Frontiers in Artificial Intelligence and Applications

HUMAN LANGUAGE TECHNOLOGIES – THE BALTIC PERSPECTIVE

Proceedings of the Ninth International Conference Baltic HLT 2020

Edited by
Andrius Utka
Jurgita Vaičenonienė
Jolanta Kovalevskaitė
Danguolė Kalinauskaitė

IOS Press

HUMAN LANGUAGE TECHNOLOGIES – THE BALTIC PERSPECTIVE

Human language technology is the study of the methods by which computer programs or electronic devices can analyze, produce, modify or respond to human texts and speech. It consists of natural language processing and computational linguistics on the one hand, and speech technology on the other.

This book presents the proceedings of the 9th International Conference, Human Language Technologies – The Baltic Perspective (Baltic HLT 2020), organised in Kaunas, Lithuania on 22 and 23 September 2020. This biennial conference offers researchers a platform to share knowledge on recent advances in human language processing for the Baltic languages, as well as promoting interdisciplinary and international cooperation in human language-technology research within and beyond the Baltic States. In addition to the traditional topics of natural language processing and language technologies, this year's conference featured a special session on resource and tool development for teaching and learning the less resourced Baltic languages. This year, 42 submissions were received, each of which was evaluated by two reviewers, resulting in a total of 34 papers being accepted for presentation and publication. The book is divided into four sections: speech and text analysis (9 papers); machine translation and natural understanding (6 papers); tools and resources (14 papers); and language learning resources (5 papers).

Providing a fascinating overview of current research in the field from a primarily Baltic perspective, the book will be of interest to all those whose work involves human language technology.

9 781643 681160

ISBN 978-1-64368-116-0 (print)
ISBN 978-1-64368-117-7 (online)
ISSN 0922-6389 (print)
ISSN 1879-8314 (online)

HUMAN LANGUAGE TECHNOLOGIES – THE BALTIC PERSPECTIVE

Frontiers in Artificial Intelligence and Applications

The book series Frontiers in Artificial Intelligence and Applications (FAIA) covers all aspects of theoretical and applied Artificial Intelligence research in the form of monographs, selected doctoral dissertations, handbooks and proceedings volumes. The FAIA series contains several sub-series, including 'Information Modelling and Knowledge Bases' and 'Knowledge-Based Intelligent Engineering Systems'. It also includes the biennial European Conference on Artificial Intelligence (ECAI) proceedings volumes, and other EurAI (European Association for Artificial Intelligence, formerly ECCAI) sponsored publications. The series has become a highly visible platform for the publication and dissemination of original research in this field. Volumes are selected for inclusion by an international editorial board of well-known scholars in the field of AI. All contributions to the volumes in the series have been peer reviewed.
The FAIA series is indexed in ACM Digital Library; DBLP; EI Compendex; Google Scholar; Scopus; Web of Science: Conference Proceedings Citation Index – Science (CPCI-S) and Book Citation Index – Science (BKCI-S); Zentralblatt MATH.

Volume 328

Recently published in this series

Vol. 327. H. Fujita, A. Selamat and S. Omatu (Eds.), Knowledge Innovation Through Intelligent Software Methodologies, Tools and Techniques – Proceedings of the 19th International Conference on New Trends in Intelligent Software Methodologies, Tools and Techniques (SoMeT_20)
Vol. 326. H. Prakken, S. Bistarelli, F. Santini and C. Taticchi (Eds.), Computational Models of Argument – Proceedings of COMMA 2020
Vol. 325. G. De Giacomo, A. Catala, B. Dilkina, M. Milano, S. Barro, A. Bugarín and J. Lang (Eds.), ECAI 2020 – 24th European Conference on Artificial Intelligence – 29 August–8 September 2020, Santiago de Compostela, Spain
Vol. 324. C. Troussas and C. Sgouropoulou, Innovative Trends in Personalized Software Engineering and Information Systems – The Case of Intelligent and Adaptive E-learning Systems
Vol. 323. L.C. Jain, X. Zhao, V.E. Balas and F. Shi (Eds.), Information Technology and Intelligent Transportation Systems
Vol. 322. M. Araszkiewicz and V. Rodríguez-Doncel (Eds.), Legal Knowledge and Information Systems – JURIX 2019: The Thirty-second Annual Conference
Vol. 321. A. Dahanayake, J. Huiskonen, Y. Kiyoki, B. Thalheim, H. Jaakkola and N. Yoshida (Eds.), Information Modelling and Knowledge Bases XXXI

ISSN 0922-6389 (print)
ISSN 1879-8314 (online)

Human Language Technologies – The Baltic Perspective

Proceedings of the Ninth International Conference Baltic HLT 2020

Edited by

Andrius Utka

Centre of Computational Linguistics
Vytautas Magnus University, Kaunas, Lithuania

Jurgita Vaičenonienė

CLARIN-LT Centre
Vytautas Magnus University, Kaunas, Lithuania

Jolanta Kovalevskaitė

Centre of Computational Linguistics
Vytautas Magnus University, Kaunas, Lithuania

and

Danguolė Kalinauskaitė

Baltic Institute of Advanced Technology, Vilnius, Lithuania
Vytautas Magnus University, Kaunas, Lithuania

IOS
Press

Amsterdam • Berlin • Washington, DC

ISBN 978-1-64368-116-0 (print)
ISBN 978-1-64368-117-7 (online)
Library of Congress Control Number: 2020945589
doi: 10.3233/FAIA328

Publisher
IOS Press BV
Nieuwe Hemweg 6B
1013 BG Amsterdam
Netherlands
fax: +31 20 687 0019
e-mail: order@iospress.nl

For book sales in the USA and Canada:
IOS Press, Inc.
6751 Tepper Drive
Clifton, VA 20124
USA
Tel.: +1 703 830 6300
Fax: +1 703 830 2300
sales@iospress.com

Preface

It is our great pleasure to introduce the Proceedings of the 9th International Conference "Human Language Technologies – the Baltic Perspective" (Baltic HLT 2020), organized by the Centre of Computational Linguistics and the CLARIN-LT centre at Vytautas Magnus University on September 22–23, in Kaunas, Lithuania. This year's conference was entirely virtual for the first time.

This biennial conference, first organized in 2004, offers researchers a space to share knowledge on recent advances in human language processing for the Baltic languages, as well as promoting interdisciplinary and international cooperation in human language-technology research within and beyond the Baltic states.

In addition to the traditional topics of natural language processing and language technologies, this year's conference featured a special session on resource and tool development for teaching and learning the less resourced Baltic languages. The keynote talk for this session, given by Elena Volodina (University of Gothenburg, Sweden), served as a good basis for sharing experiences and discussing ideas for further resource development and the application of NLP in language teaching and learning. The talk from keynote speaker Jan Rybicki (Jagiellonian University in Kraków, Poland) offered an inspiration for the growing community of Digital Humanities, whereas the keynote speaker Daniel Zeman (Charles University in Prague, Czech Republic) discussed the current state of Universal Dependencies – a community effort to define cross-linguistically applicable annotation guidelines for morphology and syntax.

We received 42 submissions this year, each of which was evaluated by two reviewers. We would like to take this opportunity to express our gratitude to the members of the Programme Committee, who worked hard to provide insightful comments. Thirty-four papers were accepted for presentation and publication. Papers in this volume cover speech and text analysis (9 papers), machine translation and natural language understanding (6 papers), tools and resources (14 papers) and language learning resources
(5 papers).

We would also like to express our gratitude to the Research Council of Lithuania for funding the conference, Vytautas Magnus University for hosting the event, the European Language Grid for organizing the pre-conference event, the Organizing Committee, our keynote speakers and all participants who, despite all the constraints, attended the virtual conference and contributed to its success.

Andrius Utka
Jurgita Vaičenonienė
Jolanta Kovalevskaitė
Danguolė Kalinauskaitė

This page intentionally left blank

Conference Organisation

The Ninth International Conference

HUMAN LANGUAGE TECHNOLOGIES – THE BALTIC PERSPECTIVE

Kaunas, Lithuania, September 22-23, 2020

Programme Committee

Andrius Utka (Chair), Vytautas Magnus University, Lithuania
Darius Amilevičius, Vytautas Magnus University, Lithuania
Ilze Auziņa, University of Latvia, Latvia
Eduard Barbu, University of Tartu, Estonia
Guntis Bārzdiņš, University of Latvia
Loïc Boizou, Vytautas Magnus University, Lithuania
Rita Butkienė, Kaunas University of Technology, Lithuania
Walter Daelemans, University of Antwerp, Belgium
Mark Fishel, University of Tartu, Estonia
Normunds Grūzītis, University of Latvia, Latvia
Heiki-Jaan Kaalep, University of Tartu, Estonia
Jelena Kallas, Institute of the Estonian Language, Estonia
Tomas Krilavičius, Vytautas Magnus University, Lithuania
Einar Meister, Tallinn University of Technology, Estonia
Kadri Muischnek, University of Tartu, Estonia
Haldur Õim, University of Tartu, Estonia
Heili Orav, University of Tartu, Estonia
Petya Osenova, Sofia University "St. Kliment Ohridski", Bulgaria
Rūta Petrauskaitė, Vytautas Magnus University, Lithuania
Gailius Raškinis, Vytautas Magnus University, Lithuania
Eiríkur Rögnvaldsson, University of Iceland, Iceland
Jūratė Ruzaitė, Vytautas Magnus University, Lithuania
Gerold Schneider, University of Zurich, Switzerland
Kiril Simov, Bulgarian Academy of Sciences, Bulgaria
Inguna Skadiņa, IMCS/Tilde, Latvia
Raivis Skadiņš, Tilde, Latvia
Kadri Vider, University of Tartu, Estonia
Martin Volk, University of Zurich, Switzerland
Elena Volodina, University of Gothenburg, Sweden
Egon W. Stemle, Institute for Applied Linguistics/Eurac Research (EURAC), Italy
Daniel Zeman, Charles University, Czech Republic
Heike Zinsmeister, University of Hamburg, Germany

Organising Committee

Andrius Utka (Chair), Vytautas Magnus University, Lithuania
Agnė Bielinskienė, Vytautas Magnus University, Lithuania
Danguolė Kalinauskaitė, Vytautas Magnus University, Lithuania
Jolanta Kovalevskaitė, Vytautas Magnus University, Lithuania
Mindaugas Petkevičius, Vytautas Magnus University, Lithuania
Erika Rimkutė, Vytautas Magnus University, Lithuania
Jurgita Vaičenonienė, Vytautas Magnus University, Lithuania

Organised by

Vytautas Magnus University

and

CLARIN-LT centre

Supported by

Research Council of Lithuania

Partner

Institute of the Lithuanian Language

This page intentionally left blank

Contents

Language Learning Resources

This page intentionally left blank

Speech and Text Analysis

This page intentionally left blank

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200595

A Study in Estonian Pronominal Coreference Resolution

Eduard BARBU [a,1], Kadri MUISCHNEK [a,b] and Linda FREIENTHAL [b]
[a] *Institute of Computer Science, University of Tartu, Estonia*
[b] *Institute of Estonian and General Linguistics, University of Tartu, Estonia*

Abstract. The first study for Estonian pronominal coreference resolution using machine learning is presented. Appropriate machine learning algorithms and techniques for balancing the data are tested on a human-annotated corpus. The results are encouraging, showing an F-score comparable with the results obtained for English before the advent of deep neural networks.

Keywords. Pronominal coreference resolution, machine learning, low resource language

1. Introduction

The pronominal coreference resolution [1] is the task of automatically finding the correct reference for a pronoun. The task is hard because the syntactic and semantic information is not enough to solve it. It constitutes the backbone of the Winograd Schema Challenge [2], a machine test intelligence that improves on the Turing Test. Given a text, for example: ”The trophy would not fit in the brown suitcase because it was too big.” a machine should answer a question like: What was too big: the trophy or the suitcase? The answer to this question amounts to solving the coreference between the pronoun ”*it*” and the noun phrase. This example shows that a pronominal coreference resolution system needs world knowledge: it needs to know that the suitcases are containers and therefore the pronoun *it* should be resolved to the noun phrase ”*the suitcase*”.

In this paper, the first machine learning study in Estonian automatic pronominal coreference resolution is presented. Appropriate machine learning algorithms and techniques intended to solve the imbalanced data problems are tested for a manually annotated pronominal coreference corpus.

Automatically resolving the coreference in Estonian is more complicated than in English. Unlike English, Estonian has no gender. Gender is a crucial feature that helps pronominal coreference resolution systems discriminate against the coreference pairs with gender agreement. For example, in English (John, **he**) could be a coreference pair, but (Ana, **he**) is not. The amount of annotated data for Estonian is much less than in English. Finally, the external knowledge than can be incorporated in a coreference system, shown to increase the performance significantly [3], is limited. The Estonian language

[1] Corresponding Author: Eduard Barbu; E-mail: eduard.barbu@ut.ee.

can only rely on the Estonian WordNet while English has a vast pool of ontologies and lexical resources.

The paper has the following structure. The next section places this study in the context of global research about automatic coreference resolution. Section 3 describes the manually annotated coreference corpus. Section 4 shows the features of the system and the machine learning algorithms tested. Section 5 presents and discusses the results. The paper ends with the conclusions.

2. Related Work

Nowadays, the best automatic pronominal coreference resolution systems are based on deep neural networks and incorporated world knowledge. Clark and Manning [4] proposed a coreference resolution algorithm that uses features defined over clusters of mentions. A two-layer model for pronoun coreference resolution leveraging the context and external knowledge is presented in a state of the art pronominal coreference system for English [3]. In particular, the authors use English Wikipedia to learn the distribution of the selectional preference of the verbs appearing in their corpus. For other languages than English a relevant study is one for Polish coreference resolution [5] that explores several deep learning architectures. For German [6], Tuggener proposes an incremental discourse processing algorithm that can address issues caused by the underspecification of mentions.

As for Baltic languages, Žitkus et al. [7] present a rule-based method for anaphora resolution in Lithuanian in the context of processing e-health records. In the same paper they provide a thorough overview of coreference/anaphora resolution in Balto-Slavic languages. Znotiņš and Paikens [8] developed a rule-based coreference resolution system for Latvian. It relies on morpho-syntactic information as well as Named Entities identification.

In Estonian, the pronominal coreference resolution was studied by Mutso [9], who adapted Mitkov's knowledge low rule-based approach [10], and Puolakainen [11] who employed Constraint Grammar [12] rules for solving the referents to pronouns. Unfortunately, neither of these experiments can be reproduced.

3. Corpus

The annotated coreference corpus used in the experiments is called EstAnaphora[2]. It contains texts from Estonian newspapers, magazines and a scientific journal spanning the years 1998 to 2007. The size of the corpus is ca 253,000 words. The following pronouns are annotated for coreference information:

- personal pronouns;
- demonstrative pronoun *see* 'it, that';
- relative pronouns *kes* 'who' and *mis* 'what'.

[2] https://github.com/EstSyntax/EstAnaphora

Each corpus file was annotated manually by two annotators, using the brat annotation tool[3]. A judge, helped by two linguists for the problematic cases, compared the annotations and provided the definitive version. For our experiments, the corpus annotations were converted to the CONLL-U format[4] where the coreference information is presented on the 10th (miscellaneous) field.

EstAnaphora contains 7,250 nominal coreference pairs, that is pairs which contain one of the above mentioned pronouns and a referent, which is a common noun, a proper noun or another pronoun, with the following distribution:

- 4,268 pairs in which a pronoun refers to a common noun;
- 2,721 pairs in which the pronoun refers to a proper noun;
- 261 pairs in which the pronoun refers to another pronoun.

Moreover, in 6,577 cases, the pronoun refers to a single referent, and in 289 cases, the pronoun refers to more than one antecedent. A case when the personal pronoun refers to more than one referent is illustrated in the following example: "**John** and **Mary** claimed that **they** are not guilty".

Figure 1 shows the percent of the pronoun referents found in a context window around the sentence containing the pronoun. Approximately 90 percent of the referent occurrences are found in a two sentences window to the left of the pronoun sentence.

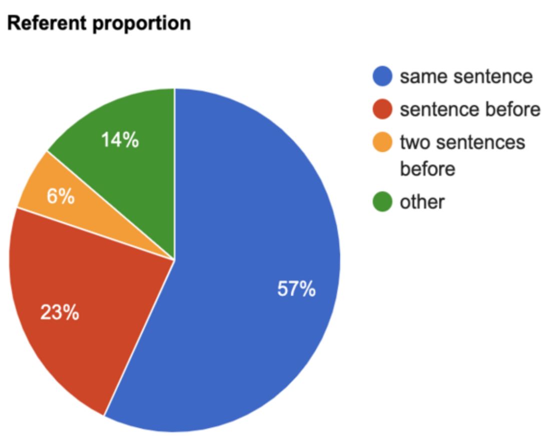


Figure 1. The percent of the referents found in the sentences in the immediate context of the sentence containing the pronoun

These figures help to set the appropriate context window for searching for candidate referents. There is a trade-off between the context window length and the algorithm performance: the wider the context window is, the more candidate pairs are generated and the less accurate the algorithm is. Vice versa, if the context is too narrow, several correct referents will be missed.

[3]https://brat.nlplab.org
[4]https://universaldependencies.org/format.html

4. Machine Learning

The coreference resolution uses the mention-pair model (for other models employed in coreference resolution see for example [13]), which is formulated as a binary classification problem. A machine learning algorithm trained on negative and positive coreference pairs learns to classify unseen coreference pairs. The coreference resolution, like fraud detection, is an imbalanced classification problem meaning that the number of positive samples is much lower than the number of negative samples. In our corpus, the proportion of positive to negative examples is roughly 1 to 24. When applied to a test set that has the same proportion of positive to negative classes, a classifier might yield an optimistic accuracy estimate. The classifier might assign every single test case to the majority class, thereby achieving an accuracy equal to the proportion of test cases belonging to the majority class.

To mitigate this known problem, techniques for dealing with the imbalanced data have been explored. As the positive class appears infrequently, extra weight has been added to it. Moreover, the standard techniques for balancing the data set (the negative class has been undersampled, and the positive class oversampled) have also been tested. The well known SMOTE (Synthetic Minority Over-sampling Technique) algorithm generates new training data for the positive class considering the k-nearest neighbors of the positive example [14]. An advanced balancing technique called Adaptive Synthetic Sampling Method for Imbalanced Data, known as ADASYN, was also tested [15]. ADASYN weighs the positive class examples based on the level of difficulty in learning. Hence more synthetic data is generated for harder to learn positive class examples. The last technique tried is One-Class SVM [16]. This algorithm is trained only on negative examples to learn the boundaries of the negative points. Any points that lie outside the boundaries are considered outliers (e.g., they correspond to the positive data examples).

4.1. Features

Four kinds of features are computed for the generated coreference pairs: distance features, morphological, syntactic, and semantic features. The distance features encode the distance between the pronoun and the referent, as well as the position of the referent in the sentence. Examples of distance features are :

- **Distance pronoun-referent**. The feature encodes the distance between the sentence of the pronoun and the sentence of the referent. If the pronoun and the referent are in the same sentence, the distance is 0.
- **Distance in nouns**. The feature counts the number of nouns separating the pronoun and the referent.
- **Referent position** gives the position of the referent in the sentence. The position can be one of the values: beginning, middle, or end.

The morphological features encode the morphological information found in a context window around the referent and the pronoun. Examples of morphological features are:

- **POS referent/pronoun**. These features encode the part of speech (POS) tag of the referent and the pronoun.
- **POS before referent/pronoun**. Theses features give the POS tag of the word found 1, 2 or 3 positions before the referent or the pronoun.

- **POS after referent/pronoun**. Theses features give the POS tag of the word found 1, 2 or 3 positions after the referent or the pronoun.

The syntactic features encode syntactic information about the coreference pairs. Examples of syntactic features are:

- **Syntactic function referent/pronoun**. The features encode the syntactic functions of the referent and the pronoun.
- **POS head referent/pronoun**. The features encode the POS tag of the syntactic heads of the referent and the pronoun.

The semantic features encode the cosine similarity scores between the embeddings corresponding to the pronouns and referents. The embeddings are trained with word2vec on the Estonian Reference Corpus [17]. For this study, 29 features have been implemented.

4.2. Algorithms

The machine learning algorithms were selected based on three criteria: resistance to data unbalancing, boundary type (linearly separable or not), and performance.

1. **Decision trees** (DT). The advantage of the decision tree algorithms is that humans can interpret their output. It is also known that they are resistant to imbalanced data because they have an inductive bias towards axis-aligned bounding boxes.
2. **Logistic regression** (LR). The Logistic Regression works particularly well when the features are linearly separable. The classifier is robust to noise, avoids overfitting, and its output can be interpreted as probability scores.
3. **K-Nearest Neighbors** (knn). This algorithm classifies a new instance based on the distance it has to k instances in the training set. The prediction output is the label that classifies the majority. Because it is a non-parametric method, it gives good results in classification problems where the decision boundary is irregular.
4. **XGBoost** is a widely used, high-performance machine learning algorithm from the tree boosting family [3]. It has won numerous Kaggle competitions, thus showing a state of the art performance in many tasks.

4.3. Experiment

The automatic coreference resolution experiment follows three steps.

1. **Candidate coreference-pair generation**. The coreference pairs between nominals and pronouns are generated. The generation algorithm allows the specification of several parameters, like the window context for each pronoun. In order to choose the best configuration, runs with different parameter values have been performed.
2. **Training**. The coreference pairs labeled in the corpus are assigned to the positive class. The rest of the coreference pairs generated based on the parameters above are assigned to the negative class. The features are computed for the training set, the machine learning algorithms are trained, and the trained model is stored.
3. **Testing**. The test set coreference pairs and their features are generated. The trained models are loaded and the test coreference pairs are assigned to the positive and negative classes by the machine learning algorithms.

Table 1. The results of the machine learning algorithms

Classifier	Parameters	Balanced	F1 score
DT	default	no	0.49
XGBoost	**default**	**no**	**0.60**
knn	neighbors =3	no	0.39
LR	solver='lbfgs', max_iter=4000	no	0.40
LR 1	solver='lbfgs', max_iter=4000, class_weight={0: 1, 1: 5}	no	0.46
LR 2		undersampling threshold 0.5	0.37
LR 3		ADASYN	0.31
XGBoost 1	**class weight={0: 1, 1: 5}**	**no**	**0.60**
XGBoost 2		undersampling threshold 0.5	0.44
XGBoost 3		ADASYN	0.51
DC		no	0.04
BC		no	0.32

5. Results and Discussions

The experiment is performed with the scikit-learn toolkit. There are two baselines. The first baseline is a weak one (abbreviated DC in the table), implemented by a dummy classifier that generates predictions according to the positive and negative class distribution in the training set. The second baseline (abbreviated BC) is a competitive baseline that resolves the mention to the closest pronoun.

Though all classifiers have been run in multiple configurations, only the best results are reported. The One Class SVM, for example, had a very low performance and we have excluded it from the analysis.

The results reported in Table 1 are for 4-fold stratified cross-validation on the annotated corpus. The parameters column gives the value of the hyperparameters for the classifiers. The Balanced column stipulates if the training set is balanced or not. There are three configurations of the Logistic Regression and XGBoost algorithms, each one with a different technique that treats the imbalanced data. LR 1 and XGBoost 1 is a configuration where the positive class receives five times more weight than the negative class. LR 2 and XGBoost 2 is a configuration where the dataset is balanced by random undersampling. LR 3 and XGBoost 3 is a configuration where the dataset is balanced by ADASYN.

The best results are obtained by the XGBoost algorithm in two configurations marked in bold in Table 1. The techniques to deal with imbalance data seem to be detrimental to the algorithm performance. However, more experiments should be performed to reach a definitive conclusion.

The Logistic Regression performance increases 6 points when we weigh the positive class 5 times more than the negative class, but undersampling imbalanced technique and ADASYN lower the algorithm performance.

It is known that the decision trees perform relatively well with imbalanced data, so the score attained slightly behind XGBoost 2 result is no surprise.

As expected, the weak baseline performs a little better than chance. The BC baseline, a competitive baseline, is soundly beaten on the test set by all machine learning algorithms, including the nonparametric lazy learning knn.

6. Conclusion and Future Work

In this paper, the first machine learning coreference study for the Estonian language has been presented. The results obtained are encouraging though they are not yet comparable to the state of the art results for the English language. The best results obtained by the best classifier XGBoost are in the same range as the results obtained for the English language by the knowledge poor systems before the advent of deep neural network revolution.

In the introduction of the paper, we have given three reasons why this might be the case. We believe that annotating more data based on the error analysis will substantially improve the results. Moreover, new, linguistically motivated features will be devised in the next version.

More importantly, we will explore advanced algorithms based on deep neural networks, which are state-of-the-art English language. There are some preliminary experiments performed by one of the authors of this paper [18]. However, for the features calculated by the system, it seems that the neural network architecture tested is not better than the XGBoost algorithm.

The incorporation of semantic information from the Estonian WordNet might improve the performance of the coreference system. However, the fact that Estonian is a low resource language and lacks the grammatical category of gender are severe limitations placed on any automatic coreference resolution system. The Estonian coreference system can be accessed from the following Github repository [5].

7. Acknowledgments

We would like to thank the annotators who have annotated the EstAnaphora coreference corpus.

This study was supported by the Estonian Language technology grant "Estonian Universal Syntax: Resources and Applications" and by the Centre of Excellence in Estonian Studies (CEES, European Regional Development Fund).

References

[1] Hobbs JR. Resolving pronoun references. Lingua 44. 1978:311–338.

[2] Levesque HJ, Davis E, Morgenstern L. The Winograd Schema Challenge. In: Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning. KR'12. Rome, Italy: AAAI Press; 2012. p. 552–561. Available from: https://cs.nyu.edu/faculty/davise/papers/WSKR2012.pdf.

[5] https://github.com/SoimulPatriei/EstonianCoreferenceSystem

[3] Zhang H, Song Y, Song Y. Incorporating Context and External Knowledge for Pronoun Coreference Resolution. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics; 2019. p. 872–881. Available from: https://www.aclweb.org/anthology/N19-1093.

[4] Clark K, Manning CD. Improving Coreference Resolution by Learning Entity-Level Distributed Representations. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Berlin, Germany: Association for Computational Linguistics; 2016. p. 643–653. Available from: https://www.aclweb.org/anthology/P16-1061.

[5] Nitoń B, Morawiecki P, Ogrodniczuk M. Deep Neural Networks for Coreference Resolution for Polish. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). Miyazaki, Japan: European Language Resources Association (ELRA); 2018. Available from: https://www.aclweb.org/anthology/L18-1060.

[6] Tuggener D. Incremental Coreference Resolution for German. University of Zurich, Faculty of Arts; 2016.

[7] Žitkus V, Butkienė R, Butleris R, Maskeliunas R, Damasevicius R, Woźniak M. Minimalistic Approach to Coreference Resolution in Lithuanian Medical Records. Computational and Mathematical Methods in Medicine. 2019 03;2019:1–14.

[8] Znotiņš A, Paikens P. Coreference Resolution for Latvian. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14). Reykjavik, Iceland: European Language Resources Association (ELRA); 2014. p. 3209–3213. Available from: http://www.lrec-conf.org/proceedings/lrec2014/pdf/729$_{paper.pdf}$.

[9] Mutso P. Knowledge-poor anaphora Resolution System for Estonian. Tartu Ülikool; 2008.

[10] Mitkov R. Robust pronoun resolution with limited knowledge. In: Proceedings of the 18th International Conference on Computational Linguistics (COLING'98)/ACL'98 Conference; 1998. p. 869–875.

[11] Puolakainen T. Anaphora resolution experiment with CG rules. In: Proceedings of the Workshop on "Constraint Grammar - methods, tools and applications" at NODALIDA 2015, May 11-13, Vilnius, Lithuania; 2015. p. 35–38.

[12] Karlsson F, Voutilainen A, Heikkilä J, Anttila A. Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text; 1995.

[13] Ng V. Machine Learning for Entity Coreference Resolution: A Retrospective Look at Two Decades of Research. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence. AAAI'17. AAAI Press; 2017. p. 4877–4884.

[14] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: Synthetic Minority over-Sampling Technique. J Artif Int Res. 2002 Jun;16(1):321–357.

[15] He H, Bai Y, Garcia EA, Li S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: IN: IEEE INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS (IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE), IJCNN 2008; 2008. p. 1322–1328.

[16] Zhang R, Zhang S, Muthuraman S, Jiang J. One Class Support Vector Machine for Anomaly Detection in the Communication Network Performance Data. In: Proceedings of the 5th Conference on Applied Electromagnetics, Wireless and Optical Communications. ELECTROSCIENCE'07. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS); 2007. p. 31–37.

[17] Kaalep HJ, Muischnek K, Uiboaed K, Veskis K. The Estonian Reference Corpus: Its Composition and Morphology-aware User Interface. In: Proceedings of the 2010 Conference on Human Language Technologies – The Baltic Perspective: Proceedings of the Fourth International Conference Baltic HLT 2010. Amsterdam, The Netherlands, The Netherlands: IOS Press; 2010. p. 143–146. Available from: http://dl.acm.org/citation.cfm?id=1860924.1860949.

[18] Freienthal L. Pronominaalsete viitesuhete automaatne lahendamine eesti keeles närvivõrkude abil. Tartu Ülikool; 2020.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200596

Structural Models of Lithuanian Plosive Consonants in Different Word Positions

Sigita DEREŠKEVIČIŪTĖ[1], Asta KAZLAUSKIENĖ
Department of Lithuanian Studies, Vytautas Magnus University, Lithuania

Abstract. This study examines the structural models of Lithuanian plosive consonants in intervocalic, word-initial and word-final positions. The research material consists of 24 sentences read three times by 6 native speakers. The results show that the plosive consonants can be composed of one to three phases, and the most frequent and common models are the closure with a burst release, which might be followed by a different degree of frication.

Keywords. Lithuanian, plosive consonant, structural model, voiced consonant, voiceless consonant

1. Introduction

It is well known that the traditional structure of plosive consonants consists of three stages in their production, involving a period of closure in the oral cavity (and of the velopharyngeal port), a burst or transient, which is generated by the sudden emission of intraoral pressure (upon constriction release), and aspiration phase in voiceless plosive consonants, which is created by noise turbulence at the glottis (and the fourth phase if uttered before a vowel – the transition into it) ([1], [2], [3], [4], [5] among others). From the acoustic point of view, a plosive consonant is a sequence of silence (a quiescent waveform) followed by a transient, i.e. a brief fricative-like noise segment and a random waveform, i.e. aspiration. However, “in between the transient and the aspiration, there is likely to be a brief interval of frication created in the mouth or at the lips as the lower articulator drags away from the upper” [6: 111]. Since both waveforms are random, normally these two phases are not distinguished from each other, rather “the term aspiration is used informally to cover the entire non-quiescent phase of a voiceless plosive” [6: 111]. Moreover, it is also known that the release stage might be modified so that no audible release occurs. This can happen “through overlap where the release occurs but is inaudible and through suppression of audible release” [7: 69]. Therefore, the aim of this study is to investigate the structure of these acoustic events and the possible modifications of the phases in greater detail. We expect that the results could be useful for developing a set of rules for automatic recognition systems. Additionally, the models are intended to be used for the future comparison of structural nature of consonants in different environments and their phonetic realizations in coarticulatory processes or consonantal clusters.

[1] Corresponding Author: Sigita Dereškevičiūtė; Vytautas Magnus University, V. Putvinskio st. 23-203, 44243 Kaunas, Lithuania; E-mail: sigita.dereskeviciute@vdu.lt.

The acoustic features of Lithuanian consonants have already been analyzed ([8], [9], [10], [11], [12], [13]); however, the structural models demonstrating the acoustic sequences in detail have not received considerable scholars' attention and have not been described. Therefore, it is interesting to find out whether the models in the Lithuanian language follow the classical patterns, or some language-specific acoustic events may occur.

2. Material and Method

This paper presents the results of production experiment which examines the acoustic structural models of Lithuanian plosive consonants in intervocalic, word-initial and word-final positions. The segments with the consonants investigated were extracted using the automatic tools created at Vytautas Magnus University; the models were defined and classified manually based on the changes in sound waveform and spectrograms.

The research material consists of 24 sentences where the consonants under the present investigation [p], [b], [t], [d], [k] and [g] appeared mostly in [a] vowel context and different positions of a word: 6 sentences were composed of the target consonants in word- and phrase-initial position, 3 sentences in word- and phrase-final position and 12 sentences in intervocalic position. The sentences were read three times by 6 native speakers (3 men and 3 women aged 24-55; all of them have higher education and different public speaking skills). In total, 432 tokens were examined. The examples are the following: [p] in [kɐ^{2}ˈpɐiˑ] (*graveyard*), [^{2}ˈpɑːdɐs] (*a sole*), [ʋɐkɐ^{1}ˈroːp] (*towards evening*); [t] in [^{2}ˈrɑːtɐs] (*a wheel*), [^{2}ˈtɑːkɐs] (*a path*), [pɐ^{1}ˈgɑˑʊt] (*to catch*); [k] in [^{2}ˈbɑːkɐs] (*a tank*), [^{2}ˈkɑːboː] (*hanging*), [pɐ^{1}ˈgɑˑʊk] (*catch*); [b] in [^{2}ˈstɑːboː] (*the idol's*), [^{2}ˈbɑːdoː] (*hunger's*); [d] in [bɐ^{2}ˈdɑːʋoː] (*starved*), [^{2}ˈdɑːroː] (*does*, *makes*); [g] in [rɐ^{1}ˈgɑˑʊtoː] (*tasted*), [^{2}ˈgɑːʋoː] (*received*).

The structural models and the duration of the consonants investigated were measured using the PRAAT software [14].

The models were defined with the labels CL – meaning a silent (or quasi-silent) closure, PW – a closure with a periodic (or quasi-periodic) waveform, PL – plosion (or transient) and FR – frication.

3. Results and Discussion

The experimental results have revealed 5 structural models, which could be generalized as CL/PW+(PL)+(FR/CL), where the first phase may be either a complete closure (CL) and, depending on the place of articulation of the voiceless consonants, may last 30-90 ms or a prevoicing interval (PW) and, depending on the place of articulation of the voiced consonants, may be characterized by 25-80 ms duration. The second possible stage is a plosion interval (PL) which, depending on the place of articulation of the voiceless and voiced consonants, may last 5-30 ms. The velar consonants [k, g] may consist of up to three bursts. Lastly, the first phase or the release stage may be accompanied by a frication period (FR)[2] of 10-20 ms or the second brief silent portion (CL, about 10-15 ms).

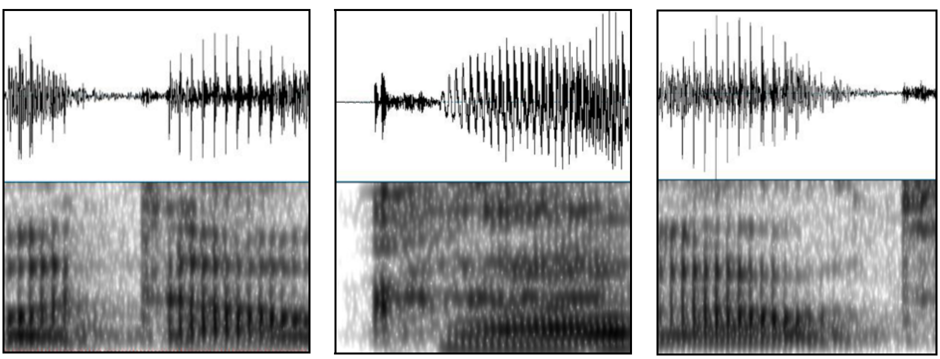
The labelling of the second/third segment as a silence closure (CL) may be a subject of discussion as, first, it may be articulatory impossible to create a complete closure after the burst release before the vowel; second, if there are perturbations in the waveform, the segment should be defined as FR. However, a decision was made to introduce the second closure portion in this experiment in order to highlight the difference between the segments with more attenuated and more intense amplitudes.

The plosive consonants are characterized by 5 models in the intervocalic position (see Table 1 and examples in Figures 1a-5a). The first phase CL is common to all voiceless consonants, while PW is common to voiced consonants. The models with plosion (CL/PW+PL+FR/CL) are relatively common (79 %). The models with an unreleased burst are less frequent (a single closure phase or a closure occurring with frication, 21 %).

Table 1. The structural models of Lithuanian plosive consonants and their distribution (%)[3]

Structural model	VCV						#CV						VC#		
	k	**t**	**p**	**g**	**d**	**b**	**k**	**t**	**p**	**g**	**d**	**b**	**k**	**t**	**p**
CL/PW+PL+FR	10	45	20	15	30		30	50	10	15	20		60	90	20
CL/PW+PL	25	30	50	55	50	60	10	30	50	80	80	80	20		65
CL/(PW)+PL+CL	50	15	15		5		60	20	40						
CL/PW+FR	5	10	5		5					5			20	10	15
CL/PW	10		10	30	10	40						20			

The structural model CL/PW+PL (see Figure 2a) is predominant (45 % of the consonants in the intervocalic position). The ternary models CL/PW+PL+FR, CL/PW+PL+CL and the monomial model CL/PW are distributed very similarly (20 %, 14 % and 17 %, respectively). The models are illustrated in Figures 1a, 3a, and 5. The structural model CL/PW+FR is rare (4 %) in the intervocalic position (see Figure 4a).



a) [d] in [^{2}ˈbɐdɑːʊoː] (*starved*) b) [k] in [^{2}ˈkɑːboː](*hanging*) c) [t] in [^{1}ˈpɐgɑˑʊt] (*to catch*)

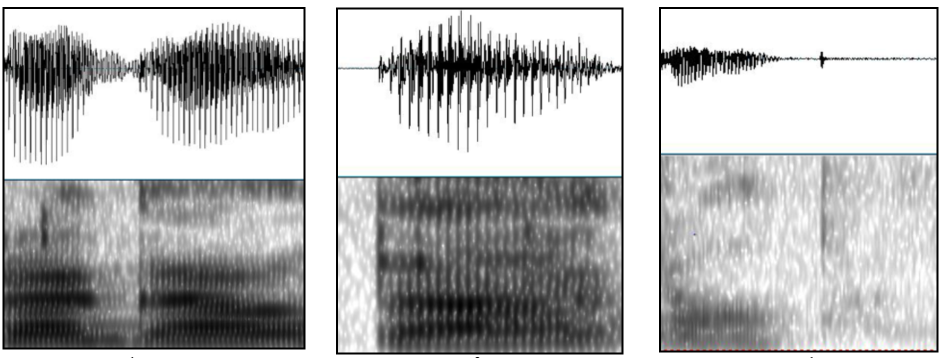
Figure 1. The structural model CL/PW+PL+FR in: a) VCV, b) #CV, c) VC#

[2] In this study, the interval between the transient and a puff of air that follows it is referred to a different degree of frication and is not regarded as aspiration. The released breath is probably too short in duration to perceive it as aspiration. Nevertheless, no perception tests were conducted to validate the phenomenon. In the Lithuanian language, only voiceless consonants used in word-final position are considered to be aspirated, i.e. the accompanied frication noise is audible.

[3] C marks consonant, V - vowel, # - word-initial or word-final position.

Almost all 5 models were found in the realization of voiceless consonants and voiced [d], whereas voiced [b] and [g] are less variable. Almost half of the examples of the voiced consonants are composed of two typical segments (PW+PL, 55 %), while voiceless consonants are characterized by a more complex structure: in the case of [t] and [k], before transition to the adjacent vowel, the PL phases may be accompanied by FR or CL (almost half of the examples of these consonants). An additional noise segment (FR) may appear in the production of the voiced consonants [d] (30 %) and [g] (15 %), but it never occurs in the production of the bilabial [b]. In the case of [p], CL+PL (50 %) is the most common model. The third part of voiced plosive consonants tend to be produced with an unreleased burst (PW, 26 %).

In the word-initial CV position, 5 models have also been identified; however, in this position, the production of consonants is more invariable. None of the consonants is defined by all models. In fact, there are three typical models for all voiceless consonants, whereas the voiced ones may be characterized by four different models but only one is predominant.



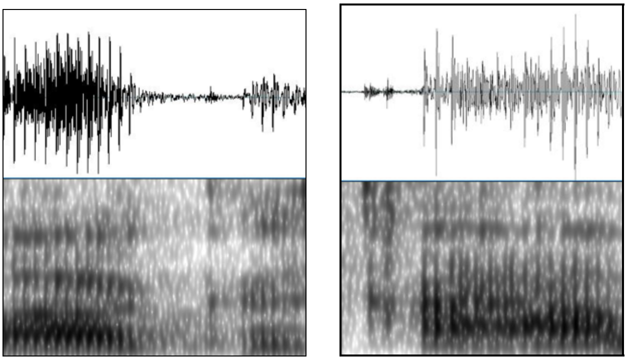
a) [g] in [[1]ˈrɐɡɑˑʊtoː] (*tasted*)　b) [t] in [[2]ˈtɑːkɐs] (*a path*)　c) [p] in [ʋɐkɐr[1]ˈoːp] (*towards evening*)

Figure 2. The structural model CL/PW+PL in: a) VCV, b) #CV

The most frequent model as in VCV sequences consists of a closure and a burst release (CL/PW+PL, 55 %, see Figure 2b), and it is the most prevailing pattern among the voiced consonants (80 %, see Table 1). A different degree of accompanying noise may follow the release phase but it is less common (CL/PW+PL+FR, 21 % and CL/PW+PL+CL, 20%, see Figures 1b and 3b, respectively). The voiceless plosives tend to have these noise segments (CL/PW+PL+FR, 30 % and CL/PW+PL+CL, 40 %, see Table 1).

Due to the laryngeal settings, the voiceless consonants do not exhibit any vibrational patterns and the silence phase of the closure merges with the prosodical pause in phrase- and word-initial positions. Therefore, it is impossible to determine the duration of the closure phase in this position. Nevertheless, the phase is not excluded in the production and, therefore, the label CL is assigned to the description of the models. The most frequent models are individual for each voiceless consonant: [k] – CL+PL+CL (60 %), [t] – CL+PL+FR (50 %) and [p] – CL+PL (50 %).

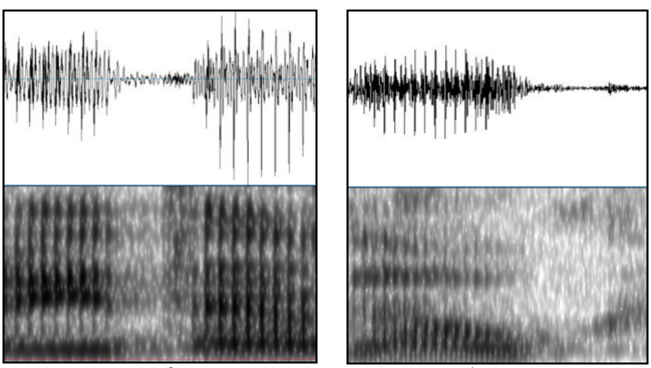
The voiced plosive consonants [b], [d] and [g] are defined by the binary model PW+PL (80 %) despite their place of articulation. The patterns where a burst release co-occurred with a CL segment were not observed. A closure accompanied by unreleased bursts is also uncommon for both voiceless and voiced consonants. It may happen only in the production of the bilabial consonant [b] (PW, 20 %) and the velar [g] (PW+FR, 5 %). However, in the latter case, it is likely to be an accidental or a speaker-specific case.



a) [k] in [^{2}ˈbɑːkɐs] (*a tank*) b) [k] in [^{2}ˈkɑːboː] (*hanging*)

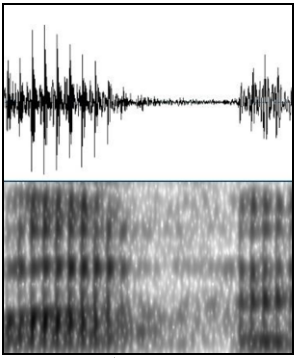
Figure 3. The structural model CL/(PW)+PL+CL in: a) VCV, b) #CV

Only voiceless consonants can be used at the end of a word in Lithuanian. The results of the experiment have shown that 3 structural models with voiceless consonants occur in word- and phrase-final position: CL+(PL)+(FR). The burst is very often followed by frication (57 % see Figure 1c), especially for [t] (90 %) and less for [k] – 60 %. Bilabial [p] is characterized by the closure with a burst release and no obvious frication (65 %, see Figure 2c). The consonants may also be produced without a strong audible plosion but rather with a hissing noise sound (CL+FR, 15 %, see Figure 4b). In word- and phrase-final position, when the articulators are getting ready to relax, the muscular tension decreases, the glottis maintains open longer, unreleased bursts could be expected. Nevertheless, the results show that the transient segment is typically present (CL+PL+(FR)), which means that consonants are rather produced with stronger buildups of pressure and/or tenser articulation.



a) [d] in [tʲɪkrɐɪ‿^{2}ˈdɑːroː] (*really does*) b) [k] in [^{1}ˈpɐgɑˑʊk] (*catch*)

Figure 4. The structural model CL/PW+FR in: a) VCV, b) VC#



[k] in [2ˈbɑːkɐs] (*a tank*)

Figure 5. The structural model CL/PW in VCV position

Considering all models regardless the position of the consonant in the word or the absence or presence of voicing, the most frequent model of plosive consonants is the binomial pattern, i.e. a closure released with a burst (46 %), confirming this structural model to be the most classical one. One third of the data has demonstrated that plosives are produced with frication (28 %). From the articulatory point of view, this means that the voicing of the following vowel (in VCV, CV sequences) starts simultaneously upon the oral release or shortly thereafter. The rarest pattern is when the closure is released not with a brief plosion but with a continuous frication interval (CL/PW+FR, 5 %). It is also possible but untypical to generate plosive consonants only by their closure (CL/PW, 8 %).

Regarding voicing despite the position of consonants in a word, the most frequent pattern is CL+PL+FR for voiceless consonants (37 %) and PW+PL for voiced consonants (68 %). As a rule, the voiceless plosive consonants are pronounced with more muscular energy and they tend to have longer and stronger pressure increases than their voiced counterparts [15], [16]. Moreover, after the voiceless plosive consonants before voiced sounds, it takes longer for the vocal folds to start vibrating [17]. Therefore, it is more frequent that during the lag between the release of obstruction and the onset of voicing, turbulent air comes through the open glottis. All these obstruent consonants probably may display stronger or weaker friction because of the articulatory gestures. Within the period of time when the articulators are getting apart and shifting to a different configuration in the vocal tract, a weaker or stronger puff of air may come along, especially in the case of voiceless consonants when the glottis is still open. Therefore, the voiceless consonants tend to be pronounced with a shorter or longer frication noise (FR) in all positions.

In contrast to voiceless consonants, the voiced plosives are relatively weaker with regard to energy, and the release of additional breath is hardly noticeable. Additionally, the energy is partly used up for the vibrational processes of the vocal folds; therefore, the pronunciation of the voiced plosives might be weaker and sometimes unreleased.

The distribution of the frequency of the structural models is irregular also in regard to the place of articulation of the plosive consonants. The dental consonants [t] and [d] tend to be the most frequently fricated (CL/PW+PL+FR) among other consonants in all positions: in the initial position [t] – 45 %, [d] – 30 %, in the medial position [t] – 50 %, [d] – 20 %, in final – [t] – 90 %. The burst accompanying the segment is also common to the velar plosive [k], but here, the plosion is followed by strongly attenuated frication acoustically similar to the phase of a complete closure (CL+PL+CL, in VCV sequences – 50 %, in CV – 60 %). The generation of a turbulent noise of a different degree may be caused by possible differences in the changes of intraoral pressure (correlated with the size of the cavity behind the supraglottal constriction) or in the speed of movements of different articulators in the production of dentals and velars. The bilabials [p] and [b], which are articulated in the most front position, are usually pronounced without a frication noise or it is very weak (PW+PL). The voiced bilabial [b], in turn, may also have no plosion (PW) at all. As Miller and Daniloff observed, the “low subglottal pressures, combined with short closure durations, might not permit the buildup of intraoral pressures sufficient for burst transients to occur at consonantal release” [18: 351]. It is also possible that the burst is released, but it overlaps with the following vowel.

4. Conclusions

In conclusion, the plosive consonants can be composed of one to three phases, or from two to three phases in the case of voiceless consonants in word final position. The most typical model consists of two phases: a closure and a burst release, which is immediately followed by the vibratory patterns of the next vowel. However, consonants require pressure buildups behind the oral closure until a rapid opening of constriction, which consequently causes the creation of a sudden brief flow of air of a different degree (FR or CL). In most cases, this reflects the production of voiceless consonants, especially articulated in the front part of the oral cavity.

The metaphorical denotation of these consonants does not always reflect their phonetic realization, and the plosive consonants are not required to have a constant number of phases as it is usually expected in Lithuanian linguistics. The next stage of this study is to investigate the structural models of plosive consonants in consonant clusters.

Acknowledgment

The research was carried out within the framework of the EU Structural Funds project "The Development of the Public Electronic Services of the Lithuanian Language Syntactic and Semantic Analysis Information System (SEMANTIKA 2)" no. 02.3.1-CPVA-V-527-01-0002, financed/funded by the European Regional Development Fund. For the help in extracting the data, we express our sincere thanks to our colleague Assoc. Prof. Gailius Raškinis from the Department of Systems’ Analysis, Vytautas Magnus University.

References

[1] Fant G. The Acoustic Theory of Speech Production. The Hague, Netherlands: Mouton; 1960. 323 p.
[2] Lass, NJ, editor. Speech and Language. Advances in Basic Research and Practice. 1st ed. vol. 11. London: Academic Press, 1984. 396 p.
[3] Lieberman P, Blumstein SE. Speech physiology, speech perception, and acoustic phonetics. Cambridge: Cambridge University Press. 1988. 249 p.

[4] Johnson K. Acoustic and auditory phonetics. 2nd ed. Hoboken, New Jersey: Wiley-Blackwell; 2003. 192 p.
[5] Stevens KN. Acoustic phonetics. Cambridge, MA: MIT Press. 1988. 624 p.
[6] Hewlett N, Beck J. An Introduction to the Science of Phonetics. New York: Routledge, 2006. 368 p.
[7] Ball MJ, Rahilly J. Phonetics. The Science of Speech. London: Arnold. 1999. 239 p.
[8] Kliukienė R. Šiaurės žemaičių tarmės konsonantizmas: akustinės ir audicinės ypatybės. Vilnius: Vilniaus universitetas. 2011. 168 p.
[9] Dereškevičiūtė S. Lietuvių kalbos priebalsių akustinės ypatybės. Doctoral Dissertation. Kaunas. 2013.
[10] Ambrazevičius R, Leskauskaitė A. Priebalsių akustinės ypatybės: palatalizacija ir balsingumas. Kaunas: Technologija. 2014. 312 p.
[11] Urbanavičienė J, Indričāne I. Lietuvių ir latvių kalbų trankieji priebalsiai: lokuso lygčių rezultatai. Baltistica. 2015; 50 (2): 261-293.
[12] Urbanavičienė J, Indričāne I. Lietuvių ir latvių kalbų uždarumos priebalsiai: sprogimo fazės trukmė ir FFT spektrai. Žmogus ir žodis. Didaktinė lingvistika, 2016; 18(1): 46-79.
[13] Jaroslavienė J, Grigorjevs J, Urbanavičienė J, Indričāne I. Baltų kalbų garsynas XXI a. pradžioje: Balsių ir garsų sąveikos instrumentinis tyrimas. Collective monograph. Vilnius: Lietuvių kalbos institutas. 2019. 317 p.
[14] Boersma P, Weenink D. 2018. Praat: doing phonetics by computer, version 6.035. www.praat.org.
[15] Sharf DJ. Duration of Post-Stress Intervocalic Stops and Preceding Vowels. Language and Speech, 1962; 5(1), 26-30.
[16] Arkebauer H, Hixon T, Hardy J. Peak Intraoral Pressures During Speech. Journal of Speech and Hearing Research. 1967; 10(2): 196-208.
[17] Speck BP, Valor MLG. A Practical Introduction to English Phonology. 2nd ed. Valencia: Universitat de Valencia, 2015. 120 p.
[18] Miller CJ, Daniloff R. Aerodynamics of Stops in Continuous Speech. Journal of Phonetics 1977; 5(4): 351–360.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200597

Evaluating Multilingual BERT for Estonian

Claudia KITTASK [1] and Kirill MILINTSEVICH and Kairit SIRTS
Institute of Computer Science, University of Tartu, Estonia

Abstract. Recently, large pre-trained language models, such as BERT, have reached state-of-the-art performance in many natural language processing tasks, but for many languages, including Estonian, BERT models are not yet available. However, there exist several multilingual BERT models that can handle multiple languages simultaneously and that have been trained also on Estonian data. In this paper, we evaluate four multilingual models—multilingual BERT, multilingual distilled BERT, XLM and XLM-RoBERTa—on several NLP tasks including POS and morphological tagging, NER and text classification. Our aim is to establish a comparison between these multilingual BERT models and the existing baseline neural models for these tasks. Our results show that multilingual BERT models can generalise well on different Estonian NLP tasks outperforming all baselines models for POS and morphological tagging and text classification, and reaching the comparable level with the best baseline for NER, with XLM-RoBERTa achieving the highest results compared with other multilingual models.

Keywords. multilingual BERT, NER, POS tagging, text classification, Estonian

1. Introduction

Large pretrained language models, also called contextual word embeddings, such as ELMo [1] or BERT [2] have been shown to improve many natural language processing tasks. Training large contextual language models is complex both in terms of the required computational resources as well as the training process and thus, the number of languages for which the pretrained models are available is still limited.

Although according to [3], language-specific BERT models are currently available for 19 languages, many more languages are supported via multi-lingual models. The aim of the multilingual models is to reduce the necessity to train language-specific models for each language separately. Experiments on various tasks, such as named entity recognition (NER) [4] or parsing pipeline tasks [5], have shown that multilingual contextual models can help to improve the performance over the baseline models not based on contextual word embeddings.

There are several multilingual models available that also include Estonian language. For instance, multilingual BERT (mBERT) [2] has been trained jointly on Wikipedia data on 104 languages, including Estonian. Estonian is also included in the cross-lingual language model (XLM-100) [6], which was trained on 100 Wikipedia languages, and

[1]Corresponding Author: Claudia Kittask; E-mail: claudiakittask@gmail.com

cross-lingual RoBERTa (XLM-RoBERTa) [4], which was trained on much larger CommonCrawl corpora and also includes 100 languages. Finally, DistilBERT [7] is a smaller version of the BERT model obtained from the BERT models via knowledge distillation, which is a compression technique where the compact model is trained to reproduce the behaviour of the larger model. The multilingual DistilBERT (DistilmBERT) has been distilled from the mBERT model featuring the same 104 Wikipedia languages.

The aim of the current work is to evaluate the existing multilingual BERT models on several NLP tasks on Estonian. In particular, we will apply the BERT models on NER, POS and morphological tagging, and text classification tasks. We compare four multilingual models—mBERT, XLM-100, XLM-RoBERTa and DistilmBERT—to find out which one of those performs the best on our Estonian tasks. We compare the results of the multilingual BERT models with task-specific baselines and show that multilingual BERT models improve the performance of the Estonian POS and morphological tagging and text classification tasks and achieve comparable results for named entity recognition. Overall, XLM-RoBERTa achieves the best results compared with other multilingual BERT models used.

2. Related Work

Although most research on multilingual BERT models has been concerned about zero-shot cross-lingual transfer [8], we are more interested in those previous works that, similar to us, evaluate multilingual BERT models in comparison to monolingual (non-English) baselines. We next review some examples of such work.

Virtanen et al. [9] evaluated multilingual BERT alongside with the monolingual Finnish BERT on several NLP tasks. In their work, multilingual BERT models outperformed monolingual baselines for text classification and NER tasks, while for POS-tagging and dependency parsing, the multilingual BERT models fell behind the previously proposed methods, most of which were utilizing monolingual contextual ELMo embeddings [1]. Baumann [10] evaluated multilingual BERT models on German NER task and found that while the multilingual BERT models outperformed two non-contextual LSTM-CRF-based baselines, it performed worse than a model utilizing monolingual contextual character-based string embeddings [11]. Kuratov et al. [12] applied multilingual BERT models on several tasks in Russian. They found that multilingual BERT outperformed non-contextual baselines for paraphrase identification and question answering and fell below a baseline for sentiment classification.

The pattern in all these works is similar: the multilingual BERT models perform better than non-neural or non-contextual neural baselines, but the multilingual BERT model is typically outperformed by approaches based on language-specific monolingual contextual comparison systems. We cannot test the second part of this observation as currently no monolingual language-specific BERT model exists for Estonian. However, we will show that the first part of this observation generally also holds for Estonian, i.e. the multilingual BERT models outperform non-contextual baselines for most of the experimental tasks used in this paper.

3. Experimental Tasks

This section describes the experimental tasks. We give also overview of the used data and the baseline models.

3.1. POS and Morphological Tagging

For POS and morphological tagging, we use the Estonian treebank from the Universal Dependencies (UD) v2.5 collection that contains annotations of lemmas, part of speech, universal morphological features, dependency heads and universal dependency labels. We train models to predict both universal POS (UPOS) and language-specific POS (XPOS) tags as well as morphological tags. We use the pre-defined train/dev/test splits for training and evaluation. Table 1 shows the statistics about the treebank splits.

Table 1. Statistics for the Estonian UD corpus

	Train	**Dev**	**Test**
Sentences	31,012	3,128	6,348
Tokens	344,646	42,722	48,491

As baselines, we report the results of Stanza [13] and UDPipe [14] obtained on the same Estonian UD v2.5 test set.

3.2. Article Type and Sentiment Classification

For text classification, we use the Estonian Valence corpus [15], which consists of 4088 paragraphs obtained from Postimees daily. The corpus has been annotated with sentiment as well as with rubric labels. The statistics of this dataset are given in Table 2. We split the data into training, testing and development set using 70/20/10 split preserving the ratios of different labels in the splits. All duplicates were removed from the corpus. In total, there were 17 duplicate paragraphs. We followed the suit of Pajupuu et al. [15] and removed the paragraphs annotated as ambiguous from the corpus. These paragraphs were shown to considerably lower the accuracy of the classification.

Table 2. Statistics of the Estonian Valence corpus

	Negative	**Ambiguous**	**Positive**	**Neutral**	**Total**
Opinion	429	242	162	139	972
Estonia	152	41	93	133	419
Life	138	47	207	128	520
Comments-Life	347	40	79	41	507
Comments-Estonia	368	27	50	56	501
Crime	170	12	11	16	209
Culture	57	40	86	79	262
Sports	76	81	152	76	385
Abroad	190	22	42	59	313
Total	1,927	552	882	727	4,088

For baseline, we trained supervised fastText classifiers [16] with pretrained fastText Wiki embeddings. The best hyperparameter values were found using the built-in fastText hyperparameter optimization.

3.3. Named Entity Recognition

The available Estonian NER corpus was created by Tkachenko et al. [17]. The corpus annotations cover three types of named entities: locations, organizations and persons. It contains 572 news stories published in local online newspapers Postimees and Delfi covering local and international news on a range of different topics. We split the data into training, testing and development set using 80/10/10 splits while preserving the document boundaries. Table 3 shows statistics of the splits.

Table 3. Statistics of the Estonian NER corpus

	Sentences	**Tokens**	**PER**	**LOC**	**ORG**	**Total**
Train	9,965	155,981	6,174	4,749	4,784	15,707
Dev	2,429	32,890	1,115	918	742	2,775
Test	1,908	28,370	1,201	644	619	2,464

As baselines, we report the performance of the CRF model [17] and the bilinear LSTM sequence tagger that was adapted from the Stanza POS tagger [13]. The tagger was trained on the NER annotations instead of POS tags, and the input was enriched with both POS tags and morphological features, i.e. the input to the NER model was the concatenation of the word, and its POS and morphological tag embeddings. The POS and morphological tags were predicted with the pre-trained Stanza POS tagger. The entity level performance is evaluated using the conlleval script from CoNLL-2000 shared task.

4. Experimental Setup

We conduct experiments with four different multilingual BERT models: multilingual cased BERT-base (mBERT), multilingual cased DistilBERT (DistilmBERT), cased XLM-100 and cross-lingual RoBERTa (XLM-RoBERTa). All these models are available via Hugging Face transformers library[2]. Each model is available with sequence lengths of 128 and 512 and we experiment with both. Table 4 shows some details of the models.

Table 4. Details of multilingual BERT models (all cased)

	Languages	**Vocab size**	**Parameters**
mBERT	104	119K	172M
XLM-100	100	200K	570M
DistilmBERT	100	119K	66M
XLM-RoBERTa	100	250K	270M

[2] `https://huggingface.co/transformers/`

To evaluate the performance of the multilingual BERT models on downstream tasks, we fine-tune all four BERT models for the NLP tasks described in Section 3. In addition to training the task-specific classification layer, we also fine-tune all BERT model parameters as well. For data processing and training, we used the scripts publicly available in the Hugging Face transformers repository. We tune the learning rate of the AdamW optimizer and batch size for each multilingual model and task on the development set using grid search. The learning rate was searched from the set of (5e-5, 3e-5, 1e-5, 5e-6, 3e-6). Batch size was chosen from the set of (8, 16). We find the best model for each learning rate and batch size combination by using early stopping with patience of 10 epochs on the development set.

5. Results

In subsequent sections, we present the experimental results on all multilingual BERT models for POS and morphological tagging, text classification and named entity recognition tasks.

5.1. POS and Morphological Tagging

The results for POS and morphological tagging are summarized in Table 5. In general, all tested multilingual BERT models are equally good and perform better than the Stanza and UDPipe baselines. DistilmBERT was the only multilingual model that did not exceed the baseline models results. On the other hand, the XLM-RoBERTa stands out with a small but consistent improvement over all other results displayed. Results also show that the sequence length of the model does not affect the performance in any way. The performance on XPOS is better than on UPOS. This is probably caused by the difference in the POS tag annotation schemes.

5.2. Text Classification

The sentiment and rubric classification task results are shown in Table 6. Multilingual models can easily outperform baseline fastText model. Similarly to POS and morphological tagging tasks, XLM-RoBERTa achieved the highest and DistilmBERT the lowest results overall. Even though there are more labels the in rubric classification task, it is still easier for the models to correctly classify than the sentiment classification task. Compar-

Table 5. POS and morphological tagging accuracy on Estonian UD test set.

Model	UPOS	XPOS	Morph	UPOS	XPOS	Morph
		Seq = 128			Seq = 512	
mBERT	97.42	98.06	96.24	97.43	98.13	96.13
DistilmBERT	97.22	97.75	95.40	97.12	97.78	95.63
XLM-100	97.60	98.19	**96.57**	97.59	98.06	96.54
XLM-RoBERTa	**97.78**	**98.36**	96.53	**97.80**	**98.40**	**96.69**
Stanza [13]	97.19	98.04	95.77			
UDPipe [14]	95.7	96.8	93.5			

Table 6. Rubric and sentiment classification accuracy

Model	Rubric	Sentiment	Rubric	Sentiment
	Seq = 128		Seq = 512	
mBERT	75.67	70.23	74.94	69.52
DistilmBERT	74.57	65.95	74.93	66.95
XLM-100	76.78	73.50	77.15	71.51
XLM-RoBERTa	**80.34**	**74.50**	**78.62**	**76.07**
fastText	71.01	66.76		

ison between the models with different sequence lengths is inconclusive—in some cases, the models with longer sequence are better, but not always.

5.3. Named Entity Recognition

The Table 7 (left) summarizes the NER results. We find that the task-specific StanfordNLP model is superior to all the multilingual BERT models, while XLM-100 and XLM-RoBERTa perform the best compared with other multilingual models. CRF based model was easily outperformed by all multilingual models except for DistilmBERT.

While performing these experiments, each sentence was treated as one sequence. This may have not optimally used the maximum sequence length available, especially in models with sequence length 512. As most sentences in our NER corpus do not reach the maximum length, we hypothesize that using longer sequences with the models of sequence length 512 would add more context for the model and thus improve the results. For that, we concatenate sentences from the same document to reach to the maximum 512 wordpiece sequence. The right-most section of the Table 7 shows the results of the experiments with longer input sequences. The numbers in the table show that concatenating the input sequences does not boost the scores. Compared with the regular results based on single sentences, only XLM-RoBERTa was able to utilize the maximum sequence length while the scores of other models decreased. The performance of the XLM-100 model suffered the most and obtained even lower results than DistilmBERT, which so far has gotten the lowest results in all tasks.

One possible reason why the multilingual BERT models were not able to improve over the Stanford tagger based NER model is that the Stanford baseline model makes use of the POS and morphological information while the BERT models do not. Adding

Table 7. NER tagging results. The right-hand part of the table shows the results with the models of sequence length 512, with the input sentences concatenated into sequences of maximum length

Model	Prec	Recall	F1	Prec	Recall	F1	Prec	Recall	F1
		Seq = 128			Seq = 512			Seq = Concatenated	
mBERT	85.88	87.09	86.51	**88.47**	88.28	88.37	86.42	89.64	88.01
DistilmBERT	84.03	86.98	85.48	85.30	86.49	85.89	83.18	87.38	85.23
XLM-100	**88.16**	88.11	88.14	87.86	89.52	88.68	73.27	80.48	76.71
XLM-RoBERTa	87.55	**91.19**	**89.34**	87.50	**90.76**	**89.10**	**87.69**	**92.70**	**90.12**
CRF	87.97	88.03	87.99						
StanfordNLP	**90.55**	**91.07**	**90.80**						

Table 8. NER F1 scores with additional POS and morphological information.

	PRE-BERT			POST-BERT			Regular
	POS+Morph	POS	Morph	POS+Morph	POS	Morph	-
mBERT	82.58	83.80	86.41	87.10	**88.59**	87.13	85.39
distilmBERT	70.30	79.39	82.16	81.84	83.51	84.97	**85.48**
XLM-100	80.26	82.48	87.36	81.25	86.76	86.42	**88.14**
XLM-RoBERTa	89.71	**89.86**	89.43	89.52	86.76	87.62	89.34

POS and/or morphological information, the BERT model has the potential to improve their results, as especially POS information can be crucial for detecting proper names that make up a large number of named entities.

We experimented with two different approaches for adding POS and morphological information to the BERT-based models. The first approach (PRE-BERT) only changes the input of the models. Here, the POS and morphological information is input directly into the BERT model by adding the embeddings of POS and morphological tags to the default input embeddings by summing all embedding vectors. The second approach (POST-BERT) requires slight changes in the sequence classification model. Here, the embeddings of POS and morphological tags are concatenated to the output vector obtained from the BERT model and the concatenated representation is then input to the classification layer. We expect the POST-BERT method to perform better because in this approach, the POS and morphological information is fed to the model closer to the classification layer and thus has the more direct influence on the classification decision. The advantage of the PRE-BERT approach, on the other hand, is its simplicity as it does not require any changes in the model architecture. For training with both approaches, we used the POS and morphological information supplied with the NER corpus. The POS and morphological tags for the test part were obtained with the open-source Estonian morphological analyzer Vabamorf [18] that uses the same annotation scheme as supplied in the NER corpus.

Table 8 shows that the results of adding POS and/or morphological features is mixed. While mBERT achieves a large improvement and XLM-RoBERTa, a marginal increase in performance, the scores of other two models actually decrease quite a bit. Overall, as expected, the POST-BERT approach, where the extra features are concatenated to the output vector of BERT, is better than the PRE-BERT approach. The exception is again the XLM-RoBERTa model that with the PRE-BERT method achieves the best NER results of all multilingual models. However, this best score is still about one percentage point worse than the Stanford tagger baseline. From the three settings adding only POS or morphological features seems the best. To conclude, adding either POS or morphological features can be helpful for the mBERT and XLM-RoBERTa models, other two models were not able to use the extra features to increase the scores.

6. Conclusions

In this work, we compared multilingual BERT and BERT-like models with non-contextual baseline models on several downstream NLP tasks. For most tasks, multilingual models outperformed the previously proposed task-specific models, XLM-

RoBERTa achieving the highest scores on all the experimental tasks, while DistilmBERT performed the worst overall. Based on these results, we can recommend using the XLM-RoBERTa as a basis for neural NLP models for Estonian. Considering the results from previous works comparing multilingual BERT with language-specific BERT models [3,9], further performance gains can be obtained from training monolignual BERT for Estonian, in particular following the RoBERTa guidelines [19].

References

[1] Peters M, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, et al. Deep Contextualized Word Representations. In: Proceedings of NAACL; 2018. p. 2227–2237.
[2] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of NAACL; 2019. p. 4171–4186.
[3] Nozza D, Bianchi F, Hovy D. What the [MASK]? Making Sense of Language-Specific BERT Models. arXiv preprint arXiv:200302912. 2020.
[4] Conneau A, Khandelwal K, Goyal N, Chaudhary V, Wenzek G, Guzmán F, et al. Unsupervised Cross-lingual Representation Learning at Scale. In: Proceedings of ACL; 2020. p. 8440–8451. Available from: `https://www.aclweb.org/anthology/2020.acl-main.747`.
[5] Kondratyuk D, Straka M. 75 Languages, 1 Model: Parsing Universal Dependencies Universally. In: Proceedings of EMNLP-IJCNLP; 2019. p. 2779–2795.
[6] Conneau A, Lample G. Cross-lingual Language Model Pretraining. In: NIPS; 2019. p. 7059–7069. Available from: `http://papers.nips.cc/paper/8928-cross-lingual-language-model-pretraining.pdf`.
[7] Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. arXiv preprint arXiv:191001108. 2019.
[8] Pires T, Schlinger E, Garrette D. How Multilingual is Multilingual BERT? In: Proceedings of ACL; 2019. p. 4996–5001.
[9] Virtanen A, Kanerva J, Ilo R, Luoma J, Luotolahti J, Salakoski T, et al. Multilingual is not enough: BERT for Finnish. arXiv preprint arXiv:191207076. 2019.
[10] Baumann A. Multilingual Language Models for Named Entity Recognition in German and English. In: Proceedings of RANLP SRW 2019; 2019. p. 21–27.
[11] Akbik A, Blythe D, Vollgraf R. Contextual String Embeddings for Sequence Labeling. In: Proceedings of COLING; 2018. p. 1638–1649.
[12] Kuratov Y, Arkhipov M. Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language. arXiv preprint arXiv:190507213. 2019.
[13] Qi P, Zhang Y, Zhang Y, Bolton J, Manning CD. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In: Proceedings of ACL System Demonstrations; 2020. .
[14] Straka M, Straková J. Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In: Proceedings of the CoNLL 2017 Shared Task; 2017. p. 88–99.
[15] Pajupuu H, Altrov R, Pajupuu J. Identifying Polarity in Different Text Types. Folklore. 2016;64.
[16] Joulin A, Grave É, Bojanowski P, Mikolov T. Bag of Tricks for Efficient Text Classification. In: Proceedings of EACL; 2017. p. 427–431.
[17] Tkachenko A, Petmanson T, Laur S. Named Entity Recognition in Estonian. In: Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing; 2013. p. 78–83. Available from: `https://www.aclweb.org/anthology/W13-2412`.
[18] Kaalep HJ. An Estonian Morphological Analyser and the Impact of a Corpus on its Development. Computers and the Humanities. 1997;31(2):115–133.
[19] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. Roberta: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:190711692. 2019.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200598

Similarities and Differences of Lithuanian Functional Styles: A Quantitative Perspective

Justina MANDRAVICKAITĖ [a,c,1] and Tomas KRILAVIČIUS [b,c]
[a] *Vilnius University, Lithuania*
[b] *Vytautas Magnus University, Lithuania*
[c] *Baltic Institute of Advanced Technology, Lithuania*

Abstract. We report an analysis of similarities and differences in terms of selected characteristics of 3 Lithuanian functional styles (FS): administrative, scientific, and publicistic. We combined 8 quantitative indicators and multivariate statistical analysis for this task. We also analyzed tendencies of indicators to be more or less pronounced in particular FS.

Keywords. Functional styles, Lithuanian, multivariate statistical analysis, quantitative indicators

1. Introduction

We report analysis of similarities and differences in terms of selected characteristics of 3 Lithuanian FS: administrative, scientific, and publicistic. 8 quantitative indicators and multivariate statistical analysis were chosen.

We define functional style as a variety of standard language that is characterized by domain, contents, functions, stylistic devices, and linguistic means [1]. Although there are 5 FS in the Lithuanian language: colloquial, administrative, fictional, publicistic and scientific (colloquial and fictional styles were not included). Colloquial was not included, because we analyze only written language, and fictional because of the impact of the author's style [2].

2. Data

We use 3 corpora, one corpus per FS: a corpus of administrative style (A), a corpus of publicistic style (P), and a corpus of scientific style (S). Corpus A is based on the administrative part of the Corpus of the Contemporary Lithuanian Language [3]. Corpus P is based on delfi.lt corpus [4], consisting of news articles. Finally, Corpus S is based on the non-fiction part of Corpus of the Contemporary Lithuanian Language, consisting

[1]Corresponding Author: Justina Mandravickaitė, Vilnius University; Lithuania; Baltic Institute of Advanced Technology, Lithuania; E-mail: justina.mandravickaite@bpti.eu

of educational and popular science texts. The latter texts were supplemented with these summaries of doctoral dissertations. Thus, all in all, A has 5.8 million words (4,527 texts), S – 20.2 million words (1,025 texts) and P – 10.4 million words (13,450 texts).

3. Methods

3.1. Indicators as Features

As characteristics of FS, based on [5–7, 9] and others, we chose 8 indicators. They address different linguistic characteristics and have lower or almost no significant dependence on the length of text [5, 6]. Also, these indicators have mathematical as well as linguistic explanation, which leads to an easier interpretation of the results. Selected indicators are the following:

1. **Average Token Length (ATL)** is used as a simple readability measure in linguistics.
2. **Indicator *a*** measures proportion of high frequency (usually function words) and lower frequency words in a text [6], [7].
3. **Indicator** R_1 was developed as a measure of vocabulary richness that is focused on less frequent words [5, 7]. We use word forms instead of lemmas, thus R_1 is more likely to measure a diversity of less frequent word forms, which belong to the main parts of speech, such as verbs, nouns, adjectives, or adverbs.
4. **Relative Repeat Rate of McIntosh (**RR_{mc}**)** measures vocabulary concentration of the text [8, 9]. RR_{mc} is normalized RR [8, 9] and this is more suitable for comparison with other indicators.
5. **Moving Average Type-Token Ratio (MATTR)** is a modification of Type-Token Ratio (TTR), which is independent of text length [10, 11].
6. **Thematic Concentration (TC)** measures the degree a text is concentrated over its topic [8, 11, 12]. TC is based on thematic words – words that are normally less frequent, but in particular text has a frequency close to the most frequent words which are usually function words – which define topic of the text.
7. **Activity (Q)** indicator expresses the dynamism of the text in terms of proportion of verbs and adjectives [8, 11, 13].
8. **Verb Distances (VD)** measure how many words on average occur in the text between 2 consecutive verbs [8], which can be interpreted, in a simplified way, as measuring complexity of syntactic structure of the text [5].

3.2. Multivariate Statistical Analysis

To analyze similarities and differences of FS, non-parametric multivariate analysis of variance was applied. We chose a non-parametric variety of analysis because of a significant number of outliers as well as different amount of texts in the corpora [14]. We applied Kruskal-Wallis test in order to test whether administrative, scientific, and publicistic functional styles have statistically significant differences among each other. Dunn's test [15] was used to evaluate the differences between pairs of functional styles in terms of each indicator. We then calculated relative treatment effects scores to estimate the scope of differences [16].

Table 1. Results of Dunn test

Indicator	Corpora pair	*Z*-value	*p*-value (adapted to multiple comparisons)
ATL	A-S	18.36141	8.027132e-75
	A-P	98.41522	0.000000e+00
	S-P	32.20224	4.925667e-227
a	A-S	-14.32894	4.329066e-46
	A-P	-93.11607	0.000000e+00
	S-P	-33.71111	1.191995e-248
R_1	A-S	3.496798	0.001412636
	A-P	-91.122320	0.000000000
	S-P	-51.653576	0.000000000
RR_{mc}	**A-S**	**0.350791**	**1**
	A-P	-89.598949	0
	S-P	-47.500637	0
MATTR	A-S	-19.85568	2.952350e-87
	A-P	-101.12481	0.000000e+00
	S-P	-32.03546	1.050061e-224
TC	A-S	45.73392	0.000000e+00
	A-P	71.06957	0.000000e+00
	S-P	-11.34295	2.411528e-29
Q	A-S	18.98684	6.574064e-80
	A-P	-26.51339	2.038068e-154
	S-P	-34.17354	1.793499e-255
VD	A-S	17.86002	7.246323e-71
	A-P	77.51638	0.000000e+00
	S-P	21.74412	2.355814e-104

4. Results

Kruskal-Wallis test showed, that all the analyzed FS differ by all the indicators significantly ($p < 0.05$). Dunn's test revealed that differences between all pairs of FS in terms of each indicator were statistically significant, except for the A-S pair in terms of indicator RR_{mc} (see Table 1). To estimate the scope of these differences, the relative treatment effects scores of those differences are presented in Table 2. A higher score indicates a higher probability of higher values for certain indicator in the texts of certain FS, i.e.:

- higher ATL values indicate longer words (more difficult to read);
- higher a values indicate lesser proportion of high frequency words;
- higher R_1 values indicate higher diversity of less frequent word forms;
- higher RR_{mc} values indicate higher vocabulary concentration;
- higher MATTR values indicate on average higher numbers of unique word forms in comparison to all word forms;
- higher TC values indicate higer thematic concentration;
- higher Q values indicate more dynamic texts (more verbs in comparison to adjectives);
- higher VD values indicate more complex syntactic structure (longer distance between 2 consecutive verbs).

Thus, longer word forms are more probable in A than in S and P. However, in this case, S is closer to A than to P. For P, lower proportion of high frequency words is more probable than for A and S. However, A and S, in this case, are closer to each other than to P. Similarly, higher diversity of less frequent words and word forms is more probable in P, however, it is less probable in A and S, which, according to relative treatment effects score, have rather similar probability. Furthermore, higher vocabulary concentration is more probable in P, than in A and S, which have the same lower probability. Higher

Table 2. Relative treatment effects

Indicator	Corpus	Relative treatment effects
ATL	A	0.85
	S	0.67
	P	0.37
a	A	0.17
	S	0.31
	P	0.63
R_1	A	0.19
	S	0.15
	P	0.63
RR_{mc}	A	0.19
	S	0.19
	P	0.63
MATTR	A	0.14
	S	0.34
	P	0.64
TC	A	0.77
	S	0.31
	P	0.42
Q	A	0.42
	S	0.23
	P	0.55
VD	A	0.78
	S	0.60
	P	0.40

proportion of different words and word forms is more probable in P and less probable in A. In this case, S is more similar to A than to P. Also, higher thematic concentration is more probable in A and less in S. P stands in between, although is more similar to S than to A. Additionally, very dynamic texts are more probable in P, although this tendency is not highly pronounced as relative treatment effects score for A in terms of this indicator is not much lower. In this case, S have a lower probability of very dynamic texts. Finally, texts with more complex syntactic structure are more probable in A. S stands slightly closer to A than to P in this matter, while P has a lower probability for syntactically complex texts.

5. Conclusions and Future Plans

We report an analysis of similarities and differences in terms of certain characteristics of 3 Lithuanian FS: administrative, scientific, and publicistic. We combined 8 quantitative indicators and multivariate statistical analysis for this task. Administrative and scientific style are closer each other in terms of indicators ATL, a, R_1, RR_{mc}, MATTR and VD. Administrative and publicistic functional styles are closer to each other in terms of indicator Q. Scientific and publicistic FS are closer to each other in terms of indicator TC.

Our future plans include experimenting with different variety of quantitative indicators as well as cross-lingual comparison in terms of scope of characteristics of FS. Future plans also include some practical applications, such as automatic text classification according to FS.

References

[1] Župerka KR. Stilistika. III pataisytas ir papildytas leidimas. Šiauliai: VšĮ Šiaulių univ. leid. 2012.

[2] Bitinienė A. Grožinio stiliaus prozos tekstų tiesioginės kalbos sakinio ilgis ir struktūra. Kalbotyra. 2001;50:17-28.

[3] Kovalevskaitė J. Dabartinės lietuvių kalbos tekstynas–10 metų kaupimo ir naudojimo patirtis. Prace Baltystyczne. 2006; 3:231-41.

[4] Bumbulienė I, Mandravickaitė J, Boizou L, Krilavičius T. An overview of Lithuanian internet media n-gram corpus. In CEUR Workshop Proc.: SYSTEM 2017, Proc. of the symposium for Young Scientists in Technology, Engineering and Mathematics, Kaunas, Lithuania, Apr 28, 2017. Aachen: CEUR-WS, 2017, Vol. 1853 2017.

[5] Kubát M. Kvantitativní analýza žánrů. Disertační práce Univerzita Palackého v Olomouci, Filozofická fakulta, Olomouc. 2015.

[6] Zörnig P, Kelih E, Fuks L. Classification of Serbian texts based on lexical characteristics and multivariate statistical analysis. Glottotheory. 2016 Jun 1; 7(1):41-66.

[7] Popescu II. Word frequency studies. Walter de Gruyter; 2009 Jun 2.

[8] Kubát M, Matlach V, Čech R. QUITA. Quantitative Index Text Analyzer. Lüdenscheid: RAM-Verlag. 2014.

[9] Popescu II, Cech R, Altmann G. Vocabulary richness in Slovak poetry. Glottometrics. 2011;22:62-72.

[10] Covington MA, McFall JD. Cutting the Gordian knot: The moving-average type–token ratio (MATTR). Jrnl of quant. ling. 2010 May 1;17(2):94-100.

[11] Kubát M, Cech R. Quantitative Analysis of US Presidential Inaugural Addresses. Glottometrics. 2016 Jan 1;34:14-27.

[12] Čech R. Tematická koncentrace textu v češtině. ÚFAL, Ústav formální a aplikované lingvistiky; 2016.

[13] Zörnig P, Altmann G. Activity in Italian presidential speeches. Glottometrics. 2016 Jan 1;35:38-48.

[14] Field A, Miles J, Field Z. Discovering statistics using R. Sage pub.; 2012 Mar 31.

[15] Dinno A. Nonparametric pairwise multiple comparisons in independent groups using Dunn's test. The Stata Jrnl. 2015 Apr;15(1):292-300.

[16] Ellis AR, Burchett WW, Harrar SW, Bathke AC. Nonparametric inference for multivariate data: the R package npmv. Jrnl. of Stat. Soft. 2017 Jan 1;76(4):1-8.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200599

Targeted Aspect-Based Sentiment Analysis for Lithuanian Social Media Reviews

Mažvydas PETKEVIČIUS [1], Daiva VITKUTĖ-ADŽGAUSKIENĖ and Darius AMILEVIČIUS
Vytautas Magnus University, Lithuania

Abstract. The paper presents research results for solving the task of targeted aspect-based sentiment analysis in the specific domain of Lithuanian social media reviews. Methodology, system architecture, relevant NLP tools and resources are described, finalized by experimental results showing that our solution is suitable for solving targeted aspect-based sentiment analysis tasks for under-resourced, morphologically rich and flexible word order languages.

Keywords. Targeted aspect-based sentiment analysis, entity recognition, deep learning, social media

1. Introduction

While initially sentiment analysis (or opinion mining) was implemented by assigning sentiment polarities on the sentence level, it gradually evolved towards fine-grained aspect-based sentiment analysis (ABSA), allowing to identify sentiments of several aspects in a sentence.

In order to achieve even better accuracy of sentiment analysis, aspect-based sentiments are classified towards target entity mentions in given sentence, leading to so-called targeted ABSA (TABSA). Such a complex approach is related to the nature of user opinions or reviews in social media texts, where usually we have several entities mentioned in a text, together with different aspects and sentiment polarities of these entities being addressed.

Following this approach, several sequential tasks must be solved – identifying entity mentions, resolving aspect categories and classifying sentiment polarities with respect to aspect categories. Different combinations of information extraction, deep learning and natural language processing (NLP) techniques are used for these tasks.

Related work. While sentiment analysis on the sentence level shows really good results, especially when applying deep learning techniques, TABSA remains a difficult task. Related research addresses different issues, including building balanced datasets [1], choosing the right model architecture [2] and coping with language specifics [3].

[1]Corresponding Author: Mažvydas Petkevičius; Vytautas Magnus university; E-mail: mazvydas.petkevicius@vdu.lt.

Different architectures are considered, especially neural sequential models, such as long short-term memory (LSTM) networks with different modifications - hierarchical attention mechanism [4,5], delayed memory update [6]. Another approach suggests convolutional neural network based (CNN-based) aspect-level sentiment classification models, consisting of two CNNs [7].

Problem statement. TABSA is normally a suitcase research problem [8] that requires tackling many natural language processing (NLP) tasks, including text preprocessing, named-entity recognition, dependency parsing, etc.

Also, dataset preparation involves the use of different NLP resources as annotated corpora, WordNet ontologies, or topical dictionaries. The quality of these resources directly influences the precision of the overall solution. While the necessary NLP tools are already mature for well-resourced languages, the situation is different for the Lithuanian language.

We propose a TABSA architecture for the Lithuanian language, combining CNN-based classification models and commonsense language knowledge embedded in NLP components. It is presented on the example of university study program sentiment analysis in social media reviews.

2. Methodology

Our TABSA approach considers sentiment analysis as aimed at different entities (targets) – organizations, products, services, persons, etc. We define entities as composite objects, having a set of components, and, also, a set of attributes. Each component may have its own sub-components and its set of attributes, and so on. Thus, an entity can be hierarchically decomposed based on the part-of relation [9]. The aspects of an entity, in general, are the components and attributes related to the entity or its constituent parts. In our example of study program sentiment analysis, entity (target) is hierarchically structured in three levels (university, study direction, and study program), and aspects, such as “teaching”, “studies”, “infrastructure”, “career”, can be attributed to any of these levels.

Overview of the approach. Joint recognition of targets and aspect-sentiment pairs in TABSA is a difficult task, and, usually, these two issues are separated in research works [6]. As we consider the case of targets having a composite and hierarchical structure, our approach handles target recognition and aspect-sentiment learning as separate tasks, sharing aspect classification results and corresponding datasets. Rule-based linking phase is included for aggregating the results of those two tasks at the end of the overall process.

Target recognition: task description. In our case, target recognition task is aimed at identifying composite structure of hierarchical named-entities and a corresponding aspect category. Rule-based matching is applied for named-entity recognition (NER), allowing either matching against a knowledge-base, or applying a partial rule-defined match [10]. Linking of all the Target components, including aspect category, obtained from aspect classifier, is performed in the context of a chunk by applying corresponding rules.

Aspect-sentiment classification: task description. Linguistic and statistic research of social media corpus texts revealed that aspect terms are expressed in nouns and noun-phrases (approx. 65 percent), and verbs and verb-phrases (around 35 percent). Both aspect and sentiment terms are explicit and implicit, however, in our case, we limit ourselves to explicit terms as they constitute the majority. Supervised learning is applied with the aim of attributing classes to aspects and sentiments.

Overall model. Conceptual architecture of the proposed solution is presented in Figure-1.

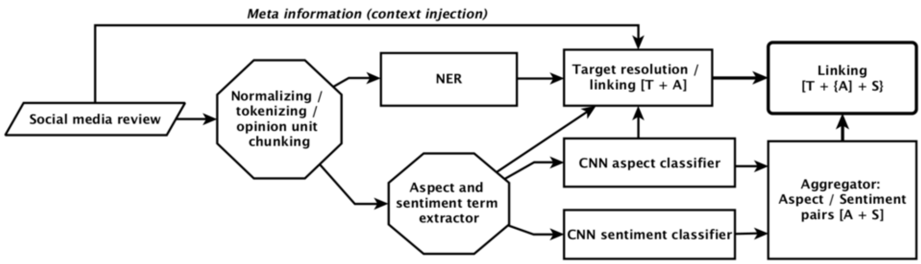


Figure 1. Conceptual TABSA architecture for the proposed solution

In our solution, we go in parallel with the target recognition and aspect-sentiment classification, linking the results afterwards. Target recognition is implemented as a sequence of NER and Target linking tasks. The NER component gets tokenized and normalized input, which is then segmented into meaningful chunks by a syntactic analyzer. Entity recognizer consists of two main components: the main logic, consisting of a ruleset and a knowledge base. Entities are recognized and linked throughout multiple phases: candidate entity generation, candidate entity ranking and unlinkable mention prediction. The recognized entities are then handed off to the target resolution and linking component, which combines target entities with the aspect. The recognized entities are then passed over to the target resolution and linking component, combining target entities with the aspect.

For the prediction of aspect and sentiment categories, we implement two Convolutional Neural Network (CNN) classifiers using the Keras [2] library with TensorFlow [3] as the backend. Dependency-based noun and verb phrase extractor feeds candidate terms from an opinion unit into neural network-based classifiers. Predictions from classifiers go to an aggregator, where classified aspect and sentiment terms are aggregated on opinion unit syntax basis.

Finally, the results from both Target Recognizer and Aspect-Sentiment Classifier are passed over to the Linking component for aggregation, using rule-based techniques and aspect category as the key.

3. Experiment

Resources (Datasets). Lithuanian language is an under-resourced language, especially considering specific topical domains. Our base dataset consists of 13,000 sentences from various Lithuanian social media sources with reviews of a selected domain. 45 percent of the dataset are complex sentences, containing two or more meaningful opinion units

[2]https://keras.io
[3]https://www.tensorflow.org

(sub-sentences). Three human experts have annotated them with aspect category labels ("teaching", "studies", "infrastructure", "career", "common", "NULL"), as well as with sentiment category labels ("very negative", "negative", "neutral", "positive", "very positive"). Generally, we assume that opinion cannot be "neutral", as only facts can be considered as neutral. However, we include aspect category "NULL" and sentiment "neutral" for mark-up of some sub-sentences that are out of rage of our domain evaluation. Experts also manually extracted aspect and sentiment terms. In order to resolve problems with insufficient dataset and to balance it for all categories, we used data augmentation techniques. For this purpose, we used synsets for aspect and sentiment terms from LitWordNet [4] [11] and Hunspell [5]engine with Lithuanian grammar and lexicon sets [6] for generating necessary morphological forms for synonym words. Our augmented corpus contains 250,000 words (19,500 unique word-types).

Table 1. Augmented corpus structure

	Infrastructure	**Teaching**	**Studies**	**Carrier**	**Common**	**Neutral**
Very positive	500	2,000	2,100	800	300	300
Positive	500	2,000	2,100	800	300	300
Neutral	500	2,000	2,100	800	300	300
Negative	500	2,000	2,100	800	300	300
Very negative	500	2,000	2,100	800	300	300
Total	2,500	10,000	10,500	4,000	1,500	1,500

Knowledge-base for entity recognition and linking was constructed on official data, including fields of studies, study programs and universities and their colloquialisms in social media. NER rule-based matcher was implemented using spaCy [7] library.

System setup. To solve aspect-based sentiment multi-class classification problem, complex classifiers were used. A complex classifier consists of two classifiers with identical CNN architecture: one for aspect classification task, and the second - for sentiment classification task. CNN-based classifiers were used as CNNs provide a faster alternative to LSTM models at a comparable performance. They are faster to train and use fewer parameters. CNN models, when applied to text, are a reasonable alternative when there is no strong dependence on distant past of the input sequence, as is in our case, since opinion units are normally very short. In our experiments, complex classifier was used in two scenarios: 1) Baseline model (S1), and 2) Advanced model (S2). For both scenarios, datasets were divided into a training subset (80 percent) and testing subset (20 percent).

In scenario S1, simple CNN architecture was used for multi-class classification with mini-batch size of 256. Our model was starting to overfit at about epoch 30, so we trained it for 25 epochs. In this scenario, aspects and sentiment dictionaries, extracted and labeled by human experts, were used for lookup. A study review corpus, labeled by human experts, was also used. Embedding was done by Keras internally, using our custom studies' review corpus.

[4] http://mackus.vdu.lt/LitWordNet
[5] http://hunspell.github.io
[6] https://github.com/Semantika2
[7] https://spacy.io

Scenario S2 was more complex. In this case, CNN consists of the following layers: embedding layer, two 1D convolutional layers, max polling layer, global max pooling layer and two dense layers. Rectified linear unit activation function is used, and the final dense layer uses the Sigmoid activation function. The following CNN hyperparameters were used: dropout rate (p) of 0.5, kernel regularizer l2=1e-4, mini-batch size of 256 and 5 epochs. Since CNNs can learn patterns in word embeddings, we used FastText word embeddings to use sub-word information. For our purpose, we created our own FastText embedding model using our custom Lithuanian review corpus. CNNs were trained only on a targeted studies' review corpus, where reviews were labeled by human experts.

For Target recognition (NER) spaCy's rule-based matcher was utilized [8], where a set of patterns comprises a ruleset. These rules were deducted from the knowledge base consisting of several types of entities:

- 57 universities and colleges;
- 107 fields of studies (with relations to universities and study programs);
- 1001 study programs (that are related to fields of studies).

These were then either lemmatized or converted into corresponding regular expressions and added as patterns to spaCy to match multi-word phrases.

Table 2. Experiment results: selected examples

Review	NER	Aspect	Sentiment
Complex sentence (two opinion units, two named entities):			
Opinion unit 1: Šiaip **[university1]** gali pasigirti geresne technika straipsnyje aiškiai parašyta tai *(In general [university1] can be proud of better equipment as stated in the article)*	[university1]	Infrastructure	Positive
Opinion unit 2: [university2] gal kai kurie dėstytojai aukštesnio lygio *(in [university2] maybe some teachers can be considered superior)*	[university1]	Teaching	Negative
Complex sentence (two opinion units, one named entity):			
Opinion unit 1: Šiaip **[university1]** dėstytojai puikūs *(In [university1] teachers are excelent)*	[university1]	Teaching	Positive
Opinion unit 2: bet technika pasenusi *(but the equipment is outdated)*	[injected university1]	Infrastructure	Negative
Non-complex sentence (one opinion unit, one named entity):			
Nieko nesupratau ką dėstė visą semestrą *(didn't understand a single thing they were teaching)*	[injected university1]	Teaching	Negative

[8] https://spacy.io/usage/rule-based-matching

Preprocessing. The preprocessing phase involves text tokenization and normalization. Since often one sentence contains two or more meaningful units with an [aspect, sentiment] pair, as a result we get a sequence of meaningful opinion units: e.g., "sentence 0: [{unit1}, {unit2}, ... {unitN}]. A hybrid approach, consisting of pattern-recognition and dependency-based parser (spaCy for noun phrase chunking and Textacy [9] for verb-phrase chunking) was applied for splitting text into opinion units. spaCy parser is trained on Vytautas Magnus University "gold standard" treebank for Lithuanian language ALKSNIS v2 [10].

Results. Our solution was tested on unseen reviews from Lithuanian social media sources. Examples of the obtained results for selected test sentences are presented in Table 2. Component-related quality measures are presented in Table 3.

Table 3. Evaluation

	NER	Aspect classifier (S1)	Sentiment classifier (S1)	Aspect classifier (S2)	Sentiment classifier (S2)
Precision	0.66	0.88	0.89	0.89	0.92
Recall	0.76	0.85	0.83	0.86	0.86
Accuracy	83 %	93 %	91 %	94 %	93 %
F1-score	0.71	0.86	0.85	0.87	0.88

4. Discussion and Conclusions

Comparing the results of both experiments, the second setup (scenario S2 with a more complex CNN architecture) has shown slightly better results, but some considerations should be taken into account. In real-life applications, not only benchmark results, but also economic aspects are important. Though scenario S1 with a simple CNN architecture has shown only slightly worse results, this scenario requires significantly less computational resources. On the other hand, scenario S1 requires much more human expert efforts (manual aspect and sentiment term extraction and annotation). The scenario S2 is the opposite. It requires much more computational resources, but, at the same time, much less human expert efforts. In addition, learning transfer can be used in scenario S2. It must also be noted, that for under-resourced language or/and under-resourced domain cases, efficient and high-quality data augmentation technique is needed in both scenarios. That, in its turn, requires additional human expert efforts for augmented data set evaluation as well as additional computational expenses and availability of certain NLP tools.

Since our final scope is an end-to-end production-ready solution, we found that our proposed model performed very well in both scenarios with topical social media reviews. Unfortunately, we have no possibility to compare the results with other similar systems,

[9] https://github.com/chartbeat-labs/textacy
[10] https://github.com/UniversalDependencies/UD_Lithuanian-ALKSNIS

because most of them are for the English language. This is the first attempt to construct TABSA for Lithuanian.

A TensorFlow friendly combo: tagger and dependency-based parser is of crucial importance for TABSA. Nowadays, the best solution is spaCy, but all the implementations for spaCy's Lithuanian support must be retrained for the Lithuanian language.

The language normalizer plays a very important role in our solution since social media texts are full of misspellings.

Our data augmentation methodology demonstrated its suitability in solving insufficient dataset, domain-oriented and imbalanced dataset problems for the Lithuanian language.

Our future work is planned towards trying and comparing different neural net architectures and different word-embedding and language representation models. For example, BERT language representation model (might provide more advanced embeddings than FastText, but is more computationally expensive). On the other hand, FastText has already proved its ability to deal well with morphologically rich languages, especially solving Out-Of-Vocabulary (OOV) problem. For this reason, we used FastText embedding for our initial TABSA trials.

Lower performance of entity recognizer can be attributed to several factors such as word ambiguity, fuzzy partial matches, incorrect entity linking due to the limited context etc., although it performs reasonably well with exact entity matches.

References

[1] Saeidi M, Bouchard G, Liakata M, Riedel S. SentiHood: Targeted Aspect Based Sentiment Analysis Dataset for Urban Neighbourhoods. Proceedings of COLING 2016, Osaka, Japan, 2016.

[2] Augustyniak L, Kajdanowizc T, Kazienko P. Comprehensive Analysis of Aspect arXiv preprint arXiv:1909.04917, 2019. https://arxiv.org/pdf/1909.04917.pdf

[3] Loukachevitch N, Kotelnikov E, Blinov P. Types of Aspect Terms in Aspect-Oriented Sentiment Labeling. Proceedings of the 5th Workshop on Balto-Slavic Natural Language Processing, Hissar, Bulgaria, 2015.

[4] Ma Y, Peng H, Cambria E. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm. In Proceedings of AAAI 2018, New Orleans, USA, 2018.

[5] Chen P, Sun Z, Bing L, Yang W. Recurrent attention network on memory for aspect sentiment analysis. In Proceedings of EMNLP 2017, Copenhagen, Denmark, 2017.

[6] Liu F, Cohn T, Baldwin T. Recurrent Entity Networks with Delayed Memory Update for Targeted Aspect-Based Sentiment Analysis. Proceedings of NAACL-HLT (2018), New Orleans, USA, 2018.

[7] Noh Y, Park S, Park SB. Aspect-Based Sentiment Analysis Using Aspect Map. Appl. Sci. 2019, 9 (16), 3239.

[8] Cambria E, Poria S, Gelbukh A, Thelwall M. Sentiment Analysis is a Big Suitcase. IEEE Intelligent Systems, Vol. 32, Issue 6, 2017.

[9] Zhang L, Liu B. Aspect and Entity Extraction for Opinion Mining. In: Chu W. (eds) Data Mining and Knowledge Discovery for Big Data. Studies in Big Data, vol 1. Springer, Berlin, Heidelberg, 2014.

[10] Shen W, Wang J, Han J. Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions, IEEE Transactions on Knowledge and Data Engineering, Vol. 27, Issue 2, 2014.

[11] Vitkutė-Adžgauskienė D, Dainauskas J, Amilevičius D, Utka A. Lithuanian Word Network – LitWordNet. Darbai ir Dienos. 2016. Vol. 64 (In Lithuanian).

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200600

Automatic Extraction of Lithuanian Cybersecurity Terms Using Deep Learning Approaches

Aivaras ROKAS [a,1], Sigita RACKEVIČIENĖ [b]
and Andrius UTKA [a]
[a] *Vytautas Magnus University, Lithuania*
[b] *Mykolas Romeris University, Lithuania*

Abstract. The paper presents the results of research on deep learning methods aiming to determine the most effective one for automatic extraction of Lithuanian terms from a specialized domain (cybersecurity) with very restricted resources. A semi-supervised approach to deep learning was chosen for the research as Lithuanian is a less resourced language and large amounts of data, necessary for unsupervised methods, are not available in the selected domain. The findings of the research show that Bi-LSTM network with Bidirectional Encoder Representations from Transformers (BERT) can achieve close to state-of-the-art results.

Keywords. Cybersecurity, terminology, automatic term extraction, deep learning, neural networks, embeddings

1. Introduction

Automatic term extraction is extensively used for the development of termbases and ontologies which are essential in translation, teaching/learning language for specific purposes, domain-specific knowledge acquisition, etc. In addition to well-established statistical, linguistic and hybrid methods, the state-of-the art automatic term extraction is performed by applying machine learning and deep learning systems. However, the latter methods are still under development and need extensive research, especially for under-resourced languages such as Lithuanian. This paper presents research results on the deep learning methods aiming to determine the most effective one for automatic extraction of Lithuanian terms from a specialized domain (cybersecurity) with restricted resources. To achieve the aim, the following objectives were set:

1. To compile a specialised corpus comprising documents on cybersecurity issues;
2. To develop the gold standard corpus (training, validation and test data) with manually labelled terminology;
3. To test various deep learning models (pre-processing of the data, automatic term extraction, and comparison of the results).

Since Lithuanian is a less resourced language, supervised and semi-supervised deep learning methods are most suitable for automatic extraction of Lithuanian terminology

[1] Corresponding Author: Aivaras Rokas; Vytautas Magnus University, K. Donelaičio st. 58, Kaunas, Lithuania; E-mail: aivaras.rokas@vdu.lt.

as unsupervised methods require very large amounts of data. Therefore, in this research, semi-supervised approach was chosen.

To our knowledge, this is the first attempt to apply deep learning approach for Lithuanian term extraction. Until now this method has been mostly used for English terminology [1], [2], [3], [4], [5].

2. Background of the Research

In our research, two types of networks are applied to terminology extraction: long short-term memory (LSTM) and Gated Recurrent Unit (GRU), as well as two types of embeddings: FastText and BERT. Below, the main features of the methods applied are discussed.

2.1. LSTM and GRU Networks

During the last decade, one of the most widely used deep learning methods has been LSTM networks, also applied for terminology extraction. In this natural language processing task, terminology extraction is seen as a sequence labelling problem, where sequence is understood as words in a sentence [1], [2], [3], [4], [5].

LSTM is a type of recurrent neural network (RNN), which uses a cell state and three gates and is able to avoid the long-term dependency problem, memorize data for a longer period of time, and is able to fix vanishing gradient problems which plague generic RNNs [6].

However, LSTM networks have their own shortcomings, for example, a simple LSTM cannot account for context from the future, only from the past. Therefore, for certain NLP tasks a bidirectional LSTM network is employed which is able to make use of both past and future inputs. A bidirectional LSTM has two LSTMs, one capturing the information from the past and another capturing the information from the future, thus potentially improving a generic LSTM network.

To ensure that tags stay consistent, a Conditional Random Fields (CRF) network can be implemented as well. CRF is a probabilistic method for marking and segmenting sequence data [7]. CRFs are able to predict tags using context and calculate the likelihood of transitioning from one tag to another.

A GRU network is yet another type of RNN which, compared to LSTM network, requires fewer parameters and less computational power. It uses only two gates (reset and update gate), whereas LSTM network uses three gates (input, output and forget gate). Therefore, GRU network potentially should be more suitable for applications where training data is scarce [8]. Similarly to LSTM, the GRU network can be potentially improved by utilizing a bidirectional GRU network and further enhancing it by combining it with the CRF network.

2.2. Word Embeddings

In order to employ neural networks for text analysis, word embeddings are a necessary prerequisite. Word embeddings are “dense, distributed, fixed-length word vectors, built using word co-occurrence statistics as per the distributional hypothesis” [9: 2]. Word embeddings can capture semantic and syntactic information of words [10]. Training word embeddings does not require a labelled dataset, but requires a substantial amount of unlabelled data. There are a variety of embeddings such as word2vec, GloVe, FastText, etc. However, word embeddings like word2vec and GloVe cannot deal with unknown or out-of-vocabulary words. FastText is an improved version of Mikolov's word2vec embedding [11]. It is able to learn morphology of words since it is based on the skip-gram model where each word is represented as a bag of n-gram characters and is able to handle unseen words. Therefore, we use FastText word embeddings in our experiments as FastText is more suited for languages such as Lithuanian with the rich vocabulary and complex morphology.

However, FastText has limitations: it creates a word vector based on all the sentences where it has occurred and does not consider different meanings which a word acquires in different contexts. This problem is solved by using contextual embeddings. Presently, the most widely used one is BERT [12]. It is a multi-layer bidirectional Transformer encoder, which is able to consider context and create a different vector for each contextual use of a word. It can potentially improve previously described networks that are using fixed embeddings like FastText. In our experiment, we compare neural networks using FastText with BERT to determine the best method for automatic terminology extraction of Lithuanian terms.

3. Experiment of Automatic Term Extraction

3.1. Datasets

For the purposes of the research, the specialised Lithuanian cybersecurity corpus was compiled. The corpus is intended to reflect the use of cybersecurity language in original and translated texts over a period of 20 years (1999-2019) and is composed of five main categories of texts grouped according to their genres:

1. Legal acts of the Republic of Lithuania: laws, resolutions of the government, orders of ministers on cybersecurity issues;
2. Administrative documents: reports of the National Cybersecurity Centre;
3. Translated EU legislation: EU secondary law acts (directives, regulations); communications of the Commission, opinions of the committees, etc.;
4. Translated international conventions: Convention on Cybercrime;
5. Academic papers: textbooks, scientific papers and books on cybersecurity;
6. Informational publications for the general public on cybersecurity.

Thus, the corpus reflects the use of cybersecurity terms both in national and international settings. The size of the corpus is over 2 mil. words (2,363,618) [13].

As a semi-supervised deep learning approach was chosen for the research, it was necessary to compile the gold standard for the training of deep neural network models used in the experiment. A very small-scale corpus of the selected documents (66,706 words) was compiled for the given purpose and 1,258 cybersecurity terms were manually annotated. The following annotation criteria were formulated: a) linguistic criterion (only nominal units were annotated – nouns, noun phrases, abbreviations, combinations of noun phrases and abbreviations, e.g., *saugumas* 'security', *integruotasis saugumas* 'security by default', *IRT produktas* 'ICT product'); b) conceptual criterion (only nominal units holding relevant terminological value, i.e. denoting concepts of or related to cybersecurity domain, were annotated, e.g. *kibernetinė grėsmė* 'cyber threat', *kibernetiniai išpuoliai* ‘cyberattacks’, *informuotumas apie kibernetinį saugumą* ‘cybersecurity awareness’).

In this research, the gold standard data were annotated using the BIESO annotation format [14].

3.2. Pre-Processing of Data

In the initial stage of the experiment, pre-processing of cybersecurity corpus and gold standard corpus was conducted. The following pre-processing tasks were performed: file conversion to plain text format, character encoding change, word tokenization, stop-word list development, and text formatting.

In order to train the deep neural network, the gold standard dataset was divided into 3 parts: 70% for training, 20% for validation and 10% for testing.

In this research, word embeddings (that capture syntactic and semantic information of a word) generated by the skip-gram method of FastText and BERT-base multilingual contextual embeddings from Google were applied to the deep neural network [12], [15]. They were selected to better represent rare words [16]. In order to have more effective FastText word embeddings, the dataset was supplemented by the entire Lithuanian Wikipedia database which contains 27,907,392 million words.

3.3. Experimental Setup

In preparation for the experiment, the following methods were analysed: Bidirectional Long Short-Term Memory with CRF (Bi-LSTM-CRF), Bi-LSTM, LSTM, as well as Bidirectional Gated Recurrent Unit with CRF (Bi-GRU-CRF), Bi-GRU and GRU. The experiments by other researchers revealed that the most suitable method to our task would be the Bi-LSTM-CRF [1], [17], [18]. The Bi-LSTM method can “take into account an effectively infinite amount of context on both sides of a word and eliminates the problem of limited context that applies to any feed-forward model” [17: 357], and the CRF layer can take into account the surrounding tags so that predictions stay consistent.

In order to determine the most optimal model, the experiment was carried out in the following stages:

- Firstly, various baseline LSTM and GRU networks were tested using Adam optimizer and FastText embeddings;
- Secondly, each of the best baseline LSTM and GRU networks were tested with various optimizers;
- Thirdly, the best model was compared with a model that has been trained using BERT contextual embeddings to test if contextual embeddings can further improve our model.

Baseline networks were tested using the following hyperparameters: batch size 32, hidden dimensions 100, word vector dimension 100, number of epochs 100, dropout 0.5. These hyperparameters were selected through experimentation of various values and combinations. For example, the increasing the number of hidden layers improves the test error, while a small number of hidden dimensions would lead to underfitting. A low dropout value would yield insignificant results, while a too high a dropout value would result in under-learning.

3.4. Results

In this section, we present the results of our terminology extraction tests performed applying LSTM and GRU networks.

3.4.1. Baseline Tests

In order to identify which of LSTM and GRU baselines perform the best, we have tested 8 baselines: LSTM, LSTM-CRF, Bi-LSTM, Bi-LSTM-CRF, GRU, GRU-CRF, Bi-GRU, and Bi-GRU-CRF.

Table 1. Results of baseline LSTM models

No.	Model	Precision	Recall	F1
1.	LSTM	63.3 %	60.7 %	62.0 %
2.	LSTM-CRF	68.2 %	66.6 %	67.4 %
3.	Bi-LSTM	70.7 %	67.5 %	69.1 %
4.	Bi-LSTM-CRF	73.5 %	67.5 %	70.3 %

Table 2. Results of baseline GRU models

No.	Model	Precision	Recall	F1
1.	GRU	64.5 %	61.7 %	63.1 %
2.	GRU-CRF	70.1 %	61.5 %	65.8 %
3.	Bi-GRU	68.5 %	67.3 %	67.9 %
4.	Bi-GRU-CRF	70.9 %	67.5 %	69.2 %

The results provided in Table 1 and Table 2 reveal that Bi-LSTM-CRF model performed best achieving F1 score of 70.3 %. The second position was taken by Bi-GRU-CRF which fell short only by 1.1 %. Bi-LSTM took the third position and fell short from Bi-LSTM-CRF by 1.2 %. The worst performing models proved to be generic LSTM reaching only 62.0 % and generic GRU reaching 63.1 %.

3.4.2. Bi-LSTM-CRF and Bi-GRU-CRF Tests with Various Optimizers

The efficiency of neural network training greatly depends on optimisation strategies. The Bi-LSTM-CRF and Bi-GRU-CRF models were tested using the following optimizers: Adam [19], SGD [20], AdaDelta [21], RMSprop [22], Adagrad [23]. It is important to note that the learning rate for each optimizer was set to 0.001, except for Adagrad and SGD for which the learning rate was set to 0.01.

The findings provided in Table 3 and Table 4 reveal that the two best variations of Bi-GRU-CRF and Bi-LSTM-CRF are the ones with RMSprop and AdaDelta optimizers respectively. The highest scores in all three categories (precision, recall and F1) were reached by Bi-LSTM-CRF with AdaDelta optimizer with 5.2 % increase, when compared to the best baseline test.

Table 3. Results of five optimizers applied to Bi-LSTM-CRF

No.	Optimizer	Precision	Recall	F1
1.	Adam	73.5 %	67.5 %	70.3 %
2.	Stochastic gradient descent	69.0 %	55.4 %	61.3 %
3.	AdaDelta	78.5 %	72.7 %	75.5 %
4.	RMSprop	76.3 %	71.6 %	73.8 %
5.	Adagrad	71.3 %	59.3 %	64.7 %

Table 4. Results of five optimizers applied to Bi-GRU-CRF

No.	Optimizer	Precision	Recall	F1
1.	Adam	70.9 %	67.5 %	69.2 %
2.	Stochastic gradient descent	68.3 %	64.7 %	66.5 %
3.	AdaDelta	65.8 %	61.6 %	63.7 %
4.	RMSprop	78.2 %	68.4 %	73.3 %
5.	Adagrad	72.5 %	63.7 %	68.1 %

3.4.3. BERT

In the last stage of the experiment, the best model (Bi-LSTM-CRF with AdaDelta optimizer and FastText embeddings) was contrasted to Bi-LSTM network with BERT embeddings.

For our test with BERT, we used Adam optimization algorithm with weight decay as it is the default optimizer that BERT was trained on. The hyperparameters remained the same as in the previous networks. Our Bi-LSTM network trained with BERT embeddings reached precision of 79.4 %, recall 77.8 %, and F1 78.6 %. This is a 3.1 % F1 increase which is significant, especially with such a small training dataset. The initial review of the extracted terms shows that BERT is able to extract more previously unseen terms compared to Bi-LSTM-CRF. Overall, BERT seems to improve our model in every aspect.

During the experiment, we discovered that having trained our neural network using multilingual BERT embeddings with monolingual (Lithuanian) training data, the model has also trained itself on 103 other languages. This phenomenon is recorded by Pires et al., as well [24]. This is possible because originally multilingual BERT embeddings were trained on 104 different languages. Therefore, it was able to recognize and extract cybersecurity terms from all 104 languages that multilingual BERT supports despite the training data being annotated only with Lithuanian terms. This can potentially be very useful in bilingual and multilingual NLP tasks such as supervised or semi-supervised terminology extraction by reducing the amount of annotation data. In order to determine its effectiveness and reliability on other languages for terminology extraction, a more extensive testing is required.

4. Conclusions

The presented experiments confirm that deep learning models can be successfully applied to automatic extraction of Lithuanian domain specific terms and enable to achieve high precision, recall, and F1 scores even with very small annotated training data.

In the first stage of the experiment where the baselines of LSTM and GRU neural networks were tested, Bi-LSTM-CRF and Bi-GRU-CRF networks showed the best performance reaching F1 scores of 70.3 % and 69.2 %, respectively.

In the second stage, Bi-LSTM-CRF with AdaDelta optimizer achieved the best results with F1 of 75.5 %. Our results can be compared to Kucza et al. [10], who similarly tackled domain-specific term extraction using neural networks as a sequence labelling problem and with Bi-LSTM reached F1 score of 86.73 %. In this case, our best performing model in the second stage of the experiment (Bi-LSTM-CRF) fell short by 11.2 %. This rather big difference could be due to the much smaller amount of annotated terms: the dataset in [10] (GENIA and ACL RD-TEC 2.0) consisted of 78,567 annotated terms vs. our dataset with 1,258 annotated terms. In Kucza et al., [10] the experiment Bi-GRU outperformed their best performing LSTM model by 0.87 %, whereas in our tests, Bi-LSTM-CRF outperforms Bi-GRU-CRF by 1.1 %. In another experiment performed by Wang et al. [4], who similarly used a LSTM network for domain-specific term extraction, the best achieved result was 69.2 % on the ACL RD-TEC dataset which is 6.3 % less than our best performing Bi-LSTM-CRF network on the Lithuanian cybersecurity dataset.

The third stage of our experiment further improved the performance of Bi-LSTM model reaching F1 score of 78.6 %. This result was achieved using Bi-LSTM with BERT embeddings. Besides, our model using multilingual BERT embeddings, which was trained with monolingual data, managed to train itself on other 103 languages.

The results of our experiments suggest that for Lithuanian term extraction, the semi-supervised deep learning approach is a way to go. Although deep neural networks were trained on a very small amount of annotated data, the highest score almost reached 80 %. In order to achieve an even higher score, the quality and quantity of annotated data have to be increased. The automation of annotation of training data would greatly reduce the workload of annotators, thus reducing time consumption and increasing the amount of training data for deep neural networks. In bilingual and multilingual term extraction, multilingual BERT might be potentially helpful as it can reduce the amount of languages to be annotated. Therefore, BERT's multilingual capabilities should be more extensively explored. Also, other word embeddings such as ELMO, GPT-2, etc., and custom BERT embeddings should also be tested.

Acknowledgements

The research is carried out under the project "Bilingual automatic terminology extraction" funded by the Research Council of Lithuania (LMTLT, agreement No. P-MIP-20-282). The project is also included as a use case in COST action "European network for Web-centred linguistic data science" (CA18209).

References

[1] Alzaidy R, Caragea C, Giles CL. Bi-LSTM-CRF Sequence Labeling for Keyphrase Extraction from Scholarly Documents. The world wide web conference; 2019 May 13; p. 2551-2557.

[2] Basaldella M, Antolli E, Serra G, Tasso C. Bidirectional LSTM Recurrent Neural Network for Keyphrase Extraction. Italian Research Conference on Digital Libraries 2018 Jan 25; p. 180-187. Springer, Cham.

[3] Kucza M, Niehues J, Zenkel T, Waibel A, Stüker S. Term Extraction via Neural Sequence Labeling a Comparative Evaluation of Strategies Using Recurrent Neural Networks. Interspeech 2018:2072-2076.

[4] Wang R, Liu W, McDonald C. Featureless Domain-Specific Term Extraction with Minimal Labelled Data. Proceedings of the Australasian Language Technology Association Workshop 2016 Dec; p. 103-112.

[5] Sahrawat D, Mahata D, Kulkarni M, Zhang H, Gosangi R, Stent A, Sharma A, Kumar Y, Shah RR, Zimmermann R. Keyphrase Extraction from Scholarly Articles as Sequence Labeling using Contextualized Embeddings. arXiv preprint arXiv:1910.08840. 2019 Oct 19.

[6] Vasilev I, Slater D, Spacagna G, Roelants P, Zocca V. Python Deep Learning: Exploring Deep Learning Techniques and Neural Network Architectures with Pytorch, Keras, and TensorFlow. Packt Publishing Ltd; 2019 Jan 16.

[7] Lafferty J, McCallum A, and Pereira FCN. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001), p. 282-289.

[8] Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555. 2014 Dec 11.

[9] Almeida F, Xexéo G. Word embeddings: A survey. arXiv preprint arXiv:1901.09069. 2019 Jan 25.

[10] de Sousa RC, Lopes H. Portuguese POS Tagging Using BLSTM Without Handcrafted Features. In Iberoamerican Congress on Pattern Recognition 2019 Oct 28 (pp. 120-130). Springer, Cham.

[11] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems 2013; p. 3111-3119.

[12] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. 2018 Oct 11.

[13] Rokas, Aivaras. 2020. Automatinis kibernetinio saugumo terminų atpažinimas / Automatic Extraction of Cybersecurity Terms. Master thesis. Vytautas Magnus University. Lithuanian.

[14] Mi C, Yang Y, Wang L, Zhou X, Jiang T. A Neural Network Based Model for Loanword Identification in Uyghur. Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018) 2018 May.

[15] Sarkar D. Text analytics with Python: a practitioner's guide to natural language processing. Apress; 2019 May 21.

[16] Brownlee J. Deep Learning for Natural Language Processing: Develop Deep Learning Models for your Natural Language Problems. Machine Learning Mastery; 2017 Nov 21.

[17] Chiu JP, Nichols E. Named entity recognition with bidirectional LSTM-CNNs. Transactions of the Association for Computational Linguistics. 2016 Jul;4:357-70.

[18] Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991. 2015 Aug 9.

[19] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. 2014 Dec 22.

[20] Bottou L. Stochastic gradient descent tricks. Neural networks: Tricks of the trade 2012; Springer, Berlin, Heidelberg. p. 421-436.

[21] Zeiler MD. Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701. 2012 Dec 22.

[22] Hinton G, Srivastava N, Swersky K. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. Cited on. 2012 Feb;14(8).

[23] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research. 2011;12(Jul):2121-59.

[24] Pires T, Schlinger E, Garrette D. How multilingual is Multilingual BERT? arXiv preprint arXiv:1906.01502. 2019 Jun 4.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200601

Using Privacy-Transformed Speech in the Automatic Speech Recognition Acoustic Model Training

Askars SALIMBAJEVS [1]
Faculty of Computing, University of Latvia, Latvia

Abstract. Automatic Speech Recognition (ASR) requires huge amounts of real user speech data to reach state-of-the-art performance. However, speech data conveys sensitive speaker attributes like identity that can be inferred and exploited for malicious purposes. Therefore, there is an interest in the collection of anonymized speech data that is processed by some voice conversion method. In this paper, we evaluate one of the voice conversion methods on Latvian speech data and also investigate if privacy-transformed data can be used to improve ASR acoustic models. Results show the effectiveness of voice conversion against state-of-the-art speaker verification models on Latvian speech and the effectiveness of using privacy-transformed data in ASR training.

Keywords. Automatic speech recognition, voice conversion, privacy, anonymization, evaluation, automatic speaker verification

1. Introduction

Voice-operated technologies and tools have multiplied in recent years; voice is rapidly replacing touch or text as the main means of interaction with modern devices.

These technologies require huge amounts of speech data to reach state-of-the-art performance. The standard today is to store the voices of end users in the cloud and label them manually. There are few guarantees (if any) regarding how data stored in the cloud is used and will be used in the future by cloud service providers. This approach raises critical privacy concerns and has led to market and data concentration in the hands of big corporations. Dramatic improvements in speech synthesis [1], voice cloning [2] and speaker recognition [3] pose severe privacy and security threats to the users.

This resulted in a growth of interest on new voice privacy-preserving transformations and voice privacy evaluations [4,5,6,7]. Recently, the VoicePrivacy initiative was started to spearhead the effort to develop privacy preservation solutions for speech technology and create a new community. [8].

The advancement of privacy-preserving methods enables the collection of anonymized speech data and raises at least two questions:

[1]Corresponding Author: Askars Salimbajevs; Tilde, Vienibas gatve 75a, Riga, Latvia, LV-1004; E-mail: askars.salimbajevs@tilde.lv.

1. Do these methods work for smaller and less-researched languages like Latvian?
2. Can privacy-transformed speech data be used to improve automatic speech recognition (ASR) acoustic models?

Therefore, first, we investigate the applicability of one of the voice anonymization methods (VoiceMask, [7]) to the Latvian language and evaluate the performance of Automatic Speaker Verification (ASV) on original and privacy-transformed speech.

Next, we train several ASR acoustic models on speech data processed by VoiceMask method. It is common among researchers to evaluate the intelligibility of privacy-transformed speech by calculating the word error rate (WER) of the ASR system. However, to the best of our knowledge, there is no research on the training of ASR acoustic models on privacy-transformed speech.

The question on the applicability of the language independent voice anonymization method for Latvian speech data might seem naive, however, we think that it is important to perform such validation as it shows that ASR acoustic models will be trained on a substantially different type of data with much of the speaker personality removed.

2. Evaluation Setup

2.1. Privacy-Preserving Voice Transformation

For the voice anonymization, we use VoiceMask voice conversion technique, which is proposed by Qian et al.[7,9]. After using standard signal processing methods to compute spectral envelope, pitch, and aperiodicity features, VoiceMask modifies the spectral envelope through frequency warping. To provide privacy, this method is based on the composition of a quadratic function and a bilinear function using two different parameters. The inverse of this transformation is much more difficult to compute, and, therefore, more resistant to attacks.

2.2. Automatic Speaker Verification

Automatic Speaker Verification (ASV) is the authentication of individuals by doing voice analysis on speech utterances. ASV has two phases: enrollment and verification. During enrollment, the speaker's voice is recorded and typically a number of features are extracted to form a voice print, template, or model. In the verification phase, a speech sample or "utterance" is compared against a previously created voice print to verify the identity of the speaker.

In this paper, x-vectors[3] are used as speaker voice-prints and PLDA[10] for x-vector classification. Model training and evaluation is done using the Kaldi toolkit [11].

In experiments with speaker verification, we use the 100h Latvian Speech Recognition corpus [13]. From this corpus, we select 50 speakers (28 - male, 22 - female) which are used to create a test set.

The test set is split into enrollment data and trials. Enrollment data consists of approximately 60 seconds of audio per each speaker recorded in different conditions. The remaining audio recordings of these 50 speakers are used for trials if they are recorded in conditions different from the enrollment. As a result, the total number of utterances in the trial set is about 7,400, which results in 164,068 trials (trials are only between speakers of the same gender).

Audio recordings from other speakers in the corpus (approximately 1,700 speakers) are combined into a training set for the Latvian x-vector model. This training set is then augmented from 80h to about 375h by adding background noises and speed perturbations.

We evaluate ASV performance on the Latvian speech in the following settings:

- English x-vector and PLDA models trained on VoxCeleb2 dataset[12];
- X-vector and PLDA models trained on Latvian data only.

In each setting, we compare the performance of ASV system on original and privacy-transformed trial speech recordings using only original enrollment data.

2.3. Automatic Speech Recognition

The two crucial parts in the standard speech recogniser architecture are the acoustic model, which encodes pronunciation information, and the language model, which encodes grammar information. The open-source Kaldi toolkit [11] is used to train and evaluate Latvian ASR models.

For training of the ASR acoustic models, we use two speech corpora:

- The 100h Latvian Speech Recognition corpus [13] as a baseline training dataset.
- The Latvian Parliament Speech corpus [14] as additional data that is appended to ASR training dataset. A subset of 100 hours was taken to make the total length of both corpora comparable.

We train end-to-end Factorized Time Delay Neural Network (TDNN-F) [15] acoustic models with Lattice-Free Maximum Mutual Information (LF-MMI) [16] in a flat-start manner [17]. The model architecture and hyper-parameters are copied from the recipe for the Wall Street Journal (WSJ) dataset [18] which has a similar size (80 hours). Because Latvian has highly phonemic orthography, word pronunciation is modelled by treating each grapheme as a separate phoneme.

For language modelling, we employ a sub-word 4-gram language model which is trained on a 40 M sentence text corpus collected from Latvian web news portals. The model has a sub-word unit vocabulary generated using the Byte-Pair Encoding (BPE) method. N-grams are pruned to about 110 MB so that the decoding process can fit in 2 GB of RAM. Correct sub-word unit combination is ensured by a modified decoding graph [19].

Because the speech recordings which are appended to the baseline training dataset come from particular domain (Latvian Parliament session recording), testing was performed on two evaluation datasets: (1) an in-domain Saeima test set and (2) an out-domain test set of queries and short messages.

The in-domain Saeima test set contains 439 utterances (1 hour) from recordings of debates in the Parliament of Latvia from 2014 to 2016, containing contributions from about 300 different speakers. The recording time period does not overlap with the previously mentioned Latvian Parliament Speech corpus, as to guarantee that all utterances in the training and test sets are distinct (some overlap in speakers is still possible).

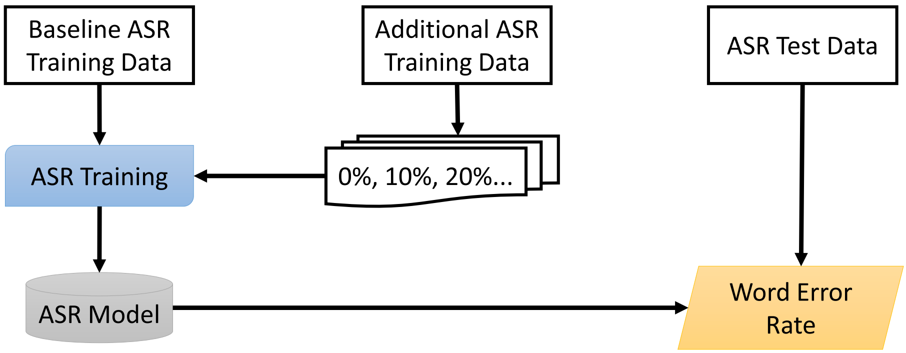


Figure 1. Baseline experimental setup of the speech recognition evaluation

To evaluate the effect of additional training data on the ASR performance in other domain, we use 1,159 utterances from real world data collected by production-level real-time Latvian ASR service.

2.4. Speech Recognition Using Anonymized Training Data

The experiment we devised compares two setups - one with and one without privacy-preserving speech transformation - with respect to the inclusion of additional training data on top of an existing baseline data set.

In the first setup, we add original, untransformed speech data to the baseline dataset by increments of 10 hours. We train a new ASR model for each increment and calculate the ASR performance on a test corpus (see Figure 1).

The second setup is similar to the first one. The only difference is that the additional training data is privacy-transformed prior to adding it to the baseline data (see Figure 2).

In both experiments, we add the respective portions of additional speech training data to the baseline dataset in the same order, i.e. ASR models are trained on exactly the same speech recordings in either setup. Also, we use untransformed test corpus in evaluation, because for privacy-preservation, we plan to operate ASR models on-device and adapt to the voice of the user. Voice anonymization will make such adaptation impossible.

3. Results

3.1. Speaker Verification

The speaker verification evaluation results presented in Table 1 show that VoiceMask voice transformation is effective on Latvian speech and can conceal the user identity (EER of ASV systems increased more than 3 times). This result corresponds to findings on the English data [4].

We can also observe that the state-of-the-art English x-vector model trained Vox-Celeb2 corpus is language-independent and performs better than model trained on the Latvian data. We believe this is due to the fact that VoxCeleb2 corpus is more than 20 times larger than corpus used to train Latvian models.

Table 1. Speaker recognition evaluation on untransformed and privacy-transformed speech

Training data	Equal error rate %	
	Original speech	Transformed speech
VoxCeleb2	10.4	32.6
Latvian	11.8	32.6

3.2. Speech Recognition

First, ASR quality evaluation of models trained using different amounts of additional data was performed in-domain Saeima test set. The results presented in Table 2 indicate that both types of additional data improve the WER and the difference between adding transformed and untransformed data is small (5 % relative between best results of both methods).

Next, the evaluation on the second test set was performed to check if additional data can improve ASR performance on out-of-domain data (see Table 3). The results are quite noisy which may be attributed to a mismatch between the domain of the original training set and the additional data. Still, it is possible to make three main observations:

- Additional data improves speech recognition quality;
- Adding untransformed data helps to achieve better WER;
- The difference between adding untransformed and privacy-transformed data is small (2 % relative between best results of both methods).

There is a noticeable WER improvement after adding the first 10 hours of privacy-transformed data which seems suspicious. Interestingly, adding the same 10h of untransformed data only has a similar effect when evaluating on in-domain test data, but not on out-of-domain data. As an additional experiment, we decided to take the last 10h of additional data instead of first 10h and to retrain the system. This time the WER improvement was smaller and fitted with other results. Therefore, we believe this result

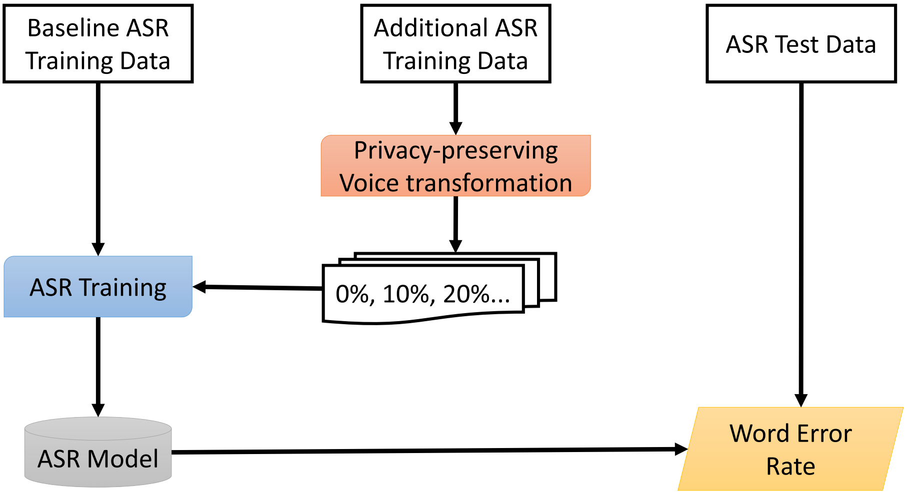


Figure 2. Experimental setup for the evaluation of the ASR trained using privacy-transformed data.

Table 2. Comparison of in-domain WER using untransformed and privacy-transformed speech in ASR training

New data, h	Word error rate %	
	Original speech	Transformed speech
0	13.6	13.6
10	12.8	12.7
20	12.5	13.3
30	12.5	13.1
40	12.4	13.1
50	11.9	13.0
60	11.9	13.0
70	12.1	12.6
80	12.2	12.7
90	11.8	12.4
100	11.8	12.8

Table 3. Comparison of out-domain WER using untransformed and privacy-transformed speech in ASR training

New data, h	Word error rate %	
	Original speech	Transformed speech
0	28.3	28.3
10	27.7	26.9
20	27.4	27.6
30	27.5	27.5
40	26.7	26.5
50	25.9	27.2
60	26.5	26.9
70	27.5	27.5
80	26.5	26.7
90	26.5	27.5
100	26.4	27.3

can be explained by irregularities in the additional data, some subsets of which are more beneficial than others.

4. Conclusions

In this paper, we investigated the use of VoiceMask voice anonymization method to protect the privacy of Latvian speakers by concealing their identity and also feasibility of using such transformed recordings in the ASR acoustic model training.

To the best of our knowledge this is a first evaluation of this kind. During the preparation of the paper, much better methods were created within Voice Privacy Challenge [8]. These will have to be evaluated in the similar way. Still, even at this early stage, the evaluation has now given us important insights.

Speaker verification experiments showed that VoiceMask method works for Latvian speech and can provide reasonable protection against attacks without knowledge of the anonymization method. This result also show that the privacy-transformed data is substantially different from the original and presents a new challenge to ASR acoustic model training.

We found that using such privacy-transformed in-domain data for acoustic model training resulted in a clear benefit for recognition quality. With a test set from another domain, the benefits of adding more training data suffered from noise artifacts. However, an improvement in WER still can be observed. While in both cases the benefit is smaller than when using the original speech data, we believe that this result proves that privacy-transformed speech data can be used to improve ASR acoustic models, therefore, allowing to collect the speech data from end users for training while preserving some privacy.

5. Acknowledgements

The work presented in this paper has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 825081 and under the name COMPRISE (Cost-effective, Multilingual, Privacy-driven voice-enabled Services).

References

[1] Székely É, Henter GE, Beskow J, Gustafson J. Spontaneous conversational speech synthesis from found data. In: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. 2019.
[2] Vestman V, Kinnunen T, González Hautamäki R, Sahidullah M. Voice Mimicry Attacks Assisted by Automatic Speaker Verification. Comput Speech Lang. 2020;
[3] Snyder D, Garcia-Romero D, Sell G, Povey D, Khudanpur S. X-Vectors: Robust DNN Embeddings for Speaker Recognition. In: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. 2018.
[4] Srivastava BML, Vauquier N, Sahidullah M, Bellet A, Tommasi M, Vincent E. Evaluating Voice Conversion-based Privacy Protection against Informed Attackers. In: ICASSP 2020 - 45th International Conference on Acoustics, Speech, and Signal Processing [Internet]. Barcelona, Spain; 2020.
[5] Lal Srivastava BM, Bellet A, Tommasi M, Vincent E. Privacy-preserving adversarial representation learning in ASR: Reality or illusion? In: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. 2019.
[6] Ribaric S, Ariyaeeinia A, Pavesic N. De-identification for privacy protection in multimedia content: A survey. Signal Process Image Commun. 2016;
[7] Qian J, Du H, Hou J, Chen L, Jung T, Li XY. Hidebehind: Enjoy voice input with voiceprint unclonability and anonymity. In: SenSys 2018 - Proceedings of the 16th Conference on Embedded Networked Sensor Systems. 2018.
[8] Tomashenko N, Srivastava BML, Wang X, Vincent E, Nautsch A, Yamagishi J, et al. The VoicePrivacy 2020 Challenge Evaluation Plan.
[9] Qian J, Du H, Hou J, Chen L, Jung T, Li X-Y, et al. VoiceMask: Anonymize and Sanitize Voice Input on Mobile Devices. 2017.
[10] Ioffe S. Probabilistic linear discriminant analysis. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2006.
[11] Povey D, Ghoshal A, Boulianne G, Burget L, Glembek O, Goel N, et al. The Kaldi Speech Recognition Toolkit. In: IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society; 2011.
[12] Chung JS, Nagrani A, Zisserman A. VoxceleB2: Deep speaker recognition. In: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. 2018.
[13] Pinnis M, Auziņa I, Goba K. Designing the Latvian Speech Recognition Corpus. In: Proceedings of the 9th edition of the Language Resources and Evaluation Conference (LREC'14). 2014. p. 1547–53.
[14] Salimbajevs A. Creating Lithuanian and Latvian Speech Corpora from Inaccurately Annotated Web Data. In: Calzolari N, Choukri K, Cieri C, Declerck T, Goggi S, Hasida K, et al., editors. Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018 [Internet]. European Language Resources Association (ELRA); 2018.
[15] Povey D, Cheng G, Wang Y, Li K, Xu H, Yarmohamadi M, et al. Semi-orthogonal low-rank matrix factorization for deep neural networks. In: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. 2018.
[16] Povey D, Peddinti V, Galvez D, Ghahremani P, Manohar V, Na X, et al. Purely sequence-trained neural networks for ASR based on lattice-free MMI. In: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. 2016. p. 2751–5.
[17] Hadian H, Sameti H, Povey D, Khudanpur S. End-to-end speech recognition using lattice-free MMI. In: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. 2018.

[18] Paul, D. B., & Baker, J. M. (1992). The design for the Wall Street Journal-based CSR corpus. In Proceedings of the workshop on Speech and Natural Language (pp. 357-362).
[19] Smit P, Virpioja S, Kurimo M. Improved Subword Modeling for WFST-Based Speech Recognition. In: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. 2017. p. 2551–5. Available from: http://dx.doi.org/10.21437/Interspeech.2017-103

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200602

Pretraining and Fine-Tuning Strategies for Sentiment Analysis of Latvian Tweets

Gaurish THAKKAR [a,1] and Mārcis PINNIS [b,c]
[a] *Faculty of Humanities and Social Sciences, University of Zagreb, Ul. Ivana Lučića 3, 10000, Zagreb, Croatia*
[b] *Tilde, Vienības gatve 75A, Riga, Latvia, LV-1004*
[c] *University of Latvia, Raiņa bulv. 19-125, Riga, Latvia, LV-1586*

Abstract. In this paper, we present various pre-training strategies that aid in improving the accuracy of the sentiment classification task. At first, we pre-train language representation models using these strategies and then fine-tune them on the downstream task. Experimental results on a time-balanced tweet evaluation set show the improvement over the previous technique. We achieve 76% accuracy for sentiment analysis on Latvian tweets, which is a substantial improvement over previous work.

Keywords. Sentiment analysis, word embeddings, BERT, Latvian

1. Introduction

Sentence-level sentiment analysis (SA) aims to classify the opinion expressed by the author into either a Positive, Negative, or Neutral class. Recently, transformer-based neural networks [1] pre-trained using self-supervision [2, 3] have shown state-of-the-art performance on downstream tasks [4]. However, most of these findings have been reported for highly resourced languages.

In this work, we focus on improving the performance of SA for Latvian tweets using a model of pre-trained multilingual Bidirectional Encoder Representations from Transformers (mBERT). We experiment further by pre-training the model with in-domain data. mBERT treats Unicode emoticons as out-of-vocabulary words. We propose adding them to the vocabulary of the model and repeating the pre-training and fine-tuning cycle. We also compare A-Lite-BERT (ALBERT) [5] and ELECTRA [6] models as light-weight variants of mBERT which we train from scratch on Latvian tweets. We release all the pre-trained language representation models and the models along with the code[2].

2. Related Work

Pre-trained word-embeddings [7, 8] have been studied extensively for improving sentiment classification scores [9]. For Latvian, Pinnis [10] performed experiments on Lat-

[1] Corresponding Author: Gaurish Thakkar; E-mail: gthakkar@m.ffzg.hr.
[2] https://github.com/thak123/bert-twitter-sentiment

vian tweets with a wide range of classifiers and features. Peisenieks and Skadiņš [11] analysed machine translation as a viable tool for performing SA for Latvian tweets. A recent study [12] performed pre-training of the BERT model from scratch for classifying sentiment of tweet representations. Earlier techniques [13, 14] used Pointwise Mutual Information (PMI) with Information Retrieval (IR) as well as Naive Bayes to classify multi-domain tweets.

3. Pre-training and Fine-tuning Strategies

We follow 3 different pre-training strategies:

- First, an existing pre-trained model trained on a large corpus is trained further on the in-domain corpus.
- Second, the model trained using the previous method is trained further by adding new tokens into the existing vocabulary. Our initial experiments showed that emoticons are treated as unknowns (*[UNK]*) as they are not present in the vocabulary of the pre-trained model. Since the mBERT model was trained on texts from Wikipedia, it is obvious to lack smileys in the text. We hypothesise that emoticons are sentiment-bearing tokens and hence important as features.
- Lastly, we pre-train (ALBERT and ELECTRA) models from scratch. The models may have vocabularies learned from the data or they may use vocabularies from existing models. We perform this step to compare the performance of pre-trained models with the models that are trained from scratch.

Using the various annotated datasets, we perform fine-tuning on the downstream task of sentiment analysis using all the models described previously.

4. Data

In our experiments, we use the sentiment annotated corpora curated by Pinnis [10]. The corpora are:

1. *Gold*: a corpus consisting of 6,777 human-annotated Latvian tweets from the period of August 2016 till November 2016.
2. *Peisenieks*: a corpus consisting of 1,178 human-annotated Latvian tweets created by Peisenieks and Skadiņš [11].
3. *Auto*: three sets of tweets from the period of August 2016 till July 2018 automatically annotated based on sentiment-identifying emoticons that are present in the tweets – 23,685 tweets with emoticons, 23,685 tweets with removed emoticons, and 47,370 tweets with both present and removed emoticons.
4. *English*: a corpus of 45,530 various human-annotated English tweets from various sources that were machine-translated into Latvian.
5. A time-balanced evaluation set that consists of 1,000 tweets from the period of August 2016 till July 2018.

To pre-train word embeddings, we use also the Latvian tweets from the Latvian Tweet Corpus[3] [10]. The corpus consists of 4,640,804 unique Latvian tweets that have been collected during the time-frame from August 2016 till March 2020.

5. Experiments

In this section, we describe the experimental setup for sentiment analysis. Our experimental setup consists of pre-training and fine-tuning steps. We perform the following pre-processing steps on the text:

1. Tokenization.
2. Removal of URLs.
3. Replacement of consecutive user mentions with a single mention.
4. User mention replacement with a placeholder (‘*mention_i*’ where the *i* stands for the ith mention in the tweet).
5. Lower-casing of the whole tweet.

5.1. Pre-training

We employ the script[4] available in the *transformers*[5] library to continue training the uncased version of the multilingual-BERT (mBERT) model. mBERT models 102 languages, which also include Latvian and other Baltic languages. This step uses the 4.6 million unique tweets from the Latvian Tweet Corpus described above. The corpus is split into train and eval and is pre-trained for 7 epochs. In the case of unknown tokens, there are around 5 thousand unique *[UNK]* tokens in the Gold dataset (train split only), which mainly are emoticons. Therefore, we sort the highest occurring emoticons and add 70 of them to the model vocabulary. Then, we perform one more cycle of pre-training with the new vocabulary in the network. Using the same Latvian Tweet Corpus, we train two more models namely ALBERT and ELECTRA from scratch[6]. This is done by tokenizing the whole corpus and joining each of the two consecutive tweets together as examples to be trained. These are used to train a discriminator to decide if each token in the corrupted input was replaced by a generator sample or not. Both embedding_size and hidden_size are set to 256.

All models use the same vocabulary as that of the pre-trained uncased mBERT model, which uses sentence-piece [15] as the method of tokenization and word splitting. We use a batch size of 16. For the pre-training step, the process was stopped once the perplexity score of ≈ 3 was achieved on the validation split of the dataset.

5.2. Fine-tuning

For this step, We have the following pre-trained language representation models:

---

[3] https://github.com/pmarcis/latvian-tweet-corpus
[4] https://github.com/huggingface/transformers/blob/master/examples/language-modeling/run_language_modeling.py
[5] https://github.com/huggingface/transformers
[6] https://github.com/shoarora/lmtuners/tree/master/lmtuners

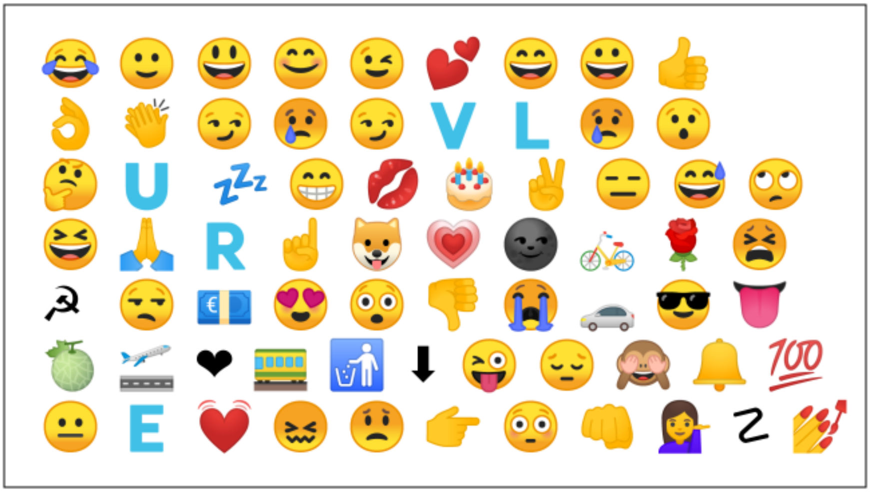


Figure 1. Examples of (non-exhaustive) list of added emoticons

1. mBERT - vanilla version.
2. mBERT - pre-trained on the Latvian Tweet Corpus.
3. mBERT - pre-trained on the Latvian Tweet Corpus plus emoticons added to the vocab.
4. ALBERT and ELECTRA.

For each tweet, the vector representing the [CLS] token is extracted and passed to the classification layer. All settings use a single 3-class softmax classification layer (*positive, negative, and neutral*) with a dropout value of 0.2. We employ a maximum sequence length of 150 and pad the shorter tweets. We randomly sample 1000 records as the validation set. Finally, we report the test accuracy on the model with the highest validation accuracy. No hyper-parameter tuning on the model is performed. Except for the emoticon augmented model, all other representation models share the same vocabulary.

6. Results

The accuracy results of the experiments are presented in Table 1. We compare our scores with previously reported scores by [10] (shown in the first column). The column named 'Base' shows the results of using the mBERT model directly in the fine-tuning task. The next column 'Pre' lists scores when the model is additionally pre-trained with in-domain Twitter data. The results show that there is an improvement over the Perceptron baseline when the mBERT model is used. There is also further improvement when we pre-train the existing model with the in-domain corpus. The results of the experiment where we incorporate emoticons into the vocabulary are presented in column 'Pre+Emo'. This setup improves over the in-domain pre-training setup except in two cases.

The best results were obtained when the *Gold* and *Auto (with ☺)* datasets were combined to train the sentiment analysis model. We see a drop in performance for all models trained on the *Gold+Auto (no ☺)* and *Gold+Auto (both)* datasets. Even though

Table 1. Results of the classifier (Accuracy Scores)

Dataset	Perceptron [10]	mBERT			ALBERT	ELECTRA
		Base	Pre	Pre+Emo		
Gold	0.661	0.678	**0.756**	0.754	0.661	0.711
Gold+Peisenieks	0.676	0.692	0.747	**0.764**	0.698	0.706
Gold+Auto (with ☺)	0.624	0.679	**0.769**	0.748	0.649	0.68
Gold+Auto (no ☺)	0.512	0.523	0.648	**0.660**	0.483	0.621
Gold+Auto (both)	0.487	0.526	0.618	**0.657**	0.509	0.564
Gold+English	0.613	0.698	0.692	**0.720**	0.669	0.684

ELECTRA has a lower number of model parameters (compared to mBERT), it is still able to perform better than the vanilla mBERT version.

7. Error Analysis

We performed error analysis using the best-performing sentiment analysis model (i.e., the mBERT model that was additionally pre-trained on the Latvian Tweet Corpus and fine-tuned using the *Gold+Auto(with ☺)* corpus. To aid the error analysis, we visualised the test set by plotting the individual tweet representations and their predictions as scatter plots. For every tweet in the test set, we use the *[CLS]* token, which is a vector of length 768, and project it down to 50 dimensions using Principal Component Analysis (PCA) [16]. The principal components are further reduced to 2 dimensions using t-SNE [17]. Each of the points is plotted as nodes. We color each of the correctly predicted tweets in green for positive, red for negative, and blue for neutral. The incorrect predictions are colored in black with the correct class and predicted class as the node text. The visualisation is depicted in Figure 2.

We started the error analysis by investigating whether we can identify if messages that are grouped in clusters that are formed in the tweet representation and prediction scatter plot have common characteristics (e.g., common syntactic structures). This did not yield positive results as the messages close to each other often contained different syntactic structures and even different vocabularies. Therefore, we continued by analysing what common characteristics can be found among a random subset of 100 misclassified tweets. From the analysis, we made the following observations:

- 32 of the misclassified tweets were ambiguous to the extent where external world knowledge would be necessary to decide on the polarity of the messages.
- 17 of the misclassified tweets featured words of the wrongly selected polarity within the messages, which may indicate that the model may have learned to use lexical polarity-identifying cues to aid classification. However, it would require further analysis to validate this hypothesis.
- 13 of the misclassified tweets featured sarcastic expressions within the messages. All of the sarcastic tweets were negative tweets. This amounts to almost 50% of all misclassified negative tweets.
- 12 of the misclassified tweets featured possible multiple polarities within the messages.
- 4 tweets featured double negation.

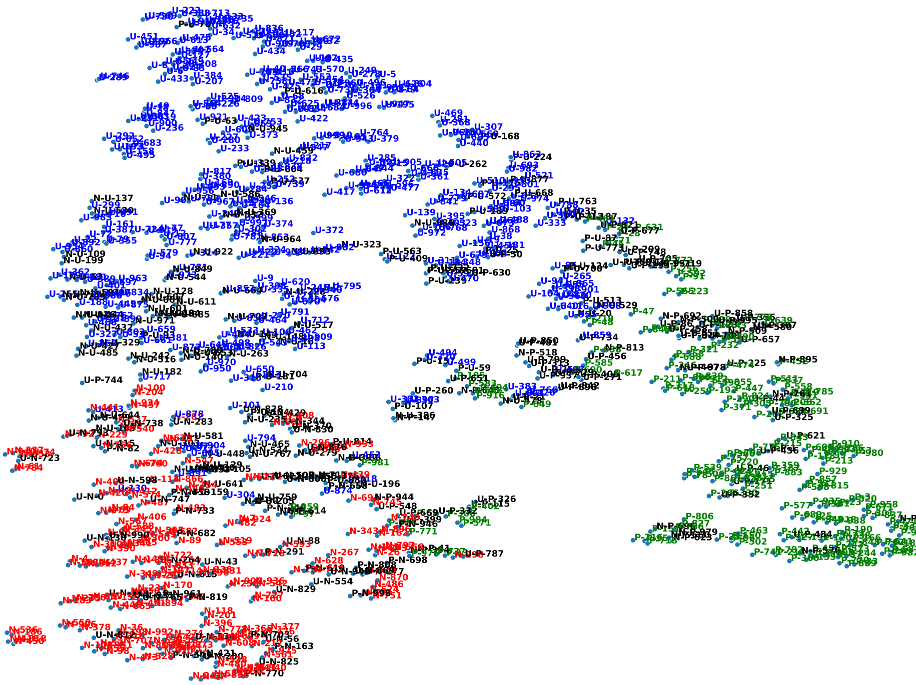


Figure 2. Tweet representation and prediction scatter plot

- 3 tweets featured spelling mistakes and a lack of diacritics that could have triggered misclassification.
- For the remaining 19 tweets, we did not identify common characteristics.

8. Conclusion

In this paper, we presented our work on improving Latvian SA for tweets. Our experiments allowed us to achieve the increase in performance when pre-training word embedding models with in-domain unlabelled data and fine-tuning the models on relatively small supervised datasets. The results surpass previous work on SA for Latvian. As future work, handling tweets with mixed emotions will be investigated. Furthermore, error analysis indicated that a large proportion of misclassified tweets can be attributed to ambiguous and sarcastic tweets for which analysis and consideration of tweet history could potentially allow expanding the context available for classification and, thereby, allow performing better-informed classification. Error analysis also raised a hypothesis that the fine-tuned models may have learned to focus on lexical polarity-identifying cues when deciding on which class to assign to tweets. This needs to be validated in further research. Lastly, there are still avenues of improvements to ELECTRA model pre-training evident that have not been explored and could be investigated in future work.

9. Acknowledgments

The work presented in this paper has received funding from the European Union's Horizon 2020 research and innovation programme under the *Marie Sklodowska-Curie* grant *agreement no. 812997* and under the name CLEOPATRA (Cross-lingual Event-centric Open Analytics Research Academy). This research has been supported by the European Regional Development Fund within the joint project of SIA TILDE and University of Latvia "Multilingual Artificial Intelligence Based Human Computer Interaction" No. 1.1.1.1/18/A/148. The work has been carried out during Gaurish Thakkar's internship in Tilde.

References

[1] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Advances in neural information processing systems; 2017. p. 5998–6008.
[2] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805. 2018.
[3] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:190711692. 2019.
[4] Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:191010683. 2019.
[5] Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R. Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:190911942. 2019.
[6] Clark K, Luong MT, Le QV, Manning CD. Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:200310555. 2020.
[7] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:13013781. 2013.
[8] Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP); 2014. p. 1532–1543.
[9] Severyn A, Moschitti A. Twitter sentiment analysis with deep convolutional neural networks. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2015. p. 959–962.
[10] Pinnis M. Latvian Tweet Corpus and Investigation of Sentiment Analysis for Latvian. In: Human Language Technologies – The Baltic Perspective - Proceedings of the Seventh International Conference Baltic HLT 2018. Tartu, Estonia: IOS Press; 2018. p. 112–119.
[11] Peisenieks J, Skadins R. Uses of Machine Translation in the Sentiment Analysis of Tweets. In: Baltic HLT; 2014. p. 126–131.
[12] Azzouza N, Akli-Astouati K, Ibrahim R. TwitterBERT: Framework for Twitter Sentiment Analysis Based on Pre-trained Language Model Representations. In: International Conference of Reliable Information and Communication Technology. Springer; 2019. p. 428–437.
[13] Gulbinskis I. Digitālo tekstu sentimenta analīze. 2010.
[14] Špats G, Birzniece I. Opinion Mining in Latvian Text Using Semantic Polarity Analysis and Machine Learning Approach. Complex Systems Informatics and Modeling Quarterly. 2016;(7):51–59.
[15] Kudo T, Richardson J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations; 2018. p. 66–71.
[16] Wold S, Esbensen K, Geladi P. Principal component analysis. Chemometrics and intelligent laboratory systems. 1987;2(1-3):37–52.
[17] Maaten Lvd, Hinton G. Visualizing data using t-SNE. Journal of machine learning research. 2008;9(Nov):2579–2605.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200603

Large Language Models for Latvian Named Entity Recognition

Rinalds VĪKSNA [a,b,1], Inguna SKADIŅA [a,b]
[a] *Tilde*
[b] *Faculty of Computing, University of Latvia, Latvia*

Abstract. Transformer-based language models pre-trained on large corpora have demonstrated good results on multiple natural language processing tasks for widely used languages including named entity recognition (NER). In this paper, we investigate the role of the BERT models in the NER task for Latvian. We introduce the BERT model pre-trained on the Latvian language data. We demonstrate that the Latvian BERT model, pre-trained on large Latvian corpora, achieves better results (81.91 F1-measure on average vs 78.37 on M-BERT for a dataset with nine named entity types, and 79.72 vs 78.83 on another dataset with seven types) than multilingual BERT and outperforms previously developed Latvian NER systems.

Keywords. Named entity recognition, NER, Latvian language, BERT

1. Introduction

Recently developed pre-trained language representation models have demonstrated significant improvement in many natural language processing tasks. The most popular are ELMo [1], BERT [2] and RoBERTa [3]. The BERT model has shown the state-of-the art performance for tasks of named entity recognition (NER), question answering, classification, and others. The multilingual BERT model (M-BERT)[2], pre-trained on Wikipedia texts in 104 languages, has demonstrated good results in zero-shot cross-lingual model transfer, where a model trained on one language (presumably, with large annotated corpora) is evaluated on another language [4]. However, some recent publications show that the monolingual BERT model could achieve significantly better results compared to the multilingual [5], [6], [7].

One of the tasks where pre-trained language representation models have been successfully applied is named entity recognition. Traditionally, NER is understood as identification of the text spans containing named entities and classifying them into predefined categories (e.g. person names (John, Barack Obama, etc.), organizations (BMW, IBM, etc.), locations (Riga, Washington, etc.) and other). NER serves as a basis for many natural language understanding tasks such as semantic annotation, question answering, ontology population, and opinion mining [8].

---

[1] Corresponding Author: Rinalds Vīksna; Tilde, Vienibas gatve 75a, LV-1004, Riga, Latvia; E-mail: rinalds.viksna@tilde.lv.

[2] https://github.com/google-research/bert.

Most of the research on the role of pre-trained language representation models for NER task has been performed on resource-rich languages, e.g., English, German, and Chinese [9], [10], while less attention has been paid to the less-resourced and morphologically rich languages like Persian [11], Finnish [6], and Portuguese [7].

In this paper, we examine the role of the BERT models in the NER task for the Latvian language. We introduce the Latvian BERT model and compare its performance to M-BERT in the NER task. Several configurations of BERT have been pre-trained on different corpora, adapted, and evaluated on two different datasets. Developed NER models have been compared with previously developed solutions (where possible). We demonstrate that pre-trained BERT models significantly improve the overall performance of NER and NER using the Latvian BERT model outperforms M-BERT.

2. Related Work

The first named entity recognizer for Latvian and Lithuanian TildeNER uses the StanfordNER random field classifier for training and tagging [12]. It also features a bootstrapping module. Evaluation on the test set using 7 class tagset (organization, person name, location, product, date, time, money) yielded F-measure of 60.19 for Latvian and 65.12 for Lithuanian. TildeNER was also compared to StanfordNER using a comparable corpus of 10 documents, where TildeNER achieved an F-measure of 56.46 while detecting location entities, 61.63 for person entities, and 65.71 detecting organizations.

The Latvian NLP tool pipeline [13] contains a NER module that tags 7 named entity classes: person, organization, geopolitical entity, location, product, time and event. It was trained on the Latvian Multilayer Corpus for NLU [14]. Data[3] [15] were serialized using a modified CoNLL-2003[4] data format, which supports hierarchical named entity annotation. Using a bidirectional LSTM neural network with CRF layer and word embeddings, the model achieves a 74.0 F1 measure on average. The model demonstrates good results on person entities (85.2 F1), while performing poorly on events (20.0 F1) and locations (45.1 F1).

Arkhipov et.al. [16] used multilingual BERT to initialize pre-training of the BERT model for Russian, Bulgarian, Czech, and Polish. The pre-trained model was extended with CRF layer to recognize five classes of entities: persons, locations, organizations, events, and products. Using additionally pre-trained SlavicBERT model together with additional CRF layer, they achieved an F1 score of 87.3 for Russian, 93.2 for Polish, 93.9 for Czech, and 87.2 for Bulgarian.

Souza et.al [7] also used a multilingual BERT model with a CRF layer to detect named entities in Portuguese. Authors compared two transfer learning approaches: feature-based and fine-tuning based. Feature-based approach uses Bi-LSTM layer and linear layer, and BERT is used to obtain word embeddings. In the fine-tuning approach, the classifier is a linear layer and all weights are updated during training. The best model uses fine-tuning approach with CRF and has achieved an F1 score of 74.15 against 70.33 F1 score obtained using a baseline Bi-LSTM.

[3] https://github.com/LUMII-AILab/FullStack/tree/master/NamedEntities.

[4] https://www.clips.uantwerpen.be/conll2003/ner/.

Virtanen et al. [6] pre-trained Finnish monolingual BERT model (FinBERT) on 234M sentences (about 3.3B tokens) crawled from the web and news. The FinBERT was evaluated on the NER task against uncased FinBERT and multilingual BERT models using the FiNER dataset, which includes nested named entities. NER model was built using the FinBERT as a base with a dense layer on top. This model achieved an F1 score of 92.4 on in-domain data and 81.47 on out of domain test set, while the multilingual BERT achieved F1 scores of 90.29 and 76.15, respectively.

3. Latvian BERT Model

3.1. Data Collection and Processing

For pre-training the Latvian BERT, unlabeled data were acquired from different sources: the EUbookshop[5], JRC-Acquis[6], Latvian Wikipedia, and various European and Latvian websites. Crawled data were then cleaned: boilerplate content and HTML tags were removed, texts converted into UTF-8 encoding, and language identification performed (documents with less than 80 % of Latvian content were removed). Documents containing long sequences of short segments or numbers were removed as well.

After cleaning, the text corpus used for BERT pre-training contained 124 million sentences or 1.6 billion tokens. In comparison, the English BERT model was pre-trained on 3.3 billion words from BookCorpus (800 million words) and English Wikipedia (2,500 million words), the Portuguese BERT was pre-trained on 2.6 billion tokens and the Finnish BERT was pre-trained on 3.3 billion tokens.

3.2. Pre-training

From the collected corpus, the byte pair encoding (BPE) vocabulary [17] was created using the sentencepiece [18] and converted to the wordpiece format used by BERT. BPE vocabulary was generated using a cased version of the corpus and its size was set to 30,000 word-pieces. BERT scripts were used to create pre-training examples with a sequence length of 128, while other parameters were set to match the original BERT model [2]. The model was pre-trained for 4M steps using learning rate 5e-5 and 10,000 warmup steps.

We also pre-trained the multilingual BERT model for 1M steps using Latvian data to evaluate the usefulness of additional pre-training.

4. NER Systems

Three NER systems using different BERT models have been trained and evaluated:

- “Multi-base”: NER model that uses the multilingual BERT model;
- “Multi-updated”: NER model that uses a multilingual model additionally pre-trained with Latvian data;

[5] http://opus.nlpl.eu/EUbookshop.php.

[6] http://opus.nlpl.eu/JRC-Acquis.php.

- “lv-base”: NER model trained using Latvian BERT.

4.1. Datasets for NER Training

Only two rather small datasets are available and were used in our experiments: proprietary TildeNER dataset and the named entity annotated layer of the publicly available Latvian Multilayer Corpus (the AILab dataset)[7].

Table 1. TildeNER dataset statistics

NE type	NE count		
	Manually created data	**Bootstrapped data**	**TOTAL**
DATE	1,590	791	2,381
LOCATION	2,611	1,759	4,370
MONEY	289	671	960
ORGANIZATION	1,649	638	2,287
PERSON	1,037	1,282	2,391
PRODUCT	866	233	1,099
TIME	353	125	478
TOTAL	**8,395**	**5,499**	**13,966**

TildeNER dataset [12] consists of two parts (Table 1). The initial dataset was manually created for training and evaluation of the Latvian NER system. Annotation was performed by 2 annotators, while the third annotator resolved the disagreement. Additional annotated data were bootstrapped during the development process and verified by a human annotator.

Table 2. Entity count in Multilayer Corpus for NLU dataset

NE type	Entity count
PERSON	3,104
GPE	2,031
ORGANIZATION	1,847
TIME	1,227
PRODUCT	293
LOCATION	677
EVENT	259
ENTITY	215
MONEY	44
TOTAL	**9,697**

[7] https://github.com/LUMII-AILab/FullStack/tree/master/NamedEntities

The second dataset is from the Balanced State-of-the-Art Multilayer Corpus for NLU [15]. It contains 3,947 paragraphs with 9,697 outer and 944 inner entities. In this work, we use only outer entities (Table 2).

For NER model training, the corpus was transformed into a CONLL-2003 format. To enable comparison with the NER model developed by Znotiņš and Cīrule [13], classes "Money" and "Entity" were labeled as "O" – Other.

4.2. Training

We use a dense+crf layer on top of the BERT for classification (Figure 1). Words are split into word pieces using BERT tokenizer, and Latvian wordpiece vocabulary. When BERT tokenizer splits words into subword tokens, we label only the first subword of each word according to BIO (identifies the Beginning, the Inside, and the Outside of text segment) labeling scheme, while other subwords get label "X". This additional label "X" is removed later, as output is words. The NER model is trained for 12 epochs, using sequence length 128, train batch size 4, and learning rate 2e-5.

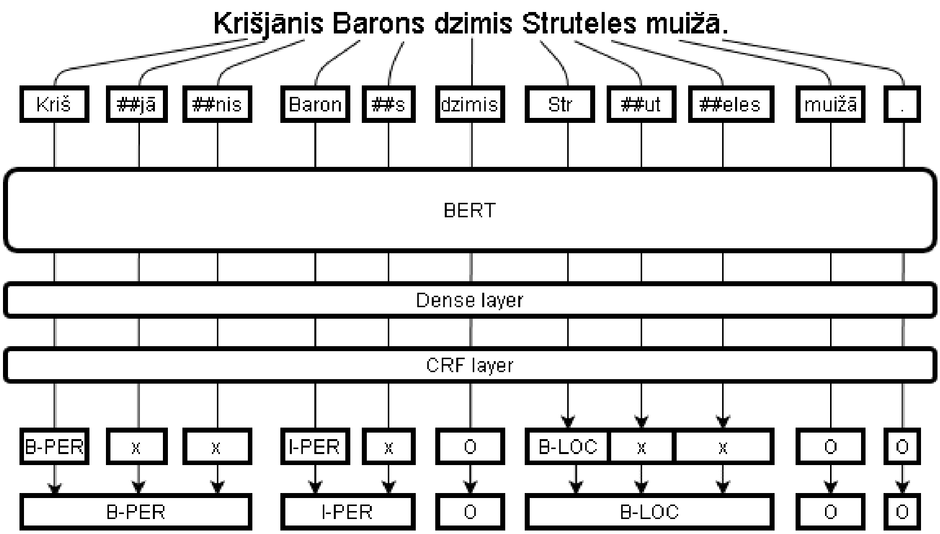


Figure 1. NER model architecture

5. Results and Evaluation

At first, we trained the NER model using data from the AILab dataset. An earlier version of this dataset was also used to train the NER model for NLP-PIPE [13][8]. This and other Latvian NER systems were evaluated by F1-measure using the CoNNL-2003 evaluation script[9]. Table 3 summarizes evaluation results, demonstrating that NER

[8] This NER model recognizes 7 out of 9 entity classes presented in the dataset.

[9] https://www.clips.uantwerpen.be/conll2002/ner/bin/conlleval.txt

model trained with the Latvian BERT (lv-base) outperforms the model, that was trained using multilingual BERT (Multi-base). The model that was trained using additionally pre-trained multilingual BERT (Multi-updated) performed poorly and thus was not used in further experiments. Probably, the learning rate 2e-5 was too high and additional pre-training did harm to the model. All models perform poorly in detecting classes that have little training data (less than 300 examples), i.e., product, event, entity, and money.

Table 3. Evaluation results (F1) on the AILab dataset

NE type	Multi-updated	Multi-base	lv-base	NLP-PIPE [13]
GPE	84.77	88.18	**89.66**	79.0
ENTITY	30.77	35.59	**44.68**	
EVENT	51.43	48.6	**59.46**	20.0
LOCATION	56.86	61.59	**67.79**	45.1
MONEY	**15.38**	11.11	0	
ORGANIZATION	71.9	77.91	**81.7**	78.5
PERSON	89.82	91.99	**94.91**	85.2
PRODUCT	45.61	**65.17**	64	40.0
TIME	64.37	63.14	65.64	**71.7**
F1	**75.20**	**78.37**	**81.91**	**74**

During experiments, we noticed that sometimes GPE and location categories are very similar and overlapping entities, and some classes are very small. Therefore, we decided to test model performance for only 4 classes. GPE and location were merged into LOCATION; person and organization classes were kept, and all the rest were merged in MISC class. The evaluation results in Table 4 show that with 4 classes, the Latvian BERT performs even better, achieving on average F1 score of 84.82.

Table 4. NER evaluation results (F1) on the AILab dataset (4 classes)

NE type	Multi-base	lv-base
LOCATION	86.23	90.49
MISC	63.11	65.93
ORGANIZATION	76.9	80.48
PERSON	91.63	95.34
F1	**81.1**	**84.82**

As next, we trained NER systems with the TildeNER dataset. As it is demonstrated in Table 5, for this dataset, the Latvian BERT achieves better F1 measure in total, but multilingual BERT also performs quite well. Although our results are not directly comparable with TildeNER[10], a huge gap of F1-score, when detecting locations and persons, is observed.

Table 5. Evaluation results (F1) on the TildeNER dataset (7 classes)

NE type	Multi-base	lv-base	TildeNER
DATE	70.74	79.07	
LOCATION	90.09	90.03	56.46
MONEY	81.54	85.5	
ORGANIZATION	70.81	70.98	65.71
PERSON	86.88	90.96	61.63
PRODUCT	67.25	56.34	
TIME	71.7	79.31	
F1	**78.83**	**79.72**	

BERT-based NER systems identify products poorly: they detect multiple products, which are not "products". Examples include button combinations ("Command+i", "Ctrl"), computer user interface parts ("Dock", "Applications", "Start"). The detection of organizations also suffers, because organization names are often complex multiword expressions, and often some of the tokens are marked as organizations wrongly. Examples which are marked wrongly as organizations include "President of Latvia" (this one counts an error for location as well), "International Bonds", "International Coordination Committees", and others.

6. Conclusion and Next Steps

In this paper, we examined the impact of large pre-trained BERT language models on named entity recognition in the case of a morphologically rich less-resourced language, specifically, Latvian. We demonstrated that large pre-trained BERT language models have a significant impact on the quality of NERs: the Latvian BERT model, pre-trained on large Latvian corpora, achieves better results (81.91 F1-measure on average vs. 78.37 on multi-BERT for the AILab dataset with nine NE types, and 79.72 vs. 78.83 on the TildeNER dataset with seven types) than multilingual BERT and outperforms previously developed NER systems for Latvian that were created using different architectures.

Acknowledgements

This research has been supported by the European Regional Development Fund within the joint project of SIA TILDE and University of Latvia "Multilingual Artificial Intelligence Based Human Computer Interaction" No. 1.1.1.1/18/A/148.

[10] Results reported for TildeNER [12] are obtained on different dataset and presented only for location, person, organization in comparative evaluation with Stanford NER classifier.

References

[1] Peters M, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, et al. Deep Contextualized Word Representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). Stroudsburg, PA, USA: Association for Computational Linguistics; 2018. p. 2227–37. Available from: http://aclweb.org/anthology/N18-1202.

[2] Devlin J., Chang M.-W., Lee K. and Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT 2019; 2019. p. 4171–4186.

[3] Wang Y., Sun Y., Ma Z., Gao L., Xu Y. and Sun T. Application of Pre-training Models in Named Entity Recognition; 2020. Available from: http://arxiv.org/abs/2002.08902.

[4] Pires T, Schlinger E, Garrette D. How Multilingual is Multilingual BERT? In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics [Internet]. Stroudsburg, PA, USA: Association for Computational Linguistics; 2019. p. 4996–5001.

[5] Martin L, Muller B, Javier P, Su O, Romary L, Villemonte E, et al. CamemBERT : a Tasty French Language Model; 2019. Available from: https://arxiv.org/abs/1911.03894

[6] Virtanen A, Kanerva J, Ilo R, Luoma J, Luotolahti J, Salakoski T, et al. Multilingual is not enough: BERT for Finnish. 2019 Dec 15; Available from: http://arxiv.org/abs/1912.07076.

[7] Souza F, Nogueira R, Lotufo R. Portuguese Named Entity Recognition using BERT-CRF. 2019; Available from: http://arxiv.org/abs/1909.10649

[8] Marrero M, Urbano J, Sánchez-cuadrado S, Morato J, Gómez-berbís JM. Named Entity Recognition : Fallacies , Challenges and Opportunities. Computer Standards & Interfaces 5, 2013, p. 482-489

[9] Yadav, V., Bethard, S.; A Survey on Recent Advances in Named Entity Recognition from Deep Learning models. Proceedings of the 27th International Conference on Computational Linguistics, August 20-26, 2018; Santa Fe, New Mexico, USA, pages 2145–2158.

[10] Li J, Sun A, Han J, Li C. A Survey on Deep Learning for Named Entity Recognition. IEEE Transactions on Knowledge and Data Engineering. 2020;1–1.

[11] Taher E, Hoseini SA, Shamsfard M. Beheshti-NER: Persian Named Entity Recognition Using BERT. 2020 Mar 19; Available from: http://arxiv.org/abs/2003.08875

[12] Pinnis M. Latvian and Lithuanian Named Entity Recognition with TildeNER. Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12); 2012. p. 1258-1265.

[13] Znotins A, Cirule E. NLP-PIPE: Latvian NLP tool pipeline. Frontiers in Artificial Intelligence and Applications. 2018;307:183–9.

[14] Chinchor N. MUC-7 Named Entity Task Definition; 1998. https://www.aclweb.org/anthology/M98-1028.

[15] Gruzitis N., Pretkalnina L., Saulite B., Rituma L., Nespore-Berzkalne G., Znotins A. and Paikens P. Creation of a balanced state-of-the-art multilayer corpus for NLU. LREC 2018 - 11th International Conference on Language Resources and Evaluation. 2019; p. 4506–4513.

[16] Arkhipov M, Trofimova M, Kuratov Y, Sorokin A. Tuning Multilingual Transformers for Language-Specific Named Entity Recognition. In: Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing. Stroudsburg, PA, USA: Association for Computational Linguistics; 2019. p. 89–93.

[17] Wu Y, Schuster M, Chen Z, Le Q V., Norouzi M, Macherey W, et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. 2016 Sep 26;1–23. Available from: http://arxiv.org/abs/1609.08144

[18] Kudo T, Richardson J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. EMNLP 2018 - Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Proceedings. 2018;66–71.

This page intentionally left blank

Machine Translation and Natural Language Understanding

This page intentionally left blank

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200605

Data Augmentation for Pipeline-Based Speech Translation

Diego ALVES [a], Askars SALIMBAJEVS [b,c]
and Mārcis PINNIS [b,c,1]

[a] *Faculty of Humanities and Social Sciences, University of Zagreb, Ul. Ivana Lučića 3, 10000, Zagreb, Croatia*
[b] *Tilde, Vienības gatve 75A, Riga, Latvia, LV-1004*
[c] *University of Latvia, Raiņa bulv. 19-125, Riga, Latvia, LV-1586*

Abstract. Pipeline-based speech translation methods may suffer from errors found in speech recognition system output. Therefore, it is crucial that machine translation systems are trained to be robust against such noise. In this paper, we propose two methods for parallel data augmentation for pipeline-based speech translation system development. The first method utilises a speech processing workflow to introduce errors and the second method generates commonly found suffix errors using a rule-based method. We show that the methods in combination allow significantly improving speech translation quality by 1.87 BLEU points over a baseline system.

Keywords. Neural machine translation, speech translation, robustness

1. Introduction

Speech translation systems are either end-to-end [1,2,3] or pipeline-based where automatic speech recognition (ASR) and machine translation (MT) systems work sequentially [4,5,6]. End-to-end speech translation systems require parallel data that has a speech audio signal on the source side and translated transcriptions on the target side. Such parallel data may be hard to obtain even for languages where the necessary components for the pipeline-based systems are readily available. Thus, the pipeline approach is often more realistic and practical. However, since ASR systems can introduce errors that a standard MT system has not seen during training and thus cannot handle, the translation quality can suffer, even to an extent where the translations are incomprehensible [7]. Therefore, in this paper, we investigate how to train neural MT (NMT) systems that are suitable for work in tandem with ASR for pipeline-based speech translation. More specifically, we propose data augmentation methods to produce data that features mistakes common to speech recognition system output, thereby improving the NMT system robustness to noise propagated within the pipeline.

[1] Corresponding Author: Mārcis Pinnis; E-mail: marcis.pinnis@tilde.lv.

2. Related Work

The adverse effects of error propagation in pipeline-based speech translation systems have been studied [7] and addressed before [4,8,9,10]. Some of the work has proposed to mitigate ASR errors by translating either N-best lists [4] or lattices [9,10]. In this work, we focus on methods that do not require drastic changes in already existing MT workflows. Therefore, closest to our work is the work on ASR noise modelling [8,9]. Different from previous work, our noise generation method tries to introduce noise that is either actually generated by the ASR system or highly probable based on ASR error analysis. Our method is also suited for morphologically rich languages, for which a large proportion of errors are inflection mistakes.

3. Synthetic Data Generation

We investigate two methods for the generation of noise typical to ASR system output. The first method (see Section 3.1) generates noise using a speech processing workflow. Since the workflow may be limited in the lexical variety of errors it can introduce (due to a limited number of speakers), we performed an analysis of the types of errors introduced by the first method and devised also a rule-based method (see Section 3.2).

3.1. Synthetic Data Generation Using Speech Synthesis and Recognition

We propose to generate data with synthetic ASR noise by using a pipeline of text-to-speech (TTS) and ASR systems. Unlike the previous work [9], our approach generates not only substitution errors, but insertion and deletion errors as well. The main limitation of generating noise using a pipeline of TTS and ASR systems is the availability of such systems and the selection of different TTS voices. Other than that, no extra resources besides those used for MT system training are required.

We generate synthetic data as follows (see Figure 1):

- First, we synthesise source language sentences from the MT training data using TTS (the gray boxes in Figure 1).
- Then, we use ASR to acquire transcriptions of the synthesised sentences (the orange boxes in Figure 1).
- Finally, we use the ASR transcriptions together with the original target sentences as the synthetic MT training data.

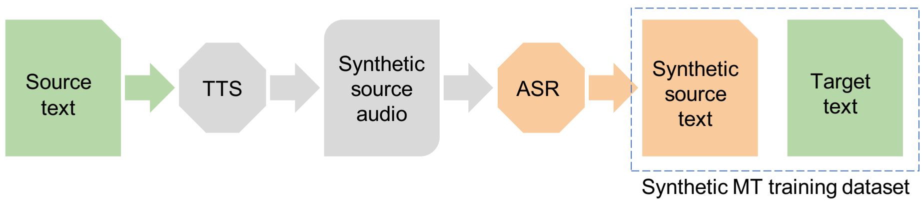


Figure 1. The synthetic data generation pipeline

3.1.1. Synthetic Data Filtering

Although the speech processing workflow introduces real noise, it is also evident that it introduces errors due to the limitations of the workflow itself. For instance, the speech synthesiser is unable to pronounce foreign named entities and complex identifiers correctly. This results in misrecognition by the ASR system (i.e., in such cases, it either deletes words or recognises the mispronounced names as some other common phrases, which are not even necessarily phonetically similar). The synthesiser also drops most Unicode characters that it cannot pronounce, which also introduces noise in the parallel corpus. Examples of errors introduced by the workflow itself are given in Table 1.

Table 1. Examples of noise introduced by limitations of speech processing tools

Source / Target in the parallel corpus	Synthetic segment (Latvian) / Translation (English)
Pierre Schapira, Attīstības komiteja	**ieviests papīra** attīstības komiteja
Pierre Schapira, Committee on Development	**introduced paper** committee on development
Mieczysław Edmund Janowski	edmunda jānoski
Mieczysław Edmund Janowski	edmund jnoski
Skatīt arī **MEMO /** 14 **/** 597.	skatīt arī **melo** 14 597
See also **MEMO /** 14 **/** 597.	see also **lie** 14 597

To address these issues, we filter the generated synthetic data by discarding sentences:

- that contain website addresses or Roman or Arabic digits,
- that contain one character,
- for which the Levenshtein [11] distance-based similarity between the original and the synthetic sentence is lower than 0.9.

We applied the Levenshtein distance-based similarity threshold to identify segments that have been too much corrupted by the synthetic data generation workflow. For instance, this allows addressing issues introduced by mis-recognising foreign named entities and complex identifiers.

3.2. Rule-Based Synthetic Noise Generation

After filtering, we performed error analysis to investigate what types of errors common to ASR systems were present in the filtered data. The results showed that 35.6 % of all errors were suffix-related. This was to be expected as Latvian is a morphologically rich language with fusional morphology and incorrect inflections are common mistakes for ASR systems. Other errors were deletions (32.9 %), insertions (25.2 %), and the remaining 6.3 % were related to other types of lexical misrecognition.

Next, we analysed what types of suffix (or inflection) errors are present in the data. For this, we used the Tilde's Latvian part-of-speech (POS) tagger[2] and extracted a list of most common inflection changes. The analysis revealed that the 26 most common inflection changes amount to 90 % of all suffix errors. We found that 26 most common inflection changes amounted to 90 % of all suffix errors. Therefore, we devised a method that in a random fashion iterates through the original dataset and randomly in each sentence

[2]The source code can be found online at: https://github.com/pdonald/latvian

generates one of the error types for a random number of words for which that particular error type can be introduced. As an additional step, we validated the generated errors using a vocabulary to make sure that the generation produced real Latvian words. After the generation, we selected only those sentences that had at least one error.

4. Experiments

We use the Latvian-English parallel data from the WMT 2017[3] shared task on news translation to train MT systems. The raw parallel corpus consists of 4,486,467 sentences. The corpus was pre-processed using a standard parallel data pre-processing workflow from the Moses statistical MT toolkit [12][4] and split into sub-word units. First, punctuation was normalised[5]. Then, the corpus was cleaned[6] by removing too long segments and segments where the source and target length ratio exceeds three times. After cleaning, 4,407,375 sentences remained in the training data. The data were further processed using the Moses tokeniser[7] and truecaser[8]. Finally, words were split into sub-word units using byte-pair encoding[9] [13,14] with 24.5K merge operations.

In order to generate the synthetic data, we use a selection of three Latvian TTS voices. For each sentence, we choose one of the voices at random. Characters that are not pronounced by TTS, are filtered from the source text (e.g., parentheses, quotation marks, various other punctuation marks).

For ASR, we use an ASR system, which is based on a hybrid Hidden Markov Model and a Time-delay Neural Network acoustic model and a sub-word level language model. The language model is implemented on a sub-word level (using byte-pair encoding, BPE) and consists of 4-grams, 6-grams, and a recurrent neural network (RNNLM). However, in order to inject more errors into the resulting synthetic training data, only the 4-gram language model is used to produce synthetic data. As the raw speech recognition output contains numbers written with words not digits, the MT model would have to learn how to translate words into digits. Therefore, to simplify the training of the MT system, we apply a number normalization tool for Latvian that re-writes numbers as digits. If number normalization tools are not available for a given language, raw transcripts from the ASR system could also be used as the source text for MT training.

The speech processing workflow produced 4,407,364 synthetic sentences. After filtering, 1,921,043 sentences remained in the synthetic dataset. The rule-based synthetic noise generation workflow produced 753,693 sentences and 961,227 sentences when using and not using vocabulary validation. The vocabulary was built using the original training data.

For validation, we use the *NewsDev2017* dataset from WMT 2017. The dataset was also processed using the speech processing workflow. For validation during training, we use a combination of the clean and noisy validation sets. For evaluation, we use a dataset, which represents real-world data from an ASR application. The evaluation set

[3]http://statmt.org/wmt17/
[4]https://github.com/moses-smt/mosesdecoder
[5]normalize-punctuation.perl
[6]clean-corpus-n.perl
[7]tokenizer.perl
[8]train-truecaser.perl and truecase.perl
[9]https://github.com/rsennrich/subword-nmt

is based on a subset of data collected by Tilde's real-time Latvian ASR service. This subset contains 8,820 utterances, which amount to 37,782 words and 39 hours of audio (including silence). Utterances originate from various domains: queries, short messages, addresses, interaction with a voice-enabled educational app, etc. It contains also a lot of "noise": laughter, English and Russian speech, untranslatable or ambiguous utterances, etc. Therefore, several rounds of semi-automatic filtering were performed to select meaningful utterances from this dataset. This resulted in a final evaluation dataset of 1,159 Latvian utterances that were manually translated to English.

All NMT systems were trained using Transformer models from the Marian NMT toolkit [15]. All models were trained till the the validation loss did not improve for 10 consecutive validation iterations.

5. Results

The results in Table 2 show that for the ASR output, supplementing the original training data with synthetic noise allows increasing the MT quality by up to 1.66 BLEU points. A similar tendency is evident when translating human-created transcripts that also may contain orthographic speech noise (e.g., truncated words, incorrect syntax, wrong punctuation, etc.). Only when translating clean publishable transcripts, the systems that are trained on noisy data show lower results than the baseline system. Nevertheless, the synthetic noise generation strategies have been successful in handling ASR output better and achieving higher translation quality than the baseline system.

The results also show that the best results were achieved when combining the filtered synthetic data and the data that is generated using rules with vocabulary validation. The combined data allow increasing the translation quality by 1.87 BLEU points over the baseline system.

Table 2. Evaluation results (bold – highest score; † – improvement over the baseline is significant with $p < 0.01$)

Training data	ASR output		Human transcripts		Transcripts + punct.	
	BLEU	ChrF2	BLEU	ChrF2	BLEU	ChrF2
a. Original parallel data (baseline)	12.73	0.4395	14.67	0.4622	**20.90**	**0.5123**
b. Noisy synthetic data	12.61	0.4160	14.08	0.4306	13.94	0.4251
c. a + b	†14.33	0.4374	†16.08	0.4577	20.45	0.5068
d. Filtered synthetic data + a	†14.39	**0.4602**	†16.79	0.4854	19.37	0.4995
e. Rule-based data (no voc.) + a	12.08	0.4243	13.12	0.4377	18.58	0.4830
f. Rule-based data (with voc.) + a	11.47	0.4231	12.75	0.4367	18.94	0.4847
g. Rule-based data (no voc.) + d	13.72	0.4484	†16.00	0.4815	18.69	0.4973
h. Rule-based data (with voc.) + d	†**14.60**	0.4547	†**17.29**	**0.4907**	19.46	0.5028

6. Conclusion

We proposed data augmentation strategies for speech translation that introduce noise typical to ASR output in parallel data for NMT systems. We showed that the methods allow generating synthetic parallel data that allows improving speech translation quality.

We believe the methods will be beneficial when developing NMT systems for speech translation purposes.

Future work on data augmentation strategies may be directed in two directions. Generation of a wider variety of ASR errors using rule-based methods (in these experiments, we covered only 26 rules) as well as investigating how much can be achieved by stripping most punctuation and symbols that are not supported by ASR systems from the parallel data.

Relevant code for re-producing the the synthetic data filtering and rule-based error generation results is published on GitHub[10].

Acknowledgements

This research has been supported by the ICT Competence Centre (www.itkc.lv) within the project "2.2. Adaptive Multimodal Neural Machine Translation" of EU Structural funds, ID n° 1.2.1.1/18/A/003. The work presented in this paper has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement no. 812997 and under the name CLEOPATRA (Cross-lingual Event-centric Open Analytics Research Academy). The presented research has been funded by the European Union's Horizon 2020 research and innovation programme project COMPRISE (http://www.compriseh2020.eu/) under grant agreement No. 3081705. The work has been carried out during Diego Alves' internship in Tilde.

References

[1] Duong L, Anastasopoulos A, Chiang D, Bird S, Cohn T. An attentional model for speech translation without transcription. In: Proceedings of NAACL 2016; 2016. p. 949–959.

[2] Bansal S, Kamper H, Lopez A, Goldwater S. Towards speech-to-text translation without speech recognition. In: Proceedings of EACL 2017; 2017. p. 474–479.

[3] Bansal S, Kamper H, Livescu K, Lopez A, Goldwater S. Low-Resource Speech-to-Text Translation. Proceedings of Interspeech 2018. 2018:1298–1302.

[4] Suhm B, Geutner P, Kemp T, Lavie A, Mayfield L, Mcnair AE, et al.. Janus: Towards Multilingual Spoken Language Translation; 1995.

[5] Waibel A, Fugen C. Spoken language translation. IEEE Signal Processing Magazine. 2008;25(3):70–79.

[6] Liu D, Liu J, Guo W, Xiong S, Ma Z, Song R, et al. The USTC-NEL speech translation system at IWSLT 2018. arXiv preprint arXiv:181202455. 2018.

[7] Ruiz N, Di Gangi MA, Bertoldi N, Federico M. Assessing the Tolerance of Neural Machine Translation Systems Against Speech Recognition Errors. 2017.

[8] Sperber M, Niehues J, Waibel A. Toward robust neural machine translation for noisy input sequences. In: Proceedings of IWSLT 2017; 2017. p. 1–7.

[9] Simonnet E, Ghannay S, Camelin N, Estève Y. Simulating ASR errors for training SLU systems. In: Proceedings of LREC 2018; 2018. p. 3157–3162.

[10] Sperber M, Neubig G, Pham NQ, Waibel A. Self-Attentional Models for Lattice Inputs. In: Proceedings of ACL 2019; 2019. p. 1185–1197.

[11] Levenshtein VI. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Soviet Physics Doklady. 1966;10(8):707–710.

[12] Koehn P, Hoang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, et al. Moses: Open Source Toolkit for Statistical Machine Translation. In: Proceedings of ACL 2007. ACL '07. Stroudsburg, PA, USA: Association for Computational Linguistics; 2007. p. 177–180.

[10] https://github.com/dfvalio/Speech_Translation

[13] Gage P. A new algorithm for data compression. C Users Journal. 1994;12(2):23–38.
[14] Sennrich R, Haddow B, Birch A. Neural Machine Translation of Rare Words with Subword Units. In: Proceedings of ACL 2016. Berlin, Germany: Association for Computational Linguistics; 2016. p. 1715–1725.
[15] Junczys-Dowmunt M, Grundkiewicz R, Dwojak T, Hoang H, Heafield K, Neckermann T, et al. Marian: Fast Neural Machine Translation in C++. In: arXiv preprint arXiv:1804.00344; 2018. p. 116–121.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200606

Robust Neural Machine Translation: Modeling Orthographic and Interpunctual Variation

Toms BERGMANIS [a,b,1], Artūrs STAFANOVIČS [a], Mārcis PINNIS [a,b]
[a] *Tilde, Riga, Latvia*
[b] *Faculty of Computing, University of Latvia, Latvia*

Abstract. Neural machine translation systems typically are trained on curated corpora and break when faced with non-standard orthography or punctuation. Resilience to spelling mistakes and typos, however, is crucial as machine translation systems are used to translate texts of informal origins, such as chat conversations, social media posts and web pages. We propose a simple generative noise model to generate adversarial examples of ten different types. We use these to augment machine translation systems' training data and show that, when tested on noisy data, systems trained using adversarial examples perform almost as well as when translating clean data, while baseline systems' performance drops by 2-3 BLEU points. To measure the robustness and noise invariance of machine translation systems' outputs, we use the average translation edit rate between the translation of the original sentence and its noised variants. Using this measure, we show that systems trained on adversarial examples on average yield 50 % consistency improvements when compared to baselines trained on clean data.

Keywords. Neural machine translation, robustness, noisy data

1. Introduction

Humans exhibit resilience to orthographic variation in written text [1,2]. As a result, spelling mistakes and typos are often left unnoticed. This flexibility of ours, however, is shown to be detrimental for neural machine translation (NMT) systems, which typically are trained on curated corpora and tend to break when faced with noisy data [3,4]. Achieving NMT robustness to human blunder, however, is important when translating texts of less formal origins, such as chat conversations, social media posts and web pages with comment sections.

In this work, we propose to augment NMT system's training data with data where source sentences are corrupted with adversarial examples of different types. There have been various studies on the impact of different types and sources of noise on NMT [5,6,7]. In this work, we focus on the noise caused by orthographic variation of words, such as unintentional misspellings and deliberate spelling alternations as well as noise due to misplaced and omitted punctuation. Thus, the closest to this study is the work on

[1]Corresponding Author: Toms Bergmanis; E-mail: toms.bergmanis@tilde.lv.

black-box adversarial training of NMT systems [3,8,9], where models are trained on adversarial examples that are generated without accessing the model's parameters. Unlike the previous work, which focuses only on adversarial examples that model unintentional changes of spelling, we also model deliberate orthographic alternation, such as omission and substitution of diacritical signs. As we show in our experiments, such orthographic variation has a more substantial negative impact on MT outputs than other types of noise and thus is more important to be accounted for. Further, to overcome the lack of curated evaluation datasets as required by the previous work [4,9], we propose an automatic evaluation method that measures the noise invariance of MT outputs without relying on a reference translation. By measuring noise invariance of MT outputs, the method also allows us to assess whether MT system translation consistency improves when facing small variations in the source text.

Table 1. Noise applied to the example sentence: "*Balta jūra, zaļa zeme.*" Were possible, noise is marked in bold, otherwise it is indicated with '_'

#	Type	Examples
1	introduce extra letters	Bal**z**ta jūra, zaļa zeme.
2	delete letters	_alta jūra, zaļa zeme.
3	permute letters	Ba**tl**a jūra, zaļa zeme.
4	confuse letters	Balta jūra, **x**aļa zeme.
5	add diacritic	Balta jūra, zaļa z**ē**me.
6	sample substitute	Balta jūra, zaļa **zemi**.
7	remove punctuation	Balta jūra_ zaļa zeme_
8	add comma	Balta**,** jūra, zaļa zeme.
9	latinize	Balta j**u**ra, za**l**a zeme.
10	phonetic latinize	Balta j**uu**ra, za**lj**a zeme.

2. Methods

We propose a simple generative noise model to generate adversarial examples of ten different types. These include incidental insertion, deletion, permutation and keyboard-based confusion of letters as well as the addition of a diacritic to letters which support them (Table 1, examples 1-5). We also explicitly model the misspellings that result in another valid word (Table 1, example 6). For interpunctual variation, we consider sentences with missing punctuation and incorrectly placed commas (Table 1, examples 7-8). For deliberate orthographic changes, we support sentence-level omission and phonetic latinization of diacritical signs (Table 1, examples 9-10).

Measure of Robustness. To measure NMT robustness and noise invariance of NMT outputs, we calculate the average translation edit rate (TER) [10] between the translation of the original orthographically correct sentence and the translations of its ten noised variants for each noise type. We refer to it as tenfold noisy translation TER, or **10NT-TER**. This measure gives a score of 0 if all ten translations of a sentence with added noise match the translation of the original sentence and a score of 100 (or more) if all of them had no word in common with the translation of the original sentence.

Table 2. The original training data sizes and data sizes with adversarial examples included

		Train	
		Original	+ adversarial noise
Small data	English-Latvian	4.5M	9M
Large data	English-Estonian	34.9M	69.8M
	English-Latvian	45.2M	90.4M
	English-Lithuanian	22.1M	44.2M

3. Experimental Setting

Languages and Data. We conduct experiments on Estonian-English, Latvian-English and Lithuania-English language pairs. We use the Latvian-English constrained data from the WMT 2017[2] news translation shared task to train **small data systems** that we use for development and analysis of our methods. To verify that our findings also hold not only for small data settings, but also for production-grade systems that are trained on much larger data, we use large datasets from the Tilde Data Library[3] to train **large data systems**. For the validation during training and testing, we use development and test sets from the WMT news translation shared tasks. For English-Estonian, we use the data from WMT 2018, for English-Latvian – WMT 2017, and for English-Lithuanian – WMT 2019[4].

We use a simplified and production-grade data pre-processing pipelines. The simplified data pre-processing consists of the standard Moses [11] scripts for tokenization, cleaning, normalization, and truecasing, while the production grade pipeline consists of Tilde MT platform's [12] implementation of the same processes.

NMT Models. We mostly use the default configuration[5] of the Marian [13] toolkit's implementation of the Transformer model [14]. We select batch sizes dynamically so that they fit in a workspace of 9000MB. Additionally, we use delayed gradient updates [15] by setting optimizer delay to 4. We stop model training after ten consecutive evaluations with no improvement in translation quality on the development set [16].

4. Experiments

Initial Experiments. To test the effect of individual noise models on MT systems' performance, we train separate Latvian-English small data systems on original data augmented in a 1-to-1 proportion with each type of adversarial examples. All in all, we obtain ten systems trained using adversarial examples and the baseline. We test each system on the original development set and development sets that have adversarial examples of each type of noise. Table 3 summarises the results. First, we note that including adversarial examples improves the overall translation quality and especially quality on development sets containing the adversarial examples that the systems have seen during training.

[2]http://www.statmt.org/wmt17
[3]https://www.tilde.com/products-and-services/data-library
[4]http://www.statmt.org/wmt17|18|19
[5]https://github.com/marian-nmt/marian-examples/tree/master/transformer

Table 3. Latvian-English development set results in BLEU [17] points for small data systems. Rows: systems trained on original data that are 1:1 up-sampled with each type of adversarial examples. Columns: development sets with each type of adversarial examples

	original data	latinize	phonetic latinize	add diacritic	delete letters	permute letters	introduce extra letters	confuse letters	sample substitute	remove punctuation	add comma	average
baseline	21.4	9.3	7.6	20.1	19.9	19.5	20.2	19.9	20.2	16.9	21.1	17.8
latinize	21.9	21.2	15.6	20.6	20.5	20.0	20.9	20.1	20.4	17.1	21.4	20.0
phonetic latinize	21.4	15.3	21.2	20.4	20.3	19.7	20.4	19.8	20.3	16.9	21.2	19.7
add diacritic	21.7	11.2	8.8	21.6	20.7	20.5	20.7	20.2	20.5	17.3	21.4	18.6
delete letters	21.8	12.0	9.5	20.8	21.1	20.7	20.9	20.2	20.5	17.0	21.2	18.7
permute letters	22.0	12.1	9.8	21.1	21.3	21.7	21.6	20.7	20.7	17.4	21.7	19.1
introduce extra letters	21.6	11.7	10.1	20.8	20.7	20.4	21.2	20.7	20.4	17.1	21.4	18.7
confuse letters	21.7	12.8	11.0	21.1	21.0	20.9	21.3	21.2	20.8	17.2	21.3	19.1
sample substitute	21.7	10.6	8.3	20.6	20.4	20.1	20.6	20.3	21.3	17.1	21.4	18.4
remove punctuation	21.6	9.5	7.6	20.0	20.2	19.3	20.4	19.9	20.5	20.4	21.5	18.3
add comma	21.3	9.3	7.5	20.2	20.0	19.6	20.5	20.0	20.2	17.3	21.5	17.9

Second, we observe that not all diagonal elements of Table 3 contain the highest BLEU score for their respective column, suggesting existing redundancies between the noise models. Examples are MT systems trained using adversarial examples from noise models that *delete letters* and *introduce extra letters*, which both when tested on their respective adversarial example development sets come second to the MT system that was trained using adversarial examples from the noise model that *permutes letters* (21.1 vs 21.3 BLEU points and 20.9 vs 21.6 BLEU points respectively). Similarly, the MT system trained using the noise model that *adds a comma* (21.5 BLEU), shows no benefit over the system that was trained using examples from the model that *removes punctuation* (21.5 BLEU). Based on these results, we decided not to include the redundant models (*delete letters*, *introduce extra letters* and *add comma*) in further experiments.

We, however, also recognize that the performance gains caused by the remaining noise models are numerically small (+0.5 BLEU) when compared against the next best performing MT system. For this reason, we use bootstrap re-sampling [18] to test if the performance gains of MT systems trained on adversarial examples generated by the noise models that *add a diacritic*, *confuse letters*, and perform *sample substitution* are statistically significant if compared against a system that is trained on adversarial examples generated by the noise model that *permutes letters*. Tests confirm that all differences are indeed significant at $p < 0.05$. Based on these tests, we include these models in our final experiments.

Large Data Systems. To test the effect of the seven productive noise models on MT system translation quality, we train Estonian-English, Latvian-English and Lithuanian-English large data MT systems. For systems trained using adversarial examples, we augment the original data with another copy of the data in which each type of noise is applied

Table 4. Test set results in BLEU points for large data MT systems

		original data	latinize	phonetic latinize	add diacritic	permute letters	confuse letters	sample substitute	remove punctuation	**average**
ET-EN	**baseline**	22.5	17.0	-	20.8	20.4	20.3	20.7	18.1	20.0
	+ adversarial noise	22.6	**22.5**	-	**22.4**	**22.2**	**22.0**	**21.8**	**21.7**	**22.2**
LV-EN	**baseline**	19.0	10.8	8.2	18.0	17.6	17.7	18.2	18.2	16.0
	+ adversarial noise	19.4	**18.8**	**18.9**	**19.2**	**19.0**	**18.9**	**19.0**	**18.6**	**19.0**
LT-EN	**baseline**	20.0	14.6	-	18.6	18.3	18.3	18.7	17.6	18.0
	+ adversarial noise	20.3	**19.3**	-	**20.0**	**19.9**	**19.7**	**19.7**	**20.5**	**19.9**

Table 5. Examples of noise in Latvian language input data causing widely different English language translations

Orig.	Twitter lietotāji nespēja noticēt, dzirdot Bairona Makdonalda neiejūtīgos komentārus.
Ref.	Twitter users did not hold back when they heard how insensitive Byron Macdonlad was being.
Src.	Twitter **leitotāji** nespēja noticēt, dzirdot Bairona Makdonalda neiejūtīgos komentārus.
Hyp.	Twitter's lieutenants couldn't believe it by hearing Byron McDonald's insensitive comments.
Src.	Twitter lietotāji nespēja noticēt, **dzidrot** Bairona Makdonalda neiejūtīgos komentārus.
Hyp.	Twitter users could not believe by clarifying Byron McDonald's insensitive comments.
Src.	**Twtiter** lietotāji nespēja noticēt, dzirdot Bairona Makdonalda neiejūtīgos komentārus.
Hyp.	Twtiter users couldn't believe it when they heard Byron McDonald's insensitive comments.
Src.	Twitter **liettoāji** nespēja noticēt, dzirdot Bairona Makdonalda neiejūtīgos komentārus.
Hyp.	The Twitter countryside couldn't believe it by hearing Byron McDonald's insensitive comments.

at an equal proportion. The evaluation results of these systems on the test data sets are provided in Table 4. First, we observe that the performance of the baseline MT systems and systems trained using adversarial examples on the original test data sets is comparable, suggesting that using adversarial examples does not harm the translation of clean data. Next, we observe, that when tested on noisy data systems that are trained using adversarial examples perform only slightly worse (about -0.4 BLEU on average) than when translating clean data, while baseline systems show an average performance drop of about 2 BLEU points.

Robustness and Noise Invariance. Besides measuring the changes in translation quality caused by noisy input data, we also would like to measure the robustness and noise invariance of the MT systems. We motivate this by the observation that often small perturbations in input data lead to widely different MT outputs (see Table 5). Desideratum, however, is that MT system outputs are noise invariant to an extent at least that unintentional changes in input data do not affect the meaning of the translation output. Results (see Table 6) of our experiments show that using adversarial examples in training improves the robustness and noise invariance of the MT systems measured in 10NT-TER (see Section 2) on average by 0.1 10NT-TER points or in relative terms an average consistency improvement of about 50 %.

Table 6. Robustness and noise invariance of large data MT systems measured in 10NT-TER (see Section 2)

		latinize	phonetic latinize	add diacritic	permute letters	confuse letters	sample substitute	remove punctuation	**average**
ET-EN	**baseline**	0.27	-	0.13	0.14	0.15	0.13	0.29	0.19
	+ adversarial noise	**0.06**	-	**0.05**	**0.06**	**0.07**	**0.08**	**0.16**	**0.08**
LV-EN	**baseline**	0.51	0.63	0.16	0.17	0.17	0.12	0.28	0.29
	+ adversarial noise	**0.16**	**0.14**	**0.07**	**0.09**	**0.10**	**0.09**	**0.15**	**0.11**
LT-EN	**baseline**	0.39	-	0.19	0.21	0.20	0.15	0.39	0.25
	+ adversarial noise	**0.12**	-	**0.08**	**0.09**	**0.10**	**0.09**	**0.29**	**0.13**

5. Conclusions

We have proposed a simple generative noise model for the generation of adversarial examples for training data augmentation of NMT systems. Our results demonstrate that NMT systems that are trained using adversarial examples are more resilient to noisy input data. We show that while for the baseline NMT systems, noisy inputs cause a substantial drop in the translation quality (a drop of 2-3 BLEU points), for the systems that are trained using adversarial examples translation quality changes comparatively little (an average of -0.4 BLEU). In terms of translation robustness, systems trained on adversarial examples on average yield 50% consistency improvement when compared to baselines trained on clean data. Methods proposed here will be useful for achieving NMT robustness to orthographic and interpunctual variation in input data. This will be especially beneficial in use cases where NMT systems are used to translate texts of informal origins, such as chat conversations, social media posts and web pages with comment sections.

6. Acknowledgments

This research has been supported by the European Regional Development Fund within the joint project of SIA TILDE and University of Latvia "Multilingual Artificial Intelligence Based Human Computer Interaction" No. 1.1.1.1/18/A/148.

References

[1] Rawlinson GE. The significance of letter position in word recognition. University of Nottingham; 1976.
[2] McCusker LX, Gough PB, Bias RG. Word recognition inside out and outside in. Journal of Experimental Psychology: Human Perception and Performance. 1981;7(3):538.
[3] Belinkov Y, Bisk Y. Synthetic and Natural Noise Both Break Neural Machine Translation. In: International Conference on Learning Representations; 2018. .
[4] Michel P, Neubig G. MTNT: A Testbed for Machine Translation of Noisy Text. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing; 2018. p. 543–553.
[5] Carpuat M, Vyas Y, Niu X. Detecting Cross-Lingual Semantic Divergence for Neural Machine Translation. In: Proceedings of the First Workshop on Neural Machine Translation. Vancouver: Association for Computational Linguistics; 2017. p. 69–79.
[6] Khayrallah H, Koehn P. On the Impact of Various Types of Noise on Neural Machine Translation. In: Proceedings of the 2nd Workshop on Neural Machine Translation and Generation; 2018. p. 74–83.
[7] Zhou S, Zeng X, Zhou Y, Anastasopoulos A, Neubig G. Improving Robustness of Neural Machine Translation with Multi-task Learning. In: Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1); 2019. p. 565–571.
[8] Sperber M, Niehues J, Waibel A. Toward robust neural machine translation for noisy input sequences. In: International Workshop on Spoken Language Translation (IWSLT); 2017. .
[9] Vaibhav V, Singh S, Stewart C, Neubig G. Improving robustness of machine translation with synthetic noise. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers); 2019. p. 1916–1920.
[10] Snover M, Dorr B, Schwartz R, Micciulla L, Makhoul J. A study of translation edit rate with targeted human annotation. In: Proceedings of association for machine translation in the Americas. vol. 200; 2006. .
[11] Koehn P, Hoang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, et al. Moses: Open source toolkit for statistical machine translation. In: Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions. Association for Computational Linguistics; 2007. p. 177–180.
[12] Pinnis M, Vasiļjevs A, Kalniņš R, Rozis R, Skadiņš R, Šics V. Tilde MT Platform for Developing Client Specific MT Solutions. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018); 2018. .
[13] Junczys-Dowmunt M, Grundkiewicz R, Dwojak T, Hoang H, Heafield K, Neckermann T, et al. Marian: Fast neural machine translation in C++. arXiv preprint arXiv:180400344. 2018.
[14] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Advances in neural information processing systems; 2017. p. 5998–6008.
[15] Bogoychev N, Heafield K, Aji AF, Junczys-Dowmunt M. Accelerating Asynchronous Stochastic Gradient Descent for Neural Machine Translation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: Association for Computational Linguistics; 2018. p. 2991–2996. Available from: https://www.aclweb.org/anthology/D18-1332.
[16] Prechelt L. Early stopping-but when? In: Neural Networks: Tricks of the trade. Springer; 1998. p. 55–69.
[17] Papineni K, Roukos S, Ward T, Zhu WJ. BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics; 2002. p. 311–318.
[18] Koehn P. Statistical significance tests for machine translation evaluation. In: Proceedings of the 2004 conference on empirical methods in natural language processing; 2004. p. 388–395.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200607

Interactive Learning of Dialog Scenarios from Examples

Daiga DEKSNE and Raivis SKADIŅŠ [1]
Tilde, Riga, Latvia
Faculty of Computing, University of Latvia, Latvia

Abstract. This paper reports on the development of a toolkit that enables collecting dialog corpus for end-to-end goal-oriented dialog system training. The toolkit includes the neural network model that interactively learns to predict the next virtual assistant (VA) action from the conversation history. We start with exploring methods for VA dialog scenario learning from examples after we perform several experiments with the English DSTC dialog sets in order to find the optimal strategy for neural model training. The chosen algorithm is used for training the next action prediction model for the Latvian dialogs in the public transport inquiries domain collected using the platform. The accuracy for the English and the Latvian dialog models is similar – 0.84 and 0.86. This shows that the chosen method for neural network model training is language independent.

Keywords. Virtual assistants, machine learning, dialog corpus

1. Introduction

A lot of manual human effort still needs to be invested to develop virtual assistants that can help specialists in customer service. Organizations serving their clients have accumulated conversation archives in both text and audio format, yet only a small portion of that data can be used for VA training [1]. Today, VAs typically work according to dialog scenarios that are executed depending on the user's intents and the data collected from the user. VAs analyze user input, determine user intent and entities, and perform dialog scenario steps. Currently, machine learning techniques are used to train the models for intent detection and entity recognition, but dialog scenarios are usually created manually.

In this research, we explore the methods that will allow VA dialog scenarios to be learned from examples. There are several studies for English in this area, such as [2]-[8]. We aim to create the methods that are as much language-independent as possible so that they can be used for the Baltic as well as other languages.

To gather the training data, we have created an environment for interactive dialog data collection. We employ the Wizard-of-Oz approach. At first, VA users communicate with a VA trainer (person pretending to be a VA), and we log these communications. VA trainer defines new VA actions and tries to reuse already defined actions as much as possible. Then we use the collected data to build a model that predicts the next VA action. The model then is used in the data collection process to predict VA actions in new dialogs,

[1] Corresponding Author: Raivis Skadiņš; Tilde, Vienibas gatve 75a, Riga, Latvia, LV1004; E-mail: raivis.skadins@tilde.lv.

where VA trainer either picks actions predicted by the model or creates new actions. We regularly retrain the model with the new data collected, thus model quality increases while we collect new data.

2. VA Action Prediction Model

There is no open-source dialog data available for Latvian. Instead, there are several valuable dialog corpora for English:

- Cambridge Restaurant Corpus [6] was designed to assist the user to find a restaurant in the vicinity of Cambridge. The corpus includes 676 dialogs.
- Stanford Driver's In-car Assistant Dialog Dataset [9] is a multi-domain corpus in three domains: calendar scheduling, weather information retrieval, and point-of-interest navigation. The corpus contains 3,031 dialogs.
- Ubuntu Dialog Corpus [10] contains 930,000 dialogs extracted from the Ubuntu chat logs in which support for various Ubuntu-related technical problems is provided.
- JD Customer Service Corpus [11] includes online retailing customer service dialogs between customers and customer service staff on the web site JD.com. There are 415,000 dialogs for training, 1,500 dialogs for validation and 5,005 dialogs for test.
- DSTC2 and DSTC3 data sets were assembled for the Dialog State Tracking Challenge[2], where participants developed state tracking algorithms using a labeled corpus of dialogs in the restaurant information domain. In total, there are 5,506 dialogs in both data sets.
- MultiWOZ Corpus [12] contains 3,406 single domain and 7,032 multi-domain task-oriented dialogs collected through the crowd work using the Wizard of Oz method.

To achieve the goals of our research, we started by building a VA action prediction model for English. We used English data to design and validate neural network models and to experiment with different training data encodings and model hyperparameters before we start collecting data for Latvian to avoid potential mistakes in the data collection process.

2.1. Training Data

For our experiments, we used DSTC2 and DSTC3 dialog sets. We transformed the data to fit the needs of our task – prediction of the next VA action. Each dialog has dictionary type records representing a single turn in a dialog. The dialog consists of several turns. Each turn has the following information:

- **actor**: ‘VA’ or ‘user’;
- **utterance**: VA or user utterance;
- **entities**: key/value pairs representing entities and their values;

[2] http://camdial.org/~mh521/dstc/

- **intents**: an array with one or several intents (only for the user's turns);
- **action**: an action of VA (only for VA turns).

There are two types of actions that the VA can perform. One that returns a textual response, another one that performs some function potentially involving outside sources – checking availability or inquiring about something, making calculations, and setting variables.

2.2. The Architecture of the Neural Network Model

We train the neural network model that predicts the next VA action when given the previous conversation history. We use the LSTM recurrent neural network architecture for this task (see **Figure 1**). We use a single layer of 100 LSTM cells. To avoid overfitting, we introduce a Dropout Layer with a dropout rate of 0.5 after the LSTM layer. We use softmax activation in the Dense Layer, categorical cross-entropy function for loss calculation, and Adam optimizer [13].

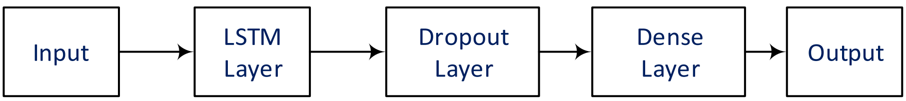


Figure 1. Layers of the neural network model

2.3. Training Settings

We performed nine experiments with different embeddings for utterance encoding and different ways how we encode the entities (see **Table 1**). The test accuracy is calculated using 10-fold cross-validation.

We evaluated two variants of representing the user input:

- One-hot vector of intents in experiments 4 and 9;
- Sentence embeddings in experiments 1-3 and 5-8.

In experiments 1, 5, and 8, we used pre-trained *fastText* embeddings [14] trained on *Wikipedia*, whereas in experiments 2, 6, and 7, we used a variation of *fastText* algorithm [15]. In experiment 3, we used uncased *BERT-Base* model [16], but as it required more computing resources and did not show better results compared to models with *fastText* embeddings, we did not use it in further experiments.

Table 1. Results of different neural network models

Nr	Vectorization	Dimensions	Entities	Test accuracy
1	wiki.en.bin	300	194	**0.8395**
2	news.wiki.en.bin	300	194	0.8303
3	BERT-Base	768	194	0.8209
4	Intents	42	194	0.8360
5	wiki.en.bin	300	20	0.8191
6	news.wiki.en.bin	300	249	0.8291
7	news.wiki.en.bin	300	498	0.8388
8	wiki.en.bin	300	498	**0.8398**
9	Intents	42	498	**0.8443**

We also evaluated several variants of encoding the entities and their values:

- We made a distinction between entities and their values that are provided by the user and entities that are provided by the VA in experiments 1-4;
- We used the same set of entities for the user and the VA in experiment 6;
- We used only entity types set by the user ignoring entity values in experiment 5;
- In experiments 7-9, we used not only entities that have been set in the last turn of a dialog, but also entities set in the previous turns.

The best results are obtained by including entities from previous turns in the current turn's input, though the results of experiments 8 and 1 are not significantly different.

3. Interactive Dialog Data Collection Environment

We have designed a platform for dialog data collection. It has two parts – the VA trainer environment and the environment for user-bot communication. Both the VA trainer and the VA user client-side environment interfaces are developed using TypeScript based JavaScript framework Angular 8[3]. The server-side solution uses the .NET Core 3.1 framework[4]. Interaction between the parts of the developed platform is provided with the SignalR asynchronous Data Processing Library[5]. Dialogs collected using the platform are stored in the SQL Server database.

Figure 2. VA trainer environment

The user environment has a chat window where a user can see all previous turns of a dialog and type in an utterance.

The VA trainer environment has three panels.

- On the left, turns of a current dialog are displayed.

---

[3] https://angular.io/

[4] https://docs.microsoft.com/en-us/dotnet/core/

[5] https://github.com/dotnet/aspnetcore/tree/master/src/SignalR

- In the middle, the VA couch can start a new dialog, save or discard a previous dialog, create new actions, and send a response to a user. At first, the action list is populated manually and all actions in the list have equal confidence index 0. After collecting an initial set of dialogs, we train a neural model for next action prediction. In further work, the model returns the list of most probable actions based on the previous conversation history. Actions are sorted by confidence index in descending order. The actions that are most appropriate for context are at the top of the list.
- On the right, the VA couch can define new entities or variables and see their values assigned in a process of conversation.

We use two types of actions. Text actions return a simple textual answer, for example, greet the user or express gratitude. Calculation actions receive variable values in the input and set one or more variable values in the output. In the future, calculation actions will be performed by external services. At this time, the values of entities and variables are set manually by the VA trainer. Examples of calculation actions are determining current time, checking the availability of some product or service, and calculating the total price of something.

4. First Experiments with Latvian Dialogs

We have started collecting dialogs for Latvian using the developed tools. Our chosen domain is transport inquiries. For now, we have collected 160 dialogs in this domain. The set of entities for the transport inquiries domain includes from/to location, transport type, route, date and time, journey length, and timetable, 23 entities in total. We have defined 29 textual actions and 14 calculation actions.

4.1. Data Collection Process

We can distinguish three steps in dialog data collection in transport inquiries domain:

- Learning what type of information is available on the public transport web page, defining types of entities and variables, and the initial set of actions to cover the domain.
- For collecting an initial set of dialogs, the VA couch played both roles – bot's and user's. In this process, the actions and entities were refined in order to improve conversation flow.
- The neural model for the next action prediction was trained and included in a workflow. Dialogs were performed between the VA trainer and different conversational partners.

Similar steps can be employed for dialog collection in any new domain.

4.2. Processing Some More Sophisticated User Utterances

In a conversation related to transport inquiries, a user may be free to ask about the timetables and routes of public transport. An "easy-to-understand" statement with all the input data required is the one containing the transport number, its type, and the stops from/to. The system does not need to ask additional questions to achieve the goal as, for example, in the statement 'What time is the 7th tram from the Ausekļa street to the National theater?'. More problematic are the statements with incomplete information:

- missing information about the start or destination stop;
- a route that involves a change of transport for one or several times;
- instead of a stop name, the user refers to an object or place name used in colloquial speech;
- the user specifies uncertain time value that needs additional calculation, for example, 'late in the evening', 'next Monday', 'as early as possible'.

4.3. Training Next Action Prediction Model in Transport Inquiries Domain

The next action prediction model's architecture for dialogs in Latvian is similar to the one used in experiments with the English data. We use the *fastText* embedding module that is trained on Latvian Wikipedia texts. The maximal number of epochs is 100 with early stopping if accuracy does not improve more than by 0.0001 for three subsequent epochs. The structure of each record in training data representing each turn in a conversation is the following:

- 41 positions are allocated for the one-hot vector representing actions. Only one position in every record has value '1' as there is a single action per turn. For the turns made by a user, the action is always the same meaning, i.e. 'the user has the word';
- 2 positions are reserved for the role of an actor – 'user' or VA;
- 66 positions are reserved for entity types. Both conversation partners can set none to several entities in their turns. Entity values are not used for training;
- 66 positions are reserved for entity types set in all previous dialog turns;
- 300 positions take the embedding vector. If a turn is made by a user it is the user's utterance's embedding vector. If it is VA's turn, all positions have value '0'.

Table 2. Results of 10-fold cross-validation

Nr	Number of dialogs	Average accuracy	Standard deviation
1	33	0.8576	0.0582
2	52	0.8415	0.0469
3	68	0.8536	0.061
4	82	0.8712	0.0617
5	97	0.8919	0.0351
6	122	0.8492	0.0378
7	128	0.8468	0.0309
8	160	0.8617	0.0231

We have trained the model several times by increasing the number of dialogs in training data. The first set of dialogs contains conversations where the bot trainer played both roles (in 48.75 % of collected dialogs) until the set of actions and entities was adjusted for the domain. Further conversations are between the bot trainer and different users (51.25 % of collected dialogs).

The models are tested using 10-fold cross-validation. See **Table 2** for the average accuracy and standard deviation (among 10 folds) of each model.

The average accuracy for the neural model trained with collected data is 0.86 with the Standard deviation 0.0231. These results are close to those we obtained in experiments with DSTC data.

5. Conclusion

In this paper, we described experiments with different training data encodings to train the next dialog action prediction model with higher accuracy. To choose the architecture of the model, the experiments were conducted with the corpus of DSTC dialogs available in English. A variety of sentence embedding algorithms and a different type of entity representation were tested. For further experiments, the architecture of the recurrent neural network with LSTM cells was selected. The training data contains information about sentence embedding, the entities used in a dialog turn, and the history of the entities set in the previous turns of a dialog. As the embedding vector is created for a whole sentence, the effectiveness of the chosen method does not depend on the length of each sentence.

The selected architecture was used to train the next dialog action prediction model using Latvian dialog data. The results are similar. The accuracy for the DSTC model is 0.84, and for the Latvian data, it is 0.86. This verifies that the method is language independent. There are only 160 dialogs in the Latvian dialog set, whereas there are approximately 5,500 in the DSTC corpus. DSTC dialogs are very similar – the user has to choose the type of food, restaurant location, price category, and a few more parameters. It can be concluded that it is not necessary to have such a large number of similar dialogs for training.

6. Acknowledgments

The research leading to these results has received funding from the research project “Competence Centre of Information and Communication Technologies” of EU Structural funds, contract No. 1.2.1.1/18/A/003 signed between IT Competence Centre and Central Finance and Contracting Agency, Research No. 2.3 “Neural network machine learning techniques for automated creating of virtual assistant dialog scenarios”.

References

[1] Deksne D, Vasiļjevs A. Collection of Resources and Evaluation of Customer Support Chatbot. In: Muischnek K, Müürisep K, editors. Human Language Technologies – The Baltic Perspective: Proceedings of the Eighth International Conference Baltic HLT 2018 (Vol. 307). 2018 Sept 27-29; Tartu, Estonia. IOS Press; c2018. p. 30-37.

[2] Jin X, Lei W, Ren Z, Chen H, Liang S, Zhao Y, Yin D. Explicit State Tracking with Semi-Supervision for Neural Dialogue Generation. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management; October 2018; Torino, Italy. New York (NY): ACM; c2018. p. 1403-1412.

[3] Li J, Monroe W, Ritter A, Jurafsky D, Galley M, Gao J. Deep Reinforcement Learning for Dialogue Generation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. c2016. p. 1192-1202.

[4] Liu B, Tür G, Hakkani-Tür D, Shah P, Heck L. Dialogue Learning with Human Teaching and Feedback in End-to-End Trainable Task-Oriented Dialogue Systems. In: Proceedings of the 2018 Conference of

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)(Vol. 1). c2018. p. 2060-2069.
[5] Mo K, Zhang Y, Li S, Li J, Yang Q. Personalizing a dialogue system with transfer reinforcement learning. In: McIlraith SA, Weinberger KQ, editors. Thirty-Second AAAI Conference on Artificial Intelligence. AAAI Press; 2018, April. p. 5317-5324.
[6] Wen TH, Vandyke D, Mrksic N, Gasic M, Rojas-Barahona LM, Su PH., ... & Young S. A network-based end-to-end trainable task-oriented dialogue system. arXiv preprint arXiv:1604.04562. 2016.
[7] Williams J, Zweig G. Deep learning of bots through examples and experience. U.S. Patent Application No. 15/154,017. 2017.
[8] Williams JD, Zweig G. End-to-end LSTM-based dialog control optimized with supervised and reinforcement learning. arXiv preprint arXiv:1606.01269. 2016.
[9] Eric M, Krishnan L, Charette F, Manning CD. Key-Value Retrieval Networks for Task-Oriented Dialogue. In: Jokinen K, Stede M, DeVault D, Louis A, editors. Proceedings of the the 18th Annual SIGdial Meeting on Discourse and Dialogue; 2017 Aug 15-17; Saarbrücken, Germany. ACL; c2017. p. 37-49.
[10] Lowe R, Pow N, Serban I, Pineau J. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In: Proceedings of the SIGDIAL 2015 Conference, The 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2-4 September 2015, Prague, Czech Republic. ACL; c2015. p. 285-294.
[11] Chen H, Ren Z, Tang J, Zhao EY, Yin D. Hierarchical variational memory network for dialogue generation. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web, c2018. p. 1653-1662.
[12] Budzianowski P, Wen TH, Tseng BH, Casanueva I, Ultes S, Ramadan O, Gasic M. Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018. Association for Computational Linguistics; c2018. p. 5016-5026.
[13] Kingma D, Ba J. Adam: A method for stochastic optimization. In: Bengio Y, LeCun Y, editors. Proceedings of 3rd International Conference on Learning Representations; 2015 May 7-9; San Diego, CA, USA.
[14] Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics. 2017; 5:135-146.
[15] Balodis K, Deksne D. Intent detection system based on word embeddings. In: Agre G, van Genabith J, Declerck T, editors. 18th International Conference on Artificial Intelligence: Methodology, Systems, and Applications; 2018 Sept 12-14; Varna, Bulgaria. Springer, Cham; c2018. p. 25-35.
[16] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. 2018.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200608

Intent Detection-Based Lithuanian Chatbot Created via Automatic DNN Hyper-Parameter Optimization

Jurgita KAPOČIŪTĖ-DZIKIENĖ [1]
JSC Tilde Information Technology, Vilnius, Lithuania
Faculty of Informatics, Vytautas Magnus University, Kaunas, Lithuania

Abstract. In this paper, we tackle an intent detection problem for the Lithuanian language with the real supervised data. Our main focus is on the enhancement of the Natural Language Understanding (NLU) module, responsible for the comprehension of user's questions. The NLU model is trained with a properly selected word vectorization type and Deep Neural Network (DNN) classifier. During our experiments, we have experimentally investigated fastText and BERT embeddings. Besides, we have automatically optimized different architectures and hyper-parameters of the following DNN approaches: Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM) and Convolutional Neural Network (CNN). The highest accuracy=~0.715 (~0.675 and ~0.625 over random and majority baselines, respectively) was achieved with the CNN classifier applied on a top of BERT embeddings. The detailed error analysis revealed that prediction accuracies degrade for the least covered intents and due to intent ambiguities; therefore, in the future, we are planning to make necessary adjustments to boost the intent detection accuracy for the Lithuanian language even more.

Keywords. NLU, Intent detection, LSTM, BiLSTM, CNN, hyper-parameter optimization, fastText and BERT embeddings, the Lithuanian language

1. Introduction

The modern society is not imaginable without conversational chatbots that are accurate, fast, tireless and available 24/7. Chatbots can successfully boost human work efficiency, but cannot replace them completely so far. However, the demand for chatbots is increasing exponentially, especially in the customer service space.

Starting from the very first keyword search-based chatbot ELIZA (asking itself instead of answering user's questions) [1], this NLP field has made tremendous progress. Nowadays, different types of chatbots exist: according to the communication channel (voice-enabled or text-based); knowledge domain (closed, general or open domain); provided service (interpersonal or intrapersonal); used learning methods (rule-based, retrieval-based or machine learning-based); and provided answers (extracted or generated answers).

The chatbot as the whole system can consist of several important modules: NLU (dealing with the machine reading comprehension), slot filling (searching for specific

[1] Corresponding Author: Jurgita Kapočiūtė-Dzikienė; JSC Tilde Information Technology, Naugarduko str. 100, LT-03160 Vilnius, Lithuania; E-mail: jurgita.k.dz@gmail.com.

pieces of information with respect to some named entity), dialog management (determining the context and guiding the dialog), even emotion detection (reacting to the user's sentiments accordingly), etc. Various chatbot features as an appearance or a friendly human-like behavior are important, however, the most important function remains the correct comprehension of user's questions and adequate responses to them. In this paper, we are solving the intent detection problem that classifies user utterances into predefined intent categories (related to chatbot answers).

Widely-known intelligent chatbots as Siri, Cortana, Alexa, Google assistant, etc., support only a small group of languages and Lithuanian is not among them. There was an attempt to solve an intent detection problem for morphologically complex languages, including Lithuanian [2]. This research describes experiments with fastText embeddings using the CNN and the Feed Forward Neural Network (FFNN) classifiers (architectures based on expert insights). Despite the significance of this research for the Lithuanian language, the intent detection model was trained and tested on the artificial data (i.e., English benchmark datasets translated into Lithuanian) and a rather small number of intents (varying from 2 up to 8).

In our research, we are solving the intent detection problem for the Lithuanian language by training and testing the NLU model on the real dataset of 41 intent. The contribution of this research is that we seek for the best intent detection model for the Lithuanian language by exploring: 1) two types of word embeddings; 2) three types of classifiers (CNN, LSTM, BiLSTM); 3) different Deep Neural Network architectures; 4) different hyper-parameters. DNN architectures and hyper-parameter values were tuned automatically in the parameter optimization process.

2. Related Work

If all outdated rule-based and keyword-based approaches are excluded, the remaining ones are mainly focused on the Information Retrieval (IR) and Machine Learning (ML) techniques. IR-based chatbots are typically used in cases when it is essential to find relevant information from huge data collections [3] or unstructured documents [4]. In our research, we experiment with the dataset containing question-answer pairs (where questions are grouped into categories representing related answers). This particular shape of the data allows us to focus on the supervised ML approaches [5].

The majority of the intent detection research is focused on the English and Chinese languages having enough resources to experiment and create accurate NLU models (a more detailed review on different intent detection methods can be found in [6]).

The research in [7] compares different chatbot platforms. Authors test the most popular NLU services on a large multi-domain dataset of 21 domains and 25K of utterances. The investigation reveals that *IBM Watson* significantly outperforms other platforms as *Dialogflow*, *MS LUIS* and *Rasa* that also demonstrate very good results. Three English benchmark datasets, i.e., *askUbuntu*, *chatbot* and *webApps* [8] were used in the experiments [9]. Authors introduce a sub-word semantic hashing technique to process input texts before classification. After vectorization, the following classifiers are explored: Ridge Classifier – RC, K-Nearest Neighbors, Multilayer Perceptron, Passive Aggressive – PA, Random Forest – RF, Linear Support Vector – LSV, Stochastic Gradient Descent, Nearest Centroid, Multinomial Naive Bayes, Bernoulli Naive Bayes, K-means. On the *askUbuntu* dataset, RC and LSV classifiers achieved the highest accuracy. PA and RF were the best on *chatbot* and *webApps*, respectively. Authors claim that with the determined classifiers and their sub-word semantic hashing technique, they were able to achieve state-of-the-art performance on these benchmark datasets. The experiments in [10] only confirm the fact that vectorization plays an important role in intent detection. Authors use the BiLSTM classifier with gloVe embeddings enriched with semantic lexicons (containing synonyms, and antonyms) and achieve superiority over the naïve one-hot vectorization on the *ATIS* and *Places* datasets from MS Cortana.

The problems often emerge when some question is out-of-scope the intent detection model is trained to cover. The supervised ML is also the right solution for this type of problems. The research in [11] solves an open intent discovery problem with the dataset of 25K of utterances in two stages: the first stage predicts if the utterance contains some intent and if it does, the second stage tries to predict it. Authors use BiLSTM with the Conditional Random Fields (CRF) method on a top of it and a self-attention mechanism to learn long distance dependencies. The offered approach outperforms state-of-the-art baselines. In the similar work [12], authors use a dataset containing queries that are out-of-scope (covering none of the existing intents) and in-the-scope (covering one of 150 available intents). A range of different approaches has been explored in this research: fastText classifier, BERT classifier, Support Vector Machine, CNN, DialogFlow, Rasa and Multi-Layer Perceptron. The research discovered the BERT classifier to perform well in-the-scope; unfortunately, all approaches struggled to identify out-of-scope queries. Our dataset contains queries only in-the-scope; therefore, we will focus on the closed domain intent detection problems only.

Intent detection problems are sometimes tackled together with the slot filling. Authors in [13] experimentally prove that the proposed joint BERT model outperforms BERT models for the intent detection and slot filling trained separately. Their proposed joint model is trained on the dataset of 72 slots and 8 intents and achieves significant improvement over the attention-based Recurrent Neural Network (RNN) models and slot-gated models. In similar research [14], authors use *SNIPS-NLU* and *ATIS* datasets and propose a Capsule-Based Neural Network model. The architecture consists of three types of capsules: WordCaps (to learn word representations), SlotCaps (to classify words by their slots) and IntentCaps (to determine the intent). The offered method outperforms other popular NN-based approaches. Authors in [15] present an attention-based encoder-decoder NN model for the joint intent detection and slot filling task. The method encodes sentences using the CNN-BiLSTM hybrid approach and decodes using the attention-based RNN with aligned inputs. The authors experimentally prove that the offered approach achieves better performance compared to the other popular DNN-based approaches. Since our dataset does not contain necessary annotations for the slot filling, we will focus on the intent detection problem only; however, the analysis of similar research especially encourages us to test BERT embeddings and DNN classifiers.

3. The Dataset

Our NLU module training and testing was done on the real Lithuanian data: i.e., dataset containing question-answer pairs about the company's Tilde product "Tildes Biuras" (questions about prices for different users, licenses, supported languages, available dictionaries, used technologies, etc.). The whole dataset contains 41 intents (chatbot outputs/answers), each covered by at least 5 instances (user inputs/questions). When solving the intent detection problem, the following Lithuanian language-dependent and spoken language-dependent features have to be considered. The Lithuanian language is morphologically complex, highly inflective, has a rich vocabulary and a rich word-derivation system. Besides, questions (to correspond the real conversation conditions) are formulated in the spoken Lithuanian language: they have relatively free word-order (that is typical for the Lithuanian language), contain stylistically irregular constructions (typical for the spoken language).

The dataset was split into training and testing sub-sets (statistics about these splits can be found in Table 1).

Table 1. Statistics about the training/testing dataset used in our experiments

Number of intents	Number of questions	Number of words	Number of distinct words	Avg. number of words per question
Training dataset				
41	365	1,857	699	5.09
Testing dataset				
41	144	751	435	5.22

4. Methodology

The nature of the dataset (presented in Section 3) makes it possible to use it with the supervised ML approaches [5]. A solving task is a typical classification problem and can be formally defined as follows:

Let $D = \{d_1, d_2, \ldots, d_n\}$ be a set of instances (user inputs/questions). Let $C = \{c_1, c_2, \ldots, c_m\}$ be a set of classes (chatbot outputs/answers/intents). Each d_i can be attached to only one c_j; besides $m > 2$ that restricts this task to a single-label multi-class classification problem. D is mapped to C according to the logic Γ ($\Gamma : D \to C$). The main goal is to train a classifier on the training dataset D^T that could find the best approximation of Γ and demonstrate as higher accuracy as possible on the testing dataset D^T.

The formulated NLU problem is considered as AI-hard problem (meaning that the software should be learned to behave as intelligent as a human). The state-of-the-art approaches focus on DNN classifiers applied on a top of neural word embeddings (see Section 2) and for this reason we have chosen to explore these approaches as well. The whole package should contain the correct selection of the vectorization type, classifier, its architecture and hyper-parameter values. Thus, we have investigated:

- FastText [2] [16] and BERT [3] [17] word embeddings. FastText embeddings (offered by Facebook AI) are suitable for the non-normative texts: they are composed of the n-gram vectors; therefore, embeddings of misspelled words are close to their correct equivalents. BERT embeddings (offered by Google) are robust to disambiguation problems (because they have mechanisms to represent homonyms with different vectors depending on their context).
- LSTM [18], BiLSTM [19], CNN [20] (with the 1D convolution adjusted for the text classification [21]) classifiers. Both LSTM and BiLSTM are adjusted to process the sequential data and to overcome limitations of the simple RNN suffering from the short memory problem (due to a vanishing gradient). LSTM

[2] The Lithuanian fastText embeddings (offered by Facebook's AI Research Lab) for our experiments were downloaded from https://fasttext.cc/docs/en/crawl-vectors.html.

[3] We have used the Google's BERT service with the base multilingual cased 12-layer, 768-hidden, 12-heads model for 104 languages (including Lithuanian) downloaded from https://github.com/hanxiao/bert-as-service.

allows the data stream only forward (i.e., from-the-past-to-the-future), whereas BiLSTM process the text in both directions (i.e., forward and backward) and can acquire how succeeding words impact the current time moment. The CNN method is adjusted to seek for the influential text patterns (sequences of words, word n-grams) the most relevant to the intent.

- DNN architectures, having different numbers of hidden layers (i.e., series of convolutional layers in CNN; simple or stacked LSTM and BiLSTM).
- Discrete and real DNN hyper-parameter values: numbers of neurons (100, 200, 300, 400); dropouts (values from an interval [0,1]); recurrent dropouts ([0, 1]); activation functions (*relu*, *softmax*, *tanh*, *elu*, *selu*); optimizers (*Adam*, *SGD*, *RMSprop*, *Adagrad*, *Adadelta*, *Adamax*, *Nadam*); batch sizes (32, 64) and numbers of epochs (20, 30, 40, 50).

All this results in a huge number of options and expert knowledge is not always capable of selecting the most accurate one. The tuning of DNN architectures and hyper-parameters was performed automatically with the parameter optimization library *Hyperas*[4] and two optimization algorithms (the optimization process took 100 iterations):

- *tpe.suggest* (Tree-structured Parzen Estimator) [22], which organizes hyper-parameters into a tree-like space. This Bayesian modelling approach decides which set of hyper-parameters should be tried in the next iteration based on the distribution of previous results.
- *random.suggest* performs random search over a set of hyper-parameters.

All methods were implemented in Python using *Tensorflow* and *Keras*[5].

5. Experiments and Results

All experiments described in Section 4 were carried out with the dataset described in Section 3. The training dataset (see Table 1) was shuffled and used for the DNN hyper-parameter optimization: 80 % was used for training and the rest 20 % for validation. The best determined model (giving the highest accuracy on the validation dataset) was evaluated with the testing dataset. The accuracy (in eq. 1, where $N_{correct}$ and N_{all} represents instances with the correctly predicted intents and all tested instances, respectively) was used as the evaluation metric. The highest achieved accuracies with different DNN methods and vectorization types are presented in Table 2.

$$accuracy = \frac{N_{correct}}{N_{all}} \tag{1}$$

The approach is considered reasonable if the accuracy is above $random = \sum_j P^2(c_j)$ (where c_j is a probability of an intent) and majority=$max(P(c_j))$ baselines in our case, equal to ~0.04 and ~0.09, respectively. The McNemar test [23] with α=0.05 was used to determine if the differences between the obtained results are statistically significant.

---

[4] More about *Hyperas* is in https://github.com/maxpumperla/hyperas.

[5] https://www.tensorflow.org/ and https://keras.io/.

The architecture of the best determined approach which happened to be the CNN classifier with BERT embeddings is presented in Figure 1 (plotted with the *plot_model* utility function in *Keras*). Since not all determined optimal hyper-parameters can be plotted in the figure, we report them here: *selu* activation function after Conv1D layer; *softmax* activation function after dense; *batch_size*=64; *epochs*=20; *optimizer*=Nadam; *dropout rate*=0.467.

Table 2. The highest achieved accuracies (after hyper-parameter optimization) with different DNN classifiers, vectorization types, and parameter optimization strategies. The best result is in bold, the underlined results determine that they do not differ significantly from the very best

	fastText embeddings		
	LSTM	**BiLSTM**	**CNN**
tpe.suggest	0.243	0.563	0.639
random.suggest	0.278	0.556	0.646
	BERT embeddings		
	LSTM	**BiLSTM**	**CNN**
tpe.suggest	0.681	0.653	**0.715**
random.suggest	0.681	0.694	0.708

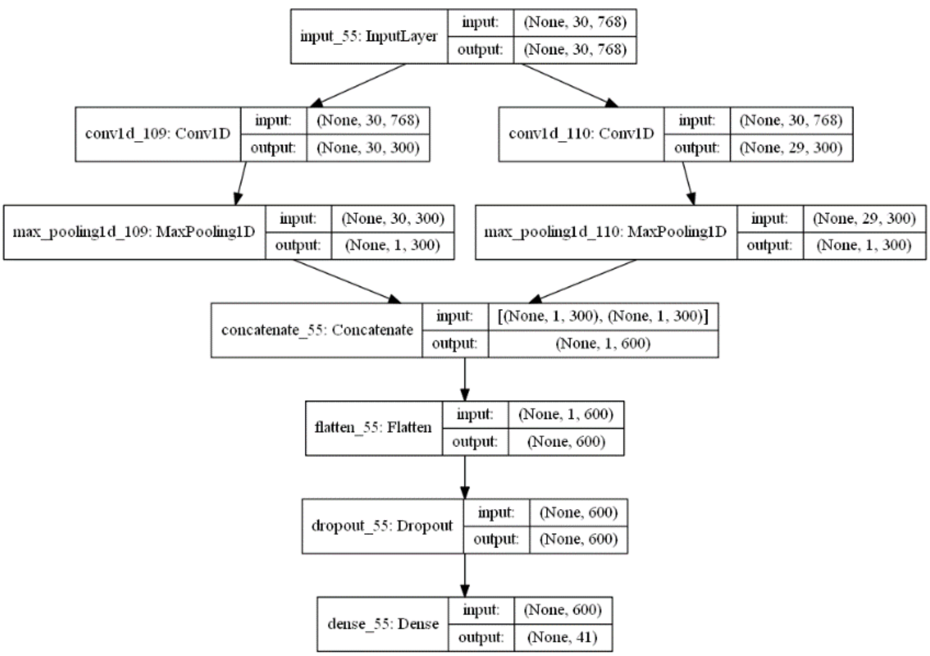


Figure 1. The best determined intent detection classifiers and its architecture: CNN with BERT embeddings

As presented in Table 2, all results are reasonable, because are above random and majority baselines. The highest accuracy=~0.715 is achieved with the BERT embeddings and the CNN classifier after parameter optimization (the best architecture is presented in Figure 1 and the best hyper-parameters are presented in Section 5). The superiority of the CNN classifier over the other approaches as LSTM and BiLSTM reveals that for this intent detection problem, to detect patterns (as word n-grams) is more important than to cope with the sequential nature of the input.

The *tpe.suggest* strategy allowed to optimize parameters even better compared to the *random.suggest*. However, there is no big difference how these parameters were optimized, the most important is that the accurate model has been found.

To see how significantly results differ from the best achieved (*accuracy*=~0.715) with BERT + CNN + *tpe.suggest,* we have performed the McNemar evaluation (calculated *p* values are in Table 3).

Table 3. The *p* values indicating how much the results are statistically significant between the best result with BERT + CNN + *tpe.suggest* strategy and other approaches

	fastText embeddings		
	LSTM	**BiLSTM**	**CNN**
tpe.suggest	4.49E-08	0.039	<u>0.300</u>
random.suggest	2.62E-07	0.032	<u>0.348</u>
	BERT embeddings		
	LSTM	**BiLSTM**	**CNN**
tpe.suggest	<u>0.668</u>	<u>0.402</u>	-
random.suggest	<u>0.668</u>	<u>0.828</u>	<u>1.000</u>

We have evaluated the accuracy for each intent separately (see the statistics in $x_y \to z$, where x, y, and z are number of testing instances, number of intents having x instances and the accuracy, respectively): $2_{10} \to 1.00$, $2_7 \to 0.50$, $2_7 \to 0.00$, $3_1 \to 0.67$, $3_1 \to 0.00$, $4_1 \to 1.00$, $4_2 \to 0.75$, $4_1 \to 0.50$, $5_1 \to 1.00$, $5_2 \to 0.80$, $5_1 \to 0.40$, $5_1 \to 0.40$, $6_2 \to 1.00$, $6_2 \to 0.83$, $6_1 \to 0.33$, $10_1 \to 0.90$, $14_1 \to 1.00$. Our solving intent detection problem is very challenging (41 intents, some of them are covered with only a few questions). The error analysis revealed that intents covered by more instances are better predicted. Besides, some intents are ambiguous (leading to very similar answers) and can be aggregated. Still, the accuracy of the NLU model is acceptable; we are planning to increase it even more by filling the training dataset with the new instances (especially by adding more instances to the least covered intents) and uptraining the new version of the model with the parameters already determined in this research.

6. Conclusions

This paper presented the intent detection problem solving for the Lithuanian language. This NLU problem for the Lithuanian language is tackled for the first time by using the real data and by comparing a wide variety of different approaches. We performed the automatic parameter optimization with three classifiers (LSTM, BiLSTM, CNN), two types of word embeddings (fastText, BERT), different DNN architectures (deeper and shallower) and various DNN hyper-parameter values.

The tackled task was especially challenging due to the following reasons: 1) the dataset contains many intents (41); 2) each intent was covered by only a few instances (~8.9 instances of which only ~7.1 are used for training and ~1.8 for validation). The experimental investigation revealed the superiority of the CNN classifier with BERT embeddings over other approaches; the best approach achieved ~0.715 of the accuracy. The error analysis revealed that intents covered by more instances are more reliable and recognized better. It allows us to assume that after adding more instances to the least covered intents and retraining the model, the accuracy will boost even more. In the future research, we are planning to expand the number of intents and instances.

Acknowledgements

The project *Bot Cloud* (No. J05-LVPA-K-03) was funded from EU funds under the measure *General Science and Business Projects* and *1st PRIORITY. Promotion of research, experimental development and innovation*.

References

[1] Weizenbaum, J. ELIZA–a computer program for the study of natural language communication between man and machine. Communications of the ACM 1996; 9:36–45.
[2] Balodis K, Deksne D. FastText-Based Intent Detection for Inflected Languages. Information 2019;10(5): 161.
[3] Lommatzsch A, Katins J. An Information Retrieval-based Approach for Building Intuitive Chatbots for Large Knowledge Bases. Proceedings of the LWDA conference 2019; 343–352.
[4] Zhao Y, Duan N, Bao J, Chen P, Zhou M, Li Z, Zhou J. DocChat: An Information Retrieval Approach for Chatbot Engines Using Unstructured Documents. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) 2016;516–525.
[5] Sebastiani F. Machine Learning in Automated Text Categorization. ACM Computing Surveys 2002;34: 1–47.
[6] Liu J, Li Y, Lin M. Review of Intent Detection Methods in the Human-Machine Dialogue System. Journal of Physics: Conference Series 2019; 1267 012059.
[7] Liu X, Eshghi A, Swietojanski P, Rieser V. Benchmarking Natural Language Understanding Services for building Conversational Agents. CoRR 2019;abs/1903.05566.
[8] Braun D, Hernandez-Mendez A, Matthes F, Langen M. Evaluating Natural Language Understanding Services for Conversational Question Answering Systems. Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue 2017;174–185.
[9] Shridhar K, Sahu A, Dash A, Alonso P, Pihlgren G, Pondeknath V, Simistira F, Liwicki M. Subword Semantic Hashing for Intent Classification on Small Datasets. CoRR 2018; abs/1810.07150.
[10] Kim J-K, Tur G, Celikyilmaz A, Cao B, Wang Y-Y. Intent Detection using Semantically Enriched Word Embeddings. IEEE Spoken Language Technology Workshop 2016 Sep; 414–419.
[11] Vedula N, Lipka N, Maneriker P, Parthasarathy S. Towards Open Intent Discovery for Conversational Text. CoRR 2019;abs/1904.08524.
[12] Larson S, Mahendran A, Peper, JJ, Clarke Ch, Lee A, Hill P, Kummerfeld JK, Leach K, Laurenzano MA, Tang L, Mars J. An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction. Proc. of the 2019 Conference on Empirical Methods in NLP and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) 2019 Nov; 1311–1316.
[13] Chen Q, Zhuo Z, Wang W. BERT for Joint Intent Classification and Slot Filling. CoRR 2019; abs/1902.10909.
[14] Zhang Ch, Li Y, Du N, Fan W, Yu PS. Joint Slot Filling and Intent Detection via Capsule Neural Networks. CoRR 2018;abs/1812.09471.
[15] Wang Y, Tang L, He T. Attention-Based CNN-LSTM Networks for Joint Intent Detection and Slot Filling. Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data 2018;LNAI;11221:250–261.
[16] Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics 2017;5:135–146.
[17] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. CoRR 2018; abs/1810.04805.
[18] Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Computation 1997;9(8):1735–1780.
[19] Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks 2005;18(5-6):602–610.
[20] Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE 1998;2278–2324.
[21] Kim Y. Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) 2014;1746–1751.
[22] Bergstra J, Bardenet R, Bengio Y, Kégl B. Algorithms for Hyper-Parameter Optimization. Advances in Neural Information Processing Systems 2011;24:2546–2554.
[23] McNemar Q. Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages. Psychometrika 1947;12(2):153–157.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200609

Towards Hybrid Model for Human-Computer Interaction in Latvian

Inguna SKADIŅA [1] and Didzis GOŠKO
Institute of Mathematics and Computer Science, University of Latvia, Latvia

Abstract. Human-computer interaction, especially in form of dialogue systems and chatbots, has become extremely popular during the last decade. The dominant approach in the recent development of practical virtual assistants is the application of deep learning techniques. However, in case of less resourced language (or domain), the application of deep learning could be very complicated due to the lack of necessary training data. In this paper, we discuss possibility to apply hybrid approach to dialogue modelling by combining data-driven approach with the knowledge-based approach. Our hypothesis is that by combining different agents (general domain chatbot, frequently asked questions module and goal oriented virtual assistant) into single virtual assistant we can facilitate adequacy and fluency of the conversation. We investigate suitability of different widely used techniques in less resourced settings. We demonstrate feasibility of our approach for morphologically rich less resourced language Latvian through initial virtual assistant prototype for the student service of the University of Latvia.

Keywords. Human-computer Interaction, goal-oriented virtual assistant, question answering, Latvian language, multi-agent systems, natural language processing

1. Introduction

Human-computer interaction, especially in form of dialogue systems and chatbots, has become extremely popular during the last decade. Success of IBM Watson, Apple Siri, Amazon Alexa and some other virtual assistants as well as new research perspectives opened by deep learning technologies have been the main drivers in human-computer interaction, question answering and even human centric AI.

The dominant approach in the recent development of practical virtual assistants is the application of deep learning techniques to learn directly from text samples and other relevant data (e.g. [1], [2], [3]). In many cases, though, the interaction problem is reduced to a text classification task, and rather basic bag-of-words classifiers, therefore, provide a strong baseline.

Because of its complexity, most of the current research focuses on end-to-end machine learning and on resource-rich languages like English, paying little attention to the less-resource languages and more intelligent approaches. The lack of training data of sufficient size, and the morphological richness and flexible word order for many languages, including Latvian, are among the main reasons why models for widely used languages are not directly applicable to the less-resourced inflected languages.

[1] Corresponding Author: Inguna Skadiņa; Artificial Intelligence Laboratory, Institute of Mathematics and Computer Science, University of Latvia, Raiņa bulv. 29, Riga, LV-1459, Latvia; E-mail: inguna.skadina@lumii.lv.

To overcome this bottleneck, we propose a hybrid approach to the development of virtual assistant by combining the data-driven approach with the knowledge-based approach. The main reason for combined approach is the lack of training data to build reasonable quality end-to-end dialogue system. Our hypothesis is that by combining different agents into single virtual assistant, we can facilitate adequacy and fluency of the conversation.

In this paper, we discus architecture of virtual assistant for the university student service, its main components and performance. We demonstrate that the chosen approach could be successfully applied to the development of domain specific virtual assistants for less resourced languages.

2. Related Work for Latvian

For Latvian, several experiments have been conducted and different virtual assistants (VAs) or their prototypes are implemented. Among the first are a multimodal assistant that teaches multiplication for Latvian children, and an assistant that teaches basic Lithuanian phrases for Latvians, both developed by the company Tilde [4]. Tilde has also recently developed customer service assistant for the State Register of Enterprises *Una*[2] and several virtual assistants for the public sector, e.g., *Justs* for the Land Registry[3], *Toms* for the State Revenue Service[4] and others. The most recent virtual assistant can answer common questions about COVID-19[5]. These virtual assistants use dialogue state tracking mechanism implemented through dialogue graph and state of the art intent detection system [5].

The Artificial Intelligence Laboratory of the Institute of Mathematics and Computer Science, University of Latvia, has developed a prototype assistant for partial automation of the customer service operations for telecommunication domain [6]. The agent consists of an intent detection system for identifying the types of customer requests that it can handle, a slot filling information extraction system that integrates with the customer service database for a rule-based treatment of the common scenarios, and a template-based language generation system that builds response candidates that can be then approved or amended by customer service operators.

3. Virtual Assistant for the University Student Service

While traditional goal-oriented dialogue systems usually use single task-oriented agent, we propose to combine several agents into virtual assistant (**Figure 1**). The proposed model includes:

- General domain chatbot – the chatbot responds to greetings, introduces with domain, and keeps conversation when user utterance is out of domain;
- Frequently asked question (FAQ) module – with the help of machine learning techniques the model is trained to answer the most common questions of students regarding the University of Latvia;

[2] https://www.ur.gov.lv/en/.
[3] https://www.zemesgramata.lv/.
[4] https://www.vid.gov.lv/en.
[5] https://covidbots.lv/.

- Goal-oriented virtual assistant – supports short dialogues, includes common constituents of virtual assistant – an intent detection, a slot filling and a dialogue state tracker.

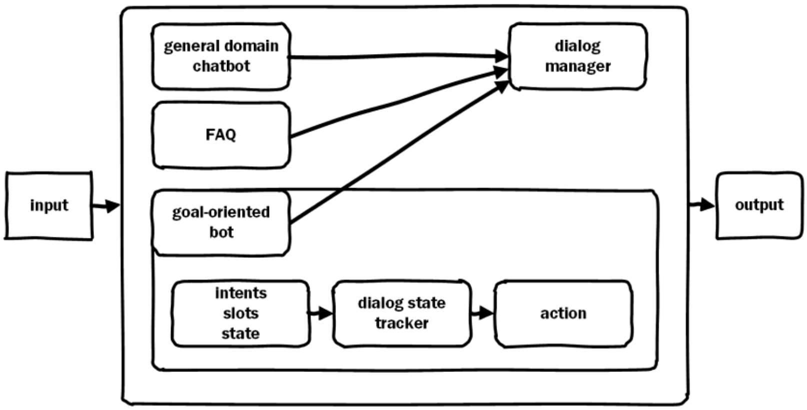


Figure 1. General overview of proposed model

The dialogue manager selects the most appropriate constituent according to the confidence score returned by the constituents. If the confidence score is below the threshold for all above-mentioned constituents, the fallback mechanism (request to reformulate question) is performed.

The virtual assistant is trained through DeepPavlov framework [7] that supports a wide range of possible model configurations and has been successfully applied in different dialogue modelling and natural language understanding (NLU) challenges.

The main reason for such combined approach is the lack of the domain data to train end-to-end dialogue system that can fully support questions related to the work of student service of the University of Latvia. In addition, our hypothesis is that combining several models that support a particular task makes communication with the virtual assistant more fluent and, thus, user friendly. Finally, such approach also allows to combine different NLP and NLU processing strategies which is very import in low resourced conditions. Similar approach, i.e., the blending of different skills for building an open-domain conversational bot in resource rich settings has been recently discussed by Facebook [8].

3.1. Chatbot Module

The chatbot constituent supports out of domain conversations. The pattern-matching approach is used to decide on most appropriate dialogue action (e.g., greeting, chit-chat, general information, etc.). This constituent also provides fallback mechanism for situations when none of task-oriented constituents have high enough confidence score to continue dialogue (in our case, less than 0.2).

3.2. Frequently Asked Questions

The frequently asked questions (FAQ) module is the main constituent in many virtual assistants. Traditionally, FAQ model calculates similarity between user's utterance and training samples.

Our FAQ module is trained to answer questions regarding studies at the University of Latvia (UL). For FAQ module, training questions (14 in total) from the UL website[6] have been paraphrased to have at least five samples for each question (as it has been suggested for low resourced conditions in [5]). Several possible approaches have been examined. Two different vectorizers (tf_idf and fasttext [9]) together with different classifiers (cosine similarity and logistic regression) have been investigated.

Evaluation results for FAQ module are summarized in Table 1. All investigated approaches have demonstrated high accuracy. However, in our case, simpler methods demonstrate slightly better results. This could be explained by the low resource settings of our experiment.

Table 1. Evaluation results of FAQ classifiers

Vectorizer	Classifier	Accuracy
tf_idf	cosine similarity	0.99
tf_idf	logistic regression	0.99
Fasttext ([10], 300 dimensions)	cosine similarity	0.98
Fasttext (Latvian, 100 dimensions)	cosine similarity	0.97

3.3. Goal Oriented Bot

The goal oriented bot is designed to support conversation about three narrow domains: working hours of different units of the University of Latvia, academic leave and university entry requirements. The goal oriented virtual assistant is based on Hybrid Code Networks [11] that combine recurrent neural networks (RNN), domain knowledge and templates for actions. It includes an intent detection, a slot filling, a dialogue state tracking and an answer selection (generation) constituents (see the bottom of **Figure 1**). The action is predicted (learned by RNN) taking into account the dialogue state, user's utterance, intents and slots (as annotated in training data).

For building a goal oriented bot, data from student forum containing several thousands of utterances were analyzed. In this dataset, the working hours of different university units (115 dialogues), academic leave (47 dialogues) and university entry requirements (24 dialogues) were identified as more frequent topics and thus were selected as data for annotation and training.

We investigated several intent detection classifiers for three selected topics. Our classifiers are built on pre-trained language models – Latvian BERT [12], MultiBERT [13][7] and Fasttext. 140 utterances from the dataset were used as training data, 15 as validation set and 31 as test set. Evaluation results in **Table 2** show that in case of small dataset of unbalanced training data, simpler mechanisms (i.e. Fasttext) and, in particular,

[6] https://www.lu.lv/gribustudet/jautajumi-un-atbildes/.

[7] https://github.com/google-research/bert/blob/master/multilingual.md.

solutions developed specifically for Latvian (i.e. Fasttext for Latvian, LV-BERT) result in better performance.

Table 2. Evaluation results for intent detection: accuracy and F1 for different settings

Embeddings	**Validations set**		**Test set**	
	accuracy	F1	accuracy	F1
Fasttext (300 dimensions)	0.8387	0.7861	0.6774	0.6724
Fasttext (Latvian, 100 dimensions)	**1**	**1**	**1**	**1**
MultiBert	0.6667	0.5411	0.6452	0.5397
LV-BERT	0.9333	0.9327	0.9677	0.9683

For slot filling, we integrated named entity recognizer (NER) trained on MultiBERT and thus allowing to perform zero-shot transfer [14]. It supports 19 tags, including organization, event, date and person that are the most important in our scenario. An example in **Figure 2** illustrates output of the named entity recognizer. Although the NER used in prototype demonstrates reasonable quality for Latvian, we are investigating recently developed Latvian NER [12], which includes the necessary named entity types and could also be adapted for the recognition of domain specific named entities.

User: Kā strādā <u>Ārija Sproģe</u>? (*What are the working hours for Ārija Sproģe?*)
Slotfill: [PERSON: Ārija Sproģe]

User: Kā strādā <u>Humanitāro zinātņu fakultāte</u>? (*What are the working hours for the Faculty of Humanities?)*
Slotfill: [ORGANIZATION: Humanitāro zinātņu fakultāte]

User: Vai <u>Humanitāro zinātņu fakultāte</u> strādā <u>jūnijā</u>? (*Is the Faculty of Humanities open in June?*)
Slotfill: [ORGANIZATION: Humanitāro zinātņu fakultāte; DATE: jūnijā]

Figure 2. Slot filling examples for working hours scenario; named entities are underlined

For dialogue state tracking and dialogue action prediction short dialogues were created for academic leave scenario using the data from the student forum. It needs to be mentioned that queries in student forum differ from typical dialogue – they are longer, one entry often contains several utterances (typical entry includes a greeting, a question and attitude statements), and sometimes even several questions. Therefore, we modified these entries as short dialogues of 2-5 turns. The dialogue action prediction on our dialogue corpus is trained with recurrent neural network model of the DeepPavlov framework that predicts dialogue action from the utterance, its slots, intents and the current state of the dialogue.

Finally, the predicted dialogue action allows us to create (select) an answer to the user's input. Currently, only fixed response is given as an output. Our next steps include data acquisition from the knowledge base and more intelligent natural language generation.

4. Initial Results of Evaluation

Our prototype currently includes general domain chatbot, FAQ model and goal oriented bot for academic leave scenario. The goal oriented bot is initialized by the dialogue manager when the academic leave intent is recognized as most probable. Example in **Figure 3** illustrates human interaction with the virtual assistant. The interaction starts with a greeting from general domain chatbot, followed by question answering by FAQ module. Finally, the goal oriented bot performs mini-dialogue regarding the academic leave in situation when a student has not finished semester successfully.

User: Labdien! (*Hallo!*)
VA (general domain bot): Labdien! Varu atbildēt uz jautājumiem par Latvijas Universitātti (*Hallo! I can answer questions related to the University of Latvia*)
User: Kam man nepieciešama akadēmiskā izziņa?" (*Why I need academic statement?)*
VA (FAQ): Akadēmiskā izziņa ir oficiāls studējošā apgūto studiju kursu, iegūto kredītpunktu un novērtējumu uzskaitījums no augstskolas studiju dokumentācijas (*An academic statement is an official inventory of study courses acquired by a student, obtained credits and assessments from the study documentation of a higher education institution*)
User: Paldies! Vai ir iespējas paņemt akadēmisko gadu? (*Thank you! Are there opportunities to take an academic year?*)
VA (goal oriented bot): Ir jāraksta iesniegums un jāiet pie programmas direktora. (*You have to write an application and go to the program director.*)
User: Ja ir parāds, ko tad? (*If there is a debt?*)
VA (goal oriented bot): Ja ir parāds, tad akadēmisko pārtraukumu nevar saņemt. (*If there is a debt, then the academic break cannot be received.*)
User: Paldies! Kur ir tuvākā kafejnīca? (*Thank you! Where is the nearest café?*)
VA (fallback): Isti nesapratu, lūdzu atkārto (*Could you, please, repeat*)
User: Atā! (*bye*)
VA (general domain bot): atā (*bye*)!

Figure 3. Dialogue sample

We evaluated different configurations of the goal-oriented bot for academic leave scenario (**Table 3**). The best result – 0.89 turn accuracy – has been achieved with the simplest configuration. This could be explained with rather small number of training examples and simplicity of dialogues.

Table 3. Evaluation results for different configurations of goal oriented bot.

Model	**Embeddings**	**Intents**	**Test (turn accuracy)**
Bot with intent detection and attention mechanisms	Latvian Fasttext	Latvian Fasttext	0.7895
Bot with intent detection	Latvian Fasttext	Latvian Fasttext	0.8421
Simple bot	Latvian Fasttext	No	**0.8947**

While obtained initial evaluation results are promising and confirm the suitability of the chosen approach, the prototype is only at its initial stage –current implementation needs to be extended to support more dialogues by the goal oriented bot. Moreover, deep evaluation by real user is necessary for better understanding of strengths and weaknesses of the chosen approach.

5. Next Steps and Conclusion

In this paper, we discussed the architecture of a hybrid virtual assistant that supports conversation in low resource settings. The virtual assistant prototype is designed to support communication between student service of the University of Latvia and students. The proposed prototype includes modules for answering frequently asked questions, a goal-oriented bot to support mini-conversations on most common topics and a chit-chat constituent that facilitates conversation in uncertain conditions.

While the prototype and evaluation results demonstrate feasibility of the proposed approach, our plan is to continue the development of the goal oriented bot by extending and deepening the topics of conversation. In particular, we plan to include knowledge base as part of the output generation constituent, making dialogue more flexible and easier extendable. Besides practical solutions, two research directions are foreseen. At first, we plan to investigate the role of pre-trained language models in low resource settings. This includes methods for NER adaptation to support slot filling for domain specific entity classes. Secondly, we aim to investigate the application of frame-semantic parsing methods for slot filling. The data for these experiments are already being annotated for the working hours scenario.

Finally, we also plan to integrate our virtual assistant into conversation platforms allowing real conversations with users. This will allow us to understand better the efficiency of our model.

Acknowledgements

This research is funded by the Latvian Council of Science project “Latvian Language Understanding and Generation in Human-Computer Interaction”, project No. lzp-2018/2-0216. We would like to thank Gunta Nešpore-Bērzkalne for data analysis and topic selection, and Raivis Skadiņš for technical support during abstract/paper preparation.

References

[1] Adiwardana D, Luong MT, So DR, Hall J, Fiedel N, Thoppilan R, et al. Towards a Human-like Open-Domain Chatbot. arXiv.org: https://arxiv.org/pdf/2001.09977.pdf, (2020).
[2] Zhang Y, Sun S, Galley M, Chen YC, Brockett C, Gao X, et al. Dialogpt: Largescale generative pre-training for conversational response generation. arXiv.org: https://arxiv.org/pdf/1911.00536.pdf, (2019).
[3] Lowe RT, Pow N, Serban IV, Charlin L, Liu CW, Pineau J. Training end-to-end dialogue systems with the Ubuntu dialogue corpus. Dialogue&Discourse. 2018. 8(1): 31–65.
[4] Vasiljevs A, Skadina I, Deksne D, Kalis M, Vira I. Application of Virtual Agents for Delivery of Information Services. New Challenges of Economic and Business Development; 2017. p. 667-678.
[5] Balodis K, Deksne D. FastText-Based Intent Detection for Inflected Languages. Information. 2019 10 (5), 161; 2019. p. 1-16.
[6] Paikens P, Znotiņš A, Bārzdiņš G. Human-in-the-loop conversation agent for customer service. Natural Language Processing and Information Systems. NLDB 2020. Lecture Notes in Computer Science. 2020; 12089: 277-284.
[7] Burtsev M, et al. DeepPavlov: Open-Source Library for Dialogue Systems. Proceedings of ACL 2018, System Demonstrations; 2018. p. 122-127.
[8] Roller S, Dinan E, Goyal N, Ju D, Williamson M, Liu Y, et al. Recipes for building an open-domain chatbot. arXiv.org: https://arxiv.org/pdf/2004.13637.pdf, (2020).

[9] Joulin A, Grave E, Bojanowski P, Mikolov T. Tricks for Efficient Text Classification. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers; 2017. P. 427-431.
[10] Grave E, Bojanowski P, Gupta P, Joulin A, Mikolov T. Learning Word Vectors for 157 Languages. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018); 2018, p. 3483-3487.
[11] Williams JD, Asadi K, Zweig G. Hybrid Code Networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. arXiv.org: https://arxiv.org/pdf/1702.03274.pdf (2017).
[12] Znotiņš A. LVBERT: Pre-trained language models for Latvian. In: Proceedings of 9th International Conference Human Language Technologies - Baltic Perspective; 2020.
[13] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT 2019; 2019. p. 4171–4186.
[14] Arkhipov M, Trofimova M, Kuratov Y, Sorokin A. Tuning Multilingual Transformers for Language-Specific Named Entity Recognition. In: Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing; 2019. p. 89-93.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200610

LVBERT: Transformer-Based Model for Latvian Language Understanding

Artūrs ZNOTIŅŠ [1] and Guntis BARZDIŅŠ
Institute of Mathematics and Computer Science, University of Latvia, Latvia

Abstract. This paper presents LVBERT – the first publicly available monolingual language model pre-trained for Latvian. We show that LVBERT improves the state-of-the-art for three Latvian NLP tasks including Part-of-Speech tagging, Named Entity Recognition and Universal Dependency parsing. We release LVBERT to facilitate future research and downstream applications for Latvian NLP.

Keywords. Transformers, BERT, language models, Latvian

1. Introduction

Pre-trained contextualized text representation models, especially BERT – the Bidirectional Encoder Representations from Transformers [1], have become very popular and helped to achieve state-of-the-art performances in multiple Natural Language Processing (NLP) tasks [2]. Previously, the most common text representations were based on word embeddings that aimed to represent words by capturing their distributed syntactic and semantic properties [3], [4]. However, these word embeddings do not incorporate information about the context in which the words appear. This issue was addressed by BERT and other pre-trained language models. The success of BERT and its variants has largely been limited to the English language. For other languages, one could use existing pre-trained multilingual BERT-based models [1], [5] and optionally fine-tune them, or retrain a language-specific model from scratch [6], [7]. The latter approach has been proven to be superior [8].

Our contributions are as follows:

- We present a methodology to pre-train the BERT model on a Latvian corpus.
- We evaluate LVBERT and show its superiority on three NLP tasks: Part-of-Speech (POS) tagging, Named Entity Recognition (NER) and Universal Dependency (UD) parsing.
- We make LVBERT model publicly available[2].

[1]Corresponding Author: Artūrs Znotiņš; Institute of Mathematics and Computer Science, University of Latvia, Riga, Latvia; E-mail: arturs.znotins@lumii.lv.
[2]https://github.com/LUMII-AILab/LVBERT

Table 1. Number of sentences and tokens in the pre-training dataset from each source

Source	#Sentences (M)	#Tokens (M)
Balanced Corpus	0.7	12
Wikipedia	1.3	25
Comments	5	80
News	20	380
Total	27	500

2. LVBERT

2.1. Model

In our experiments, we used the original implementation of BERT on TensorFlow with the whole-word masking and the next sentence prediction objectives. We used $\mathrm{BERT}_{\mathrm{BASE}}$ configuration with 12 layers, 768 hidden units, 12 heads, 128 sequence length, 128 mini-batch size and 32,000 token vocabulary. The model was trained on a single TPUv2 for 10,000,000 steps that took about 10 days.

2.2. Pre-training Dataset

The original BERT was trained on 3.3B tokens extracted from English Wikipedia and the Book Corpus [9]. Latvian Wikipedia dump is relatively small compared to English. To increase lexical diversity, we included texts from the Latvian Balanced corpus LVK2018 [10], Wikipedia, comments and articles from various news portals (see Table 1). Dataset contains 500M tokens.

2.3. Sub-Word Unit Segmentation

Sub-word tokenization is one of the problems of the multilingual BERT model that uses 110k shared sub-word token vocabulary. Because Latvian is under-represented in the training dataset, tokenization into sub-word units is very fragmented, especially for less frequent words. We trained SentencePiece model [11] on the pre-training dataset to produce a vocabulary of 32,000 tokens that was then converted to WordPiece format used by BERT. For sentence tokenization, we used LVTagger [12]. mBERT's sub-words tend to be shorter and less interpretable, for example:

```
mBERT:  So #fi #ja a #ši ie #sl #ē #dz #ās sa #vā m #ā #jā .
LVBERT: Sofija     a #ši ieslēdz #ās       savā   mājā     .
```

3. Evaluation

We evaluated LVBERT on three Latvian NLP tasks: POS, NER, UD. We compared LVBERT model results with the multilingual BERT model (mBERT) results and the current state-of-the-art on each task. We also fine-tuned multilingual BERT model (mBERT-adapted) on our pre-training dataset and evaluated it to assess usefulness of additional target language data. All model results were averaged over three runs.

Table 2. Named entity dataset statistics

	Train	Dev	Test
GPE	1600	218	207
entity	168	18	29
event	214	22	22
location	538	54	84
money	29	3	12
organization	1354	237	251
person	2466	320	306
product	231	31	31
time	967	144	115

3.1. Part-of-Speech Tagging

For POS tagging, we used bidirectional LSTM architecture and compared results to the current state-of-the-art Latvian morphological tagger [13]. We only evaluated the POS tag accuracy ignoring full morphological tag.

3.2. Named Entity Recognition

For training and evaluating NER, we used a recently published multi-layer text corpus for Latvian [14]. Named entity layer includes annotation of nine entity types: person, organization, geopolitical entity (GPE), location, product, event, time (relative or absolute date, time, or duration), money, and unclassified entity. In this work, only the outer level entities are considered, ignoring hierarchical annotation of named entities. We use the same train/development/test data split as for Universal Dependency layer to preserve corpus distribution of genres and to prevent document overlap between splits (see table 2).

We used a standard neural architecture consisting of bidirectional LSTM with a sequential conditional random fields layer above it. IOB2 (Inside, Outside, Beginning) tagging scheme was used to model named entities that span several tokens. The current best Latvian NER model based on GloVe word embeddings [15] was re-evaluated on the same dataset and compared to the BERT-based models.

3.3. Universal Dependency Parsing

For dependency parsing, we used a model based on biaffine classifiers on top of a bidirectional LSTM [16], specifically, AllenNLP[3] implementation, and compared results to the current state-of-the-art [15] re-evaluated on the latest Universal Dependency release.

4. Results

The main results of our experiments are presented in Table 3. LVBERT models achieved the best results in all three tasks. The most significant improvement was visible in the UD task. Multiligual BERT model under-performed in the NER and POS tagging tasks

[3]https://allennlp.org/

Table 3. Performance of LVBERT on Latvian NLP tasks compared to multilingual BERT and previous state-of-the-art

Task	Metric	Previous Best	mBERT	mBERT-adapted	LVBERT
POS	Accuracy	97.9	96.6	98.0	98.1
NER	F1-score	78.4	79.2	81.9	82.6
UD	LAS	80.6	85.7	88.1	89.9

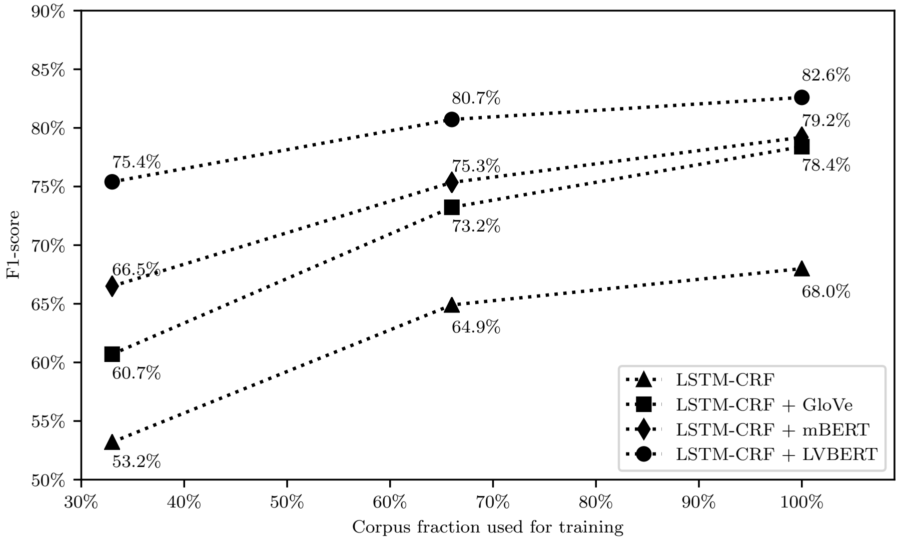


Figure 1. NER learning curve

compared to relatively simple word embedding based models showcasing its shortcomings for less resourced languages. To fully utilize the potential of BERT, the multilingual model should be at least fine-tuned on the specific language texts. Fine-tuned BERT model performed surprisingly well when compared to LVBERT given its vocabulary disadvantage. NER learning curve (see Figure 1) shows that pre-trained contextualized text representation models can more fully utilize limited amount of training data compared to simple word embeddings, especially if just a relatively small part of the annotated training corpus is used.

5. Conclusion

This paper showcases that even a relatively small language specific BERT model can significantly improve results over non-contextual representations and also multilingual BERT model. LVBERT sets a new state-of-the-art for several Latvian NLP tasks. By publicly releasing LVBERT model, we hope that it will serve as a new baseline for these tasks and that it will facilitate future research and downstream applications for Latvian NLP.

Acknowledgements

This research is funded by the Latvian Council of Science, project "Latvian Language Understanding and Generation in Human-Computer Interaction", project No. lzp-2018/2-0216.

References

[1] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics; 2019. p. 4171–4186. Available from: https://www.aclweb.org/anthology/N19-1423.

[2] Howard J, Ruder S. Universal language model fine-tuning for text classification. arXiv preprint arXiv:180106146. 2018.

[3] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems; 2013. p. 3111–3119.

[4] Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP); 2014. p. 1532–1543.

[5] Conneau A, Khandelwal K, Goyal N, Chaudhary V, Wenzek G, Guzmán F, et al. Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:191102116. 2019.

[6] de Vries W, van Cranenburgh A, Bisazza A, Caselli T, van Noord G, Nissim M. BERTje: A Dutch BERT Model. arXiv preprint arXiv:191209582. 2019.

[7] Martin L, Muller B, Suárez PJO, Dupont Y, Romary L, de la Clergerie ÉV, et al. CamemBERT: a Tasty French Language Model. arXiv preprint arXiv:191103894. 2019.

[8] Virtanen A, Kanerva J, Ilo R, Luoma J, Luotolahti J, Salakoski T, et al. Multilingual is not enough: BERT for Finnish. arXiv preprint arXiv:191207076. 2019.

[9] Zhu Y, Kiros R, Zemel R, Salakhutdinov R, Urtasun R, Torralba A, et al. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In: Proceedings of the IEEE international conference on computer vision; 2015. p. 19–27.

[10] Levane-Petrova K. Līdzsvarotais mūsdienu latviešu valodas tekstu korpuss, tā nozīme gramatikas pētījumos. Language: Meaning and Form. 2019;10:131–146. The Balanced Corpus of Modern Latvian, its role in grammar studies. Available from: https://www.apgads.lu.lv/fileadmin/user_upload/lu_portal/apgads/PDF/Valoda-nozime-forma/VNF-10/vnf_10-12_Levane_Petrova.pdf.

[11] Kudo T, Richardson J. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. arXiv preprint arXiv:180806226. 2018.

[12] Paikens P, Rituma L, Pretkalniņa L. Morphological analysis with limited resources: Latvian example. In: Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013); 2013. p. 267–277.

[13] Paikens P. Deep Neural Learning Approaches for Latvian Morphological Tagging. In: Baltic HLT; 2016. p. 160–166.

[14] Gruzitis N, Pretkalnina L, Saulite B, Rituma L, Nespore-Berzkalne G, Znotins A, et al. Creation of a Balanced State-of-the-Art Multilayer Corpus for NLU. In: Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC); 2018. p. 4506–4513. Available from: http://www.lrec-conf.org/proceedings/lrec2018/pdf/935.pdf.

[15] Znotins A, Cirule E. NLP-PIPE: Latvian NLP Tool Pipeline. In: Human Language Technologies - The Baltic Perspective. vol. 307. IOS Press; 2018. p. 183–189. Available from: http://ebooks.iospress.nl/volumearticle/50320.

[16] Dozat T, Manning CD. Deep biaffine attention for neural dependency parsing. arXiv preprint arXiv:161101734. 2016.

Tools and Resources

This page intentionally left blank

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200612

An Online Linguistic Analyser for Scottish Gaelic

Loïc BOIZOU [a] and William LAMB [b,1]
[a] *Centre of Computational Linguistics, Vytautas Magnus University, Lithuania*
[b] *Celtic and Scottish Studies, University of Edinburgh, Scotland, UK*

Abstract. This paper describes the Gaelic Linguistic Analyser, a new resource for the Scottish Gaelic language. The GLA includes a tagger, a lemmatiser and a parser, which were developed largely on the basis of existing resources. This tool is available online as the first component of the Scottish Gaelic Toolkit.

Keywords. Scottish Gaelic language, lemmatiser, tagger, parser, web portal, language technology

1. Introduction

This paper presents a new resource for the under-resourced Celtic language, Scottish Gaelic (henceforeth, Gaelic). Although only spoken by about 1 % of the Scottish population, there have been efforts to revitalise Gaelic recently, in domains such as culture, media, education and government. Hereditary Gaelic communities are endangered [1], but the language's profile has been boosted of late by its inclusion in a major television program (Outlander) and within certain language technology platforms (e.g. Duolingo and Google Translate). As native domains contract, it has become crucial for the language's long-term survival to develop data and cornerstone tools to support its use in technologically mediated ones.

In the field of Natural Language Processing, Gaelic is less advanced than two related languages, Irish [2,3] and Welsh [2]. However, some key resources are available, such as oral archives [3], online dictionaries [4], and corpora such as DASG [5], ARCOSG [6] and the UD Gaelic Treebank. The Gaelic Linguistic Analyser (GLA) – which combines a tagger, a lemmatiser, and a parser – strongly relies on these resources. Sharing existing resources widely and building new tools on top of them is of crucial importance for lesser-resourced languages, to minimise reduplication and fully exploit work already done.

[1]Corresponding Author: William Lamb; University of Edinburgh, 50 George Square, Scotland, UK, EH8 9LH; E-mail: w.lamb@ed.ac.uk.
[2]http://techiaith.cymru/cloud/api/parts-of-speech-tagger-api/?lang=en
[3]www.tobarandualchais.co.uk/en/
[4]www.faclair.com
[5]sg.ac.uk
[6]www.github.com/Gaelic-Algorithmic-Research-Group/ARCOSG

2. The Part-of-Speech Tagger

Lamb and Danso [4] developed the first tagger for Gaelic using a small part of ARCOSG, later extending it with the full corpus [5]. The present analyser uses the whole of ARCOSG for training and evaluation, in addition to further training files (105,456 tokens in total). The tagger was developed in Python3 using the ML scikit-learn library. The model was trained on 96.6 % of ARCOSG. One sentence in 20 was randomly picked for evaluation to ensure that all of the genres present in ARCOSG appear in the evaluation set.

For each word-form, the following features were included in the conditional random field model: 1) the original word-form and the lowercase one; 2) the prefix and suffix up to three letters; 3) information about symbols used in the word-form (e.g. capitals, numbers, hyphens, non-Gaelic letters); 4) the position in the sentence (initial, final, intermediate); and 5) the two previous and following word-forms in the sentence.

The accuracy is currently 0.907 (cf. 0.84 in [5]). Following the experience of [5], we retrained the tagger on the same corpus, but with a restricted set of tags (41 tags versus 246 in the full tagset), and achieved an accuracy of 0.947 (cf. 0.92 in [5]). The simplified tagger avoids the features that are difficult to grasp from the context in Gaelic, such as gender and case, for users who require less morphological granularity and desire a higher accuracy.

Except for the size of the training corpus, tagging mistakes come from different sources. As specific issues for Gaelic, we should mention the high frequency of English words mixed in Gaelic sentences (cf. [6]). We should also point inconsistencies in Gaelic orthography in the training data (cf. [7]). Among technical issues, a discrepancy exists between the segmentation used by ARCOSG, with many multiword tokens, and the general tokeniser used for segmenting input text, which is based mainly on orthographic words. We intend to address this issue in future works.

3. The Lemmatiser

The GLA lemmatiser is the first publicly presented for Gaelic. During the development stage, we instantiated two versions: a rule-based one and a lexicon-based one. Evaluation work is ongoing, but the lexicon-based one has two benefits at present: 1) the mistakes seem more acceptable, since it avoids impossible word forms and 2) the lexicon (courtesy of Michael Bauer and Will Robertson of www.faclair.com) was almost ready-to-use; it contains about 177,000 word-forms associated with their lemmas and parts of speech.

The main challenge was to locate an efficient searching algorithm. We began with a standard dictionary (or 'letter') tree – an alphabetically sorted binary tree – where node values are letters and lemmas at the current position in the tree. Nonetheless, the time for loading the tree was unacceptably high because of its recursive structure. It was possible to correct the problem with suspending the garbage collector, but this approach was suboptimal. Therefore, we decided to apply the dictionary tree principle to a standard Python dictionary, using letters as sorted key, as in the following example:

```
{ 'a': { 'b':  { 'a': {...} },
                 ...
                 'lemmas': [('aba', 'N')]
```

```
                },
            'c':  { 'a': {...} },
                    ...
                },
            ...
            'lemmas': [('a', 'P')]
        },
    'à': {...},
    'b': {...},
    ...
}
```

In the dictionary, letters at the same depth are considered alternatives, e.g. the first letter of a word, which appears in the first column of letters, can be *a*, *à*, *b*, and so on. Then the value of the key is to be understood as the group of possible letters at the next position (here, the second position), e.g. an initial *a* can be followed by *b* or *c* (for *ab...* or *ac...*), then *a* and *b* can be followed by *a*. The resulting string (*aba*) is a word-form that corresponds to the (unchanged) noun lemma *aba*, which means ‘abbot’. The one-letter string *a* is also a word-form: it corresponds to the third person possessive pronoun (lemma is identical *a*). As shown in the example, lemmas are grouped under the dictionary key ‘lemmas’. The value is a list, since a word-form can be related to several lemmas of one or several parts of speech. In the current version of the lemmatiser, we return the first lemma matching the requested word-form with its assigned part of speech; this provides simple results with one lemma for each analysed word. In the future, we may return all possible lemmas or start disambiguating lemmas via context.

The letter dictionary is about 50 % slower than the dictionary tree in the worst cases, when searched items are at the end of the alphabet, but loading the structure is faster and does not require suspending the garbage collector.

The letter dictionary is the core of the lemmatisation process. Searching involves the word-form and the part of speech previously indicated by the tagger. It is completed by a lower-case search: if the original word-form is not found in the dictionary, a second attempt is performed with the lowercase word-form. Fused words, such as prepositional pronouns, are treated somewhat differently, in that a lemma is provided for each fused element, e.g., *annad* ‘in you’ is lemmatised as *ann* ('in') + *thu* ('you').

4. The Parser

The GLA parser is based on the ready-to-use Python UDPipe library [8]. The syntactic model was trained using UDpipe executable on the Gaelic UD treebank made by Colin Bachelor [9]. The parser accuracy was evaluated with the same tool while selecting different transition systems: with the link2 parser option – UAS: 97.11 %, LAS: 96.40 %, with the swap option – UAS: 97.10 %, LAS: 96.35 %, with the projective option – UAS: 92.95 %, LAS: 91.33 %. The link2 model, which gave the best accuracy, is the one used by the parser.

These evaluations are meant to be compared together and not to other tools or data sets, since they are measured on the training set. This non-standard decision was motivated by the choice to keep all the limited data for training, because of their relative scarcity. We are aware that it will be necessary in the future to set aside control data for a proper evaluation.

5. The Web Portal

The GLA is the first component of the Scottish Gaelic Toolkit (SGT), which is accessible at the following adress: [7]. The website, which is fully bilingual in Gaelic and English, is based on a Python server solution that relies on Flask[8] and Gunicorn [9]. It provides access to the GLA through a text area window, where Gaelic sentences can be written or pasted, or through a web service with a POST request.

6. Final Remarks

Our objective with the GLA was to provide useful NLP tools online and demonstrate that these tools can run efficiently, if basic resources such as lexicons and annotated corpora are shared. A more comprehensive evaluation of the different tools, especially the lemmatizer and the parser is a future desideratum.

To increase functionality, soon we hope to provide an option to manually correct analyses. This should increase available training data via crowd-sourcing, although the pipeline required for integrating the extra data requires additional work.

References

[1] Ó Giollagáin C., Camshron, G., Moireach, P., Ó Curnáin, B., Caimbeul, I., MacDonald, B. and Péterváry, T. The Gaelic Crisis in the Vernacular Community. 2020. Aberdeen: Aberdeen University Press.

[2] Lynn T, Scannell K, Maguire E. Minority Language Twitter: Part-of-Speech Tagging and Analysis of Irish Tweets. In: Xu W, Han B, Ritter A, editors. Proc. of the First Workshop on Noisy User-generated Text; 2015 July 31, Beijing, China. European Association for Machine Translation; c2015. p 1-8

[3] Lynn T, Scannell K. Code-switching in Irish Tweets: A Preliminary Analysis. In: Lynn T, Prys D, Batchelor C, Tyers F, editors. Proceedings of the Celtic Language Technology Workshop; 2019 August 19, Dublin, Ireland. European Association for Machine Translation; c2019. p. 32-9.

[4] Lamb W, Danso S. Developing an Automatic Part-of-Speech Tagger for Scottish Gaelic. In: Judge J, Teresa Lynn T, Ward M, Ó Raghallaigh B, editors. Proceedings of the First Celtic Language Technology Workshop; 2014 August 23, Dublin, Ireland. Association for Computational Linguistics and Dublin City University; c2014. p. 1-6.

[5] Lamb W, Danso S, Lawson A. Evaluating a Gaelic Part-of-Speech Tagger and Reference Corpus [Internet]. 2016. Available from: https://www.academia.edu/26589071/Evaluating_a_Gaelic_Part-of-Speech_Tagger_and_Reference_Corpus.

[6] Smith-Christmas C. Stance and Code-Switching: Gaelic-English Bilinguals on the Isles of Skye and Harris, Scotland. In: Auer P, Caro Reina J, Kaufmann G, editors. Language Variation – European Perspectives IV. 2011 June; Freiburg. Amsterdam: John Benjamins; c2013. 229–17

[7] Ross S. The standardisation of Scottish Gaelic orthography 1750-2007: a corpus approach [PhD thesis]. [Glasgow]: University of Glasgow; 2016. 265 p.

[8] Straka M, Straková J. Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In: Hajič J, Zeman D, editors. Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies; 2017 August, Vancouver, Canada. Association for Computational Linguistics; c2017. p 88-12.

[9] Batchelor C. Universal dependencies for Scottish Gaelic: syntax. In: Lynn T, Prys D, Batchelor C, Tyers F, editors. Proceedings of the Celtic Language Technology Workshop; 2019 August 19, Dublin, Ireland. European Association for Machine Translation; c2019. p. 7-9.

[7] https://klc.vdu.lt/sgtoolkit/

[8] https://flask.palletsprojects.com

[9] https://gunicorn.org/

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200613

Corpus-Based Methods for Assessment of Traditional Dictionaries

Virginijus DADURKEVIČIUS[a,1] and Rūta PETRAUSKAITĖ[b]
[a] *Vilnius University*
[b] *Vytautas Magnus University*

Abstract. The paper presents the investigation of *The Dictionary of Modern Lithuanian* (6th edition) from the point of view of its coverage in comparison with a Joint Corpus of Lithuanian. Resources, methods and procedures are described together with the results revealing that only 81 % of the dictionary lemmas have their counterparts in the corpus.

Keywords. Corpus-based lexicography, updates of traditional dictionaries, Hunspell platform, comparison of dictionaries and corpora

1. Introduction

From its very start, corpus linguistics was used for different purposes of lexicography. At first, raw corpora served as sources of authentic data, then annotated corpora provided different patterns of usage, and finally, lists of entry headwords for newly compiled dictionaries were derived from corpus-based frequency lists. There were other numerous applications of corpora and corpus-based methods of language description, however, they were applied for the compilation of new dictionaries and not for updating the old ones. Nevertheless, traditional dictionaries can be updated and made more efficient with the help of corpora and computational linguistics. This paper presents methods and procedures exploiting corpora for the update of traditional dictionaries, specifically, the list of their entry words. A case study of *The Dictionary of Modern Lithuanian* (6th edition, hence, the DML6) [1] and the Joint Corpus of Lithuanian (hence, JCL) serve as an example.

2. Resources and Procedures

JCL is a merge of three corpora (see Table 1 below): Vilnius university corpus (VU) representing the Lithuanian internet content from 2014 and primarily used for machine translation, a legal document corpus in a form of wordlist (courtesy of the Office of the Seimas of the Republic of Lithuania, 2011, hence, LRSK) and a balanced corpus of present-day Lithuanian of Vytautas Magnus University (VMU). The terms "*tokens*" (all

[1] Corresponding Author: Virginijus Dadurkevičius; Vilnius University Faculty of Physics Institute of Photonics and Nanotechnology, M. K. Čiurlionio str. 29, 03100 Vilnius, Lithuania; E-mails: virginijus.dadurkevicius@tmi.vu.lt, dadurka@gmail.com.

words, including repeated), "*types*" (only distinct words) and "*Type to Token Ratio*" (TTR) are used while discussing corpora, comparing them, assessing their size, scope and representativeness [2]. Usually, TTR is expressed as percentage and tends to decrease as the corpus gets larger. With reference to these terms, the overall size of JCL is 1,334,845,080 tokens, 4,968,125 types, and 0.37 % TTR. The size of JCL is approximately equal to 10,000 books, i.e. the number of books published in Lithuanian in three years.

Table 1. Composition of JCL

Specific corpus	Tokens	Types	TTR	Contribution to JCL
VU	779,154,268	3,958,963	0.51 %	58.4 %
LRSK	443,114,936	1,092,473	0.23 %	33.2 %
VMU	112,575,876	1,778,259	1.58 %	8.4 %

DML6 [1] contains ~600,000 entries with ~86,000 lemmas. The difference in numbers can be explained by the fact that only part of naturally existing lemmas is presented as entry headwords, others are explicitly mentioned in the entries while some of them are not mentioned at all. The latter are called implicit lemmas based on regular word formation patterns. In the Introduction to the dictionary, they are described as belonging to the regular derivational patterns therefore assumed "by default". Thus, the entry with the headword "gailėti" contains 13 lemmas:

a) explicit lemmas of

1. the verb "gailėti" from the derivational paradigm "gaili, gailėjo";
2. the verb "gailėti" from the derivational paradigm "gailėja, gailėjo";
3. the noun "gailėjimas";
4. the noun "gailėjimasis";
5. the verb "gailėtis" derivational paradigm "gailisi, gailėjosi" (a hint of the existence of such a reflexive form is given in the entry "|| sngr.");

b) implicit lemmas of

1. the prefixed derivative verbs "negailėti", "tegailėti", "nebegailėti", "tebegailėti" (regular derivational pattern of the above form is discussed in the Introduction of the dictionary, hence, these forms are not presented in the respective entries);
2. reflexive forms of the above prefixed verbs "nesigailėti", "tesigailėti", "nebesigailėti", "tebesigailėti".

Lithuanian is a synthetic language rich of flexions. First, for the comparison of the dictionary with the corpus, all inflected forms which could be theoretically derived from dictionary lemmas and morphological information provided there had to be generated. As a tool for this task, the Hunspell platform [3] has been chosen. The primary goal of this platform is spelling, but after substantial modification [4], it can also be successfully applied to morphological analysis and synthesis. Successful application of Hunspell platform for Lithuanian was described by Dadurkevičius [5]. Using Hunspell formalism, the scope of a particular language is represented in two files: affixes (morphological rules) and dictionary (words with references to its rules). In our case, the Hunspell dictionary was built by obtaining all the possible lemmas from DML6 entries (both explicitly stated and implied). That made about 200,000 entries in total. The file of morphology rules is used to generate all the theoretically possible word forms. In our case, these rules (about 5,000 items) were based on the *Grammar of Modern Lithuanian* [6]; they are described in detail by Dadurkevičius [5]. References from the Hunspell dictionary to the rules were derived on the basis of information provided in DML6 entries. More than 50 million word forms of DML6 can be generated combining a Hunspell dictionary and its rules. This is how the tool is made suitable for both spelling and morphological analysis based on DML6.

Assessing the coverage of DML6 of the contemporary Lithuanian language represented by JCL, two research questions were asked:

1. What part of JCL is covered by DML6? As a measure for such assessment, the percentage of JCL tokens overlapping with grammatical word forms of DML6 (50+ millions of possible word forms) was calculated. Looking at this facet of the assessment, 100 % overlap would mean a perfect dictionary, able to identify every single word of a corpus. To simplify and speed up the calculation processes, we used the spelling feature of the Hunspell platform to find out if the token in JCL has the matching word form in DML6. A correctly spelled token means that it can be derived from DML6 content. An incorrectly spelled token means a failure to find the match in DML6 and would mark a possible lexical gap in the dictionary. The list of possible gaps [7] could be a valuable resource for updating DML6.

2. How up to date the full list of headwords and other explicit entry lemmas of DML6 really is? A measure for such assessment is the percentage of DML6 explicit lemmas having counterparts (any form, at least one occurrence) in JCL. 100 % would mean a perfect dictionary, with every single headword being used in the corpus that covers a major part of the present-day Lithuanian language. To make this estimation, the list of JCL types has been lemmatized using the functionality of Hunspell platform; implicit lemmas have been ignored. The number of DML6 lemmas having counterparts in the corpus has been compared to the total number of lemmas in DML6. Failure to find DML6 lemma in JCL would mark presently unused words. The fact of such a failure cannot be sufficient to state that headwords, absent in JCL, are out of use nowadays. Nevertheless, the list of unused headwords [7] should be tested applying other methods, e.g. linguistic experiment or introspection.

3. Results

In reply to the first research question concerning lexical gaps and the coverage of DML6, the results, provided below, were obtained. DML6 based Hunspell spell-checker accepted 1,191,815,754 tokens (89.3 %) and 1,252,370 (25.2 %) types of JCL. See Tables 2 and 3 for the distribution of the results in the constituent parts of JCL.

Table 2. Corpora tokens covered by DML6

Corpora	Number of tokens covered by DML6	Total number of tokens in the corpora	%
VU	694,405,495	779,154,268	89.1
LRSK	393,344,588	443,114,936	88.8
VMU	104,065,671	112,575,876	92.4
JCL	1,191,815,754	1,334,845,080	**89.3**

Table 3. Corpora types covered by DML6

Corpora	Number of types covered by DML6	Total number of types in the corpora	%
VU	1,081,818	3,958,963	27.3
LRSK	426,958	1,092,473	39.1
VMU	789,982	1,778,259	44.4
JCL	1,252,370	4,968,125	**25.2**

The reply to the second research question concerning unused lemmas in DML6 provides information about the lemmatization of the corpus that allows to identify 81.1 % of DML6 lemmas. Thus, about one fifth of DML6 lemmas can be regarded as presently unused lexis. See Table 4 for a detailed part of speech analyses of the overlapping lemmas in the compared resources.

Table 4. Number of overlapping lemmas and their POS features in the compared resources

Part of speech	Number of explicit lemmas in DML6	Number of explicit lemmas present in JCL	Number of explicit lemmas absent in JCL	% of the DML6 lemmas having their counterparts in JCL
Adjective	7,398	6,885	513	93.1
Adverb	3,063	2,591	472	84.6
Noun	49,801	37,503	12,298	75.3
Numeral	85	82	3	96.5
Proper noun	2,717	2,706	11	99.6
Pronoun	59	59	0	100.0
Verb	22,020	19,161	2,859	87.0
Other	927	826	101	89.1
TOTAL	86,070	69,813	16,257	**81.1**

A detailed qualitative analysis of the lexical gaps of DML6 as well as its unused dictionary lemmas is planned as the next stage of this research hoping that it should help lexicographers to update the dictionary.

References

[1] The Dictionary of Modern Lithuanian. Edited by Keinys S. 6th (3 electronic) edition of the Dabartinės lietuvių kalbos žodynas. 2006.
[2] Scott, M. WordSmith Tools version 8, Stroud: Lexical Analysis Software, 2020.
[3] Hunspell platform https://hunspell.github.io
[4] Németh L, Trón V, Halácsy P, Kornai A, Rung A, Szakadát I. Leveraging the Open Source Ispell Codebase for Minority Language Analysis. SALTMIL Workshop at LREC 2004: First Steps in Language Documentation for Minority Languages. Computational Linguistic Tools for Morphology, Lexicon and Corpus Compilation. Edited by Julie Carson-Berndsen, 2004:56-59.
[5] Dadurkevičius V. Lietuvių kalbos morfologija atvirojo kodo "Hunspell" platformoje [Lithuanian Morphology in the "Hunspell" Framework]. Bendrinė kalba**.** 2017:1-15.
[6] Lithuanian Grammar. Edited by Ambrasas V. (in English). 1997.
[7] Dadurkevičius V. Assessment Data of the Dictionary of Modern Lithuanian versus Joint Corpora, CLARIN-LT digital library in the Republic of Lithuania, http://hdl.handle.net/20.500.11821/36. 2020.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200614

Lessons Learned from Creating a Balanced Corpus from Online Data

Roberts DARĢIS [1], Kristīne LEVĀNE-PETROVA, and Ilmārs POIKĀNS
Institute of Mathematics and Computer Science, University of Latvia, Latvia

Abstract. This paper describes lessons learned from developing the most recent Balanced Corpus of Modern Latvian (LVK2018) from various online sources. Most of the new corpora are created from data obtained from various text holders, which requires cooperation agreements with each of the text holders. Reaching these cooperation agreements is a difficult and time consuming task and may not be necessary if the resource to be created is not of hundred millions of size. Although there are many different resources available on the Internet today for a particular language, finding viable online resources to create a balanced corpus is still a challenging task. Developing a balanced corpus from various online sources does not require agreements with text holders, but it presents many more technical challenges, including text extraction, cleaning and validation.

Keywords. Balanced corpus, general corpus, corpus development, metadata

1. Introduction

Nowadays, the research of different scientific disciplines would not be possible without the use of corpora, especially a reference corpus, that is designed to provide comprehensive information about a language [1].

A corpus is used in linguistics to conduct language research, create dictionaries and grammars; in sociology to analyze mass opinion and behavior and in computer science to develop natural language processing components, such as machine translation, speech recognition and various text taggers.

Most of the new corpora are created from data obtained from various text holders that makes corpus creation much more easier, because the texts are of high quality, in easily parsable formats with structured metadata [2], [3]. Obtaining cooperation agreements from different text holders is a difficult and time consuming task and may not be necessary if the resource to be created is not of hundred millions or even billions of size.

This paper describes the development of the latest corpus in the LVK series. The Balanced Corpus of Modern Latvian (LVK2018) [4] is a new 10 million representative corpus of contemporary Latvian, created mostly from various online sources. Although there are many different resources available on the Internet today for a particular language, finding viable online resources to create a balanced corpus is still a challenging

[1]Corresponding Author: Roberts Darģis, Artificial Intelligence Laboratory, Institute of Mathematics and Computer Science, University of Latvia, Raiņa bulv. 29, Riga, LV-1459, Latvia; E-mail: roberts.dargis@lumii.lv.

task. Extracting text and metadata is a much more complicated task, and much more effort is required to clean and validate the data. This paper describes lessons learned from developing the most recent balanced corpus of LVK series from various online sources, which do not require agreements with text holders, but it presents many more technical challenges.

2. Background of LVK Series

The Balanced Corpus of Modern Latvian (LVK) has been developed in multiple rounds. The history of the LVK series goes back to 2007 when the first 1 million corpus was created. The experience from the designing of other general corpora was taken into account as well. The reviewed list of corpora includes British National Corpus [5], [6], Czech National Corpus [3], [7], [8], Corpus of the Contemporary Lithuanian Language [9], [10], and others. The same corpus design criteria were also used for the subsequent LVK series. The previous corpus from this series (LVK2013) was released on 2013 with 4.5 million words [11]. All corpora are morphologically annotated [12], [13], [14] and with metadata descriptions.

Previous corpora in LVK series were manually created. The main innovation in LVK2018 is autmatization process in all corpus development steps.

3. Design Principles of LVK

LVK2018 is designed as a general-language, representative and publicly available corpus. It is a monolingual, fully morphologically and partly syntactically and semantically annotated corpus. Presently, it consists of 10 million tokens. Characteristics of LVK2018:

- **General** – the corpus includes sources from different domains, styles, genres, etc.
- **Balanced** – the corpus that aims to cover the variety of existing texts in estimated proportions.
- The corpus represents the **synchronic** state of the language. It covers sources as from the end of the last century until the present.
- **Originality** – the corpus should only contain texts originally written in Latvian. The obvious translations of the different texts into Latvian will not be included in LVK2018.
- The corpus is **representative**, it contains texts from all language styles, major domains and many subdomains.

The corpus contains five different sections – *journalism, fiction, scientific, legal and parliamentary transcripts* (figure 1).

To cover different magazines and newspapers, subsequently the Journalism section also has been divided into the following subsections: *nationwide media, regional media, leisure media, and popular science media* (figure 2). The previously defined corpus sections were not enough to achieve fully balanced and representative corpus, so multiple additional text selection criteria were set.

- **Time** – the corpus should contain texts created and published after 1991.

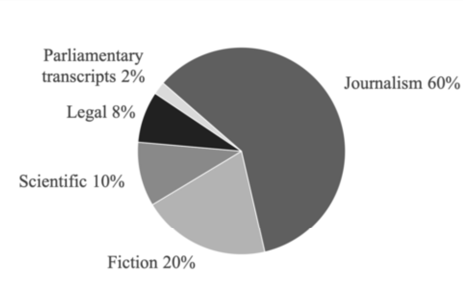


Figure 1. Composition of LVK2018

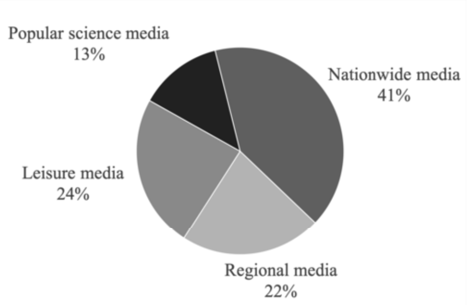


Figure 2. Composition of Journalism section

- The corpus should contain **full-text**.
- **Diversity** – texts should cover as wide range of topics as possible. The sample cannot exceed more than 5 % and 50,000 words of the particular section of the corpus to not dominate one author or domain in the corpus. If the text is longer than the limit, it should be cut from the end of the sample only.
- **Uniqueness** – the corpus sample should be represented in corpus just once.
- **Quality** – samples should only contain clean text written in literary language with appropriate usage of diacritics and punctuation in Latvian. Tables and other non-text parts should be removed.

4. Corpus Development Process

All the criteria set previously were implemented in the corpus development process which was divided in three steps.

- **Data collection** – for each section of the corpus, the most suitable data sources were identified, taking into consideration time and originality selection criteria. Text with all of the available metadata were extracted from each of the data sources.
- **Data processing** – data quality and uniqueness assurance processes were used in data processing. Texts were filtered through multi-level system to ensure that only unique and qualitative texts are considered as candidates for final data selection.
- **Final data selection** – a subset was selected from the remaining texts to be included in the corpus, taking into account corpus size limitation and diversity criteria.

Each corpus section (journalism, fiction, scientific, legal, and parliamentary transcripts) required multiple exceptions in different corpus development steps due to the diverse nature of data sources and structure. Next sections of the article will go into more details on how each step was implemented.

5. Data Collection

The most challenging tasks in the whole corpus development process were finding the most suitable freely available online data sources for each of the corpus sections and gathering the data in a structured format with metadata. Texts that are available online are considered to be freely available for the corpus creation purposes. Although the texts themselves are subject to copyright, the concordances that are available through the corpus query interface and used for educational or research purposes do not infringe copyrights [2].

Data for the Journalism section was gathered from various online media sites through a media monitoring system. The data also included forum posts and articles' comments that needed to be removed. The only available metadata for each article were the source URL and the date of publication. From the source URLs, a list of domains were obtained and manually categorized into the one of the four subcategories according to the most common content type.

Fiction is the only section with almost no freely available online data sources. Some sites publish freely available works of new authors and hobby writers. Book samples are also perfect for corpus creation. Unfortunately, there was no enough easily available data for Latvian, so the data was taken from books digitized in previous projects. The data also included metadata (title, author, publisher and year).

Data for the scientific section was collected from online freely available doctoral theses in PDF format. Thematic domain and publishing year were extracted as metadata.

Legal documents were crawled from the database of legal acts of the Republic of Latvia. They contain different types of documents (such as law, regulations, protocols, rulings and orders) from multiple institutions.

Parliamentary transcripts are extracted from The Corpus of the Saeima, which consists of the transcription of Latvian parliamentary debates crawled from the official website [15].

6. Data Processing

Gathering data from online sources can introduce many errors. A quality filter must be put in place to ensure only literary correct texts are included in the corpus. A multi-level system was used to filter out all the texts that did not meet the quality standards. Each of the quality filters were implemented due to a specific problem.

Some documents were written in poor language without the correct use of diacritical signs. Some documents besides the Latvian version also contained versions in a different language. A dictionary approach was used to filter out these kinds of documents. The document was considered to be of poor quality if fewer than 85 % of the words were found in a dictionary. This percentage was empirically calculated from LVK2013, which was manually created and validated.

Some documents did not contain correct Latvian: insufficient use of punctuation, too many punctuation marks or garbage symbols due to some parsing errors or embedded content with many hashtags. Documents having an unreasonable percentage of punctuation marks calculated against the number of tokens were filtered. Reasonable proportion

[2] Copyright Law, 48/150 (2059/2061), 27.04.2000. https://likumi.lv/ta/en/en/id/5138-copyright-law

was considered to be between 12 % and 36 %. These percentage were also empirically calculated from LVK2013.

In rare cases, the conversion from PDF corrupted some letters with diacritics. Documents of a reasonable length are expected to contain every letter of the alphabet. Documents that did not meet the requirement were filtered out.

After the quality filters, the next step was uniqueness filters. In the media industry, it is a common practice to republish the same in multiple sites with little or no modifications at all. Uniqueness issues are also common in the parliamentary section due to standard phrases and in the legal section due to templates, especially in rulings. To filter out too similar articles Bray-Curtis similarity over bag of words models [16] was used. If the similarity between any two documents was greater than 0.8, only the longest document was kept. This threshold was also calculated from LVK2013. Vocabulary in legal documents tends to be much more limited than in journalism, so the threshold in legal documents was set to 0.65 instead of 0.80.

7. Data Selection

The last step of the corpus development process is the final document selection, taking into account the corpus size limitation and diversity criteria. The main challenge is finding the right balance between diverse and representative subsets. Each section of the corpus was balanced according to the available metadata, such as date, author, industry, and others. The way how each section of the corpus was balanced was different due to different metadata properties.

Documents from journalism section were chosen based on the date of publication. To keep the original balance between article categories (local news, global news, sports, finance, etc.), articles were grouped by date of publication and the whole day was included or excluded from the corpus. To obtain the most diversified subset, documents were chosen evenly across the available timespan.

To choose the most representative subset of fiction, documents were chosen so that each author is included in the corpus. If some author had less data available, than the remaining quota was evenly distributed to other authors.

Subset of documents for scientific section was chosen the same way as the subset for fiction section were chosen only instead of choosing by author the documents were chosen based on scientific discipline of the document. In total, there were 30 scientific disciplines.

Data for legal section was already as balanced as it could be because the threshold for the longest common word string was chosen in such a way that the remaining amount of data was only a bit bigger than the required amount for the corpus. In final selection, a few random documents were removed to obtain the target word count.

In final data selection for parliamentary section, documents were grouped by date to cover as wide time span as possible. To achieve higher diversity in the selected subset, iteratively the shortest document from each date was selected until the total word count reached the goal.

8. Annotation of LVK2018

LVK2018 has three publicly visible metadata fields – unique identifier (id), section and reference. A different reference template was designed for each of the five sections to incorporate all the relevant metadata fields for that sections.

- for legal texts – *{title}*, adoption *{date of adoption}*, published on *{date of publication}*;
- for Parliamentary transcripts:
  - for the samples of LVK2013 – from Parliamentary transcripts *{date}*, *{speaker}* (*{parliamentary group}*);
  - for the samples of LVK2018 – from Parliamentary transcripts theme: "*{theme}*", *{sample URL}*;
- for science – *{author}*, *{title}* (*{the branch of science}*), *{year}*;
- for fiction – *{author}*, *{title}* (*{chapter}*). *{publishing place}*, *{publisher}*, *{year}*;
- for journalism – *{naming}*, *{source}*, *{published}*, *{subsection}*.

LVK2018 contains morphosyntactic annotation by the IMCS morphological tagger [12][13][14]. Morphosyntactic annotations contain PoS tag, lemma and other Latvian specific morphological and syntactic information.

A balanced subcorpus of LVK2018 (10,000 sentences), containing samples of texts from the different styles, domains and subdomains existent in the corpus, is also syntactically manually annotated [17], using hybrid dependency-constituency grammar formalism developed in the previous Latvian Treebank pilot project [18]. Afterwards, the hybrid annotation is automatically converted to Universal Dependencies to achieve the cross-lingual compatibility, as well as to provide training data for efficient and robust parsers [19]. The same subset of 10,000 sentences is also manually named-entity, coreference and FrameNet annotated [20].

9. Availability

LVK2018 has been released in the framework of Latvian National Corpus. LVK2018 is freely available via the corpus query interface NoSketch Engine [21].

10. How to quote LVK

The corpus material is to be quoted in the bibliography in the following way: The Balanced Corpus of Modern Latvian – LVK2018. The Institute of Mathematics and Computer Science, University of Latvia. Riga, 2018. Available at: www.korpuss.lv

11. Conclusions

LVK2018 was developed in the FullStack project framework and served as a basis for multilayered syntactically and semantically annotated text corpus for Latvian [19]. Creating LVK2018 from online data allowed successfully complete the project due to time constraints, because the online approach is much faster for developing a corpus of this size compared to the typical approach via signing the cooperation agreements with text holders.

Although nowadays a 10 million balanced corpus is not considered as a large corpus, it is useful for many Latvian language studies [22]. The success of LVK2018 has helped to secure a new project which is fully dedicated to the development of a new 100 million balanced corpus of contemporary Latvian. The new corpus will be created in cooperation with text holders, because there is not that much freely available online data for Latvian. The LVK2018 serves as a great example in the conversations with the text holders.

Acknowledgements

This work has received financial support from the Latvian Language Agency through the grant agreement No. 4.6/2019-029.

References

[1] Sinclair J. EAGLES. Preliminary recommendations on corpus typology. EAGLES Document EAG TCWG-CTYP/P. 1996.
[2] Mititelu VB, Irimia E, Tufis D. CoRoLa – The Reference Corpus of Contemporary Romanian Language. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14); 2014. p. 1235–1239.
[3] Křen M, Cvrček V, Čapka T, Čermáková A, Hnátková M, Chlumská L, et al. SYN2015: Representative corpus of contemporary written Czech. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16); 2016. p. 2522–2528.
[4] Levāne-Petrova K. Līdzsvarotais mūsdienu latviešu valodas tekstu korpuss, tā nozīme gramatikas pētījumos. Valoda: nozīme un forma. 2019;(10):131–146.
[5] Aston G, Burnard L. The BNC handbook: exploring the British National Corpus with SARA. Capstone; 1998.
[6] Burnard L. Reference Guide for the British National Corpus (xml edition), 2007. URL http://www natcorp ox ac uk/XMLedition/URG. 2007.
[7] Čermák F. Today's corpus linguistics: Some open questions. International Journal of Corpus Linguistics. 2002;7(2):265–282.
[8] Hnátková M, Kren M, Procházka P, Skoumalová H. The SYN-series corpora of written Czech. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14); 2014. p. 160–164.
[9] Kovalevskaitė J. Dabartinės lietuvių kalbos tekstynas – 10 metų kaupimo ir naudojimo patirtis. Prace Baltystyczne. 2006;3:231–241.
[10] Rimkutė E, Kovalevskaitė J, Melninkaitė V, Utka A, Vitkutė-Adžgauskienė D. Corpus of contemporary Lithuanian language–the standardised way. In: Human Language Technologies–The Baltic Perspective: Proceedings of the Fourth International Conference Baltic HLT; 2010. p. 154–160.
[11] Levāne-Petrova K. Līdzsvarots mūsdienu latviešu valodas tekstu korpuss un tā tekstu atlases kritēriji. Baltistica. 2012;(8):89–98.
[12] Paikens P. Lexicon-based morphological analysis of Latvian language. In: Proceedings of the 3rd Baltic Conference on Human Language Technologies (Baltic HLT); 2007. p. 235–240.

[13] Paikens P, Rituma L, Pretkalniņa L. Morphological analysis with limited resources: Latvian example. In: Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013); 2013. p. 267–277.

[14] Paikens P. Deep Neural Learning Approaches for Latvian Morphological Tagging. In: Human Language Technologies–The Baltic Perspective: Proceedings of the Seventh International Conference Baltic HLT 2016. vol. 289. IOS Press; 2016. p. 160–166.

[15] Darģis R, Auziņa I, Bojārs U, Paikens P, Znotņš A. Annotation of the corpus of the Saeima with multilingual standards. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018); 2018. .

[16] Bray J, Curtis J. An ordination of upland forest communities of southern Wisconsin. Ecological Monographs (27). PRIMER-E Plymouth; 1957.

[17] Rituma L, Saulīte B, Nešpore-Bērzkalne G. Latviešu valodas sintaktiski marķētā korpusa gramatikas modelis. Valoda: nozīme un forma. 2019;(10):200–216.

[18] Pretkalniņa L, Nešpore G, Levāne-Petrova K, Saulīte B. A Prague Markup Language profile for the SemTi-Kamols grammar model. In: Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011); 2011. p. 303–306.

[19] Gruzitis N, Pretkalnina L, Saulite B, Rituma L, Nespore-Berzkalne G, Znotins A, et al. Creation of a balanced state-of-the-art multilayer corpus for NLU. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018); 2018. p. 4506–4513.

[20] Gruzitis N, Nespore-Berzkalne G, Saulite B. Creation of Latvian FrameNet based on Universal Dependencies. In: Proceedings of the International FrameNet Workshop (IFNW); 2018. p. 23–27.

[21] Rychlỳ P. Manatee/Bonito-A Modular Corpus Manager. In: RASLAN; 2007. p. 65–70.

[22] Holvoet A. The Middle Voice in Baltic. vol. 5. John Benjamins Publishing Company; 2020.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200615

Creation of Language Resources for the Development of a Medical Speech Recognition System for Latvian

Roberts DARĢIS [1], Normunds GRŪZĪTIS, Ilze AUZIŅA and Kaspars STEPANOVS
Institute of Mathematics and Computer Science, University of Latvia, Latvia
Riga East University Hospital, Latvia

Abstract. This paper describes an ongoing work on the creation of Latvian language resources for the medical domain focusing on digital imaging to develop a medical speech recognition system for Latvian. The language resources include a pronunciation lexicon, a text corpus for language modelling, and an orthographically transcribed speech corpus for the (i) adaptation of the acoustic model, (ii) evaluation of the speech recognition accuracy, (iii) development and testing of rewrite rules for automatic text conversion to the spoken form and back to the written form. This work is part of a larger industry-driven research project which aims at the development of specific Latvian speech recognition systems for the medical domain.

Keywords. Speech recognition, language resources, medical domain, Latvian language

1. Introduction

This paper describes the creation of domain-specific language resources required for the development of a medical speech recognition system. The language resources of interest are: an anonymised text corpus of medical reports, namely digital imaging reports and epicrisis reports (excerpts from an archive) for language modelling; a pronunciation lexicon of medical terms, abbreviations and named entities for their recognition and consistent transcription; an anonymised and orthographically transcribed speech corpus for adapting the acoustic model, evaluating the adapted speech recognition systems, developing and testing automatic pre-editing and post-editing rules to rewrite the existing final reports to their spoken form (as if dictated) and dictations to the expected written form.

Since the creation of a relatively large[2] general-domain speech corpus for Latvian [1], various automatic speech recognition (ASR) systems of industrial applicability have been developed for Latvian [2], [3]. ASR systems trained on general-purpose

[1]Corresponding Author: Roberts Darģis, Artificial Intelligence Laboratory, Institute of Mathematics and Computer Science, University of Latvia, Raiņa bulv. 29, Riga, LV-1459, Latvia; E-mail: roberts.dargis@lumii.lv.

[2]100 hours of annotated and orthographically transcribed balanced audio data, of which 4 hours are also phonetically transcribed.

speech and text corpora, however, are not applicable[3] for the very specific language of medical reports. Domain-adapted language model and pronunciation lexicon (both derived from a text corpus of written medical reports) give the most significant boost in ASR accuracy, while a domain-adapted acoustic model (derived from a speech corpus of dictated medical reports) makes a considerable impact as well [4].

The work presented in this paper is a part of an ongoing collaborative project between a language technology research group and the largest hospital in Latvia on Latvian ASR for medical applications. Although modern medical technology is widely used in Latvia, particularly for imaging diagnostics, medical reports are still produced completely manually. The largest healthcare institutions in Latvia maintain or outsource services of transcriptionist centres to produce medical reports. However, the number of diagnostic examinations is constantly growing, and clinicians and patients must wait up to several days for the reports. Moreover, transcriptionist services are expensive, and regional healthcare institutions cannot afford them. Inspired by the successful implementation of Estonian ASR for radiology [4], [5], our goal is to create the essential language resource and technology components for Latvian ASR in the medical domain, particularly in radiology, and showcase their usage by developing and validating an automated dictation platform for radiology reporting. The platform will be customised for two usage scenarios. First, we expect that for subdomains that typically produce simpler and more fluent dictations (e.g. X-ray and ultrasonography), radiologists will use ASR in a self-service manner. Second, we expect that a transcriptionist centre might still be preferred by radiologists in the case of more complicated dictations (e.g. computed tomography and magnetic resonance); however, a centralised transcriptionist centre would become more productive since part of the workload would be moved to the self-service scenario, and draft reports (via ASR) would be available for the complex dictations.

2. Text Corpus

The data source for the creation of the text corpus is an archive of medical reports produced over the last decade in a large multi-profile hospital.

The corpus creation involves several steps. The first step is the extraction of plain-text paragraphs from the description part and the conclusion part of the actual reports stored in the archive. The hospital's transcriptionist centre produces, so far manually, Microsoft Word documents from audio files dictated and submitted by radiologists, clinicians and other doctors.

Over the time, there are many different templates used for the documents, and multiple nested tables are used to create layout for the templates. The documents also contain various text formatting elements, such as underlines, bold, and italic. In the text extraction step, it is important to preserve the text segmentation, since language modelling relies on correct text division into sentences. The documents contain multiple short-text segments without any punctuation marks, such as table column names and field names. If all the extracted text segments would be concatenated as is, such short-text segments would be incorrectly added to other sentences. An opposite issue would occur if larger text paragraphs would be divided into smaller segments due to mistakenly recognised text formatting elements as paragraph separators.

[3]The word error rate (WER) is way too high for efficient usage of non-adapted ASR systems.

Multiple third-party plain-text extraction tools were tested. Unfortunately, all of them had the same issue: either too long or too short text segments are extracted. Therefore a custom plain-text extraction tool was developed using the underlying XML structure of a Word document. Each paragraph is represented by a *p* tag, and each text segment inside the paragraph is encoded by a *t* tag. This method correctly extracts text segments from table cells as well, since text in each cell is encoded in its own paragraph.

The second step is the anonymisation of the extracted texts to avoid any sensitive personal data (name, surname, ID number, etc.) to be included in the text corpus and, thus, in the language model. Although the description and conclusions parts of medical reports should not contain any personal data, it is still possible that some irrelevant text segments containing personal data are extracted by mistake.

The available amount of archived text is more than necessary for language modelling, therefore, text anonymisation was prioritised over text preservation. To reduce the risk of personal information leaking into text corpus, anonymisation is done at the paragraph level. If a paragraph contains any tokens recognised as potential personal data, the whole paragraph is excluded from the corpus. The remaining paragraphs are split into sentences, since language modelling is done at the sentence level.

The first two steps are executed in the hospital's IT infrastructure, so that no sensitive data is handed over to the research partner. Anonymised plain-text were extracted from 100k reports covering 8 years of reporting. The number of reports produced each year is steadily increasing, reaching 15k in 2018.

The third step is text normalisation w.r.t. the tokenization and correction of typical typos. If a sentence contains words recognised as typos, they are automatically corrected according to the annotations made during the lexicon development (Section 3).

The fourth step is automatic expansion (verbalization) of numbers, abbreviations, symbols and other such tokens. In the result, we effectively acquire a parallel corpus of collapsed (original) and expanded texts. We follow the successful approach used by Alumäe et al. [5]. A small but representative part of the corpus is expanded manually by domain experts. Additional parallel texts are acquired through the speech corpus creation (Section 4). Based on these subsets, context-sensitive rewrite rules are defined for automatic expansion of the rest of the text corpus. Additionally, random punctuation marks are verbalised as commands for structural formatting. The language models are acquired from the verbalised version of the text corpus.

3. Pronunciation Lexicon

Development of the lexicon is the next step after text extraction (Section 2). The lexicon contains a list of words and their pronunciation that the speech recognition system should recognise.

The lexicon is derived from the text corpus. First, the texts are tokenized, and all the unique words are extracted. Their pronunciation is generated semi-automatically.

There are four types of words w.r.t. the pronunciation generation:

- words in Latvian, which are pronounced exactly as written (the majority of words);
- words containing typos, which should not be included in the lexicon as they are;
- abbreviations and symbols that need to be expanded to generate pronunciation;

- words in a language other than Latvian (e.g. drug names and Latin terms), which are specifically pronounced in the Latvian context, i.e., phonetic transcription based on Latvian phonemes has to be provided.

Radiologists tend to pronounce some words incorrectly because it is easier, faster or just more convenient. Additional pronunciation variants are extracted from the speech corpus (Section 4) to better reflect pronunciation variation in the actual dictations.

Words that are part of personal data (name, surname, address) are semi-automatically separated and used as a supplementary filter in text anonymisation (Section 2).

In total, there are 1.8M unique tokens in the extracted text corpus, of which 1.1M are tokens that contain at least one letter and are, thus, considered for inclusion in the lexicon.

Manual categorisation of all words into the above mentioned four groups would be an inefficient and time consuming task.

Therefore, several existing dictionaries were used to categorise most of the words automatically. First, words that are included in the largest open dictionary of Latvian Tezaurs.lv [6] were automatically marked as standard words. In the next iteration, an open-source Latvian NLP pipeline [7] (namely, a morphological tagger and a named entity recogniser) was used to recognise words that are possibly part of personal data. Words that were not classified automatically need to be manually classified and transcribed. Words that occur at least 1,000 times in the text corpus (almost 14k words)[4], were selected for the manual review process in the first iteration.

The lexicon and the frequency information is also used as a filter in the development of the language model. Rarely or incorrectly used words and words that possibly constitute personal data are excluded from the language model.

4. Speech Corpus

A domain-specific orthographically transcribed speech corpus is a key component in the adaptation of the acoustic model as well as for the evaluation of the specialised ASR systems.

Adaptation of the acoustic model generally helps to reduce the word error rate (WER). It makes even a bigger impact in ASR settings where the potential user pool is limited, as it is in the medical domain, and the ASR system can be adapted to the speakers. In the past year, there have been 71 unique speakers.

Speech corpus also allows to evaluate how well all the ASR components work together: how well the acoustic model predicts the environment, how well the pronunciation modelled in the lexicon matches the actual pronunciation, and how well the language model predicts the text.

Actual dictation records are used for the creation of the speech corpus. Records are selected from the hospital's transcriptionist centre's archive of recordings. Low quality records that have been dictated over a telephone, or contain significant background noise or parallel speech, or are dictated by non-native speakers with sever accent and many pronunciation errors are omitted.

[4] An empirical threshold based on the word frequency distribution in the extracted text corpus.

```
1 Izmeklējums CT         vēdera dobumam -
2 izmeklējums CT [cē tē] vēdera dobumam {dash}
3 izmeklējums CT         vēdera dobumam domuzīme
4 Izmeklējums CT         vēdera dobumam -

1 Natīvs izmeklējums ar p/o      kontrastētu kuņģa-zarnu traktu
2 natīvs izmeklējums ar perorāli kontrastētu kuņģa zarnu traktu
3 natīvs izmeklējums ar perorāli kontrastētu kuņģa zarnu traktu
4 Natīvs izmeklējums ar p/o      kontrastētu kuņģa-zarnu traktu

1 izmeklējums pēc i/v         ievadīta Ultravist 300
2 izmeklējums pēc intravenozi ievadīta Ultravist trīssimt [trīssimti]
3 izmeklējums pēc intravenozi ievadīta Ultravist trīssimt
4 izmeklējums pēc i/v         ievadīta Ultravist 300

1 Pancreas            difūza tauku involūcija , veidojumus        neredzu
2 pancreas [pankreas] difūza tauku involūcija   veidojumus nesas* neredzu
3 pancreas            difūza tauku involūcija   veidojumus        neredzu
4 Pancreas            difūza tauku involūcija   veidojumus        neredzu

1 kreisā niere mazāka apjomā ,       plānāku parenhīmu .
2 kreisā niere mazāka apjomā {comma} plānāku parenhīmu {full-stop}
3 kreisā niere mazāka apjomā komats  plānāku parenhīmu punkts
4 kreisā niere mazāka apjomā ,       plānāku parenhīmu .

1 Kreisās       nieres konkrements 0,39             cm          / Ø
2        labās nieres konkrements nulle trīs deviņi centimetri   diametrā
3        labās nieres konkrements nulle trīs deviņi centimetri   diametrā
4        Labās nieres konkrements 0,39             cm          / Ø
```

Figure 1. Sample excerpts from the aligned text and speech corpora. Lines: 1 – a text span from an archived written report; 2 – an orthographically transcribed and annotated segment from the corresponding dictation record; 3 – the corresponding text span from the derived text corpus for language modelling; 4 – the expected final output of the ASR system

We are aiming at a 30 hour orthographically transcribed corpus, part of which will be used for the evaluation purposes. The transcriptions are bootstrapped by aligning the corresponding expanded text segments of the extracted text corpus (Section 2). The automatic alignment is, in general, partial, and it is manually post-edited by experts. Simple annotation guidelines are used, e.g. on how to annotate pronunciation of specific terms, based on the experience gained in the creation of the general-purpose Latvian speech corpus [1]. Additionally, text formatting commands are annotated, based on the previous work on designing a general-purpose dictation corpus of Latvian [8].

Figure 1 illustrates the orthographic transcription and annotation of the speech corpus (see Line 2 in each of the six samples). For each token, pronunciation is given in square brackets next to the token if its pronunciation differs from the written form.[5] For instance, *CT* ('computed tomography') is pronounced as *cē tē* in the 1st sample. Pronunciation is given also in cases where a word that should be pronounced as it is written, is

[5] In Latvian, words are mostly pronounced as they are written.

pronounced incorrectly, as it often occurs with numbers: e.g., *trīssimt* ('three hundred') is pronounced as *trīssimti* in the 3rd sample. Pronunciation annotations are used for automatic extension of the pronunciation dictionary. Text formatting instructions (radiologist-to-transcriptionist) are segmented within braces as in the 1st and 5th samples (in Figure 1, instructions are translated in English for clarity). Numbers are expanded to their spoken form as in the 3rd and 6th samples. In each sample, Line 1 represents the corresponding text span in the archived report. Observing the differences between Line 1 and Line 2 in the aligned speech and text corpora, rewrite rules are specified to automatically expand the text corpus of archived reports into a derived text corpus (Line 3) for language modelling. Rewrite rules typically deal with abbreviations, symbols and numbers (see the 2nd, 3rd and 6th examples). Line 4 represents the expected final output of the ASR system, where transcriptions are converted from the spoken form into the standard form of written reports by using reverse rewrite rules.

Anonymisation of specific segments in audio files is a lot more challenging task than text anonymisation. It would involve transcripts of inaccurate ASR, therefore, it could not be guaranteed that the anonymisation is done accurately. Since personal data should be dictated only at the beginning of a report, we are using a more simple and reliable approach instead: given the anonymised text corpus (Section 2), we extract the main body of the report (cropping off any metadata before and after the main text) and align it with the ASR result to find the segment of the given audio file, which corresponds to the main body of the text; the rest of the audio file is trimmed.

5. Conclusion

The language resources described in this paper are the most crucial part in the adaptation of a speech recognition system for the medical domain or any other highly specialised domain. The expected end result is dual: a self-service platform for instant radiology reporting, and a semi-automated platform for a more productive work at the transcriptionist centre. The next steps are further development of the text rewriting system for automatic verbalisation and deverbalisation, adaption of the ASR system for the medical domain, and the platform development, followed by evaluation and user studies. High accuracy ASR accompanied by productive user interfaces must be achieved for the overall system to be accepted and used in practice.

Acknowledgements

This work has received financial support from the European Regional Development Fund under the grant agreement No. 1.1.1.1/18/A/153 (*Latvian Speech Recognition and Synthesis for Medical Applications*).

References

[1] Pinnis M, Auzina I, Goba K. Designing the Latvian speech recognition corpus. In: Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC). Reykjavik, Iceland; 2014. p. 1547–1553.

[2] Salimbajevs A, Strigins J. Latvian speech-to-text transcription service. In: Proceedings of the 16th Annual Conference of the International Speech Communication Association (INTERSPEECH). Dresden, Germany; 2015. p. 722–723.

[3] Znotins A, Polis K, Dargis R. Media monitoring system for Latvian radio and TV broadcasts. In: Proceedings of the 16th Annual Conference of the International Speech Communication Association (INTERSPEECH). Dresden, Germany; 2015. p. 732–733.

[4] Paats A, Alumäe T, Meister E, Fridolin I. Retrospective Analysis of Clinical Performance of an Estonian Speech Recognition System for Radiology: Effects of Different Acoustic and Language Models. Journal of Digital Imaging. 2018;31(5):615–621.

[5] Alumäe T, Paats A, Fridolin I, Meister E. Implementation of a Radiology Speech Recognition System for Estonian Using Open Source Software. In: Proceedings of the 18th Annual Conference of the International Speech Communication Association (INTERSPEECH); 2017. p. 2168–2172.

[6] Spektors A, Auzina I, Dargis R, Gruzitis N, Paikens P, Pretkalnina L, et al. Tezaurs.lv: the largest open lexical database for Latvian. In: Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC). Portoroz, Slovenia; 2016. p. 2568–2571.

[7] Znotins A, Cirule E. NLP-PIPE: Latvian NLP Tool Pipeline. In: Human Language Technologies - The Baltic Perspective. vol. 307 of Frontiers in Artificial Intelligence and Applications. IOS Press; 2018. p. 183–189.

[8] Pinnis M, Salimbajevs A, Auzina I. Designing a speech corpus for the development and evaluation of dictation systems in Latvian. In: Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC). Portoroz, Slovenia; 2016. p. 775–780.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200616

Towards the Development of Language Analysis Tools for the Written Latgalian Language

Daiga DEKSNE [a,1] and Anna VULĀNE [b]
[a] *Tilde, Latvia*
[b] *Latvian Language Institute of the University of Latvia, Latvia*

Abstract. This paper reports on the development of spell checking and morphological analysis tools for Latgalian. The Latgalian written language is a historic variant of the Latvian language. There is a wide range of language analysis tools available for Latvian, whereas the Latgalian language lacks such tools. The work is done by the joint effort of linguists who work on morphologically marked lexicon creation and IT specialists who work on language tool development. For the creation of a morphological analysis tool, we reuse the FST technology used for the Latvian morphological analyzer. We create a spelling dictionary that can be used with the Hunspell engine. All tools are accessible via Web Service. For now, the Latgalian lexicon contains 13,139 lemmas marked by 105 inflection groups. The work of lexicon replenishment still continues.

Keywords. Proofing tools, morphological analysis, spell checking, FST transducer

1. Introduction

In the last decade, various essential language tools and technologies have been developed for the Latvian language, such as spelling checkers, morphological analyzers, taggers and parsers, speech recognition and synthesis tools, and machine translation systems [1]. Unfortunately, there is a lack of such tools for Latgalian.

There are three dialects in the territory of Latvia: the Livonian Dialect, the Middle Dialect, and the High Latvian dialect. Latvian literary language has been formed on the Middle Dialect. Latgalian is based on the Latgalian subdialects of the High Latvian dialect. It differs significantly from the Middle and Livonian Dialects as well as the Latvian literary language. The Latgalian written language is a historic variant of the Latvian language. In the 16th century, German clergymen began developing a language of writings on the basis of the low Latvian dialects. Later, this variety became the normed language of Latvian. On the other hand, in the 18th century, Catholic clerics began to form the language of Latgalian writings on the basis of the high Latvian (Latgalian) dialects.

Latgale is the eastern region of Latvia which was separated by a state border from the rest of Latvia's territory for almost 300 years, and it was then that the Latgalian written language developed. These historical circumstances determined the development

[1] Corresponding Author: Daiga Deksne; Tilde, Vienibas gatve 75a, Riga, Latvia, LV1004; E-mail: daiga.deksne@tilde.lv.

of a second written language tradition. Nowadays, the Latgalian written language is regularly used in Roman Catholic churches and some schools in the region, by book publishers and media, it can be heard in theatres and at concerts, on radio and television, at public events and gatherings around the country, as well as on the Web. Therefore, it is necessary to create a digital spelling tool to maintain the high quality of language use. Latgalians use either the native dialect or the Latvian literary language in their spoken communications. According to the Census 2011, 8.8 % of the Latvian population speak Latgalian on a daily basis, with 5.7 % of all children up to the age of 17. “Latgalian is spoken the most in Latgale region – 35.5 % of the population, although this reduces to 27 % amongst children up to the age of 17” [4].

The goal of this project is to create morphological analysis and spelling checking tools for the written Latgalian language. The work is done by the joint effort of linguists and IT specialists. Linguists with Latgalian language knolwedge develop a morphologically marked lexicon. The task of the IT specialists is to create tools that will help users to learn Latgalian and aid in text production without spelling mistakes.

The workflow for the creation of the language analysis tools consists of several steps:

- Establishing an environment for the creation of the morphologically marked lexicon (done by IT specialists and linguists);
- Creation of the lexicon (done by linguists);
- Development of language analysis tools (done by IT specialists);
- Development of a client-side application (done by IT specialists);
- Checking errors in the lexicon and identifying missing entries using a client-side application (done by linguists);
- Fixing errors and replenishing the lexicon (done by linguists);
- Rebuilding the tools including changes made in the lexicon (done by IT specialists).

There are several ways how morphologically marked lexicon can be created. The authors of *Grammatical Dictionary of Polish* [1] define inflection groups in a relational database. Words in the dictionary are linked to inflection groups. *Croatian Morphological Lexicon* [2] has three parts: a list of lemmas containing stems and inflectional pattern numbers, a list of endings including morphosyntactic category values, and a list of transformations applied on stems when concatenating words from morphemes. To facilitate the work of lexicographers, we used a simple approach. Template examples for paradigm definition for different part-of-speech were created specifying word forms used in Latvian. Linguists modified them and filled in with example word forms to cover all inflection patterns used in Latgalian. Defined inflection groups were used for word marking in the lexicon.

2. Creation of Morphologically Marked Lexicon

Some linguistic features of Latgalian not found in Latvian are as follows:

- Complete opposition of hard and soft consonants;
- The use of graphemes *y* and *uo*, which are not in the Latvian alphabet;
- Endings *-ys*, *-is* in place of Latvian *-as*, *-es*;
- Change of vowels under the influence of the next syllable vowel (*g**o**b**o**ls* and *g**a**b**a**leņš*);

- Prefix *da-* and the use of formants *-za-* and *-sa-* in reflexive prefixed verbs and nomens with endings *-inis*, *-ine*;
- A wider participle system;
- Many words have preserved more ancient meanings.

Given the fact that Latgalian written language is only taught in a handful of schools in Latgale on an optional basis or, alternatively, can only be learned through courses or self study, only a small proportion of the population are familiar with the orthographic norms of the Latgalian written language. Therefore, errors are prevalent in informal communications such as social media, text messaging, on-line comments, unedited literary works, etc. The most common of these include:

- The softening of consonants before the letters *e* or *i* (e.g. ***ņ**edeļa* : ***n**edeļa*);
- The use of an inappropriate root, suffix vowel or consonant, which is influenced by the specificity of pronunciation in a subdialect. For example, *ir – jir*, *jer*, *prīd**ē** – prīd**ī***, *kol**ni** – kol**ny***, *ga**rs** – gar**š***, *puiš**ki**ns – puiš**ky**ns*, *skrēj**a** – skrēj**e***;
- The use of dialect specific person forms of verbs, such as *ūgoiļom – ūgojom*, *dzīduo – dzīduoja*;
- The lack of consistency due to simultaneous use of the orthographic norms of 1929 [5] and 2007 [6];
- The phonetic translation of words from Latvian literary language;
- The use of written language according to the pronunciation in a subdialect, thereby, disregarding orthography.

A number of parallel forms were allowed to be in use during the transition to the improved and refined orthography. However, variants of graphemes, morphemes and forms were not created during the development of the spelling tool as unsubstantiated variability of word forms can lead to delayed embedding of the spelling rules.

Given the spelling tool has been developed based on the spelling rules of 2007, a number of typical mistakes were identified in the articles that were published in line with the orthographic norms of 1929 (see Table 1).

Table 1. Orthographic norms of 2007 and 1927

Language Unit	2007	1927
Diphthong designation	*uo – m**uo**te* *iu – **iu**diņs*	*ō – m**ō**te* *yu – **yu**diņs*
Diphthong of a root	*p**ie**rts*	*p**ē**rts*
Ending following the hard or palatalized consonant (in singular genitive, plural nominative, and plural accusative forms)	*-ys*, *-is* *lop**ys***, *kuoj**is***, *muot**is***	*-as*, *-es* *lop**as***, *kōj**as***, *muot**es***
Suffix	*-eja – Latv**eja***	*-ija – Latv**ija***
Past and future forms of 2nd and 3rd conjugation verbs with *-ēt*	*kav**ē**t*, *redz**ē**t – kav**ie**ju*, *redz**ie**ju*; *kav**ie**šu*, *redz**ie**šu*	*kav**ē**ju*, *redz**ē**ju*; *kav**ē**šu* *redz**ē**šu*

All parts of speech were covered and the following sources were used during the creation of the lexicon:

- Published Latgalian language dictionaries of spelling [7], [8];
- *Kalupe Subdialect Dictionary* [9];
- Scientific articles on Latgalian vocabulary and word-formation;
- Unpublished material on Latgalian subdialects and written language;
- Press and fiction texts;

- The meeting minutes and decisions of the Latgalian Written Language Sub-committee meetings on the correct spelling of work positions and professions, names of residents and toponyms, and other material.

In addition to the widely used vocabulary, different variants of the same concept encountered in dialects were included, however, spoken vocabulary was not provided. The notation on word-forms observed the spelling rules of the Latgalian language that stem from parallel forms and are found in dialects. Furthermore, older word-forms were included in the spelling tool in order to facilitate and preserve their use. In the case of homography, and if a word is part of two different paradigms, both words were included and the appropriate morphological classification was provided.

The language material was arranged in two files containing a description of a morphological system and a vocabulary. Paradigms of all word classes that can be inflected were developed and classified; a lexicon was created and morphologically marked. If applicable, the diminutive forms of nouns were included, as well as the present, past and participle forms of verbs. See Table 2 for examples of verb records in the lexicon. All columns are not filled for verbs belonging to the groups where inflected forms have the same stem as infinitive or can be derived by regular rules.

Table 2. Example of verb records in the Latgalian lexicon

word	group	pres1p	pres2p	pres3p	past1p	past3p	ppmasc	ppfem
bēgt	V12a	*bāgu*	*bēdz*	*bāg*	*biegu*	*bāga*	*biedzs*	*bāguse*
cyluot	V2uot							
badeit	V3eit							

Participles were not included as separate entries in the lexicon as they are automatically generated from the verb stems. See Table 3 for complete statistics of different part-of-speech words in the lexicon and the inflection groups defined.

Table 3. Statistics of the Latgalian lexicon

Part-of-speech	Number of lemmas	Number of groups
noun	5,010	29
verb	5,435	29
adjective	1,302	15
pronoun	109	15
adverb	931	1
numeral	140	12
particle	34	1
conjunction	23	1
preposition	18	1
interjection	137	1
Total	13,139	105

3. Development of Language Analysis Tools

We reuse Finite state transducer (FST) technology used in the development of the Latvian morphological analyzer [10].

3.1. Finite State Transducer

For the lexicon description, we use the Stuttgart Finite-State Transducer Toolkit (SFST) [11] as it allows the use of regular expressions, variables and different operators for text string transformation – concatenation, composition, insertion, and others. For transducer compilation, we use OpenFst toolkit[2]. For verbs, nouns and adjectives, we define inflection classes containing information about every word-form in the paradigm – form identifier, word-form's ending, corresponding lemma's ending, tags signalizing to which stem an ending can attach (see Figure 1). Once defined, this part of the transducer is reused when new entries are added to the lexicon.

```
$N5pl$ = <normEnd>{is}:{is}<414>:<n> |\
<altEnd1>{u}:{is}<415>:<n> |\
<normEnd>{em}:{is}<416>:<n> |\
<normEnd>{is}:{is}<417>:<n> |\
<normEnd>{em}:{is}<418>:<n> |\
<normEnd>{ēs}:{is}<419>:<n> |\
<normEnd>{is}:{is}<420>:<n>
```

Figure 1. Example of noun declension class definition

The dynamic part of the transducer is a set of stems linked to the declension groups (see Figure 2). This set is recreated when the lexicon is changed. Nouns can have up to three stems. Verbs have up to 11 stems according to conjugation paradigm. Stems not specified in the lexicon are generated according to regular palatalization rules.

```
<N5pl>    Dekšuor|is Dekšuor|u
<N5pl>    pušdīn|is  pušdīņ|u
<N5pl>    zuol|is    zuoļ|u
```

Figure 2. Example of noun stem representation

The non-inflectional part-of-speech words are represented as lexical entries followed by form identifiers. Adverbs, numerals, and pronouns are also included as lexical entries along with information on how to generate lemma from inflected form (see Figure 3).

```
tu<1456>:<p> |\
{teve}:{tu}<1457>:<p> |\
{tev}:{tu}<1458>:<p> |\
{tevi}:{tu}<1459>:<p> |\
{tevim}:{tu}<1460>:<p> |\
{tevī}:{tu}<1461>:<p> |\
```

Figure 3. Lexical entries for pronoun *tu* ('you')

Words in the transducer are represented as concatenation of separate parts. For example, verbs are represented as concatenation of items from a prefix set, a stem set,

[2] http://www.openfst.org/.

and an ending set. The prefix set has only three items – a prefix for negation, a prefix for the debitive mood form, and an empty prefix. The stem set has verb stems sorted by conjugation classes. The ending set has endings sorted by conjugation classes. The correct word forms are obtained by matching tags of constituent parts. For example, there are tags in the prefix part and the stem part that must match the ending part.

In the compiled transducer, the form identifiers are replaced by the morphological description strings that are based on MULTEXT-East format [12]. Each form's description is 28 symbols long string. Each position in a string is reserved for the value of a particular grammatical feature. The first position is reserved for part-of-speech, the second – for tense, the third – for gender, the fourth – for number, the fifth – for case, etc. Values of features are represented by a single symbol in a particular position. For example, in the second position, symbol 'p' (present tense), 's' (past tense), or 'f' (future tense) can be found. Not all positions are filled in for every word as each part-of-speech word has a different set of features. Verbs have tense, number, person, and mood. Nouns have gender, number, case, and diminutive marker. Adjectives have gender, number, case, definite ending marker, and comparative forms. Positions that are not relevant for the particular part-of-speech word are filled with value '0'.

We built two transducers. One provides morphological analysis description for a given word, whereas another generates all word-forms for a given lemma. The output is presented in XML format (see Figure 4).


```
<document><source_info original='Afrika' />
<word pos='n' baseform='Afrika'>
<form descr='n0fsn000000000n0000000000000' spelling='Afrika' />
<form descr='n0fsg000000000n0000000000000' spelling='Afrikys' />
<form descr='n0fsd000000000n0000000000000' spelling='Afrikai' />
<form descr='n0fsa000000000n0000000000000' spelling='Afriku' />
<form descr='n0fsi000000000n0000000000000' spelling='Afriku' />
<form descr='n0fsl000000000n0000000000000' spelling='Afrikā' />
<form descr='n0fsv000000000n0000000000000' spelling='Afrika' />
</word></document>
```


Figure 4. Example of form generation result in xml format

In the case of homoforms, description of every form for a given word is provided as well as the lemma of a particular word form. For example, the word 'molu' can be a verb in past tense, first person singular, indicative mood form and a noun in singular accusative, singular instrumental or plural genitive form (see Table 4).

Table 4. Morphological analysis of word 'molu'

Lemma	Part-of-speech	Form description
mola (side)	noun	<n0fsa000000000n0000000000000>
mola (side)	noun	<n0fsi000000000n0000000000000>
mola (side)	noun	<n0fpg000000000n0000000000000>
maļt (to grind)	verb	<vs0s00100i000000000000000000>

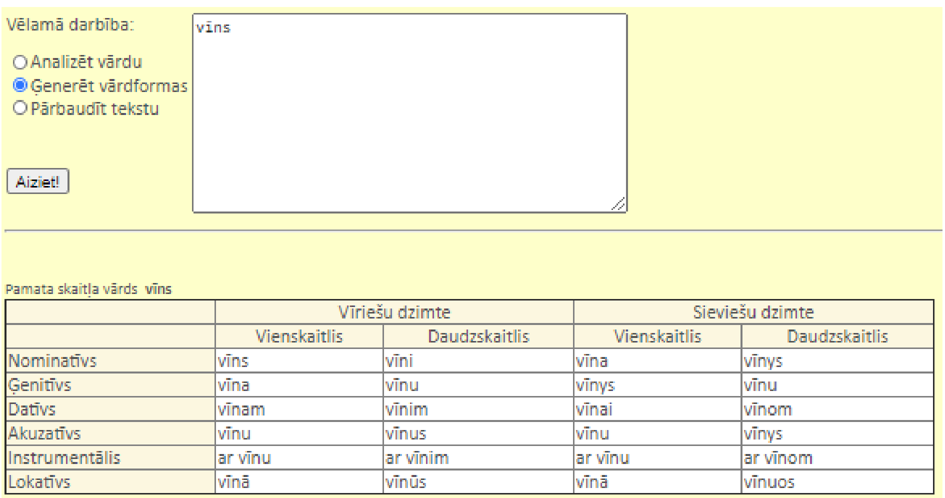
3.2. Hunspell Dictionary

We build the spell checking tool using the Hunspell library[3]. The dictionary for the spell checking tool is compiled from the files prepared for transducer compilation. The spelling tool checks text from the standard input (*stdin*). The produced output is in the HTML format. The misspelled words are included in <*span*> tags containing spelling suggestions in the *title* attribute. In such a way, spelling suggestions are shown as a tooltip when the mouse moves over a particular <*span*> element in any HTML browser application.

Another way how to use a compiled dictionary is by using plug-ins supporting Hunspell format, for example, DSpellCheck[4] plug-in.

4. Language Analysis Web Service

A Web Service is created to access the functionality of the developed tools from the Web environment. We have created an initial version of the Web form that enables users to check the correctness of a text, to see the morphological description of a desired word-form, and to see the tables displaying the full paradigm for a given word (see Figure 5).



	Vīriešu dzimte		Sieviešu dzimte	
	Vienskaitlis	Daudzskaitlis	Vienskaitlis	Daudzskaitlis
Nominatīvs	vīns	vīni	vīna	vīnys
Ģenitīvs	vīna	vīnu	vīnys	vīnu
Datīvs	vīnam	vīnim	vīnai	vīnom
Akuzatīvs	vīnu	vīnus	vīnu	vīnys
Instrumentālis	ar vīnu	ar vīnim	ar vīnu	ar vīnom
Lokatīvs	vīnā	vīnūs	vīnā	vīnuos

Figure 5. Web form with Inflection table for numeral *vīns* ('one')

5. Conclusion

In this paper, we described the creation of morphological analysis and spelling tools for the Latgalian written language. The work is still in progress. We have finished the first steps in the project as a result of which 105 inflection paradigms used in Latgalian have been defined and the basic lexicon containing 13,136 entries created. The definitions of inflection paradigms as well as the lexicon entries have been transferred to the finite state

[3] https://github.com/hunspell/hunspell.
[4] https://github.com/Predelnik/DSpellCheck.

transducer, and morphological analysis and spelling checking components created. The initial version of the Web Service allows analyzing a word, seeing its inflection paradigm, and checking the spelling of a text. The next phase involves an active work from linguists in checking the correctness of the words in the Latgalian lexicon and identifying missing entries.

Acknowledgments

Creation of the morphologically annotated Latgalian lexicon was supported by National Research Programme project “Latvian Language” (№ VPP-IZM-2018/2-0002).

References

[1] Woliński, M. A relational model of Polish inflection in Grammatical Dictionary of Polish. In: Language and Technology Conference; October 5-7, 2007; Poznan, Poland. Springer, Berlin, Heidelberg; c2007. p. 96-106.

[2] Tadić, M, Fulgosi, S. Building the Croatian morphological lexicon. In: Proceedings of the 2003 EACL Workshop on Morphological Processing of Slavic Languages; April 13th, 2003; Budapest, Hungary. ACL; c2003. p. 41-45.

[3] Skadiņa I. Languages of Baltic Countries in Digital Age. In: Lupeikiene A., Vasilecas O., Dzemyda G. (eds) Databases and Information Systems. Proceedings of the 13th International Baltic Conference on DB&IS 2018; 1-4 July 2018, Trakai, Lithuania. Communications in Computer and Information Science, vol 838. Springer, Cham. p. 32-40.

[4] Spārīte L. (ed.) Bērnu tautība un mājās lietotā valoda. Bērni Latvijā; 2013. Available: http://www.csb.gov.lv/sites/default/files/publikacijas/nr_13_berni_lat-vija_2013_13_00_lv_en.pdf

[5] Strods, P. (kom. pr.) Nuteikumi par latgališu izluksnes ortografiju. In: Zīdūnis, 1929, Nr. 7, 21. (1669)–22. (1670) lpp.

[6] LR Tieslietu ministrijas Valsts valodas centrs. Latgaliešu rakstības noteikumi=Latgalīšu raksteibys nūsacejumi. Rīga, 2007. Available: https://m.likumi.lv/doc.php?id=164904

[7] Bukšs M. Placinskis J. Latgaļu volūdas gramatika un pareizraksteibas vōrdneica. Minhene: Latgaļu izdevnīceiba; 1973. 420 lpp.

[8] Strods P. Pareizraksteibas vōrdneica. Rēzekne: Dorbs un Zineiba; 1933. 213 lpp.

[9] Reķēna A. Kalupes izloksnes vārdnīca I–II. Rīga: Latviešu valodas institūts; 1998. 601 lpp.

[10] Deksne, D. Finite State Morphology Tool for Latvian. In: Proceedings of the 11th International Conference on Finite State Methods and Natural Language Processing; July 2013; St Andrews, Scotland. Association for Computational Linguistics; c2013; p. 49-53.

[11] Schmid H. A Programming Language for Finite State Transducers. In: Yli-Jyrä A., Karttunen L., Karhumäki J. (eds) Proceedings of the 5th International Workshop on Finite State Methods in Natural Language Processing (FSMNLP 2005); 2005 September 1-2; Helsinki, Finland. Lecture Notes in Computer Science, vol 4002. Springer, Berlin, Heidelberg.

[12] Erjavec T. MULTEXT-East: Morphosyntactic Resources for Central and Eastern European Languages. Language Resources and Evaluation. 2012; 46/1:131-142.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200617

Adding Compound Splitting and Analysis to a Semantic Tagger of Modern Standard Finnish – On the Way to FiSTComp

KIMMO KETTUNEN[1]
University of Eastern Finland, Joensuu, Finland

Abstract. This study continues a work in progress for implementing a full-text lexical semantic tagger for Finnish, FiST. The tagger is based on a 46,226 lexeme semantic lexicon of Finnish that was published in 2016 [1]. Kettunen [2], [3] describes the basic working version of FiST. FiST is based on freely available components: the first implementation uses Omorfi and FinnPos for morphological analysis and disambiguation of Finnish words. The current paper describes work with compound splitting for semantic tagging and its effects on the lexical coverage of the tagger. We try out two different approaches to morphological analysis and disambiguation of words for an improved version of FiST, FiSTComp: FinnPos [4], and Turku Dependency Parser [5], [6], UD1. Both these tools disambiguate morphological interpretations of words and provide boundary markings for compounds, but details and granularity of constituent decomposition vary. Our results with two-, three and four-part compounds show that analysis of compounds through their constituents with UD1 may improve the lexical coverage of the tagger with about 6.6 % units at best. Although we are able to proceed in basic problems of compound splitting, the results are still initial and further work is needed as compounds are a complex phenomenon.

Keywords. Semantic tagging, compounds, Finnish

1. Introduction

Kettunen [2], [3] has introduced the first version of a lexical semantic tagger of modern standard Finnish called FiST. Details of the tagger's implementation and first evaluation results are described in [2], and [3] continues with more evaluation. [7] have used the tagger for analysis of Finnish parliamentary speeches related to rights of everyman in three different decades. [1] describes the Finnish semantic lexicon and principles of its compilation in detail[2]. [1] also evaluates a now obsolete Finnish semantic tagger of Kielikone Ltd. that was the first semantic tagger for Finnish.

So far, the lexical coverage of FiST has been evaluated with about 30 different texts of various genres and sizes. Most of the texts are modern Finnish, but also texts older than 100 years have been analyzed successfully, e.g. the prose of several late 19th and early 20th century Finnish authors. The largest analyzed texts so far have been the Finnish Europarl documents v.6 with 28.6 million words and part of the Open Subtitle

[1] Corresponding Author: Kimmo Kettunen; University of Eastern Finland, Joensuu, Finland; E-mail: Kimmo.kettunen@uef.fi.

[2] The Finnish semantic lexicon is available at https://github.com/UCREL/Multilingual-USAS

collection with 45.2 million words. Both of these analyses achieve lexical coverage of 90.9 % [2], [3].

The first version of FiST analyzed only compounds that were included in its lexicon. As Finnish language uses compounding amply and formation of compounds is quite free, any Finnish lexicon is lacking a great part of compounds found in texts. In this study, we improve the compound handling of FiST by also using the constituents of compounds in semantic analysis. It is obvious that analysis of constituents of compounds should improve the coverage of compounds and the lexical coverage of the tagger, but it is not self-evident what the best practice for performing the analysis is and how much improvement can be achieved.

Our research topic in this paper is twofold: first, we need to find out what type of compound splitting with the available morphological analyzers is most beneficial for semantic tagging of texts. Second, we want to examine how much compound splitting improves the lexical coverage of our Finnish semantic tagger. The solutions we will offer for compound analysis are preliminary, but they are a step forward to a better lexical coverage of the tagger and more comprehensive than the basic compound engine introduced in [1]. [1] introduces a simple compound engine where the last constituent of the compound is separated from the beginning of the word and the two parts are given semantic tags, if possible.

Our test and development data consists of a corpus of speeches at the Finnish Parliament during 1991–2015. The data is part of the The ParlSpeech data set [8]. The data we use is a part of the whole Finnish corpus of 245,852 speeches, and it contains speeches where innovation has been mentioned in the speech. The size of the test and development data is 4,220 speeches and about 2.17 million word tokens. Parliamentary speeches contain probably more compounds than e.g. newspaper texts and thus these texts suit well for our analyses.

2. Compounds in Finnish

2.1. Basics

Creation of compounds, words which are formed by concatenating two or more words without a space between them[3], is a very productive means of making up new words in Finnish and many other languages [9], [10]. Finnish compounds are most often formed from nouns, but other parts of speech can also appear in compounds [10], [13]. Most common are Noun+Noun and Adjective+Noun compounds. According to [14], about 89 % of compounds in *Nykysuomen sanakirja* (Dictionary of Modern Finnish), are nouns. Typical examples of Finnish compounds are e.g. *puutalo* (puu+talo, 'wooden house', literally wood+house), and *ihmisoikeus* (ihmis+oikeus, 'human right'). By adding more words to the beginning or end of two-part compounds, new more complex compounds can be formed: *puutalorakentaminen* (puu+talo+rakentaminen, 'building of wooden houses'), *ihmisoikeusloukkaus* (ihmis+oikeus+loukkaus, 'violation of human right(s)').

There is no clear upper limit to recursive concatenation of constituents in compound creation, but compounds with five constituents are already on the upper limit of

[3] A hyphen is used to separate compound constituents in several cases for clarity This happens, for example, when the constituents in a compound have adjacent same vowels, i.e. a hiatus between, e.g. *kilpa-auto ('racing car')* [10:401]. Also abbreviations, numbers and special signs are written with hyphen in a compound.

concatenation in frequency [10: 405]. [15] has analyzed compounds that have four or more constituents and in her newspaper data of ca. 13,000 tokens about 84 % of the long compounds consist of four constituents and ca. 12 % have five constituents. In Tyysteri's [12] data of new compounds from years 2000–2009 (over 28,000 tokens), two constituent compounds are the norm: 83.6 %; three constituent compounds form 15.5 % of the data and four constituent compounds only 0.9 %. Longer compounds are almost negligent in the data [12].

2.2. Types of Compounds

The largest modern Finnish grammar, *Iso suomen kielioppi* [10], describes compound forming of Finnish in detail. Here, we concentrate only on the basics of compounds for our purpose and do not try to cover all the varieties, as part of the compound classes are rare[4]. In addition to [10], [11] and [12] have been useful sources in details of Finnish compound forming.

The most common type of compound is a determinative compound (aka. a subordinate compound). In a determinative compound, the last constituent of the compound specifies the basic meaning of the word and is the head of the whole construction, whereas the first constituent modifies the whole. The meaning of the whole is more or less the sum of the constituents, i.e. the meaning is transparent and compositional. *Puutalo* is a type of a house, where *puu* ('wood') modifies the basic meaning of *talo* ('house'). In determinative compounds, constituents have thus a semantically non-symmetrical relationship with each other. The compound denotes a subordinate concept to the head of the compound [11].

In some determinative compounds, the meaning of the compound cannot easily be deduced from the sum of the meanings of the compound constituents. Examples of such compounds are e.g. *tietokone* (tieto+kone, 'computer', literally 'knowledge machine') and *potkuhousut* (potku+housut, 'playsuit' (for a baby), literally 'kick trousers') [1]. Such items are many times referred to as "lexicalized compounds", and their meaning is non-transparent.

Copulative or co-ordinate compounds are the second main compound type [10], [11]. They consist of two or more compound constituents, which are in a symmetrical relationship with each other. Constituents of copulative compounds represent the same part of speech and their relationship is semantically additive. A hyphen is often used to separate the constituents. Examples of copulative compounds are e.g., *kanttori-urkuri* ('cantor and organist') and *parturi-kampaamo* ('barber and hairdresser').

Out of these two basic compound types, determinative compounds are far more common than copulative compounds. On the basis of the 94,110 word basic lexicon of Finnish, the Kotus lexicon[5], we would estimate that whereas determinative compounds are counted in tens of thousands in a basic lexicon, copulative compounds are counted in about 30–50 in the same lexicon. Lantee's [15] compound analysis data consists of ca. 13,000 compounds. About 85 % of tokens in the data are determinative compounds**.** The rest 15 % are either copulative compounds or determinative compounds that have as a determinative part either a phrase or a copulative compound.

---

[4] Usually three types are distinguished: determinative, copulative and appositive [10-11, 13].

[5] http://kaino.kotus.fi/sanat/nykysuomi/

2.3. Share of Compounds

The total share of compounds has been counted for the largest Finnish dictionaries. [14] states that the still largest but nowadays slightly outdated Finnish dictionary, *Nykysuomen sanakirja,* has about 65 % of compounds out of its ca. 201,000 lexemes. The more modern dictionary, *Perussanakirja*, has 94,100 lexemes out of which 52,269 (55.5 %) are compounds according to [16].

The number of compounds in dictionaries is one aspect of productivity of compounds in Finnish, another is their frequency in texts and speech. We estimated this with analyses of large corpora with an automatic morphological analyzer. The largest data we had available were Europarl's Finnish data v.7[6] with ca. 31.95 million words, Open subtitle corpus [7] with ca. 144.48 million Finnish words, and The Finnish Parliamentary data with ca. 57.32 million words [8]. Out of these, Open subtitles represents spoken language data, although it is slightly artificial.

We ran the texts through morphological analyzer [17]. Europarl v7 had 4,125,947 (12.9 %) unique compounds in Omorfi's [17] analysis and Open subtitles 5,845,351 (4.1 %). The Finnish Parliamentary data had 7,692,148 (13.4 %) unique compounds. In the analysis of [15], the ca. 31,270,992 million token Helsingin Sanomat 2000–2001 newspaper corpus contained about 2.5 million compounds, which is 8 % out of the total words. These figures are similar to the older data of [18]: they had a 3.8 % share of compounds in speech and 14.6 % in texts.

2.4. Structure of Compounds

Compounding is based on the concatenation of two or more words together. A two-part simple determinative compound consists of two simple words, and its structure is straightforward: *kivi+talo*. However, more complex compounds can consist of other compounds or word combinations. These complex compounds have a layered structure where relations of the constituents are hierarchic. According to [10], multiple constituents are more common for the first constituents of a compound. As examples, [10] list the multipart compounds shown in Table 1.

Table 1. Multipart compounds with hierarchic structural analysis: Det and Cop refer to determinative and copulative compounds

1)	[isän+maan]+rakkaus	'love for homeland'	Det
2)	ala+[ikä+raja]	'minimum age limit'	Det
3)	[maa+talous]+[oppi+laitos]	'rural institute'	Det
4)	[[aika+kaus]+lehti]+katsaus	'survey of periodicals'	Det
5)	sähkö+[[parran+ajo]+kone]	'(electric) razor'	Det
6)	[palo+päällikkö]+[[väestön+suojelu]+ohjaaja]	'fire chief and civil defense instructor'	Cop

The problems that multipart determinative compounds bring to automatic analysis can be further illuminated with examples from [15]. Three-constituent compounds can in principle be decomposed in two ways:

[6] https://www.statmt.org/europarl/
[7] http://opus.nlpl.eu/OpenSubtitles.php

7) [koira+valjakko]+kilpailut 'dog sled race'
8) kirjasto+[tieto+kanta] 'library database'

It seems that the first type is more common, but the latter one is also frequent in our data. Four constituent compounds are still more complex, as they can be decomposed in three ways. Lantee [15: 30] gives the following examples:

9) [arvo+paperi]+markkina]+laki 'securities market law'
10) [mäki+hyppy]+[viikon+loppu] ' ski-jump weekend'
11) kesä+[kauppa+[korkea+koulu]] ' summer school of economics'

These decompositions can still be decomposed further, which increases the number of possible combinatorial analyses to five. If the compound has five or more parts, possibilities for analyses would increase.

3. Marking of Compounds in FiSTComp

3.1. An Initial Strategy

We have seen so far that compound structures may be complicated and a certain type of compound, i.e. the determinative compound, is the most frequent one. The number of constituents in a determinative compound is in theory unlimited, but two- and three-constituent determinative compounds are the most frequent ones. Four-constituent compounds occur to some extent too, but from five constituents on the frequencies are negligible [10], [12], [15]. Thus, we will concentrate only on compounds that have maximally four constituents in our compound tagging strategy.

In this paper, we use two different approaches to morphological analysis of words for FiSTComp: FinnPos [4], and Turku Dependency Parser [5–6], UD1. Both these tools disambiguate multiple morphological interpretations of words and provide word boundary markings for compounds, but the details and granularity of constituent decomposition vary. FinnPos's style in compound splitting could be called cautious whereas UD1 is more prolific in splitting.

Most of the compounds – easily up to 85 % in different data – consist of two constituents, and these are easy to handle: FiSTComp tries first to analyze all split compounds as wholes, and if the whole is found in the lexicon, the program stops analysis and returns the result found in the lexicon. If the whole is not found in the lexicon, the two constituents are sought for in the lexicon and tagged, if possible.

As example analyses, we use words *pää+ministeri* ('prime minister' sg. nom.) and *oppositio+puolue* ('opposition party', sg. nom.). FiSTComp tries first to find the two-constituent word as a whole in the lexicon, and only after failure of that, the constituents *pää* and *ministeri* or *oppositio* and *puolue* would be sought for. Results of the analysis after FiSTComp look like this:

12) pääministeri Noun G1.1/S2
13) puolue Noun G1.2/S5+ oppositio Noun G1.2/S5+ COMP1

The first compound has been found in the semantic lexicon, and thus its meaning is one tag for the whole; the slash in the tag shows that the word belongs to two semantic categories. The second compound was not in the lexicon, and it is given the meaning of its constituents, *party* and *opposition*. The main constituent is presented first in the output of FiSTComp to mark its saliency for the meaning of the whole. Tag COMP1 is also attached to analyses where constituents of a two-part compound have been sought for in the lexicon.

In our test data of 2.17 million tokens, FinnPos analyses 91,139 tokens as compounds. Out of these 83,419 (91.5 %) are split to two constituents. In the same data, UD1 analyses 265,437 tokens as compounds, and out of these 227,549 (85.7 %) have two constituents. Analyses of UD1 seem to be far more useful for FiSTComp: 121,965 (45.9 %) of the marked compounds could be analyzed as wholes by FiSTComp, and the rest, 143,472 (54.1 %) were given a constituent analysis. This implies that UD1's compound splitting performs well. In comparison, out of FinnPos's compound analyses only about 3 % could be analyzed as wholes by FiSTComp.

3.2. A Refined Strategy

For two constituent compounds, the analysis is straightforward, but for more complex compounds other solutions are needed. The simple solution, treating the last constituent of the compound as the main part of the compound, works in many cases, but there are also lots of cases where the main internal word boundary should be set differently. The following examples depict this. In examples 14–15, the main constituent of the compound consists of the last two constituents, and the first constituent is a modifier for the whole.

14) aalto+[sulku+merkki] (‘curly bracket’)
15) aamu+[jumalan+palvelus] (‘morning worship’)

In example 16, however, the first two constituents should be kept together:

16) [aika+kaus]+julkaisu (‘magazine’)

This applies to four-constituent compounds, too. The last constituent may be sometimes the main part as in example 17, but many times, the last two constituents form the main part of the compound: this is the case with the examples 18–19.

17) [aika+kaus+lehti]+artikkeli (‘magazine article’)
18) [asian+ajo][valta+kirja] (‘power of attorney’)
19) [elo+hopea][lämpö+mittari](‘mercury thermometer’)

For three- and four-part compounds, a more elaborate initial strategy could be like this: as earlier, the whole word is first sought for in the lexicon. If it is not found, then two splittings are tried in this order for three-part compounds, keeping in mind that Finnish compounds are right-headed [13]:

1/2+3	try to find the longest possible end match first
1+2/3	if the longer end match does not succeed, try to find the last constituent first and then the initial combined part

If these do not bring results, then all the constituents need to be sought for separately in the lexicon. The same kind of strategy applies to four-part compounds, although this does not cover all the possibilities [15].

1+2/3+4 longest plausible match
1+2+3/4 the last constituent and the beginning as a whole

3.3. Results and Problems

We saw earlier that compound splitting with FinnPos was not very useful for FiSTComp even with two constituent compounds, and thus we use only results of UD1's compound analyses with FiSTComp in the analysis of the 2.17 million token corpus.

We compared FiSTComp's analysis with basic FiST which has no elaborated compound handling. FiSTComp achieved lexical coverage of 93.4 % with the corpus, whereas FiST achieved lexical coverage of 86.8 %. The gain was thus quite clear: 6.6 % units. As the only difference between the tagger versions is compound handling, splitting of compounds improves lexical coverage of the tagger significantly if the morphological analysis phase performs well.

UD1 marked 265,437 (12.23 %) words as compounds in the data. 85.7 % of these were marked as two-constituent compounds, 12.9 % as three-constituent compounds and 1.3 % had four constituents. The data had also 106 five-constituent and eight six constituent compounds in UD1's analysis.

There are some clear problems in compound analysis that rely on a morphological analyzer which is not integrated with the semantic tagger. First and foremost is the case when the morphological component does not produce good enough boundary markings for compounds, which seemed to be the case with FinnPos. Another problem is that morphological analyzers may, e.g., produce inaccurate analyses for some parts of the compounds. These include category changes in word class, e.g. from deverbal nouns to verbs: noun *tuottavuus+ohjelma* ('productivity program') is analyzed as *tuottaa+ohjelma* ('to produce + program'). Many times, the tags in semantic categories of the constituents are right even in these cases, but, anyhow, a whole word analysis would be better. Base forming of compound constituents may also make the analyzed word impossible to find in the lexicon as a whole. *Veron+kevennys (*'tax cut', the first constituent in sg. gen*.),* e.g., is analyzed as *vero+kevennys,* where the first constituent is lemmatized to sg. nom, and thus the word could not be found as a whole in the lexicon even if it is there. The form of the first constituent of a Finnish compound is most of the times sg. nom., but also genitive forms are common. Also, clear misanalyses occur in the morphological analysis phase: *tulevaisuus+valiokunta* ('future committee') becomes *tulla+valiokunta* ('come + committee') which blurs the meaning of the compound.

4. Conclusion

This paper has described an initial version of FiSTComp, a semantic tagger for Finnish with advanced compound analysis via compound constituents. As was shown, constituent analysis improves the lexical coverage of the tagger markedly – with 6.6 % units – in comparison to a tagger version with no compound constituent analysis. In a corpus with about 227,000 found compounds, FiSTComp was able to give some level of constituent analysis to 54 % of the compounds which otherwise would have been left unanalyzed. FiSTComp thus improves compound handling, but further improvement is needed. Compounds are a multifaceted phenomenon, and so far we have scratched the surface of their structural composition. The question of representing the analysis results from a lexical semantic point of view, for example, would need separate discussion, which needs to be left for later development.

References

[1] Löfberg. L. Creating large semantic lexical resources for the Finnish language. Lancaster University, 2017. 422 pages. https://doi.org/10.17635/lancaster/thesis/3

[2] Kettunen, K. FiST – towards a Free Semantic Tagger of Modern Standard Finnish. In: Proceedings of the Fifth International Workshop on Computational Linguistics for Uralic Languages, 2019. p. 66-76.

[3] Kettunen, K. Nykysuomen automaattisesta semanttisesta merkitsemisestä. In: Jantunen, JH, Brunni, S, Kunnas, N, Palviainen, S, Västi, K, editors. Proceedings of The Research Data And Humanities (Rdhum) 2019 Conference: Data, Methods And Tools. Studia humaniora ouluensia. p. 215–228.

[4] Silfverberg, M, Ruokolainen, T, Lindén, K, Kurimo, M. FinnPos: an open-source morphological tagging and lemmatization toolkit for Finnish. Lang Resources & Evaluation 2016 50: 863–878.

[5] Haverinen, K, Nyblom, J, Viljanen, T, Laippala, V, Kohonen, S, Missilä, A, Ojala, S, Salakoski, T, Ginter, F. Building the essential resources for Finnish: the Turku Dependency Treebank. Lang Resources & Evaluation 2014 48: 493–531.

[6] Pyysalo, S, Kanerva, J, Missilä, A, Laippala, V, Ginter, F. Universal Dependencies for Finnish. In: Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA 2015, May 11-13, 2015, Vilnius, Lithuania. p. 163-172.

[7] Kettunen, K, LaMela M. Digging Deeper into the Finnish Parliamentary Protocols – Using a Lexical Semantic Tagger for Studying Meaning Change of Everyman's Rights (allemansrätten). In: Reinsone, S, Skadiņa, I, Baklāne, A, Daugavietis, J, editors. Proceedings of the Digital Humanities in the Nordic Countries 5th Conference Riga, Latvia, October 21-23, 2020. p. 63-80. http://ceur-ws.org/Vol-2612/

[8] Rauh, C, De Wilde, Pieter, Schwalbach, J. The ParlSpeech data set: Annotated full-text vectors of 3.9 million plenary speeches in the key legislative chambers of seven European states. 2017. https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/E4RSP9

[9] Finkbeiner R, Schlücker B. Compounds and multi-word expressions in the languages of Europe. In: Schlücker, B, editor. Compounds and Multi-Word Expressions. De Gruyter. 2018. p. 1–43.

[10] Hakulinen, A et al. Iso suomen kielioppi. Helsinki: Kotimaisten kielten tutkimuskeskus, 2008. http://scripta.kotus.fi/visk/etusivu.php

[11] Hyvärinen, I. Compounds and multi-word expressions in Finnish. In Schlücker, B, editor, Compounds and Multi-Word Expressions. De Gruyter. 2018. p. 307–337.

[12] Tyysteri, L. Aamiaiskahvilasta ötökkätarjontaan. Suomen kirjoitetun yleiskielen morfosyntaktisten yhdyssanarakenteiden produktiivisuus. 2015. Annales Universitatis Turkuensis C 408.

[13] Niemi, J. Compounds in Finnish. Lingue e linguaggio 2009 VIII2: 237-256.

[14] Saukkonen, P. Suomen kielen yhdyssanojen rakenne. In: Commentationes Fenno-Ugricae in honorem Erkki Itkonen sexagenarii die XXVI mensis aprilis anno MCMLXXIII: Erkki Itkonen 60 v. 1973. SUST. 150. Helsinki: SUS. p. 332–339.

[15] Lantee, A. Pitkät yhdyssanat Helsingin Sanomissa. M.A. thesis, Kieli-ja käännöstieteiden laitos - School of Modern Languages and Translation Studies. 2010. University of Tampere.

[16] Nikolaev, A, Niemi, J. Suomen nominien taivutusjärjestelmän produktiivisuuden indekseistä. Virittäjä 2008 112(4): 518–544.

[17] Pirinen, TA. Development and Use of Computational Morphology of Finnish in the Open Source and Open Science Era: Notes on Experiences with Omorfi Development. SKY Journal of Linguistics 2015 28: 381–393.

[18] Pajunen, A, Palomäki, U. Tilastotietoja suomen kielen rakenteesta, 1 / Frequency Analysis of Spoken and Written Discourse in Finnish. 1984. Helsinki: KOTUS (Research Institute for the Languages of Finland).

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200618

Lexicon-Enhanced Neural Lemmatization for Estonian

Kirill MILINTSEVICH [1] and Kairit SIRTS
Institute of Computer Science, University of Tartu, Tartu, Estonia

Abstract. We propose a novel approach for Estonian lemmatization that enriches the seq2seq neural lemmatization model with lemma candidates generated by the rule-based VABAMORF morphological analyser. In this way, the neural decoder can benefit from the additional input considering that it has a high likelihood of including the correct lemma. We develop our model by stacking two interconnected layers of attention in the decoder—one attending to the input word and another to the candidates obtained from the morphological analyser. We show that the lexicon-enhanced model achieves statistically significant improvements in lemmatization compared to baseline models not utilizing additional lemma information and achieves a new best result on lemmatization on the Estonian UD test set.

Keywords. Lemmatization, seq2seq model, Estonian, morphological analyser

1. Introduction

High quality lemmatization can drastically improve the quality of other more high-level NLP tasks such as information extraction [1] or named entity recognition [2]. This is even more important for languages with a rich word inflection paradigm like Estonian. Mapping word forms to their lemma greatly reduces the size of the training vocabulary and improves the learning capabilities of the models trained on such data.

State-of-the-art lemmatization systems are nowadays based on sequence-to-sequence neural architectures that can capture contextual word similarities better than purely statistical models. They are also not dependent on fixed lexicons and rule-based systems. Neural network based lemmatization systems have already achieved very high results. For instance, the best Estonian lemmatizer at the CONLL 2018 Shared Task achieved the accuracy of 96.57 %,[2] leaving little room for improvement. However, we propose that it is still possible to reduce the errors even further by utilizing existing linguistic resources such as rule-based systems and lexicons. Our proposal enables to unite the strengths of both approaches—the neural representation learning and the symbolic rules of otherwise hard to predict words.

VABAMORF [3] is a rule-based Estonian morphological analyser that for each word generates all possible morphological analyses consisting of lemma, part-of-speech and morphological features. We propose to integrate the lemmas generated by the

[1]Corresponding Author: Kirill Milintsevich; E-mail: kirill.milintsevich@ut.ee
[2]`https://universaldependencies.org/conll18/results-lemmas.html`

VABAMORF analyser directly into the neural lemmatizer, thus enabling the model to rely on both sources of information—the regularities learned by the neural model and the candidates proposed by the analyser. Our model encodes both the input word and the lemma candidates generated by VABAMORF, and passes both representations into a decoder. The decoder then learns to benefit from the second input by passing it through an additional layer of attention.

We conduct experiments on the Estonian Universal Dependencies (UD) dataset and show that our model with additional VABAMORF inputs achieves significantly higher results compared to the baseline model trained only on the UD training set. Moreover, our VABAMORF enhanced model also surpasses the best Estonian lemmatization result of the CoNLL 2018 Shared Task.

2. Previous Work on Estonian Lemmatization

The previous work on Estonian lemmatization derives from two sources. The first is the rule-based VABAMORF morphological analyzer [3] that in addition to POS tags and morphological features also produces lemmas. The lemmatization module is based on a lexicon which, according to Kaalep and Vaino [3], is estimated to cover ca. 97 % of tokens in any Estonian text. The system also has a guesser module that attempts to generate lemmas for unknown words. Although VABAMORF also features a statistical disambiguator, approximately 13.5 % of all words are expected to remain ambiguous for various reasons [3]. Some of these ambiguities are solved by considering the wider textual context [4]. According to our knowledge, there is only one previous work that has evaluated the performance of VABAMORF lemmatizer on common benchmark UD datasets [5]. According to Leman [5], the lemmatization accuracy of the VABAMORF system on the UD v2.3 test set is about 95.2 %.

The second line of work involving lemmatizing Estonian originates from the CoNLL 2017 and 2018 Shared Tasks [6, 7] and the SIGMORPHON 2019 Shared Task [8]. The most widely known systems from these competitions are the Stanford Stanza [9], UdPipe [10] and TurkuNLP [11] neural lemmatizers. These systems also exemplify the two main approaches used in neural lemmatization systems. Both Stanza and TurkuNLP are based on the sequence-to-sequence architecture, where the lemma for a word is generated character by character. The UDPipe model, on the other hand, utilizes a classification approach. Based on training set, a set of rules for transforming a word into its lemma are extracted. On the current Estonian UD v2.5 test set, the lemmatization results are 96.05 % for Stanza[3] and 90.6 % for UDPipe.[4] The TurkuNLP achieved the best performance on Estonian UDv2.2 dataset in the CoNLL-2018 Shared Task with 96.57 %. However, it is unknown how well it performs on the UD v2.5 test set.

3. Lexicon-Enhanced Lemmatization Model

The core of our model is the Stanza lemmatizer [9] which is a sequence-to-sequence encoder-decoder model. Stanza takes the character-level word representation and the

[3] `https://stanfordnlp.github.io/stanza/performance.html`

[4] `http://ufal.mff.cuni.cz/udpipe/models`

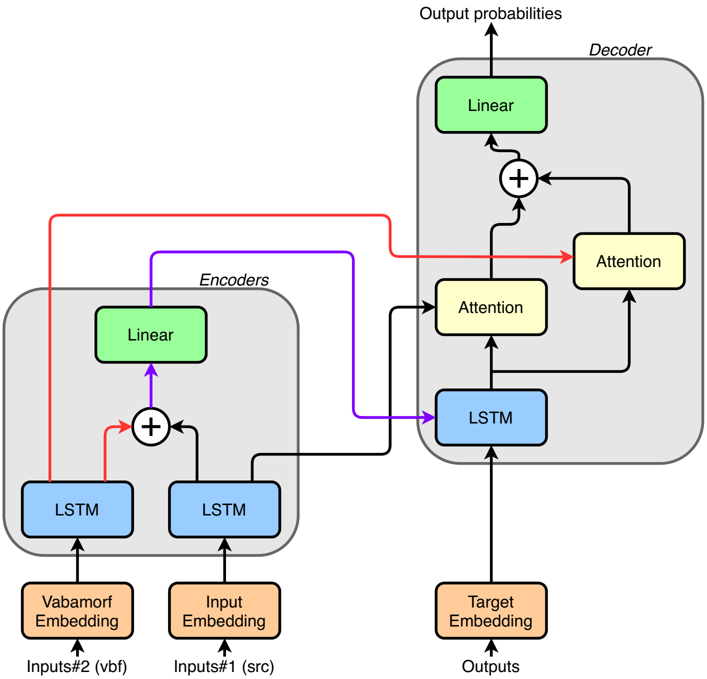


Figure 1. Dual encoder architecture for the lexicon-enhanced lemmatizer

POS tag embedding as input and processes them with a bidirectional LSTM encoder. Then, it passes the encoder outputs to a LSTM decoder. The decoder applies a soft dot attention layer after every LSTM step. Finally, the output is constructed with the greedy search over the decoder outputs.

The overall architecture of our model is presented on **Figure 1**, which shows the changes made to the Stanza lemmatizer. In particular, we add another encoder that takes the candidates generated by the VABAMORF analyser as input. The outputs of both encoders are combined with a linear layer and fed to the decoder. Moreover, we add another attention layer to the decoder that attends to the VABAMORF candidates encoded by the second encoder. This helps the model to better choose the appropriate features from both encoders. The outputs of both attention layers are finally combined with a linear layer.

Finally, in addition to POS tag, we also add morphological features to the input of the first encoder. We will show that it improves the lemmatization for Estonian and can potentially be helpful for other highly inflectional languages as well.

4. Experimental Setup

For training and testing the models we use the Estonian UD v2.5 treebank [12]. The treebank is in the CoNLL-U format and conforms to the Universal Dependencies project annotation standards [13]. It contains 437 769 tokens annotated with POS tags, morphological features, lemmas, and syntactic information. The treebank is based on the Estonian Dependency Treebank (EDT), created at the University of Tartu.

Table 1. Experimental results for our Vabamorf-enhanced model, the baseline with the empty second encode (Non-Enhanced), and the Stanza baseline. The OOV column shows the accuracy on the out-of-vocabulary words only

Rank	Model	Accuracy	
		All words	OOV
1	LexEnLem (Vabamorf-enhanced)	96.87 ± 0.17	88.64
2	LexEnLem (Non-Enhanced)	96.36 ± 0.20	86.16
3	Stanza	96.11 ± 0.20	83.86

All models were evaluated on the Estonian UD test set with POS tags and morphological features predicted by the Stanza pipeline [9] that achieved 94.54 F1-score for all tags (upos, xpos, and feats). The predictions for each run were ranked and tested with paired bootstrap resampling [14]. Each score is accompanied by 95 % confidence intervals obtained with 10,000 resamples. The p-value shows if the difference with the next system is statistically significant. If the p-value < 0.05, we rank the system higher than the following one. As baseline, we use the default Stanza model for Estonian that has been trained on the same UD v2.5 dataset.

5. Results

We conducted experiments with our LEXicon-ENhanced LEMmatization (LexEnLem) model in two different settings that differ in the input to the second encoder. In the baseline version (LexEnLem Non-Enhanced), the second encoder receives no input. The Vabamorf-Enhanced version receives via the second encoder all distinct lemma candidates generated by the Vabamorf morphological analyser. If there are several lemmas then they are simply concatenated.

As can be seen from the **Table 1**, the Stanza baseline model was ranked the least. Our Non-Enhanced model providing better results can be explained by the addition of morphological features to the input of the first encoder. The enhanced model with the second encoder utilizing the Vabamorf predictions outperforms both baselines. According to the bootstrap test, the differences between all models are statistically significant at the level of $p < 0.05$. The difference between the Vabamorf-enhanced LexEnLem and the Stanza baseline are especially visible when inspecting the accuracy of the out-of-vocabulary words. Overall, our Vabamorf-Enhanced lemmatization model achieves a new best result on the Estonian UD test set.

5.1. The Effect of Word Formation Symbols

Lemmas in Estonian EDT are additionally annotated with the word formation information, specifically compounding and morphological derivation. For example, the lemma for the word *ostusedelisse* (in the shopping list) is *ostu_sedel*. The underscore in the lemma shows that the word is compound and in fact consists of two words: *ostu* (shopping) and *sedel* (a list).

We analyzed the errors made by all models and found that indeed, many errors are related to the misplacement of the word formation symbols. **Table 2** shows the distribution

Table 2. Division of different types of errors made by the models

Model	Missing		Misplaced		Misc	Total
	COM	DER	COM	DER		
LEXENLEM (Vabamorf)	82	225	197	124	890	1518
LEXENLEM (Non-Enhanced)	107	211	166	120	1163	1767
Stanza	146	192	150	82	1316	1886

of the errors. In the table, **COM** signifies the symbol "_" separating the compound parts in a compound word, **DER** denotes the the derivational symbols "+" and "=". **Missing COM** and **Missing DER** stand for the errors when the respective symbol is present in the gold lemma but not present in the predicted lemma and if removed, the prediction is correct (e.g. correct: *"laua_naaber"*; predicted: *"lauanaaber"*). **Misplaced COM** and **Misplaced DER** stand for the errors when the respective symbol is present in both gold and predicted lemmas but is misplaced in the predicted (e.g. correct: *"ostu_sedel"*; predicted: *"ostus_edel"*) or it is not present in the gold but is present in the predicted and if removed, the prediction is correct (e.g. correct: *"seostamine"*; predicted: *"seosta=mine"*). **Misc** stands for all the other errors. As it can be seen, the number of errors related to word formation annotation symbols is roughly the same across all models, while the best performing model reduces the number of MISC errors.

Thus, we created another version of the data where all word formation symbols were removed, and trained our models on this modified dataset. **Table 3** shows the results of these experiments. The Stanza baseline results are obtained by removing the word formation symbols before evaluation. All the models retain their rankings and show improvement in accuracy for more than 1 % for all words and more than 5.5 % for out-of-vocabulary words.

Table 3. Experimental results for our Vabamorf-enhanced model, the baseline with the empty second encode (Non-Enhanced), and the Stanza baseline without word formation symbols. The OOV column shows the accuracy on the out-of-vocabulary words only

Rank	Model	Accuracy	
		All words	OOV
1	LEXENLEM (Vabamorf-enhanced)	98.11 ± 0.13	94.14
2	LEXENLEM (Non-Enhanced)	97.66 ± 0.15	91.88
3	Stanza	97.29 ± 0.15	89.99

5.2. The Effect of the Vabamorf Settings

VABAMORF in its basic setting is a morphological analyser that for each word returns all possible morphological analyses, including lemmas. In addition to this basic setting, there are several additional modules that can be applied. The disambiguator module uses a statistical HMM model to select the most likely analysis for each word in context. The proper name module attempts to recognise proper names. Finally, the guesser module attempts to guess the lemmas for words that are not present in the system's dictionaries.

Table 4. Results of the VABAMORF settings ablation experiments. Basic is the VABAMORF system without additional modules; +PN adds the proper name module, +DIS adds the disambiguation module and +Guesser adds the guesser module

Setting	Dev Accuracy	Test Accuracy	Test OOV
Basic	**99.14**	**98.11**	**94.14**
+PN	99.09	98.04	93.82
+DIS	99.04	97.99	93.79
+PN +DIS	99.05	97.95	93.46
+PN +DIS +Guesser	99.05	97.95	93.46

The application of all these modules can have an effect on our lexicon-enhanced model as well. Thus, we performed a set of ablation experiments with the different settings of the Vabamorf system. All these experiments were done on the dataset with the word formation symbols removed.

Table 4 presents the results on both development and test set. On both evaluation sets, the basic VABAMORF without additional modules performs the best, suggesting that giving more ambiguous input to the second encoder improves our model. This may be because increasing the ambiguity of the morphological analyses raises the likelihood that the additional input includes the correct lemma. When the guesser module is turned on, VABAMORF attempts to predict the lemma for unknown words using an inferior algorithm while without the guesser it skips unknown words and thus leaves the prediction task to our model.

5.3. *The Effect of the Vabamorf Candidates*

The central idea of the proposed approach is to use an external system or lexicon to influence the lemmatization model towards correct predictions if the external systems knows the correct lemma. To analyze if our model succeeded in that, we analyzed the errors made by the Vabamorf-Enhanced model and the baseline Non-Enhanced model that did not receive any input into the second encoder. **Figure 2** demonstrates the effect of the VABAMORF candidates on the model's performance. As can be seen from the graph, in the majority of the cases when the Vabamorf-enhanced model predicts the correct lemma and the Not-Enhanced model's prediction is wrong (the first column), at least one of the candidates passed to the Vabamorf-enhanced model is correct, suggesting that the additional input indeed influenced our model towards making a correct prediction.

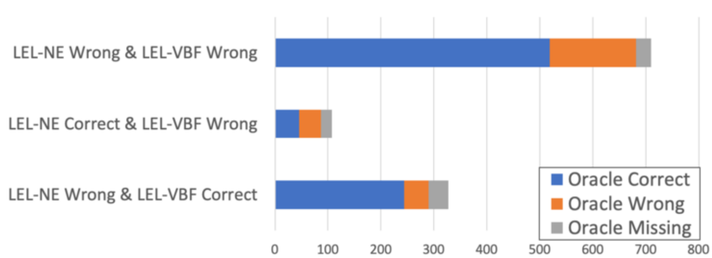


Figure 2. Comparison of the errors made by theLEXENLEM Not-Enhanced (LEL-NE) and the LEXENLEM Vabamorf-enhanced (LEL-VBF) models. Oracle stands for all the candidates produced by VABAMORF

The third column shows the number of cases when both models predicted wrong lemmas even though the Vabamorf-enhanced model received the correct candidate during the inference. This was mostly caused by the wrong capitalization of loan words (e.g. Jazz, Rock), adding -i ending to the foreign names (e.g. *Johni, *Pauli, *Kölni), and wrong disambiguation of *see* and *tema* which have the same plural form *neid*. Another reason for this type of errors is incorrectly predicted POS and/or morphological tags. For example, the form *praeme_Verb.Present.1per.Plur* was wrongly tagged as **praeme_Noun.Part.Plur* and thus lemmatized as **praed* while the correct lemma would be the infinitive *praadima*. This can signify that the information encoded in the morphological tags has more weight than the additional lemma candidates provided to the model.

There are also a small number of cases where the Non-Enhanced model predicts correctly while the Vabamorf-Enhanced model generates a wrong lemma (the middle column), even though there is a correct lemma in the VABAMORF candidates. These cases are also mostly related to the erroneous POS and morphological tags.

6. Conclusion

We presented a novel approach to Estonian lemmatization that enables a seq2seq encoder-decoder model to benefit from the external VABAMORF morphological analyser. Our hybrid model achieves a new state-of-the-art results in lemmatization on the Estonian UD test set with 96.87 % when the word formation symbols are considered and 98.11 % with word formation symbols removed. We also analyzed the error patterns of our model and found that many errors are related to incorrect POS and morphological predictions which influence the model to generate incorrect lemmas even when the correct lemma candidate is provided. This suggests that in order for the Vabamorf-enhanced model to fully gain from the extra input, it might be beneficial to downweight the relative importance of the POS and morphological info when the external candidates are provided.

References

[1] Kanis J, Skorkovská L. Comparison of different lemmatization approaches through the means of information retrieval performance. In: Proceedings of TSD 2010; 2010. p. 93–100.

[2] Konkol M, Konopík M. Named entity recognition for highly inflectional languages: effects of various lemmatization and stemming approaches. In: Proceedings of TSD 2014; 2014. p. 267–274.

[3] Kaalep HJ, Vaino T. Complete morphological analysis in the linguist's toolbox. Congressus Nonus Internationalis Fenno-Ugristarum Pars V. 2001:9–16.

[4] Kaalep HJ, Kirt R, Muischnek K. A trivial method for choosing the right lemma. In: Proceedings of Baltic HLT 2012. vol. 247; 2012. .

[5] Leman LK. Comparative analysis of neural Network Based Lemmatizers in the Estonian Language [Bachelor's Thesis]. University of Tartu; 2019.

[6] Zeman D, Popel M, Straka M, Hajič J, et al. CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In: Proceedings of the CoNLL 2017 Shared Task; 2017. p. 1–19.

[7] Zeman D, Hajic J, Popel M, Potthast M, Straka M, Ginter F, et al. CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. In: Proceedings of the CoNLL 2018 Shared Task; 2018. p. 1–21.

[8] McCarthy AD, Vylomova E, Wu S, Malaviya C, Wolf-Sonkin L, Nicolai G, et al. The SIGMORPHON 2019 Shared Task: Morphological Analysis in Context and Cross-Lingual Transfer for Inflection. In: Proceedings of the SIGMORPHON; 2019. p. 229–244.

[9] Qi P, Zhang Y, Zhang Y, Bolton J, Manning CD. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. arXiv preprint arXiv:200307082. 2020.

[10] Straka M, Hajic J, Straková J. UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In: Proceedings of LREC'16; 2016. p. 4290–4297.

[11] Kanerva J, Ginter F, Miekka N, Leino A, Salakoski T. Turku neural parser pipeline: An end-to-end system for the CoNLL 2018 Shared Task. In: Proceedings of the CoNLL 2018 Shared Task; 2018. p. 133–142.

[12] Muischnek K, Müürisep K, Puolakainen T, Aedmaa E, Kirt R, Särg D. Estonian dependency treebank and its annotation scheme. In: Proceedings of TLT'13; 2014. p. 285–291.

[13] Zeman D, Nivre J, Abrams M, et al. Universal Dependencies 2.5; 2019. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. Available from: `http://hdl.handle.net/11234/1-3105`.

[14] Koehn P. Statistical significance tests for machine translation evaluation. In: Proceedings of EMNLP'04; 2004. p. 388–395.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200619

Berri Corpus Manager: A Corpus Analysis Tool Using MongoDB Technology

Hugo SANJURJO-GONZÁLEZ [1]
Department of Information Technology, Electronics & Communications, University of Deusto, Spain

Abstract. Nowadays, there are many options for corpus linguistic analysis that make use of different approaches for corpus storage. There are tools based on SQL databases, dedicated implementations such as CQP/CWB and others that employ plain-text corpora. NoSQL databases have been widely used for big data, data mining and even sentiment analysis. However, as far as we can see, there is a lack of a widespread concordancer or consolidated framework that makes use of MongoDB architecture for the purposes of corpus linguistics. This paper aims to describe the architecture of a software that allows users to analyse monolingual and bilingual parallel corpora with grammatical annotation using MongoDB technology. Our premises are that MongoDB is ideal for non-structured data and provides high flexibility and scalability, so it may be also useful for corpus linguistic research. We analyse functionalities of MongoDB such as text search indexes and query format in order to examine its suitability.

Keywords. Corpus analysis tool, concordancer, NoSQL database, MongoDB, corpus linguistics

1. Introduction

There are many approaches for corpus storage regarding corpus analysis software. Sanjurjo-González [1] offers a comprehensive survey of the available corpus analysis software characterised by linguistic and technological features. In this survey, we can find tools based on SQL databases, mixed approaches that employ SQL, software that makes use of dedicated implementations for corpus indexing and querying [2] and last, software that employs plain-text corpora.

The most popular approaches for concordancers are those that are SQL-based, or dedicated implementations that use CQP/CWB [3]. For instance, SQL is used in corpus.bye.edu[2] and PELCRA[3], CQP/CWB is used in CQPWeb [4] and ACM[4] [5], among others. Although these approaches offer different performances using extreme scale cor-

[1]Corresponding Author: Hugo Sanjurjo-González; University of Deusto, Unibertsitate Etorbidea 24, 48007 Bilbao, Bizkaia, Spain; E-mail: hugo.sanjurjo@deusto.es

[2]`http://corpus.byu.edu/`

[3]`http://nkjp.uni.lodz.pl/`

[4]`https://actres.unileon.es/wordpress/?page_id=663&lang=en`

pora, there is no significant difference employing a smaller corpus for a linguistic oriented user carrying out the most common corpus linguistic operations such as KWIC (Key Word In Context) concordances, frequency lists, collocations, etc.

MongoDB[5] was officially released for production purposes in 2011 and is one of the most used NoSQL databases. It is a document-oriented database with high scalability and flexibility that stores data in BSON (binary JSON) documents. It is widely used to store data for sentiment analysis [6,7,8], data mining [9,10,11] or even big data [12,13,14]. In [15], Coole, Rayson and Mariani carried out different experiments using NoSQL databases, SQL databases and CWB/CQP with extreme scale corpus, showing that NoSQL databases like MongoDB or Cassandra are viable solutions for performing KWIC searches employing several servers. For this reason, their performance may be good enough using a large corpus even without clustering.

Consequently, this paper aims to describe the architecture of the software that allows users to analyse monolingual and bilingual corpora with grammatical annotation using MongoDB technology. Our premises are that MongoDB is ideal for non-structured information and corpus texts might be considered in this way, providing high flexibility and scalability.

The remainder of this article is organized as follows. In Section 2, we survey relevant research to the scope of the paper. We then describe the architecture and technology of the proposed software in Section 3. We introduce experiment results in Section 4. Finally, we describe conclusions in Section 5.

2. Related Work

NoSQL databases have gained popularity with emerging demands of scalable databases, mainly related to big data research [16]. This emerging trend of NoSQL paradigm for handling big data information systems should be at least considered for corpus linguistics research. Our premises are that MongoDB is ideal for non-structured information such as corpus texts thanks to its high flexibility, vertical scalability and schemmaless architecture. However, as far as we can see, there is no widespread concordancer or consolidated framework that makes use of MongoDB architecture for the purposes of corpus linguistics. In fact, some approaches make use of MongoDB for ad hoc solutions. We can mention Perkins [17] that presents a proof of concept that combines NLTK [18] with MongoDB database; Frey, Glaznieks and Stemle [19] download and merge data as a corpus zero using MongoDB; Gutierrez-Vasques, Sierra and Pomp [20] combine MongoDB with Lucene/Solr[6] to build and query their corpus; and last, Dorantes et al [21], develop a search interface using custom Python scripts and MongoDB database without describing any additional details about the implementation.

As previously mentioned, the most relevant work using MongoDB technologies in corpus analysis may be Coole, Rayson and Mariani [15]. In this case, they made their experiments in a clustering environment as a consequence of the extreme size of the corpus. In the present work, we do not use cluster components as the size of a typical corpus does not require as much power. In addition, we want to test MongoDB technology in the

[5] https://www.mongodb.com/

[6] https://lucene.apache.org/

```
{
  "corpus": "Europarl.English",
  "language": "English",
  "pos": 0,
  "parallel": ["Europarl.Spanish"],
  "texts": [
    {
      "_id": 1,
      "sentence": "Resumption of the session"
    }

  ]
}
```

Figure 1. Data model of Berri Corpus Manager

most common software development environment, composed of only one server, which does not require complex configurations or high hardware specifications.

3. Architecture and Technology

3.1. Data Model

Berri Corpus Manager employs MongoDB for corpus storage. For each corpus, we store language, size, wheter it includes any type of annotations or if it is part of a parallel corpus (Figure 1). As a consequence of the flexibility and schema-less architecture of the database, the design is very simple and can be easily modified if required. Grammatical annotations are included using prefix notation. Last, we employ sentence alignment for parallel corpora.

3.2. User Interface

User interface provides a way to query the corpus. This interface is based on Sanjurjo-González and Izquierdo [22] and offers a user-friendly layout that enables effective corpus analysis and simplifies the pattern (Figure 2) and parallel queries (Figure 3) that may or may not include annotations.

3.3. Implementation

To implement the software, we employ Python technologies, more concretely Flask[7] and PyMongo.[8]

Flask is a lightweight micro-framework based on Werkzeug web server and Jinja2 templates. It allows us to employ Python language to develop web applications. Flask maps url to different Python functions. Thus we can use Python in the server for natural

[7] https://flask.palletsprojects.com/en/1.1.x/
[8] https://docs.mongodb.com/drivers/pymongo

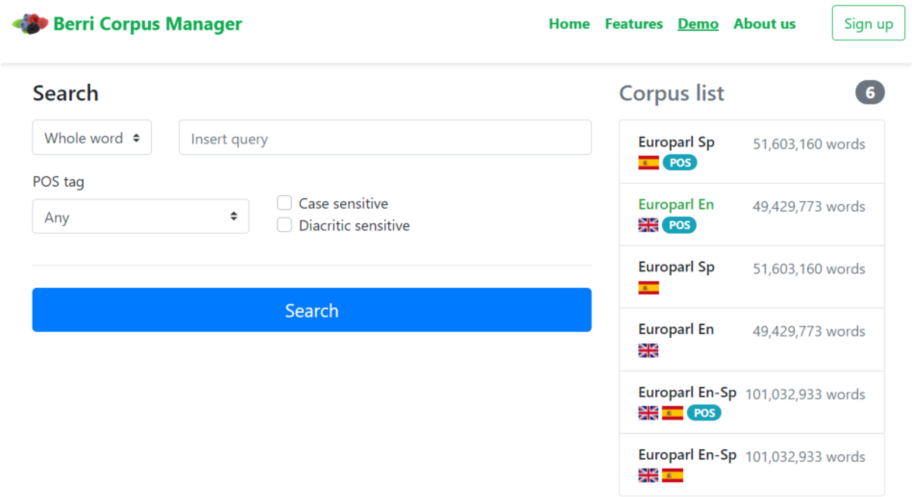


Figure 2. User interface for monolingual queries

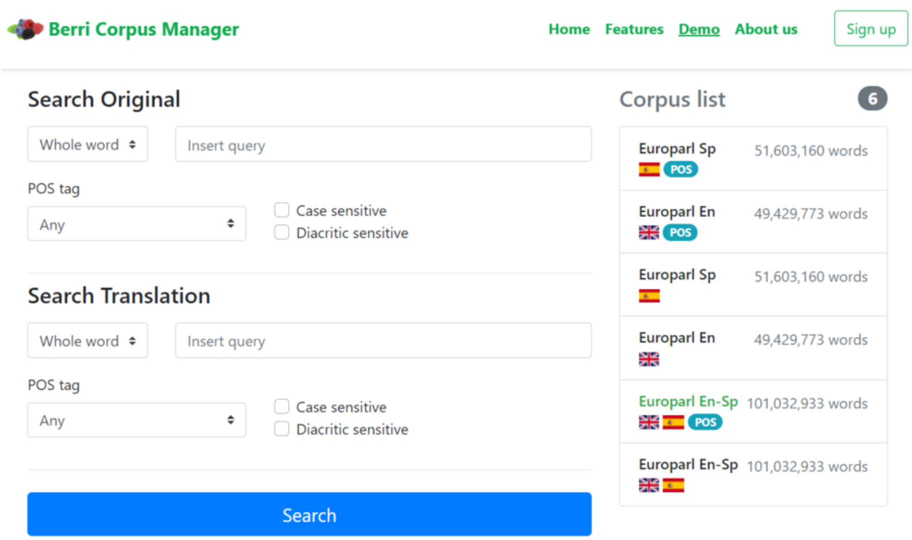


Figure 3. User interface for bilingual parallel queries

language processing tasks such as grammatical tagging, tokenisation and alignment, as well as for scripting tasks such as formatting and document corpus selection.

PyMongo is a Python distribution containing tools to work with MongoDB. By PyMongo we can easily interact with the MongoDB database using Python. As a consequence, we populate and query corpora employing available methods. It is also necessary to parse queries from the user interface to the MongoDB query language. To do that, we make use of the text search feature of MongoDB query language that includes a text index. In Figure 4, text variable refers to the query including or not including annotations.

```
data = cursor.find(
            {"$text": {"$search": ''+text+''}}, {"_id": 0, "sentence": 1}
        ).skip(int(offset)).limit(int(limit))
```

Figure 4. PyMongo code for full text search

```
data = cursor.find(
                {"sentence": {'$regex': '\w*'+text+'\w*', '$options': 'i'}},
                {"_id": 0, "sentence": 1}
            ).skip(int(offset)).limit(int(limit))
```

Figure 5. PyMongo code for regex search

As it would be expected, text search increases performance, but it also has several issues regarding the most common queries in corpus linguistics:

1. It provides language specification, however, it uses stemming and removes stop words, so it might be useless if the user queries for a particular phrase of word. For this reason, it is better not to employ language specification. Size of the database increases significantly for this selection.
2. Text indexes do not support partial word searches, so they cannot be used to search patterns such as prefixes, suffixes, or others. To overcome this issue, we must employ $regex operator, which affects negatively the performance of the query (Figure 5).
3. As a consequence of the designed data model, grammatical annotations are included using prefix notation and some queries may be affected, for instance, if a user wants to search for all the words that are proper or common names.
4. As is the case with all the databases that are not linguistically oriented, the count function returns a count of documents, in our case sentences, that would match a query. However, one sentence may include more than one instance, so we need an additional processing. To do that, we make use of the MongoDB aggregation pipeline functionality. This functionality allows us to create a framework for data aggregation modelled on the concept of data processing pipelines. It works relatively well but it takes one minute or more if the number of results is high.

Parallel queries functionality of Berri Corpus Manager supports queries in both subcorpora at the same time in order to search the correspondences. For instance, we may be interested in searching what type of expressions translate the English word ”table” into Spanish (`tabla` or `mesa`). As MongoDB is not designed for this type of parallel queries, we search in both subcorpora at the same time, and obtain sentences for which identifiers are common in both results (Figure 6).

3.4. System Architecture

Berri Corpus manager operates across a simple Model-View-Controller pattern (Figure 7): Flask (controller), MongoDB (model), Jinja2 and Bootstrap v.4 (view). Berri Corpus Manager interacts with the corpus by means of PyMongo utilities. Python scripts are

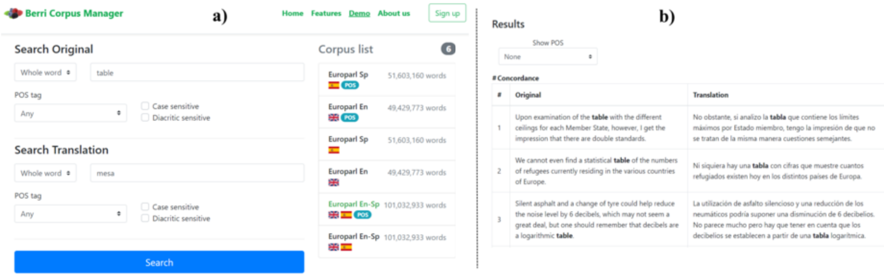


Figure 6. Results of a parallel concordance

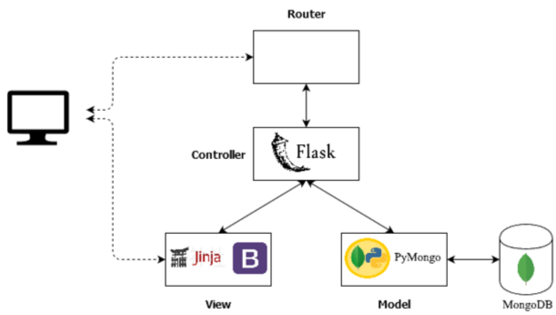


Figure 7. Architecture of Berri Corpus Manager

used to parse user queries into understandable commands using MongoDB format as well as natural language processing tasks.

4. Experiments

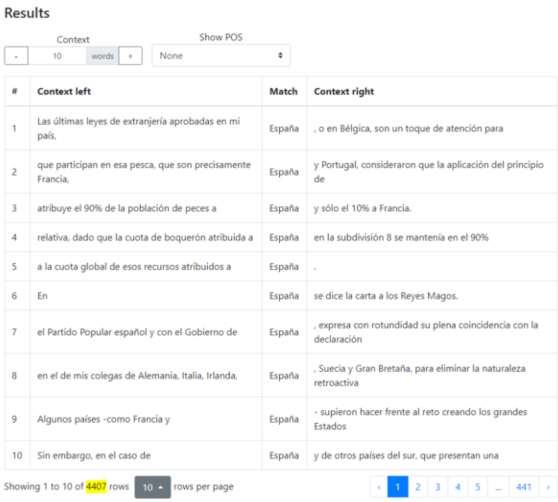
Berri Corpus Manager is able to handle monolingual and bilingual corpora with grammatical annotation. For our experiment, we selected the well-known corpus Europarl [23]. More concretely, we employed English and Spanish subcorpora that were already aligned and have a size of 49,429,773 and 51,603,160 words, respectively. Grammatical annotation has been carried out using Spacy. [9]

MongoDB presents a good performance if the search makes use of the text search index. It takes more time if it makes use of regex patterns. For instance, in our experiments, searching for all the occurrences of the word "a" takes 20 seconds using the text index and 30 seconds if it is not used.

Berri Corpus Manager employs server pagination, so there are no issues returning results. However, as it was previously mentioned, returning the total number of matches might take some time if there is a high number of results.

Last, it should be mentioned that query plan cache methods save a lot of time for recurrent operations. Therefore, high performance queries can takes less time than expected if they have been previously executed.

[9] https://spacy.io/



Results

Context: - 10 words + Show POS: None

#	Context left	Match	Context right
1	Las últimas leyes de extranjería aprobadas en mi país,	España	, o en Bélgica, son un toque de atención para
2	que participan en esa pesca, que son precisamente Francia,	España	y Portugal, consideraron que la aplicación del principio de
3	atribuye el 90% de la población de peces a	España	y sólo el 10% a Francia.
4	relativa, dado que la cuota de boquerón atribuida a	España	en la subdivisión 8 se mantenía en el 90%
5	a la cuota global de esos recursos atribuidos a	España	.
6	En	España	se dice la carta a los Reyes Magos.
7	el Partido Popular español y con el Gobierno de	España	, expresa con rotundidad su plena coincidencia con la declaración
8	en el de mis colegas de Alemania, Italia, Irlanda,	España	, Suecia y Gran Bretaña, para eliminar la naturaleza retroactiva
9	Algunos países -como Francia y	España	- supieron hacer frente al reto creando los grandes Estados
10	Sin embargo, en el caso de	España	y de otros países del sur, que presentan una

Showing 1 to 10 of 4407 rows 10 rows per page ‹ 1 2 3 4 5 ... 441 ›

Figure 8. Results of a monolingual concordance

5. Conclusions

This paper presented one of the first implementations of a concordancer using MongoDB technology. Employing lightweight technologies such as Flask makes development extremely quick. MongoDB has several advantages: it is schema-less, flexible and scalable. However, as it has not been designed for specific purposes of corpus linguistics, it also presents some issues, for instance, the unavailability of the partial search using text indexes, which has a serious effect on the queries' performance, or in returning the number of occurrences of a concordance, as it is based on the number of documents that match the query instead of the number of occurrences of the query. If these issues are solved, MongoDB can become a prominent technology for corpus linguistic software development as a consequence of its flexibility and fast deployment.

References

[1] Sanjurjo-González H. Desarrollo de un framework para el tratamiento de corpus lingüísticos [Development of a framework for corpus linguistic analysis] [Doctoral dissertation on the Internet). León, Spain: University of León; [cited 2020 Apr. 22]. Available from: http://hdl.handle.net/10612/6920

[2] McEnery T, Hardie A. Corpus Linguistics: Method, Theory and Practice. Cambridge University Press; 2012. 312p.

[3] Christ O. A modular and flexible architecture for an integrated corpus query system. In: Kiefer F, Kiss G, Pajzs J. editors. Proceedings of the 3rd International Conference on Computational Lexicography; 1994 Jul 7-10; Budapest, Hungary. Research Institute for Linguistics, Hungarian Academy of Sciences; p. 23-32.

[4] Hardie A. CQPweb—combining power, flexibility and usability in a corpus analysis tool. International Journal of Corpus Linguistics. 2012; 17(3):380-409.

[5] Sanjurjo-González H. Desarrollo de un framework para el tratamiento de corpus lingüísticos [Development of a framework for corpus linguistic analysis]. University of León; 2018. 116 p.

[6] Bai A, Hammer H, Yazidi A, Engelstad P. Constructing Sentiment Lexicons in Norwegian from a Large Text Corpus. In: Proceedings of the IEEE 17th International Conference on Computational Science and Engineering; 2014 Dec 19-21; Chengdu, China. IEEE Computer Society; p. 231-37.

[7] Gkontzis AF, Karachristos CV, Panagiotakopoulos CT, Stavropoulos EC, Verykios VS. Sentiment analysis to track emotion and polarity in student fora. In: Proceedings of the 21st Pan-Hellenic Conference on Informatics ; 2017 Sept 28-30; Larissa, Greece. The Association for Computing Machinery; p. 1-6.

[8] Ramzan M, Mehta S, Annapoorna E. Are tweets the real estimators of election results? In: Proceedings of the Tenth International Conference on Contemporary Computing (IC3); 2017 Aug 10-12; Noida, India. IEEE. p. 1-4

[9] Niekler A, Wiedemann G, Heyer G. Leipzig Corpus Miner–A Text Mining Infrastructure for Qualitative Data Analysis. In: Proceedings of the 11th Terminology and Knowledge Engineering 2014 proceedings, 39-47 2014 Jun 19-21; Berlin, Germany. Asociación Española de Terminología. p. 39-47.

[10] Santhiya K, Bhuvaneswari V. An automated MapReduce framework for crime classification of news articles using MongoDB. International Journal of Applied Engineering Research. 2018; 13(1):131-36.

[11] Staar PW, Dolfi M, Auer C, Bekas C. Corpus Conversion Service: A machine learning platform to ingest documents at scale. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2018 19-23 Aug; London, United Kingdom. The Association for Computing Machinery; p. 774-82.

[12] Rakib TBA, Soon LK. Using the Reddit Corpus for Cyberbully Detection. In: Nguyen N, Hoang D, Hong TP, Pham H, Trawiński B. editors. Proceedings of the 10th Asian Conference Intelligent Information and Database Systems (ACIIDS 2018); 2018 Mar 19-21; Dong Hoi City, Vietnam. Cham (Switzerland): Lecture Notes in Computer Science; 10751. p. 180-189.

[13] Kang Y. Park, I, Rhee J, Lee Y. MongoDB-Based Repository Design for IoT-Generated RFID/Sensor Big Data. IEEE Sensors Journal. 2016; 16(2):485-97.

[14] Plugge E. Hows D, Membrey P, Hawkins T. The Definitive Guide to MongoDB: A complete guide to dealing with Big Data using MongoDB. Apress; 2015. 336 p.

[15] Coole M, Rayson P, Mariani J. Scaling out for extreme scale corpus data. In: Proceedings of the 2015 IEEE International Conference on Big Data (Big Data). 2015 29 Oct - Nov 1; Santa Clara, CA. IEEE Computer Society. p. 1643-1649.

[16] Lee M, Jeon S, Song M. Understanding user's interests in nosql databases in stack overflow. In: Proceedings of the 7th International Conference on Emerging Databases. 2017 Aug 7-9; Busan, South Korea. Berlin (Germany): Springer. p. 128-137

[17] Perkins J. Python 3 text processing with NLTK 3 cookbook. Packt Publishing Ltd; 2014. 304 p.

[18] Bird S, Klein E. Loper E. Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit. O'Reilly Media Inc; 2009. 504 p.

[19] Frey, JC, Glaznieks A, Stemle EW. The DiDi Corpus of South Tyrolean CMC Data. In: Proceedings of the 2nd Workshop on Natural Language Processing for Computer-Mediated Communication/Social Media at GSCL2015 (NLP4CMC2015); 2005 Sep 25; Essen, Germany: German Society for Computational Linguistics & Language Technology; p. 1-6.

[20] Gutierrez-Vasques X, Sierra G, Pompa IH. Axolotl: a web accessible parallel corpus for spanish-nahuatl. In: Calzolari, N, Choukri K, Declerck T, Goggi S, Grobelnik M, Maegaard B, Mariani J, Mazo H, Moreno A, Odijk J, Piperidis S, editors. Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16) 2016 May 23-28: Portorož, Slovenia: European Language Resources Association; p. 4210-4.

[21] Dorantes A, Sierra G, Pérez TYD, Bel-Enguix G, Rosales MJ. Sociolinguistic corpus of whatsapp chats in spanish among college students. In: Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media. 2018 July 20; Melbourne, Australia: Association for Computational Linguistics; p. 1-6.

[22] Sanjurjo-González H, Izquierdo M. P-ACTRES 2.0: A parallel corpus for cross-linguistic research. In: Doval I, Sánchez-Nieto MT, editors. Parallel Corpora for Contrastive and Translation Studies: New resources and applications 90. Amsterdam/Philadelphia: John Benjamins; 2019. p. 215-31.

[23] Koehn P. Europarl: A parallel corpus for statistical machine translation. In: Proceedings of Machine Translation Summit X Vol. 5. 2005 Sep 12-15; Phuket, Thailand. Asia-Pacific Association for Machine Translation. p. 79-86.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200620

Evaluating Sentence Segmentation and Word Tokenization Systems on Estonian Web Texts

Kairit SIRTS [1] and Kairit PEEKMAN
Institute of Computer Science, University of Tartu, Estonia

Abstract. Texts obtained from web are noisy and do not necessarily follow the orthographic sentence and word boundary rules. Thus, sentence segmentation and word tokenization systems that have been developed on well-formed texts might not perform so well on unedited web texts. In this paper, we first describe the manual annotation of sentence boundaries of an Estonian web dataset and then present the evaluation results of three existing sentence segmentation and word tokenization systems on this corpus: EstNLTK, Stanza and UDPipe. While EstNLTK obtains the highest performance compared to other systems on sentence segmentation on this dataset, the sentence segmentation performance of Stanza and UDPipe remains well below the results obtained on the more well-formed Estonian UD test set.

Keywords. Sentence segmentation, sentence boundary detection, web texts, orthographic sentence boundary, syntactic sentence boundary, Estonian

1. Introduction

Sentence segmentation and word tokenization are the first pre-processing steps for most NLP tasks. Existing methods for addressing these steps are either rule-based or trained on annotated corpora. Rule-based systems for sentence and word tokenization make certain assumptions about the well-formedness of the text, for instance, that the sentences start with a capital letter and end with one of the valid sentence-final punctuation marks. Data-driven systems rely on the specifics of the annotated training corpora, most of which are also based on well-formed text genres like fiction or news articles.

Web texts, such as forum or blog posts and user commentaries, are typically unedited and thus the well-formedness of such texts can not be guaranteed. Some types of social media texts, tweets, for instance, have very distinct characteristics and thus, specific tools have been created for their token and sentence segmentation [1]. Although some examples of tokenizers and sentence segmentation systems developed specifically for other types of web and social media texts exist [2], it is generally assumed that standard tools are applicable to web texts as well.

For Estonian, both rule-based and model-based tokenizers and sentence segmentation systems are available. The rule-based system is part of the EstNLTK library [3] that gathers existing NLP tools for Estonian. Data-driven solutions are available in several

[1]Corresponding Author: Kairit Sirts; E-mail: kairit.sirts@ut.ee.

pre-trained parsing pipeline models such as Stanford Stanza [4] and UDPipe [5]. Both of these systems have achieved good results on both tokenization and sentence segmentation on Universal Dependencies (UD) Estonian treebank test set as evaluated in the CoNLL 2018 Shared task on parsing raw text [6].

In addition to the Estonian UD treebank consisting of standard texts, also a smaller Estonian Web Treebank (EWT) [7] is available, the tokenization and sentence segmentation of which are based on automatic analysis with EstNLTK. Thus, although this corpus can be used for evaluating the performance of NLP tools on web texts, the correctness of the token and sentence segmentation of this corpus cannot be guaranteed.

Our goal in this paper is to evaluate the existing sentence segmentation and word tokenization systems on Estonian web texts. For that purpose, we first manually annotated the token and sentence boundaries of the EWT corpus. In the following, we will first describe the annotation process and then present the evaluation of token and sentence segmentation performance using both EstNLTK and the pretrained Stanza and UDPipe models.

2. Related Work

Although sentence boundary detection is usually regarded as a solved problem or, at least, a problem that does not warrant much effort and attention, [8] showed that this is not the case. They performed experiments with all sentence segmentation systems available at the time on various English corpora and showed that results are far from perfect and vary depending on the domain of the corpus. Although at the time of conducting their study, models based on neural architectures were not available yet, it is highly likely that when adding the current existing neural models to the experimental plan of [8], the overall picture would remain roughly the same.

Tweets constitute a very specific domain of social media texts and thus, various attempts have been made to develop sentence segmentation and word tokenization systems specifically for handling tweets [1]. [9] describe an annotation effort to create a sentence and token boundary annotated tweet corpus in Slovene. They mention several annotation issues also relevant in our work, in particular, in relation to ambiguous sentence endings. For instance, depending on the context, multiple dots can be interpreted as a pause or as sentence-final punctuation.

Annotating and predicting sentence boundaries in social media texts is in many ways similar to handling sentence boundaries in speech transcriptions, especially when only textual features extracted from the speech transcripts are used [10]. Several previous works that have studied the issues related to sentence boundary segmentation in speech transcripts have acknowledged the issue related to the question of what constitutes a sentence, i.e. what units have to be annotated. For instance, [11] describe a sentence boundary annotation effort on speech transcripts based on syntactic information, i.e. a sequence annotated as a sentence should form a syntactially complete unit. As another example, [12] assessed the inter-annotator agreements of annotating sentence boundaries in Russian transcribed speech data. They found that when using a threshold of 60 % for deciding the majority annotations, more than 70 % of the annotations would have been lost, suggesting that the inter-annotator agreements were relatively low. Our task is simpler compared to these works in a sense that while we cannot assume that the writers of the web texts have followed conventional grammatical norms, they nevertheless have used various punctuation marks and emoticons that help to determine potential sentence boundaries.

3. Corpus Annotation

As a part of this project, the texts included in the Estonian Web Treebank (EWT) [7] were manually annotated with sentence boundaries. The sentence boundary annotation involves determining where one sentence ends and the next one starts and, typically, this is decided based on orthographic sentence boundary markers. However, as the web texts are unedited and the writers might not adhere to usual orthographic rules, relying on common orthographic sentence end markers might, in some cases, result in overly long sentences. Thus, in the context of web texts, we propose to distinguish between two types of sentences:

1. Orthographic sentences that typically start with a capital letter and end with one the commonly used sentence final punctuation marks (.?!);
2. Syntactic sentences that might not adhere to the orthographic rules but that are syntactically complete, i.e. the syntactic head of each word resides inside the sentence.

Originally, our corpus did not contain paragraph boundaries. However, paragraph information, if available, can be very helpful for sentence segmentation systems as it gives a free sentence boundary at the end of every paragraph. The EWT texts originate from the EtTenTen 2013 corpus [13] which contains the paragraph information and thus, we reintroduced the paragraph boundaries back into the corpus. The final corpus consists of 32 documents and in total of 522 paragraphs.

We instructed the annotators to separately annotate the locations of both types of sentence boundaries. For instance, consider a paragraph that orthographically consists of a single long sentence, but syntactically contains several shorter syntactically independent parts, each separated from each other with a comma. The expected annotation of this paragraph was to add an orthographic sentence boundary at the end of the paragraph and syntactic sentence boundaries after each syntactically complete clause.

The corpus was annotated in two equal parts; five undergraduate linguistic students were recruited for the annotation of both parts. The annotators were provided with written annotation guidelines supplied with examples. As a result, we obtained five annotations for one half of the corpus and three annotations for the second half as two annotators annotating the second half were not able to complete their work. The corpus was word-tokenized using the EstNLTK tokenizer. The tokenization was manually checked and corrected by the second author of this paper.

The inter-annotator agreements of the annotations in terms of both Dice coefficient and Fleiss kappa are reported in Table 1. As can be seen from the table, both Dice and kappa values refer to very high inter-annotator agreement. The agreement is the highest for orthographic sentences and the lowest for syntactic sentences. This is expected as annotating the syntactic sentence boundaries assumes more subjective judgement compared to orthographic boundaries.

Table 1. Inter-annotator agreements of sentence boundary annotations. Binary boundary refers to the presence or absence of any boundary annotation, regardless whether it is orthographic or syntactic

	Dice	**Fleiss κ**
Binary boundary	0.92	0.91
Orthographic boundary	0.96	0.89
Syntactic boundary	0.90	0.95

4. Sentence Segmentation and Tokenization Evaluation

For further evaluations, we constructed a corpus containing majority annotations. The decision to retain the boundary was done for both types of sentence boundaries separately—the orthographic boundary was retained in places where at least three (or, in the second half of the corpus, two, respectively) annotators had annotated the orthographic sentence boundary. Similarly, syntactic boundaries were retained in places where at least three (or, in the second half, two) annotators had annotated the syntactic boundary. As a result, the corpus with majority annotations contains three types of sentence boundaries: the majority of annotators had annotated 1) both orthographic and syntactic boundary, 2) only orthographic boundary, and 3) only syntactic boundary.

We used this corpus to evaluate the performance of three tokenization and sentence segmentation systems: EstNLTK version 1.6.5 [14], Stanza [4] and UDPipe [5]. EstNLTK tokenizer is rule-based and the sentence segmenter is based on the Punkt system [15] with additional post-processing rules. For Stanza and UDPipe we used the models pretrained on Estonian UD corpus available on their respective web sites.[2][3]

We compared all systems in three evaluation scenarios:

1. *All boundaries*: both orthographic and syntactic boundaries are considered as gold sentence boundaries;
2. *Orthographic boundaries*: only orthographic boundaries are considered as gold sentence boundaries;
3. *Relaxed boundaries*: both orthographic and syntactic boundaries are considered as gold sentence boundaries but the system was not penalized when it did not mark the syntactic boundary.

Table 2 shows the precision, recall and F1-scores of both sentence segmentation and tokenization results computed with the official evaluation script of the CoNLL 2018 Shared Task [6], with necessary modifications made for the relaxed boundary evaluation.

In all evaluation scenarios, the EstNLTK segmentation system performed the best while Stanza and UDPipe perform similarly, with Stanza being slightly worse. The best F-score of orthographic boundaries on this corpus (87.58) is considerably lower than 92.87, which was the best sentence segmentation score reported in the CoNLL 2018 Shared Task on the UD Estonian test.[4] When considering both orthographic and syntactic boundaries as gold boundaries, the performance is considerably lower, confirming that existing systems are oriented to detecting orthographic sentence boundaries.

[2] https://stanfordnlp.github.io/stanza/available_models.html

[3] http://ufal.mff.cuni.cz/udpipe/models

[4] https://universaldependencies.org/conll18/results-sentences.html

Table 2. Sentence segmentation and tokenization evaluation results. Orth. stands for orthographic

	EstNLTK			Stanza			UDPipe		
	Prec	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
All boundaries	**80.11**	**68.91**	**74.09**	77.12	68.25	72.41	79.24	66.89	72.54
Orth. boundaries	**88.40**	**86.77**	**87.58**	83.64	84.46	84.05	86.80	83.59	85.17
Relaxed boundaries	**88.40**	**85.05**	**86.70**	84.73	83.30	84.00	87.58	83.29	85.38
Tokenization	**98.13**	**98.93**	**98.53**	96.68	96.89	96.78	97.03	96.16	96.59

Table 3. Comparison of sentence segmentation systems on the Estonian UD test set and the reannotated EWT corpus with and without paragraph boundaries

	Estonian UD			EWT			EWT w/o paragraphs		
	Prec	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
EstNLTK	88.60	82.92	85.66	**88.40**	**86.77**	**87.58**	**83.64**	74.70	**78.91**
Stanza	**94.77**	**91.91**	**93.32**	83.64	84.46	84.05	81.89	**76.03**	78.85
UDPipe	93.40	89.79	91.56	86.80	83.59	85.17	81.43	72.21	76.55

In terms of tokenization, again the EstNLTK system performs the best and here, also the Stanza and UDPipe systems perform roughly the same. While the tokenization results are very high on all systems, they still remain below the best results reported in the CoNLL 2018 Shared Task, where the best systems achieved an F-score of 99.96 on the Estonian UD test set.

4.1. Sentence Segmentation Comparison on the UD and EWT Corpora

To assess how much the out of domain characteristics of the web texts affect the performance of different sentence segmentation systems, we next present in Table 3 the sentence segmentation and tokenization results on Estonian Universal Dependencies v2.5 test set. Note that both UDPipe and Stanza systems have been trained on the Estonian UD training set and thus one can expect these systems to perform much better in this scenario. For EWT, the orthographic sentence boundary results have been copied from Table 2.

Table 3 shows that Stanza obtains the best sentence segmentation results on the more well-formed UD test set, while EstNLTK performs considerably worse than both Stanza and UDPipe. The Stanza performance is also better than the best sentence segmentation result reported on the CoNLL 2018 Shared Task (92.87), which might be due to two reasons: 1) the tokenizer that was part of the Stanford parsing pipeline and now repackaged as the Stanza system has been improved, and 2) the CoNLL 2018 Shared Task systems were trained and evaluated on an older version of the Estonian UD corpus and the performance differences stem from the differences in the different versions of the corpus.

Overall, these results confirm that both rule-based and model-based systems are vulnerable to textual domain characteristics. While the EstNLTK seems to be more biased towards noisy web domain, the supervised model-based Stanza and UDPipe perform much better on the well-formed domain they were trained at. This leads to a quite an obvious suggestion that the performance of the Stanza and UDPipe systems on web texts might be improved if they would be trained on the data consisting of annotated web texts.

4.2. The Effect of Paragraph Boundaries

If the text contains paragraph boundaries, then sentence segmentation systems get one sentence boundary per paragraph for free. Therefore, maintaining paragraph boundaries in the input text has potentially large effect on sentence segmentation accuracy. Unfortunately, on the Estonian UD datasets, the paragraph boundaries have not been retained and thus, systems evaluated on the UD test set cannot make use of this so-called free lunch.

As explained above in Section 3, the EWT dataset did not initially contain paragraph boundaries, but these were reintroduced when preparing the corpus for sentence boundary annotation. Thus, we can evaluate the effect the presence or absence of paragraph boundaries has to different sentence segmentation systems. The right-most section of the Table 3 shows these results. While the performance of all systems degrade when paragraph boundaries are not available, the EstNLTK scores decrease the most, suggesting that it is most sensitive to the missing paragraph boundaries. This can be also a partial explanation of why the EstNLTK achieves the lowest results on UD test set that does not contain paragraph boundaries.

5. Error Analysis

To get an idea about where the systems struggle the most on the web corpus, we manually analyzed and categorized the orthographic sentence boundary errors made by the systems. The common error types and the proportional division of errors into these types of different systems are given in Table 4. The overall pattern of error division is roughly similar for Stanza and UDPipe and somewhat different for EstNLTK.

A large category of errors for all systems consists of missing boundaries after multiple punctuation marks (1), typically three dots (...). The complement to these errors is the category 6: wrongly placed boundary after multiple punctuation marks. These situations are often ambiguous also to the human annotators and can be interpreted either as a sentence boundary or a pause in the middle of the sentence, depending on the context and the perception of the individual annotator.

Another complementary pair of error categories is no boundary due to the missing sentence-final punctuation (3) and boundary in the middle of a sentence (5). The latter error often occurs when the next word starts with a capital letter (like a name) and so the systems erroneously decide that this must be the first word of the next sentence. This error only occurs with Stanza and UDPipe and was never observed with EstNLTK. EstNLTK, on the other hand, makes more of the errors of category 3 where the sentence boundary is missed due to the absence of the sentence final punctuation in the text.

Other error categories are mostly due to incorrect tokenization. For instance, both Stanza and UDPipe fail to occasionally put a sentence boundary after a valid sentence final punctuation mark (2). These errors occur mostly in situations where there is no space character between the punctuation mark and the next word, thus leading these systems to predict the whole sequence as a single token. On the other hand, EstNLTK makes a larger proportion of errors compared to Stanza and UDPipe by placing a sentence boundary inside a token that contains a punctuation mark (8). These problems are again possible due to the tokenization errors. Also, abbreviations can confuse the systems. Sometimes the punctuation symbol right after the abbreviation should be part of the abbreviation token

Table 4. The proportion of errors belonging to different categories made by different sentence segmentation systems. The type indicates whether a boundary was missing (M) or an extra boundary was added (A)

No	Error category	Type	EstNLTK	Stanza	UDPipe
1	No boundary after multiple punctuation marks	M	37 %	16 %	19 %
2	No boundary after sentence final punctuation	M	0 %	19 %	34 %
3	No boundary due to missing sentence-final punctuation	M	18 %	8 %	12 %
4	Sentence boundary inside repeated punctuation marks	A	14 %	17 %	7 %
5	Boundary in the middle of a sentence	A	0 %	20 %	12 %
6	Wrong boundary after multiple punctuation marks	A	5 %	9 %	7 %
7	Emoticon has been segmented as a separate sentence	A	6 %	6 %	4 %
8	Sentence boundary inside a token	A	12 %	2 %	1 %
9	Missing boundary after sentence-final abbreviation token	M	3 %	2 %	2 %
	Others		3 %	2 %	2 %

and thus, does not denote the sentence boundary and, in some cases, the punctuation mark signifies the end of the sentence (9). Finally, if the tokenization system splits the repeated punctuation marks (4) or multiple punctuation marks making up emoticons (7) into several tokens, then these can also cause errors in the subsequent sentence segmentation.

6. Discussion and Conclusions

In this paper, we described a sentence boundary annotation effort of the EWT corpus and presented the sentence segmentation and word tokenization results of three segmentation systems on this dataset. These results were compared to the sentence and token segmentation performance of the same systems on the more well-formed UD dataset. We found that on the newly annotated web corpus, the EstNLTK system based on the Punkt model with additional rules performs the best, while on the more well-formed UD test set, the neural models based Stanza and UDPipe systems performed better. Based on these results, our suggestion would be to use EstNLKT system for processing noisy web texts and prefer model-based Stanza or UDPipe systems to segment well-formed written texts. Moreover, the error analysis presented in the previous section indicates that the performance of supervised UDPipe and Stanza systems might be improved on web texts when the training data of these models would also contain web domain data.

Sentence segmentation is a crucial preprocessing step for syntactic analysis systems, which rely on properly detected sentence boundaries. Although model-based syntactic parsers can deal with long orthographic sentences often found in noisy web texts, we hypothesise that syntactic analysers will perform better if these long orthographic sentences are split into several syntactically independent parts that can be analysed independently. The dataset presented in this paper also contains the annotations of syntactic sentence boundaries which enables future work to test out this hypothesis. That would involve evaluating the accuracy of dependency parsing performane on two versions of this dataset: one containing only orthographic sentence boundaries and another containing both orthographic and syntactic boundaries.

Acknowledgements

We thank Kadri Muischnek for the help in developing the annotation guidelines and the students of linguistics who annotated the corpus.

References

[1] O'Connor B, Krieger M, Ahn D. Tweetmotif: Exploratory Search and Topic Summarization for Twitter. In: Fourth International AAAI Conference on Weblogs and Social Media; 2010. .

[2] Proisl T, Uhrig P. SoMaJo: State-of-the-Art Tokenization for German Web and Social Media Texts. In: Web as Corpus Workshop; 2016. p. 57–62.

[3] Orasmaa S, Petmanson T, Tkachenko A, Laur S, Kaalep HJ. EstNLTK - NLP Toolkit for Estonian. In: LREC; 2016. p. 2460–2466.

[4] Qi P, Zhang Y, Zhang Y, Bolton J, Manning CD. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. arXiv preprint arXiv:200307082. 2020.

[5] Straka M, Hajic J, Straková J. UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing. In: LREC; 2016. p. 4290–4297.

[6] Zeman D, Hajic J, Popel M, Potthast M, Straka M, Ginter F, et al. CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In: CoNLL 2018 Shared Task; 2018. p. 1–21.

[7] Muischnek K, Müürisep K, Särg D. CG Roots of UD Treebank of Estonian Web Language. In: NoDaLiDa 2019 Workshop on Constraint Grammar-Methods, Tools and Applications; 2019. p. 23–26.

[8] Read J, Dridan R, Oepen S, Solberg LJ. Sentence Boundary Detection: A Long Solved Problem? In: COLING; 2012. p. 985–994.

[9] Čibej J, Fišer D, Erjavec T. Normalisation, Tokenisation and Sentence Segmentation of Slovene tweets. Normalisation and Analysis of Social Media Texts. 2016:5–10.

[10] Tilk O, Alumäe T. Bidirectional Recurrent Neural Network with Attention Mechanism for Punctuation Restoration; 2016. p. 3047–3051.

[11] Westpfahl S, Gorisch J. A Syntax-Based Scheme for the Annotation and Segmentation of German Spoken Language Interactions. In: Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions; 2018. p. 109–120.

[12] Stepikhov A. Resolving Ambiguities in Sentence Boundary Detection in Russian Spontaneous Speech. In: TSD; 2013. p. 426–433.

[13] Muischnek K. Estonian Web 2013 (etTenTen) Corpus. Center of Estonian Language Resources; 2016.

[14] Laur S, Orasmaa S, Särg D, Tammo P. EstNLTK 1.6: Remastered Estonian NLP Pipeline. In: LREC; 2020. p. 7152–7160.

[15] Kiss T, Strunk J. Unsupervised Multilingual Sentence Boundary Detection. Computational linguistics. 2006;32(4):485–525.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200621

Language Technology Platform for Public Administration

Raivis SKADIŅŠ [a,b], Mārcis PINNIS [a,b], Artūrs VASIĻEVSKIS [a], Andrejs VASIĻJEVS [a,b,1],
Valters ŠICS [a], Roberts ROZIS [a] and Andis LAGZDIŅŠ [a]
[a] *Tilde, Riga, Latvia*
[b] *Faculty of Computing, University of Latvia, Latvia*

Abstract. The paper describes the Latvian e-government language technology platform HUGO.LV. It provides an instant translation of text snippets, formatting-rich documents and websites, an online computer-assisted translation tool with a built-in translation memory, a website translation widget, speech recognition and speech synthesis services, a terminology management and publishing portal, language data storage, analytics, and data sharing functionality. The paper describes the motivation for the creation of the platform, its main components, architecture, usage statistics, conclusions, and future developments. Evaluation results of language technology tools integrated in the platform are provided.

Keywords. Language technology infrastructure, machine translation, speech recognition, speech synthesis, terminology, language resources

1. Introduction

Machine translation (MT) and other language technologies (LT) are invaluable tools for the public sector to reach out and connect with its various constituents in a cost effective and secure way. Language technologies can simplify, automate, and broaden the way public administration interacts with the public in their language.

Technologies like machine translation can significantly reduce the time and costs of translation [1][2] in public sector institutions. In many scenarios, machine translation is the only feasible way to provide access to e-government services in multiple languages. For instance, MT can be used as an assistive technology for vital information distribution in crisis situations [3].

There is a growing pressure to find an efficient solution to tackle language barriers in the multilingual European Union with its 24 official languages, many of which are spoken by less than 10 million people [4]. This is highlighted in the European Parliament resolution on language equality in the digital age adopted on 11 September 2018 that calls on member states and European Commission to boost the development and application of translation technologies and other LT for all EU languages, including languages that are less widely spoken [5]. The language technology community has proposed development of a Pan-European infrastructure for language tools and services to address the multilingual needs of the public sector, industry and society [6].

[1] Corresponding Author: Andrejs Vasiļjevs; Tilde, Vienibas gatve 75a, Riga, Latvia, LV1004; E-mail: andrejs@tilde.com.

The European Commission, with the support of multiple companies and research organisations, addresses the translation needs of public administrations with its online machine translation service eTranslation. eTranslation provides MT functionality from/to any official EU language. It supports plaintext and formatting-rich document translation in asynchronous translation mode.

The platform approach for addressing multilingual needs in an intergovernmental context is exemplified by the EU Council Presidency Translator[2] – a custom-tailored multi-functional translation solution to support the hosting countries of the presidencies of the Council of the European Union [7]. This machine translation platform supports translation from/to all 24 official European languages. The platform supports plaintext, formatting-rich document, and website translation. Registered users from public administration institutions have access to the SDL Trados Studio plug-in, enabling MT support in CAT tool environments. The initial development of the EU Council Presidency Translator was funded by the European Commission through the Connecting Europe Facility (CEF) Telecom programme.

In Lithuania, a language technology platform *versti.eu* provides similar machine translation functionality to translate plaintext, formatting-rich documents (by supporting the most popular MS Office formats), and website translation. The *versti.eu* platform is freely available without registration and is maintained by Vilnius University.

The government of Latvia is among the pioneers in advancing a platform approach to meet multilingual needs on a national level. For the Latvian government, a particular challenge is to ensure that public information and e-services are accessible to all linguistic groups living in Latvia or having business, cultural, or private relationships within the country. To address the need for an automated solution to the multilingual challenge, a centralized language technology platform has been created. The platform, named HUGO.LV[3], developed by Tilde[4] and maintained by the Culture Information Systems Centre, addresses the multilingual needs of public institutions for their internal and external communication.

The paper further describes the motivation for the creation of the platform HUGO.LV, its main components, architecture, usage statistics, as well as presents conclusions and future developments. Evaluation results of the language technology tools integrated in the platform are provided.

2. Motivation

The goal of the language technology platform is to provide the latest developments in language technology in order to help public administrations:

- To reach various audiences and communities by providing instant access to information and e-government services in various languages;
- To exchange information across borders;
- To provide real-time secure translation of confidential texts, documents and websites;
- To boost operational productivity of translation work in public institutions;

[2] http://presidencymt.eu.
[3] http://www.hugo.lv.
[4] http://www.tilde.com.

- To facilitate access to online information and e-services to disabled people;
- To advance the Latvian language in the digital age by making state-of-the-art language technologies developed in Latvia widely accessible and used.

3. Components of the Platform

The core functionality of the platform is machine translation with various usability and integration tools, automated speech recognition and synthesis, and a terminology management and publishing portal.

3.1. Machine Translation Systems

Neural machine translation (NMT) systems for the HUGO.LV platform were trained iteratively during a timeframe of two years. Therefore, the platform supports multiple NMT decoders, including AmuNMT [8] for models trained using multiplicative long short-term memory (MLSTM) [9] based recurrent neural networks, and Transformer [10] models from Sockeye [11] and Marian [12] toolkits. The platform features a total of 12 NMT systems for translation to/from Latvian, English and Russian in the general domain as well as culture and legal domains. The AmuNMT models were trained using the Nematus [13] toolkit. For training of NMT systems, data were prepared using Tilde's parallel data pre-processing workflows (see [14] for more details). For English-Latvian and Latvian-English NMT general domain systems, morphology driven word splitting [15] was applied instead of the simple byte-pair encoding [16].

The quality of the NMT systems was validated using automatic and manual evaluation methods. For the automatic evaluation, we calculated BLEU [17] scores using the ACCURAT balanced evaluation set[5] [18]. The results for the 12 systems are provided in Table 1. The results show that translation quality according to BLEU is lower when translating into morphologically rich languages. BLEU provides even lower scores when analyzing translations between morphologically rich languages. This can be explained by the fact that both Latvian and Russian allow variations in word order that allow the same sentence to be translated using different syntactic structures. However, error analysis could be performed in future work to assess whether the overall error level is comparable when translating from/to morphologically simpler and more complex languages. The legal and cultural domain systems show subpar translation quality when evaluated on the balanced evaluation set, however, this is expected as these are systems adapted on specific datasets.

We also performed manual comparative evaluation at the time of development of the NMT systems. The evaluation was performed by comparing HUGO.LV general domain systems and Google Translate. The evaluation was carried out on (at that time) current news. The results (see Figure 1) show that the translations of the HUGO.LV general domain NMT systems were more preferred by the evaluators (professional translators) than the translations of Google Translate.

---

[5] ACCURAT balanced test corpus for under resourced languages, available for download in the META-SHARE repository http://www.meta-share.org.

Table 1. Automatic evaluation results (in terms of BLEU scores) of HUGO.LV NMT systems using the ACCURAT balanced evaluation set

Translation direction	Legal	Culture	General
English-Latvian	25.64	21.46	28.69
Latvian-English	31.87	32.93	34.31
Russian-Latvian	16.71	16.07	16.94
Latvian-Russian	15.02	15.40	15.65

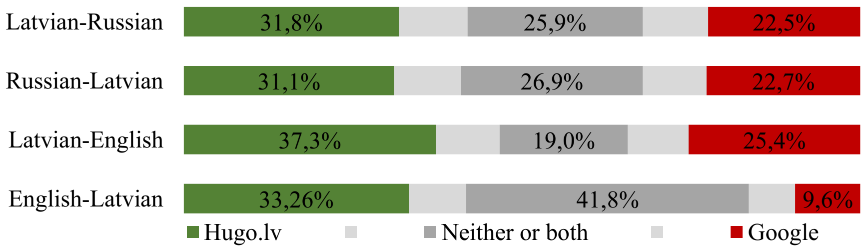


Figure 1. Human comparative evaluation of general domain systems of HUGO.LV and Google Translate

3.2. Text and Document Translation Facilities

The language technology platform provides a translation workspace to translate texts and documents. Users can translate entire documents with a click of a button. Translated documents preserve their original formatting. Multiple formats are supported – rtf, docx, xlsx, pptx, odt, odp, ods, html, etc. A specialized workflow for pre-processing, translating, and post-processing of format-rich documents was developed [19].

3.3. Tools for Website Translation

Two options for website translation are available. A browser add-on for Chrome lets end users to translate any website. For website owners and developers, a translation widget integrates the machine translation functionality into their websites. This lets public administration bodies to provide instant translations of all of their content.

3.4. Tools for Translators

An online computer assisted translation (CAT) tool is integrated in the platform to support semi-professional translation work done by public sector employees. It has been developed by adapting the open-source MateCAT tool [20]. All the segments translated by human translators are stored in a centralized translation memory within the platform. For professional translators, the platform has a plug-in for integrating HUGO.LV machine translation systems in SDL Trados Studio – a computer aided translation tool used by Latvian public administrations.

3.5. Speech Technologies

The text-to-speech functionality provides information for visually impaired or dyslectic people by reading out the written text. Man, woman and youngster voices are provided based on the concatenation approach using diphone synthesis, multiple diphone variations, and LPC residual modification [21].

Automatic speech recognition for Latvian enables text dictation and transcription of audio recordings. It is created with the Kaldi toolkit [22] using an HMM-DNN acoustic model [23], [24] and the Latvian Speech Recognition Corpus [25], [26]. The quality of the Latvian ASR reaches a word error rate (WER) of 9 % as measured on a test corpus.

3.6. Terminology Portal

The terminology portal is a separate platform component that provides an open access to consolidated national terminology resources and supports correct and consistent use of terminology in human and machine translations.

The terminology component has facilities for storing, managing, and accessing national terminology data – 435,000 Latvian terms and 250,000 English terms as well as terms in other languages. Term collections are organized in 22 domains specified by the State Language Center of Latvia. Currently 95 public term collections are available ranging from data digitalized from paper format books and dictionaries to live term collections that are frequently updated by domain and language experts.

The terminology component was created with the following functionality: 1) terminology metadata and term data management; 2) terminology creation workflow; 3) user and their different rights management to ensure online and easy terminology sharing; 4) publishing of news on terminology work, latest protocols and official decisions, some theoretical materials and other content.

The main functionality of the portal is term data and metadata management. All terms are organized in collections. A collection contains concepts that can store the term and its related information in multiple languages. Import and export functionality reuses terminology data in different solutions. TBX, CSV, TSV, MS Excel file format support was created, and these exports are powered with a manual mapping functionality between the file data structure and term database structure. Also, the single term collection view is very important as it provides a full list of term entries within the collection with their data editorial function in place. The terminology portal provides term data export in MT compatible formats for immediate use in training and customising of MT systems.

The terminology creation workflow starts with entering a term candidate and other raw data into the system. When the raw terminology data is prepared, the discussion process can be started. The workflows can be public or private. Public terminology creation workflow enables every Latvian citizen to take part in the discussion about new terms. Private workflows let experts cooperate while keeping the discussions and term candidates confidential. Term creation workflow encourage suggestions for new term translation equivalents, comments on existing ones, as well as comments on a whole terminological concept. During the discussion process, everyone can vote for the best term candidate translation. Finally, the term workflow manager can manually review the list and approve the agreed terms.

Providing content related to the terminology field, the solution helps to form a community of terminologists, and attract their attention with the latest developments in terminology. Also making terminology collections publicly searchable and discoverable allows every citizen to become acquainted with the latest approved terminology.

4. Architecture of the Platform

HUGO.LV is based on the recent version of the LetsMT! platform [27] and has a multi-level architecture (See Figure 2):

- Client level;
- Interface level;
- Logic level;
- Data level;
- High performance computing cluster.

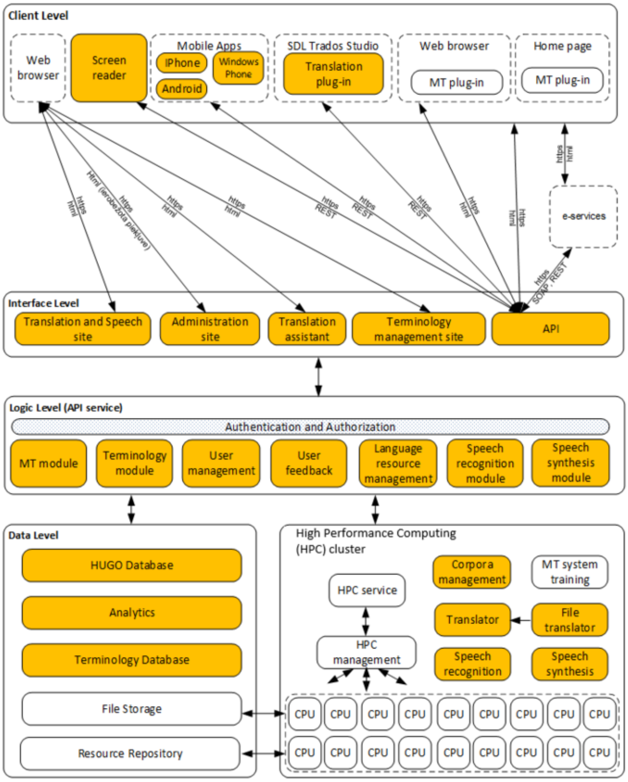


Figure 2. Logical architecture of the platform

4.1. Client Level

The client level includes components that provide HUGO.LV translation functionality on user devices. The client level functionality includes the widget, web browser, and the SDL Trados Studio plug-in components, as well as mobile applications that run on the user's computer or mobile device.

4.2. Interface Level

The interface level includes all system components that are necessary for the system interaction with both human and machine users (the website user interface and the APIs that provide integration across different internal and external systems).

This level provides an interface between external systems, the widget, the web browser plug-in, the translation assistant, and mobile applications. The API, including its OData Service, can be used in external systems. The components at the interface level cooperate with the logic level. The system API has been implemented as a SOAP[6] and/or REST[7] web service (both XML and JSON[8] format). To ensure the security of the data to be transferred, HUGO.LV communication takes place using the HTTPS[9] protocol. The modules are developed in the ASP.NET environment at the interface level.

4.3. Logic Level

The logic level includes modules that provide all functions needed to operate the system. Logic level modules are called only from interface level modules or the 4.5. High Performance Computing (HPC) cluster. Modules are developed in ASP.NET environment that creates separate web services or, in some cases, are included as modules in an interface level application. External users are not allowed to direct access to the level.

4.4. Data Level

The LetsMT! resource repository is used for the storage of language data, their metadata and MT systems. This repository is dedicated to the storage, processing and management of MT language assets. Meanwhile, trained MT systems, which are multiple binary files that together can take several gigabytes, are stored in file storage. Various data that are not directly related to MT systems are stored in the SQL database (MySQL[10]), such as user data, user feedback, terminology, recommended translation fixes, system settings, analytics, etc. External users are not allowed to access this level.

4.5. High Performance Computing Cluster

In order to train MT systems, several model calculations and optimization tasks occur in parallel, which can take from several hours up to two weeks to complete. These computing tasks are performed in a high-performance computing cluster that works on the Oracle Grid Engine[11] platform on the Linux operating system. The HPC cluster performs a variety of processes that require high computing capacities, such as data preparation and processing tasks, text alignment tasks, training tasks for MT systems, translation tasks for texts and files, speech recognition and synthesis tasks. The use of the HPC cluster ensures the scalability of the system, i.e. increasing the performance of the system, if necessary, by automatically adding new computing resources to the HPC cluster during operation.

[6] SOAP: http://www.w3.org/TR/soap/, http://en.wikipedia.org/wiki/SOAP.
[7] REST: http://en.wikipedia.org/wiki/Representational_State_Transfer.
[8] JSON: http://www.json.org/, http://en.wikipedia.org/wiki/JSON.
[9] HTTPS: http://en.wikipedia.org/wiki/HTTP_Secure.
[10] MySQL: http://www.mysql.com/.
[11] Oracle Grid Engine, formerly Sun Grid Engine (SGE): http://gridengine.org/, http://en.wikipedia.org/wiki/Sun_Grid_Engine.

5. Usage of the Platform

Since January 2019, when the fully functional platform was launched, the HUGO.LV website has been visited 1.3 million times, 26.94 million translation requests have been made, and more than 552 million words have been translated. The most frequently used translation direction is from English to Latvian, the second most popular translation direction is from Russian to Latvian.

Speech technologies of the platform have also been popular by users, with 37.70 million words recognized by transcribing 6,686 hours of audio recordings, and 6.2 million words generated using the speech synthesis functionality.

The HUGO.LV machine translation service is integrated in several Latvian government websites, providing multilingual access for information and e-services. Machine translation services are integrated in the state service portal latvija.lv providing descriptions of services in English and Russian languages, Latvian Electronical declaration system eds.vid.gov.lv, electronical auction website izsoles.ta.gov.lv, and the website of the city library of Valmiera biblioteka.valmiera.lv. Speech recognition technology is used by the national radio Latvijas Radio, enabling transcriptions of audio broadcasts in textual form.

The usability of the platform has been recognised in several national and international contests and events. In 2019, the HUGO.LV platform was awarded the "Platinum Mouse", which is the main award of the IT industry in Latvia, curated by the Latvian Information and Communication Technology Association (LIKTA). In 2015, the HUGO.LV machine translation service was nominated for the World Summit on Information Society (WSIS) Project Prize in the category "Cultural Diversity and Identity, Linguistic Diversity and Local Content".

6. Conclusions and Future Work

The considerable use of the HUGO.LV services and their integration in various public online systems clearly demonstrate the value and importance of the platform for public administration and society as a whole.

Future activities include expanding the platform with new components for creating multilingual chatbots that can serve multiple public institutions. The chatbots will use new components such as natural language understanding, natural language generation, intent detection, and dialog management, as well as existing components of the platform for machine translation and speech processing.

The technological architecture and modularity of HUGO.LV platform makes it adaptable to other languages and usage contexts. This makes it possible to introduce a similar solution in other countries by using the same framework and integrating the necessary language tools and services. This can significantly boost speed and decrease costs of adapting feature-rich multilingual platform solution across EU member states.

References

[1] Samuel L, Amrhein C, Düggelin P, Gonzalez B, Zwahlen A, Volk M. Post-editing Productivity with Neural Machine Translation: An Empirical Assessment of Speed and Quality in the Banking and Finance Domain. arXiv preprint arXiv:1906.01685. 2019.

[2] Screen B. Productivity and quality when editing machine translation and translation memory outputs: an empirical analysis of English to Welsh translation. Studia Celtica Posnaniensia 2, no. 1; 2017. p. 113-36.
[3] Lewis W, Munro R, Vogel S. Crisis MT: Developing a Cookbook for MT in Crisis Situations. In Proceedings of the Sixth Workshop on Statistical Machine Translation; 2011. p. 501-511.
[4] Rehm G, Uszkoreit H, Dagan I, Goetcherian V, Dogan MU, Váradi T. An update and extension of the META-NET Study "Europe's Languages in the digital age". 2014. p. 1-8.
[5] European Parliament. Resolution of 11 September 2018 on language equality in the digital age (2018/2028(INI)). P8_TA(2018)0332. 2018.
[6] Vasiljevs A, Hajic J, Hummel J, van Genabith J, Kalnins R. European Platform for the Multilingual Digital Single Market: Conceptual Proposal. In Baltic HLT, p. 20-27. 2016.
[7] Pinnis M, Kalniņš R. Developing a neural machine translation service for the 2017-2018 european union presidency. In Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 2: User Papers); 2018. p. 72-83.
[8] Junczys-Dowmunt M, Dwojak T, Hoang H. Is Neural Machine Translation Ready for Deployment? A Case Study on 30 Translation Directions. Arxiv. http://arxiv.org/abs/1610.01108. 2016
[9] Krause B, Lu L, Murray I, Renals S. Multiplicative LSTM for sequence modelling. arXiv preprint arXiv:1609.07959. 2016.
[10] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention Is All You Need. Advances in Neural Information Processing Systems; 2017. p. 5998-6008.
[11] Hieber F, Domhan T, Denkowski M, Vilar D, Sokolov A, Clifton A, et al. Sockeye: A toolkit for neural machine translation. ArXiv e-prints; 2017.
[12] Junczys-Dowmunt M, Grundkiewicz R, Dwojak T, Hoang H, Heafield K, Neckermann T, et al. Marian: Fast Neural Machine Translation in C++. ArXiv Preprint ArXiv:1804.00344. https://arxiv.org/abs/1804.00344; 2018.
[13] Sennrich R, First O, Cho K, Birch A, Haddow B, Hitschler J, et al. Nematus: a Toolkit for Neural Machine Translation. In Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics; 2017. p. 65-68.
[14] Pinnis M, Rikters M, and Krišlauks R. Tilde's Machine Translation Systems for WMT 2018. Proceedings of the Third Conference on Machine Translation; 2018. p. 477-485.
[15] Pinnis M, Krišlauks R, Deksne D, Miks T. Neural Machine Translation for Morphologically Rich Languages with Improved Sub-word Units and Synthetic Data. Proceedings of TSD 2017: Text, Speech and Dialogue; 2017. p. 237-245
[16] Gage P. A New Algorithm for Data Compression. C Users Journal, 12(2); 1994. p. 23-38.
[17] Papineni K, Roukos S, Ward T, Zhu WJ. BLEU: a Method for Automatic Evaluation of MT. Proc. of the 40th Annual Meeting on Association for Computational Linguistics; 2002. p. 311-318.
[18] Skadiņš R, Goba K, Šics V. Improving SMT for Baltic Languages with Factored Models. In Frontiers in Artificial Intelligence and Applications, volume 219; 2010. p. 125-132.
[19] Pinnis M, Skadiņš R, Šics V, Miks T. Integration of Neural Machine Translation Systems for Formatting-Rich Document Translation. Natural Language Processing and Information Systems. NLDB 2018. Lecture Notes in Computer Science, vol. 10859; 2018. p. 494-497.
[20] Federico M, Bertoldi N, Cettolo M, Negri M, Turchi M, Trombetti M, et al. The Matecat Tool. In COLING (Demos); 2014. p. 129-132.
[21] Goba K, Vasiļjevs A. Development of Text-To-Speech System for Latvian. Proceedings of NODALIDA 2007, Tartu, Estonia.
[22] Povey D, Ghoshal A, Boulianne G, Burget L, Glembek O, Goel N, et al. The Kaldi speech recognition toolkit. In IEEE 2011 workshop on automatic speech recognition and understanding (No. CONF). IEEE Signal Processing Society; 2011.
[23] Salimbajevs A, Strigins J. Latvian Speech-To-Text Transcription Service. Proceedings of Interspeech 2015; p. 22-723.
[24] Salimbajevs A. Towards the First Dictation System for Latvian Language. Frontiers in Artificial Intelligence and Applications, vol. 289: HLT – The Baltic Perspective; 2016. p. 66-73.
[25] Pinnis M, Auziņa I, Goba K. Designing the Latvian Speech Recognition Corpus. Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14); 2014. p. 1547-1553.
[26] Pinnis M, Salimbajevs A, Auzina I. Designing a Speech Corpus for the Development and Evaluation of Dictation Systems in Latvian. Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2016); 2016. p. 775-780.
[27] Vasiļjevs A, Skadiņš R, Tiedemann J. LetsMT!: A Cloud-Based Platform for Do-It-Yourself Machine Translation. Proceedings of the ACL 2012 System Demonstrations; 2012. p. 43-48.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200622

What Can We Learn from Almost a Decade of Food Tweets

Uga SPROĢIS [a,1], Matīss RIKTERS [b]
[a] *Faculty of Computing, University of Latvia, Latvia*
[b] *The University of Tokyo, Japan*

Abstract. We present the Latvian Twitter Eater Corpus - a set of tweets in the narrow domain related to food, drinks, eating and drinking. The corpus has been collected over time-span of over 8 years and includes over 2 million tweets entailed with additional useful data. We also separate two sub-corpora of question and answer tweets and sentiment annotated tweets. We analyse the contents of the corpus and demonstrate use-cases for the sub-corpora by training domain-specific question-answering and sentiment-analysis models using the data from the corpus.

Keywords. Annotated corpora, social networks, food data, Latvian

1. Introduction

Even though the usage and popularity of Twitter have stopped rapidly growing and even dropped in recent years[2], it still has a considerable amount of loyal users who keep on sharing everything from worldwide events to random personal details with their followers. We decided to focus on one of the random personal details that people share, specifically, anything to do with food consumption and related topics.

Several corpora of Latvian tweets exist in prior work, but none of them are domain-specific and have been collected over an extensive period of time. Milajevs [1] collected and analysed 1.4 million tweets geo-located in Riga, Latvia from April 2017 to July 2018 and 60 thousand tweets [2] from November 2016 to March 2017. Pinnis [3] collected and analysed 3.8 million tweets of Latvian politicians, companies, media, and users who interacted from August 2016 to July 2018 There are also several data sets of general sentiment-annotated tweets [4], [5], [3][3] amounting to 14,781 tweets in total.

In this paper, we describe the Twitter eater corpus (TEC) and analyse its contents. We also provide two sub-corpora: one consisting of question and answer tweets and one with sentiment-annotated tweets. More details can be found in Section 2. In Sections 3.1 and 3.2, we describe question answering and sentiment analysis experiments using our corpus. Finally, we conclude the paper in Section 4.

[1]Corresponding Author: Uga Sproģis; E-mail: ugasprogis12@inbox.lv.
[2]https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users
[3]https://github.com/nicemanis/LV-twitter-sentiment-corpus

2. The Twitter Eater Corpus

The corpus consists of tweets that have been collected from October 2011 [6] until April 2020. They are tracked using 363 keywords, which are various inflections of Latvian words associated with eating, tasting, breakfast, lunch, dinner, etc. The main keywords are shown in Table 1: the words in bold are mostly verbs that describe eating - these were inflected to all usable forms and included in the full keyword list. The rest of the keywords are a set of the top 60 food-related words that were the most popular in the first month of collecting the tweets.

Figure 1 illustrates the contents of a single tweet from the TEC in JSON notation. Each tweet consists of primary fields - *"tweet_id"*, *"tweet_text"*, *"tweet_author"* and *"created_at"*, which will always be present, and optional fields, which depend on the tweet text and metadata. We separate three groups of optional fields: 1) "*media_url*" and *"expanded_url"*, which contain information about the media files from the tweet; 2) *"location_name"*, *"location_lng"*, *"location_lat"* and *"location_country"*, which specify where the tweet was created; and 3) *"food_surface_form"*, *"food_nominative_form"*, *"food_group"* and *"food_english_translation"*, which contain semicolon-separated lists of foods or drinks that appear in the tweet.

At the beginning of the project, approximately 15,000 food and drink words from collected tweets were manually annotated with their respective nominative forms, English translations and food groups according to the food guide pyramid [7]. The food groups are: bread, cereal, rice, pasta (6); vegetables (5); fruit, berries (4); milk products (3); meat, eggs, fish (2); fats, oils, sweets (1). There are two additional groups for drinks: alcoholic drinks (7) and non-alcoholic drinks (8).

The corpus is available on Github[4], in accordance with the content redistribution section of the Twitter Developer Agreement and Policy[5]. The public release includes tweet IDs along with data fields created within the scope of this project (starting with *"location_lng"* in Figure 1). The complete version is available upon individual request for research purposes. The repository also includes data processing scripts and details on how to reproduce our experiments.

Table 1. List of main keywords used to collect the corpus

taste	**lunch**	beet	potato	mandarin	sweet
eat	**feast**	bun	cabbage	sauce	mushroom
breakfast	**drink**	carrot	candy	pancake	onion
dine	**treat**	chips	sour cream	dumpling	chocolate
dinner	**nom**	vegetable	cream soup	gingerbread	tea
bite	**appetite**	meat	cake	rice	tomato
meal	orange	Hesburger	drink	salad	grape
food	apple	coffee	McDonald's	ice cream	strawberry

[4] https://github.com/Usprogis/Latvian-Twitter-Eater-Corpus
[5] https://developer.twitter.com/en/developer-terms/agreement-and-policy

```
{
  "tweet_id": 1213025400273735680,
  "tweet_text": "Gulašzupa #receptesĪsumā gulašzupa ir gana
     vienkārša liellopu gaļas bāzēta zupa https://t.co/
     OnqDwotQr0 https://t.co/Z2tAodyj9M",
  "tweet_author": "receptes_eu",
  "created_at": "2020-01-03 11:12:54",
  "media_url": "http://pbs.twimg.com/media/ENWIKb8WsAAiLKE.
     jpg",
  "expanded_url": "https://twitter.com/receptes_eu/status/12
     13025400273735680/photo/1",
  "location_name": "Ogresgals",
  "location_lng": "24.7377",
  "location_lat": "56.8079",
  "location_country": "Latvia",
  "food_surface_form": "Gulašzupa;liellopu;gaļas;zupa;",
  "food_nominative_form": "gulašs;liellops;gaļa;zupa;",
  "food_group": "2;2;2;6;",
  "food_english_translation": "Goulash;Cattle;Meat;Soup;",
}
```


Figure 1. An example of a tweet from the TEC with all available metadata

2.1. Content Overview

The corpus contains 2,275,787 tweets, of which 155,057 contain media information, 165,335 contain location information and 1,297,159 tweets mention foods or drinks. Table 2 shows the 10 most popular foods and drinks from the TEC. Looking from a Latvian consumer perspective[6], it is very typical that Latvians mostly drink water, tea, juice, beer and eat meat, vegetables and fruits. Interesting, however, is the high popularity of sweets such as chocolate, cakes, ice cream and Coca-Cola.

Table 2. List of foods and drinks which are the most popular overall

Food	Count	Drink	Count
Chocolate	117,235	Tea	163,338
Ice cream	86,109	Coffee	120,040
Meat	85,574	Juice	18,179
Potatoes	70,135	Water	15,692
Salads	61,616	Beer	14,845
Cake	52,267	Cocktails	8,207
Soup	46,545	Coca-cola	5,016
Pancakes	40,203	Alcohol	4,766
Sauce	40,201	Champagne	3,673
Apple	36,571	Vodka	2,802

[6]https://enciklopedija.lv/skirklis/4980-nacion%C4%81l%C4%81-virtuve-Latvij%C4%81

Figure 2 shows the yearly count of collected tweets along with the potential trend (since for 2011 and 2020, only a part has been collected) and the general popularity of Twitter and Instagram (a competing social network) for Latvia from Google Trends [7]. There was a stable income of food tweets up until 2015, however, it seems that the following decrease correlates with the overall drop in the popularity of Twitter in Latvia, which seems to be directly opposite to the popularity of Instagram in Latvia according to Google Trends.

In Figure 3, we have visualised four of the largest tweet trends over the past years from the Latvian speaking twitter users. The most recent one just a month ago - panic buying of buckwheat due to the CoViD19 pandemic of 2020, followed by the doubling of butter prices in 2017, Latvian sprat import ban to Russia in 2015, and, finally, the horsemeat scandal in 2013. If we look closer at the 2823 tweets about meat in week 9 of 2013, we can see multiple inflexions of the word "horse" along with words like "scandal" and "investigation" among the most common words.

Figure 4 shows a selection of seasonal trends averaged from data between 2012 and 2019. Most trends have one peak zone indicating parts of the year when they are more popular. Examples of this are gingerbread and tangerines in December, and strawberries and ice cream in the summer. We were expecting to see chocolate peak high on Valentine's day, but while it does peak, the difference is not as high.

2.2. *Question - Answer Sub-corpus*

We noticed that there are a number of tweets in our corpus that express questions. To highlight one of the uses of the corpus, we selected a subset of tweets which include

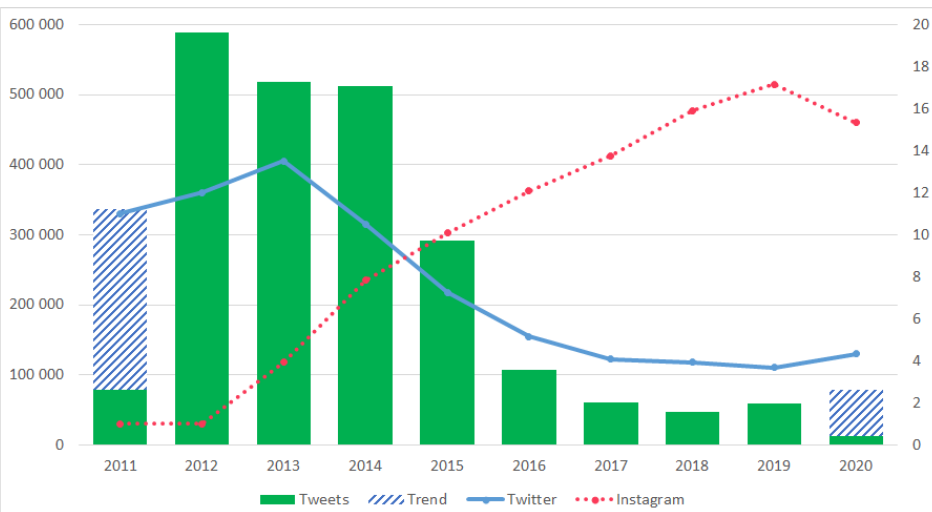


Figure 2. Collected tweet count by year

[7]https://trends.google.com/trends/explore?hl=en-US&tz=-540&date=2011-10-06+2020-03-14&geo=LV&q=%2Fm%2F0fjd36,%2Fm%2F0289n8t,%2Fm%2F02y1vz,%2Fm%2F0glpjll&sni=3

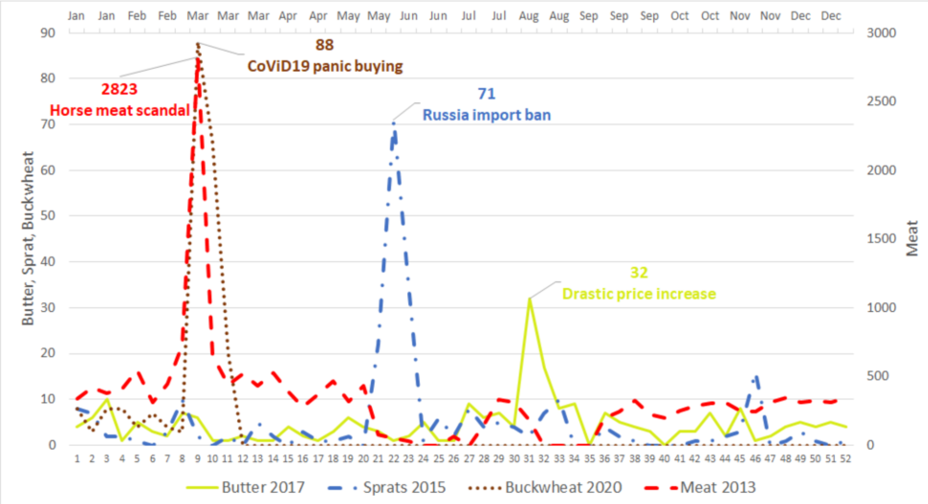


Figure 3. Four of the large trends noticeable in the TEC

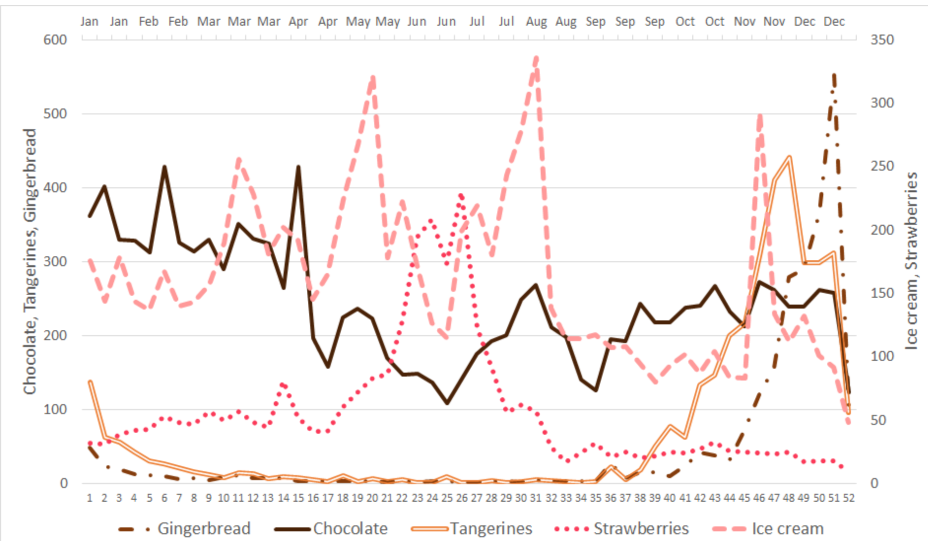


Figure 4. Five of the yearly seasonal trends noticeable in the TEC

at least one of typical Latvian question words[8] or phrases along with a question mark. This resulted in 215,233 question tweets. To gather answers for them, we scraped Twitter's web version[9], which resulted in 19,871 tweets with at least one reply. Since there were many tweets with multiple answers, we eventually wound up with 42,744 question-answer pairs. We randomly selected subsets of 1,000 and 500 question-answer pairs to use as the development set and evaluation set respectively.

[8] http://valoda.ailab.lv/latval/vidusskolai/SINTAKSE/sint3jaut.htm
[9] https://github.com/luodaoyi/TwEater

2.3. Sentiment Annotated Sub-corpus

We manually annotated 5,420 tweets, marking them as positive, neutral or negative. This gave us 1,631 positive, 2,507 neutral and 1,282 negative tweets. We further split these into a test set of 250 tweets from each class and a training set.

3. Experiments

3.1. Question Answering

Typical question answering systems are trained using paragraphs of text, questions about the paragraphs and answers to those questions [8]. Since we only had question-answer, we chose to train an encoder-decoder model similar to machine translation using questions and answers as source and target languages, respectively. We used Sockeye [9] to train transformer architecture models with the base parameters until they reached convergence on development data.

Our initial experiments using only TEC data showed rather poorly generated answers due to the lack of general-domain training data. To mitigate this, we used the same approach to select question-answer tweets from the Latvian Tweet Corpus [3]. This gave us 546,982 additional question-answer pairs to add to our training data.

3.1.1. Data Pre-processing

We performed tokenisation or truecasing using scripts from the Moses Toolkit [10]. We used Sentencepiece [11] to create a shared subword vocabulary of 8,000 tokens. We replaced all Twitter-specific *@user* mentions with *@USR* and URLs with *@URL*, as these usually do not contain relevant linguistic data for the model to learn. We also replaced multiple consecutive *@USR* or *@URL* tags with a single one and removed them completely if they were either at the start of the tweet or in the end.

3.1.2. Results

Figure 5 shows several examples of generated answers to the evaluation data questions. There were many hypothesis answers that were not even close to the reference ones but still made some sense in relation to the questions, such as the first two. There were also just as many or even more answers that made no sense at all like the last one.

We performed a small-scale human evaluation on the results by asking 5 annotators to evaluate a random 10 % of the evaluation set by marking generated answers as either OK or not good (NG). The evaluators marked 46.40 % of answers as OK. The evaluators had an overall agreement of 66.27 % (Free-marginal kappa [12] - 0.33), which indicates moderate agreement.

3.2. Sentiment Analysis

For the sentiment analysis experiment, we used 3,116 of our sentiment-annotated tweets. 250 random tweets from each sentiment class were separated as an evaluation corpus and the remaining 2,368 were used for training.

Question: Bet es jau tevis dēļ visu daru. Cikos jāiet ēst torti?
Hypothesis: Es jau paēdu.: D
Reference: uzzvanīšu kad varēs nākten ciemen.

Question: Man šķūnī nesen piedzima ežuļi. Tāpēc SOS jautājums - ko ēd mazi eži?
Hypothesis: Jā, tas ir ļoti labs jautājums.:)
Reference: tikai nebaro ar krūti, viņi nepanes pienu!

Question: kāpostu tīteņi vai cepelīni?
Hypothesis: Jā, svaigēdājus.
Reference: kāpostu tīteņi

Figure 5. Example questions with generated hypothesis answers and true reference answers

For sentiment analysis, we performed similar data pre-processing steps as for question answering, except for splitting words in sentence pieces. We also experimented with stemming[10] and lemmatizing [13] words.

Table 3 shows the results of our sentiment analysis experiments. We compared a Python implementation of the Naive Bayes classifier from NLTK[14] against Pinnis [3] implementation of the Perceptron classifier. We also experimented with several combinations of training data sets - TE (our Twitter Eater dataset), MP [3], RV [5], PE [4], NI[11]. We found that the highest classification accuracy - 61.23 % - is achieved by using all but NI data sets for training and only stemming all words.

Table 3. Accuracy of our sentiment analysis experiment results on scale of 0 to 100

Training Data	TE	MP	MP.PE	TE.MP	All	TE.MP.RV.PE
Naive Bayes	53.21	43.32	45.72	56.55	59.63	58.02
Perceptron	53.07	52.67	53.47	57.87	57.33	58.27
				Stemmed		
Naive Bayes	53.74	46.39	50.67	58.16	60.56	**61.23**
Perceptron	56.67	53.73	54.13	60.00	56.93	57.73
				Lemmas		
Naive Bayes	53.88	45.45	49.60	56.42	58.42	59.63
Perceptron	54.41	51.07	53.07	57.35	56.95	56.95
				Stemmed Lemmas		
Naive Bayes	54.41	45.99	49.33	57.62	59.63	59.63
Perceptron	53.34	51.47	52.67	58.29	56.68	57.09

[10] https://github.com/rihardsk/LatvianStemmer
[11] https://github.com/nicemanis/LV-twitter-sentiment-corpus

4. Conclusion

In this paper, we described the creation of a fairly large narrow-domain corpus of Twitter posts related to the topic of eating. We gave some insights in overall observations gained from the corpus contents and various trends that we noticed from the data. We believe that the data would be useful in many linguistic, sociological, behavioural and other research areas.

We experimented with creating a food-related question answering system using one subset of our data and a sentiment analysis system using another subset to highlight potential use-cases of our corpus. While the results did not break new ground, we hope that they inspire related future research.

Acknowledgements

We would like to thank Mārcis Pinnis for sharing his collected tweet dataset with us as well as running experiments with his model using our data.

References

[1] Milajevs D. Language use in a multilingual tweet corpus. In: Human Language Technologies–The Baltic Perspective: Proceedings of the Eighth International Conference Baltic HLT 2018. vol. 307. IOS Press; 2018. p. 88.
[2] Milajevs D. Toward a Comparable Corpus of Latvian, Russian and English Tweets. In: Proceedings of the 10th Workshop on Building and Using Comparable Corpora. Vancouver, Canada: Association for Computational Linguistics; 2017. p. 26–30.
[3] Pinnis M. Latvian Tweet Corpus and Investigation of Sentiment Analysis for Latvian. In: Proceedings of the 8th Baltic Conference on Human Language Technologies (Baltic HLT 2018); 2018. p. 112–119.
[4] Peisenieks J, Skadiņš R. Uses of Machine Translation in the Sentiment Analysis of Tweets. IOS Press. 2014.
[5] Vīksna R. Sentiment Analysis in Latvian Tweets [Master's Thesis]. Rīgas Tehniskā universitāte; 2018.
[6] Rikters M. Universālas metodes Twitter datu analīzei [Bachelor's Thesis]. Latvijas Universitāte; 2012.
[7] Duston D. Food guide pyramid is built on a base of grains. Daily News. 1992 Apr:8–8.
[8] Rajpurkar P, Zhang J, Lopyrev K, Liang P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Austin, Texas: Association for Computational Linguistics; 2016. p. 2383–2392.
[9] Hieber F, Domhan T, Denkowski M, Vilar D, Sokolov A, Clifton A, et al. Sockeye: A Toolkit for Neural Machine Translation. ArXiv e-prints. 2017 dec.
[10] Koehn P, Hoang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, et al.. Moses: open source toolkit for statistical machine translation. Association for Computational Linguistics; 2007.
[11] Kudo T, Richardson J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations; 2018. p. 66–71.
[12] Randolph JJ. Free-Marginal Multirater Kappa (multirater κfree): An Alternative to Fleiss' Fixed-Marginal Multirater Kappa. In: Presented at the Joensuu Learning and Instruction Symposium. vol. 2005; 2005. .
[13] Paikens P. Lexicon-based morphological analysis of Latvian language. In: Proceedings of the 3rd Baltic Conference on Human Language Technologies (Baltic HLT 2007); 2007. .
[14] Bird S, Loper E, Klein E. Natural language processing with python O'reilly media Inc; 2009.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200623

OCR Challenges for a Latvian Pronunciation Dictionary

Laine STRANKALE, Pēteris PAIKENS
University of Latvia, Institute of Mathematics and Computer Science

Abstract. This paper covers the devlopment of a custom OCR solution based on the Tesseract open source engine developed for digitization of a Latvian pronunciation dictionary where the pronunciation data is described using a large variety of diacritic markings not supported by standard OCR solutions. We describe our efforts in training a model for these symbols without the additional support of preexisting dictionaries and illustrate how word error rate (WER) and character error rate (CER) are affected by changes in the dataset content and size. We also provide an error analysis and postulate possible causes for common pitfalls. The resulting model achieved a CER of 2.07%, making it suitable for digitization of the whole dictionary in combination with heuristic post-processing and proofreading, resulting in a useful resource for further development of speech technology for Latvian.

Keywords. OCR, pronunciation, Tesseract

1. Problem Description

Accurate speech recognition and speech synthesis are greatly helped by a large database of words and their phonetic notations. For Latvian the most comprehensive resource of phonetic information currently available is *"Latviešu valodas pareizrakstības un pareizrunas vārdnīca"* (LVPPV)[1], "Latvian spelling and pronunciation dictionary", which contains over 80000 words with full pronunciation transcription. Unfortunately, a machine-readable version of this dictionary is not available because of historical reasons of how this dictionary was originally developed.

During earlier digitization efforts, the LVPPV was scanned. As seen in Figure 1, each dictionary entry contains a pronunciation section enclosed in square brackets. However, in the earlier digitization only the spelling part of the entries was suitable for OCR technologies available at that time, as the pronunciation information is encoded in a custom set of symbols not expected by the available OCR models.

The availability of this scanned data and the requirement for a large machine-readable resource of Latvian pronunciation motivates our research to develop an accurate OCR model for this custom phonetic alphabet to assist a full digitization of this dictionary. Although OCR results are rarely perfect and would be expected to contain mistakes that need human review, an effective custom OCR model

iesalt [ìesàlt], *sk.* salt
iesāļezers [iesā̀ļęzę̀rs]
iesāļš [iesā̀ļš]
uzvelties [uzveļ̂tiês], *sk.* velties
uzvērpties [uzvḕrptiês], *sk.* vērpties
uzversmot [uzvęr̂smuôt], *sk.* versmot
uzvērst [uzvḕrst], *sk.* vērst¹
uzvērt [uzvẽrt], *sk.* vērt
uzvēsmot [uzvę̃smuôt], *sk.* vēsmot

Figure 1. Extracts from Latvian spelling and pronunciation dictionary (LVPPV)

would ensure that this data can be reviewed and corrected with a reasonable amount of manual work.

In LVPPV the pronunciation is denoted by using symbols that extend the standard Latvian alphabet (Figure 2) with additional symbols with diacritic modifiers illustrated in Figure 3. Accurate recognition of this phonetic alphabet is challenging because meaningful differences in pronunciation are expressed by minor variations of diacritic marks – for example, there is a common need to distinguish the pronunciation of the letter 'a' as 'ã' or 'â'. Standard OCR systems distinguish visually similar characters through assistance of dictionaries and statistical n-gram frequency models, which is not possible in this scenario as we need to digitize the only dictionary which has this data.

a ā b c č d e ē f g ģ h i ī j h k ķ
l ļ m n ņ o p r s š t u ū v z ž

Figure 2. Symbols in Latvian alphabet (only lowercase shown)

ì ĩ î à ã â ā̀ è ẽ ê ę ę̀ ę̃ ę̂ ę̄ ę̄̀ ḕ
ù ũ û û̯ ò õ ô
l̃ l̂ ļ̃ ļ̂ r̃ ŗ r̂ ñ ņ̃ m̃ m̂ ŋ̃ ŋ j̣ ˙

Figure 3. Additional symbols in phonetic transcriptions (LVPPV uses only lowercase letters)

2. Related Work

There has been no known work in digitizing this style of printed phonetic transcription of Latvian. There is phonetic transcription available for a limited number of words in digital resources *"Mūsdienu latviešu valodas vārdnīca" Dictionary of contemporary Latvian* (MLVV) and tezaurs.lv online dictionary [7]. These resources are much smaller than LVPPV, but they can be used as a test set for verifying the OCR accuracy on the subset of words contained in both resources.

There is earlier work on rule-based approaches to derive phonetic transcriptions of Latvian[5]. However, these methods are limited and would also directly benefit from a larger digital database of phonetic transcriptions such as the final result of digitizing LVPPV.

In reviewing research on OCR technology applications, we were not able to identify any useful publications on digitizing phonetic transcription of other languages. However, there is extensive literature on the more general task of developing OCR solutions for new scripts. The prevailing paradigm for such OCR systems relies on retraining existing general-purpose OCR systems on a targeted set of training data examples for the new script using supervised machine learning and deep neural networks. While there are also examples using custom systems implemented from scratch, especially for commercial solutions, we consider that it is reasonable to adapt an existing system in order to reuse existing functionality of recognizing the latin alphabet and only adjust the specific characters (diacritic combinations) that are used in this pronunciation dictionary.

The two leading OCR tools that support training additional languages are Abby Finereader[1] and Tesseract[6]. For this research we have chosen Tesseract as it is a free open-source solution and has been the basis for multiple successful implementations of OCR for a new script[2,4].

3. Method

Since version 4 Tesseract uses a LSTM[3] based neural network architecture which significantly outperforms previous versions. Tesseract 4 provides three training methods[8]:

1. fine-tuning for impact (adding a few extra characters to an existing model);
2. training just a few layers (removing the top layers from an existing model and replacing with new ones);
3. training from scratch.

These approaches differ substantially by how much training data is required. As the pronunciation data introduces a significant number of new characters and we want to limit training data size, the second training method is considered the most appropriate. LVPPV largely contains characters and letter patterns characteristic to the Latvian language, thus Tesseract's pre-trained Latvian language model was chosen as the base model.

3.1. Data Preparation

The Tesseract training process requires training data that consists of scanned images annotated with character bounding boxes aligned with appropriate characters. In this work, whole pages of the scanned LVPPV dictionary were used as the basic units of data. The input data was selected, so that the chosen images contain all the new phonetic characters frequently enough to minimize errors in the final model.

[1] https://www.abbyy.com/en-us/finereader/

OCR requires good quality images and LVPPV was scanned in 600 dpi resolution. Additionally, as the goal of this research is to recognize text from this particular dictionary, we know that all data will be a single font, thus we did need to adapt it for font variation as is commonly required for general purpose OCR solutions.

3.2. Post-processing Heuristics

We developed a method to determine a particular character's error rate. It uses Levenshtein distance, also known as edit distance, that measures the difference between two sequences, in this case, the expected character string and the OCR generated character string.

This allowed to both strategically choose pages from LVPPV for training data and detect instances where rule-based post-processing could be applied using known phonetic transcription characteristics and gained insights.

Analysis of OCR errors during development revealed that a common error pattern involves character duplicates – for example, 'ã' which was mistaken for 'ãâ'. Some examination revealed that this is an unresolved issue in Tesseract implementation wherein if two characters have a similar probability of being the correct then both are output[2]. Some proposed solutions suggested looking at the character bounding boxes but because of the way Tesseract is implemented this still would not be a complete solution. However, as the model improves, the frequency of these errors should decrease. A large part of such errors can be automatically corrected with heuristic post-processing methods because these mistakes generate character sequences that are not plausible in Latvian words.

4. Training

We performed experiments to analyze the effect of training set size on OCR accuracy to determine when the effort put into preparing training examples exceeds the effort needed to check for errors in the processed pages manually.

Multiple models were trained with various limitations on training data size. Data was increased gradually, with one LVPPV page as the step size. The final model was trained on a dataset of 2949 lines (dictionary entries with a word spelling, its pronunciation, and auxiliary comments); Table 1 illustrates how the LVPPV page count corresponds to the line count in the training data.

Table 1. Training Set Size. Line Count vs. LVPPV Page Count

Line Count	1175	1314	1433	1551	1666	1784	1910	2026
LVPPV Page Count	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*
Line Count	2141	2256	2372	2483	2601	2716	2832	2949
LVPPV Page Count	*18*	*19*	*20*	*21*	*22*	*23*	*24*	*25*

The dataset was split into three parts; separate training and test sets were used during development to train the models and choose when to stop the training

[2] https://github.com/tesseract-ocr/tesseract/issues/2738

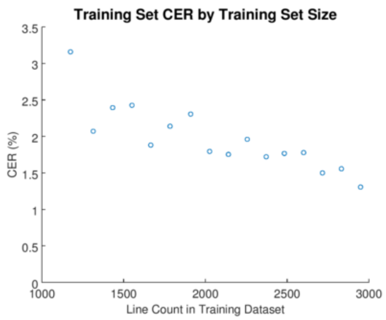


Figure 4. Training Set Character Error Rate

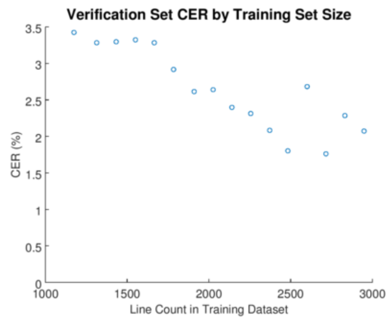


Figure 5. Verification Set Character Error Rate

process, while a separate verification set was used in final accuracy assessments for this paper.

It is important to note that we must evaluate not only the OCR accuracy of the pronunciation OCR. LVPPV entries contain both regular Latvian words and phonetic transcriptions and both categories need to have a high accuracy - it is not sufficient to obtain a correct pronunciation if it can not be automatically mapped to the proper dictionary word because of an OCR mistake in the spelling part.

5. Result Assessments

It was expected that (a) larger training size would correspond to a smaller error rate, and (b) increased character frequency would decrease the character's error rate.

Initially we trained some models for exploratory research to eliminate and fix any persistent errors in character set and input data. For the experimental data included in this paper, 16 models were trained with different amounts of training data, from 10 pages (1175 lines) to with 25 pages (2949 lines).

The analysis uses *character error rate* CER and *word error rate* WER to measure the quality of the trained models. Figures 5, 7 show that both rates fell with the increase in the training set. However, the trend is not consistent and has some significant outliers. Overall, error rate patterns in training and verification sets are similar, although, as expected, the training set has higher accuracy.

5.1. Character Level Analysis

On average, the total error rates fell which support expectation (a), however, character level errors exhibit a different pattern. An important hypothesis was the question of whether a particular character's error rate would decrease with the increase of its frequency in the training set.

Table 2 shows the character frequencies in the training data in the various experiments as more pages of training data were added. Additionally, by examining the verification set character errors (insertions, deletions, and substitutions) in the table 3 a few things can be noted:

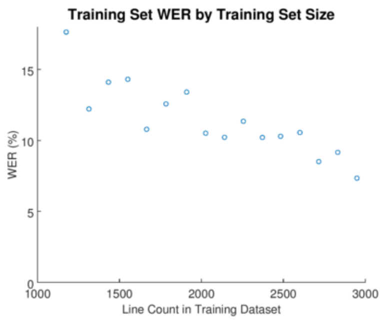


Figure 6. Training Set Word Error Rate

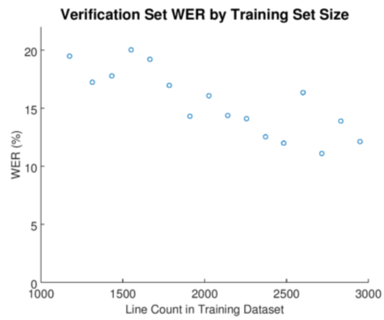


Figure 7. Verification Set Word Error Rate

Table 2. Character frequencies in the training set after the addition of n-th page to the dataset

N-th Page Added / Symbol	16	17	18	19	20	21	22	23	24	25
ì	228	231	250	252	262	269	270	270	373	407
î	132	134	142	147	152	152	154	164	183	193
ĩ	195	218	236	244	260	268	279	285	292	319
ê	269	277	296	303	305	311	314	346	353	358
ẽ	210	227	260	268	271	276	284	296	310	326
ȩ	105	105	106	107	108	108	109	127	128	128
è	102	104	104	113	115	118	119	124	125	125
ȩ̃	100	100	101	101	101	101	101	112	112	124
l̂	21	21	22	22	33	33	40	50	54	59
l̃	33	39	40	40	54	55	105	124	127	172
ã	180	196	202	227	234	260	284	288	288	296
â	100	101	103	104	106	109	110	116	124	125
à	8	8	8	12	12	13	13	13	23	23
à	71	73	74	103	104	106	106	109	110	116

(a) The general trend is towards fewer errors as training data increases, particularly for characters with a smaller diversity of diacritic marks like ”r”

(b) Larger frequency does not consistently correspond to fewer errors. For instance, the frequency of *letter l with tilde* almost doubled when page 22 was added, however, its error actually increased and the letter was consistently not being recognized.

(c) An inappropriately large frequency can cause ’overconfidence’. This can be seen in the last models for the letter ”ì” where it went from being missed to being overused in place of ”i”

As exemplified, the data illustrates an instability of the models towards different specific errors, as training a new model on a different, larger set of training data may result in a very different pattern of errors, possibly because of random initialization effects on the neural network model.

In this particular application, a significant subset of errors could be easily corrected in heuristic post-processing as some OCR output character combinations are not used in Latvian pronunciation. It might also be plausible to use a small statistical language model for this disambiguation, but due to technical difficulties (limited quantity of available data, and this would need to be a subword model for short character n-grams inside a single word) of integrating this modification, it was not attempted at this time.

Table 3. Most common character errors in verification set by training set size

Pages		14	15	16	17	18	19	20	21	22	23	24	25
OCR	**GT**												
è	ê	2	3	8	2	2	1	2	2	3	4	6	3
ẽ	ẽ̦	19	0	0	0	2	2	6	0	6	0	13	1
e	ȩ	9	5	2	3	5	6	3	1	5	4	11	6
ẽ	ê	4	9	4	15	6	6	7	3	27	7	4	1
ã	à	9	11	10	10	13	12	8	13	11	11	11	8
â	à	48	5	3	7	1	1	5	0	17	4	21	2
l̃	l	2	0	0	0	1	1	5	0	2	1	8	20
l̃	l̂	1	0	4	0	14	0	3	12	0	3	2	14
l	l̂	18	1	3	20	7	9	2	4	2	2	0	2
l̂	l̃	0	57	10	0	0	21	5	1	40	2	6	0
l	l̃	25	6	19	36	18	21	14	5	13	17	2	4
î	ì	15	4	4	12	10	1	3	5	1	1	1	1
ì	i	12	16	34	10	19	0	0	18	0	0	10	31
i	ì	29	28	19	21	22	34	46	27	37	21	19	13
CER		*3.28*	*2.92*	*2.61*	*2.64*	*2.40*	*2.31*	*2.08*	*1.80*	*2.68*	*1.76*	*2.29*	*2.07*

6. Conclusion

Our research demonstrates that freely available OCR models can be adapted for a new, specific set of diacritic markings with reasonable accuracy (2.07% character error rate) even with no availability of preexisting dictionaries or language models and only a small quantity of training data, 25 pages in our experiments.

However, our error analysis indicates that the training of Tesseract OCR models is not clear cut: more data might not necessarily mean better accuracy for any specific erroneous characters. Care should be given to the frequencies of characters in the training data to avoid detrimental effects, and a specific character's accuracy can fluctuate widely.

Whether these effects can be diminished by using other training methods could be explored by testing different OCR engines and their versions or experimenting with character frequencies in the training data.

The resulting OCR models will be applied to the full LVPPV data and, as the manual post-processing and proofreading work is finalized, result in up to 80000 phonetic transcriptions added to the publicly available lexical resources for Latvian. This data will facilitate the development of accurate speech synthesis and speech-to-text solutions for Latvian.

Acknowledgements

This research work was supported by the Latvian Council of Science, project "Latvian WordNet and word sense disambiguation", project No. LZP-2019/1-0464.

References

[1] Porīte T. Raģe S. Ceplītis L., Miķelsone A. *Latviešu valodas pareizrakstības un pareizrunas vārdnīca*. "Avots", 1995.
[2] Md Hasnat, Muttakinur Rahman Chowdhury, Mumit Khan, et al. Integrating bangla script recognition support in tesseract ocr. 2009.
[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
[4] Isabell Hubert, Antti Arppe, Jordan Lachler, and Eddie Antonio Santos. Training & quality assessment of an optical character recognition model for northern haida. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3227–3234, 2016.
[5] M. Pinnis and I. Auzina. Latvian text-to-speech synthesizer. In *Human Language Technologies - The Baltic Perspective*, volume 219. IOS Press, 2010.
[6] Ray Smith, Daria Antonova, and Dar-Shyang Lee. Adapting the tesseract open source ocr engine for multilingual ocr. In *Proceedings of the International Workshop on Multilingual OCR*, pages 1–8, 2009.
[7] A. Spektors, I. Auzina, R. Dargis, N. Gruzitis, P. Paikens, L. Pretkalnina, L. Rituma, and B. Saulite. Tezaurs.lv: the largest open lexical database for latvian. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*, 2016.
[8] Tesseract. How to use the tools provided to train tesseract 4.00. https://tesseract-ocr.github.io/tessdoc/TrainingTesseract-4.00. Accessed: 2020-04-03.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200624

Morfio – A Corpus-Based Perspective on Latvian Morphology

Michal ŠKRABAL[a,1], Pavel VONDŘIČKA[a] and Václav CVRČEK[a]
[a] *Institute of the Czech National Corpus, Charles University, Czech Republic*

Abstract. Our paper introduces Morfio, a corpus-based online tool for the study of derivation and morphological productivity. Originally, Morfio was created for Czech, in this paper, however, we would like to introduce its Latvian implementation. Apart from the tool description, we want to showcase its possibilities for describing Latvian morphology by way of several examples.

Keywords. Morfio, morphological base, formants, alternations

1. Introduction

This demo is a follow-up on the [1] paper presented at the last Baltic HLT conference in Tartu. New corpora for the Baltic languages and tools for exploiting these corpora were introduced in the paper: namely the Latvian component [2] of the InterCorp parallel corpus [3], [4] and Araneum Lettonicum [5], as well as two tools based on these corpora: the translation equivalents database Treq [6] and a word-sketch grammar for Latvian [7]. The current paper presents Morfio, a new tool adjusted for Latvian (and, hopefully, also for other languages, including Lithuanian, in the future).

Morfio is a corpus-based online tool for the study of derivation and morphological productivity available within the Czech National Corpus portal www.korpus.cz. It can be used to identify pairs (or triplets/quadruplets) of words which follow the same derivational pattern. This pattern is specified by a user using regular expressions in two ways: 1) "common parts" or the derivational *base* (i.e. parts which are common for both words) and 2) "distinct parts" or the derivational *formants* in which they differ (e.g. ***darb****s* – *no****darb****e*, the bold parts are shared, while the non-bold parts signal the differences). The tool substitutes the common parts with a wild card and searches the corpus for word pairs that a) share the common parts and, at the same time, b) differ in the way specified by distinct parts (using the example introduced above: the *Xs – noXe* pattern). This results in a list of word pairs having the same derivational relation (*darbs – nodarbe*, *gals – nogale*, *jums – nojume*, *kalns – nokalne*, *kars – nokare*, *laids – nolaide*, *rīts – norīte*, *vakars – novakare*, *vietns – novietne*, *zars – nozare*)[2] with their absolute frequencies in the relevant corpus (also see Figure 2). Furthermore, Morfio estimates the productivity of each word-formation pattern according to an index proposed by [8] (see Figure 3).

[1] Corresponding Author: Michal Škrabal, Institute of the Czech National Corpus, Panská 890/7, Prague, Czech Republic; E-mail: michal.skrabal@ff.cuni.cz
[2] The results are accessible within the Morfio tool at http://morfio.korpus.cz/Crb4KojP.

When conducting a derivational research on a corpus which is not semantically annotated, we have to stick to the semasiological approach, i.e. proceeding from the form to function/meaning. This can pose several problems (besides potentially inaccurate morphological annotation and/or lemmatization, also homonymy) whose solutions are outside the scope of this tool and require a thorough manual analysis carried out by linguists (also see Section 4). However, tools such as Morfio can help a researcher by sifting through a large amount of corpus data and identifying potentially relevant candidates for further analysis.

Originally, Morfio was created for Czech [10], [11], yet nothing prevents it from extending its functionality to other languages,[3] including the Baltic ones. For a fully-fledged non-Czech version of the tool, we had to implement configurability for different tagsets [12] and add an inventory of relevant vocal and consonant alternations (according to [13], [14]).

Figure 1. Morfio's main menu, with the inventory of relevant morphological alternations for Latvian

---

[3] In fact, Morfio has already been successfully applied to the Polish part of InterCorp [15]. We chose Latvian next because it is a morphologically rich language, yet a non-Slavic one.

2. Morfio Interface

After entering a valid query in the form, Morfio provides four types of results which are organized in separate tabs: Summary, List, Productivity and Pattern 1 (2, 3, 4).

2.1. Summary Tab

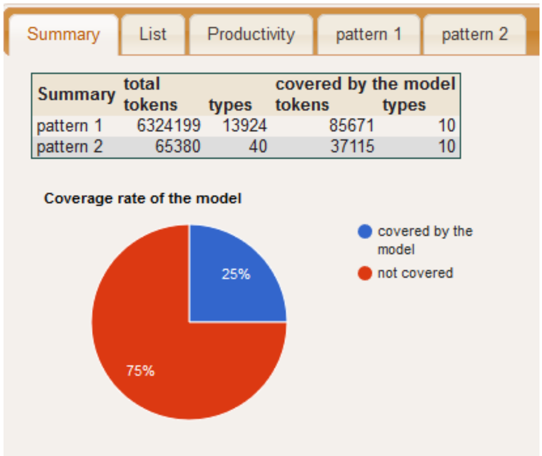
The Summary tab shows three types of information: number of types (with frequency above the limit specified by the user), sum of their occurrences, and an estimation of model completeness. One set of results (column "Total") refers to each of the isolated patterns itself, while the other set (column "Covered by the model") refers to those words following the given pattern which also fall into the analysed word-forming model, i.e. words for which a derivative counterpart was identified by the second pattern. The estimation of model completeness is based on the following assumptions:

a) Each pattern in the model identifies a certain number of items in the corpus (either wordforms or lemmas). We assume that most word-forming or derivational relations are asymmetric: there will be fewer words which are derived than those serving as derivation bases, as not every base produces a derivative. Thus, we can distinguish patterns that are basic in the model (those that include a larger number of items in the corpus) and those which are conditional (with a smaller number of items). In other words, the pattern which identifies a smaller set of word-types is considered a derivative of the pattern that identifies more word-types.

b) The completeness of the model is then calculated as the proportion of the total number of word pairs in the model to the total number of types of the conditional pattern, i.e. how many words in the less represented pattern find a derivative counterpart in the second pattern.

When inspecting our example model that can be characterized by a pair of words *darbs – nodarbe*, or by a pair of formants *-s* and *no-e* respectively, we can identify 10 lemma pairs involved in this word-forming process (see their list in Section 1 or in Figure 3). The first pattern (*Xs*) alone provides 13,924 different noun lemmas, while the second pattern (*noXe*) alone provides 40 different noun lemmas. Each of the patterns contains words that do not enter the model (e.g. the noun *vīrs* does not meet our requirements due to the non-existence of the noun **novīre*; similarly, we cannot find the noun **bīds* as a counterpart to the noun *nobīde*, etc.). The pattern with a smaller number of identified words (in our case, *noXe*) represents a greater limitation for the whole model than the pattern identifying more words (in our case, *Xs*). The condition for the existence of the word-forming relation specified by the example model is the existence of the noun ending with *-s*; however, the derivation process is limited mainly by the number of nouns following the pattern *noXe*, i.e. the pattern *noXe* is considered conditioned by the existence of the pattern *Xs*. In such a case, it makes sense to estimate the completeness of the proposed model according to how much the words of the conditional model contribute to it. In this case, the estimate is calculated as the ratio of the types of the pattern *noXe* that enter the model to all types of this pattern, i.e. 10/40, which corresponds to 25 %.

For word-formation analysis, it is obviously optimal if the coverage of the model is close to 100 %. This means that for all words in the conditional pattern, we have identified derivational bases in the other pattern. If such coverage cannot be achieved, the word-formation model explains only a part of the words formally defined by the conditional pattern; to relate the uncovered words with their bases, it is necessary to modify the existing model or create another, a complementary one.



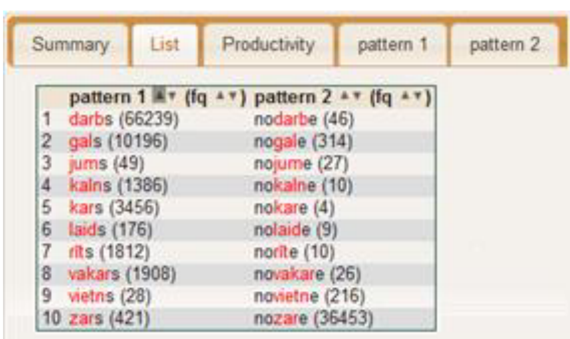
Summary	total tokens	types	covered by the model tokens	types
pattern 1	6324199	13924	85671	10
pattern 2	65380	40	37115	10

Figure 2. Summary tab

2.2. List Tab

The table in this tab lists all occurrences from all patterns that enter the specified model. The red part of the words indicates a common base (which may differ only if alternations are applied). The numbers in parentheses represent the total frequency of the lemma in the selected corpus. The table can be sorted according to any column using the arrows in the table header, both alphabetically and by frequency. At the same time, each word functions as a link to an example concordance in the selected corpus.

Pairs created only due to the application of alternation rules are highlighted by a coloured background (not shown in Figure 3). If more than one word corresponds to one pattern in a given pair (i.e. due to the application of alternations), all these words are listed collectively in one row of the table.



	pattern 1 (fq)	pattern 2 (fq)
1	darbs (66239)	nodarbe (46)
2	gals (10196)	nogale (314)
3	jums (49)	nojume (27)
4	kalns (1386)	nokalne (10)
5	kars (3456)	nokare (4)
6	laids (176)	nolaide (9)
7	rīts (1812)	norīte (10)
8	vakars (1908)	novakare (26)
9	vietns (28)	novietne (216)
10	zars (421)	nozare (36453)

Figure 3. List tab

2.3. Productivity Tab

The estimation of the productivity of both patterns and their mutual comparison is based on Baayen's theoretical remarks [8]. Morphological productivity is measured by estimating the increment of new types with the growing number of tokens for each pattern separately. The comparison shows which pattern is more productive, because the number of its types grows faster as new words are being created using its formants and, on the contrary, which pattern is less productive or potentially closed (albeit frequented and large).

Productivity in this approach can be understood as the total probability of all types of a given pattern that are not represented in the corpus. If such a probability is high for a pattern after examining a certain number of occurrences, it means that the pattern is productive; otherwise the pattern seems to be relatively closed. The total probability of unrepresented types for a given pattern can generally be calculated using the Good-Turing estimate [9], as the number of hapaxes related to the total number of tokens. In our case, hapaxes are those types that occur exactly once in a given pattern. If we plot the data of increasing number of types with the growing number of tokens for a given pattern, this total probability will be, as a consequence of the construction of the Good-Turing estimate, the slope of its tangent at the last point.

However, to compare patterns that are of unequal size, type and token data must be normalized. The results shown in the graph (see Figure 4) are thus normalized on both axes, for the median value of tokens and types, respectively. This means that a value of 1 for the normalized number of tokens on the x-axis represents the median for tokens of a given pattern, and, similarly, a value of 1 for a normalized number of types on the y-axis corresponds to the median for the number of different words.

In order to compensate for the influence of the order of texts in the corpus, the data are shuffled several times. The number of random permutations of concordance lines within the corpus is variable and ranges from one shuffle to a maximum of ten repeated randomization cycles.

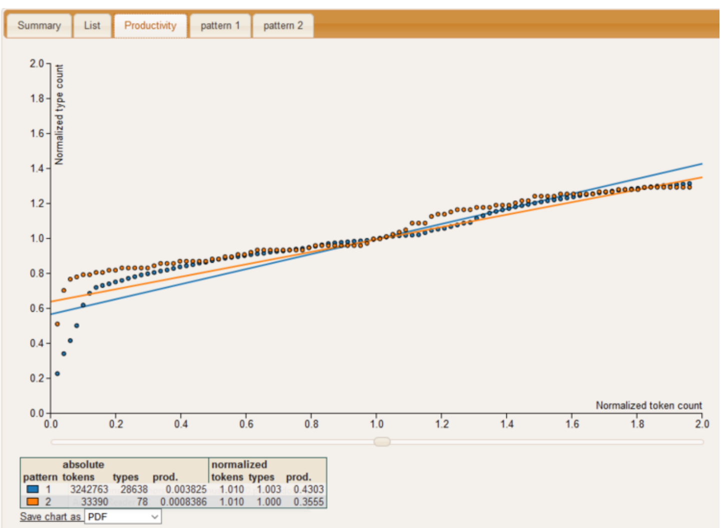


Figure 4. Productivity tab (Pattern 1 has a slightly higher slope and is, therefore, more productive than pattern 2.)

2.4. Patterns Tab

The words (wordforms or lemmas) corresponding to individual patterns are presented in the form of frequency lists in separate tabs. The list can also be supplemented with words that were not taken into account in the model, because their frequency was lower than the threshold set by the user. Data highlighted by a coloured background are involved in the word-forming model (i.e. there is a counterpart with the same base in the second pattern, differing only in formants).

The lists are mainly used to modify the model. If the list contains a word that that is not a part of the model (although it should be), it is signal for the user that it might be appropriate to change the model specification in order to increase its completeness and productivity.

The lists can be sorted in ascending or descending order, not only according to frequency, but also alphabetically, both commonly and retrogradely (i.e. from the end of the word). For better orientation in alphabetically sorted data, it is possible to turn on the grouping switch: the lines are then grouped according to the same start or end sequence of characters (see the left part of Figure 5). The number of grouping levels can be adjusted using the +/- element (each additional letter from the beginning or end, by which the words differ, can form another (sub)level for group division). For each group, data on the number of types and tokens appear, in total as well as those that participate in the word-formation model (shown in parentheses).

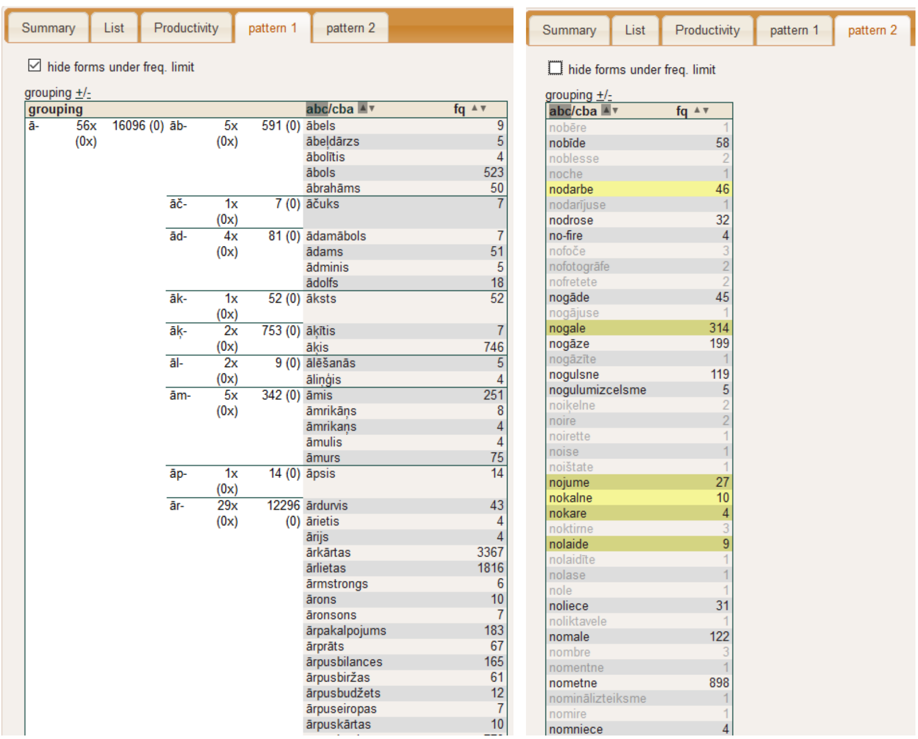


Figure 5. Pattern 1 and Pattern 2 tabs

3. Demo (Morfio-based Queries)

In the HLT Baltic demo session we aim to present the use of Morfio for the study of Latvian morphology. We use Morfio with the data from Araneum Lettonicum corpus [5] (over 671 M tokens)[4] to extract words involved in the following derivational models:

- prefixes/circumfixes: we are looking for triplets of 1) non-prefixed (*X*), 2) prefixed (*saX*) and 3) both prefixed and reflective verbs (*saXies*) with the same stem (lemmas, minimum frequency 5) – 504 types: *adīt* – *saadīt* – *saadīties* … *žņaugt* – *sažņaugt* – *sažņaugties*;[5]
- prefixoids: e.g. non-substantive lemmas *X* × *pašX* – 273 types: *aizdedzināties* – *pašaizdedzināties* … *zīmēt* – *pašzīmēt*;[6]
- suffixes: pairs of lemmas with the same stem, yet different suffix: e.g. nouns *Xums* × *Xība* (255 types: *absurdums* – *absurdība* … *žultainums* – *žultainība*)[7] or adjectives *Xains* × *Xīgs* (24 types: *acains* – *acīgs* … *zīdains* – *zīdīgs*);[8]
- alternations: feminines of the 5th declination class and their (non-)alternation of the stem in genitive plural form – 1565 types: *ābece* – *ābeču* … *žubīte* – *žubīšu*;[9]
- noun diminutives ending with both formants *-iņa* and *-ele* (184 types: *acs* – *aciņa* – *ačele* … *zupe/zupa* – *zupiņa* – *zupele*) or *-iņš* and *-elis* respectively (109 types: *auns* – *auniņš* – *aunelis* … *žurnālists* – *žurnālistiņš* – *žurnālistelis*);[10] adjective diminutives *X* × *Xiņš* (only 2 types: *kluss* – *klusiņš*; *mazs* – *maziņš*).[11]

4. Limits and Advantages of Morfio

It goes without saying that Morfio – as a tool based solely on analysis of the form and ignoring the meaning of the words – cannot produce error-free and ready-to-use results without the need for further manual inspection. The tool focuses on providing maximal *recall* by pre-processing a large amount of data and yielding a list of morphologically related candidates, making analysis faster and more accessible for researchers. Relevance of results (*precision*) is left solely to the judgment of the user: i.e. to the actual query formulation and the subsequent interpretation of the findings.

Yet, these data are hardly accessible by a linguist's introspection, and, especially in some cases, a corpus-driven approach is the only possible way to obtain them. The

---

[4] The Czech-Latvian components of the parallel corpus InterCorp (IC) [2] are another two searchable datasets, unfortunately, their size is still quite small (v9 – 40,6 M tokens, v12 – 32,7 M tokens). The size of a corpus will understandably affect the size of the results. E.g. the list for the above-mentioned pattern *Xs* – *noXe* is almost 20 times bigger in Araneum Lettonicum (with the same frequency threshold of 3), yielding 186 word pairs, although the precision itself decreases significantly. See http://morfio.korpus.cz/EWWr9pfi for the results of the query.

[5] http://morfio.korpus.cz/Bv2wcPQT; cf. 69 types for ICv12 (http://morfio.korpus.cz/TfyBMwBs).

[6] http://morfio.korpus.cz/9wQcQKUM; cf. 19 types for ICv12 (https://morfio.korpus.cz/Pa9kBXWE).

[7] http://morfio.korpus.cz/kRH8xniz; cf. 41 types for ICv12 (http://morfio.korpus.cz/OMUCmsbD).

[8] http://morfio.korpus.cz/KLSrOvJH; cf. 0 types for ICv12 (http://morfio.korpus.cz/JK0CGAIx).

[9] http://morfio.korpus.cz/mt11TzQg; cf. 253 types for ICv12 (http://morfio.korpus.cz/RBtmCGjb).

[10] http://morfio.korpus.cz/CvHsAxq9; cf. 6 types for ICv12 (http://morfio.korpus.cz/LaKgeXWg) and http://morfio.korpus.cz/dseYxsxC; cf. 3 types for ICv12 (http://morfio.korpus.cz/oQzn3MVS)

[11] Cf. 1 type (*mazs* – *maziņš*) for ICv12 (http://morfio.korpus.cz/aEvOVVdW).

frequency of word pairs gives an overall idea about the productivity of the respective phenomena in the contemporary Latvian lexicon and may differ significantly from existing descriptions of Latvian.

Acknowledgements

This work has been supported by the European Regional Development Fund-Project "Creativity and Adaptability as Conditions of the Success of Europe in an Interrelated World" (No. CZ.02.1.01/0.0/0.0/16_019/0000734). During its creation we used the tools developed within the Czech National Corpus project (LM2015044) funded by the Ministry of Education, Youth and Sports of the Czech Republic within the framework of Large Research, Development and Innovation Infrastructures.

References

[1] Škrabal M, Benko V. Czech & Slovak Corpus Resources Go (not only) Latvian. In: Muischnek K, Müürisep K, editors. Frontiers in Artificial Intelligence and Applications. Proceedings of the 8th International Conference Baltic HLT 2018. Amsterdam: IOS Press; 2018. p. 158-165.
[2] Lazar M, Škrabal M, Vavřín M. Korpus InterCorp – lotyština, verze 12 z 12. 12. 2019. Praha: Ústav Českého národního korpusu FF UK; 2019. Available at: http://www.korpus.cz.
[3] Čermák F, Rosen A. The case of InterCorp, a multilingual parallel corpus. International Journal of Corpus Linguistics 2012;13(3):411-427.
[4] Rosen A. InterCorp – a look behind the façade of a parallel corpus. In: Gruszczyńska E, Leńko-Szymańska A, editors. Polskojęzyczne korpusy równoległe. Polish-language Parallel Corpora. Warszawa: Instytut Lingwistyki Stosowanej; 2016. p. 21-40.
[5] Benko V. Aranea: Yet Another Family of (Comparable) Web Corpora. In: Sojka P, Horák A, Kopeček I, Pala K, editors. Text, Speech and Dialogue. 17th International Conference, TSD 2014, Brno, Czech Republic, September 8-12, 2014. Proceedings. LNCS 8655. Springer International Publishing Switzerland; 2014. p. 257-264.
[6] Škrabal M, Vavřín M. The Translation Equivalents Database (Treq) as a Lexicographer's Aid. In: Kosek I et al., editors. Electronic lexicography in the 21st century. Proceedings of eLex 2017 conference. Leiden: Lexical Computing CZ s. r. o.; 2017. p. 124-137.
[7] Benko V. Compatible Sketch Grammars for Comparable Corpora. In: Abel A, Vettori C, Ralli N, editors. Proceedings of the XVI EURALEX International Congress: The User in Focus. 15-19 July 2014. Bolzano/Bozen: Eurac Research; 2014. p. 417-430.
[8] Baayen H. Quantitative aspects of morphological productivity. In: Booij GE, van Marle J, editors. Yearbook of Morphology 1991. Dordrecht: Kluwer Academic Publishers; 1992. p. 109-149.
[9] Good IJ. The Population Frequencies of Species and the Estimation of Population Parameters. Biometrika. 1953;40:237-264.
[10] Cvrček V, Vondřička P. Morfio. Praha: Ústav Českého národního korpusu FF UK; 2013. Available at: http://morfio.korpus.cz.
[11] Cvrček V, Vondřička P. Nástroj pro slovotvornou analýzu jazykového korpusu. In: Gramatika a korpus 2012. Hradec Králové: Gaudeamus; 2012.
[12] Paikens P, Rituma L, Pretkalnina L. Morphological analysis with limited resources: Latvian example. In: Oepen S, Hagen K, Johannessen JB, editors. Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA) 2013. Linköping: Linköping University Electronic Press; 2013. p. 267-277.
[13] Auziņa I et al. Latviešu valodas gramatika. Rīga: LU Latviešu valodas institūts; 2015.
[14] Laua A. Latviešu literārās valodas fonētika. Rīga: Zvaigzne ABC; 1997.
[15] Zasina AJ. Konkurence koncovek -a a -u v genitivu singuláru neživotných maskulin v polštině. In: Stluka M, Škrabal M, editors. Liſka a czban – Sborník příspěvků k 70. narozeninám prof. Karla Kučery. Praha: NLN; 2017. p. 90-98.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200625

Development and Research in Lithuanian Language Technologies (2016-2020)

Andrius UTKA [a,1], Jurgita VAIČENONIENĖ [a], Monika BRIEDIENĖ [b] and Tomas KRILAVIČIUS [b]

[a] *Vytautas Magnus University, Lithuania*
[b] *Vytautas Magnus University and Baltic Institute of Advanced Technology, Lithuania*

Abstract. The paper presents an overview of the development and research in Lithuanian language technologies for the period 2016-2020. The most significant national and international LT related initiatives, projects, research infrastructures, language resources and tools are discussed. The paper also surveys research production in the field of language technology for the Lithuanian language. The provided analysis of scientific papers shows that machine translation and speech technologies were the most trending research topics in 2016-2019.

Keywords. Lithuanian language, language technology research, language resources and tools, research infrastructures

1. Introduction

The scientific paradigm of *language technology* (LT) is changing in the world, intelligent technologies are developing at an incredible rate, robotisation and the Internet of Things are emerging. The amount of data is increasing exponentially and will soon reach hundreds of zeta bytes [1]. Most of the technologies are designed for the wider-used languages such as English, Chinese, Spanish, Arabic, or German, and we may observe groundbreaking achievements and breakthroughs in all complex language technology areas: machine translation, speech recognition and synthesis, dialogue systems, and natural language processing, among others.

On the other hand, the European Parliament resolution on “Language Equality in the Digital Age” adopted on September 11, 2018 [2] highlights that there is a “widening technology gap between well-resourced languages and less-resourced languages” maintaining that “European lesser-used languages are at a significant disadvantage on account of an acute lack of tools, resources and research funding” [2]. It is, therefore, important for smaller and less resourced languages to assess the current situation correctly and set up adequate goals for the future.

The state of art of LT in Europe and the Baltic states has been overviewed by [3, 4]. Discussion on language resources and technologies in Lithuania (2012-2015) was presented in [5], while this paper focuses on the landscape of human language technology developments in Lithuania, in 2016-2020. We survey Lithuania’s involvement in inter-

[1]Corresponding Author: Andrius Utka; Vytautas Magnus University; E-mail: andrius.utka@vdu.lt.

national and national LT related initiatives and research infrastructures, and overview the advancements in language resources and technologies, as well as key projects and research.

2. Language Technology Related Initiatives and Projects

"Guidelines on the Development of Lithuanian Language in Digital Environment and Advancements in Language Technologies (2021-2027)", adopted by the State Commission of Lithuanian Language, provide a thorough overview of the European and Lithuanian strategic documents, funding instruments and institutions regulating LT development in the country [6]. The main goal of the guidelines is to overview and facilitate the full use of the Lithuanian language in the digital environment. Drawing on the information provided in this as well as other relevant documents, this section will briefly highlight the major initiatives, projects and developed language resources.

Over the course of the last five years, Lithuania became involved in various European LT initiatives such as European Federation of National Institutions for Language (EFNIL)[2], European Language Resource Coordination (ELRC)[3], European Open Science Cloud (EOSC)[4], and FAIR data[5]. In 2019, together with other European countries, Lithuania signed the "EU Declaration on Cooperation on Artificial Intelligence" [7] and prepared "Lithuanian Artificial Intelligence Strategy" [8], which highlights the necessity of language technologies for the development of economy and science. Increasing international cooperation and researcher mobility are also evidenced by the participation of Lithuanian higher educational institutions in language technology related COST actions (e.g., IC1408, IC1207, CA18209, IC1002, CA18209)[6].

In the context of the most significant projects and their results, "The Lithuanian Information Society Development Programme 2014-2020" [9], funded by the EU Structural Funds, needs to be emphasized. The programme has launched 5 large-scale projects that develop language technology solutions and services in different areas: "Lithuanian Language Speech Services (LIEPA 2)", "The Information System for Syntactical and Semantic Analysis of the Lithuanian Language (SEMANTIKA 2)", "Integrated Information Systems of the Lithuanian Language and Language Resources (Raštija 2)", "Modernization and Development of Machine Translation Systems and Localization Services", and "Lithuanian Language Resources (E.kalba)". The following language technology resources have been created while implementing the projects: syntactically annotated corpus (ALKSNIS)[7], morphologically annotated corpus (*gold standard*) (MATAS)[8], 1,000 hour speech corpus, Internet corpus (BIT), as well as a number of modernized corpora and lexicons in the E.kalba project. The developed tools and services include: neural MT system (Lithuanian, German, Polish, French, and Russian language pairs), automatic transcription of speech files, speech recognition of computer

[2]http://www.efnil.org/
[3]http://www.lr-coordination.eu/
[4]https://www.eosc-portal.eu/
[5]https://www.go-fair.org/fair-principles/
[6]https://www.cost.eu/
[7]https://clarin.vdu.lt/xmlui/handle/20.500.11821/21
[8]https://clarin.vdu.lt/xmlui/handle/20.500.11821/33

commands, Lithuanian language synthesizer, advanced internet search, automatic text summarisation, and hate speech detection.

The opportunities provided by EU Structural Funds, specifically, instruments encouraging companies to invest in research and experimental development (R&D) of innovative products (e.g., the instruments "Inočekiai", "Intelektas", "Eksperimentas") are also exploited. A number of language technology projects were implemented by the Baltic Institute of Advanced Technologies, Vilnius University, and Vytautas Magnus University in cooperation with JSC Amberlo, JCS Tilde Information Technology, and other companies.

On a smaller scale, the Research Council of Lithuania, responsible for monitoring national science development and research funding, has financed six language technology projects for the period 2016-2020, which resulted in new language resources, tools and scientific publications. We should mention here the project PASTOVU[9] in which tools and methodology for the extraction of Lithuanian multi-word expressions were created and the project "Modern Spoken Lithuanian"[10] for modernizing a searchable spoken language corpus.

Important for language services and technologies are private business initiatives such as open frameworks and tools that can be tested and adapted for less resourced languages. A crucial impact on the development of LT is made by world business leaders (e.g., *Google*, *Microsoft*, *Facebook*, *Amazon*, *IBM*, etc.), as well as by data collection initiatives such as *Mozilla Common Voice* and *Glosbe*. The largest language technology companies established the *LT-Innovate* language technology industry association, where they can share ideas and develop strategic solutions. Lithuanian business initiatives can be exemplified by the Lithuanized *SpaCy*[11] library developed by *JSC TokenMill*, and Lithuanian speech recognition, speech synthesis[12], and machine translation[13] demo online services created by *JSC Tilde*.

3. Language Technology Infrastructures

As one of the measures which could contribute to decreasing the LT breech among wider and lesser used languages is the role of EU-funded research networks such as FLaReNet, CLARIN, HBP and META-NET [2]. Lithuania began joining international research infrastructures (RIs) in 2013. In the report by the Research Council of Lithuania, the Research and Higher Education Monitoring and Analysis Centre (MOSTA), and the Ministry of Education, Science and Sport, it is agreed that Lithuania's participation in RIs has to be based on strategic goals of the state and long-term sustainability of RI [10]. In other words, RI should be relevant at the national level, ensure scientific excellence and effective governance, have sufficient numbers of users, long-term financing and technological development. Participation in RIs for research progress and breakthrough in human language technologies also corresponds with the strategic aims of such legislation as [6], [9], [10], [11], [12], [13], [14]. In line with this, Lithuania has so far joined two

[9] http://mwe.lt/en_US/

[10] http://sakytinistekstynas.vdu.lt/

[11] https://spacy.io/models/lt

[12] https://www.tilde.lt/snekos-technologijos

[13] https://translate.tilde.com/en

international language technology related initiatives: Common Language Resources and Technology Infrastructure (CLARIN ERIC) and the European Language Grid (ELG).

Lithuania became a full member of the European Research Infrastructure for Language Resources and Technology CLARIN ERIC in 2014. At present, CLARIN-LT is a consortium of five institutions, which maintains a repository[14] and provides open access (under the academic, public and restricted licenses) to specialized and well-annotated language resources used by language researchers, teachers and students of various Lithuanian higher educational institutions. Besides, CLARIN-LT centre is actively involved in knowledge sharing activities. In 2020, CLARIN-LT became a member of the CLARIN Knowledge Centre for Systems and Frameworks for Morphologically Rich Languages (SAFMORIL) coordinated by the University of Helsinki[15] by committing to share the knowledge in corpus linguistics and natural language processing methods for Lithuanian.

The maturity of the infrastructure has been acknowledged by the Research Council of Lithuania which recommends including CLARIN-LT into the renewed "Lithuanian Research Infrastructures Roadmap for 2020-2023". In the nearest future, CLARIN-LT aims to achieve the Service Providing Centre (CLARIN B centre) certification, especially important for the full integration into the international infrastructure. CLARIN B centres guarantee sustainable storage of language tools, resources and open access to other CLARIN services for various research communities which would increase the visibility and uptake of Lithuanian contribution to LT development on national and international levels. Lithuania's membership fee for CLARIN ERIC paid by the Ministry of Education, Science and Sport is ensured until 2021, thus, the continuity of CLARIN-LT activities is largely dependent on the strategic decisions of the ministry.

In 2019, the Institute of Lithuanian Language signed a subcontract with the EU-funded project European Language Grid (2019-2021) becoming one of the National Competence Centres of the network[16]. By establishing a scalable cloud platform, ELG aims to become the leading platform for Language Technology in Europe offering both commercial and non commercial LT communities to store, use and promote their services. Currently, there are 32 National Competence Centres responsible for implementing the successful operation of ELG and pursuing the main goals. Being in their active stage of establishment, the Lithuanian National Competence Centre provides information on the national level about the ELG consortium, the European Language Grid cloud platform and organizes knowledge sharing events.

In addition to joining international initiatives, three national (Raštija LT[17], LKS-SAIS[18], E.kalba[19]) infrastructures and a few smaller LT portals were launched or modernized in 2016-2020. Services offered by international and national research and LT related infrastructures are increasingly being integrated in university studies, distance learning, and development of new technologies.

Moreover, the role of RIs in the present AI hype cannot be underestimated as in order to efficiently employ new machine learning methods, massive amounts of accessible and

[14] https://clarin.vdu.lt/xmlui/?locale-attribute=en
[15] https://www.kielipankki.fi/safmoril/
[16] https://www.european-language-grid.eu/
[17] raštija.lt
[18] semantika.lt
[19] ekalba.lt

quality language data are needed. Further development and modernization of RIs which may ensure better language data storage and sharing conditions are especially important for lesser used languages that strive to be visible in the digital space.

4. Language Technology Research

This section surveys different research topics in the field of language technology for Lithuanian. We have collected and analyzed papers and studies on language technology research for 2016-2019. We do not include publications that were published in 2020, as the data would only present partial information.

Figure 1 illustrates the trends, i.e. more and less popular topics in language technology research for Lithuanian as well as the progression of the topics by year. The information was retrieved from the major subscribed databases (such as arXiv, Google Scholar, IEEE Xplore, Mendeley, Scopus, Semantic Scholar, SpringerLink, Web of Science, VDU DSpace/CRIS). The search was organized in two ways: verification by entering the surnames of well-known Lithuanian language technology specialists and by using the basic terms (e.g., Lithuanian/ Lithuanian language + language technologies/ corpora/ speech/ lexical/ morphological/ multiword/ authorship/ chatbot/ wordnet/ embeddings/ media/ NER/ NLP/ NLG/ NLU/ classification/ clusterization) with AND operator). In total, 91 publications were retrieved, which were then grouped thematically according to their content and keywords. The full list of the bibliography is accessible from the CLARIN-LT repository[20]. It should be noted that the collected data includes publications not only by Lithuanian researchers, but all work on Lithuanian language technology for the discussed period.

In terms of categorization, we avoided overgeneralizing labels so that the specificity of the LT field would be reflected. Naturally, the presented taxonomy of language technology research topics is not absolute and many other classification schemes are possible. Also, the same paper is ascribed to several categories in some cases (for example, corpora and morphological analysis), thus Figure 1 does not reflect the exact number of papers in each category, but rather the thematic scope of research distributed across all papers.

Judging by the number of scientific papers for 2016-2019 in Figure 1, the most researched topic is machine translation (12 papers), where one of the languages is Lithuanian. A possible reason for this trend is that despite a considerable progress in machine translation achieved as a result of neural machine translation developments, expectations for MT quality continue to increase and numerous experimentation attempts are done testing different NMT frameworks for many languages (including Lithuanian). As a result, presently, MT research for the Lithuanian language is mostly conducted by international groups.

Other popular research topics are: traditional corpus-based research (9), word embeddings (7), analysis of multiword expressions (7), media monitoring (7), stylometry (6), morphological analysis (6), and authorship classification problems (6), followed by speech synthesis (5), speech recognition (5), and language technology overviews (5). However, if we combine speech-related topics of speech synthesis, speech recognition,

[20]http://hdl.handle.net/20.500.11821/38

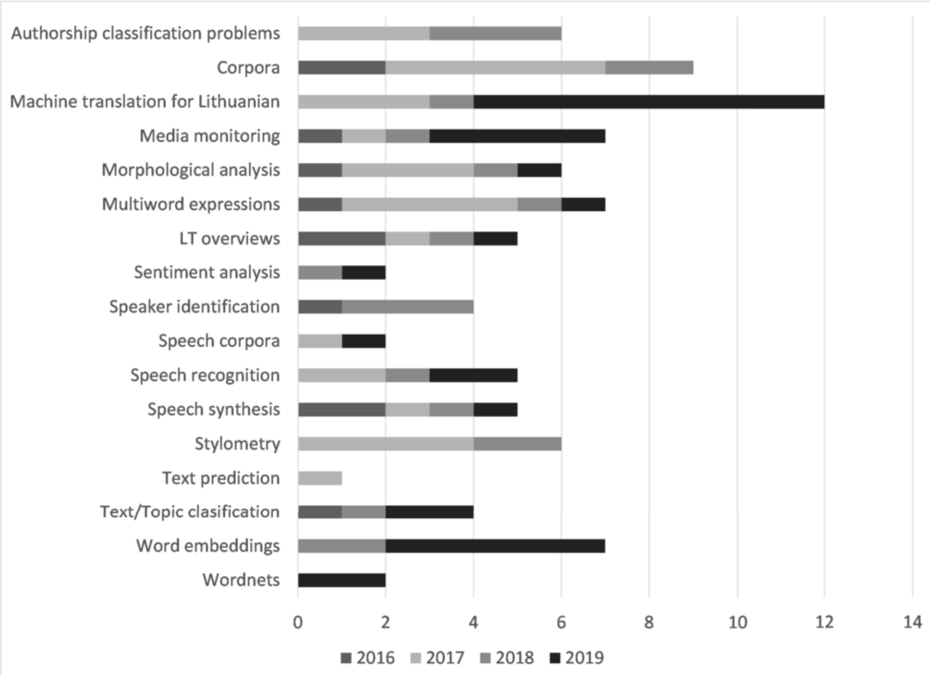


Figure 1. Research related to language technology for Lithuanian (2016-2019)

speaker identification, and speech corpora into one group of speech technologies, then this group becomes the most popular topic in this period with 18 publications.

It has also been observed that the rise of deep learning techniques created an increasing demand for specially designed data. Thus, beside traditional data sets such as treebanks, news corpora, speech corpora, spoken language corpora, or lexical databases, considerable research has been carried out on developing and implementing different word embeddings for Lithuanian as reflected in scientific publications on the topic.

It should be noted that even though several research groups and companies are involved in developing practical applications for the Lithuanian language in such worldwide trending areas as natural language generation, natural language understanding, automatic summarization, and chatbots, very few scientific research papers have been published on these topics.

In terms of methodology, we found out that the number of publications which apply deep learning and other advanced machine learning techniques has surpassed the number of publications on traditional, symbolic and rule-based approaches (53 % vs 47 %). The increasing use of machine learning methods can be highlighted as one of the achievements of the analyzed period.

As to the limitations of this survey, although we tried to include all relevant LT papers, naturally, not all publications might have been identified. Still, we hope that this overview provides a general insight on the changing trends of LT research for Lithuanian, determined by societal changes, financed projects and their aims, private sector initiatives and other factors.

5. Conclusions

The paper has shown that the last five years in Lithuania were productive in LT-related policies, infrastructure development, projects and cooperation initiatives both nationally and internationally.

Developments contributing to the integration of Lithuanian as less resourced language in the digital environment are as follows: inclusion of language technology problems into strategic documents and legislation, developing national and international language technology infrastructures, promoting the international visibility of language resources or technologies adapted to or created for the Lithuanian language, implementing Lithuanian language services in large scale projects, offering knowledge sharing services to interested parties, developing open access possibilities of language technologies and resources, and pursuing advanced research goals.

The presented review of scientific publications has shown that machine translation and speech-related research were the most researched topics in 2016-2019. In the Lithuanian language technology research, similarly to other European countries, we can see increasing number of publications that apply innovative machine learning methods for language analysis.

References

[1] European Commission. White paper on artificial intelligence–a European approach to excellence and trust. 2020. Available from: https://ec.europa.eu/info/sites/info/files/commission-white-paper-artificial-intelligence-feb2020_en.pdf.

[2] European Parliament. European Parliament resolution of 11 September 2018 on language equality in the digital age. 2018. Available from: https://www.europarl.europa.eu/doceo/document/TA-8-2018-0332_EN.html.

[3] Skandina I. Some Highlights of Human Language Technology in Baltic Countries. In: Databases and Information Systems X: Selected Papers from the Thirteenth International Baltic Conference, DB&IS 2018. vol. 315. IOS Press; 2019. p. 18.

[4] Rehm G, Marheinecke K, Hegele S, Piperidis S, Bontcheva K, Hajič J, et al. The European Language Technology Landscape in 2020: Language-Centric and Human-Centric AI for Cross-Cultural Communication in Multilingual Europe. arXiv preprint arXiv:200313833. 2020.

[5] Utka A, Amilevičius D, Krilavičius T, Vitkutė-Adžgauskienė D. Overview of the Development of Language Resources and Technologies in Lithuania (2012-2015). In: Human Language Technologies-the Baltic perspective: Proceedings of the 7th International Conference, Baltic HLT 2016, Riga; 2016. p. 12–19.

[6] Lietuvos Respublikos Seimas. Seimo nutarimo "Dėl Lietuvių kalbos plėtros skaitmeninėje terpėje ir kalbos technologijų pažangos 2021–2027 m. gairių patvirtinimo" projektas + gairės; 2020. Available from: https://e-seimas.lrs.lt/portal/legalAct/lt/TAP/ee468a02945611eaa51db668f0092944?positionInSearchResults=0&searchModelUUID=2a0d20bc-b1f2-4573-a67a-78f9e820afdb.

[7] European Commission. EU Declaration on Cooperation on Artificial Intelligence. 2018. Available from: https://ec.europa.eu/jrc/communities/en/node/1286/document/eu-declaration-cooperation-artificial-intelligence.

[8] Ministry of the Economy and Innovation of the Republic of Lithuania. Lithuanian Artificial Intelligence Strategy. 2018. Available from: http://kurklt.lt/wp-content/uploads/2018/09/StrategyIndesignpdf.pdf.

[9] Government of the Republic of Lithuania. Information Society Development Programme for 2014-2020 'Digital Agenda for the Republic of Lithuania'. 2014. Available from: https://eimin.lrv.lt/uploads/eimin/documents/files/30310_LRV%20nutarimas(en).pdf.

[10] LMT. Lietuvos MTI kelrodis; 2015. Available from: https://www.lmt.lt/lt/mokslo-politika/mokslinių-tyrimu-infrastrukturos/lietuvos-mti-kelrodis/2358.

[11] Ministry of Finance of the Republic of Lithuania. Operational Programme for the European Union Funds' Investments in 2014-2020; 2014. Available from: https://ec.europa.eu/regional_policy/en/atlas/programmes/2014-2020/lithuania/2014lt16maop001.

[12] Research Council of Lithuania. Directions for the Lithuanian Studies Research Development 2012-2020; 2012. Available from: https://www.lmt.lt/en/doclib/ujv8xc7kauxwnnp3e6r7dw5dxwqdy6hq.

[13] Ministry of Education and Science of the Republic of Lithuania. State Lithuanian Studies and Dissemination Programme for 2016-2024; 2015. Available from: https://www.lmt.lt/en/research-commissioned-by-the-state/state-lithuanian-studies-and-dissemination-programme-for-2016-2024/803.

[14] Parliament of the Republic of Lithuania. Guidelines for the State Language Policy 2018-2022; 2018. Available from: https://e-seimas.lrs.lt/portal/legalAct/lt/TAD/cd0584707b6e11e89188e16a6495e98c?positionInSearchResults=0&searchModelUUID=2fa062c8-0d9b-4b80-9a44-8938b12fe0a4.

Language Learning Resources

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200627

Quantitative Analysis of Language Competence *vs.* Performance in Russian- and Lithuanian-Speaking 6 Year-Olds

Ingrida BALČIŪNIENĖ [a,b1] and Aleksandr N. KORNEV [b]
[a] *Vytautas Magnus University, Lithuania*
[b] *Saint-Petersburg State Pediatric Medical University, Russia*

Abstract. The paper deals with a comparative analysis of the *Part-of-Speech Profile* between different languages and discourse genres in 6-year-old typically developing Russian- *vs*. Lithuanian-speaking children. Results of the study inspire a discussion on a possibility to evaluate both language competence and language performance of the same subject on the basis of his/her distribution of parts of speech in the discourse.

Keywords. Corpus linguistics, language acquisition, part of speech, language competence, language performance

1. Introduction

Among numerous studies in child language, the development of separate linguistic patterns such as morphology and morphosyntax of some parts of speech as well as morphological derivation and compounding has been well described both in Russian and Lithuanian. In Russian, the essential longitudinal studies [1, 2, 3, 4, 5, 6, 7] have focused on the acquisition of nouns, adjectives, pronouns, and verbs. In Lithuanian, longitudinal studies have been devoted to the acquisition of nouns [8], verbs [9], and adjectives [10]. Some comparative studies [11, 12] should also be noted. Much less is still known about relationships between different parts of speech (PoSs) along the developmental course and their role in the acquisition of discourse skills. [13] discussed the effect of grammatical, lexical, and pragmatic categories on the mean length of utterance (MLU) rate. Authors proposed that during the 2nd-3rd years of life, function words play the central role in the syntactic development. Verbs appeared to be especially important for the late syntactic development [14, 15, 16], whereas nouns are important for the nominal function development. The acquisition of PoSs means "…knowing how to use the word in the language. The grammatical category of a word determines (1) the position it is allowed to occupy in the clause /…/; (2) the range of syntactic functions it can occupy /…/; (3) the types of words with which it co-occurs /…/; (4) the types of morphemes it requires or accepts /…/" [17: 434].

[1] Corresponding Author: Ingrida Balčiūnienė; Department of Lithuanian Studies, Vytautas Magnus University, V. Putvinskio st. 23-206, LT-44243 Kaunas, Lithuania; E-mail: ingrida.balciuniene@vdu.lt.

As Slobin proposed, "…child acquires more than a system of grammatical forms and semantic/communicative functions. In acquiring the grammar of a particular language, the child comes to adopt a particular framework for schematizing experience" [18: 7]. In different languages, categorical/syntactic function of word plays different roles in entity assignment. On the other hand, to become a proficient native speaker, a child has to learn language-specific rhetorical style which, in turn, influences lexical and morphosyntactic features of discourse [19]. Speakers of two different languages will organize the same reality in slightly different ways and, thus, they will employ the PoSs in some different proportions.

The **aim** of our study was to compare PoS distribution in the discourse of Russian-speaking children and their Lithuanian-speaking peers. The point of our interest was to analyze quantitively the PoS distribution from both static (language knowledge/competence) and dynamic (language behavior) perspectives in different genres. It was hypothesized that *lemma distribution* to more extent reflects the *language competence* of a subject, while *word token distribution* is more sensitive to *language behavior* demands in the given discourse context.

Among various quantitative approaches to corpus data, the distributive PoS-analysis should reveal some syntactic pattern information [20, 21, 22]. Following Lyashevskaya, the "grammatical behavior" of language units in corpus data manifests in the item distribution in a context. This is relevant to PoS Profile (PoSP), i.e. the distribution of word types [23: 7].

2. Methodology

For this comparative quantitative study, we accessed two corpora of child language. The *Corpus of Lithuanian Children Language* has been developed at Vytautas Magnus University and comprises morphologically annotated longitudinal and semi-experimental data (~106 hours) of the Lithuanian L1 development [24]. The *Corpus of Russian Children Language* has been compiled at Saint-Petersburg State Pediatric Medical University and comprises morphologically annotated semi-experimental data (~65 hours) of Russian L1 development [25].

For this study, we selected 24 typically developing (TD) 6-year-olds and analyzed their PoSs in different discourses (Table 1).

Table 1. The data

Subjects	Russian TDs (n = 12)	Lithuanian TDs (n = 12)
Morphologically annotated transcripts:		
Fictional narratives	2466 word tokens	2975 word tokens
Conversational dialogues	3074 word tokens	13279 word tokens

Namely, we selected (1) narratives told by the subjects according to the picture sequence and (2) conversational dialogues. As for narratives, Lithuanian children told stories according the *Cat Story* picture sequence developed by [26]; Russian children told stories according the sequence slightly modified in the framework of the COST Action IS0804 (http://bi-sli.org). Conversational dialogues were elicited in a slightly different way: Russian children were asked to answer 10 comprehensions about the story they told. Lithuanian children were not controlled for story comprehension. Their conversational dialogues were based on brief semi-structured interviews about the daily activities at the kindergarten.

Word tokens included only words and excluded punctuation marks, symbols, and acronyms. Linguistic disfluencies, such as hesitations, incomplete/revised words, were also excluded from the analysis (on linguistic disfluencies, see [27]). Morphological multiwords (such as Lithuanian *vos ne vos* 'hardly', *iš tikrųjų* 'in fact' or Russian *kak budto* 'as it were' *vse ravno* 'even so') were analyzed as entire units (on morphological multiword units, see [28]). All word tokens were lemmatized by means of the CLAN [29]. During the analysis, all the children (a) word tokens and (b) lemmas were classified into PoSs. The distribution of them was compared from the perspective of the language (Lithuanian *vs.* Russian) and the genre (narrative *vs.* dialogue).

3. Results

3.1. PoSP in Different Genres in Russian-speaking Children

The between-genre comparison of *word token distribution* in Russian-speaking children revealed multiple distinctions. The majority of PoSs (with the exception of adjectives, participles and prepositions) significantly discriminated narratives from conversations (Figure 1).

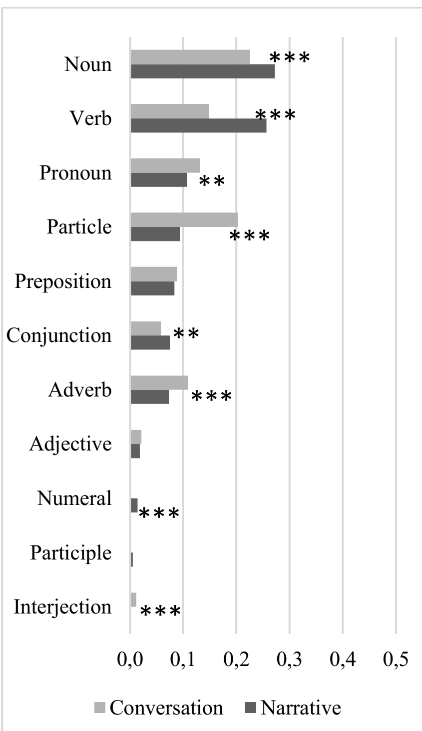


Figure 1. Distribution of *PoSs (word tokens)* in Russian-speaking discourse

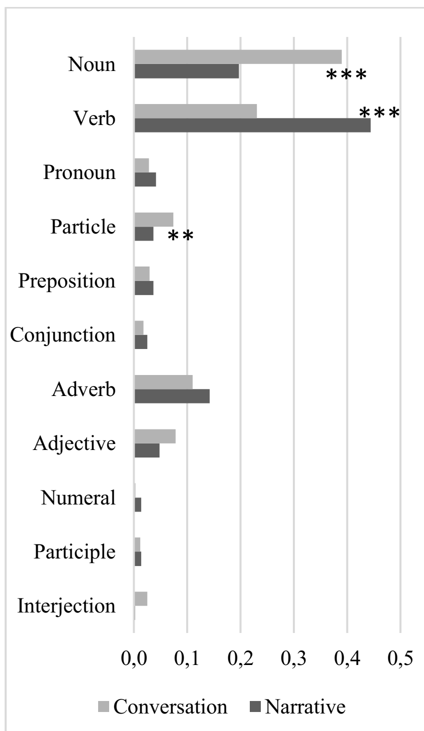


Figure 2. Distribution of *PoSs (lemmas)* in Russian-speaking discourse

Notes: *** means $p \leq 0.001$; ** means $p \leq 0.01$; * means $p \leq 0.05$

The directions of these distinctions were different: in conversations, significantly less verbs, numerals, and conjunctions, but significantly more pronouns, particles, interjections, and adverbs were produced. However, in *lemma distribution,* only three PoSs (verbs, nouns and particles) discriminated the genres (Figure 2).

3.2. PoSP in Different Genres in Lithuanian-speaking Children

Lithuanian-speaking children demonstrated partially similar PoS distribution as their Russian-speaking peers. In conversations, more adverb, pronoun, adjective, numeral, and, especially, particle *word tokens* were produced, while in narratives, nouns, verbs, and conjunctions were significantly more frequent (Figure 3).

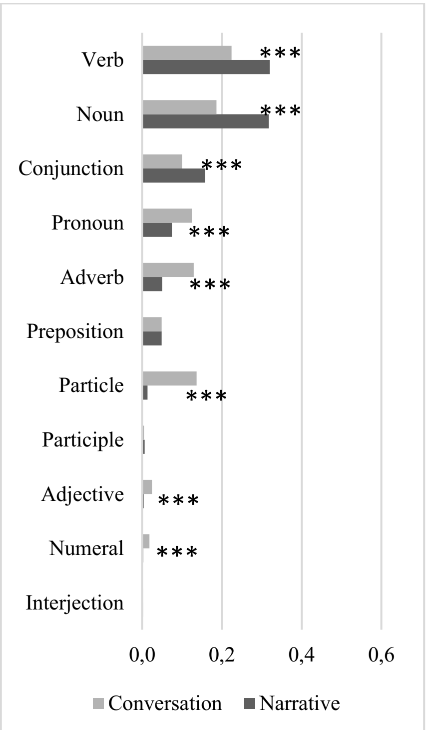


Figure 3. Distribution of *PoSs (word tokens)* in Lithuanian-speaking discourse

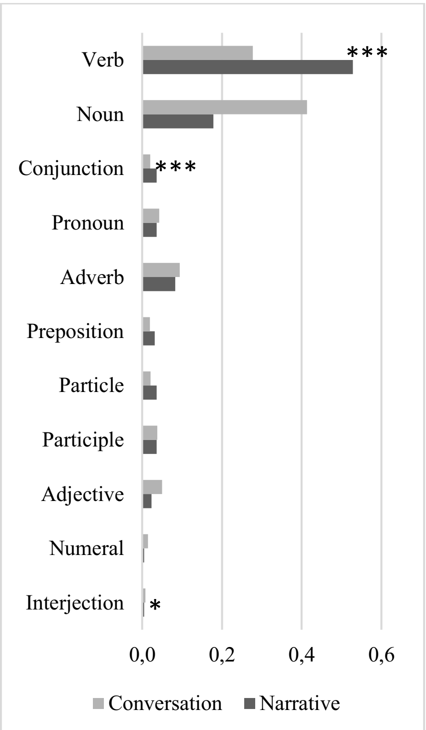


Figure 4. Distribution of *PoSs (lemmas)* in Lithuanian-speaking discourse

In *lemma distribution* (Figure 4), only interjections were more frequent in conversations, while verbs and conjunctions were more frequent in narratives.

3.3. Between-group Analysis of the PoSP

Many between-group distinctions in *word token distribution* were revealed in both genres.

In *narratives*, Lithuanian-speaking children produced more nouns, verbs, and conjunctions, whereas Russian-speaking peers produced more pronouns, prepositions, adjectives, numerals, and, especially, particles (Figure 5).

In *conversations*, the main patterns of PoS distribution were similar to narratives, but nouns were more frequent in the Russian data, while adverbs and numerals were more frequent in the Lithuanian one (Figure 6).

Between-group comparative analysis of the *lemma distribution* revealed only two distinctions in narratives (Figure 7) where adverbs and particles were more frequent in the Russian data; slightly more differences were revealed in conversations (Figure 8) where verbs were more frequent in the Lithuanian data, while particles, adjectives and interjections were more frequent in the Russian one.

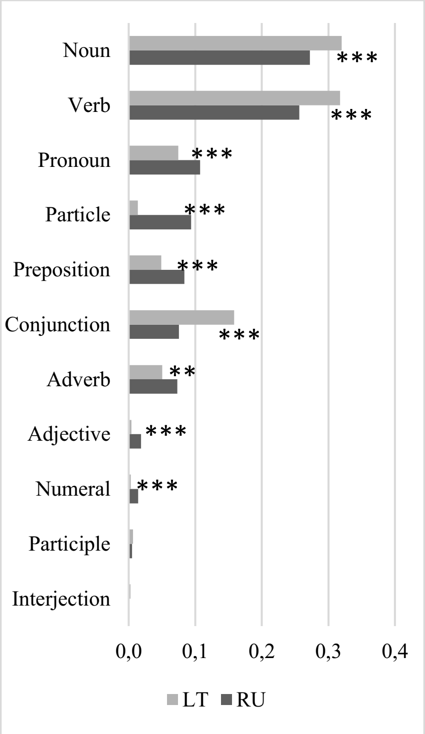


Figure 5. Distribution of *PoSs (word tokens)* in narratives

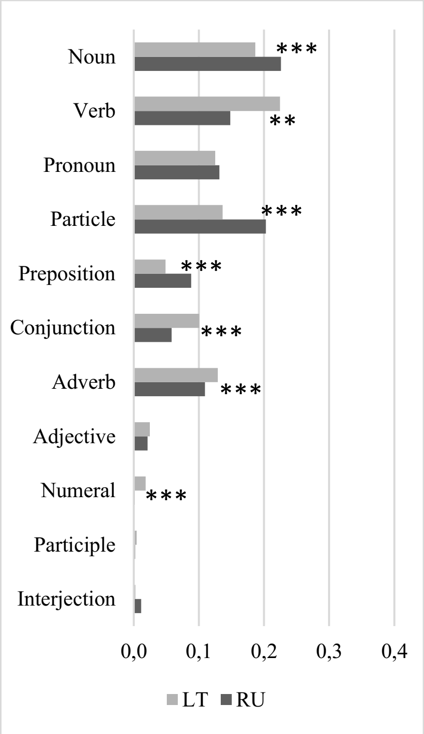


Figure 6. Distribution of *PoSs (word tokens)* in conversations

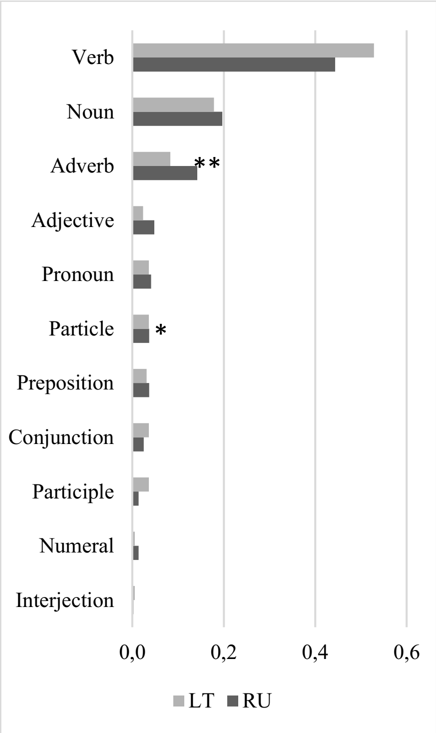


Figure 7. Distribution of *PoSs (lemmas)* in narratives

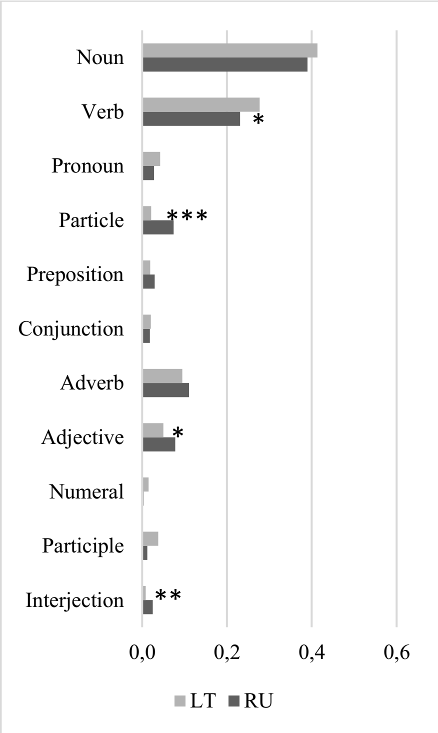


Figure 8. Distribution of *PoSs (lemmas)* in conversations

4. Conclusions and Discussion

Results of the PoS distribution analysis in Russian- and Lithuanian-speaking TD children in different discourses evidenced that PoSP was a rather sensitive measure that discriminated both genres and languages. The genre had a strong influence on the distribution of several PoS (especially, in word token items) in both languages. This probably means that genre demands govern the PoS distribution in language behavior. On the other hand, the PoS distribution in the sample of lemmas revealed much less between-genre distinctions. In other words, within the variety of different lemmas used in the data, only a few PoS discriminated narratives and conversations. It seems reasonable to consider this measure relevant to *language competence* (the variety of acquired lemmas). As for the between-group comparison, new data were obtained about discourse language distinctions. Despite the very similar PoS distribution between contemporary Russian [30] and Lithuanian [31], our study evidenced several between-group distinctions (especially, in the tokens sample). In the narratives, Lithuanian-speaking children produced significantly more nouns, verbs, conjunctions and adverbs, while Russian-speaking children were significantly more productive in pronouns, prepositions, adjectives, numerals, and particles. As for nouns, verbs, and conjunctions, this difference was close to the distinctions between the Russian [32] and Lithuanian national corpora [33]. Hence, despite a rather similar language competence, Russian and

Lithuanian children tended to recruit some PoSs in slightly different ways in different genres.

In addition to the between-group distinctions in language behavior, we found some minor distinctions in language competence. In the narratives, Lithuanian-speaking children used more verbs, while Russian-speaking children used more adverbs. In the conversations, Lithuanian-speaking children used more verbs, conjunctions, and numerals, whereas Russian-speaking peers used more particles and interjections.

In the lemma's PoSP, only two PoSs (verb and particle) discriminated the languages. To sum up, it should be concluded that genres of discourse govern the PoS distribution in both languages and this manifests in *language behavior* measures much stronger than in *language competence* measures.

In multiple publications related to child discourse development, many age-related features have been described. However, a syntactic role of the lexicon in different genre patterns still remains the least analyzed. In some quantitative studies, lexical diversity (e.g. type/token ratio) has been discussed as a language competence measure. However, the lexical (PoSs) richness and diversity (i.e. language competence) and word production (i.e. language behavior) have almost never been disentangled in the same discourse. Our results inspire an assumption that PoS variety in the mental lexicon of the narrator is not the same as the PoSs variety he/she produces in discourse. Also, distinctions between languages and the related pattern of using PoS in child discourse should be considered.

Acknowledgements

The study was supported by a research grant No. 18-18-00114 from the Russian Science Foundation. Our sincere thanks are due to the anonymous reviewers, for their helpful suggestions and comments on previous versions of this paper.

References

[1] Gvozdev AN. Formirovanie u rebenka grammatičeskogo stroja russkogo jazyka. Moskva: Akad. pedag. nauk RSFSR; 1949.
[2] Voeikova MD. Kvalitativnye semantičeskie kompleksy i ih vyraženie v sovremennom russkom literaturnom jazyke i v detskoj reči. SPb.: RGPU; 2004.
[3] Voeikova MD. Rannie ètapy usvoenija det'mi imennoj morfologii russkogo jazyka. Moskva: Znak; 2011.
[4] Dobrova GR. Ontogenez personal'nogo dejksisa (ličnye mestoimenija i terminy rodstva). SPb.: RGPU im. A.I.Gercena; 2005.
[5] Gagarina NV. Stanovlenie grammatičeskih kategorij russkogo glagola v detskoj reči. SPb.: Nauka; 2008.
[6] Ceytlin SN. Očerki po slovoobrazovaniju i formoobrazovaniju v detskoj reči. Moskva: Znak; 2009
[7] Eliseeva MB. Stanovlenie individual'noj jazykovoj sistemy rebenka. Rannie ètapy. Moskva: JaSK; 2014.
[8] Savickienė I. 2003: The Acquisition of Lithuanian Noun Morphology. Wien: Verlag der Österreichischen Akademie der Wissenschaften
[9] Wójcik P. The Acquisition of Lithuanian Verb Morphology: A Case Study. Kraków: Quartis, 2000.
[10] Kamandulytė L.Lietuvių kalbos būdvardžio įsisavinimas: leksinės ir morfosintaksinės ypatybės. Kaunas: VDU; 2009.
[11] Voeikova MD. The acquisition of case in typologically different languages. In: MD Voeikova, WU Dressler (eds) Pre- and protomorphology. Early phases of morphological development in nouns and verbs. Vienna: University of Vienna, 2007; p.
[12] Dabašinskienė I, Voeikova M. Diminutives in Spoken Lithuanian and Russian: Pragmatic functions and structural properties. In: P Arkadiev, A Holvoet, B Wiemer (eds) Contemporary approaches to Baltic linguistics. Moscow: RAS, 2015; p.203-234.

[13] Le Normand MTh, Moreno-Torres I, Parisse C, Dellatolas G. How do children acquire early grammar and build multiword utterances? A corpus study of French children aged 2 to 4. Child Development 2013;84(2):647-661.
[14] Lieven EVM, Pine JM, Baldwin G. Lexically-based learning and early grammatical development. Journal of Child Language 1997;24(1):187-219.
[15] Tomasello M. A usage-based approach to child language acquisition. Annual Meeting of the Berkeley Linguistics Society 2000;6(1):305-319.
[16] Tomasello M. Constructing a language: A usage-based approach to child language acquisition. Cambridge, MA: Harvard University Press, 2003.
[17] Labelle M. The acquisition of grammatical categories: The state of the art. In: Cohen H, Lefebvre C, editors, Handbook of Categorization in Cognitive Science. Amsterdam: Elsevier; 2005. p. 433-458.
[18] Slobin DI. Learning to think for speaking: Native language, cognition, and rhetorical style. Pragmatics 1991;1:7-26.
[19] Berman R, Slobin DI. Relating Events in Narrative: A Crosslinguistic Developmental Study. Hillsdale, NJ: Erlbaum, 1994.
[20] Rayson P. Matrix: A statistical method and software tool for linguistic analysis through corpus comparison. Lancaster: Lancaster University; 2002.
[21] Rayson P, Garside R. Comparing corpora using frequency profiling. Paper presented at the Workshop on Comparing Corpora, held in conjunction with the 38th annual meeting of the Association for Computational Linguistics (ACL 2000); 2000.
[22] Rayson P, Berridge D, Francis B. Extending the Cochran rule for the comparison of word frequencies between corpora. Paper presented at the 7es Journées internationales d'Analyse statistique des Données Textuelles; 2004.
[23] Liashevskaya ON. Korpusnye instrumenty v grammatičeskih issledovanijah russkogo jazyka. Moskva: IRYA im. V.V.Vinogradova RAN; 2014.
[24] Balčiūnienė I, Kamandulytė-Merfeldienė L. The Corpus of Lithuanian Children Language: Development and application for modern studies in language acquisition. Kalbotyra. 2018;(71):7-25.
[25] Balčiūnienė I, Kornev AN. Osobennosti ustnogo diskursa u detej 4-5 let: aprobacija novogo metoda polučenija korpusnych dannych. In: 8 meždizciplinarnyj seminar «Analiz razgovornoj russkoj reči 2019»; 2019; SpB: SpBSU; p. 31-38.
[26] Hickmann M. Children's Discourse: Person, Time and Space across Languages. Cambridge: Cambridge University Press, 2003.
[27] Balčiūnienė I, Kornev AN. Linguistic disfluency in children discourse: Language limitations or executive strategy? In: Computational Linguistic and Intellectual Technologies: Proceedings of the International Conference «Dialogue 2016»; 2016; Online resource. http://www.dialog-21.ru/media/3381/bal%C4%8Di%C5%ABnien%C4%97ikornevan.pdf.
[28] Homola P, Rimkutė E, Jarašiūnaitė G. Morfologinių samplaikų atpažinimas ir klasifikavimas. Lituanistica. 2005;(2): 58-75.
[29] MacWhinney B. The CHILDES Project: Tool for Analyzing Talk. Mahwah, NJ: Lawrence Erlbaum Associates, 2000.
[30] Sičinava DV. Časti reči [Parts-of-speech]. Russkaja korpusnaja grammatika. 2011; Online resource. http://rusgram.ru/
[31] Utka A. Dažninis rašytinis lietuvių kalbos žodynas 1 milijono žodžių morfologiškai anotuoto tekstyno pagrindu. Kaunas: VDU; 2009.
[32] Bogdanova-Beglarian NV, Šerstinova TYu, Baeva EM, Blinova OV, Martynenko GYa, Ermolova OB, Ryko AI. et al. Russkij jazyk povsednevnogo obŝenija: osobennosti funkcionirovanija v raznyh social′nyh gruppah. SPb.: LAIKA; 2016.
[33] Dabašinskienė I. Šnekamosios lietuvių kalbos morfologinės ypatybės. Acta Linguistica Lithuanica 2009; LX:1-15.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200628

Lithuanian Pedagogic Corpus: Correlations Between Linguistic Features and Text Complexity

Loïc BOIZOU, Jolanta KOVALEVSKAITĖ and Erika RIMKUTĖ[1]
Vytautas Magnus University, Lithuania

Abstract. This paper discusses the problem of automatic CEFR[2] level assignment to texts. We address the correlations between the lexical, morphological and syntactic features and the different CEFR levels of the texts in the Lithuanian Pedagogic Corpus. Only the texts from coursebooks showed the correlation of investigated linguistic features with text complexity. In the coursebook sub-part of the corpus, we observed that higher language proficiency levels are associated with more complex linguistic features: their number increases in texts of higher CEFR levels from A1 to B2 (e.g., non-finite verb forms, participles, adverbial participles and half participles, dative and instrumental noun cases or longer sentences).

Keywords. Lithuanian language, Lithuanian Pedagogic Corpus, automatic text classification, text complexity, linguistic features, Common European Framework of Reference for Languages (CEFR)

1. Introduction

This paper discusses the problem of automatic CEFR level assignment to texts. Specifically, we address the linguistic features of the Lithuanian Pedagogic Corpus[3] and their correlation with text complexity. The Lithuanian Pedagogic Corpus is a small monolingual specialized corpus which provides material relevant to learning and teaching Lithuanian as a foreign language. The corpus consists of 669,000 tokens and includes 111,000 tokens of A1-A2 level texts (96,000 tokens of written and 15,000 tokens of spoken samples) and 558,000 tokens of B1-B2 level texts (523,000 tokens of written and 35,000 tokens of spoken samples) [1]. For this study, only the sub-corpus of written texts (618,637 tokens) has been used (in the corpus, A1 level texts make up 6.93 %, A2 – 8.52 %, B1 – 10.99 %, and B2 – 73.56 %).

The data for this sub-corpus was collected from 1) coursebooks of the Lithuanian language (17.2 %) and 2) a variety of authentic Lithuanian material (82.8 %): news portals, popular science books, advertisements, stories, fairy tales, letters, songs, public information (travelling, health care, and other), etc. In total, the corpus includes 29 genres.

[1] Corresponding Author: Erika Rimkutė; Vytautas Magnus University, V. Putvinskio st. 23-216, Kaunas LT-44243, Lithuania; E-mail: erika.rimkute@vdu.lt.

[2] CEFR – Common European Framework of Reference for Languages: https://www.coe.int/en/web/common-european-framework-reference-languages/level-descriptions.

[3] The project *Lithuanian Academic Scheme for International Cooperation in Baltic Studies*: http://baltnexus.lt/en/baltic-studies-project.

In previous research [2], texts taken from coursebooks were used to automatically predict the CEFR level of texts from other sources. The procedure involved the classification of texts into two (A1-A2 and B1-B2) or four (A1, A2, B1, B1) CEFR levels using machine learning (ML) methods (as described by [2]). Different experiments with various ML methods were carried out with a combination of surface quantitative features (number of sentences, average sentence and word length, ratio between longer and shorter items, etc.) and linguistic features (word length and type/token ratio for selected parts of speech; proportions of selected morphological features). The best results were obtained using the logistic regression. According to [2], the difference between the most beneficial and the least beneficial features was very small. In addition, the training allowed to reclassify texts that were previously defined in broader categories (A1-B1, B1-B2) into four categories. The relatively low efficiency of automatic classification (cross validation score of about 0.6 with four CEFR levels) did not allow to be confident about the results. Additional examination of the data reinforced the reservations about the validity of classification for non-didactic material.

In this study, we aim to reassess the results of the automatic text classification experiment reported by [2] and to get a better understanding of the representativeness of the Lithuanian Pedagogic Corpus and its sub-parts. Furthermore, we want to re-evaluate the previously analysed linguistic features important for text complexity assessment and to identify additional useful features.

After the first step of the research, the distribution of morphological, syntactic and lexical features has shown considerable discrepancies between the coursebooks and the texts from other sources that confirmed the weakness of the classification for non-didactic material (non-coursebook texts did not show the correlation between moving from lower to higher levels and the growing complexity of lexis and grammar). For this reason, the scope of the analysis was restricted to texts from coursebooks. In this paper, we first provide information about the distribution of linguistic features presented in the materials for the teachers of Lithuanian as L2 (i.e., which grammatical features and how they are described in the materials for relevant language levels) (Section 2); and compare the number and manner of the distribution of linguistic features in the Lithuanian Pedagogic Corpus (Section 3).

2. Grammatical Features in the Teaching Material of Lithuanian as L2 Prepared According to CEFR: Learning, Teaching, and Assessment

The discussion of the connections between grammatical forms and language levels is based on the CEFR materials designed for levels A1, A2, B1, and B2, see respectively [3], [4], [5], [6].

Noun gender is introduced in level A1, while more complex cases of expressing notions are discussed in level A2. Number[4] is also explained in level A1; cases are introduced in level A1 (only the most frequent meanings are taught, e.g., locative to indicate location and time, nominative to indicate a thing, a person, a phenomenon or a state, vocative to address a person or an animal) as well as in level A2, while shortened forms of cases are introduced in level B2.

[4] The number of other words that agree with nouns is not specified, because CEFR emphasizes agreement; other agreement categories – gender and case – are discussed separately, as their acquisition differs and is more difficult.

Adjective gender, cases, semantic classification into qualitative and relational (this distinction determines some grammatical features), and degree are discussed in level A1; pronominal forms are introduced in level A2.

Pronoun cases are introduced in level A1 (only forms important for learner communication at level A1 are taught, e.g., *manęs* ('me' Gen.), *tavęs* ('you' sg. Gen.), *jus* ('you' pl. Acc.), *mane* ('me' sg. Acc.), only in nominative, genitive, and accusative); the topic is continued in level A2; pronominal forms are instructed in level B2.

Numeral cases are introduced in level A1 (only forms relevant to learner communication are taught, e.g., *Reikia dviejų stiklinių miltų* ('I need two glasses of flour'); *Turiu penkis eurus* ('I have five euros'); *Yra keturios kėdės* ('There are four chairs'): only nominative, genitive, and accusative) as well as in level A2. Cardinal numerals are introduced in level A1; multiple and ordinal numerals are taught in A2 level, while in level B2, learners get acquainted with collective cardinal numerals and fractions. The structure of numerals (simple, combination, and compound) is described in level B2. The numeral governance over other words (e.g., in *dešimt vyrų* ('ten men'), with the noun in genitive because of the numeral) is explained in level A1.

Adverb degree is introduced in the material for level A1.

Verb tenses are explained in different levels: present, past simple and future tense of finite forms in A1; past frequentative of finite forms and compound tense forms in A2. The notion of mood is introduced in level A1 (only the politeness aspect of the subjunctive is explained) and level A2. Reflexive forms are discussed in levels A1 (only the forms that are part of phrases to be learned by heart[5]), A2 and B2 (reflexive participles). Participles are introduced in levels A1 (as multi-word lexical units[6]) and A2; pronominal forms of participles are taught only in level B2; participle voices are described in levels A2 (with more emphasis only on passive forms due to easier declining; while future participles are not included), B1 (active participle forms of all tenses are also presented) and B2 (passive future participles are introduced). Syntactic features of participles (predicative, half predicative and attributive usage) are discussed in the material for level A2. Verb transitivity is introduced in levels A1 (only the fact that verbs usually govern genitive and accusative is mentioned) and A2; aspect is discussed in level A2. Non-finite forms are introduced from levels B1 (half participles, present and past adverbial participle) and B2 (adverbial participle of past frequentative tense, and necessity participles).

The formation of various parts of speech, i.e., derivatives is introduced in coursebooks for level A2. Thus, starting from level A2 learners are taught morphologically more complex and longer words. **Sentence types, word order and sentence parts** are also discussed in the material for level A2.

3. Correlations between Lexical, Morphological and Syntactic Features and the CEFR Levels

We start the analysis by showing the correlation between the morphological features and the different CEFR levels and continue with the discussion of the results of syntactic and

[5] E.g., *Kaip sekasi?* ('How are you?') *Mokausi Vilniaus universitete.* ('I am studying at Vilnius University.') *Man patinka maudytis.* ('I like swimming.').

[6] E.g., *rūkyta žuvis* ('smoked fish'), *rauginti agurkai* ('pickled cucumbers'), *pavargęs* ('tired' (masc.)), *ištekėjusi* ('married' (fem.)).

lexical features. The assessed syntactic features include sentence length and lexical features as type/token ratio, word length (this correlates with the fact that higher language proficiency levels contain more complex derivatives) and the lexical coverage of the most frequently used vocabulary.

3.1. Morphological Features

Morphological features are especially important as Lithuanian is a highly inflected language: learners are taught many grammatical categories and inflected forms of the noun, verb and other parts of speech. All texts in the corpus were morphologically annotated automatically. For this reason, some inaccuracies can be found; however, they are not numerous and their quantity does not invalidate the general tendencies.

In this sub-section we discuss the distribution of some forms of verbs and nominal words.

3.1.1. Verb Forms

Table 1. Finite and non-finite verb forms

Text level/verb forms	Finite forms	Infinitives	Participles	Adverbial participles	Half participles
A1	**80.94**	15.98	2.83	0.09	0.16
A2	79.45	14.75	5.42	0.13	0.25
B1	68.34	**16.91**	12.86	0.97	0.90
B2	64.02	16.18	**16.51**	**1.63**	**1.63**

As we can see in Table 1, the distribution of finite and non-finite verb forms correlates with a language level: it is obvious that **finite forms** prevail in lower levels (in level A1 texts – 80.94 %), while the number of these forms decreases with the rising language level where they are replaced by grammatically more complex forms. We have also noticed a significant difference between the least used finite verb form cases (64.02 % in B2) and the mentioned largest number of usage instances (80.94 %).

Infinitives are used quite consistently: from 14.75 to 16.91 % (the highest frequency is for level B1 texts). However, the infinitive in the Lithuanian language is one of the fundamental forms acquired in order to be able to use a verb. Moreover, the infinitive is used in diverse areas (not only as a predicate, but also as an object, a subject, an attribute or an adverbial). For these reasons, the usage of infinitives cannot accurately reflect the complexity or simplicity of the language.

Participles, on the other hand, can be considered as an important indicator of the higher language proficiency; as we see in Table 1, their number varies considerably: from 2.83 % (A1) to 16.51 % (B2). Although **adverbial participles** and **half participles** are not abundant, they are mostly used in level B2 texts which shows the growing complexity of the grammar.

Table 2. Verb moods

Text level/mood	Indicative	Imperative	Subjunctive
A1	90.34	6.16	3.50
A2	85.16	**9.63**	**5.21**
B1	91.28	4.08	4.64
B2	**93.08**	2.82	4.10

The pedagogic corpus reflects a high usage of the feature typical to both spoken and written Lithuanian – **indicative** forms with variation ranging from 85.16 % to 93.08 %

across language levels. The frequencies of **imperative** reveal that it is the most frequent in level A2 texts (9.63 %). This can be explained by the fact that these forms are common to dialogues, which make up a major part of lower level texts. The distribution of **subjunctive** forms does not show a significant variation across language levels (from 3.5 to 5.21 %).

We can draw a conclusion that the data of verb moods confirms the general tendencies of the Lithuanian grammar features and does not provide reliable information about the correlation between complex forms and higher language levels.

Table 3. Tenses of finite forms

Text level/tense	Present	Simple past	Past frequentative	Future
A1	**78.34**	13.91	0.16	7.60
A2	54.22	**26.61**	5.51	**13.66**
B1	58.15	25.66	**5.72**	10.46
B2	43.31	47.10	4.44	5.14

The usage of present and simple past tenses cannot indicate the correlation between the complexity of grammatical forms and the language level because these forms are very common and their distribution in other corpora (e.g. MATAS[7]) is very diverse. On the other hand, the forms of past frequentative suggest the following correlation: because of their complexity, they are less frequent in lower level texts and more frequent in higher levels (with the highest usage in level B1). The fact that the most future forms (13.66%) appear in A2 texts is not enough to show the correlation of these forms with a language level – a larger corpus to highlight this correlation should be used.

Table 4. Voice and tense of participles

Text level/ voice and tense	Active present	Active simple past	Active past frequ-entative	Active future	Passive present	Passive past	Passive future	Necessi-ty
A1	1.61	33.06	0.00	0.00	**49.19**	15.32	0.00	0.81
A2	3.53	17.06	0.00	0.59	47.06	**30.59**	0.00	1.18
B1	**15.99**	21.88	0.55	0.18	33.09	27.57	0.18	0.55
B2	10.81	**29.58**	0.00	0.00	29.77	28.99	0.39	0.46

Due to the low numbers of instances, we cannot draw conclusions about the usage of past frequentative and future tenses of active participles, passive future and necessity participles, as only several cases or none of these occurred in the texts of every level.

According to CEFR, the usage of participles should increase from level A2. In this level, the focus is put on passive participles, because they are simpler than active ones. This fact is supported by the data in Table 4: A1-A2 level texts contain more passive than active participles. Admittedly, high frequency of passive present participles in A1 texts is surprising – even 49.19 %. This could be explained by the necessity for learners, even at the beginning, to acquire certain multi-word lexical units containing passive present participles, e.g., *rašomasis stalas* (‘a desk’), *valgomasis šaukštas* (‘a tablespoon’).

The largest number of active present participles in level B1 (15.99 %) and active simple past participles in A2 texts (29.58 %) allows the presumption of a correlation between grammatically more complex forms and a higher language level.

[7] Lithuanian morphologically annotated corpus MATAS: https://clarin.vdu.lt/xmlui/handle/20.500.11821/33.

3.1.2. Nominal Forms

We paid a particular attention to such grammatical features of nominal words as noun cases, numeral types, pronominal forms, and adjective and adverb comparative forms, because CEFR quite clearly prescribes when and which numerals should be used or when pronominal forms are taught. Although according to CEFR, all noun cases are introduced in level A1, we can presume that texts of lower levels will contain fewer instances of rarer cases (especially dative or instrumental).

Table 5 provides only **noun cases**. The distribution of other parts of speech was not analysed as most adjectives, pronouns, numerals and participles agree with nouns, thus the choice of their cases (as well as gender and number) depends on the form of a noun.

Table 5. Noun cases

Text level/ case	Nom.	Gen.	Dat.	Acc.	Ins.	Loc.	Voc.	Ill.
A1	**38.75**	27.59	1.46	**19.02**	3.53	**8.35**	1.24	0.04
A2	31.56	33.21	2.79	18.65	5.98	6.47	**1.35**	0.00
B1	30.05	36.03	2.77	16.88	**6.43**	7.38	0.40	0.06
B2	27.24	**39.32**	**3.29**	17.08	6.00	6.81	0.22	0.03

We assume that the three most frequent cases (nominative, genitive, and accusative) will not reveal the correlation between the grammatical complexity and language level. The correlation is not indicated by very rare cases – vocative and illative (a type of locative, not included into the grammar system of Modern Lithuanian). Even though locative is not a frequent case, it is inevitable even at the beginning of language acquisition, because one has to learn to indicate a place or time. For this reason, it is not surprising that most locative instances occur in texts for level A1 (8.35 %).

The link between the complexity of grammar and language level can be demonstrated by two rarely used cases: dative and instrumental. These cases are usually used to express the facultative valency; they can be often replaced by prepositional constructions with frequently used cases of genitive and accusative. Based on the data, we can state that our earlier hypothesis was confirmed and both cases show the relation between the growing grammatical complexity and language level: most dative forms occur in level B2 texts (3.29 %), while instrumental in level B1 texts (6.43 %).

Table 6. Types of numerals

Text level/type of numerals	Cardinal	Multiple	Collective	Ordinal
A1	26.75	0.56	0.00	**72.69**
A2	91.05	1.43	0.00	7.52
B1	**92.97**	0.67	0.00	6.35
B2	85.73	**1.72**	**0.08**	12.48

As to the **numeral** usage, we can maintain that even though multiple and collective numerals are not common, their higher frequency in level B2 suggests the correlation between more difficult grammatical forms and a higher language level. It was surprising though to see that most ordinal numerals are used in level A1 texts (72.69 %). Especially common are the same first ordinal numerals: *pirmas* (‘the first’), *antras* (‘the second’), and *trečias* (‘the third’). Presumably, they were learned as individual lexical items. We cannot draw any conclusions about cardinal numerals, because we analyse only numerals written in words and exclude numerals written in a numerical form.

Table 7. Degrees of adjectives and adverbs

Text level/ degree	Adj. positive	Adj. comparat.	Adj. superlative	Adv. positive	Adv. comparat.	Adv. superlative
A1	**96.55**	1.44	2.00	**94.73**	4.02	1.25
A2	89.71	2.90	7.39	92.58	4.85	2.56
B1	87.95	3.35	**8.70**	87.56	**7.87**	**4.57**
B2	89.52	**4.21**	6.27	90.04	6.89	3.06

We can see that **degree** might be important in determining the relation between the grammatical complexity and the language level, because the positive degree is more frequent in lower level texts, while more complex forms – the comparative and superlative degrees – in higher level texts.

3.2. Syntactic and Lexical Surface Features

Table 8 shows the number of **sentences** and their length in the analysed part of the pedagogic corpus. The average sentence length is 12.15 words. The sentence length substantially correlates with the complexity of texts: A1 level texts contain the shortest sentences – 8.08 words, while the longest sentences are found in level B2 – 15.94 words.

Table 8. Length of sentences

Text level/syntactic features	Number of sentences	Average sentence length (in words)
A1	5,591	8.08
A2	2,864	10.12
B1	2,575	14.44
B2	4,599	**15.94**

Although the **word length** (in terms of the number of letters) is not very diverse, it is evident that words in higher levels are longer, thus, morphologically or derivationally more complex (see Table 9).

Table 9. Lexical surface features

Text level/lexical features	Average word length (in characters)	3,075 most frequent word forms (coverage)	Type/token ratio
A1	5.39	**69.13%**	0.26
A2	5.59	61.05%	0.39
B1	5.95	55.86%	**0.43**
B2	**6.16**	54.23%	0.37

In this study, the most **frequent vocabulary**, i.e., the 3,075 most frequent word forms of the corpus, was integrated into the assessment of text complexity. The results confirm that vocabulary is larger in higher level texts: 69.13 % of words are from the most frequent vocabulary list for A1 level texts; in B2 level texts, the most frequent vocabulary comprises 54.23 % of all words. For future work, it will be important to have a better-defined reference word list, since most experiences on automatic level assignment stress the primary importance of the lexicon for this task.

Type/token ratio also indicates the correlation between higher level texts and higher lexical diversity. However, as we can see in Table 9, the highest diversity, as one might expect, is not found in level B2 texts but, rather, in level B1 texts (0.43). This might be explained by the repetition of similar topics in level B2, e.g., Lithuania, customs, holidays, the same famous people; thus, the lexical diversity becomes lower. Furthermore, several coursebooks included in the corpus were of transitional level, e.g., B1-B2. Such texts were automatically classified into B1 or B2 according to the experiment described in [2]. This might have influenced the fact that the highest lexical diversity was not in level B2 texts, although these texts are characterized by the most complex grammar.

4. Conclusions

In this work, as well as in the experiments described by [2], we focused mostly on the morphological features. During the study [2], proportions of various morphological features were calculated, and the obtained data was used to assign the language level to text. Also, lexical features, specifically, which part of all vocabulary is covered by the most frequent words, were considered.

The linguistic features described revealed that the automatic text classification applied earlier by [2] was not sufficiently precise; therefore, non-coursebook texts in the corpus should be reclassified. As [2] suggested and as [7] demonstrated, a wider set of lexical information could strongly improve the quality of a renewed prediction on non-didactic materials.

We can state that in order to determine the text level automatically, it is worth considering the correlation described in this article – the link between the language level and properties indicating more complex forms (participles, adverbial participles and half participles) in comparison with all verb forms; the usage of finite forms of past frequentative tense; the usage of present and past simple tense participles of the active voice; the usage of multiple and collective numerals; the usage of dative and instrumental for nouns in comparison with other cases; the usage of comparative and superlative degree. It is also important to consider the length of a sentence, word length, type/token ratio and the distribution of the most frequent words of the analysed corpus. Nevertheless, in order to determine clear values of each aforementioned linguistic properties in automatic text level assignment, more texts and additional experiments are needed.

References

[1] Kovalevskaitė J, Rimkutė E Mokomasis lietuvių kalbos tekstynas: naujas išteklius lietuvių kalbos besimokantiesiems (Pedagogic Corpus of Lithuanian: a New Resource for Learning and Teaching Lithuanian as a Foreign Language). Sustainable multilingualism (forthcoming). 2020.

[2] Grigonytė G, Kovalevskaitė J, Rimkutė E. Linguistically-Motivated Automatic Classification of Lithuanian Texts for Didactic Purposes. In: Muischnek K, Müürisep K, editors. Proc. of the 8th International Conference Baltic HLT 2018; 2018 Sep 27-29; Tartu (Estonia). Frontiers in Artificial Intelligence and Applications, vol. 307. Amsterdam: IOS Press. P. 38-46.

[3] Stumbrienė V. Lūžis (Breakthrough – A1). Vilnius: Vilniaus universiteto leidykla; 2016. 67 p.

[4] Ramonienė M, Pribušauskaitė J, Vilkienė L. Pusiaukelė (Waystage – A2). Europos Taryba; 2006. 148 p.

[5] Ramonienė M, Pribušauskaitė J, Vilkienė L. Slenkstis (Threshold – B1). Vilnius: Vilniaus universiteto leidykla; 2016. 244 p.

[6] Ramonienė M, Pribušauskaitė J, Vilkienė L. Aukštuma (Vantage – B2). Vilnius: Vilniaus universiteto leidykla; 2016. 247 p.

[7] Pilán I, Vajjala S, Volodina E. A Readable Read: Automatic Assessment of Language Learning Materials based on Linguistic Complexity. International Journal of Computational Linguistics and Applications. 2016 7(1):143–59.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200629

Detailed Error Annotation for Morphologically Rich Languages: Latvian Use Case

Roberts DARĢIS [1], Ilze AUZIŅA, Kristīne LEVĀNE-PETROVA and Inga KAIJA
Institute of Mathematics and Computer Science, University of Latvia
Rīga Stradiņš University

Abstract. This paper presents a detailed error annotation for morphologically rich languages. The described approach is used to create Latvian Language Learner corpus (LaVA) which is part of a currently ongoing project *Development of Learner corpus of Latvian: methods, tools and applications*. There is no need for an advanced multi-token error annotation schema, because error annotated texts are written by beginner level (A1 and A2) who use simple syntactic structures. This schema focuses on in-depth categorization of spelling and word formation errors. The annotation schema will work best for languages with relatively free word order and rich morphology.

Keywords. Leaner corpus, error annotation, language acquisition, corpus development

1. Introduction

Learner corpora constitute a new resource for second language acquisition and foreign language teaching specialists. They are particularly useful if they are error-tagged with consistently annotated errors. Annotation schema is one of the most important aspects of a learner's corpus. A detailed error annotation schema provides a wide range of statistical analysis, enabling researchers to conduct numerous kinds of quantitative research, and allows the development of fine-grained search that enables research to quickly find the information of interest for qualitative analysis with no need to go through a lot of redundant information.

This paper presents the error annotation schema used in the development of Learner Corpus of Latvian (LaVA). The LaVA corpus is developed as a part of an ongoing project *Development of Learner corpus of Latvian: methods, tools and applications*, started in September 2018. Latvian is a language with rich morphology and a relatively free word order. Latvian can be generally considered a phonetic language, i.e. a language with a relatively simple relationship between orthography and phonology. From the language acquisition perspective, Latvian has several specific properties: short and long vowels

[1]Corresponding Author: Roberts Darģis, Artificial Intelligence Laboratory, Institute of Mathematics and Computer Science, University of Latvia, Raiņa bulv. 29, Riga, LV-1459, Latvia; E-mail: roberts.dargis@lumii.lv.

and diphthongs, a high degree of inflection and a rather free word order. These properties have to be taken into account in the error-annotation process.

2. Related Work

Learner corpora have been collected and analyzed for more than 25 years now and their popularity is increasing. There are many learner corpora for English, such as the International Corpus of Learner English [1] among others. However, more and more learner corpora are being developed for other languages as well, many of which are morphologically rich [2], [3].

Usually, errors are grouped according to language level (phonetics, morphology, syntax, etc.); the linguistic category to which the error belongs and the changes that occur when comparing the original and corrected texts (omission, addition, misformation, etc.) [4], [5], [6]. Although error annotation schemas for morphologically rich languages have more detailed error categories and subcategories [7], [8], [9], manually defined categories will never be comprehensive. In Latvian, there are more than 2,000 morphological tags of which about 200 are used to describe nouns and more than 1,000 to describe verbs, which leads to many possible error combinations. A lot of information would be lost using even as many as 100 error codes. In the LaVA corpus, a different approach is used after text correction. Instead of using a limited set of error codes, only morphological information is annotated and more fine-grained error codes are automatically extrapolated from the morphological information.

3. Error Annotation Schema

The error annotation is done on the alignment between the original text and the corrected text [10]. A commonly used error taxonomy for Latvian includes 5 types (Spelling errors, Punctuation errors, Grammatical errors, Syntactical errors and Lexical errors) and multiple subtypes (for example, subtypes for Spelling errors: Upper/lower case letter, Diacritics, Separately/together spelled words, Missing letter, Redundant letter, Other spelling errors) [6], [10]. These error codes are not directly used in the LaVA corpus; instead, more detailed error codes are extrapolated for five other properties, which are much more easier to annotate. These properties are: original token without typos (the token written by the learner with corrected spelling errors), original lemma, original tag, corrected lemma, and corrected tag (figure 1).

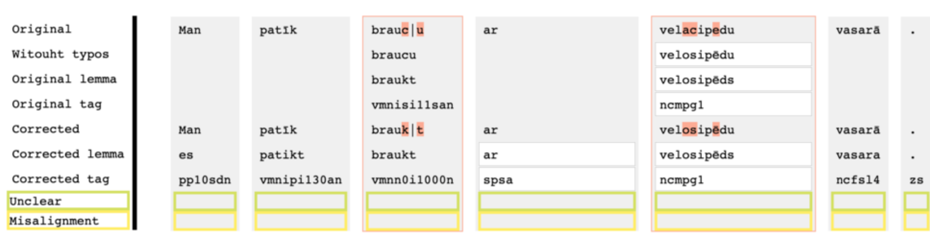


Figure 1. Error annotation interface

The spelling errors can be determined automatically by comparing the original token with the original token without typos. Character level alignment combined with a rule based system allows to extract exactly which character pairs are used incorrectly. This information can later be used to facilitate qualitative research by providing fine-grained search or quantitatively grouping the extracted character errors. To specify misspellings of together or separately written words, adjacent units are marked/pulled together.

A morphological tag contains a lot of information, including part of speech (Pos) tag. There is a tag for punctuation marks, so recognizing punctuation errors is straightforward – if the corrected token is different from the original token and the tokens are punctuation marks, it is a punctuation error.

Lexical errors mean that the lemma of the corrected token is different from the lemma of the original token. Subtype cannot be determined automatically, but it can be added later for unique token pairs only, because the subtype is not context dependent.

The remaining errors are grammatical errors. A very detailed grammatical error analysis can be done based on the morphological tags.

In addition, two more properties can be annotated – *unclear* and *misalignment*. Both of these properties are there just as percussion. *Misalignment* is meant for cases where alignment is not correct, for example, in the alignment, it shows that one token is replaced with another, but actually one is removed and the other one is added independently. *Unclear* is used for cases in which it is not clear what annotations should be added or there is a wider context that impacts the error and that cannot be annotated in the current scheme, for example, a prepositional construction should be used instead of the word form, or an analytic form of verb is used. Such cases are summarised and discussed to decide the correct annotations and update the error annotation scheme if necessary, and to annotate errors at the syntax level.

4. Semi-automatic Annotation Generation

All the values of the properties mentioned in the previous section are generated automatically. There are two types of values: those that can be edited and those that are read-only (figure 1).

The read-only values are obtained from a manually annotated and verified list of tokens which occurred at least 3 times and have only one possible lemma and tag regardless of context.

The suggestions for the rest of the property values are acquired from a morphological annotator [11]. Tag and lemma for punctuation marks and numerals are considered to be correct and are also read-only.

The morphological suggestions for the original tokens are highly inaccurate due to the high amount of typos. If the original form is not in the dictionary and the words are similar, it is considered to be the same token with typos and annotations from the corrected token are suggested.

5. Conclusion

The error annotation method proposed in this paper is tested in the LaVA corpus development. The corpus consists of error annotated texts written by beginner level (A1 and

A2) language learners. There is no need for an advanced multi-token error annotation schema, because beginners use simple syntactic structures. Most of the errors are limited to individual tokens. This schema has more detailed categorization of spelling and word formation errors. These errors are more common and much more diverse for beginner level compared to intermediate and advanced level. Further work includes review of unclear segments and extending annotation schema to support syntax errors and more complex multi-word structure annotation if necessary.

Acknowledgment

The work reported in this paper is a part of the project *Development of Learner Corpus of Latvian: methods, tools and applications* (Project No. lzp-2018/1-0527) that is being implemented at the Institute of Mathematics and Computer Science, University of Latvia (IMCS UL) since September 2018. The project is financed by Latvian Council of Science.

This work also a part of the Latvian State Research Programme "Latvian Language" (No. VPP-IZM-2018/2-0002) subproject "Acquisition of Latvian Language" that is being implemented at IMCS UL.

References

[1] Granger S, Dagneaux E, Meunier F, Paquot M, et al.. International corpus of learner English. UCL, Presses Univ. de Louvain; 2009.

[2] Siemen P, Lüdeling A, Müller FH. FALKO-ein fehlerannotiertes Lernerkorpus des Deutschen. In: Proceedings of Konvens. vol. 2006; 2006. p. 107.

[3] Rakhilina EV, Vyrenkova A, Mustakimova E, Ladygina A, Smirnov I. Building a learner corpus for Russian. In: Proceedings of the joint workshop on NLP for Computer Assisted Language Learning and NLP for Language Acquisition; 2016. p. 66–75.

[4] Tono Y. Learner corpora: design, development and applications. In: Proceedings of the Corpus Linguistics 2003 conference. University Centre for Computer Corpus Research on Language Lancaster; 2003. p. 800–809.

[5] James C. Errors in language learning and use: Exploring error analysis. Routledge; 2013.

[6] Znotiņa I. Otrās baltu valodas apguvēju korpuss: izveides metodoloģija un lietojuma iespējas. Liepājas Universitāte; 2018.

[7] Štindlová B, Škodová S, Rosen A, Hana J. A learner corpus of Czech: Current state and future directions. Twenty years of learner corpus research: Looking back, moving ahead. 2013:435–446.

[8] Ledbetter S, Dickinson M. Automatic morphological analysis of learner Hungarian. In: Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications; 2015. p. 31–41.

[9] Gayo IDR, Antunes S, Mendes A, Janssen M. Towards error annotation in a learner corpus of Portuguese. In: Proceedings of the joint workshop on NLP for Computer Assisted Language Learning and NLP for Language Acquisition; 2016. p. 8–17.

[10] Darģis R, Auziņa I, Levāne-Petrova K. The use of text alignment in semi-automatic error analysis: use case in the development of the corpus of the Latvian language learners. In: Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018); 2018. p. 4111–4115.

[11] Paikens P, Rituma L, Pretkalniņa L. Morphological analysis with limited resources: Latvian example. In: Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013); 2013. p. 267–277.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200630

The First Corpus-Driven Lexical Database of Lithuanian as L2

Jolanta KOVALEVSKAITĖ [1], Loïc BOIZOU, Agnė BIELINSKIENĖ,
Laima JANCAITĖ, Erika RIMKUTĖ
Vytautas Magnus University, Lithuania

Abstract. The article presents a new resource for A2-B2 learners of Lithuanian as L2 to improve their lexical competence and language production skills. The lexical database is a lexicographic application of the Lithuanian Pedagogic Corpus which was used both to develop headword lists and to collect word usage information. For this study, we adopt the inductive procedure of Corpus Pattern Analysis which was partly automated using the Lithuanian Sketch Grammar in Sketch Engine. We explain the model for pattern recognition and description, sense division, the selection of examples and give some details concerning the user interface.

Keywords. Lithuanian, Lithuanian Pedagogic Corpus, learner lexicography, foreign language learning resources, corpus pattern, Lithuanian Sketch Grammar

1. Introduction

The paper aims to present a new lexical resource for learning and teaching Lithuanian – the first Corpus-driven Lexical Database (henceforth, the database). The target group of the database is A2-B2 learners of Lithuanian (according to *the Common European Framework of Reference for Languages* (henceforth, CEFR)). This on-going lexicographic work is a part of the project "Lithuanian Academic Scheme for International Cooperation in Baltic Studies"[2].

In the first part, we briefly describe the Lithuanian Pedagogic Corpus (henceforth, the corpus), which was used to study word patterning, and explain our strategy for headword list development. As the integration of available Lithuanian corpora in L2 classroom is rather limited, and explanatory Lithuanian dictionaries often lack data that reflects the modern language usage, the main purpose of the database is to provide learners and teachers with relevant material on language use. For the description of word usage, we adopted the inductive procedure of Corpus Pattern Analysis (CPA; Hanks [1, 2]), which consists of several steps: 1) preparation of the Lithuanian Sketch Grammar; 2) adoption of a model for pattern recognition, sense division and pattern description; 3) guideline setting for example selection (see Section 3).

All the information related to word usage was recorded in the XML database MONGO: the entry structure contains organizational/technical data (status, comment, and editor), frequency data from each A1-B2 sub-corpus, a phonetic container (pronunciation, transcription, and the accentuation type for nominal words), a

[1] Corresponding Author: Jolanta Kovalevskaitė; Vytautas Magnus University, V. Putvinskio st. 23-216, Kaunas LT-44243, Lithuania; E-mail: jolanta.kovalevskaite@vdu.lt.
[2] http://baltnexus.lt/en/baltic-studies-project.

morphological container (the part of speech of a headword, inflected forms that appear in the corpus, and the frequency for each form). In the usage container, the record depends on the headword frequency in the corpus: a) for words (and derivatives) with frequency of 100 and above, a full-record is prepared: with patterns, examples and derivatives related to particular word senses; b) for derivatives with frequency below 100, a short-record is prepared: with examples and derivatives related to particular word senses.

Due to financial project limitations, we chose to use open source and (at least, for the project period 2018-2020) freely available tools: the text analysis software Sketch Engine and the Dictionary Writing System Lexonomy. However, after the testing stage, we had to reject Lexonomy because of data buffering problems. Besides, the experience with some experimental Sketch Engine functionalities (e.g., OneClick Dictionary) showed that in a real lexicographic project, their use is (still) rather limited.

2. The Structure of the Database

In this part of the paper, we explain our procedure for the headword list development that is based on the word frequency distribution in the Lithuanian Pedagogic Corpus.

2.1. The Lithuanian Pedagogic Corpus

The size of this monolingual specialized corpus is 669,000 tokens. The corpus consists of two sub-parts: written language (A1-A2: 96,000 tokens; B1-B2: 523,000 tokens) and spoken language (A1-A2: 15,000 tokens; B1-B2: 35,000 tokens) (see [3]). The material for the database was collected from the written part of the corpus (618,637 tokens), thus, the database represents the usage of the written Lithuanian language. The data for the written part of sub-corpus was collected from 1) the Lithuanian language coursebooks (17.2%) and 2) a variety of authentic Lithuanian material (82.8%) selected using the criteria of learner-relevant communicative function and genres: news portals, popular science books, advertisements, public information (travelling, health care and other services), etc.

Although the corpus we used is rather limited, the probable usage and meanings [2] of frequent words can be seen in small corpora as well. As pattern recognition for the low frequency words might be more problematic, we analyzed the concordances of at least 100 occurrences of the node word. Accordingly, word patterns were provided only for words with corpus frequency of 100 and above (see 2.2). The other shortcoming is related to the fact that although the corpus includes 29 genres (texts from news portals, stories, fairy tales, advertisements, letters, songs, and others), their distribution is unbalanced.

Some genres make up a bigger part of the corpus than others, and this can influence word patterning: e.g., advertisements amount to 6.4 % of the data, whereas public information texts (e.g., notices in public transport, public catering and healthcare institutions, etc.) are short, and their percentage in the corpus is low (0.3 %). As a result, patterns in advertisements are detected automatically because of frequent usage, and patterning in public texts may not be automatically recognized because of infrequent usage. Thus, frequency is important, but as less frequent usage patterns could also be worth lexicographers' attention, they should be verified by consulting a bigger corpus.

2.2. Core Vocabulary and the Headword List

Since we were aware that corpus division according to CEFR levels is problematic (see [4]), we decided to dismiss the approach of level-linking (e.g., CEFR-graded lexical resources [5]). Instead, we attempted to define a relative core vocabulary, i.e., words that appear in each level or at least in three levels (appr. 7,700 items). However, for the pattern analysis, only words of particular word classes with frequency of 100 and above were selected from this core vocabulary (see motivation in 2.1). Thus, our headword list for CPA analysis includes appr. 700 items that appeared in all CEFR levels from A1 to B2 or from A2 to B2. The headword list consists of nouns, lexical verbs (except auxiliary and modal verbs), adjectives, adverbs (except such deictic adverbs as *čia* 'here', *ten* 'there', *dabar* 'now', *kodėl* 'why') and some numerals (*šimtas* 'hundred', *tūkstantis* 'thousand', *milijonas* 'million'). In the case of primary verbs (e.g., *siekti* 'to seek', *imti* 'to take', *likti* 'to stay'), which can function both as auxiliary and lexical verbs, full record was prepared to show usage differences.

The headword list is being extended with two kinds of headwords: a) some multiword expressions (henceforth, MWEs) including an item from the headword list, b) some word formations (derivatives and compounds) from the core vocabulary related to the items from the headword list. With all MWEs and related word formations added to the headword list, we expect the final headword list of appr. 3,000 lexical items.

In the database, 3 types of MWEs are included: idioms, two-word compounds and proverbs (sayings and greetings). MWEs are identified manually and represented with a short record. The derivatives and compounds are selected from the core vocabulary, but as their frequency is below 100, their usage is demonstrated only with a short record. Word formations are selected manually from the headword list. Words that are not in the headword list are also sometimes indicated as base words and then they are labelled with a special symbol. Occasionally, the relation between a base word and its derivatives is just formal, but not semantic, e.g., *sakyti* 'to express thoughts in words' - *užsakyti* 'to commission, to arrange'; *eiti* 'to go from one location to the other on foot' - *apsieiti* 'to get by'. In such cases, both a base word and its derivative are assigned a special symbol.

3. Recognizing and Defining Corpus Patterns

For the description of word usage, one of the corpus-driven methods, Corpus Pattern Analysis (CPA) [1, 2], was adopted. CPA describes a pattern as a syntagmatic structure with semantic values for arguments, i.e., semantic types populated by lexical sets. CPA draws on the insights of the corpus-driven language analysis and the contextual and functional theory of meaning. Maintaining Sinclair's position that "not single words, but rather words in their contextual patterns are the true bearers of meaning" [6], the meaning of a word is associated with a specific lexical and grammatical environment.

Relying on a slightly modified CPA approach, in this project, the observation and definition of meaning-related patterning were performed using the Sketch Engine[3] and the specially designed Sketch Grammar for Lithuanian. During the analysis of the patterning of a word, we applied both automatic (grammatical patterns) and manual procedures (semantic type identification, sense identification, and example selection).

[3] https://www.sketchengine.eu/.

3.1. Lithuanian Sketch Grammar

The corpus had been previously automatically morphologically annotated with *Semantika.lt* analyser; therefore, it was possible to prepare a morphology-based Sketch Grammar for Lithuanian. Given the intended purpose to extract lexico-grammatical patterns, the aim was to capture some syntactic relations using such categories as the part of speech and case with verb forms (infinitive, participle) and neutral gender for adjectives playing an auxiliary role.

The rules are based on expected typical dependents for given parts of speech:

- For verbs: nouns/pronouns in different cases (except vocative), adjective (for the verb *būti* 'to be'), preposition, infinitive, conjunctions;
- For nouns: preposed adjectives/participles with case agreement, preposed nouns in genitive, some left dependents in dative or genitive (e.g., *įtaka kam*, 'influence on sth/sb') or related through a conjunction (e.g., *klausimas, ar…* 'the question whether…') or a preposition (e.g., *priemonė nuo ko* 'measure against sth/sb');
- For adjectives: prepended adverbs, some left dependents in instrumental or genitive (e.g., *išdidus kuo*, 'proud of sth/sb'), infinitive (e.g., *svarbu matyti* 'important to see'), preposition (e.g., *greitesnis už ką* 'faster than sth/sb') or related through a conjunction (e.g., *keista, kad…* 'it is strange that…', for neutral adjectives only);
- For adverbs: prepended adverbs (e.g., *labai akivaizdžiai* 'very obviously').

For Lithuanian Sketch Grammar, the following 14 dual relations were defined:

has_adj_modifier/is_adj_modifier_of	has_gen_noun/is_gen_noun_of
has_right_modifier/is_right_modifier_of	has_dat_noun/is_dat_noun_of
has_adv_modifier/is_adv_modifier_of	has_loc_noun/is_loc_noun_of
has_noun_modifier/is_noun_modifier_of	has_inf_compl/is_inf_compl_of
has_acc_noun/is_acc_noun_of	has_pred_adj/is_pred_adj_of
has_nom_noun/is_nom_noun_of	has_adp/is_adp_of
has_ins_noun/is_ins_noun_of	has_conj/is_conj_of

To reduce the number of false relations, the syntactic relations were defined inside the sentences and with a limited number of tokens inserted between the considered words. In general, verb-centered relations allow for a maximum of 3 token gap, whereas other structures – for a one-token gap or no gap at all. Another difference of verb-centered relations is that they are bilateral (the related word can be before or after a verb), while for other relations, the related word is expected either to the right or to the left (depending on the relation).

Such an approach has well-known shortcomings. Given the lack of syntactic annotation, some relations can be postulated on the basis of the Sketch Grammar between words that are not directly syntactically related. Furthermore, some cases are quite ambiguous, especially the genitive case, which is mostly used for attributes, but may also express a complement for negative verbs, as well as for several positive verbs. WordSketch selects only those combinations that are described by the rules in the Sketch Grammar. As a consequence, occasionally such automatic WordSketch analysis might not reveal some portion of typical usage (e.g., the parenthesis usage function of the adjective *aiškus* 'clear', combinations with numerals, e.g., *dveji metai* 'two years'). Nevertheless, the Sketch Grammar acts as a filter which prepares a WordSketch for a lexicographer that is then used for the manual analysis of headword patterning.

3.2. Corpus Pattern Analysis Applied

In the database, we provide a systematic description of word usage patterns formed by grammar and lexis while analyzing words with the frequency of 100 and above. A corpus pattern includes grammatical (syntactic functions and morphological categories of case and verb forms), lexical (words and collocations) and semantic (semantic types) components. A language feature (collocate, grammatical form, and syntactic function) has to occur at least 3 times in the corpus to be analyzed as a pattern element.

Pattern recognition. First, frequent syntagmatic pattern(s) for each word were identified by the Sketch Grammar. Word usage overview provided in the WordSketch helps lexicographers to make initial hypotheses about meaning-related patterning. Sometimes valency alone is sufficient to make the semantic distinction [2]: in the case of the verb *reikšti* ('to mean'), it is the object in dative which differentiates two verb meanings – *nurodo, žymi* ('indicates, signifies') and *turi vertę* ('has a value'):

[Sub_nom] [REIKŠTI] [Obj_acc]: Geltona spalva reiškia saulę (The yellow colour means the sun.)

[Sub_nom] [Obj_dat] [REIKŠTI] [Obj_acc]: Ką Tau reiškia pokalbis? (What does a conversation mean to you?)

Generally, collocations and semantic types are also needed for sense distinction, thus, the final sense binding to patterns is performed only at the second stage of the analysis.

As the WordSketch provides two-element grammatical patterns, a lexicographer decides where the boundaries of the pattern are. Usually, due to their government structure, verb patterns have more elements than noun, adjective and adverb patterns. On the other hand, it is known that some nouns are used as adverbs (e.g., *daugybė* 'multitude', *daugelis* 'most, many'), predicative adjectives are used as verbs (e.g., *vertas* 'worthy', *pilnas* 'full'), thus, the analysis and description of their patterns are different in comparison with those of nouns or adjectives.

Sometimes several elements function as one pattern component (adverbial, subject, object, attribute). In such cases, manual work is needed to identify them properly and integrate them into the pattern, e.g., *gero būdo žmogus* 'an easygoing person' was described by an attributive pattern which consists of two components [AtrA BŪDAS_sg.gen] [Mod]: attribute [AtrA BŪDAS_sg.gen] and modifier [Mod], which is realized semantically by a semantic type [human] and lexically – by a collocate *žmogus* 'a person'.

The information of morphological forms is very important in the identification of patterns, because sometimes the forms that realize a word in the corpus show that the word is only used as a parenthesis, e.g., only the form *vadinasi* ('it turns out') of *vadintis* ('refl. to call') is used.

After the grammatical patterns are sketched, the second part of the procedure begins: a lexicographer examines collocates provided by WordSketch in each grammatical pattern (unlike in CPA, we do not evaluate collocates by statistical significance), and sorts collocates into lexical sets – a group of words that share one or more semantic feature, e.g., collocates *wedding, festival, concert* form a lexical set, which is then used to define a semantic type 'event' of one of the arguments in a particular pattern.

Semantic types are often the main separators of meanings, especially when two verb senses are associated with the same grammatical pattern, e.g., when using the verb *skambinti*, both meanings 'to phone' and 'to play' are realized by the grammatical pattern [Sub] [SKAMBINTI] [Obj_ins], but the [Obj_ins] is a semantic separator, because for the first meaning it is realized by a semantic type [device] (to call <u>by telephone</u>) and for the second meaning – by a semantic type [musical instrument] (to play the <u>piano</u>).

Besides, the meaning ‘to phone’ is realized by 5 patterns, while the meaning ‘to play’ has one pattern.

The analysis of semantic types according to the CPA procedure has to be performed with a preliminary ontology. We did not use any ontology, but for collocates that are verbs and adjectives, we used a predefined finite set of semantic types: 3 types for verbs (active, state, independent) and 3 types for adjectives (physical, classifying, evaluative). For nouns, following the bottom-up approach, the list of semantic types was non-finite: more types can be added depending on the context of a word.

Sometimes, as mentioned by the CPA practitioners [2], it is problematic to decide on the appropriate level of semantic generalization for a semantic type, e.g., too broad semantic types like [abstract] could be not sufficient to make important semantic distinctions. In our case, this problem is sometimes related with the size of the corpus: e.g., for adjective *naujas* (‘new’) we have to semantically categorize the collocates *kontaktai* ‘contacts’, *giminės* ‘relatives’, *augintinis* ‘a pet’, *galerija* ‘a gallery’, *skonis* ‘a taste’, and because each of them goes to a different semantic type, it is difficult to generalize them semantically.

When lexical and semantic elements are integrated into the grammatical patterns, every pattern (or patterns) is linked to a specific sense. Sense division is based only on patterns, while the existing explanatory dictionaries of Lithuanian were used only in problematic cases.

Pattern description. We used the model for pattern description which consists of grammatical categories and some rules how to show separate elements and their variability.

While learning such a morphologically rich language as Lithuanian, it is important to master the cases and grammatical forms. In pattern description, it is necessary to indicate a case and, quite often, verb forms. For this reason, we provide a multilevel description of a pattern, i.e., grammatical (gramForm), semantic (semForm) and lexical (collocates) realizations are given separately, e.g.,

"gramForm": [ARBATA] [su AtrN_ins] / [TEA] [with AtrN_ins]
"semForm": [Mod] [maistas] / [Mod] [food]
"collocates": [ARBATA] [su citrina] / [TEA] [with lemon]

In the collocate line, collocates are indicated as a lemma; however, fixed forms, and multi-word constructions (cf. *su citrina*) can also be given.

We can see in the pattern above that the word under analysis is capitalized, and separate components in fields "gramForm, "semForm" and "collocates" are surrounded by square brackets. The variability in the pattern is indicated with a vertical slash ‘|’ (‘either – or’). For example, in [Pred] [CENTRAS_acc]||[CENTRAS_ins]: the object is expressed in accusative or instrumental. For the grammatical description of the pattern, morphological categories are marked using Leipzig glossing rules and syntactic categories are marked by international abbreviations (Sub, Obj, Pred, etc.), taken from the syntactically annotated Lithuanian corpus ALKSNIS [7].

Linking patterns to senses and examples. As already mentioned, each sense of a headword is represented with one (or more) pattern(s). The examples were sorted according to different corpus patterns. Lexicographers can provide a sense description for one or more patterns, but the database users will be provided only with patterns and examples.

Explanations of the meaning are not included for several reasons. First, this resource is meant to train learner’s encoding skills. Findings from several studies presented by [8] support the idea that examples indeed seem to help language production. Second, there seems to be no consistent correlation between learners' preferences of dictionary explanations and their success in encoding (see the discussion by [9]). Added equivalents could be a good option for resource development, as they can be used for the same purpose as explanations.

Our approach to example selection was not automated by GDEX facility of Sketch Engine, because our example selection principle was based on described corpus patterns. The grammatical, lexical and semantic components of a pattern help to collect corpus examples that are typical. To ensure that examples are informative and clear, we avoided rare words, figurative usage, and field-related terms. Some examples were slightly edited (shortened or with inserted explanations to clarify anaphora). Usually, the example is one sentence, but in some cases more, than one sentence is given – this helps to illustrate some MWEs (sayings or idioms) in a broader context.

The number of examples depends on the number of collocates for each semantic type; therefore, the approximate frequency of a pattern can be seen from the number of examples. Encoding examples which illustrate patterns and collocations dominate. On the other hand, decoding examples which contain contextual clues about the meaning are also included. As our headword list is not CEFR-level graded, we do not aim to select examples which correspond to the level of an item.

4. User Interface

As the development of user interface is now in progress, we will only provide its brief description. We plan to offer two search options – the search in the headword list and the search in the collocates list. In the pattern description, we provide frequently used collocates from each semantic type (see Section 3), but not all of those collocates are headwords, thus, the collocate search gives the user a possibility to see more collocational networks, which is important for the development of lexical competence. By selecting a particular word, the users will be provided with patterns and examples associated with each word sense. The current form of grammatical realization description may be difficult to understand for learners, thus, we are searching for options to simplify the representation for the end-user.

In the development of some possible end-user scenarios, we address two user groups – language learners (who have reached intermediate level A2) and teachers. For A2 learners, examples, pronunciation, and inflection could be the first relevant option. More advanced learners could be interested in corpus patterning, examples or derivatives. Meanwhile, teachers might benefit from both examples and corpus patterns: they can be used to explain lexical and grammatical environment related to a word (word sense) or to prepare learning material (e.g., exercises with pattern analysis, comparison, pattern-sense relation detection, etc.).

5. Conclusions

In the paper, we explained the application of CPA for Corpus-driven Lexical Database, and mentioned some problematic issues concerning its application. While recognizing and defining corpus patterns, a real challenge for lexicographers is to remain flexible in their observation task (not to be limited only to the repertoire of preselected categories) and, at the same time, to follow the guidelines. To ensure the consistency in pattern description, we apply the cross-validation approach commonly used in dealing with corpus annotation – every full-record is checked by two lexicographers. Working with the morphologically annotated corpus, we partly automated the grammatical pattern recognition at least at the beginning of using the Sketch Grammar, but for the broader application of CPA in (learner) lexicography, more tools could be used in the process of both pattern recognition and description (e.g., [10]).

Given the limited scope of the project and the mentioned limitations of the corpus, we consider our approach for headword list as reasonable. Nonetheless, it would be important to do more research in the future to evaluate the extent to which this headword list represents the basic vocabulary of Lithuanian as L2. One of the promising approaches could be the one demonstrated by [11].

Word patterns may provide valuable data for language learning and teaching, but application possibilities depend on the functionalities of the user interface which is now under development. The lexical database will be freely available for users on kalbu.vdu.lt in 2021.

References

[1] Hanks P. Corpus pattern analysis. In: Williams G, Vessier S, editors. Proceedings of the 11th EURALEX International Congress. Vol. 1; 2004 Jul 6-10; Lorient, France: Université de Bretagne-Sud; 2004. p. 87-97.
[2] Hanks P. How people use words to make meanings: semantic types meet valencies. In: Boulton A, Thomas J, editors. Input, process and product: developments in teaching and language corpora. Brno, CZ: Masaryk University Press; 2012.
[3] Kovalevskaitė J, Rimkutė E. Mokomasis lietuvių kalbos tekstynas: naujas išteklius lietuvių kalbos besimokantiesiems. (Pedagogic Corpus of Lithuanian: a new resource for learning and teaching Lithuanian as a foreign language.) Sustainable Multilingualism. Forthcoming 2020.
[4] Boizou L, Kovalevskaitė J, Rimkutė E. Lithuanian Pedagogic Corpus: correlations between linguistic features and text complexity. In: Proceedings of the 9th International Conference Human Language Technologies – the Baltic Perspective, Baltic HLT; 2020 Sep 22-23; Kaunas, Lithuania. Forthcoming 2020.
[5] François Th, Gala N, Watrin P, Fairon C. FLELex: a graded lexical resource for French foreign learners. In: Calzolari N, et al., editors. Proceedings of International Conference on Language Resources and Evaluation, LREC 2014; 2014 May 26-31; Reykjavik, Iceland: European Language Resources Association (ELRA); 2014. p. 91-102.
[6] Sinclair J. Corpus, concordance, collocation. Oxford, UK: Oxford University Press; 1991.
[7] Rimkutė E, Bielinskienė A, Kovalevskaitė J, Boizou L, Aleksandravičiūtė G. Lithuanian Treebank ALKSNIS, CLARIN-LT digital library in the Republic of Lithuania [Data file]. Kaunas, Lithuania; 2017. [cited 9 Jul 2020]. Available from: http://hdl.handle.net/20.500.11821/10
[8] Frankenberg-Garcia A. Dictionaries and encoding examples to support language production. International Journal of Lexicography. 2015; 24(4):490-512.
[9] Moon R. Explaining meaning in learners' dictionaries. In: Durkin Ph, editor. The Oxford Handbook of Lexicography. Oxford, UK: Oxford University Press; 2016. p. 123-143.
[10] Baisa, V, El Maarouf I, Rychlý P, Rambousek A. Software and data for corpus pattern analysis. In: Horáck A, Rychlý P, Rambouseket A, editors. Proceedings of the 9th Workshop on Recent Advances in Slavonic Natural Language Processing, RASLAN 2015; 2015 Dec 4-6; Brno, Czech Republic: Tribun EU; 2015. p. 75-86.
[11] Brezina V, Gablasova D. Is there a core vocabulary? Introducing the New General Service List. Applied Linguistics. 2015 Feb; 36(1):1-22.

Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200631

Error Tagging in the Lithuanian Learner Corpus

Jūratė RUZAITĖ [1], Sigita DEREŠKEVIČIŪTĖ, Viktorija KAVALIAUSKAITĖ-VILKINIENĖ and Eglė KRIVICKAITĖ-LEIŠIENĖ
Vytautas Magnus University, Lithuania

Abstract. This paper is a work-in-progress report on error annotation in the Lithuanian Learner Corpus (LLC), which has been developed using the TEITOK environment. The LLC is the first electronic corpus of learner Lithuanian that represents learners of very diverse native language backgrounds and different proficiency levels. In this paper, we have a double aim: firstly, we present the structure of the corpus in its current state; and secondly, we describe the main principles, procedures, and challenges of error annotation in the LLC. The main types of errors that are tagged in this corpus and analysed in this paper are orthographic, lexical, and syntactic.

Keywords. Error annotation, leaner corpus, Lithuanian, TEITOK

1. Introduction

The present study is a work-in-progress report on error annotation in the Lithuanian Learner Corpus (LLC), which is currently still under development but is approaching its final stages. In this paper, we shortly overview the structure of the corpus in its current state and lay our primary focus on the main principles, procedures, and challenges of the process of error annotation mainly focusing on written texts.

Learner corpora have become a conventional empirical resource in studies of Second Language Acquisition (SLA) and language teaching/learning (e.g. [7]). The earliest and most numerous learner corpora have been compiled for English, e.g. the International Corpus of Learner English ([10]), TOEFL11 ([1]), Longman Learners' Corpus ([6]), or the Cambridge Learner Corpus ([15]). In recent years, however, learner corpora have been developed for a large variety of other languages, such as Arabic, Jinan Chinese, Korean, Persian, Czech, Dutch, Portuguese, Spanish, Italian, German, Estonian, Gaelic, Hungarian, Norwegian, Latvian and Lithuanian, Russian, and Slovene ([2]).

The landscape of learner corpora is currently quite diverse not only in terms of the target languages that such corpora represent but also regarding their overall size (ranging from around 50,000 to over 1 million words) and internal constitution. Concerning the latter, learner corpora can represent a different variety of text types (ranging from homogeneous corpora of, for example, solely academic writing to corpora comprising all types of written assignments, exams, and oral communication), L1 backgrounds

[1] Corresponding Author: Jūratė Ruzaitė, Centre of Intercultural Communication and Multilingualism, Vytautas Magnus University, K. Donelaičio g. 58, LT-44248 Kaunas, Lithuania; E-mail: jurate.ruzaite@vdu.lt.

(ranging from a single L1 to more than 60 languages), medium of communication (ranging from exclusively spoken or written texts to both spoken and written texts), educational institutions (covering a single institution or involving multiple institutions), or proficiency level (ranging from a single level to the full scope of A1-C2).

Lithuanian as a foreign language (henceforth LFL), being a lesser used and lesser taught language, in general has been studied to a rather limited extent (e.g. [3], [16], [17], [18]), and learner corpora were not available for a rather long time. This new corpus is the only digital text repository that represents a broad spectrum of LFL in terms of text types, native language backgrounds, and institutions where LFL is taught. It is also the only corpus of this size to be annotated for errors. The corpus ESAM (https://esamtekstynas.wordpress.com/) also represents learner Lithuanian, but it is limited to the beginner level and only Latvian as L1; it is also considerably more limited in size (52,000 tokens) ([22]).

It has become well established that error tagging is important in learner corpus annotation, since it allows for identifying standard and deviant forms, which in turn can help to pinpoint problematic areas in the language learning/teaching process ([9]). Error annotation has been done in a variety of languages, and error taxonomies have been developed for French ([8]), Czech ([11], [19]), Portuguese ([5]), Norwegian ([20]), Hungarian ([14]), Latvian ([4], [22]), and to some extent Lithuanian ([22]). The TEITOK interface, applied in this project, has been used for error annotation in the Croatian Learner Text Corpus (CroLTeC), the Baltic language corpus ESAM, and the Learner Corpus of Portuguese L2 (COPLE2; [5]).

2. Design and Main Features of the Lithuanian Learner Corpus

The LLC contains written and spoken data collected from LFL learners not only in Lithuania but also other countries, such as Germany, Sweden, Georgia, and China. It includes texts written by beginning (level A1; 102,952 tokens), pre-intermediate (level A2; 99,303 tokens), intermediate (level B1; 62,940 tokens), and upper-intermediate learners of Lithuanian (level B2; 37,639 tokens). In total, the corpus consists of 302,834 tokens. The disbalance between the lower and upper levels results from the fact that there are relatively few learners of Lithuanian who reach levels B1 and B2.

As the distribution of spoken and written texts presented in Table 1 shows, written texts form the majority of texts in each level (from 80 % to 62 %) and are more numerous in A1-A2 mainly because the oral output at this level is still rather restricted in length.

Table 1. Distribution of spoken and written texts

Mode	A1	A2	B1	B2
Written	75,561 (73 %)	79,842 (80 %)	39,514 (63 %)	23,165 (62 %)
Spoken	27,193 (27 %)	19,461 (20 %)	23,426 (37 %)	14,474 (38 %)
Total	**102,952**	**99,303**	**62,940**	**37,639**

Since written texts dominate in the LLC, we focus here on error tagging in this mode; besides, the scope of the paper does not allow discussing in greater detail the amendments that the spoken part requires.

The age span in the LLC ranges from 16 to 70 years of age, but most of the speakers are 18-26 (totalling 220,025 tokens, or 72.7 % of the entire corpus); see Figure 1.

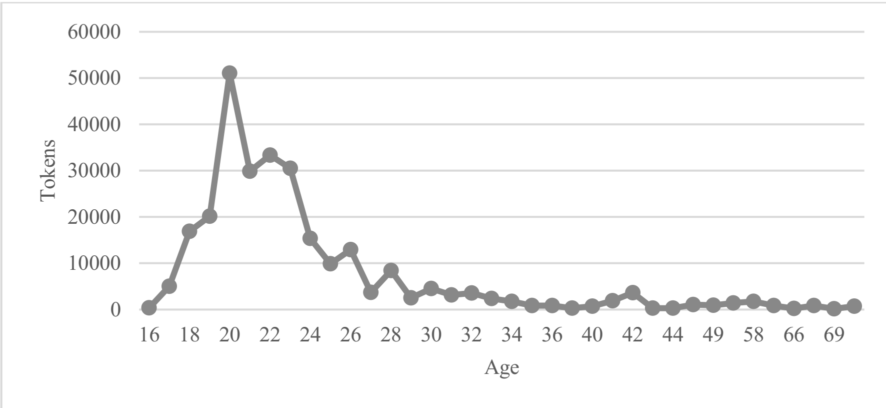


Figure 1. Corpus distribution by age

The dominant age span reflects the fact that the majority of learners in the LLC are undergraduate and graduate students. Approximately two thirds of the learners are female speakers (208,755 tokens as opposed to 94,079 tokens produced by male speakers). The learners come from over 50 different L1 backgrounds, and 55 learners indicated that they are bilingual or multilingual.

In terms of genres, the written subcorpus contains mainly descriptive essays (184,022 tokens), epistolary texts including letters, postcards, and emails (15,963 tokens), argumentative essays (6,409 tokens), and narrative texts (6,118 tokens). None of the other genres (literary essays, chats in a social network, or written dialogues) exceeds 1,000 words, and thus they form only a small minority of texts. In the spoken part, most of the recordings are semi-structured interviews of a teacher with a student, and only a small portion includes presentations (5,609 tokens).

The corpus uses the TEITOK programme developed by Maarten Janssen (2014-, http://www.teitok.org/), which is "a web-based framework for corpus creation, annotation, and distribution, that combines textual and linguistic annotation within a single TEI based XML document" ([13]). The TEITOK interface integrates linguistic annotation and search functions and offers the function of error tagging (for an overview of error tagging options, see [5]).

Thus, the transcriptions in the LLC are stored as TEI compliant XML files consisting of the transcription and a header with metadata. The latter includes the proficiency level, genre of the text, mode of communication, type of the task, use of reference tools, age, sex, the first language(s), foreign languages, mother's and father's first language, home language, education, educational institution, and the length of the text in words. The files are visualised in a user-friendly way in the TEITOK environment, as shown in Figure 2.

Figure 2. Visualisation of a written transcription

The transcribed text appears together with the scanned original, which has multiple advantages: it allows for verifying the accuracy of the transcription, presents the stimulus (the task of the assignment), displays the teacher's corrections (including not only verbal, but also non-verbal mark-up such as underlining, question marks, or explanatory comments), and provides possibility for multimodal research on learner data, which sometimes includes some drawings, schemes, graphs, and other visual elements.

3. Analysis of Error Categories in the LLC

Error tagging in the TEITOK environment is performed on tokenised data by following different types of taxonomies, which include taxonomies marking the source of error (orthography, lexis, and syntax) and taxonomies based on formal types of alternation of the source text (omission, addition, splitting, and merging) (cf. [11], [12], [21]). In addition to the forms suggested by the annotator, the software allows for marking the student form of each token versus the teacher form of the token (see also [5]). TEITOK also provides the possibility to normalise the learner's text by inserting omitted tokens, splitting, and merging them.

Drawing on the model followed in other TEITOK-based corpora, the annotation of deviant language forms in the LLC works at the token level and distinguishes three types of errors: syntactic, lexical, and orthographical errors. Having the same taxonomy for different corpora using the same environment allows for more systematic comparisons across different languages.

The taxonomy used in the LLC is thus based on rather broad categories and offers quite coarse granularity. However, even such a limited level of detail involves challenging tasks and even more so for a language such as Lithuanian, which has rich inflection, derivation, and agreement. Error tagging, which is inevitably guided by the annotator's intuition at least to some extent, does require an analytical framework that would be based on grounded choices made by the annotator to minimise arbitrariness. Thus, further on we overview which more specific categories, or subtypes, fall into the three broad error types by discussing how different ambiguities in error identification and interpretation were solved and what choices were made by the research team.

3.1. Orthographic Errors

At the orthographical level, errors are limited to the word form. In the LLC, punctuation marks, differently from the Portuguese learner corpus ([5]), are not considered. They are tagged as a distinct error category in Znotina's work ([22]), which we consider to be more relevant than categorising it under orthography. However, in the initial stages of the corpus development we decided not to annotate punctuation, since it often does not receive a sufficiently systematic approach in the language teaching curriculum.

Orthographical errors in the LLC mainly include misspellings resulting in omitted/substituted letters, misuse of capitalisation, missing or misused diacritics, misuse of long and short vowels, misspelt diphthongs, merging or splitting morphemes (agglutination), and spelling peculiarities arising due to sound assimilation (for examples, see Table 2).

Table 2. Subtypes of orthographic errors

Error subtype	Example in LFL	EN translation
Omission	*negali nuspęsti* (=nuspręsti)	'you can't decide'
Addition	*mokyklios* (=mokyklos)	'school' (sg.gen)
Substitution	*productus* (=produktus); į sporto clubą (=klubą)	'products (pl.acc); 'to the sports club'
Diacritics	*nera* (=nėra)	'is not'
Capitalisation	*apie Amerikiečių* (=amerikiečių) *kultūrą*	'about American culture'
Long vs. short vowels	*mažas kambaris* (=kambarys)	'small room' (sg.nom)
Sound assimilation	*bendrabutio* (=bendrabučio)	'dormitory' (sg.gen)
Diphthongs	*studijouju* (=studijuoju); *vasių* (=vaisių) *pyragas*	'study' (3sg.pres); 'fruit cake'
Agglutination	*ne susitiksi* (=nesusitiksi); *vistiek* (=vis tiek)	'you will not meet'; 'anyway'

Some of these error subtypes also appear in Znotina's ([22]) taxonomy for Lithuanian and Latvian as second languages; she identifies diacritics, agglutination, upper / lower case (for capitalisation), and 'other spelling errors'.

Perhaps the most challenging are those instances when a deviant form is ambiguous and can be interpreted as an error in orthography or syntax, e.g. *Mano šalis turi jūra.* ('My country has a sea.'). Here the noun *jūra* should appear in the accusative form *jūrą* but '-a' is used without the diacritic and thus has the form of the nominative case. However, it is impossible to know if the learner misused the inflection of the nominative case (which would result in a syntactic error) or intended to use the inflection for accusative but did not add the diacritic to it (which would result in an orthographic error). We followed the principle that if the student form exists in native Lithuanian (e.g. *jūra*), but it does not fit in the grammatical context of the sentence, it is considered to be an error in syntax, not orthography, since a difference in the word form results in a different grammatical form. An orthographic error appears when a deviant form results in a non-existent form in standard native Lithuanian.

3.2. Syntactic Errors

The syntactic level covers grammatically deviant forms, that is, errors that affect syntactic structures. Most of the errors in this category include morphology errors (illustrated in Table 3). Examples of such errors mainly comprise agreement problems (subject-verb, verb-object, modifier-noun, etc.), inaccuracies in the verb form (mood, voice, conjugation, reflexivity, etc.) and noun form (case, number, declension, gender, etc.), part of speech errors (e.g. adjective vs. adverb), errors in the use of prepositions, and agreement between prepositions and nouns. We also ascribe question words (as a category of function words) to the area of syntax.

Table 3. Subtypes of syntactic errors

Error subtype	Example in LFL	EN translation
Case ending	*Nebeturiu vietą* (=vietos)	'I don't have the place anymore'
Noun declension	*pirkti užsienietiškus prekius* (=užsienietiškas prekes) *savo šalyje*	'buy foreign goods in one's own country'
Number	*Ventspilis turi daug tako ir parko* (=takų ir parkų)	'Ventspils has a lot of paths and parks'
Countable/uncountable	*Aš valgau bandeles, tartus ir duonas* (=tortus ir duoną)	'I eat buns, cakes and bread'
Reflexive verb	Leiskite prisistatyti apie mano šalį (=pristatyti mano šalį)	'Let me introduce my country'
Person	*Kai aš buvo vaikė* (=buvau vaikas)	'When I was a child'
Agreement	*visokie skirtingos renginai* (=skirtingi renginiai)	'all sorts of different events'
Derivation	*Valdauja* (valdo); *radau toksį suoliuką* (radau tokį suoliuką)	'rules'; 'I found such a bench'
Verb conjugation	*Ji užaugė* (=užaugo) *kaime*	'She grew up in the countryside'
Voice	*kada autobusas bus atvažiuotas* (=atvažiuos) *pagal tvarkaraštį*	'when the bus comes according to the schedule'
Mood	*daugelis iš mūsų konservuoja agurkus ..., kad žiemą yra* (=būtų) *atsargos.*	'many of us can cucumbers … so that we stock up for the winter.'
Prepositions	Daug jaunų žmonių išvažiavo užsienyje (=į užsienį)	'Many young people went abroad'
Pronoun form	*Aš* (=Man) *patinka mano miestas*	'I like my city'
Adverb vs adjective	*Jie ieško darbo ir geriau* (=geresnio) *gyvenimo*	'They look for work and a better life.'
Question words	*Is kur tu studiojuje?* (=Kur tu studijuoji?)	'Where are you studying?'

In general, syntactic errors also include word order errors, but these were corrected in the LLC only when absolutely necessary. Lithuanian is a highly synthetic language and thus allows for a high degree of flexibility in word order, since usually more than one morpheme indicates the relations between different syntactic units. Alternatives in syntactic patterns in Lithuanian are difficult to assess since they can be used for different stylistic effects but strictly grammatically are still acceptable. Our approach seems to be more flexible than Znotina's ([22]) taxonomy; in her research, a stricter approach to word order is applied and some syntactic patterns presented as examples of inaccurate word order would not be counted as errors in our corpus.

3.3. Lexical Errors

Lexical errors (illustrated in Table 4) are restricted to word choice and meaning. At this level, the word used by the learner is orthographically and grammatically correct but is not the most natural choice for a native speaker in terms of word meaning and/or collocability. In some rarer cases, a lexical unit does not follow the word formation rules (a derivational affix is misused) or a foreign word is used as a loan with a Lithuanian inflection.

Table 4. Subtypes of lexical errors

Error subtype	Example in LFL	EN translation
Prefixation	*Ilgai* supgalvojau (=galvojau) *apie tai; suspaustas* (=išspaustas) *sultis*	'I was thinking long about it'; 'squeezed juice'
Collocability	*tai yra dalykas, kuris keičiasi laikui* *skrendant* (=bėgant)	'this is something that changes as times passes'
Word choice	*ne vienas negali keltis nuo stalo kol* *vienas* (= kas nors) *dar valgo.*	'no one can leave the table while someone is still eating.'
Word formation	*šaltakariu* (=šaltojo karo) *pabaiga*	'the end of the cold war'
Loan	*Stvetaforas* (=šviesoforas) *yra prie teatro.*	'The traffic lights are near the theatre.'

As demonstrated in Table 4, we consider misuse of prefixation as a lexical error. It is an ambiguous subtype since some prefixes can also mark perfectivity (as in *galvojau* vs *sugalvojau*, where the latter refers to a completed action and is perfective) and as such can be assigned to the syntactic error category (cf. [22]). However, we take the stance that prefixation in many cases leads to semantic changes and lexicalization, and its impact on word meaning cannot be explained solely in grammatical terms (as in *suspaustas* vs *išspaustas*, where both forms are perfective, but there is an important semantic difference between the two).

Finally, it needs to be noted that a typical learner of Lithuanian makes errors across all linguistic levels, and a single token may result in more than one correction, e.g. a misspelt word may also be used with a non-standard inflection. Such multi-level errors are also marked using the TEITOK annotation tool.

4. Conclusion

This new error-tagged Lithuanian learner corpus with a rich XML-encoding opens new research areas as well as possibilities for practical applications in language teaching/learning. Error tagging can provide qualitative data about the types of errors in LFL and quantitative information about the distribution of these error types across different learner groups/texts. Such data can help develop an inventory of difficulties typical of the learner population in general and those that are restricted to a certain L1 background. By containing complete metadata, it allows for relating learners' errors to sociolinguistic parameters, e.g. the person's linguistic background, age, or gender.

The error taxonomy discussed here still needs refining as well as further testing by performing an inter-annotator agreement evaluation to assess the accuracy of the system. A more fine-grained annotation could be developed to account for more types/subtypes of errors. Further quantitative analysis of error types could lead to some insights about learners' difficulties; however, such analysis needs to be carried out with caution especially when comparisons between different languages are made since there are some differences in the internal structure of learner corpora and annotation systems even if a common tool for developing them is used. Despite the slippery areas that exist in such research, we believe that this new corpus will provide language instructors and researchers with valuable authentic data about learners' interlanguage so that better-grounded teaching and testing materials can be developed.

References

[1] Blanchard D, Tetreault J, Higgins D, Cahill A, Chodorow M. TOEFL11: A Corpus of Non-Native English. Technical report, Educational Testing Service; 2013.
[2] Centre for English Corpus Linguistics: Learner Corpora around the World. Louvain-la-Neuve: Université catholique de Louvain; 2020. https://uclouvain.be/en/research-institutes/ilc/cecl/learner-corpora-around-the-world.html.
[3] Dabašinskienė I, Čubajevaitė L. Acquisition of case in Lithuanian as L2: error analysis. Eesti Rakenduslingvistika Uhingu Aastaraamat. 2005; 5:47-66.
[4] Deksne D, Skadina I. Error-annotated corpus of Latvian. In: Utka A, Grigonytė G, Kapočiūtė-Dzikienė J, Vaičenonienė J, editors. Human Language Technologies – The Baltic Perspective; 6. Amsterdam: IOS Press; 2014. p. 163-166.
[5] Del Río I, Antunes S, Mendes A, Janssen M. Towards error annotation in a learner corpus of Portuguese. In: Proceedings of the Joint Workshop on NLP for Computer Assisted Language Learning and NLP for Language Acquisition. Umeå: LiU Electronic Press; c2016. p. 8-17.
[6] Gillard P, Gadsby A. Using a learners' corpus in compiling ELT dictionaries. In: Granger S, editor. Learner English on Computer; London: Longman; 1998. p. 159-171.
[7] Granger S. A bird's-eye view of learner corpus research. In: Granger S, Hung J, Stephanie Petch-Tyson S, editors. Computer learner corpora, second language acquisition and foreign language teaching. Amsterdam: John Benjamins; 2002. p. 3-33.
[8] Granger S. Error-tagged learner corpora and CALL: a promising synergy. CALICO Journal. 2003; 20(3): 465-480.
[9] Granger S. Computer learner corpus research: current status and future prospects. In: Connor U, Upton T, editors. Applied Corpus Linguistics: A Multidimensional Perspective. Amsterdam & Atlanta: Rodopi; 2004. p. 123-145.
[10] Granger S, Dupont M, Meunier F, Naets H, Paquot M, editors. International Corpus of Learner English. Version 3. UCL: Presses Universitaires de Louvain; 2020. 227 p.
[11] Hana J, Rosen A, Škodová S, Štindlová B. Error-tagged learner corpus of Czech. In: Proceedings of the Fourth Linguistic Annotation Workshop. Uppsala: Association for Computational Linguistics; c2010. p. 11-19.
[12] James C. Errors in language learning and use. Exploring error analysis. London: Longman; 1998. 320 p.
[13] Janssen M. TEITOK: Text-faithful annotated corpora. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Portorož; c2016. p. 4037-4043.
[14] Langman J. Analyzing second-language learners' communication strategies: Chinese speakers of Hungarian. Acta Linguistica Hungarica 1997;44(1/2):277-299.
[15] Nicholls, D. The Cambridge Learner Corpus – error coding and analysis for lexicography and ELT. In: Archer D, Rayson P, Wilson A, McEnery T, editors. Proceedings of the Corpus Linguistics 2003 Conference; Lancaster: Lancaster University; c2003. p. 572-581.
[16] Ramonaitė J. Kaip lietuviškai šneka užsieniečiai? Lietuvių kaip antrosios kalbos veiksmažodžio įsisavinimas. Baltistica 2015; L (2):295-330.
[17] Ramonaitė J. Bendratis lietuvių kaip antrojoje kalboje. Baltistica 2017; LII(1):81-104.
[18] Ramonaitė J. Ką sako tokios užsieniečių sudaromos formos kaip *valgu* ar *žinėjau*? Lietuvių kalba 2017;11:1-25.
[19] Rosen A, Hana J, Štindlová B, Feldman A. Evaluating and automating the annotation of a learner corpus. Language Resources and Evaluation 2014; 48:65-92.
[20] Tenfjord K, Meurer P, Hofland K. The ASK corpus – A language learner corpus of Norwegian as a second language. In: Proceedings from 5th International Conference on Language Resources and Evaluation (LREC), Genova: ELRA; c2006. p. 1821-1824.
[21] Tono Y. Learner corpora: design, development and applications. In: Proceedings of the Corpus Linguistics 2003 Conference. Lancaster: Lancaster University; c2003. p. 800-809.
[22] Znotina I. Computer-aided error analysis for researching Baltic interlanguage. In: Dislere V, editor. Proceedings of the 11th International Scientific Conference. Education. Personality (REEP); Jelgava: Latvia University of Life Sciences and Technologies; c2017. p. 238-244.

Subject Index

Author Index

This page intentionally left blank