Human Language Technologies – The Baltic Perspective
K. Muischnek and K. Müürisep (Eds.)

doi:10.3233/978-1-61499-912-6-150

General-Purpose Lithuanian Automatic Speech Recognition System

Askars SALIMBAJEVS [a,1], Jurgita KAPOČIŪTĖ-DZIKIENĖ [b]
[a] *Tilde, Vienibas gatve 75a, Riga, Latvia*
[b] *Tilde IT, Naugarduko g. 100, Vilnius, Lithuania*

Abstract. This paper describes the development of a general-purpose automatic speech recognition system for Lithuanian. The system is capable of performing both the transcription of user submitted audio recordings and real-time speech-to-text conversion. The comparative evaluation results prove that the presented system outperforms all other ASR systems for the Lithuanian language. The system also includes number and date normalization and is paired with an automatic punctuation restoration model that achieves state-of-the-art results for the Lithuanian language. Importantly, the system is publicly available to any Lithuanian speaker for testing via its web-page and mobile application.

Keywords. automatic speech recognition, Lithuanian language, general domain, post-processing, punctuation restoration

1. Introduction and Related Work

Automatic speech recognition (ASR) technology is becoming an integral part of modern life: it is used to control devices, search the web, dictate e-mails or ideas, transcribe meetings, etc. All this demand encourages many academic researchers and global companies (such as Google, Microsoft, IBM, Baidu, Apple, Amazon, Nuance, SoundHound, IflyTek, etc.) to work on ASR. However, the main focus of research is on widely used languages, e.g., present research on English yields human parity by achieving word error rates (WER) of 5.8% and 11% [1]. The challenge of adapting the technology and methods to other languages (and their dialects) still remains.

Unfortunately, the Lithuanian language (having 3.2 million speakers worldwide) is not very attractive from the commercial point of view. Moreover, the Lithuanian language specifics create additional challenges and complexity, e.g., absolutely different phoneme list, much larger vocabulary, rich word derivation system, and free word order in a sentence. Despite all this, the first important steps have already been taken in the area of the Lithuanian ASR, experimenting with dictated speech (isolated words, short phrases, and continuous speech) and spontaneous speech (partially spontaneous and telephone speech).

ASR research for the Lithuanian language started in 2002, when the system for the automatic digit recognition (described in [2]) was offered. This system used Dynamic

[1] Corresponding Author. E-mail: askars.salimbajevs@tilde.lv

Time Warping with Linear Predictive Coding. One of the earliest researches for isolated word recognition (described in [3]) uses the triphone Gaussian Hidden Markov Model applied on Mel Frequency Cepstral Coefficients (MFCC). The pronunciation dictionary of isolated words contains only $\sim$750 words. The large vocabulary ASR for broadcast speech recognition for the Lithuanian language is described in [4].

A lot of the previous research on Lithuanian ASR is based on the idea of adapting existing other language ASR, the Lithuanian words are transcribed using the Universal Phone Set, and no Lithuanian acoustic models are trained. Experiments were performed with the Microsoft Office Communication Server package for the English, German, French, and Spanish languages on digits [5] or medical phrases [6]. In [7], the recognition is done by the neural network model, which computes the final recognition decision from predictions of several other language recognizers.

Also, Lithuanian is one of the development languages within the IARPA BABEL research program [8], and broadcast speech recognition for Lithuanian was addressed within the Quaero research program. In [9], a semi-supervised method was used to train an acoustic model with only 3 hours of transcribed data and 360 hours of untranscribed data. Unfortunately, we can not compare this result with our ASR, as Quaero systems and evaluation data are not publicly available.

Summarizing all previously described research works, the major efforts are directed towards 1) low-resource, semi-supervised methods, 2) adaption of existing acoustic models, and 3) broadcast speech recognition and keyword spotting. For most of the described systems, it is not clear if they are being used in practice. The important research, representing a real working ASR system for the Lithuanian language, is described in [4]. This system is not publicly available but is currently used to monitor various broadcast speech sources.

Hence, the Lithuanian language still requires accurate dictation software, ASR for stenography, and mobile dictation applications. Consequently, the main contribution of our research is to offer the accurate general-purpose large-vocabulary Lithuanian ASR, which (if needed) could be adapted to some specific tasks/domains. Moreover, the system is also publicly available to Internet users `https://www.tilde.lt/snekos-technologijos` and as the mobile application "Tildės Balsas" (`https://play.google.com/store/apps/details?id=com.tilde.tildesbalsas`).

2. Tilde ASR System

We present the Lithuanian large-vocabulary ASR system, developed by *Tilde*, which is based on the open-source Kaldi toolkit [10].

2.1. Pronunciation Model

Lithuanian language has a highly phonemic orthography, i.e., pronunciation and spelling mostly correspond in a predictable way. For our ASR system, we use a simplified grapheme-based approach by treating each grapheme as a separate phoneme. Consequently, the phoneme repository consists of 29 grapheme-based phonemes, 1 unified filler-silence model, and 1 model for fragmented and out-of-vocabulary words.

However, not all words (e.g., acronyms and brand names) can be correctly transcribed using this model. For instance, acronym *JAV* (USA) is pronounced as “jav”, but

Table 1. Text corpora used for the LM training.

Corpus	Sentences	Tokens
Sentences with brand names	347,082	105,097,735
Sentences with acronyms+proper nouns (Google search)	2,393,057	245,020,034
Sentences with acronyms+proper nouns (Yahoo!)	142,672	2,566,817
Lithuanian web news portals	55,531,263	769,309,016

LNK as "el-en-ka", *NBA* as "en-bė-a" or "en-bi-ei". Due to this, these specific text elements are stored in the additional pronunciation list (1,527 acronyms and 6,828 brand names, with at least one pronunciation option for each). Candidates (mostly acronyms and brand names) for this list were collected semi-automatically by 1) scanning texts and searching for non-generic words in the upper or title cases and 2) manually reviewing the collected list and adding the correct pronunciation (or several possible pronunciation options). Besides, foreign brand names (simple and compound) are inflected (the same as common nouns in the Lithuanian language) by adjusting the appropriate inflection paradigm; therefore, the brand name pronunciation list contains not only the original form (*Facebook*) but also all inflected forms (*Facebookas*, *Facebooko*, *Facebookui*, *Facebooką*, *Facebooku*, *Facebooke*).

2.2. Language Model

For the language model (LM) training, we have used several text corpora that were automatically harvested from the web (see Table 1). The main text corpus is the 50M sentence Lithuanian web news corpus, and other corpora were collected to improve the recognition of specific types of words. These corpora include sentences (crawled from *Google* and *Yahoo!*) containing specific keywords: acronyms (1,527 tokens), person names (33,979), place names (5,284), and brand names (6,828).

To make these corpora suitable for our speech recognition task, the following preprocessing procedures were taken:

1. Tokenization, filtering (e.g., mixed case tokens, non-alphanumeric tokens), true casing.
2. Rewriting number tokens from digits to words in Lithuanian language by using the num2words Python library.[2]
3. Automatic spell-checking with the proprietary Tilde morphology tool: sentences with $\geq$2 incorrect words were filtered out.
4. Vocabulary extraction. The vocabulary is created from words that are marked by spell-checker as correct.
5. Limiting vocabulary. After the previous steps, the size of the vocabulary is about 2M word forms (because of inflections and complex morphology), which is too large for practical ASR. Due to this, we reduce the vocabulary to the $\sim$600K most frequent words, which gives the out-of-vocabulary rate $<= 1\%$. These words were concatenated with the handcrafted list of exceptions (containing person names, places, brand names, etc.), i.e. words that spell-checker frequently misrecognizes, and the final vocabulary contains $\sim$670K word forms.

[2] num2words library can be found at `https://github.com/savoirfairelinux/num2words`.

The pre-processed text corpus was used to train the following n-gram LMs with the Kneser-Ney smoothing:

- A heavily pruned 2-gram model for the first-pass decoding.
- A big 3-gram unpruned model for the lattice rescoring.

2.3. Acoustic Model

The acoustic model (AM) is an important integral part of ASR, which is used to represent the relationship between an audio signal and phonemes.

This ASR system uses the Kaldi recipe for sequence discriminative training of Time-Delay Deep Neural Network (TDNN) acoustic models[11] and iVectors for speaker adaptation [12]. For training AM, three speech corpora were used:

- $\sim$100h Lithuanian speech corpus (52h of pure speech, 11K word forms, 61K utterances, 360 speakers) was collected during the LIEPA project.[3] The small-vocabulary LIEPA corpus is valid for the limited domain ASR only: i.e., for queries, voice commands.
- $\sim$192h Seimas corpus (308K utterances) (more about it in [13]) was automatically created from crawled Seimas session video recordings from 2015-2017. Recordings and aligned edited text transcripts are taken from the Seimas webpage.[4]
- $\sim$20h dictated speech corpus (21K utterances) contains manually annotated dictated speech recorded using various mobile devices (smartphones, voice recorders, laptops).

Each corpus is augmented by performing three-way speed-perturbation (0.9x and 1.1x speed) and reverberation (using simulated impulse responses [14]). As the result, the total approximate training data size is $\sim$1500h. For additional robustness, about a half of the data is randomly passed through a low-pass filter that emulates telephone recordings.

2.4. Postprocessing

To make the ASR transcript more readable and to decrease the human post-editing time, a number of post-processing procedures were implemented.

2.4.1. Punctuation Restoration

A transcript without punctuation is difficult to read even if it is segmented by the speaker diarisation system. For punctuation restoration, we use the machine translation approach and the Transformer neural network model to translate a non-punctuated ASR text into a sequence of punctuation labels (one label for each word, e.g., "leave as is", "prepend comma", "prepend period", etc.). More details on this can model will be published in [15].

The model relies only on the pure textual data and is trained on texts from Lithuanian web news portals. The text is filtered similarly as for the LM training, excluding the punctuation filtering. Punctuation is retained but processed using special handcrafted

[3] The project site is `https://www.xn--ratija-ckb.lt/liepa`.

[4] The Seimas webpage `http://www.lrs.lt/`.

filters that discard sentences containing dubious punctuation, e.g., unclosed quotation marks, repeating punctuation (like ",,,,."), and emoticons.

The punctuation restoration quality of the model was evaluated on the held-out set from the training data, which contains ~25K sentences. The results are presented in Section 3.2.

2.4.2. Number and Date Normalization

All numbers in the ASR transcript are written in words but have to be converted into a digit form, e.g., *du šimtai penkiasdešimt trys → 253*. This conversion is done using a finite state transducer (FST) defined by a handcrafted JSGF grammar[5].

The grammar rule set processes and converts cardinal and ordinal numbers in different inflection forms (including pronominal). Also, it contains special rules for date formating and can produce the following abbreviations (from different inflection forms): “m.”, “mėn.”, “d.” from “metai” (year), “mėnuo” (month), “diena” (day), respectively. For instance, *du tūkstančiai aštuonioliktųjų metų kovo trečiąją dieną → 2018 m. kovo 3 d.*. The grammar rule set can be further filled and modified depending on the need.

2.4.3. Word Capitalization

Some words can be written only in capitalized form, e.g., “Lietuva” (Lithuania) and “Vytauto Didžiojo universitetas” (Vytautas Magnus University), so they are present in the LM only in capitalized form. However, there are words that can be written both lowercased or capitalized, depending on the context. For example, common words like “energetika” (energetics) or “komitetas” (committee) have to be in the title-case in meaningful phrases like “Energetikos komitetas” (Energy Committee). Moreover, these phrases can be up to several words in length, e.g., “Kriminalinės žvalgybos parlamentinės kontrolės komisija” (Criminal Intelligence Parliamentary Control Commission), and according to the Lithuanian grammar rules, only the first word in the phrase (if it does not contain proper nouns) is title-cased.

We tackle this problem in 3 ways:

1. We allow the ASR vocabulary to contain words both in capitalized and lowercased form. This enables the ASR system to choose a capitalized word by looking at the context. For example, 2-gram “Lietuvos Respublika” (Republic of Lithuania) probably will have smaller LM cost than 2-gram “Lietuvos respublika”, but both will have equal AM cost.
2. We train our Transformer model to do both punctuation and capitalization by forcing it to output labels like "prepend comma and capitalize current word".
3. Finally, there is a semi-automatically created list of exceptions that can be tweaked for specific applications. The current list contains 1,025 unique items of party and parliamentary groups, committees, laws, ministries, departments, etc. Usually, only the last word of such phrases is inflected (e.g., “Energetikos ministerija”, “Energetikos ministerijos”, “Energetikos ministerijai”, etc.), so they can be presented as their stem followed with the specific identifier “*” (i.e., “Energetikos ministerij*”). Such representation allows searching for phrases in all inflection forms. Post-processing script searches for such phrases in the transcript and replaces them with the correct casing.

[5] More details about the JSpeech Grammar Format can be found at `https://www.w3.org/TR/jsgf/`

3. Experiments and Results

3.1. Speech Recognition

The developed ASR system for Lithuanian was evaluated using the standard word error rate (*WER*) metric on the following manually annotated test corpora:

- *test_general*, a 1-hour "general domain" set of audio segments from various radio and TV shows (mainly *Atviras pokalbis* and *Labas rytas, Lietuva*).
- *test_seimas*, a 6-hour set of randomly picked utterances from Seimas sessions recordings from 2016-2017.
- *test_lt_radio*, a 2-hour set of audio segments from Lithuanian radio (news, talk shows, and advertisements).

Unfortunately, test corpora and ASR systems used in many Lithuanian speech recognition research papers usually are not publicly available. Therefore, we had to perform evaluation on our test sets and compare only to the systems we could get access to: Google Cloud Speech and Alumäe&Tilk [4]. The results are summarized in Table 2.

3.2. Punctuation and Capitalization Restoration

The Transformer model for punctuation restoration was evaluated separately on the held-out set of web-portals text corpus (see Table 3). For comparison, we also evaluated the Punctuator2 [16] model trained on the same data (vocabulary size is increased to 200K and default values are used for other parameters).

The same sentences were used to evaluate the model's capitalization restoration performance. The achieved precision, recall, and F-score are equal to 98.9%, 99.3%, and 99.1%, respectively.

Table 2. The WER(%) evaluation of ASR systems on different data sets.

Test set	Domain	Google	Alumäe&Tilk [4]	Ours
test_general	General domain	40 (ignoring deletions - 26)	25.2	**21.8**
test_lt_radio	Radio broadcast	54 (ignoring deletions - 27)	32.3	**29.2**
test_seimas	Seimas	41 (ignoring deletions - 26)	28.4	**21.3**

Table 3. Evaluation of punctuation restoration on held-out set from web-portals corpus.

Model	Classes	Precision, %	Recall, %	F1-score, %
Transformer	Comma	86.2	85.9	86.1
	Period	89.6	90.1	89.9
	Average	87.9	88.0	88.0
Punctuator2	Comma	86.0	82.5	84.2
	Period	85.1	85.5	85.3
	Average	85.6	84.0	84.8

3.3. Keyword Recognition

Brand names, acronyms, and other keywords were added to the pronunciation and LM; therefore, it is important to evaluate the recognition of such potentially problematic words. Since test sets used to evaluate WER are not keyword-rich, we prepared a separate small corpus of short audio recordings ($\sim$10 words per recording). Each recording contains at least one keyword ($\sim$1,200 keywords in total). These recordings are excerpts from news, television shows, and advertisements. In order to make the evaluation more objective, we do not force keywords in our test set to be present in the pronunciation list and LM. The WER calculated only on the keywords is $\sim$52%, meaning that every second problematic word (acronym, foreign person, brand name, etc.) can be recognized with our generic ASR system.

4. Discussion

The results demonstrate that the developed ASR system for the morphologically complex Lithuanian language achieves reasonable and comparable WER on the test corpora of diverse domains. Since the presented ASR system is not adapted to some particular task or domain, it can be treated as “general-purpose”.

The comparative analysis revealed that our system outperforms all other evaluated ASR systems on all test data sets, even on the broadcast speech radio test set, where Alumäe&Tilk [4] should have an advantage.

Google Cloud Speech ASR demonstrated the worst WER. This is mainly due to a high number of deletions, which occur because Google Cloud Speech aggressively filters out words for which it has low confidence. However, even if deletions are not considered as errors, the WER of Google ASR is still high (26% on *test_general* and on *test_seimas*).

The results also show that our neural network-based Transformer punctuation model outperforms the Punctuator2 model (trained on the same corpus), while having more than 4 times less parameters (12M vs 53M), and therefore can be considered as state-of-the-art. The results achieved both for our model and for Punctuator2 are better compared to the results described in [4]; this is probably because we have more training data and restore only commas and periods. Also, it is possible that this test set is just simpler because we apply filtering of potentially incorrect punctuation on both training and testing sentences.

5. Conclusion and Future Work

In this paper, we presented the general-purpose ASR for the Lithuanian language. The implemented system achieved promising results on the general domain test corpus (WER=$\sim$21.8) and other test corpora. Moreover, our system is publicly available (from web-page or mobile application) and outperforms other publicly available ASR systems for the Lithuanian language.

The presented ASR system could be improved by increasing the amount/diversity of data for the AM training, by filling exception lists with new keywords, by expanding the corpus for the LM training, by improving text corpora processing, by adding new post-processing rules, etc. All these directions for improvement are in our future plans.

6. Acknowledgments

The research has been supported by: 1) EU Investment Fund under 2014–2020 Economic Growth Operational Programme's 1st priority "Scientific Research, Experimental Development and Innovation Promotion" measure "Intellect. Joint Science-Business Projects" (invitation No. J05-LVPA-K-01) within the project *Automatic speech recognition, synthesis and creation and improvement of adaptation technologies in R & D ("Speech Cloud")*, No. J05-LVPA-K-01-0034; 2) European Regional Development Fund within the project *Neural Network Modelling for Inflected Natural Languages*, No. 1.1.1.1/16/A/215.

References

[1] Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu and Geoffrey Zweig. Achieving Human Parity in Conversational Speech Recognition. *IEEE/ACM Trans. Audio, Speech & Language Processing* **25(12)** (2017), 2410–2423.

[2] Antanas Lipeika, Joana Lipeikienė and Laimutis Telksnys. Development of Isolated Word Speech Recognition System. *Informatica*, **13(1)**, 2002, 37-46.

[3] Gailius Raškinis and Danutė Raškinienė. Building Medium-Vocabulary Isolated-Word Lithuanian HMM Speech Recognition System. *Informatica*, **14(1)**, 2003 75-84.

[4] Tanel Alumäe and Ottokar Tilk. Automatic Speech Recognition System for Lithuanian Broadcast Audio. *Baltic HLT, Frontiers in Artificial Intelligence and Applications*, 289 (2016), 39–45.

[5] Rytis Maskeliūnas, Algimantas Rudžionis, Kastytis Ratkevičius and Vytautas Rudžionis. Investigation of Foreign Language Models for Lithuanian Speech Recognition. *Electronics and Electrical Engineering*, **3(91)**, 2009, 15-20.

[6] Vytautas Rudžionis, Gailius Raškinis, Rytis Maskeliūnas, Algimantas Rudžionis and Kastytis Ratkevičius. Comparative Analysis of Adapted Foreign Language and Native Lithuanian Speech Recognizers for Voice User Interface. *Elektronika ir elektrotechnika*, **19(7)**, 2013, 90-93.

[7] Tomas Rasymas and Vytautas Rudžionis. Combining Multiple Foreign Language Speech Recognizers by using Neural Networks. *Human Language Technologies – The Baltic Perspective*, 2014, 33-39.

[8] Mary Harper. The BABEL program and low resource speech technology. *ASRU*, (2013)

[9] Rasa Lileikyte, Arseniy Gorin, Lori Lamel, Jean-Luc Gauvain, Thiago Fraga-Silva. Lithuanian Broadcast Speech Transcription Using Semi-supervised Acoustic Model Training. *SLTU 2016*, (2016) 107-113.

[10] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Nagendra Goel, Mirko Hannemann, Yanmin Qian, Petr Schwarz, Georg Stemmer. The Kaldi Speech Recognition Toolkit. *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding* (2011).

[11] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, Sanjeev Khudanpur. Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI. *Interspeech*, pp. 2751-2755. 2016.

[12] Miao Yajie, Hao Zhang, and Florian Metze. "Towards speaker adaptive training of deep neural network acoustic models." *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[13] Askars Salimbajevs. Creating Lithuanian and Latvian Speech Corpora from Inaccurately Annotated Web Data. *11th edition of the Language Resources and Evaluation Conference (LREC 1018)*, 2018.

[14] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L. Seltzer, Sanjeev Khudanpur. A study on data augmentation of reverberant speech for robust speech recognition. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5220-5224. IEEE, 2017.

[15] Andris Varavs and Askars Salimbajevs. Restoring Punctuation and Capitalization Using Transformer Models. *6th International Conference on Statistical Language and Speech Processing (SLSP 2018)*, (in press).

[16] Tanel Alumäe and Ottokar Tilk. Bidirectional Recurrent Neural Network with Attention Mechanism for Punctuation Restoration. *Interspeech 2016*, (2016).