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

doi:10.3233/978-1-61499-442-8-236

Towards Large Vocabulary Automatic Speech Recognition for Latvian

Askars SALIMBAJEVS[a,1] and Mārcis PINNIS[a]
[a] *Tilde, Vienibas gatve 75a, Riga, Latvia*

Abstract. In this paper, the authors present the results of ongoing research on Large Vocabulary Automatic Speech Recognition for the Latvian language. The paper describes the initial acoustic model, phoneme set, filler and noise models, and grapheme-to-phoneme modelling. The second part of this work is focused on language modelling. Different word and class-based n-gram models are evaluated in terms of perplexity and word error rate in a speech recognition task. The authors also train a recurrent neural network language model and use it for n-best rescoring.

Keywords. Speech recognition, large vocabulary, Latvian, acoustic modelling, language modelling, grapheme-to-phoneme modelling

Introduction

In recent years, the success of spoken interfaces in smartphones and tablets has prompted new excitement about automatic speech technologies. This success has stimulated many developers to embrace speech technologies for their native languages.

However, most of the research in speech recognition and speech synthesis (as well as ready-made tools and language resources) is usually available only for "big" languages, like English, French, German, and Spanish [1]. There are many smaller languages for which speech recognition is not available.

Among the languages of the Baltic countries, Estonian is the most researched language in speech recognition. There have been successful efforts in Estonian speech recognition for both limited and large vocabulary speech recognition tasks [2, 3].

Unfortunately, there is lack of research on Latvian language speech recognition. To the best of our knowledge, there has been only one publicly known research project on Latvian speech recognition, Oparin et al. [4] efforts on broadcast news transcription using acoustic model bootstrapping methods.

The lack of effort can be largely explained by the absence of a labelled speech corpus, which has only recently become available. This paper describes our first steps towards the goal of creating a Large Vocabulary Automatic Speech Recognition (LVASR) system for Latvian.

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

1. Acoustic Modelling

The quality of speech recognition depends on many parameters, most of which can be grouped into two independent categories: acoustic (e.g., feature extraction parameters, number of states in HMM, phoneme set) and language model (e.g., modelling method, vocabulary size, language model size, etc.).

We started developing our speech recognition system by fixing language model parameters in order to search for the best acoustic model. We used the language model created from the test data at this stage.

The initial acoustic model developed was a 40 phoneme, 3-state Hidden Markov Model (HMM) with 3,000 tied states, each described by 8 Gaussian mixture components. For the initial acoustic modelling, we did not use speaker adaptation, feature transformation, or multi-pass decoding methods, because our goal was to identify how much can be achieved by using core methods before investigating more advanced acoustic modelling and acoustic model adaptation methods.

The model was trained using the CMU Sphinx toolkit [5] with 13-dimensional Mel Frequency Cepstral Coefficient (MFCC) features on a recently published 100-hour Latvian Speech Recognition Corpus (LSRC) [6].

After performing multiple experiments with different feature extraction parameters, we identified parameters which work best with LSRC. The most notable changes were: (1) using DCT-II transform instead of the default "legacy" transform and (2) adding liftering (cepstrum filtering).

When the initial model was trained, we used it in experiments with different phoneme sets, voiced fillers, and grapheme-to-phoneme models. The PocketSphinx [7] decoder from the CMU Sphinx toolkit was used for decoding and evaluating speech recognition quality according to Word Error Rate (WER). At this stage, we achieved a WER of 14% when using the language model from test data.

After all of the abovementioned acoustic and G2P model parameters were determined, the number of Gaussians per state was increased to 32, and the acoustic model was retrained. The retrained model achieved 12% WER when using the language model from test data and was later used in experiments with different language models.

2. Phoneme Set, Filler Word, and Noise Models

As large vocabulary ASR models use phonemes to recognise words, it is important to select an optimal phoneme set which would allow us to (1) acquire sufficient statistics for phonemes and build more accurate models and (2) unambiguously and effectively recognize words from recognised phoneme sequences.

We performed multiple experiments with different phoneme sets and identified a baseline phoneme set that contains: (1) 33 phonemes which have a one-to-one correspondence to letters of the Latvian alphabet and (2) 4 diphthongs ([ai], [au], [ɛi] and [iɛ]) which were selected empirically.

This phoneme set allowed for the best WER to be achieved. Any small deviation from this baseline set resulted in a small increase of WER (see Table 1).

Humans rarely communicate using read or prepared speech, therefore a good speech recognition system must be able to deal with the defects of spontaneous speech [8].

Table 1. Experiments on changing the baseline phoneme set

Phoneme set description	Word error rate
Baseline phoneme set	13.7%
Phonemes [o] and [uɔ] are distinguished	14.2%
Phoneme [ss] is added	14.1%
Phoneme [c] is removed and combination of [t]+[s] is used instead	14.2%
Long phoneme [ā] is removed	14.9%
[ɛi] diphthong is removed	13.9%
[ui] diphthong is added	14.2%

Table 2. Noise and filler models

Noise/filler	Occurrences in training corpus
[e], [ē], and their variations	13,192
[m] and its variations	1,060
[a], [ā], and their variations	1,263
[h], [hmm], [kh], etc.	126
Mix of [n], [en], [s], [u]	162
Mix of rare voiced fillers	114
Non-speech noise	4,481
Breathing	45,041
Laughing	431
Silence	18,290

In this work, we train noise and filler models for filtering out non-speech noises and voiced pauses. The LSRC has 107 unique labels for voiced pauses and non-speech events. It quickly became evident that training such a large amount of noise/filler models is not effective. Some of the models are almost identical, while some are too rare for acquiring sufficient statistics for training and generalisation of reliable acoustic models. Therefore, only 9 generic models were trained (see Table 2).

3. Grapheme-to-phoneme Model

A grapheme-to-phoneme (G2P) model describes the mapping between a sequence of phonemes and a word. In its simplest form, a G2P model is a dictionary – a list of words and their corresponding canonical phonetic pronunciations.

Since there is a very strong correspondence between graphemes and phonemes in Latvian spelling, we used a grapheme based model. A simple rule-based G2P algorithm was developed, which basically maps letters directly to phonemes using one-to-one correspondences. The algorithm also tries to recognise and use the 4 diphthongs mentioned earlier, which is its only difference from a pure grapheme based approach.

When training the initial acoustic model, the phone tree crossing option was turned on. This allowed CMU Sphinx to tie HMM states of different phonemes, which is very useful for grapheme based models. When using this option in our setup, together with our basic G2P mapping, the WER is improved by 3-5% over a setting where the option is turned off.

3.1. Evaluating G2P

The LSRC includes 4 hours of data that are annotated both phonetically and orthographically. This makes it possible to evaluate the quality of the automatic G2P conversion algorithm.

Table 3. Effect of different G2P errors

Error description	G2P phoneme error rate	Word error rate
Insertion of extra phoneme "g" after "k" in some cases	+ 24% absolute	+ 0.5% absolute
Insertion of extra phoneme "k" after "e" in some cases	+ 15% absolute	+ 3% absolute
Using separate phonemes instead of diphthongs	+ 4.8% absolute	+ 0.5% absolute
Deletion of phoneme "o" after consonants in some cases	+ 0.5% absolute	+ 2% absolute
Substitution between similar sounds "p" and "b"	+ 4.8% absolute	+ 0.2% absolute

The G2P and LSRC phoneme sets are not identical, but we can transform the LSRC phoneme sets into the phoneme set used by the rule-based G2P. Depending on how we define this transformation, we can obtain a phoneme error rate of 9-13% for our G2P algorithm. Almost half of these errors are substitutions, deletions constitute less than 1% of the errors, and the remaining errors are insertions (i.e., a phoneme is inserted by the rule-based G2P where the human transcription does not have one).

We also performed several experiments in order to understand how different G2P errors can influence the WER. In these experiments, we injected synthetic errors into the G2P algorithm, retrained the acoustic model, and calculated the WER. Table 3 shows a few examples of these experiments. It was concluded that there is no simple correspondence between the G2P error rate and the WER, because the resulting WER depends on the character of specific errors, e.g., two G2P algorithms can have a similar phoneme error rate, bet a very different WER. Some errors such as substitutions between similar phonemes have little effect on WER, while errors such as the insertion of an extra phoneme can lead to noticeable WER degradation.

Given that about 50% of current G2P errors are insertions, it seems that there is a potential for improvement of WER.

3.2. Advanced G2P

In order to address and lower the number of insertion errors, we trained two more complex G2P models. The first one was trained with Phonetisaurus [9], which utilises weighted finite-state transducers for decoding a representation of a grapheme-based n-gram model trained on data aligned by an advanced many-to-many alignment algorithm (which is a variant of the EM algorithm [10]). The second one is a statistical machine translation (SMT) model which translates from "grapheme" language to "phoneme" language [11].

Both models were evaluated on a small held out data set from the phonetically annotated corpus. The phoneme error rate for both models is given in Table 4.

Table 4. Advanced G2P models

G2P model	Phoneme error rate
Phonetisaurus WFST model	5.24%
Statistical machine translation	3.26%

Both models achieved significantly better results than our rule-based G2P algorithm. The superiority of the SMT model can be explained by the fact that the Phonetisaurus model is trained on a pronunciation dictionary which was extracted from the phonetically annotated 4 hour corpus and includes all pronunciations from this corpus. At the same time, the SMT model was trained on the full phoneme transcriptions of the training set, not just isolated words. This allows the SMT model to take into account word boundaries and phonemes from adjacent words. After training is done, the SMT model is used to translate all transcriptions. This translation is then processed, and a static pronunciation dictionary with multiple pronunciation variants is created.

Despite the better phoneme error rate, no improvements in WER were observed. Moreover, the result degraded significantly in the case of the SMT-based G2P model, because it introduced a lot of ambiguous pronunciation variants. This ambiguity comes directly from the nature of human speech which is captured by precise training labels. As a result, words in our static dictionary have a large number of pronunciations, many of which overlap with the pronunciations of other words, making the "recovery" of the right word strings from such a dictionary difficult. This result corresponds with the findings in [12].

As we were unable to improve upon the rule-based G2P, it was also used in later experiments with language modelling.

4. Language Modelling

4.1. Corpus

In order to train language models, we have used a large text corpus, which was collected from several sources (see Table 5).

The text corpus was pre-processed before training language models:

- First, the text was tokenised.
- Then, punctuation, digits, and URL tokens were removed. Only word tokens were kept.
- Finally, vocabularies of different sizes were selected from the most frequent words in the corpus.

After pre-processing, the corpus consisted of 38.5M sentences (40% of them were 10-20 words long) and 592M running words. The vocabulary of the complete text corpus was 2.8M word surface forms.

Table 5. Sources of Latvian monolingual text corpora

Corpus	Type	Description
DGT-TM[13]	Translation memories	Public EU law text collection
OPUS EMEA[14]	Monolingual part of parallel text	European Medicines Agency documents
Localisation TM	Translation memories	Translation memories from software and user manual localisation
WebNews corpus	Monolingual corpus	Collection of texts from Latvian news portals
Other	Monolingual corpus	Some texts from books and internet

Table 6. Experiments with 3-gram models

Model	Perplexity	OOV	WER, %
Test transcripts	70.786	0%	12.5%
Training transcripts	410.456	4.6%	19.9%
Large corpus, 50,000 vocabulary	423.964	8.1%	37.5%
Large corpus, 100,000 vocabulary	528.162	4.3%	48.5%
Small corpus, 50,000 vocabulary	496.401	7.3%	41.7%
Small corpus, 100,000 vocabulary	610.328	4.3%	45.5%

From this corpus, we created a smaller corpus of 3M sentences. We used the Moore&Lewis [15] data selection method and training transcripts from the audio corpus as adaptation data for this task.

4.2. N-gram Models

From both corpora, we trained several 3-gram language models and evaluated those in terms of perplexity on test data transcripts and speech recognition WER (see Table 6).

The best result was achieved by a 3-gram model with the vocabulary size of 50,000 words (results of LM from audio transcripts are given for comparison only). The idea of creating a smaller corpus from adapted sentences turned out to be unsuccessful, while the perplexity of training data transcripts significantly reduced test WER, and perplexity results showed a negative change. The big difference between training perplexity and test perplexity can be a sign of poor generalisation and\or overfitting.

We also tried training 4-gram word and 2-gram class-based language models for lattice rescoring (see Table 7). 200, 500, 1000, 2500, and 5000 classes were induced by automatic word clustering [16]. However, all of these models exhibit very similar perplexity, and therefore, results are given only for the 200 classes.

Rescoring with the 4-gram model resulted in an absolute WER improvement of 1%, while a smaller improvement (about 0.7%) was achieved by rescoring with class-based and 3-gram model interpolation. Further improvement can be achieved (0.22%) by combining 4-gram and class-based models, though this improvement is not significant.

4.3. RNN Language Model

Recurrent neural networks (RNN) are considered as a state-of-the-art language modelling method [17] which gives significantly smaller perplexity than traditional n-gram models.

Table 7. Rescoring with class-based and 4-gram models

Model	Perplexity	WER, %
Rescoring with 200 class model	753.155	41.05%
Rescoring with 4-gram word model	327.022	36.50%
Rescoring with interpolated 200 class + 3-gram model	460.638	36.83%
Rescoring with interpolated 200 class + 4-gram model	441.629	36.28%

Table 8. N-Best (N=200) list rescoring with RNN language model

Model	Perplexity	WER, %
Baseline(no rescoring)	423.964	37.50%
Rescoring with RNN LM	174.682	36.62%

In this work, we use the "rnnlm" toolkit [18] for training and using RNN language models. The goal of this experiment was to validate the fact that RNN models are significantly better than classic n-gram models. Therefore we used the smaller 3M sentence corpus for training, because training an RNN language model on the complete 38M corpus would take a large amount of time. We are planning to train an RNN model on a larger corpus in future research.

As seen in Table 8, the perplexity of the RNN language model is very low in comparison to n-gram models, however, applying n-best list rescoring with RNN LM resulted in a WER improvement of less than 1%. This can be partially explained by the fact that pure acoustic scores were not available in the n-best list. It should be noted that the minimum error rate of n-nest lists used in this test is 20.4%, which suggests that much bigger improvement after rescoring should be possible. The cause of this difference will be investigated in future research.

5. Conclusions

In this paper, we presented a summary of the ongoing research on automatic speech recognition for the Latvian language. We started with basic and classic acoustic modelling methods and also trained multiple n-gram models.

A significant amount of time was spent on searching for the best grapheme-to-phoneme model. We trained a Phonetisaurus WFST based G2P model and an SMT-based G2P model, however, the best result was achieved with the grapheme-based approach extended with several straightforward rules.

Our initial acoustic model was trained using core methods like grapheme based tied-state continuous HMM and MFCC features. For language modelling, we focused on different n-gram models. The best word error rate of 37.5 in a LVASR scenario was achieved by a 3-gram model with a vocabulary of 50,000 words. This result can be further improved by applying rescoring with 4-gram and class-based models. We also tried rescoring with the state-of-the-art RNN language model. While the RNN LM showed a large improvement in reduction of perplexity, the improvement in WER was smaller than with the 4-gram model. The cause of this needs to be investigated.

By comparing our results with [4], we see that the possibilities of achieving significant further improvement with basic methods is limited. In our future work, we are therefore planning to gradually introduce more advanced and modern methods like speaker adaptation, bottleneck features, and others.

Acknowledgements

The research leading to these results has received funding from the research project "Information and Communication Technology Competence Centre" of EU Structural funds, contract nr. L-KC-11-0003 signed between the ICT Competence Centre (www.itkc.lv) and the Investment and Development Agency of Latvia, research No. 2.4 "Speech recognition technologies".

References

[1] Rehm, G. & Uszkoreit, H., editors. (2012). META-NET White Paper Series: Europe's Languages in the Digital Age. Springer, Heidelberg etc. 32 volumes on 31 European languages.

[2] Alumäe, T., Võhandu, L. (2004). Limited-Vocabulary Estonian Continuous Speech Recognition System using Hidden Markov Models. Informatica 15, 3, pp. 303-314.

[3] Alumäe T., Meister E. (2010). Estonian Large Vocabulary Speech Recognition System for Radiology. In *proceedings of the 2010 conference on Human Language Technologies – The Baltic Perspective*. pp. 33-38. Amsterdam, The Netherlands: IOS Press.

[4] Oparin, I., Lamel, L., & Gauvain, J. (2013). Rapid Development of a Latvian Speech-to-text System. In *proceedings of ICASSP'13.* pp. 2-6. Vancouver, Canada.

[5] Lee, K.F., Hon, H. W., & Reddy, R. (1990). An overview of the SPHINX speech recognition system, IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 38, no. 1, pp. 35–45.

[6] Pinnis, M., Auziņa, I., & Goba, K. (2014). Designing the Latvian Speech Recognition Corpus. In *proceedings of the 9th edition of the Language Resources and Evaluation Conference (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA).

[7] Huggins-Daines, D., Kumar, M., Chan, A., Black, A.W., Ravishankar, M., Rudnicky, A.I. (2006). Pocketsphinx: A Free, Real-Time Continuous Speech Recognition System for Hand-Held Devices. In *proceeding of the 2006 IEEE International Conference on Acoustics Speed and Signal Processing*.

[8] Butzberger, J., Murveit, H., Shriberg, E., & Price P. (1992). Spontaneous speech effects in large vocabulary speech recognition applications. In *proceedings of the workshop on Speech and Natural Language (HLT '91)*, pp. 339-343, Stroudsburg, PA, USA.

[9] Novak, J. (2011). Phonetisaurus: A WFST-driven Phoneticizer (Version 0.8). Available at http://code.google.com/p/phonetisaurus/.

[10] Jiampojamarn, S., Kondrak S., & Sherif, T. (2007). Applying Many-to-Many Alignments and Hidden Markov Models to Letter-to-Phoneme Conversion. In *proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics NAACL-HLT*. Rochester, NY, USA.

[11] Auziņa, I., Pinnis, M., & Darģis, R. (2014). Comparison of Rule-based and Statistical Methods for Grapheme to Phoneme Modelling. In *Human Language Technologies – The Baltic Perspective - Proceedings of the Sixth International Conference Baltic HLT 2014*. Kaunas, Lithuania: IOS Press.

[12] Saraçlar, M., Nock, H., & Khudanpur, S. (2000). Pronunciation modeling by sharing Gaussian densities across phonetic models. Computer Speech & Language 14, pp. 137-160.

[13] Steinberger, R., Eisele, A., Klocek, S., Pilos, S. & Schlüter P. (2012). DGT-TM: A freely Available Translation Memory in 22 Languages. In *proceedings of the 8th international conference on Language Resources and Evaluation (LREC'2012)*. Istanbul, Turkey.

[14] Tiedemann, J. (2009). News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. In N. Nicolov, K. Bontcheva, G. Angelova & R. Mitkov (eds.) Recent Advances in Natural Language Processing (vol V), pp. 237-248, Amsterdam/Philadelphia: John Benjamins

[15] Moore, R. C., & Lewis, W. (2010). Intelligent Selection of Language Model Training Data. In *Proceedings of the ACL 2010 Conference.* pp. 220–224.

[16] Brown, P., Della Pietra, V., deSouza, P., Lai, J., and Mercer, R. (1992). Class-Based n-gram Models of Natural Language, Computational Linguistics 18(4), pp. 467-479.

[17] Mikolov, T., Karafiát, M., Burget, L., Cernocky, J. (2010). Recurrent neural network based language model. In *proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)*. Makuhari, Chiba, Japan.

[18] Mikolov, T., Kombrink, S., Deoras, A., Burget, L., & Černocký, J. (2011). RNNLM - Recurrent Neural Network Language Modeling Toolkit. *ASRU 2011 Demo Session*.