# SENTENCE-ADAPTED FACTORED LANGUAGE MODEL FOR TRANSCRIBING ESTONIAN SPEECH

*Tanel Alumäe*

Department of Phonetics and Speech Technology
Institute of Cybernetics at Tallinn University of Technology
Akadeemia tee 21, 12618 Tallinn, Estonia

## ABSTRACT

This work presents a 2-pass recognition method for highly inflected agglutinative languages based on an Estonian large vocabulary recognition task. Morphemes are used as basic recognition units in a standard trigram language model in the first pass. The recognized morphemes are reconstructed back to words using hidden event language model for compound word detection. In the second pass, the vocabulary from N-best sentence candidates from the first pass is used to create an adaptive sentence-specific word-based language model which is applied for rescoring the N-best hypotheses. The sentence specific language model is based on the factored language model paradigm and estimates word probabilities based on the preceding two words and part-of-speech tags. The method achieves a 7.3% relative word error rate improvement over the baseline system that is used in the first pass.

## 1. INTRODUCTION

Statistical word n-gram models are the most commonly used language models used in today's large vocabulary speech recognizers. However, in languages where the number of word inflections is large and compound words are written together, the number of distinct words is very large. Therefore, a high out-of-vocabulary (OOV) rate is expected when words are used as recognition units in composing a statistical language model. To overcome this problem, subword units are used as basic language modeling units for agglutinative languages. Subword units may be found using morphological analysis (e.g. [1]), or discovered automatically from large corpora [2]. The first approach has also been used for Estonian speech recognition [3].

The morpheme-based standard trigram language model can fairly well represent the relationships between word particles, such as compounds, stems, suffixes and endings. However, because the span of the language model is limited, it often fails to incorporate relationships between word forms (as a large part of word forms consist of many subwords units). Thus, the language model represents the morphosyntactic information needed to reconstruct word forms, but much less semantic and syntactic relationships between words than for non-agglutinative languages. There have been attempts to overcome this problem by modeling the relationships between stems and endings separately [4]. A rather different approach uses a limited word form based language model already in the first pass, and applies a dynamically generated sentence-specific language model of acoustically similar words in the second pass [5].

In this paper we focus on building a two-pass decoder that incorporates word-level statistical morphosyntactic information. In the first pass, a morpheme-based standard trigram language model is used to generate a N-best list of sentence hypotheses for each utterance. The morpheme sequences are reconstructed to word sequences by using special morpheme tags and a hidden event n-gram model for compound word recognition. In the second pass, a morphological analyzer is used for adding part-of-speech (POS) and stem tags to each word form. The list of all word forms from all hypotheses are used to construct a new word form based language model. The language model uses backing off to the POS tags of the corresponding word, if there is not enough data in the training corpus to estimate the relationship between word forms. Finally, the N-best hypotheses are rescored using the dynamically generated language model, and the resulting scores are combined with the scores from the first pass.

## 2. ARCHITECTURE OF MORPHOLOGY-BASED RESCORING

The objective of this method is to improve the estimation of word form sequence probabilities for agglutinative and compounding languages. The words in such languages are heavily inflected depending on their syntactic role. This makes the number of distinctive words in the language very large. In addition, new words can be freely created by compounding. Thus, the number of different word forms is huge. This creates two problems from the point of statistical language modeling. First, the huge number of unique words makes a full word-based language model very space-consuming. Second, the training data sparsity becomes even a bigger problem, as the number of valid word sequences is much larger than for non-agglutinative languages, and the available text corpora cannot have enough evidence for estimating all needed trigrams, and we would need to back off to bigrams and unigrams in most cases. This would reduce the language model accuracy by a large extent. The proposed recognition strategy tries to overcome those problems.

The architecture of the recognition system using adaptive morphosyntactic language model is shown on figure 1. A decoder using a language model of subword units is used in the first pass. It outputs an N-best list of sentence hypotheses for each sentence, together with their acoustic and language model scores. Each hypothesis is originally a sequence of subword units. The word ending morphemes are specially tagged in the lexicon and can be thus easily concatenated to the preceding stems. It is however not possible to easily identify units that should form compound words, as most compound particle pairs occur also as individual words in the language. To illustrate this, the morpheme sequence ”*ma kohtu _sin kala mehe _ga*” should be transformed to word sequence ”*ma kohtusin*

This research was partly funded by the Estonian Information Technology Foundation as part of the Tiigriülikool program and by the Estonian Association of Information Technology and Telecommunications.

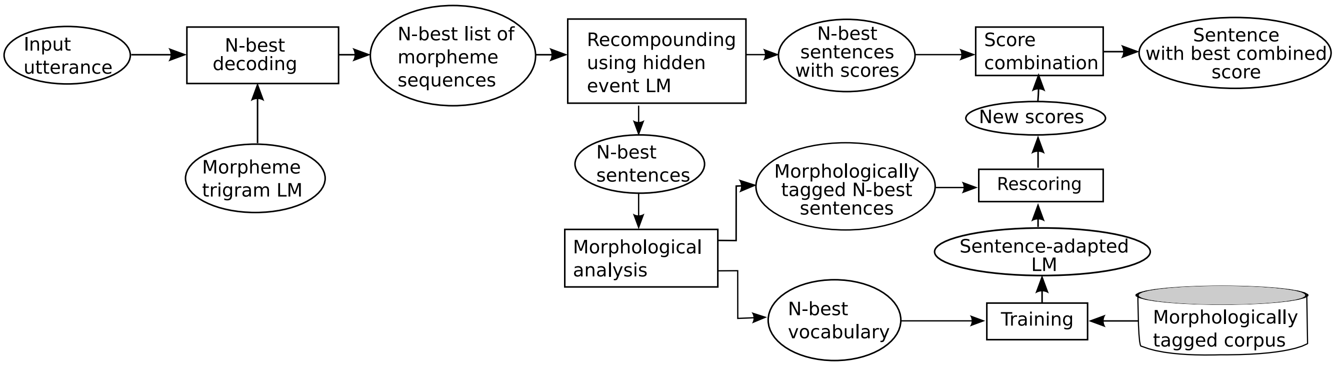
Input utterance
N-best decoding
Morpheme trigram LM
N-best list of morpheme sequences
Recompounding using hidden event LM
N-best sentences with scores
Score combination
Sentence with best combined score
New scores
N-best sentences
Morphologically tagged N-best sentences
Rescoring
Sentence-adapted LM
Morphological analysis
N-best vocabulary
Training
Morphologically tagged corpus

**Fig. 1**. *Architecture of the two-pass recognizer*

*kalamehega*” (meaning ”I met a fisherman”), but a sequence ”*ma näg _in kala mehe _ga*” could transform to ”*ma nägin kala mehega*” (meaning ”I saw a fish with a man”). To disambiguate between compound and non-compound words, a hidden-event language model [6] is used. This language model is a trigram model over all the subword units in our main language model and the hidden ”compound word border” hidden tag. When applied to the output of the decoder, it marks the most likely places where compound words should be formed. The accuracy of this approach has not been measured in detail, but by broad investigation it seems to work reasonably well.

In the next phase, the N-best sentence hypotheses are processed by a morphological analyzer and disambiguator, that attaches POS tags and stem information to each word form.

The morphologically tagged sentence hypotheses are used to create a vocabulary for the dynamic sentence language model. The sentence language model is estimated from morphologically tagged training corpora. To speed up the computation of the model, the n-gram counts of all word forms may be precomputed. The resulting language model will only contain the probabilities that are needed for estimating the scores for the sentence hypotheses for the current sentence, thus the vocabulary is fixed and small, and the OOV rate is effectively zero. However, many of the word form sequences in the N-best list are usually never or only very rarely seen in the training corpora. Therefore, we propose the using of a factored language model (FLM) [7] as the dynamic sentence model. In an FLM, a word is seen as a collection of parallel factors. Factors for a given word can be any linguistic features that correspond to a word. In our case, the factors are the word itself, its POS tag, and it’s stem. In a factored model, a word probability can be estimated based on the preceding POS-tags and/or stems, whenever there is insufficient data to fully estimate the probability based on the preceding words.

After generating the dynamic sentence LM, all N-best sentence hypotheses from the 1st pass are rescored using the new model. The resulting sentence scores are combined with the scores from the first pass and the N-best hypotheses are reordered using the combination of scores. The weights for the scores can be optimized on a development set so as to minimize the word error rate.

This process must be executed for each utterance. In practice, the dynamic language model can be generated for a large batch of sentences, as long as the size of the dynamic vocabulary stays in the allowed bounds of the software and the size of the language model is reasonable. In our experiments, the N-best vocabulary of 320 test sentences was only around 8000, and the dynamic language model was generated for all sentences in one batch.

## 3. BASELINE SYSTEM

### 3.1. Training data

The acoustic models for recognition experiments were trained on the Estonian SpeechDat-like phonetic database [8], collected from volunteer speakers over telephone. Each recording session contains read sentences from a handout sheet, answers to simple questions, short utterances, etc. The number of ”good” recording sessions, including truncated but otherwise acceptable sessions, is 2969. Because many speakers were asked to call 10 times, the total number of different speakers is 1332. The number of acceptable utterances is 177 793. This totals in about 241.1 hours of audio data.

The speech data is recorded at 8kHz sampling rate and coded using 8-bit mono A-law. The recording sessions consist of a fixed set of utterance types, such as isolated and connected digits, natural numbers, money amounts, spelled words, time phrases, date phrases, yes/no answers, person and company names, application words and phrases, phonetically rich words and sentences.

For speech recognition experiments, the database was divided into training, development and test set. Development and test set were composed by randomly selecting 80 speakers out of those who only called once. For speech recognition experiments, only the long sentence utterances were used (8 from each speakers) The utterances were divided into development and test set, which both contained 320 utterances.

For language model training, a part of the mixed corpus of Estonian was used [9]. The used part consists of the texts from two national newspapers (40.5 million words), an academic journal (7 million words) and a corpus of Estonian original fiction (4.2 million words).

For language model evaluation, the transcriptions from the long sentences in the development set of the before-mentioned speech database were used. The texts are relatively neutral in style, resembling more fiction than newspaper articles.

### 3.2. Acoustic modeling

The open source SphinxTrain toolkit was used for training the acoustic models. Models are created for 25 phonemes, the five filler/noise types and silence. All audio was converted from 8-bit A-law to 16-bit linear encoding, as the feature extractor program cannot handle A-law data. For acoustic features, MFCC coefficients were used. The coefficients were calculated from a frequency band of 130 Hz - 3400 kHz, using a preemphasis coefficient of 0.9. The window size was 0.0256 seconds and the frame rate was 100 frames/second. A

512-point FFT was used to calculate 31 filter banks, out of which 13 cepstral coefficients were calculated. All units are modeled by continuous left-to-right HMMs with three emitting states and no skip transitions. The output vectors are 39-dimensional and are composed of 13 cepstral coefficients, delta and double delta coefficients. The final tied-state triphone models have 8000 shared states in total. Each state is modeled by 8 Gaussian mixture components.

The pronunciation dictionary was created from word orthography using a set of context sensitive rewrite rules.

### 3.3. Language modeling

Before language model training, the training corpora were processed by the morphological analyzer and disambiguator that tagged words in the texts with their respective POS and stem tags. The analyzer also marks the boundaries between word compounds and between stems and endings.

Next, the corpus-specific counts for all uniquely tagged words were generated. The list was filtered using the following strategy: all words tagged as abbreviations or numbers were removed (as there is currently no reliable tool for expanding abbreviations and numbers in texts, and therefore we cannot specify pronunciations for such words). Next, all words tagged as names, except for the top 500 for each corpus, were also removed (this was done in order to avoid the over-population of names in the vocabulary, as the newspaper texts contain a lot of names). From the filtered word counts, counts for all unique word particles were generated, again one per training corpus. Finally, the maximum-likelihood vocabulary of 60 000 particles was selected from the mixture of the counts, using the technique implemented in the SRILM toolkit [10]. The transcripts of the sentences in the development set of our speech database were used as the held-out text. The resulting vocabulary has a particle OOV-rate of 2.4%.

Using the vocabulary of 60 000 particles, a trigram language model was estimated for each training corpus. The cutoff value was 2 for bigrams and 3 for trigrams. A modified version of Kneser-Ney smoothing as implemented in SRILM was applied. Finally, a single LM was built by merging the four models, using interpolation coefficients optimized on the SpeechDat development set transcripts. The resulting LM has 2 608 862 bigrams and 4 457 057 trigrams. The unnormalized perplexity of the LM on the development sentences is 480. The hit ratio of trigrams, bigrams and unigrams is 34.2%, 44.0% and 21.8%, respectively.

### 3.4. Baseline performance

The decoding was performed using the open source Sphinx 3.5 toolkit. For each utterance, 1000 sentence hypotheses were generated, except for a few utterances, for which the beam width suppressed the number of available sentence candidates.

Error rates including substitution, deletion and insertion were measured. The language model uses word particles as basic units, but the recognized particles are merged back to words before comparing them with reference transcripts. See table 1 for the word error rate and the oracle word error rate of the N-best hypotheses of the baseline system.

As can be seen, the N-best error rate is much lower from the 1-best error rate. Thus, it can be hoped that the rescoring the hypotheses with a good LM can improve the error rate.

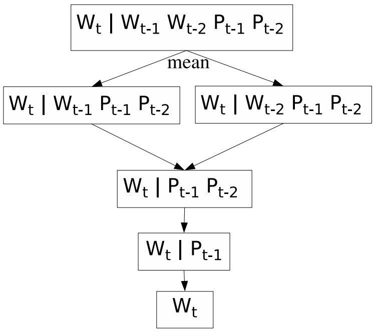
$W_t$ | $W_{t-1}$ $W_{t-2}$ $P_{t-1}$ $P_{t-2}$
mean
$W_t$ | $W_{t-1}$ $P_{t-1}$ $P_{t-2}$
$W_t$ | $W_{t-2}$ $P_{t-1}$ $P_{t-2}$
$W_t$ | $P_{t-1}$ $P_{t-2}$
$W_t$ | $P_{t-1}$
$W_t$

**Fig. 2**. Back-off graph of the rescoring language model. $W_x$ stands for word at time $x$ and $P_x$ for part-of-speech tag at time $x$

## 4. RESCORING EXPERIMENTS

Before rescoring, all N-best sentence hypotheses were processed by a morphological analyzer [11], that tagged all words with their respective POS and stem tags. As many words can have different meanings, the tool also performs statistical disambiguation based on the sentence where the word occur.

The score combination weights of the original particle-based LM, the acoustic model, and the dynamically created morphosyntactic LM were optimized on the development set, using the word error rate of the word-level posterior probability maximizer as the minimization function. The simplex-based ”Amoeba” search strategy implemented in SRILM was applied. Finally, the scores from the different knowledge sources were combined and the final sentence hypothesis was selected using the SRILM-implementation of the ROVER algorithm. The scores of the test set were combined using the optimized weights from the development set.

The stochastic language model that gave the best results in our experiments is outlined on Figure 2. At first, the probability for each word in a sentence is attempted to be calculated based on the history of two previous words and their respective POS tags ($Pr(w_t|w_{t-1}, w_{t-2}, p_{t-1}, p_{t-2})$). As the POS tag is a deterministic function of only the word in most cases (except for ambiguous words), this probability can be viewed as a standard trigram estimation ($Pr(w_t|w_{t-1}, w_{t-2})$). If the string $w_{t-2}$ $w_{t-1}$ $w_t$ did not occur in the training data at least 2 times, the model branches into 2 back-off paths by dropping the parent $w_{t-1}$ or $w_{t-2}$, respectively, and using the mean score from the 2 branches as the final probability. In each of the branches, the word probability is attempted to be computed based on either the previous or the before-previous word, and the 2 previous POS tags. If there is again not enough evidence in the training corpus for those estimates, both branches back off to use only the two, or at least only one previous POS tag for calculating the probability. Finally, the model backs off to the unigram probability of the word.

Table 1 gives the word error rates before and after the rescoring together with the relative improvements. The score combination weights are optimized on the development data, thus the improvement on the test data is lower than on development data, but yet a significant 7.3%.

The described factored language model model uses only the occurrences of words and their respective POS tags for probability estimation. The morphoanalytical tool also attaches word stems to each word but using the stem statistics didn’t give any improvements

| | Dev set | Test set |
|---|---|---|
| Baseline | 42.8 | 45.2 |
| N-best oracle WER | 22.8 | 24.9 |
| After rescoring | 37.2 <13.1%> | 41.9 <7.3%> |

**Table 1**. Word error rates and relative improvements before and after rescoring using the dynamically generated FLM

in our experiments. Quite contrary, it had a negative effect on the rescoring results.

The natural discounting law was used when estimating the probabilities of all back-off nodes except for the final unigram node. The unigram node was discounted using constant discounting with an empirically tuned discounting constant. The rationale behind this is that a significant part of the words in the N-best hypotheses are never seen in the training data. Such words include both very rare words in rare inflections, but more frequently words that are result of lexically illegal concatenation of compound-stem-suffix particles. Also, sometimes the concatenated words are lexically legal, but don't make any sense semantically. The probability calculation of such words always ends up in the unigram node of the factored language model (as they are never seen in the training corpus), where they are given a very small probability.

Table 2 lists hit counts for different back-off nodes in the rescoring factored LM described on Figure 2. The hit counts were computed by rescoring the N-best sentences in the test set and looking were the probability calculation for each word ends up. Note that all nodes except the first get an artificially higher hit count because the backoff path branches into 2 paths and always 2 nodes are hit if the highest node is backed off from. Therefore, it's fair to normalize their hit counts by dividing them by two.

| Node | Hits | Normalized hits |
|---|---|---|
| $w_t \| w_{t-1}, w_{t-2}, p_{t-1}, p_{t-2}$ | 486 | 486 |
| $w_t \| w_{t-1}, p_{t-1}, p_{t-2}$ | 312 | 166 |
| $w_t \| w_{t-2}, p_{t-1}, p_{t-2}$ | 350 | 175 |
| $w_t \| p_{t-1}, p_{t-2}$ | 1000 | 500 |
| $w_t \| p_{t-1}$ | 642 | 321 |
| $w_t$ | 744 | 372 |

**Table 2**. *Hit counts (in thousands) for different backoff nodes in the rescoring FLM that uses a history of 2 previous words ($w_{t-n}$) and POS-tags ($p_{t-n}$), as illustrated on Figure 2*

## 5. DISCUSSION AND SUMMARY

This paper has investigated a 2-pass recognition strategy for agglutinative languages. Word particles are used as language modeling units in the first pass. Words and their POS-tags are used via a factored language model in the second pass to rescore the N-best hypotheses from the first pass. A relative word error rate improvement of 7.3% was obtained on a large vocabulary speaker independent recognition task.

The described recognition method uses a word and POS based FLM to rescore the sentences from the first pass. Instead of an FLM, or in addition to it, other kinds of sentence probability estimators could be used in the second pass. One approach worth investigating is the use of latent semantic analysis (LSA) based LM. The use of word particles as modeling units makes it very difficult to integrate LSA-based LM into the first pass decoder. Also, huge number of different words in the language make the standard LSA approach probably quite ineffective. However, in the 2nd pass, we could use word stems as LSA modeling units. The number of different stems is much less than the number of different inflected word forms, and given the long-distance nature of the the LSA technique, they are more suitable modeling units than the actual words.

The main weakness of the method is that it is not suitable for integration into the 1st pass, as it relies on a limited vocabulary from the N-best results from the 1st pass. Also, the morphological analyzer needs full sentences in order to disambiguate some words.

The experiments presented here were applied on an Estonian speech recognition task. Similar strategy could be used for other agglutinative languages. However, a language specific morphological analyzer is needed for attaching POS-tags to words.

## 6. REFERENCES

[1] O.-W. Kwon and J. Park, “Korean large vocabulary continuous speech recognition with morpheme-based recognition units,” *Speech Communication*, vol. 39, pp. 287–300, 2003.

[2] Vesa Siivola, Teemu Hirsimäki, Mathias Creutz, and Mikko Kurimo, “Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner,” in *Proceedings of Eurospeech*, Geneva, Switzerland, 2003, pp. 2293–2296.

[3] Tanel Alumäe, “Large vocabulary continuous speech recognition for Estonian using morpheme classes,” in *Proceedings of ICSLP 2004 - Interspeech*, Jeju, Korea, 2004, pp. 389–392.

[4] Mirjam Sepesy Maucec, Zdravko Kacic, and Bogomir Horvat, “Modelling highly inflected languages.,” *Information Sciences*, vol. 166, no. 1-4, pp. 249–269, 2004.

[5] P. Geutner, M. Finke, and P. Scheytt, “Adaptive vocabularies for transcribing multilingual broadcast news,” in *Proceedings of ICASSP 1998*, Seattle, Washington, 1998.

[6] A. Stolcke, E. Shriberg, R. Bates, M. Ostendorf, D. Hakkani, M. Plauche, G. Tur, and Y. Lu, “Automatic detection of sentence boundaries and disfluencies based on recognized words,” in *Proceedings of ICSLP*, Sydney, Australia, 1998, vol. 5, pp. 2247–2250.

[7] J. Bilmes and K. Kirchhoff, “Factored language models and generalized parallel backoff,” in *Proceedings of HLT/NACCL*, 2003, pp. 4–6.

[8] Einar Meister, Jürgen Lasn, and Lya Meister, “Development of the Estonian SpeechDat-like database,” in *Proceedings of Eurospeech*, Geneva, Switzerland, 2003, vol. 2, pp. 1601–1604.

[9] Heiki-Jaan Kaalep and Kadri Muischnek, “The corpora of Estonian at the University of Tartu: the current situation,” in *The Second Baltic Conference on Human Language Technologies : Proceedings*, Tallinn, Estonia, 2005, pp. 267–272.

[10] Andreas Stolcke, “SRILM – an extensible language modeling toolkit,” in *Proceedings of ICSLP*, Denver, USA, 2002, vol. 2, pp. 901–904.

[11] Heiki-Jaan Kaalep and Tarmo Vaino, “Complete morphological analysis in the linguist's toolbox,” in *Congressus Nonus Internationalis Fenno-Ugristarum Pars V*, Tartu, Estonia, 2001, pp. 9–16.