Weighted Finite-State Transducer Approach to German Compound Words Reconstruction for Speech Recognition

Nickolay Shamraev, Alexander Batalshchikov, Mikhail Zulkarneev, Sergey Repalov, Anna Shirokova
Stel – Computer Systems Ltd.
Moscow, Russian Federation
ncam1977@yahoo.com, abatalshikov@gmail.com, zulkarneev@mail.ru, repalov_sa@stel.ru, anna_a@stel.ru

Abstract—**An approach is proposed for German Large Vocabulary Speech Recognition, dealing with the problem of compound words, based on unsupervised word decomposition for German words and a probabilistic method for combining the words using finite state transducers. The basic idea of the method is to train n-gram language model on the texts where compound words are substituted by their parts plus concatenation symbol. Thus, the context information is taken into account for the compound words and is used in the process of recombination to find most probable variant for recognition result. The advantage of this approach is the improvement of the word recognition accuracy and a more precise recombination of compound words.**

I. Introduction

One of the important problems in German Large Vocabulary Continuous Speech Recognition (LVCSR) system is the large amount of compound words in the language. Although their relative frequency is low, compound words make up a significant part of German vocabulary. This leads to unproportionally large size of LVCSR lexicon and high out-of-vocabulary (OOV) rates compared to other European languages.

State of the art approaches dealing with the compound word problem are based on decomposition of compounds into smaller parts. Decomposition helps to increase lexical coverage, lower OOV rate and leads to decrease of Word Error Rate (WER) on shorter words. Most approaches can be classified as using supervised [1], [4], [11], [12] or unsupervised word partition [2], [7], [8], [9], [13]. Some of the efficient methods include using fragment based language models with tagged fragments as a part of language model [7], [8], [11], [12], sub-lexical language model (LM) approach with graphones [5], [6], and methods which imply splitting compound words into smaller elements [3], [7], [10].

An approach, which allows using the complete vocabulary of the language model training data as a valuable knowledge resource, was presented in article [2]. In [2] the list of compounds which are recombined in lattices is generated manually, but the results show an increase in word error rate (WER).

This idea was used and developed in work [14]. In [14], the recombination process is restricted to the recognition resultonly. Thus, a lattice is generated by recombining compounds from the recognition result using a large vocabulary. Then the lattice is rescored with a unigram language model and the most likely compound sentence in the lattice is chosen as the recognition result.

In this paper we propose a method which allows using contextual information for compound words in recognition lattice. The practical implementation of the method is based on the weighted finite-state transducers (WFST) as a convenient tool for hypothesis presentation, integrated in Kaldi toolkit [15].

The weighted finite-state transducers are now widely used in application to different speech recognition problems ([21], [22], [23]). The proposed method of compound recombination is related to the method applied in the Estonian LVCSR system [22].

The paper is organized as follows: in Section 2 we provide a description of a modified recognition lattice, details of text pre-processing, an algorithm forword partition, and a method of reconstruction of compound words with the help of WFST. Section 3 contains the experimental setup and results. In Section 4 we discuss results of experiments.

II. Description of The Method

The basic idea of the proposed approach is close to the method of recombination of compound words from the recognition result, based on the use of lattices and a unigram language model [14].

In [14] a set of possible options for combining compound words H is presented in the form of a lattice and the result of recombination is the most probable path $\hat{h}$ in the lattice, which is calculated using a unigram language model:

$$\hat{h} = \arg\max_{h \in H} p_{1-gram}(h)$$

In the method proposed in this paper, n-gram language model (n=4) is used to estimate the probability of potential variants of compound words recombination, which makes it possible to take into account the context of words. By using contextual information this method can significantly improve

ISBN 978-952-68397-0-7 (paperback), ISBN 978-952-68397-1-4 (PDF)

the accuracy of speech recognition. Set of possible options for combining compound words H is presented as a lattice of a special type, an example is shown in Fig. 1.

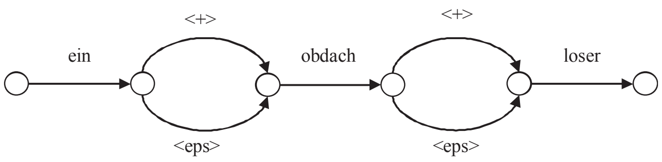

Fig. 1. Example of the lattice, representing possible options of word recombination for recognition result "ein obdach loser" ("a homeless")

The use of n-gram language model gives an opportunity to select the most probable variant of the recombination. Then, further post-processing combines words, which have a special symbol of concatenation "<+>" in between. To use n-gram model the initial lattice and the language model should be presented in the form of two WFST, H and G respectively. Then the procedure of appending probabilities to the lattice arcs is reduced to the composition operation between the WFSTs:

$$HG = H^{\circ}G$$

and the selection of the most probable variant of recombination is reduced to the problem of finding the shortest way in WFST HG.

In other words, the reconstruction of compounds in the proposed method is based on a 4-gram language model trained on the data with a special concatenation symbol <+> and the use of the weighted finite-state transducers to find the optimal way of recombination.

The proposed method can be divided into the following stages:

- pre-processing of the text,
- selection of a compound words vocabulary (lexicon) and partition of the text data,
- construction of the hybrid Hidden Markov Model – Deep Neural Network (HMM – DNN) speech recognition system,
- to obtain hypotheses on the short words lexicon using speech recognition system output,
- training an n-gram language model on the texts with short words containing the symbol of word concatenation,
- construction of a finite state transducer (FST) performing the re-composition of compound words,
- conversion of short word hypotheses into full word hypotheses using the FST (word reconstruction).

The pre-processing of the text is necessary for correct operation on the next steps. It converts all of the available German text materials into homogeneous data.

Selecting a dictionary (lexicon) of partition words defines the set of admissible words – fragments of compound words. In this paper we propose to use a dictionary of admissible partitions which contains partition for each word into two shorter sub-words, with some restrictions. In the general case, the algorithm of word decomposition and the contents of the dictionary can be selected using other methods, algorithms described further in the article are not dependent on the algorithm of word decomposition. When the dictionary is constructed all the input texts are transformed into texts with separated words.

After that we build speech recognition system on the basis of Kaldi toolkit [15], using a hybrid HMM-DNN approach, and perform speech recognition over data with separated words.

As the output of the speech recognition system, we obtain recognition hypotheses for all possible values of the language model weight. On the next step the resulting text hypotheses are converted into hypotheses with full words using a probabilistic word-merging model based on the WFST. This probabilistic model is trained on a corpus of texts in which compound words are explicitly replaced by the parts of compound words and the concatenation symbol.

A. *Text Pre-processing*

At the stage of pre-processing the training text is normalized and converted into a standard format. All words containing characters that are not included in the German alphabetare transformed into <unk>. The words, which had various writing variants before German writing reform in 1996, are converted into the correct form using a special dictionary.

We consider three major cases of substitutions:

ß/ss: According to the new spelling rules, *ß*, when preceded by a short vowel sound, should be written with a double s. We applied this rule, changing all words in training and text bases into the correct variant according to the orthography reform (*Rechtschreibreform*).

Umlauts *ue/ü, ae/ä* and *oe/ö*: It is common to write ue, ae or oe instead of ü, ä and ö as different variants in case of non-German keyboards because it may be difficult to print Umlauts. To remove different variations of the same word we applied substitution selecting the variant with the highest frequency.

Triple Consonants: triple consonants, when followed by a vowel, were reduced to two. In the new spelling, triple consonants are preserved no matter the following letter. Therefore, we normalize spelling variants differing only on the number of consonants according to the new rule.

In addition, the transcription of numerals, dates and currency symbols with numerals is performed. Pronunciation rules for numbers and numerals are not followed strictly; in fact, complex numerals have several variants of pronunciation in processed speech data. Therefore, sentences in all training data containing numerals are considered separately and processed by the program converting numerical expressions into the text form. To complete this, numerical expressions were split into the minimal allowable elements, i.e. individual numbers for each grade. The numbers from 1 to 100 were taken as the minimal elements (e.g., "*Zweihundertsechsundvierzig*" ("two hundred forty-six") separated as "*zwei hundert sechsundvierzig*"). Combining numerals in compound numerals is executed in the same way as for the rest of the words, and based on a probabilistic model.

In case when words with numbers are compounded with a hyphen sign, for example *20-Tonner*, *35-mal* ("20-tonner", "35 times"), we replace the hyphens in such combinations with a white space to separate the word and the number. For some German compound words, comprising initial or trailing hyphen, for example *Getrennt- und Zusammenschreibung* ("separate and fused writing"), the hyphens were deleted as well.

B. Word partition

Partition of compound words into simple words is performed by executing the following steps (flowchart is presented in Fig. 2):

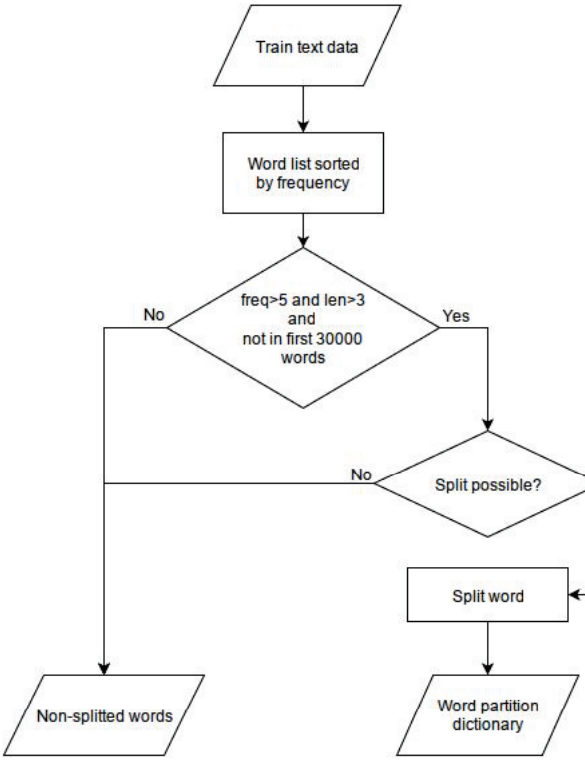
Train text data
Word list sorted by frequency
freq>5 and len>3 and not in first 30000 words
No
Yes
Split possible?
No
Yes
Split word
Non-splitted words
Word partition dictionary

Fig. 2. Word partition algorithm.

- Extraction of words from the available training text and sorting them by frequency of occurrence.
- Selection of word-candidates for the elements of the partition. Words are chosen as candidates if they consist of at least 4 characters and have the number of occurrences in the text not less than 5.
- The most frequent 30000 words are not split.Since these words are most often used, their partition could worsen n-gram language model. All of them remain in the recognition vocabulary.
- For each of the remaining words of the general list we check the possibility of separation into candidate words, already contained in the lexicon.
- If the partition is possible, we split the compound word intothe most probable pair (probability is determined by the product of the frequency of occurrence of parts). Then the partition of the compound word is added to the partitions dictionary.
- If no possible partition is found for the current word, it remains in the recognition vocabulary.

The selection of parameters in words partition was carried out on the basis of the guidelines given in previous researches, and tested empirically with a series of preliminary experiments. Thus, the length of words and number of occurrences (N=5) were considered taking into account the results of [1, 2, 16]. The number *r* (r=30000) of the discarded most frequent words was chosen according to the results in [14].

The result is a lexicon of the words and their corresponding partitions. Then we convert all text materials applying the operation of decomposition to each compound word in the text. The operation of decomposition is repeated several times (four iterations is usually sufficient), until the lexicon vocabulary extracted from the converted text does not change (that means, all compound words are separated into constituent elements).

The total volume of texts before partition was 48.942 million words, with the vocabulary containing about 2.85 million words. After splitting the text volume grew up to 51.16 million words and the size of the vocabulary reduced to 744,000 words.

As the text data for training language models we used articles from Wikipedia (370 MB), texts from the web collected news resources (Euronews, 55 MB) and 45 books of various genres (20 MB) in addition to the training data.

C. Baseline speech recognition system

The experiments were performed using a speech recognition system based on Kaldi [15]. Recognition was carried out in two stages, and it can be described as a hybrid HMM-DNN approach.

At the first stage, a recognition system based on HMM-GMM was used to build the adaptation of Maximum Likelihood Linear Regression (MLLR) matrix, then the MLLR transformation was applied to Perceptual Linear Prediction (PLP) feature vectors plus first and second derivatives (39-dimensional vector). At the second stage, recognition of MLLR-transformed features was performed using the DNN, consisting of 4 hidden layers, each hidden layer composed of 1024 elements. Unsupervised pre-training and back propagation training based on statistical gradient descent algorithm with cross entropy as target function are used for training the DNN. In general, our DNN training algorithm follows a given technique, namely Karel Vesely's DNN implementation [15].

As a language model, 3-gram model was used in speech recognition system, trained on the text data with full or split words depending on conditions of the experiment.

The training data (31.4 hours) is based on two sources: Globalphone German speech data [17] and German Voxforge open speech corpus.

D. *Reconstructing compound words using Weighted Finite State Transducers*

Simple methods for words reconstruction such as dictionary of reverse replacement, dictionary using only compound words with uniquely defined parts (i.e. compound word always splits into them), and recombination with probability threshold do not produce the desired gain in accuracy when compared with recognition accuracy on short words.

In this regard, we propose a method of word reconstruction similar to the method described in the work [14] but based on the application of the weighted finite state transducers (WFST).

Reconstruction of full words is based on the use of a 4-gram language model, presented in the form of WFST and trained on the text data, in which the places of separation are filled with the concatenation symbol "<+>". An example of the original text before separation is provided below:

> "*die deutsche kriegsfotografin anja niedringhaus ist nach polizeiangaben in ostafghanistan von einem polizisten erschossen worden ein polizeisprecher sagte die kanadische journalistin kathy gannon sei bei dem vorfall schwer verletzt worden*".

The transformed text:

> "*die deutsche kriegs <+> fotografin anja niedringhaus ist nach polizei <+> angaben in ostafghanistan von einem polizisten erschossen worden ein polizei <+> sprecher sagte die kanadische journalistin kathy gannon sei bei dem vorfall schwer verletzt worden*"
>
> (translation from German: "the German war photographer Anja Niedringhaus was shot dead by a policeman according to the police in eastern Afghanistan, a police spokesman said the Canadian journalist Kathy Gannon had been seriously injured in the incident")

A 4-gram language model is trained on the transformed text with the help of SRILM utilities [18].

In general, the algorithm of fullword reconstruction consists of the following steps:

1) Training of an n-gram language model on the text data with sub-words separated by the concatenation symbol ("<+>") instead of compound words (trigram or 4-gram language models were trained).

2) Generation of the WFST corresponding to this language model. The *"arpa2fst"* command executes this step in Kaldi.

Steps 2 and 3 are executed once for a fixed language model, and then we iterate steps 4-5 for each sentence obtained as ASR output for shorter words to get a full-text recognition results.

3) Generation of the small linear WFST for each sentence - recognition hypothesis. The sentence-level WFST contains additional transitions: <eps> and <+>. The transition <eps> corresponds to a space between normal words, the transition <+> corresponds to the concatenation of fragments of compound word. More detailed description on creation of the WFST for the sentence and language model can be found in OpenFST library tutorial [20] and [19].

Example of the recognition hypothesis sentence:

"<s> die sind die größten busch <+> feuerin der provinz new south wales </s>"

(translation: "which are the largest bush fire in the province of New South Wales")

We generate the corresponding ".txtfst" file:

```
0 1 <s> <s>
1 2 die die
2 3 <eps> <eps>
2 3 <+> <+>
3 4 sind sind
4 5 <eps> <eps>
4 5 <+> <+>
5 6 die die
6 7 <eps> <eps>
6 7 <+> <+>
7 8 größten größten
8 9 <eps> <eps>
8 9 <+> <+>
9 10 busch busch
10 11 <eps> <eps>
10 11 <+> <+>
11 12 feuer feuer
12 13 <+> <+>
13 14 in in
14 15 <eps> <eps>
14 15 <+> <+>
15 16 der der
16 17 <eps> <eps>
16 17 <+> <+>
17 18 provinz provinz
18 19 <eps> <eps>
```

```
18 19 <+> <+>
19 20 new new
20 21 <eps> <eps>
20 21 <+> <+>
21 22 south south
22 23 <eps> <eps>
22 23 <+> <+>
23 24 wales wales
24 25 </s> </s>
```

Then we apply *"fstcompile"* command from OpenFST library and get small FST, corresponding to this hypothesis.

4) The composition of the two WFSTs obtained in steps 2 and 3, results in a new WFST, which has a property that shortest way in it contains the most probable variant (with the possible insertions of "<+>") in terms of the n-gram model obtained in step 1. Thus, we compose two FSTs and calculate the shortest path in theresulting FST at this stage.

We use *"fstcompose"* and *"fstshortestpath"* commands from OpenFST for this step.

5) Finally, an extraction of the text with compound words from the shortest path in the FST from stage 4 is made.

We use Kaldi procedure "fst-to-transcripts" to perform this step.

III. Experiments and Results

Four types of speech recognition experiments on the test data were conducted. In the first experiment a model trained on full word text data without any partition of compound words (Baseline model in the Table I) was applied.

The second experiment is based on the recognition with the model trained on short words – fragments of compound words. Both training and test data are transformed so that all composite words are split (SHORT_WORD model in the Table I). Language model parameters of both baseline and SHORT_WORD were pruned to generate WFSTs of the same size (after pruning Baseline model contained approximately1.4 million 1-grams, 1.8 million 2-grams and 750,000 3-grams, while SHORT_WORD contained 744,000 1-grams, 2.47 million 2-grams and 0.98 million 3-grams).

The next three experiments are set up as the full words recognition based on the model trained on "short" words and the reconstruction of compound words using statistics of sequences. The reconstruction is performed by comparing the sequences of words in hypotheses with the dictionary of word partitions. Two variants of the dictionary of split words were used. The first variant of the dictionary consists of unique splits (i.e. compound word always splits into the two parts). This leads to a smaller dictionary of partitions, but makes recombination straightforward (UNIQUE_SPLIT model in Table I). The dictionary in the fourth experiment includes the compound word partition only if it is a more probable variant (PROB model in Table I).

The fifth experiment is the full-words recognition based on the model trained on "short" words followed by reconstruction of the composite words with a model based on proposed method of recombination (WFST model in Table I).

As test data, two different speech bases were used. The first test base consists of web collected audio transcriptions (Euronews) and comprises about 74.9 minutes speech. Second speech base includes default test samples from Globalphone German ([17], 77.3 minutes).

The experimental results (WER) are presented in the following table.

TABLE I. Word error rate of recognition using different methods

Test data	Baseline	SHORT_ WORD	UNIQUE _SPLIT	PROB	WFST
Broadcast News	23.66	19.83	22.52	20.37	20.09
Global Phone	15.17	9.89	-	10.54	10.17

The Baseline model in Table I shows the recognition results for the case when compound words were not split, i.e. recognition was carried out on full words. It is shown in many works, that recognition on components of compounds is more accurate and robust. Thus, SHORT_WORD model in the second column shows much better results compared to thebaseline model. However, these are not valid recognition results, because short "words" in fact do not correspond to real spoken words of German language.

Then, the next three columns in Table I represent different methods of word reconstruction for the base language model, and these are valid recognition results. In the case of exact word recombination for compounds, the accuracy of recognition (and WER) would converge to that of SHORT_WORD model. The WFST-based method of recombination gives better results than PROB model, which can be considered as a unigram-based method of recombination.

IV. Conclusion

In this article, we present a detailed description of two methods: decomposition of German compound words and their reconstruction. Decomposition is beneficial to increase lexical coverage, lower OOV rate, and it also gives better results at the recognition stage on shorter words. If it is not applied, the size of an n-gram language model and the number of OOV words become the major limiting factors for the German LVCSR system development.

The method of compound word reconstruction is based on the idea of training an n-gram language model on the texts in which compound words are substituted with their parts plus a concatenation symbol. The method uses the composition of the WFSTs and is directly applied to the output of a recognition system with a short-word vocabulary.

Experimental results show that the proposed method significantly improves the recognition accuracy compared to methods based on the use of non-modified composite words. In this case, the absolute reduction in WER is 3.4%.

Given the fact that the recognition accuracy for reconstructed fullwords is almost equal to the accuracy of the model on the reduced (short-word) vocabulary (the difference is about 0.2%) it can be concluded that the proposed method of recovery is close to optimal. The remaining errors can be attributed to the ambiguity of writing for some German words (i.e. both options of word writing - separate and joint, are allowed), and at last, to the inaccuracy of the probabilistic model.

The method of the reconstruction of compound words described in the article can be used to restore other special characters in other languages (for example, an apostrophe in Turkish or a hyphen in other languages).

ACKNOWLEDGMENT

Research is being conducted with the support of Ministry of Education and Science of Russian Federation (Contract №14.579.21.0058). Unique ID for Applied Scientific Research (project) RFMEFI57914X0058. The data presented, the statements made, and the views expressed are solely the responsibility of the authors.

REFERENCES

[1] M. Adda-Decker, and G. Adda, “Morphological decomposition for ASR in German”, *Workshop on Phonetics and Phonology in ASR*, Saarbrucken, Germany, 2000, pp. 129-143.

[2] M.A. Adda-Decker, “A corpus-based decompounding algorithm for German lexical modeling in LVCSR”, *in Proc. European Conf. on Speech Communication and Technology*, Geneva, Switzerland, 2003, pp. 257-260.

[3] I. Bazzi, J.R.Glass, “Modeling out-of-vocabulary words for robust speech recognition”, *in Proc. Int. Conf. on Spoken Language Processing*, Beijing, China, Oct. 2000.

[4] A. Berton, P. Fetter, P. Regal-Brietzmann, “Compound words in large-vocabulary German speech recognition systems”, *in Proc. Of International Conference on Spoken Language Processing*, Philadelphia, PA, USA, Oct. 1996, vol. 2, pp. 1165 – 1168.

[5] M. Bisani, H. Ney, “Open vocabulary speech recognition with flat hybrid models”, *in Proc. of Interspeech*, Lisbon, Portugal, Sept. 2005, pp. 725-728.

[6] M. Bisani, H. Ney, “Joint-sequence models for grapheme-to-phoneme conversion”, *Speech Communication*, vol. 50, no. 5, May 2008, pp. 434 – 451.

[7] M. Creutz, T. Hirsimki, M. Kurimo, A. Puurula, J. Pylkknen, V. Siivola, M. Varjokallio, E. Arisoy, M. Saraclar, A. Stolcke, “Morph-based speech recognition and modeling of out-of-vocabulary words across languages”, *ACM Transactions on Speech and Language Processing*, vol. 5, no. 1, Dec. 2007.

[8] A. El-Desoky Mousa, H. Ney, “Sub-Lexical Language Models for German LVCSR”, *in Proc. Of IEEE workshop on Spoken Language Translation*, Dec. 2010, pp. 803-806.

[9] L. Galescu, ”Recognition of out-of-vocabulary words with sub-lexical language models”, *In Proc. of European Conference on Speech Communication and Technology*, Geneva, Switzerland, Sept. 2003, pp. 434-451.

[10] D. Klakow, G. Ros, X. Aubert, “OOV-detection in large vocabulary system using automatically defined word-fragments as fillers”, *In Proc. of European Conference on Speech Communication and Technology*, Budapest, Hungary, Sept. 1999, vol. 1, pp. 49-52.

[11] J. Kneissler, D. Klakow, “Speech recognition for huge vocabularies by using optimized sub-word units”, *In Proc. of European Conference on Speech Communication and Technology*, Aalborg, Denmark, Sept. 2001, vol 1, pp 69-72.

[12] L. Lamel, A. Messoudi, J.L. Gauvain, “Investigating morphological decomposition for transcription of Arabic broadcast news and broadcast conversation data”, *In Proc of Interspeech,* Brisbane, Australia, Sept. 2008, vol 1, pp 1429-1432.

[13] R. Ordelman, A.V. Hassen, F.D. Jong, “Compound decomposition in Dutch large vocabulary speech recognition”, *In Proc. of European Conference on Speech Communication and Technology*, Geneva, Switzerland, Sept. 2003, pp. 225-228.

[14] M. Nußbaum-Thom, A. El-Desoky Mousa, R. Schluter, H. Ney, “Compound word recombination for German LVCSR”, *In Proc. of Interspeech*, Florence, Italy, 2011, pp. 1449–1452.

[15] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, K. Vesely, ”The Kaldi Speech Recognition Toolkit”, *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, IEEE Signal Processing Society, 2011.

[16] R. Hecht, J. Riedler, G. Backfried, “Fitting German into N-Gram Language Models”, *Lecture Notes in Computer Science*, Volume 2448, 2002, pp. 341-346.

[17] T. Schultz, “Globalphone: A Multilingual speech and text database developed at Karlsruhe University”, *In Proceedings of the International Conference of Spoken Language Processing*, ICSLP 2002, Denver.

[18] A. Stolcke, “SRILM - an extensible language modeling toolkit”, *In Proc. of International Conference on Spoken Language Processing*, Denver, Colorado, USA, Sept. 2002, vol. 2, pp. 901 – 904.

[19] M. Mehryar, C.N.P. Fernando, R. Michael,“Weighted Finite-State Transducers in Speech Recognition”, *Computer Speech and Language*, 16(1), pp. 69-88, 2002.

[20] C. Allauzen et al., “OpenFST: A General and Efficient Weighted Finite-State Transducer Library” *In Proc. CIAA*, Prague, Czech Republic, July 16–18, 2007, pp. 11–23.

[21] I. Chen et al., “A Keyword-Aware Language Modeling Approach to Spoken Keyword Search”, *Journal of Signal Processing Systems*, 01/2015, pp 1-10.

[22] T. Alumae, “Recent improvements in Estonian LVCSR”, *In SLTU 2014* , Saint Petersburg, Russia, 2014.

[23] B. Réveil, K. Demuynck , J. Martens, “An improved two-stage mixed language model approach for handling out-of-vocabulary words in large vocabulary continuous speech recognition”, *Computer Speech & Language*, vol. 28, no. 1, Jan. 2014, pp. 141–162.