INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

The STC Keyword Search System For OpenKWS 2016 Evaluation

Yuri Khokhlov[1], *Ivan Medennikov*[1,2], *Aleksei Romanenko*[1,2], *Valentin Mendelev*[1,2]
Maxim Korenevsky[1,2], *Alexey Prudnikov*[1†], *Natalia Tomashenko*[1,2‡], *Alexander Zatvornitsky*[1,2]

[1] STC-innovations Ltd, St.Petersburg, Russia
[2] ITMO University, St.Petersburg, Russia

{khokhlov,mendelev,medennikov,romanenko,
korenevsky,tomashenko-n,zatvornitskiy}@speechpro.com
alexey.prudnikov@corp.mail.ru

Abstract

This paper describes the keyword search system developed by the STC team in the framework of OpenKWS 2016 evaluation. The acoustic modeling techniques included i-vectors based speaker adaptation, multilingual speaker-dependent bottleneck features, and a combination of feedforward and recurrent neural networks. To improve the language model, we augmented the training data provided by the organizers with texts generated by the character-level recurrent neural networks trained on different data sets. This led to substantial reductions in the out-of-vocabulary (OOV) and word error rates. The OOV search problem was solved with the help of a novel approach based on lattice generated phone posteriors and a highly optimized decoder. This approach outperformed familiar OOV search implementations in terms of speed and demonstrated comparable or better search quality.

The system was among the top three systems in the evaluation.

Index Terms: keyword search, speech recognition, low-resource, OpenKWS 2016, OOV search

1. Introduction

The problem of speech recognition for low-resource languages has been receiving a lot of attention in recent years. This interest was greatly facilitated by IARPA Babel program which "is developing agile and robust speech recognition technology that can be rapidly applied to any human language in order to provide effective search capability for analysts to efficiently process massive amounts of real-world recorded speech" [1]. As part of the program National Institute of Standards and Technology (NIST) organized annual OpenKWS evaluations from 2013 till 2016. The evaluation campaigns were accessible to all speech recognition community members. In the beginning of each evaluation NIST distributed a limited amount of language resources among participants to prepare their technology. In the final stage of each evaluation the participants were requested to build a speech recognition system for a new (surprise) language in several weeks.

In the OpenKWS 2016 the surprise language was Georgian and the participants were provided with the training data without the phonetic lexicon. Each team had to build a system, process the 75 hrs long evaluation set and submit keyword search (KWS) and speech recognition results to NIST.

In this paper we describe the STC system that took part in the evaluation. The main highlights are the acoustic part of the system which consists of 9 acoustic models of different architectures, character level recurrent neural network (*char-rnn*) [2] application to improve the language modeling part and a novel approach to out-of-vocabulary (OOV) words detection problem. All results in the paper are reported on Georgian language. We use actual/maximum term weighted value ([A/M]TWV) and word error rate (WER) to measure keyword search and speech recognition quality respectively. Details on the metrics can be found in [3].

The paper is organized as follows. Section 2 describes some previous works devoted to the low-resource speech recognition and keyword spotting. Sections 3 and 4 are about the acoustic and language modeling components of the system respectively. Keyword search and OOV handling techniques are covered in Section 5. Section 6 describes the techniques used to combine the results from the separate systems. The discussion and conclusions are in Section 7.

2. Related work

Previous work on speech recognition for low-resource languages has shown that the combination of different systems [4] and multilingual bottleneck features [5, 6, 7] among other techniques provide notable performance gains for keyword search. In [8, 9, 10] very low WER values were obtained on Russian and English conversational telephone speech recognition tasks using speaker dependent bottleneck features, deep neural networks (DNN) and deep maxout networks (DMN) trained using annealed dropout regularization [11]. We decided to adapt recipes developed in [8, 9, 10, 12] for multilingual training and apply them to the OpenKWS 2016 task.

In [13] word-based recurrent neural network (RNN) was used to generate more text in order to improve a language model employed on the lattice generation phase. We had decided to use character level neural network [2] for the same purpose and obtained very good results with this approach.

The possible methods to perform keyword search are described in [14], [15] and [16]. We report results on our implementation of [15] with some modifications. We also use word and subword units on the lattice generation phase. Some previous work on subword units and their benefits for OOV search can be found in [17], [18] and [19]. The new approach to OOV words detection outlined in Section 5.2 is based on features representing lattice posterior probabilities explained in [20], [21], and [22].

[†]Alexey Prudnikov is now with Mail.Ru Group, St.Petersburg, Russia

[‡]Natalia Tomashenko is now also with LIUM, University of Le Mans, France

http://dx.doi.org/10.21437/Interspeech.2017-1212

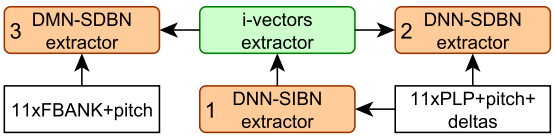

Figure 1: *Bottleneck features and i-vectors extraction scheme*

3. Acoustic Modeling

The in-house version of the Kaldi toolkit [23] that supports integration with our proprietary speech recognition training system was used to train acoustic models.

3.1. Feature extraction

In order to utilize different representations of sound 3 sets of raw features were used in our system, namely 40-dimensional log mel filterbank energy (FBANK) features, 40-dimensional mel frequency cepstral coefficients (MFCC), and 13-dimensional perceptual linear prediction (PLP) features. All raw features were appended with pitch values.

In addition to these raw features, our system utilizes multilingual i-vectors and 3 types of high-level multilingual bottleneck features:

1. Deep Neural Network (DNN) based speaker-independent 80-dimensional bottleneck (DNN-SIBN) features extractor trained using 11xPLP+pitch+deltas features.
2. DNN based speaker-dependent 80-dimensional bottleneck (DNN-SDBN) features extractor trained using 11xPLP+pitch+deltas features appended with i-vector.
3. DMN based speaker-dependent 80-dimensional bottleneck (DMN-SDBN) features extractor trained using 11xFBANK+pitch features appended with i-vector.

Extractor of 200-dimensional i-vectors was based on Universal Background Model (UBM) with 2048 Gaussians. The extraction scheme of bottleneck features and i-vectors is presented in Figure 1. All extractors were trained using build datasets for 18 languages from the IARPA Babel Program language collection (overall duration is 860 hours). DNN and DMN based SDBN extractors were trained in multi-task style with language-specific parts consisting of only softmax layer with about 5000 grapheme senones as outputs. DNN training was performed with greedy layer-wise discriminative pretraining, DMN training was carried out using annealed dropout regularization technique [11] without pretraining. The other details of extractors training procedure are omitted due to the lack of space, and because the procedure is quite similar to the SDBN approach presented in our previous papers [8, 9, 10, 12].

3.2. Acoustic models

Our system comprises 9 acoustic models trained to classify grapheme senones. The models are briefly described below:

1. **DNN**$_1$: 6x1024 sigmoidal DNN; 11x FMLLR-adapted LDA-MLLT transformed PLP+pitch features; sequence training with state-level Minimum Bayes Risk (sMBR) criterion.
2. **DNN**$_2$: 4x2048 sigmoidal DNN; 31x FMLLR-adapted DNN-SDBN taking every 5th frame; sMBR sequence training.
3. **DMN**$_3$: 6x1536 DMN with maxout group size of 2; 31xDMN-SDBN taking every 5th frame; cross-entropy (CE) training with annealed dropout regularization followed by sMBR sequence training (see details in our previous paper [10]).
4. **DMN**$_4$: the same as **DMN**$_3$, but initialized with the shared part of multilingual DMN (18 langs).
5. **TDNN**$_5$: Time Delay Neural Network (TDNN) [24] with 4x1024 ReLU layers; 5xMFCC+pitch appended with i-vector; CE criterion.
6. **BLSTM**$_6$: Bidirectional Long Short-Term Memory recurrent neural network (BLSTM) with projection layers [25]; 3x512(cell,hidden)x128(recurrent proj.,non-recurrent proj.) hidden layers; 5xFBANK+pitch appended with i-vector; CE criterion.
7. **DNN**$_7$: 6x1024 sigmoidal DNN; 11x PLP+pitch appended with i-vector; initialization with the shared part of multilingual DNN (18 langs); CE criterion.
8. **DMN**$_8$: 10x1024 DMN with maxout group size of 2; 11x FBANK+pitch appended with i-vector; initialization with the shared part of multilingual DMN (18 langs); CE training with annealed dropout regularization.
9. **DMN**$_9$: the same as **DMN**$_8$ with semi-supervised learning. 40 hours of untranscribed training data were recognized using the best acoustic model in terms of WER (**DMN**$_3$).

All models except the first one were trained with the use of speed perturbed data [26] (two additional copies of the training data were created by adjusting the speed by +-10% of the original value). Performance of these models on the development set in terms of WER and ATWV on in-vocabulary (IV) words of the official development keywords list is reported in Table 1. It should be noted that these results were obtained using the best language model from section 4.

Table 1: *Performance of acoustic models on the development set. ATWV scores are reported for IV words of the official development keywords list*

Acoustic Model	ATWV	WER,%
DNN$_1$	0.643	44.2
DNN$_2$	0.634	41.5
DMN$_3$	0.675	**39.4**
DMN$_4$	0.660	44.3
TDNN$_5$	0.658	42.3
BLSTM$_6$	0.652	41.1
DNN$_7$	0.669	43.0
DMN$_8$	0.680	42.4
DMN$_9$	**0.685**	41.8

4. Language modeling

The basic dataset used to build the lexicon and the language model consisted of the acoustic training transcriptions. It amounted to about 4.5 Mb of text and the observed OOV rate on the development list was about 20%. In order to handle this problem and hopefully increase the n-gram statistics quality, we decided to augment the training set with artificially generated texts produced by the character-level recurrent neural network

(char-rnn). It was demonstrated [2] that char-rnn language models (LM) can be used to provide artificial texts that look quite naturally with some minor inconsistencies.

To train and apply the model we used the open source *char-rnn* tool [2] slightly patched to support UTF-8 input files. Our model had 2 LSTM layers with 256 neurons each and was trained with dropout rate of 0.3. Gradients were propagated up to 200 steps backward in time. After training the model we generated up to 100 artificial texts 1 million characters each. We found that using lexicons of size 100–150K (only the most frequent words were retained) can reduce the OOV rate on a pre-selected development set more than twice compared to the original lexicon.

The other source of extra data was the Web texts (about 380 Mb) provided by the organizers (BBN part). Overall we had four datasets for LM training: basic dataset (baseline), the Web texts, and two sets of artificial data. The last two sets were generated with two char-rnn models: CRNN_1 and CRNN_2. The acoustic training transcriptions were included in the training sets of both models and the latter also had seen some texts selected by perplexity from the Web texts corpus. The LMs were trained on each dataset separately and then interpolated. The size of the lexicon for all the initial LMs was limited by 150K words.

Table 2 shows OOV rates on the official development keyword list and the values of WER and ATWV obtained with different LMs with the $\textbf{TDNN}_5$ AM. It can be seen that extending set of texts for training LMs reduces the OOV rate and improves both WER and ATWV significantly[1]. Moreover, real and artificial texts seem to be complementary.

Table 2: *Language models comparison on the development set and the official development keywords list*

LM	OOV,%	ATWV(ALL/IV)	WER,%
Baseline	20.0	0.563 / 0.638	46.6
+CRNN_1	9.3	0.614 / 0.653	43.7
+CRNN_2	10.0	0.598 / 0.652	46.8
+Web	8.6	0.624 / **0.658**	43.5
+CRNN_1+Web	4.4	**0.639 / 0.658**	**42.3**

5. Keyword search and OOV Handling

5.1. Keyword search in confusion networks using proxies

In our keyword search implementation we used a word-level confusion network (CN) [14] to index audio-data. The idea is based on the *proxies approach* proposed in [15], where a special WFST is constructed from a CN, and used to search for IV and OOV words.

We applied several modifications to the original algorithm ([15]) in order to speed up the search process and to improve the search quality. First, we construct WFST from CN in such a way that all paths from the start node lead only to non-epsilon arcs, and all paths to the final node pass through the non-epsilon arcs only. This construction allows to skip the last (third) step mentioned in [15], since no overlapping hits occur. Also this speeds up the composition operation and reduces memory consumption. Second, we use the same proxies approach for IV search, as for OOV search, but without using phone confusability transducer (phone-to-phone, P2P). This gives additional terms to look for in lattices and thus helps to improve search accuracy. Third, we apply pruning of the phone confusability transducer (P2P). And finally, we prune the proxy word automaton (P2P composed with phone-to-word in [15] notation). This procedure speeds up the query result extraction from the outcome of the CN WFST composition with the proxy word automation and increases efficiency of the sum-to-one normalization. It was our main approach to search for IV words. For OOV search we used it and a novel technique described in more details below.

[1]The modest results on Baseline+CRNN_2 combination are explained by the presence of the predominantly non-conversational Web data in the CRNN_2 training set. This led to much lower interpolation weight for CRNN_2 LM. The weight was selected to optimize the interpolated model perplexity on the official development set.

5.2. Decoder on high-level features for OOV search

The proposed approach to OOV search is based on using a modified Viterbi decoder working with the new type of high-level features derived from speech recognition lattices.

Table 3: *Comparison of the OOV search quality and speed for the decoder and for the proxies-based approach on the STC-dev list and different acoustic models*

AM	MTWV decoder	RTF decoder	MTWV proxies	RTF proxies
$\textbf{DMN}_9$	0.630	5.8e-05	0.528	0.0015
$\textbf{DMN}_8$	0.615	5.8e-05	0.491	0.0017
$\textbf{DMN}_3$	0.591	5.7e-05	0.512	0.0015

The extraction of the proposed features for audio files consists in three major steps:

1. Perform speech recognition based on words or subword units to produce the lattices.
2. Calculate phoneme posterior probabilities from word/subword lattices with phoneme alignments. Posterior probabilities are calculated using the forward-backward algorithm ([20, 21, 22]).
3. Smooth the obtained probabilities. Smoothing is performed by taking the weighted mean of a given feature vector with a constant one obtained from the confusion model which is trained in an unsupervised manner. This step allows to improve the accuracy of OOV search, especially in the case of sparse lattices.

The smoothed features are passed to the decoder.

We experimented on the 10 hour development set with the internal OOV keyword list. We found the list provided by the organizers containing too little amount of OOV words so we generated our own STC-dev list as prescribed in [27] with minor changes. The STC-dev list contained 742 OOV words. The recognition lexicon contained subword units which are described in Section 5.3.

The results in Table 3 show that the proposed approach outperforms proxies-based implementations available to us in terms of search quality and speed when applied on features derived from lattice generated by a single ASR system. When features are generated as a linear combination of posteriors derived from the lattices produced by all 9 our acoustic models, maximum attainable MTWV for proxies is slightly higher than for the decoder. However, the decoder achieves much better search quality when both approaches work with approximately

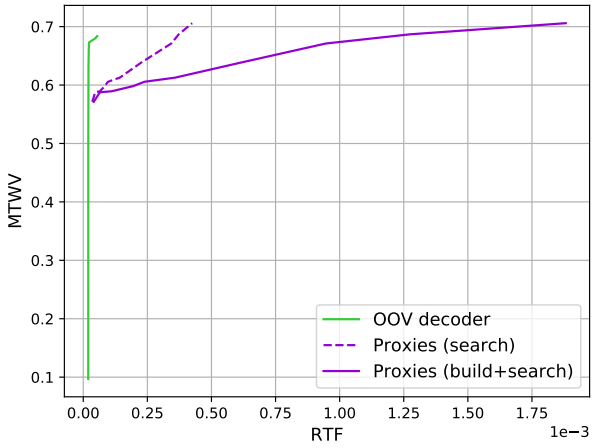

Figure 2: *The dependencies of MTWV against processing time (RTF) for the OOV search decoder and the proxies-based approach on combination of lattices from the 9 systems*

the same speed. The dependencies of MTWV against processing time represented as real time factor (RTF) are depicted in Figure 2.

Finally, the list-level combination of the two approaches gives an additional improvement in search quality, as shown in Table 5. The more detailed description of the OOV decoder is given in [28].

5.3. Subword units

In order to build a subword lexicon we employed the Factor toolkit (FTK) [29]. The allowed maximum unit length was 3 letters. We estimated frequencies of letter sequences on the training transcriptions and BBN clean data, fed the statistics to the FTK and trained the segmentation model on the acoustic training transcriptions only. The 4-gram subword language model was trained on the training transcriptions converted to the subword units.

Table 4: *The OOV search results on word and subword units*

Decoding units	**ATWV**	
	dev	**STC-dev**
words	0.398	0.281
subword units	**0.541**	**0.575**

In Table 4 results for different decoding units are presented. subword units based system clearly outperforms the word based analogue.

6. Systems combination

We used two system combination techniques: list-level and lattice-level ones. The former is implemented in Kaldi toolkit and was described in [30]. The latter is done by building a single confusion network from the lattices belonging to different systems and performing the search on it. Unlike list-level fusion, one doesn't have to run search for every system. When building a CN we scale each lattice posterior probabilities with weights obtained from the tuning procedure aimed to optimize keyword search accuracy.

Table 5: *Results for different system combination and OOV search techniques. In the table "prx" and "dec" refer to keyword search approaches described in Section 5.1 and Section 5.2 respectively*

		dev	**STC-dev**
	Fusion type	**ATWV(IV/OOV)**	**ATWV(OOV)**
1	prx lists	**0.749**/0.646	0.676
2	dec lists	-/-	0.727
3	prx lattices	-/-	0.711
4	dec lattices	0.581/0.666	0.688
	1+4 lists	0.731/**0.753**	0.766
	1+2 lists	-/-	**0.788**
	2+3 lists	-/-	0.785

As can be seen from the Table 5, the best result is achieved when the lists produced by different search techniques are combined. The list-level and lattice-level combination techniques do not seem to be complementary so we don't present ATWV obtained for 1+3 and 2+4 experiments in Table 5. Most figures in the table refer to our internal list because it contains much more OOV words than the official one, and our primary interest was to combine results from the different OOV search approaches.

7. Discussion and Conclusions

As follows from Table 1 the fully connected DNNs with maxout activations outperformed TDNN and BLSTM models, although the latter provided significant contribution into the final fusion result.

The language model used for lattice generation was built with the use of text generated by a character-level RNN. Initially we expected to reduce the OOV rate only, but the texts produced proved to be beneficial for word level statistics estimation as well. This may be attributed to the fact that Georgian is an agglutinative language. We plan to explore this approach on languages with different word generation patterns.

The OOV search approach introduced in Section 5.2 is shown to vastly outperform proxies-based implementations available to us in terms of processing speed while achieving comparable or better search accuracy values. Combination of the two approaches allows to increase ATWV of the whole system by 8% in comparison with the best result between the two approaches.

The presented system was among the top three systems in the OpenKWS 2016 evaluation with ATWV score of 0.821.

8. Acknowledgments

The work was financially supported by the Ministry of Education and Science of the Russian Federation. Contract 14.579.21.0121, ID RFMEFI57915X0121.

This effort uses the IARPA Babel Program language collection release IARPA-babel{101b-v0.4c, 102b-v0.5a, 103b-v0.4b, 201b-v0.2b, 203b-v3.1a, 205b-v1.0a, 206b-v0.1e, 207b-v1.0e, 301b-v2.0b, 302b-v1.0a, 303b-v1.0a, 304b-v1.0b, 305b-v1.0c, 306b-v2.0c, 307b-v1.0b, 401b-v2.0b, 402b-v1.0b, 403b-v1.0b, 404b-v1.0a}, set of training transcriptions and BBN part of clean web data for Georgian language.

9. References

[1] https://www.iarpa.gov/index.php/research-programs/babel.

[2] A. Karpathy, "The Unreasonable Effectiveness of Recurrent Neural Networks," http://karpathy.github.io/2015/05/21/rnn-effectiveness.

[3] KWS16 Evaluation Plan, https://www.nist.gov/sites/default/files/documents/itl/iad/mig/KWS16-evalplan-v04.pdf.

[4] W. Lee , J. Kim, and I. Lane, "Multi-stream combination for LVCSR and keyword search on GPU-accelerated platforms," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 3296–3300.

[5] S. P. Rath et al., "Combining tandem and hybrid systems for improved speech recognition and keyword spotting on low resource languages," in *Proc. INTERSPEECH 2014*, pp. 835839.

[6] J. Cui et al., "Multilingual representations for low resource speech recognition and keyword search," in *Proc. 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015.

[7] P. Golik, Z. Tuske, R. Schluter, and H. Ney, "Multilingual features based keyword search for very low-resource languages," in *Proc. INTERSPEECH 2015, Dresden, Germany, Sep. 2015*, pp. 1260-1264.

[8] I. Medennikov and A. Prudnikov, "Advances in STC Russian Spontaneous Speech Recognition System," in *Proc. SPECOM 2016*, pp. 116–123.

[9] I. Medennikov, A. Prudnikov, and A. Zatvornitskiy, "Improving English Conversational Telephone Speech Recognition," in *Proc. INTERSPEECH 2016, San Francisco, USA, Sep. 2016*, pp. 2–6.

[10] A. Prudnikov and M. Korenevsky, "Training Maxout Neural Networks for Speech Recognition Tasks," in *Proc. TSD 2016, LNAI 9924*, pp. 443–451.

[11] S. J. Rennie, V. Goel, and S. Thomas, "Annealed dropout training of deep networks," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pp. 159–164.

[12] A. Prudnikov, I. Medennikov, V. Mendelev, M. Korenevsky, and Y. Khokhlov, "Improving Acoustic Models For Russian Spontaneous Speech Recognition," in *Proc. SPECOM 2015*, pp. 234–242.

[13] G. Huang, A. Gorin, J. L. Gauvain, and L. Lamel, "Machine translation based data augmentation for Cantonese keyword spotting," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 6020–6024.

[14] L. Mangu, E. Brill, and A. Stolcke, "Finding Consensus in Speech Recognition: Word Error Minimization and Other Applications of Confusion Networks," *Computer Speech & Language*, vol. 14, pp. 495-498, October 2000.

[15] L. Mangu, B. Kingsbury, H. Soltau, H.-K. Kuo, and M. Picheny, "Efficient Spoken Term Detection Using Confusion Networks," in *Proc. of Intl Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2014.*

[16] D. Can and M. Saraclar, "Lattice indexing for spoken term detection," *IEEE Transactions on Audio, Speech, and Language Processing*, 19(8), 2011, pp. 2338–2347.

[17] D. Karakos and R. M. Schwartz, "Subword and phonetic search for detecting out-of-vocabulary keywords," in *Proc. INTERSPEECH 2014*, pp. 2469-2473.

[18] I. Bulyko, J. Herrero, C. Mihelich, and O. Kimball, "Subword speech recognition for detection of unseen words," in *Proc. INTERSPEECH 2012*.

[19] W. Hartmann, V. B. Le, A. Messaoudi, L. Lamel, and J.-L. Gauvain, "Comparing decoding strategies for subword-based keyword spotting in low-resourced languages," in *Proc. INTERSPEECH 2014*, pp. 2764-2768.

[20] L. Uebel and P. C. Woodland, "Improvements in linear transform based speaker adaptation," in *Proc. ICASSP 2001*, pp. 49-52.

[21] C. Gollan and M. Bacchiani, "Confidence scores for acoustic model adaptation," in *Proc. ICASSP 2008*, pp. 4289-4292.

[22] G. Evermann and P. C. Woodland, "Large vocabulary decoding and confidence estimation using word posterior probabilities," in *Proc. ICASSP 2000*, pp. 1655-1658.

[23] D. Povey et al., "The Kaldi speech recognition toolkit," in *Proc. 2011 IEEE workshop on automatic speech recognition and understanding (ASRU)*, 2011.

[24] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. INTERSPEECH 2015, Dresden, Germany, Sep. 2015*, pp. 2440–2444.

[25] H. Sak, A. Senior, and F. Beaufays, "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling," in *Proc. INTERSPEECH 2014*.

[26] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proc. INTERSPEECH 2015, Dresden, Germany, Sep. 2015.*

[27] J. Cui, J. Mamou, B. Kingsbury, and B. Ramabhadran, "Automatic keyword selection for keyword search development and tuning," in *Proc. ICASSP 2014*, pp. 7839–7841.

[28] Y. Khokhlov, N. Tomashenko, I. Medennikov, A. Romanenko "Fast and accurate OOV decoder on high-level features," in *Interspeech, 2017.*

[29] M. Varjokallio and M. Kurimo, "A Toolkit For Efficient Learning of Lexical Units for Speech Recognition," in *Proc. of The 9th edition of the Language Resources and Evaluation Conference (LREC 2014), 26-31 May, Reykjavik, Iceland, 2014.*

[30] J. Trmal et al., "A keyword search system using open source software," in *Proc. IEEE Workshop on Spoken Language Technology, South Lake Tahoe, NV; USA, Dec. 2014, IEEE*