Syllable based Lattice Transduction for Keyword Search

Hang Su

December 23, 2015

Acknowledgement

I would like to thank my mentors James Hieronymus and Steven Wegmann for their detailed guidance on this work. I learned a lot through collaboration with both of them. I also want to thank my advisor Nelson Morgan for his support and patience. He gave me a lot of freedom to explore different research ideas. I thank Ryan Yanzhang He and Eric Fosler-Lussier for their help on implementing the framework of our work, and thank Brian Hutchinson, Peter Baumann and Aaron Jaech for their help on boosted language model. I also thank Korbinian Riedhammer for his generous help on syllable decoding, and Daniel Povey for his enlightening advice on lattice alignment.

Syllable based Lattice Transduction for Keyword Search

by Hang Su

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor Nelson Morgan
Research Advisor

(Date)

* * * * * * *

Doctor James Hieronymus
Second Reader

(Date)

Abstract

Low-resource speech recognition and keyword search (KWS) are important topics for speech technologies. However, their performance often suffers from out-of-vocabulary (OOV) keywords. Subword units like syllables are useful in handling this issue. This report introduces a weighted finite state transducer (WFST) based syllable transduction framework for OOV handling in KWS. Syllable lattices are generated by performing syllable decoding and OOV keywords are entered into a pronunciation dictionary using a word-to-syllable pronunciation prediction system. Syllable lattices are then transduced into word lattices using both in-vocabulary word pronunciations and OOV pronunciations. Experiments on 5 languages provided by IARPA Babel project [1] are presented, and it is shown that syllable transduction can effectively spot OOV keywords. Combination of this approach with two other OOV handling methods further improves keyword search performance.

[1]Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0014. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

Acknowledgement

I would like to thank my mentors James Hieronymus and Steven Wegmann for their detailed guidance on this work. I learned a lot through collaboration with both of them. I also want to thank my advisor Nelson Morgan for his support and patience. He gave me a lot of freedom to explore different research ideas.
I thank Ryan Yanzhang He and Eric Fosler-Lussier for their help on implementing the framework of our work, and thank Brian Hutchinson, Peter Baumann and Aaron Jaech for their help on boosted language model. I also thank Korbinian Riedhammer for his generous help on syllable decoding, and Daniel Povey for his enlightening advice on lattice alignment.

Contents

1 Introduction

Nowadays, keyword search in speech/audio is becoming more and more important as lots of multimedia are available on the Internet. However, unlike text search, audio/speech search is a much harder task. Researchers have been developing Keyword Search (KWS) systems based on speech recognition technology, which itself is an unsolved artificial intelligence problem. Thanks to the progress made by the speech community, speech search is becoming a more practical technique in recent years.
Multilingual speech search presents a unique set of challenges, including novel speech sounds, agglomerative morphology which causes large vocabularies, and lack of transcribed data for model training. Especially, when training data is limited, there will be lots of out-of-vocabulary (OOV) words in the test set that makes it impractical to search them without special treatment.
The IARPA Babel program [1] instantiates this scenario well in providing a limited amount of transcribed training data and lexicons for words and syllables in several minority languages. For the five languages provided in 2014 the numbers of OOV keywords are from 10% to 25%. This report investigates KWS in the Babel low-resource condition, with an emphasis on the OOV issue.
The organization of the report is as follows. Section 2 introduces the background of speech recognition and keyword search, together with two OOV handling methods. Section 3 illustrates the WFST framework for syllable transduction. Section 4 introduces the BABEL project and Section 5 presents experimental results on five languages. Section 6 is the conclusion.

2 Background

KWS in audio is usually based on speech recognition. A set of audio files and their corresponding transcriptions are given as training data to train a speech recognizer. Test audio files are then transcribed via Automatic Speech Recognition (ASR) and word lattices containing possible transcriptions are generated (this procedure is called decoding). KWS is conducted based on the produced word lattices.

2.1 WFST Framework for ASR

The speech recognition task could be well formulated as a composition task in the framework of weighted finite state transducers (WFST). A typical WFST decoding graph generation in speech recognition [2] is represented as

$$H \circ C \circ L \circ G \tag{1}$$

where H, C, L and G are WFSTs for a state network of triphone HMMs, context-dependency transducer of phones, pronunciation lexicon for words, and an n-gram word LM, respectively; operation $\circ$ represents the composition operator.

2.2 Keyword search

Keyword search is usually done in four phases:

- Index generation. Given a list of keywords and word lattices, generate indexes for keywords that appear in word lattices.
- Score normalization. Normalize posterior scores (or other confidence scores) for each hypothesis in the index.
- System combination. Combine hypotheses from several different systems.
- Thresholding. Set thresholds for keywords and output searching results based on hypotheses and the thresholds.

There are two widely-used keyword search strategies for index generation. Both of them are based on decoded word lattices from ASR, but they use different searching algorithms.

2.2.1 WFST based indexation

Decoded lattices are converted into finite-state acceptors with posterior score, start time and end-time for each word encoded as weights. An inverted index is then created from these individual acceptor, with paths to accept every possible word sequence in the original lattice. KWS is done via composition of keyword acceptors K and the inverted index, and posterior score, start time and end time for keywords could be obtained at the same time. Finally, a yes/no decision is made according to the posterior score from the search.

2.2.2 Word lattice based indexation

Decoded lattices are traversed and a word based index is created tracking all of the words that occur in the lattices. For single word keywords, a list of all the keyword occurrences is returned. For multiword keywords, one retrieve the individual words from the index in the correct order with respect to their start and end times but discard occurrences where the time gap between adjacent words is bigger than a predefined threshold.

2.3 Methods for OOV handling

Out-of-vocabulary (OOV) words are words in the test set that do not appear in the training set. When an OOV word turns out to be a keyword to be searched, a standard searching pipeline on word lattices will not produce any hypotheses.
One way to mitigate OOV issue is to find in-vocabulary (IV) words which are closest in pronunciation to the OOV keywords. Confusion matrices are used in [3–6] to generate alternatives words or string of words to stand as proxies for OOV keywords. These systems provide a way of dealing with OOV keywords which have a close IV keyword or series of IV keywords in the training data.
Another way to handle OOV keywords is to use sub-word units. It is possible to use phones, syllables and morphs as subword units for representing OOV words. Subword index is created either by performing subword decoding [7–9] or by converting word lattice into sub-word lattice [10,11]. Hartmann et al. compares converting word-based lattices to subword lattices, separate decoding for each subword type and single decoding using all possible subword units, reporting a best performance by carrying out a separate decoding for each subword type [12].

2.3.1 Word Proxy

This approach tries to find IV word proxies which are closest in pronunciation to the OOV keywords. Let K represent a finite-state acceptor for an OOV keyword, L_2 a finite state transducer for lexicon containing OOV keywords, E be an edit-distance transducer that maps a phone sequence to another phone sequence with costs estimated from a phone confusion matrix, L_1 denote the pronunciation lexicon for IV words. The proxy keyword K' can be generated by

$$K' = Project(ShortestPath(K \circ L_2 \circ E \circ (L_1)^{-1})) \tag{2}$$

E is a phone to phone confusion matrix expressed in a finite-transducer form, estimated through standard maximum likelihood estimation. Training data for the conditional probabilities estimation are collected by aligning the reference phone string to the ASR hypothesis on a held-out set.

2.3.2 Subword Search

Direct search of OOV keywords in subword lattices serves as another baseline method. It follows the pipeline in [8]. Instead of doing mixed word and subword decoding, we decode with subwords only (i.e., syllables) in this work. We create a syllable-based index from the lattices, tracking all of the syllables that occur in the lattices, their start and end times, and their lattice posterior probabilities. Keywords can be searched from the index with their corresponding syllable representation.

For multiword keywords, their representation would be the cross product of all the representations of each component word.

3 Syllable Lattice Transduction for KWS

This method follows the syllable decoding procedure first, generating syllable lattices. Instead of searching for keywords directly in subword lattices, we transduce syllable lattices to word lattices, using G2S systems to produce syllable pronunciations for the OOV keywords. This method has the advantage of using larger, less confusable units than phones for the decoding, and has a weak LM because many possible syllable sequences are allowed within a language. For a number of OOV words, the pronunciation contains syllables which were not seen in the training data. For those OOV syllables predicted by G2S system, we choose the perceptually nearest IV syllable to substitute for the OOV syllable. The match requires that the vowel nucleus and the onset consonants match closely and the coda consonants be nearest in place and manner of articulation, following the results of phone perceptual experiments [13].

3.1 Syllable Decoding

To perform a syllable decoding, we substitute word transducers with syllable transducers in the standard word decoding graph generation

$$H \circ C \circ L_{phn2syl} \circ G_{syl} \tag{3}$$

where $L_{phn2syl}$ denotes lexical transducer for syllable-phone pronunciations and G_{syl} is syllable LM.
$L_{phn2syl}$ can be easily constructed using a syllable lexicon given by experts. For syllable language model G_{syl}, a simple way is decomposing words into syllable sequences using a word to syllable lexicon. As words may have several pronunciations, randomly picking could be used. However, a more appropriate approach is first aligning transcriptions with acoustics using trained acoustic models, and then mapping words to syllable sequences that match phone alignments.
With the resulting syllable transcription, we can build an n-gram language model and use it for decoding. Note that in a low-resources condition, syllable LMs tend to be better modeled by n-gram than word LMs because the number of syllables is usually less than the number of words, making training data relatively sufficient.

3.2 Grapheme-to-syllable prediction

Our pronunciation prediction utilizes the Phonetisaurus G2P system [14] and trains on IV pronunciations. This system itself is WFST-based, and predicts pronunciations based on a multigram alignment between graphemes and phonemes. Our initial experiments aligning multiple character sequences to syllable symbols proved that the space is too sparse to learn syllables directly. In order to utilize the better accuracy of G2P when predicting syllables, we exploit the fact that Phonetisaurus is WFST-based, and impose additional constraints on the output of the system to produce syllables.

For each language, we collect statistics over which phones can appear in onset, nucleus, or coda positions; we also collect statistics over the different kinds of syllable structures (including frequency of onset clusters or coda clusters). Then two transducers are created: one that maps phones to the same phone with possible syllable positions, and another that maps the phone/syllable position pairs to the syllable position. We also create an acceptor that provides a unigram language model over valid syllable structures. When these three constraints are composed and realized as a phone to phone/syllable position pair transducer, this can be used as a constraint to be composed with the original Phonetisaurus G2P system, but produces phones annotated with syllable positions. We can then read off the syllable structure of the predicted phone pronunciation easily.

This G2S system described above can produce syllables that do not appear in training data set (i.e. OOV syllables). Once these OOV syllables have been found, it is necessary to find the perceptually nearest IV syllable to be a proxy for them. We use a syllable to phone system to find the phone pronunciation of the OOV syllable and then match it to the pronunciations of the IV syllables using a metric which weighs the vowel identity highest, the onset consonants the next highest and the coda consonants the lowest. This weighting is justified by perceptual experiments which show humans perceive the vowel and prevocalic consonants better than the postvocalic consonants [13]. As a first step, we only selected one IV syllable per OOV syllable.

3.3 Syllable to Word Transduction

After generating syllable lattices, we can construct a syllable to word lexical transducer $L_{syl2wrd}$ using the Babel lexicon (we will cover OOV part in next section) and then compose it with syllable lattices to get word lattices. This last step concludes the lattice generation part, while for KWS, we need one extra step (word alignment) to retrieve time information from composed word lattices.

3.4 Boosted Language Model

To better exploit the knowledge of keywords, a unigram language model is trained on all keywords and then interpolated with original word language model. This boosted language model is compiled into a grammar WFST and then composed with syllable to word lexical transducer. To use the composed lexical transducer, we first remove language model scores in syllable lattices, and then perform transduction to word lattices via composition, i.e.

$$\hat{Lat}_{syl} \circ L_{syl2wrd} \circ G_{boost} \tag{4}$$

where $\hat{Lat}_{syl}$ denotes syllable lattices without language model score, $L_{syl2wrd}$ denotes lexical transducer for word-syllable pronunciations and G_{boost} is boosted LM.

4 The Babel Project

4.1 Background

The Babel Program focuses on 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]. It provides language packs on several different minority languages. Each language pack contains a limited amount of training data which has been transcribed at the word level and lexicons for words and syllables. The data we used in this study is divided into subsets called the full language pack (FLP) and the limited language pack (LLP) which have approximately 65 hours and 10 hours of training data respectively. Around 10 hours of development data are provided for performance testing and parameter tuning. In addition, an evaluation set is provided for final system evaluation, and around one third of the evaluation set, called eval-part1, is given for evaluation analysis. In this work, we used the LLP for training and eval-part1 for testing.
In the work we reported here, performance evaluation is split into 3 stages:

- training stage (training data and development data is released and researchers begin building models and tuning recognition parameters);
- ingestion stage (evaluation data is released and used for decoding);
- search stage (keywords are given and KWS is performed).

A so-called ”No test audio re-use” (NTAR) condition requires no decoding operation be performed in search stage, reducing the total amount of time it takes to perform KWS. Doing syllable decoding in the second stage and adding in OOV pronunciations before transduction allows the OOV words to be recognized in the third stage, while satisfying the NTAR condition. OOVs are handled via G2S and a mapping procedure previously described.

4.2 Languages

Here I am reporting experiments on 5 different languages: Assamese, Bengali, Creole, Zulu and Tamil. The version of the language packs are summarized in Table 1, and Table 2 records actual speech time (after segmentation).

	version	keyword list
Assamese	IARPA-babel102b-v0.5a	conv-eval.kwlist4
Bengali	IARPA-babel103b-v0.4b	conv-eval.kwlist4
Creole	IARPA-babel201b-v0.2b	conv-eval.kwlist4
Zulu	IARPA-babel206b-v0.1e	conv-eval.kwlist4
Tamil	IARPA-babel204b-v1.1b	conv-eval.kwlist5

Table 1: Babel data for OP1 languages

	LLP-training	dev	evalp1
Assamese	10.03	8.67	3.69
Bengali	10.32	8.83	4.81
Creole	11.36	9.63	4.27
Zulu	10.38	9.95	4.22
Tamil	11.77	10.33	13.11

Table 2: Babel audio data in hours

4.3 Statistics

Table 3 shows language pack and keyword list statistics. In general, #Words-to#Syls ratio is between 2 to 10, and OOV rate varies from 16% to 22%[2].

	#Words	#Syls	#KWs	#OOV KWs
Assamese	7661	1679	1608	259
Bengali	7933	2082	1594	283
Creole	4897	1981	1533	287
Zulu	13674	1345	1412	380
Tamil	14265	2620	2188	500

Table 3: Statistics for language packs

[2]OOV are counted with regard to `eval-part1`

5 Experiments

5.1 Setup

The Kaldi toolkit [15] is used for the speech recognition part. A standard 13 dimensional PLP feature, together with 3-dim Kaldi pitch feature [16], is extracted and used for maximum likelihood GMM model training. Features are then transformed using LDA+MLLT before SAT training. With 'standard' GMM training recipe performed, a tanh-neuron DNN-HMM hybrid system is trained using the same features. Details of DNN training are documented in section 2.2 in [17]. The major difference between our setup and default Kaldi setup is that we use word position-independent phones for acoustic models. This is necessary for syllable transduction with word alignment because position-dependent phones would blow up the alignment lexicon for lattice word alignment.

5.2 Syllable Decoding

To evaluate the performance of syllable decoding, we need to map transcriptions from word to syllables. Just as described in Section 3.1, we force align transcriptions with the acoustic training data to reach a more accurate result.
Table 4 shows WERs of word decoding and Syllable Error Rates (SERs) of syllable decoding (first two columns)[3]. These two metrics are not comparable in general – we present them here just for a quick reference. It is shown that syllable error rate is usually lower than word error rate, which is reasonable in that a correct word recognition requires all syllables within the word to be correctly recognized.

	WER	SER	S2W ER
Assamese	66.6	64.1	70.7
Bengali	68.9	63.4	73.7
Creole	61.6	57.3	67.1
Zulu	71.5	73.7	77.5
Tamil	79.0	77.9	81.0

Table 4: Syllable Error Rate and Syllable-to-word Error Rate

5.3 Grapheme to Syllable

G2S prediction is evaluated by comparing the ground truth lexicon with predicted lexicon. BABEL FLP language pack provides a lexicon that covers more words

[3]"S2W ER" refers to Syllable-to-word Error Rate which will be introduced in 5.4

than those appear in LLP. Since the G2S training only uses pronunciations of words in LLP, all other words can serve as evaluation set. Table 5 shows the phone error rate (PER) and syllable error rate (SER) on that 'OOV' set (i.e. FLP v.s. LLP). Note that the predicted SER concept in this section is different from the recognized SER.

	PER	SER
Assamese	7.0	18.7
Bengali	9.7	25.2
Creole	5.7	31.5
Zulu	5.9	11.9
Tamil	2.2	7.6

Table 5: Pronunciation Prediction Error Rate

It is shown in Table 5 that SERs for Creole and Bengali are quite high. Actually, those high SERs are mainly caused by mis-assign of phones to successive syllables, and these may not influence following procedure much since syllable lattices may contain different assignments as well and may compensate for that.

5.4 Lattice Transduction

WER is used as a validation metric for lattice transduction. In this part, we do not use a keyword boosted language model as it is designed for KWS task rather than recognition. Table 4 shows WERs for baseline word decoding versus syllable transduction word error rate (first and last column). It can be observed that syllable transduced lattices have a higher word error rate, which indicates word language models are still stronger than syllable ones in terms of word recognition.

5.5 Keyword Search

5.5.1 ATWV

KWS performance is usually measured by the metric ATWV, which was developed for the NIST 2006 Spoken Term Detection evaluation [18]. The score computed in the following way: for a given keyword kw in a posting list (a list of putative hits with start, end times and detection confidence) and detection threshold θ, the probabilities of miss and false alarm rates are computed using

$$\begin{aligned} P_{Miss}(kw,\theta) &= 1 - N_{Correct}(kw,\theta)/N_{Ref}(kw) \\ P_{FA}(kw,\theta) &= N_{Spurious}(kw,\theta)/N_{NT}(kw) \end{aligned} \tag{5}$$

where $N_{NT}(kw, \theta)$ is the number of correctly hypothesized posting list entries with detection scores $\geq \theta$, $N_{Spurious}(kw, \theta)$ is the number of incorrectly hypothesized posting list entries with detection scores $\geq \theta$, $N_{Ref}(kw)$ is the number of reference occurrences, and $N_{NT}(kw)$ is the number of non-target trials for kw in the data, which is computed as

$$N_{NT}(kw) = T_{Audio} - N_{Ref}(kw) \tag{6}$$

For a specific keyword kw, the Term Weighted Value (TWV) is defined as

$$TWV(kw, \theta) = 1 - P_{Miss}(kw, \theta) - \beta P_{FA}(kw, \theta) \tag{7}$$

Assuming there are K keywords in total, ATWV is defined as the average TWV of all keywords at a given detection threshold $\hat{\theta}$.

$$ATWV = \sum_{kw} TWV(kw, \hat{\theta})/K \tag{8}$$

For every keywords, $TWV(kw, \theta)$ is in the range of $(-\infty, 1]$. Unlike WER, a higher ATWV is better. It achieves its upper bound when there are no misses and no false alarms.

5.5.2 IV Keyword Search

Though the main focus of this work is on OOV KWS, IV ATWV still serves as an indicator for evaluating effectiveness of different methods. Table 6 shows IV ATWV for baseline word system (Word), syllable search (SylS) and syllable transduction (SylT). It shows that the syllable transduction method gives reasonable IV

	Word	SylS	SylT
Assamese	0.3064	0.2539	0.2958
Bengali	0.3094	0.2523	0.2914
Creole	0.3759	0.3367	0.3640
Zulu	0.3139	0.2401	0.2572
Tamil	0.2595	0.2123	0.2203

Table 6: IV ATWV

ATWV, indicating that the transduction method is effective in spotting both IV and OOV keywords.

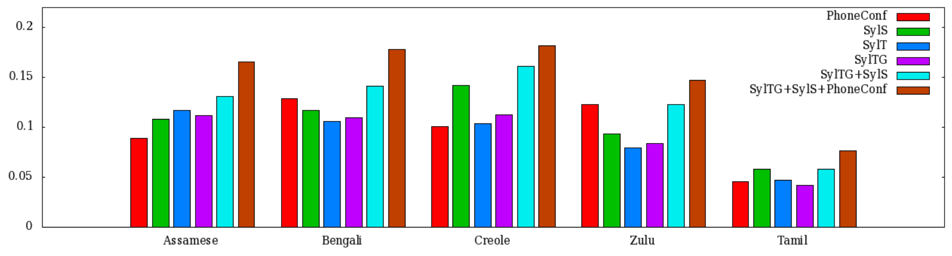

Figure 1: OOV ATWV for different languages

5.5.3 OOV Keyword Search

Figure 1 shows OOV ATWV for all 5 languages. We see that syllable transduction (SylT) generally gives comparable results to the other two methods, i.e. phone confusion (PhoneConf) and syllable search (SylS).
To get a feeling for the characteristics of these three methods, we present miss rate and false alarm rate on Assamese in Table 7. It shows that the syllable transduction method tends to give lower false alarm rate, indicating more accurate hypotheses.

	PhoneConf	SylS	SylT
PMiss	0.853	0.827	0.859
PFA	0.00006	0.00006	0.00003

Table 7: PMiss and PFA for Assamese

5.5.4 Combination

Figure 1 also shows the OOV ATWV on system combination. It is shown that system combination can effectively improves the OOV ATWV [19].

5.6 Analysis

The phone confusion method usually generates many more hypotheses than the other two methods, giving a higher hit rate as well as false alarm rate. The generated index file is usually much bigger than the other two methods, and it slows down the KWS. In addition, it uses the dev set for confusion matrix estimation, which makes tuning of hyper-parameters a bit complicated.
Syllable search usually generates a reasonable amount of hypotheses and gives good search results on OOV keywords. This method does not require post-processing for syllable lattices. On the other hand, searching syllables in lattices usually takes more time than searching in word lattices, especially when lattices are dense.

Compared with the above methods, syllable transduction usually gives fewer but more accurate hypotheses, and is good at spotting both IV and OOV keywords. It has been shown that combination of a word system and a syllable transduction system gives better results than combining the word system with syllable search [20]. This method also provides word lattices that are useful for OOV recognition. While it does not require any modification of the KWS template, this method requires a G2S system and a boosted language model for better performance.
In general, these three methods combine well in terms of ATWV. This combination strategy does not require training multiple acoustic models, which reduces the training time and computation greatly.

6 Conclusion

We show that syllable transduction is helpful in handling OOVs in low resources KWS tasks. Its good performance in both IV and OOV ATWV makes it a serious alternative to direct-searching in subword lattices. A keyword boosted language model further improves ATWV by mixing in unigram trained from keywords. Future work may benefit from improving syllable decoding accuracy, adding in more OOV pronunciations, and developing a composition algorithm that preserves time information in lattices.

References

[1] "Iarpa babel program broad agency announcement," 2014, http://www.iarpa.gov/index.php/research-programs/babel.

[2] Mehryar Mohri, Fernando Pereira, and Michael Riley, "Speech recognition with weighted finite-state transducers," in *Springer Handbook of Speech Processing*. 2008.

[3] Lidia Mangu, Brian Kingsbury, Hagen Soltau, Hong-Kwang Kuo, and Michael Picheny, "Efficient spoken term detection using confusion networks," in *ICASSP*, 2014.

[4] Guoguo Chen, Oguz Yilmaz, Jan Trmal, Daniel Povey, and Sanjeev Khudanpur, "Using proxies for oov keywords in the keyword search task," in *ASRU*, 2013.

[5] Yuk-Chi Li, Wai-Kit Lo, Helen M Meng, and PC Ching, "Query expansion using phonetic confusions for chinese spoken document retrieval," in *Proceedings of the fifth international workshop on on Information retrieval with Asian languages*, 2000.

[6] Murat Saraclar, Abhinav Sethy, Bhuvana Ramabhadran, Lidia Mangu, Jia Cui, Xiaodong Cui, Brian Kingsbury, and Jonathan Mamou, "An empirical study of confusion modeling in keyword search for low resource languages," in *ASRU*, 2013.

[7] Olivier Siohan and Michiel Bacchiani, "Fast vocabulary-independent audio search using path-based graph indexing," in *Interspeech*, 2005.

[8] Yanzhang He, Brian Hutchinson, Peter Baumann, Mari Ostendorf, Eric Fosler-Lussier, and Janet Pierrehumbert, "Subword-based modeling for handling oov words inkeyword spotting," in *ICASSP*, 2014.

[9] Ivan Bulyko, José Herrero, Chris Mihelich, and Owen Kimball, "Subword speech recognition for detection of unseen words.," in *Interspeech*, 2012.

[10] Damianos Karakos, Ivan Bulyko, Richard Schwartz, Stavros Tsakalidis, Long Nguyen, and John Makhoul, "Normalization of phonetic keyword search scores," in *ICASSP*, 2014.

[11] Murat Saraclar and Richard Sproat, "Lattice-based search for spoken utterance retrieval," in *HLT-NAACL*, 2004.

[12] William Hartmann, Viet-Bac Le, Abdel Messaoudi, Lori Lamel, and Jean-Luc Gauvain, "Comparing decoding strategies for subword-based keyword spotting in low-resourced languages," in *Interspeech*, 2014.

[13] Melissa A Redford and Randy L Diehl, "The relative perceptual distinctiveness of initial and final consonants in cvc syllables," *The Journal of the Acoustical Society of America*, 1999.

[14] Josef R Novak, Nobuaki Minematsu, and Keikichi Hirose, "Wfst-based grapheme-to-phoneme conversion: open source tools for alignment, model-building and decoding," in *10th International Workshop on Finite State Methods and Natural Language Processing*, 2012.

[15] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *ASRU*, 2011.

[16] Pegah Ghahremani, Bagher BabaAli, Daniel Povey, Korbinian Riedhammer, Jan Trmal, and Sanjeev Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *ICASSP*, 2014.

[17] Xiaohui Zhang, Jan Trmal, Daniel Povey, and Sanjeev Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *ICASSP*, 2014.

[18] Jonathan G Fiscus, Jerome Ajot, John S Garofolo, and George Doddingtion, "Results of the 2006 spoken term detection evaluation," in *Proc. SIGIR*, 2007.

[19] Hang Su, Van Tung Pham, Yanzhang He, and James Hieronymus, "Improvements on transducing syllable lattice to word lattice for keyword search," 2015.

[20] Hang Su, James Hieronymus, Yanzhang He, Eric Fosler-Lussier, and Steve Wegmann, "Syllable based keyword search: Transducing syllable lattices to word lattices," in *Spoken Language Technology Workshop*, 2014.