IMPROVEMENTS ON TRANSDUCING SYLLABLE LATTICE TO WORD LATTICE FOR KEYWORD SEARCH

Hang Su[1,2], Van Tung Pham[3], Yanzhang He[4], James Hieronymus[1]

[1] International Computer Science Institute, Berkeley, California, USA
[2] Dept. of Electrical Engineering & Computer Science, University of California, Berkeley, CA, USA
[3] Nanyang Technological University, Singapore
[4] Dept. of Computer Science & Engineering, The Ohio State University, Columbus, OH, USA

ABSTRACT

This paper investigates a weighted finite state transducer (WFST) based syllable decoding and transduction method for keyword search (KWS), and compares it with sub-word search and phone confusion methods in detail. Acoustic context dependent phone models are trained from word forced alignments and then used for syllable decoding and lattice generation. Out-of-vocabulary (OOV) keyword pronunciations are produced using a grapheme-to-syllable (G2S) system and then used to construct a lexical transducer. The lexical transducer is then composed with a keyword-boosted language model (LM) to transduce the syllable lattices to word lattices for final KWS. Word Error Rates (WER) and KWS results are reported for 5 different languages. It is shown that the syllable transduction method gives comparable KWS results to the syllable search and phone confusion methods. Combination of these three methods further improves OOV KWS performance.

Index Terms— Speech Recognition, Keyword Search, OOV Keywords, Syllable Transduction, WFST

1. INTRODUCTION

KWS for multilingual speech is challenging because of the presences of novel speech sounds, agglomerative morphology and the lack of transcribed data for training. The IARPA Babel program [1] aims to solve these problems by providing a limited amount of transcribed training data and lexicons for words and syllables in several minority languages. In a resource limited setting like this, spotting OOV keyword becomes essential for high performance measured by the metric Actual Term Weighted Value (ATWV) [2].

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.

One way to handle OOV keywords is to find IV words which are closest in pronunciation to the OOV keywords [3, 4]. Confusion matrices can be used [5–8] to generate alternative words or strings of words to stand as proxies for OOV keywords. Another method is to use sub-word units like phones, syllables or morphs for representing OOV words. Subword lattices are created either by performing subword decoding [9–12] or by converting the word lattice into a subword lattice [13,14]. Hartmann et al. compare different strategies and report best performance by carrying out separate decodings for each subword type [15]. Our method uses syllables as the decoding subword unit, and generates syllable lattices first. Instead of searching subword lattices, we transduce syllable lattices to word lattices, using a G2S system to produce syllable pronunciations for the OOV keywords. Syllables produced by the G2S system which do not appear in the training data and therefore do not have pronunciations are replaced by the perceptually nearest in vocabulary syllable. Syllable lattices are then transduced to word lattices using the lexical transducer and a boosted language model.

This work covers 5 different languages and compares syllable transduction with the other two OOV handling methods mentioned above. Both recognition WER and ATWV are reported and the pros and cons of these three methods are discussed in detail. Our experiments also show that these methods combine well. Section 2 describes the WFST framework for lattice transduction, Section 3 discusses related work. Section 4 describes our experimental setup and Section 5 gives the recognition and KWS results on 5 different Babel languages. Section 6 gives analysis and Section 7 presents our conclusion.

2. SYLLABLE TRANSDUCTION FRAMEWORK

2.1. Syllable decoding

The typical WFST decoding framework in speech recognition [16] 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, a context-dependent transducer of phones, a pronunciation lexicon for words, and an n-gram word LM, respectively; $\circ$ represents the composition operator. To perform a syllable decoding, substituting syllable transducers for word transducers gives

$$H \circ C \circ L_{phn2syl} \circ G_{syl} \quad (2)$$

where $L_{phn2syl}$ denotes a lexical transducer for syllable-phone pronunciations and G_{syl} is a syllable LM.

$L_{phn2syl}$ can be constructed using the syllable lexicon given by the BABEL program. For a syllable language model G_{syl}, we need to decompose word transcriptions to syllable transcriptions. Since each word may have multiple pronunciations, we first align word sequences with acoustics using the trained acoustic models, and then map words to syllable sequences that match the phone alignments. Consider the word record in English, it can be pronounced r i k o r d or r E k er d depending on whether it is a verb or noun. For good trisyllable modeling knowing which syllables were produced is important. Preliminary experiments showed that a syllable language model trained on aligned syllable transcriptions gives better performance than randomly picking word-to-syllable pronunciations during decomposition.

2.2. Handling OOVs via G2S

Our pronunciation prediction utilizes the Phonetisaurus G2P system [17] trained on IV pronunciations. There is too little data to train an accurate G2S system for an average of 2000 syllables per language. It is better to use the better accuracy of G2P and find a way to put in syllable boundaries. We exploit the fact that Phonetisaurus is WFST-based, and impose additional constraints on the output of the system to produce syllables. We collect statistics over which phones can appear in onset, nucleus, or coda positions; and also statistics over the different kinds of syllable structures found in the data. Languages are very specific on which consonents are allowed in onset and coda positions as well as what syllable structure is allowed. 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. These three constraints are composed and used as a constraint to be composed with the original Phonetisaurus G2P system. We can then read off the syllable structure of the predicted phone pronunciation easily, selecting the most likely syllable boundary.

For OOV syllables predicted by the G2S system described above, we match them to nearest IV syllables using a metric over phone pronunciations which weights 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 [18].

2.3. Syllable to Word Transduction

After generating syllable lattices and OOV pronunciations, we construct a syllable to word lexical transducer $L_{syl2wrd}$ and then compose it with syllable lattices to get word lattices. For KWS, words in the lattices are aligned to states using the lexicon to retrieve time information.

2.4. Boosted Language Model

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

$$\hat{Lat}_{syl} \circ L_{syl2wrd} \circ G_{boost} \quad (3)$$

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.

The use of keyword information in this routine satisfies the so-called "No test audio re-use" (NTAR) condition in BABEL program because decoding is done before keyword information are used.

3. OTHER KWS METHODS

3.1. Direct Search for Keywords in Syllable Lattices

Direct search for OOV keywords in subword lattices serves as a baseline in this paper, following the pipeline in [10]. However, instead of doing mixed word and subword decoding, we decode with syllables only. We create a syllable-based index from the lattices, tracking all the syllables 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 word.

3.2. OOV Keyword Proxy using Phone Confusion

Using phone confusions for proxy keyword generation serves as another baseline in this paper. The general framework is described in [6]. Specifically, a list of proxies for each OOV keyword K is generated using the following procedure:

$$K' = Project(ShortestPath(K \circ L_2 \circ E \circ (L_1^*)^{-1})) \quad (4)$$

where L_1 is the original pronunciation lexicon, L_2 is L_1 augmented with phone pronunciations of OOV words, E is the phone-to-phone confusion transducer and K' are the proxies of K. The OOV pronunciations are automatically obtained using the G2P tool [19].

The phone confusion model E reflects the error patterns of ASR system. Thus it is necessary to estimate E on development data. For each utterance of the dev set, the reference phone string (from forced-alignment) is aligned with the ASR phone hypothesis (best decoding path) to obtain confusion statistics. These statistics are then encoded into transducer E. Because the composed transducer $K \circ L_2 \circ E' \circ (L_1^*)^{-1}$ can be very large, we prune it using the ShortestPath algorithm with beam 5, and then select the top 500 proxies for each OOV keyword.

4. EXPERIMENTAL SETUP

4.1. Data

BABEL data shown in Table 1 are used for the recognition and KWS experiments. Each language pack divides into subsets of full language pack (FLP) and the limited language pack (LLP). Around 10 hours of development data is provided for parameter tuning, and test sets are used for final evaluation. In this work, WER and ATWV are reported all based on `eval-part1` data, which is a subset of evaluation set. 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

4.2. Recognition System

The Kaldi toolkit [20] was used for speech recognition in this work. The standard 13-dim PLP features, together with 3-dim Kaldi pitch feature [21], were extracted and used for maximum likelihood GMM model training. Features are then transformed using LDA+MLLT before the SAT training. After a standard GMM training recipe is 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 [22]. 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 because position-dependent phones make the lexicon too large for lattice word alignment.

4.3. KWS System

The KWS experiments for syllable transduction and syllable lattice search generally follow the method described in [23]. Tuning of the KWS parameters (i.e. LM scale, posterior scale and fraction) are done on development data using the Nelder Mead optimization method [24]. For the phone confusion word proxy method, we use the Kaldi toolkit for proxy generation and OOV search. KST score normalization [25] was used in all our scoring methods.

5. EXPERIMENTS

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%[1].

	#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 pack

5.1. Syllable Transduction

After transducing the syllable lattices to word lattices, we compute WER based on word lattices and compare them with word based recognition systems. Table 4 shows WER comparison for all 5 languages. We see that the syllable transduction method is able to produce good speech recognition performance. In general, transduced lattices do give a higher WER than the original word based lattices, but they help spot OOV words.

	WER	S2W ER
Assamese	68.8	72.3
Bengali	66.5	72.3
Creole	64.0	70.1
Zulu	72.8	78.0
Tamil	78.9	80.9

Table 4. WER with Lattice Transduction

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

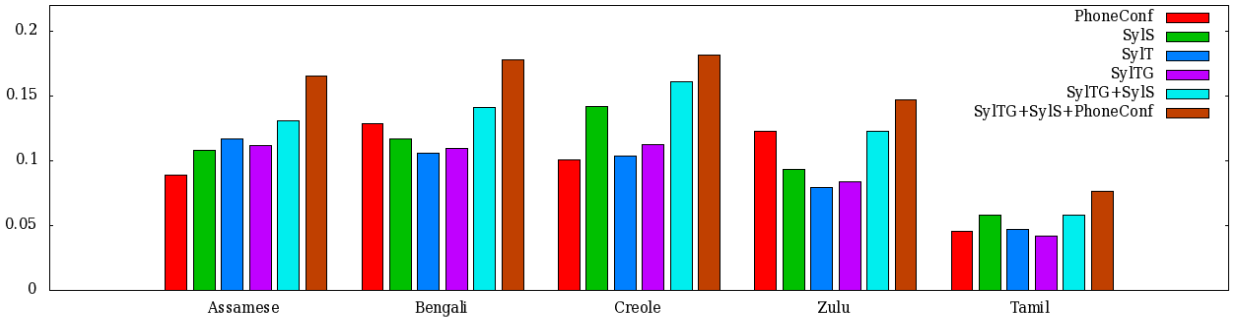

Fig. 1. OOV ATWV for different languages

5.2. 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), and a combination of these methods improves the overall performances.

To get a feeling for the characteristics of these three methods, we present miss rate and false alarm rate on Assamese in Table 5. 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 5. PMiss and PFA for Assamese

Although the primary focus of this paper is on OOV KWS, IV ATWV is also an important metric. 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 ATWV, indicating that the transduction method is effective in spotting both IV and OOV keywords.

	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

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 [26]. 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 by a greatly.

7. CONCLUSION

We show that syllable transduction is an effective way of dealing with OOV issue for KWS. It does not require modification on KWS template and is good at spotting both IV and OOV keywords. Analysis on miss rate and false alarm is presented, along with KWS results on 5 different languages. Experiments show that it is complimentary to syllable search and phone confusion method, and they can give much better OOV ATWV in combination.

8. ACKNOWLEDGEMENTS

We thank Brian Hutchinson, Aaron Jaech for providing the boosted language model, and Steven Wegmann, Guoguo Chen for their help on KWS and the phone confusion method.

9. REFERENCES

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

[2] J. Fiscus, J. Ajot, J. Garofolo, and G. Doddingtion, "Results of the 2006 spoken term detection evaluation," in *Proc. SIGIR*, 2007.

[3] C. Liu, A. Jansen, G. Chen, K. Kintzley, J. Trmal, and S. Khudanpur, "Low-resource open vocabulary keyword search using point process models," in *Proc. Interspeech*, 2014.

[4] D. Xu and F. Metze, "Word-based probabilistic phonetic retrieval for low-resource spoken term detection," in *Proc. Interspeech*, 2014.

[5] L. Mangu, B. Kingsbury, H. Soltau, H. Kuo, and M. Picheny, "Efficient spoken term detection using confusion networks," in *Proc. ICASSP*, 2014.

[6] G. Chen, O. Yilmaz, J. Trmal, D. Povey, and S. Khudanpur, "Using proxies for oov keywords in the keyword search task," in *Proc. ASRU*, 2013.

[7] Y. Li, W. Lo, H. Meng, and P. 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.

[8] M. Saraclar, A. Sethy, B. Ramabhadran, L. Mangu, J. Cui, X. Cui, B. Kingsbury, and J. Mamou, "An empirical study of confusion modeling in keyword search for low resource languages," in *Proc. ASRU*, 2013.

[9] O. Bacchiani and M. Siohan, "Fast vocabulary-independent audio search using path-based graph indexing," in *Proc. Interspeech*, 2005.

[10] Y. He, B. Hutchinson, P. Baumann, M. Ostendorf, E. Fosler-Lussier, and J. Pierrehumbert, "Subword-based modeling for handling oov words inkeyword spotting," in *Proc. ICASSP*, 2014.

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

[12] D. Karakos and R. Schwartz, "Subword and phonetic search for detecting out-of-vocabulary keywords," in *Proc. Interspeech*, 2014.

[13] D. Karakos, I. Bulyko, R. Schwartz, S. Tsakalidis, L. Nguyen, and J. Makhoul, "Normalization of phonetic keyword search scores," in *Proc. ICASSP*, 2014.

[14] M. Saraclar and R. Sproat, "Lattice-based search for spoken utterance retrieval," in *Proc. HLT-NAACL*, 2004.

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

[16] M. Mohri, F. Pereira, and M. Riley, "Speech recognition with weighted finite-state transducers," 2008.

[17] J. Novak, N. Minematsu, and K. 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.

[18] M. Redford and R. Diehl, "The relative perceptual distinctiveness of initial and final consonants in cvc syllables," in *The Journal of the Acoustical Society of America*. 1999, Acoustical Society of America.

[19] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," in *Speech Communication*, 2008.

[20] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, et al., "The kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[21] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *Proc. ICASSP*, 2014.

[22] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *Proc. ICASSP*, 2014.

[23] S. Wegmann, A. Faria, A. Janin, K. Riedhammer, and N. Morgan, "The tao of atwv: Probing the mysteries of keyword search performance," in *Proc. ASRU*, 2013.

[24] J. Lagarias, J. Reeds, M. Wright, and P. Wright, "Convergence properties of the nelder–mead simplex method in low dimensions," in *SIAM Journal on optimization*, 1998.

[25] D. Karakos, R. Schwartz, S. Tsakalidis, L. Zhang, S. Ranjan, et al., "Score normalization and system combination for improved keyword spotting," in *Proc. ASRU*, 2013.

[26] H. Su, J. Hieronymus, Y. He, E. Fosler-Lussier, and S. Wegmann, "Syllable based keyword search: Transducing syllable lattices to word lattices," in *Spoken Language Technology Workshop*, 2014.