SYLLABLE BASED KEYWORD SEARCH: TRANSDUCING SYLLABLE LATTICES TO WORD LATTICES

Hang Su[1,2], James Hieronymus[1], Yanzhang He[3], Eric Fosler-Lussier[3], Steven Wegmann[1]

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

ABSTRACT

This paper presents a weighted finite state transducer (WFST) based syllable decoding and transduction framework for keyword search (KWS). Acoustic context dependent phone models are trained from word forced alignments. Then syllable decoding is done with lattices generated using a syllable lexicon and language model (LM). To process out-of-vocabulary (OOV) keywords, pronunciations are produced using a grapheme-to-syllable (G2S) system. A syllable to word lexical transducer containing both in-vocabulary (IV) and OOV keywords is then constructed and composed with a keyword-boosted LM transducer. The composed transducer is then used to transduce syllable lattices to word lattices for final KWS. We show that our method can effectively perform KWS on both IV and OOV keywords, and yields up to 0.03 Actual Term-Weighted Value (ATWV) improvement over searching keywords directly in subword lattices. Word Error Rates (WER) and KWS results are reported for three different languages.

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

1. INTRODUCTION

Multilingual speech recognition and keyword search presents a unique set of challenges, including novel speech sounds, agglomerative morphology which causes very large vocabularies, and lack of transcribed data for training. 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 languages provided so far the numbers of OOV keywords are from 10% to 25%, so that just spotting IV keywords has resulted in reasonable performance in ATWV [2]. However, a higher OOV rate could be expected in such low-resource conditions, so a method for finding these keywords needs to be developed.

One way to mitigate OOV issue is to find 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].

Our method is to use syllables as the decoding subword unit, generating syllable lattices first. 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 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].

Previous work by Liu, Hieronymus, Gale and Woodland [14] used syllable recognition and syllable to word transduc-

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.

tion to improve speech recognition performance for Mandarin Chinese in the DARPA GALE project. GALE project provided 1,673 hours of training data so that there were very few OOV syllables. Collecting billions of words of text allowed very good syllable and word LM's to be trained. OOV words are in Chinese characters, so no G2S system was needed.

For our present work, the LM training data is limited to the transcribed data provided by the Babel project. This means that both the syllable and word LM's are weak because they are trained on a very small amount of data. In the word transduction stage, OOV keywords are added to the LM as unigrams. Their probabilities can be boosted to make them appear more likely in the LM. We evaluate our method on three languages from the Babel corpus, Haitian Creole, Zulu and Tamil, and report both recognition WER and ATWV.

The organization of the paper is as follows. Section 2 describes the WFST framework for syllable recognition and lattice transduction, Section 3 discuss construction of syllable to word lexical transducer. Section 4 discusses the experimental setup and recognition results. Section 5 gives the results of KWS experiments on the Babel languages. Section 6 concludes.

2. SYLLABLE DECODING AND TRANSDUCTION

Acoustic models for syllable decoding are trained on forced word alignments, and syllables are recognized using the syllable pronunciation dictionary and LM. Our intuition is that forced word alignments create more accurate acoustic models than expanding the words to syllables and training acoustic models based on them. Using common acoustic models allows us to compare word and syllable recognition and requires training acoustic models only once. Transducing syllable lattices to word lattices reduces lattice size by a large amount, making it possible to perform fast and accurate search.

In the Babel languages the average number of syllables in the training data ranges from 1300 to 2,600 while the number of vocabulary words varies from 5,000 to 15,000. Performing syllable decoding means that the number of decoding units is relatively stable across all the languages, even as the word vocabulary size changes greatly.

2.1. Syllable Decoding

A typical WFST decoding framework in speech recognition [15] 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; $\circ$ represents the composition operator. To perform a syllable decoding, substituting word transducers to syllable transducers gives

$$H \circ C \circ L_{phn2syl} \circ G_{syl} \tag{2}$$

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 word to syllable lexicon. As words may have several pronunciations, randomly picking might be needed. 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 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 that of words, making training data relatively sufficient.

2.2. 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.

2.3. Keyword 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.

$$\hat{Lat}_{syl} \circ L_{syl2wrd} \circ G_{boost} \tag{3}$$

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.

3. HANDLING OOVS VIA G2S

In the BABEL project, 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 add in OOV pronunciation 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 described as follows.

3.1. Grapheme to Syllable Prediction

Our pronunciation prediction utilizes the Phonetisaurus G2P system [16] 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.

3.2. Mapping OOV Syllables

This G2S system described above can produce syllables that have not been seen before. 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 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 [13]. As a first step, we only selected one IV syllable per OOV syllable.

4. RECOGNITION EXPERIMENTS

BABEL data in Table 1 are used for experiments. Each language pack provides a limited amount of training data which has been transcribed at the word level and lexicons for words and syllables. The data 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. Development data sets are provided for performance testing and parameter tuning. In this work, we used the LLP for training and dev set for evaluation.

	version
Creole	IARPA-babel201b-v0.2b
Zulu	IARPA-babel206b-v0.1e
Tamil	IARPA-babel204b-v1.1b

Table 1. Babel data for different languages

4.1. Setup

The Kaldi toolkit [17] is used for the speech recognition part. Standard 13-dim PLP feature, together with 3-dim Kaldi pitch feature [18], 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 feature. Details of DNN training are documented in section 2.2 in [19]. 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.

4.2. Syllable Decoding

To evaluate the performance of syllable decoding, we need to map transcriptions from word to syllables. Just as described in 2.1, we force align transcriptions with the acoustic training data to reach a more accurate result.

Table 2 shows WERs of word decoding and Syllable Error Rates (SERs) of syllable decoding. These two metrics are not comparable in general – we present them here just for a quick reference. Table 3 shows the number of words and syllables in decoding.

4.3. G2S Prediction

G2S prediction is evaluated by comparing ground truth lexicon with predicted lexicon. BABEL language pack provides

	WER	SER
Creole	61.6	67.1
Zulu	71.5	73.6
Tamil	79.0	77.9

Table 2. Syllable Error Rates

	# of Words	# OOV KW	# of Syllables
Creole	4897	884	1981
Zulu	13674	1109	1345
Tamil	14265	1449	2620

Table 3. Vocabulary statistics

a lexicon that covers more words 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 4 shows the phone error rate (PER) and syllable error rate (SER) on that 'OOV' set (i.e. FLP v.s. LLP). Note that the SER concept in this section is different from that in previous section.

	PER	SER
Creole	5.7	31.5
Zulu	5.9	11.9
Tamil	2.2	7.6

Table 4. Pronunciation Prediction Error Rate

We could see from Table 4 that SERs for Creole and Zulu 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.

4.4. Lattice Transduction Recognition Performance

WER is used as a validation metric for lattice transduction. In this part, we do not use keyword boosted language model as it is designed for KWS task rather than recognition. Table 5 shows WERs for baseline word decoding and syllable transduction. It can be observed that 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. KWS EXPERIMENTS

We use `babel201b-v0.2b_conv-eval.kwlist4.xml` (Creole), `babel206b-v0.1e_conv-eval.kwlist4.xml` (Zulu), `babel204b-v1.1b_conv-eval.kwlist5.xml` (Tamil) for the KWS task. These keyword lists are provided by NIST for the IARPA Babel project. IV/OOV keywords statistics are reported in Table 6. Note that OOV keywords in this table are actually OOV keywords that appear in the dev set.

	Word	Syl2Wrd
Creole	61.6	67.1
Zulu	71.5	77.7
Tamil	79.0	81.0

Table 5. WER with Lattice Transduction

KWS experiments generally follows the method described in [20].

	IV keywords	OOV keywords	Total
Creole	1307 (91.3%)	124 (8.7%)	1431
Zulu	1066 (77.5%)	310 (23.5%)	1376
Tamil	1211 (82.0%)	266 (18.0%)	1477

Table 6. Keywords statistics relative to dev set

5.1. Direct Search in Syllable Lattices

Direct search of OOV keywords in subword lattices serves as a baseline method in this paper. 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.

5.2. Search Results

Figure 1 reports IV ATWV for all three languages using different decoding/searching methods. Word, Sylsearch, Syl2word and Syl2wordG denote word baseline, searching keywords in syllable lattices, transducing syllable lattices to word lattices and transducing with keyword boosted LM respectively [1]. It shows that lattice transduction works better than syllable search in IV set, and boosted language model further improves search performance.

Figure 2 shows OOV ATWV in different settings. Note that word decoding in NTAR condition gives no OOV hits. It is shown that in Tamil and Zulu, syllable transduction gives slightly lower OOV ATWV than syllable search, but reaches a much higher score in Haitian Creole. Detailed search statistics shows that the syllable transduction method always yield fewer hypotheses with better posteriors than direct search in syllable lattices, resulting in fewer correct hits and false alarms. This fact might indicate advantage over syllable

[1] As ATWV is highly influenced by lattice size, to ensure a fair comparison, we control lattice size so that baseline word lattice are of similar size as transduced syl2wrd lattices, and syllable lattices used for direct search are the same as those for transduction.

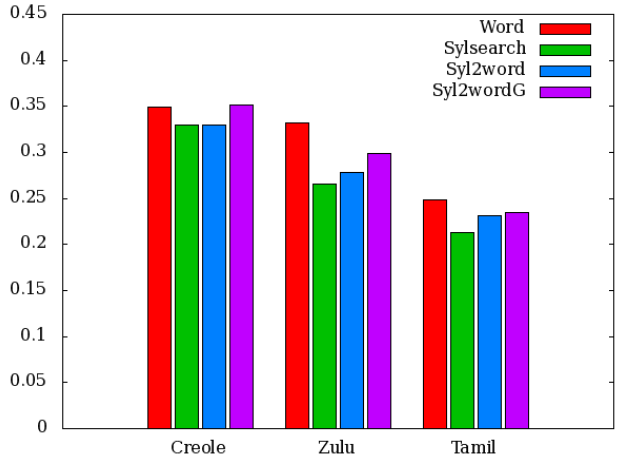

Fig. 1. IV ATWV

search in system combination condition, as combination itself tend to favor accurate posterior and compensate for lack of hypotheses.

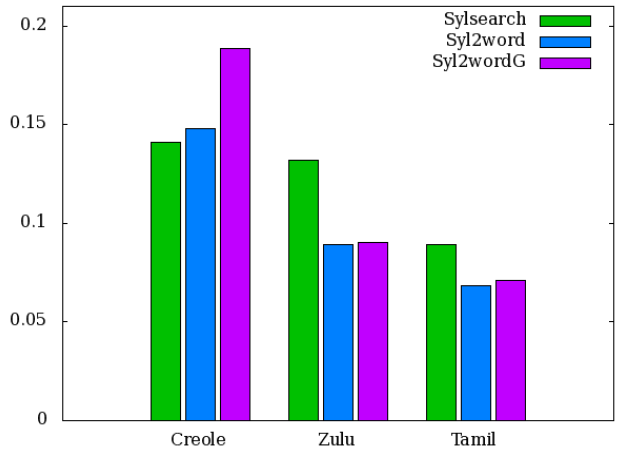

Fig. 2. OOV ATWV

Combining IV and OOV ATWV we arrive at Figure 3 for overall ATWV. In addition, we present two system comination results: Sylsearch+Word denotes combination of word baseline and syllable search, Syl2wordG+Word denotes combination of word baseline and syllable transduction with boosted language model. Our combination strategy follows KST normalization present in [21]. This figure shows that syllable transduction method generally outperforms syllable search, yielding up to 0.03 ATWV increase in Haitian Creole, and system combinations for syllable search and transduction both helps.

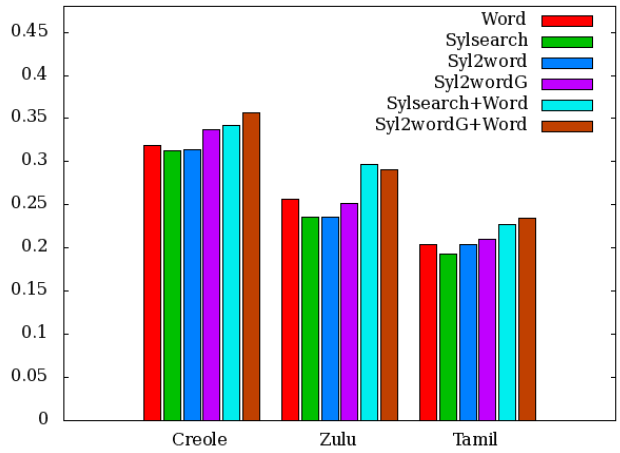

Fig. 3. Overall ATWV

6. CONCLUSION AND FUTURE WORK

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.

7. ACKNOWLEDGEMENTS

We gratefully thank Brian Hutchinson, Peter Baumann and Aaron Jaech for their help on boosted language model. We also thank Korbinian Riedhammer for his generous help on syllable decoding and Daniel Povey for his enlightening advice on lattice alignment.

8. REFERENCES

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

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

[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] Xunying Liu, James L Hieronymus, Mark JF Gales, and Philip C Woodland, "Syllable language models for mandarin speech recognition: Exploiting character language models," *The Journal of the Acoustical Society of America*, 2013.

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

[16] 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.

[17] 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.

[18] 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.

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

[20] Steven Wegmann, Arlo Faria, Adam Janin, Korbinian Riedhammer, and Nelson Morgan, "The tao of atwv: Probing the mysteries of keyword search performance," in *ASRU*, 2013.

[21] Damianos Karakos, Richard Schwartz, Stavros Tsakalidis, Le Zhang, Shivesh Ranjan, Tim Ng, Roger Hsiao, Guruprasad Saikumar, Ivan Bulyko, Long Nguyen, et al., "Score normalization and system combination for improved keyword spotting," in *ASRU*, 2013.