OUT-OF-VOCABULARY WORD RECOVERY USING FST-BASED SUBWORD UNIT CLUSTERING IN A HYBRID ASR SYSTEM

Ekaterina Egorova, Lukáš Burget

Brno University of Technology
{iegorova, burget}@fit.vutbr.cz

ABSTRACT

The paper presents a new approach to extracting useful information from out-of-vocabulary (OOV) speech regions in ASR system output. The system makes use of a hybrid decoding network with both words and sub-word units. In the decoded lattices, candidates for OOV regions are identified as sub-graphs of sub-word units. To facilitate OOV word recovery, we search for recurring OOVs by clustering the detected candidate OOVs. The metrics for clustering is based on a comparison of the sub-graphs corresponding to the OOV candidates. The proposed method discovers repeating out-of-vocabulary words and finds their graphemic representation more robustly than more conventional techniques taking into account only one best sub-word string hypotheses.

Index Terms— Out-of-vocabulary Words, Robust ASR

1. INTRODUCTION

Human speech is by nature non-finite: new words are constantly emerging, and it is therefore impossible to describe a language fully. Words which are not accounted for in the language model (LM) are called out-of-vocabulary (OOV) words, and they constitute one of the biggest challenges in ASR and other speech processing tasks. The problem is that if a word is not in the dictionary and language model, the system cannot output it during the decoding. Instead, the system will try to find the (acoustically) closest in-vocabulary (IV) word, often confusing the end user and interfering with the proper decoding of the words around it. The problem gets even more interesting if one considers that OOVs are usually topic-specific words or proper names, meaning they are often key words important for proper understanding of the text.

OOV research is most prominent within the frameworks of speech recognition and key word spotting (KWS). While in KWS the graphical representation of the query is known [1, 2, 3], in ASR task OOVs are completely unseen and thus have to be discovered and modeled without any knowledge of what they are. This is the task that we tackle in this paper. One of the most common ways to deal with OOVs in the framework of ASR is to use lattices of sub-word units, which can be either linguistically motivated (phonemes, syllables, etc.) or data driven [4]. For efficiency the system can first work on a word level and get to a sub-word level only in case the output does not fit pre-set conditions (e.g. minimum confidence score etc.) [5]. A successful method of discovering and learning OOV words exploits a hybrid system which combines word and sub-word units in its language model [6, 7, 8, 9]. In this case if the system output is, in some region, a string of sub-word units instead of a string of words, it means that an OOV word is discovered. A variation upon this is a flat hybrid sub-lexical model that substitutes rare words with their phoneme strings to obtain training data for the language model, thus combining word and sub-word levels into a single hybrid LM [10].

The main focus of this article will be on what happens after a successful discovery of OOV words. If they are discovered in the form of strings of sub-word units, a free clustering of these hypothesized strings can be performed in order to discover recurring sequences and to add them to the dictionary as new words [11, 12, 13]. The clustering criteria may include phonetic and acoustic features and context information [14]. Unlike previous approaches, the work covered in this article attempts to automatically discover new words in a decoding lattice rather than on one-best hypothesis. A sub-word decoding lattice may contain paths that correspond to slightly different pronunciations. Thus, using clustering lattices instead of one-best output strings allows us to discover OOV patterns even if the same OOV is pronounced somewhat differently on different occasions. Moreover, this approach should be more robust in the case of an ASR output of low quality.

2. OOV RECOVERY PROCEDURE

2.1. OOV Detection with Hybrid Decoding Graph

Weighted Finite-State Transducer (WFST) based decoders [15] limit the decoding search space by constructing a decoding graph out of different knowledge sources. In Kaldi setup

The work was supported by European UnionâĂŹs Horizon 2020 project No. 645523 BISON, Technology Agency of the Czech Republic project No. TJ01000208 "NOSICI", and by Czech Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science - LQ1602".

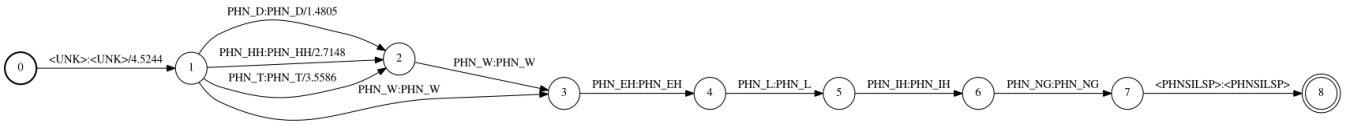

Fig. 1. OOV candidate in lattice form

[16], the decoding graph is a composition $H \circ C \circ L \circ G$, which combines different components of the ASR model into a huge Hidden Markov Model (HMM), on which decoding is performed. It represents search space constraints on each of the levels in the system: G is a weighted acceptor that encodes language model, L is the lexicon, C represents the context-dependency and H contains the HMM definitions of acoustic units. Thus, a path through the decoding graph encodes a mapping from a string of input symbols (encodings of acoustic units) to a string of output symbols (words). The weights on the links constituting a path in the HCLG graph are combined probabilities from the word LM, phoneme LM, and HMM transition probabilities, while acoustic model probabilities (HMM emission probabilities) are calculated during decoding using the PDFs corresponding to HCLG's input symbols. The HCLG graph thus defines transitions and their probabilities of the decoding HMM, while the input symbols of the graph identify HMM state output distributions. For more information on WFSTs for ASR, see [15].

The issue with OOV words in such a framework is that although we can have OOVs represented by a special word in the LM, there is no dictionary entry that specifies what string of acoustic units represents each particular OOV. Phonetically OOVs are often modeled with a garbage acoustic model. While practical, such a solution does not provide good acoustic representation of an OOV, so in our hybrid system we model the likely phonetic sequences representing an OOV by n-gram subword LM trained on the lexicon. Our hybrid system is built according to [17] with phonemes chosen as subword acoustic entities. The phoneme FSTs representing OOVs in the hybrid G graph allow an OOV to be correctly modeled as the combination of the probability of the OOV being in the utterance that is obtained from the language model and the probability of the OOV being realized as a specific phoneme path. In the hybrid graph, we can manually control the preference that the system shows towards paths containing phonemes. This is achieved by boosting the probability of following the OOV link and by increasing or decreasing probabilities inside of the phoneme sub-graph to encourage choosing paths through OOV [17].

2.2. OOV Extraction from Lattices

In order to find recurring OOVs, we decode the data with the hybrid graph described above, which gives us hybrid decoded lattices containing both paths with words and paths with phonemes representing OOVs. First, we have to extract OOV candidates in form of phoneme sub-lattices from our decoded lattices, i.e. separate sub-lattices of phonemes from the rest of the lattices containing words that will not be used in further information extraction. We will search for sub-lattices starting with an OOV symbol that marks the entry point of phoneme lattice and ending with <phnsilsp> output symbol which marks the exit point. It is memory and time consuming to do so as we have to traverse lattices forward and backward to ensure that all phoneme paths are found.

We can speed up the process of OOV extraction if we first apply indexing to decoded lattices. In our experiments, we used reverse indexing proposed in [18] for the task of key word spotting. The output of this indexing procedure is a tree-like WFST which contains all partial paths through the lattice, of which we care only about subword ones. On the output there are paths' weights and start/end timings. A tree-like structure is fast to traverse, and after the OOV candidates are extracted , the minimization operation will return them to the structure they had in the decoded lattices (see Fig. 1).

After they are extracted from the index, OOV candidates are represented as probabilistic phoneme lattices that now contain both the decoding graph probabilities and acoustic scores. The weight of every path of an OOV candidate lattice can be transformed to the posterior probability via weight-pushing in log semiring [15]. Lattices are better as OOV candidates than one-best paths, as they can encode the uncertainty in the OOV's acoustic realization. We discover lattices that represent the same OOV with the help of clustering.

2.3. OOV Candidates Clustering

As the metric which decides if two OOV candidates belong to the same word, we introduce composition score, which is calculated as following: after we perform composition of two lattices, the composition score is zero if the resulting lattice is empty. This means that there are no common paths between these two lattices. If the composition output is not empty, though, it means that there is at least one common path in these two lattices. The composition score is then the probability of the shortest path of this composition, which can be interpreted as the probability of both OOV candidates being present in the recording and both being pronounced as the same sequence of phonemes.

In the beginning, pairwise composition scores of all the OOV candidates are put into a matrix. At each step of the clustering, the system looks for the biggest composition score and performs the union of the two corresponding OOV candidates. The newly united lattice preserves paths from all of the initial lattices that were merged into it at different steps.

At the end of a clustering step, pairwise composition scores involving the two merged candidates are recalculated by performing the composition of the newly merged candidates with all other candidates. The composition score is a natural stopping criterium for the clustering - if it is too small, we might not want to merge these candidates.

After the clustering, the system looks at the OOV candidates that have more than 2 occurrences based on the clustering. These candidates are represented by the union of all the phoneme paths in all these occurrences. Now further actions can be taken to recover the words and add them to the system. For example, a phoneme-to-grapheme (p2g) model trained on the dictionary [19] can be applied to the best path or to the whole lattice of a candidate to propose new entries to the dictionary.

3. EXPERIMENTAL SETUP

3.1. Data and System Description

For the experiments, we have selected LibriSpeech ASR corpus of audiobooks[1] for its size and quality. The language model provided with LibriSpeech dataset is a 3-gram ARPA LM trained on 14500 public domain books. The dictionary contains 200000 words. The phonotactic language model is also a 3-gram ARPA LM trained on the dictionary. The system is trained using Kaldi toolkit [16], slightly altered to accommodate for hybrid training. From LibriSpeech, 100 hours of clean data are used for system training, and a separate test set of 5 hours and 20 minutes for system performance evaluation. There is no speaker overlap between test and train data. Further discovery of OOVs is done on a bigger dataset of 360 hours. This gives enough data to find repeating OOV patterns and learn their acoustic representation.

The sequence of systems in Kaldi baseline (nnet3 recipe) is: (i) HMM/GMM monophone system, (ii) triphone system with MFCC + Δ + $\Delta\Delta$ features, (iii) triphone system with LDA+MLLT, (iv) triphone system with LDA+MLLT+SAT (v) DNNs on top of the fMLLR features, using the decision tree and state alignments from the LDA+MLLT+SAT system as supervision for training. The WER that is possible to obtain with this setting is 11.61%. Note that it is more than was reported by [20], but this baseline is already for the system with artificially chosen and excluded OOVs at 1.5% OOV rate.

3.2. OOV Simulation on LibriSpeech Dataset

In a real-world scenario, OOVs would be newly-coined words and names, but in audiobooks, this is not a viable setup. The corpus majorly consists of free domain books, which are predominantly from 19th century. One of the possible simulations of the real-world scenario is to "reverse" the task and designate archaic and out-of usage words as OOV words. These words are not likely to be in a modern LM trained on Internet data.

In order to choose OOV words, we used Google ngram dataset of word usage statistics in books [21]. For each word, the database provides its number of occurrences in sources published each year over the last 5 centuries. By normalizing this number by the total number of words in this year's publications, the relative frequency of this word in this year is obtained.

For our purposes, words with twice as much frequency before year 1900 than after 1900 are chosen as OOVs. Moreover, all names are also added to the OOV list. If the word is a name can be checked by the relative number of its occurrences in the ngram with a capital letter and without.

The resulting list of OOVs picked as described above consists of 1000 designated OOVs, which present an example of 19-century bookish English. For example, it includes such words as INTERPOSED, HASTENED, MADEMOISELLE, INDIGNANTLY, COUNTENANCE, etc. The words' frequency in the 360 hours dataset ranges from 0 to 296 reference occurrences, with the mean of 51 occurrences.

With the OOV list obtained as a result of this method, the OOV rate (percentage of OOVs in all the words) reaches 1.5%.

4. DISCUSSION OF THE RESULTS

4.1. OOV Detection Results

The baseline system as described in Section 3 achieves 11.61% WER and zero f-score (as proposed in [13]) if estimated on one-best output on the 5.3 hour test set. This means it is not capable of detecting OOV candidates at all. In a hybrid system, hybrid parameters described in [17] can be changed to trade-off between detecting more OOV candidates and retaining a good WER. The following experiments are performed on a hybrid system with OOV cost = -10, phoneme LM scaling factor = 0.8 and phoneme insertion penalty = 0. This system provides an f-score of 1% at 11.77% WER on the test set.

When evaluating the OOV detection performance on full lattices in comparison to a one-best output, benefits of the full lattice approach can be observed. On 360 hours of data, extracting OOV candidates as phoneme strings from one-best decoding output results in just 1247 OOV candidates, while from the full lattices we get 15991, which is more than 12 times more.

4.2. Clustering Procedure Analysis

To evaluate the hierarchical clustering of OOV hypoteses and estimate the stopping point, we've looked at the clustering quality with Adjusted Rand Index, a standard way of analyzing comparisons between two clusterings [22]. At each step of the clustering procedure, we compared the current hypothesis clustering with the true clustering given by the word la-

[1] http://www.openslr.org/12/

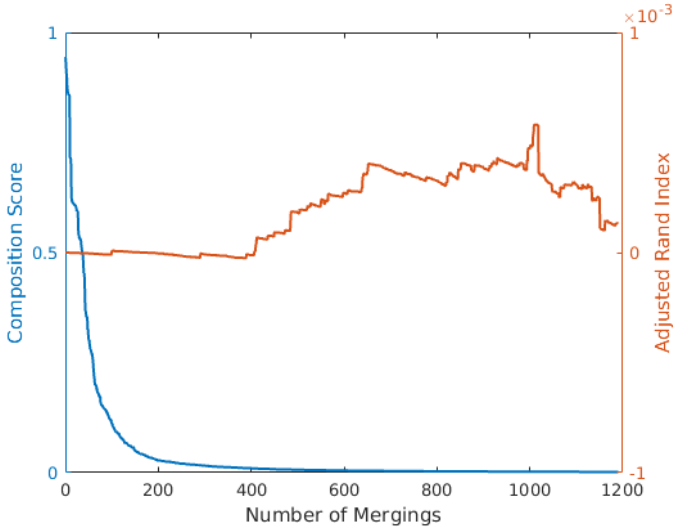

Fig. 2. Composition Score and Adjusted Rand Index Score in hierarchical clustering of OOV hypotheses.

bels obtained from alignment with a full dictionary. While ARI is small due to the huge number of initial clusters, it keeps growing while the composition score is big enough to create mergings that make sense and falls when the system starts over-merging clusters (See Fig.2).

4.3. OOV Recovery Results

Having proved that full lattices are better than one-best output for OOV detection, we will now prove the same for OOV recovery based on subword FST clustering.

Below are the OOVs that are obtained from the clustering with composition score threshold 0.01 of phoneme strings from one-best decoding output results. The corresponding graphemic representation is obtained from a p2g system trained on the same dictionary as the phonotactic LM [19]. Only the OOVs that are a result of clustering of more than 2 candidates are considered, which produces the following 7 recovered words:

COURAGE	K ER IH JH
VOYAGE	V OY IH JH
FLANE	F L EY N
KLEY'S	K L IY Z
SALOOSKI	S AH L UW S K IY
IMETHEUS	IH M IY TH IY AH S
ANCTIOUSLY	AE NG K SH AH S L IY

As can be seen, less than half of the words make sense. Only 2 of the OOV words out of 1000 are recovered correctly and one (ANCTIOUSLY) is close enough to be recognizable. This gives us a recovery rate of 0.3%. To compare, below are the 20 OOVs that are obtained with clustering candidates obtained from full decoding lattices with 0.01 threshold. In the brackets is the number of the word's reference occurrences where applicable. Again, only OOVs that are the result of merging more than 2 candidates are considered:

COURAGE (288)	K ER IH JH
VOYAGE (120)	V OY IH JH
THRACE	TH R EY S
THRONG (48)	TH R AO NG
SNESS	S N AH S
UNESE	AH N IY Z
ATHO'S	AE TH OW Z
HITHER (95)	HH IH DH ER
IGARLY (176)	IH G ER L IY
SAVAGE (182)	S AE V IH JH
WELLING (82)	W EH L IH NG
ANXIOUS (296)	AE NG K SH AH S
BOLDLY (68)	B OW L D L IY
TRICHERY (43)	T R IH CH ER IY
DIGNITY (190)	D IH G N AH T IY
ERLOGINGS	ER L AA JH IH NG Z
ERNESSNESS	ER N AH S N AH S
ANCTIOUSLY (99)	AE NG K SH AH S L IY
CORMALIS	K AO R M AE L AH S
HITHERINTHITHER	HH IH DH ER IH N TH IH DH ER

Of these 20, 8 are ideally recovered words from the 1000 on the OOV list, which is four times as many as with one-best approach. Furthermore, there are some close-to-ideal recoveries, like a name from *The Three Musketeers*, and 6 words that are still recognizable, although the graphemic representation is not completely right. So the OOV recovery rate in full-lattice clustering equals 1.4%, which is more than 4 times better than one-best clustering. Adding these newly-discovered OOVs to the dictionary with the learned pronunciation and to the LM as unigrams with the same probability as an OOV reduces WER from 11.77% to 11.62%.

Of special interest are entries "SNESS" and "HITHERINTHITHER". The first is a suffix, which can help with the recognition of nouns that are derived from adjectives using this morpheme. The second is a phrase "hither and thither", which repeats itself more often in the data than any of its parts separately. As this phrase has a distinct meaning and usage, it may be profitable to treat it as an individual lexical entity in a language model.

5. CONCLUSION

This article has explored the possibilities of OOV recovery through fst-based sub-word unit clustering. It has been shown that the newly-proposed lattice-based approach outperforms one-best approaches both in terms of OOV detection and in terms of the recovery of phonetic and graphemic representations of OOV words. The proposed system shows promise of enhancing ASR user experience by bringing to her attention newly discovered words that may be added to the dictionary almost without adjustments.

6. REFERENCES

[1] S. Novotney, I. Bulyko, R. Schwartz, S. Khudanpur, and O. Kimball, "Semi-Supervised Methods for Improving Keyword Search of Unseen Terms," *INTERSPEECH*, 2012.

[2] D. Karakos and R. Schwartz, "Subword and Phonetic Search for Detecting Out-of-Vocabulary Keywords," *INTERSPEECH*, 2014.

[3] D. Karakos and R. Schwartz, "Combination of Search Techniques for Improved Spotting of OOV Keywords," *ICASSP*, 2015.

[4] W. Hartmann, A. Roy, L. Lamel, and J.-L. Gauvain, "Acoustic Unit Discovery and Pronunciation Generation From a Grapheme-based Lexicon," *Proceedings of ASRU 2013*, 2013.

[5] L. Lee, J. Glass, H. Lee, and C. Chan, "Spoken Content Retrieval - Beyond Cascading Speech Recognition with Text Retrieval," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2015.

[6] A. Yazgan and M. Saraclar, "Hybrid language models for out of vocabulary word detection on large vocabulary conversational speech recognition," *ICASSP*, 2004.

[7] A. Rastrow, A. Sethy, and B. Ramabhadran, "A New Method for OOV Detection Using Hybrid Word/Fragment System," *ICASSP*, 2009.

[8] L. Qin and A. Rudnicky, "Finding Recurrent Out-of-Vocabulary Words," *Proceedings of INTERSPEECH 2013*, 2013.

[9] S. Kombrink, M. Hannemann, and L. Burget, *Out-of-Vocabulary Word Detection and Beyond*, pp. 57–65, Studies in Computational Intelligence, 384. 2012.

[10] M. Bisani and H. Ney, "Open vocabulary speech recognition with flat hybrid models," *INTERSPEECH*, 2005.

[11] L. Qin and A. Rudnicky, "Learning Better Lexical Properties for Recurrent OOV Words," *Proceedings of ASRU 2013*, 2013.

[12] M. Hannemann, S. Kombrink, M. Karafiát, and L. Burget, "Similarity scoring for recognizing repeated out-of-vocabularywords," in *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)*, 2010, vol. 2010, pp. 897–900.

[13] S. Kombrink, M. Hannemann, L. Burget, and H. Heřmanský, "Recovery of rare words in lecture speech," in *Proc. Text, Speech and Dialogue 2010*, 2010, vol. 2010, pp. 330–337.

[14] L. Qin and A. Rudnicky, "Finding Recurrent Out-of-Vocabulary Words," *INTERSPEECH*, 2013.

[15] M. Mohri, F. Pereira, and M. Riley, "Speech Recognition with Weighted Finite-State Transducers," *Handbook on Speech Processing and Speech Communication, Part E: Speech recognition*, 2008.

[16] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, Petr Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The Kaldi Speech Recognition Toolkit," *Proceedings of ASRU 2011 Conference*, 2011.

[17] I. Szőke, *Hybrid word-subword spoken term detection*, Ph.D. thesis, 2010.

[18] D. Can and M. Saraçlar, "Lattice Indexing for Spoken Term Detection," *IEEE Transactions on Audio, Speech, and Language Processing*, 2011.

[19] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication 50*, pp. 434–451, 2008.

[20] V. Panayotov, G. Chen, Povey D, and S. Khudanpur, "LibriSpeech: an ASR corpus based on public domain audio books," *ICASSP*, 2015.

[21] Y. Lin, J.-B. Michel, E.L. Aiden, J. Orwant, W. Brockman, and S. Petrov, "Syntactic Annotations for the Google Books Ngram Corpus," *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pp. 169–174, 2012.

[22] J.M. Santos and M. Embrechts, "On the Use of the Adjusted Rand Index as a Metric for Evaluating Supervised Classification," *ICANN 2009. Lecture Notes in Computer Science*, 2009.