OUT-OF-VOCABULARY WORD DETECTION IN A SPEECH-TO-SPEECH TRANSLATION SYSTEM

Hong-Kwang Kuo, Ellen Eide Kislal, Lidia Mangu, Hagen Soltau, Tomas Beran

IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA

ABSTRACT

In this paper we describe progress we have made in detecting out-of-vocabulary words (OOVs) for a speech-to-speech translation system for the purpose of playing back audio to the user for clarification and correction. Our OOV detector follows a strategy of first identifying a rough location of the OOV and then merging adjacent decoded words to cover the true OOV word. We show the advantage of our OOV detection strategy and report on improvements using a real-time implementation of a new Convolutional Neural Network acoustic model. We discuss why commonly used metrics for OOV detection do not meet our needs and explore an overlap metric as well as a Jaccard metric for evaluating our ability to detect the OOVs and localize them accurately in time. We have found different metrics to be useful at different stages of development.

Index Terms— OOV, metric, speech-to-speech translation system

1. INTRODUCTION

Using speech-to-speech translation systems, two speakers of different languages can use a computer to try and communicate with each other. Such systems typically include component technologies such as automatic speech recognition (ASR), machine translation (MT), and text-to-speech (TTS). More recently there is new interest in incorporating an intelligent agent or dialogue manager to detect errors or ambiguity, e.g. ASR errors, homonyms, word sense, etc., and engage in a dialogue with the user to correct the errors [1, 2, 3, 4, 5].

An important type of ASR error concerns out-of-vocabulary (OOV) words, which are often content words that are important to the conversation. One possible strategy is to detect the ASR errors and ask the user to correct them using a clarification dialogue such as a "Paraphrase or Spell" module. Once the errors are resolved, the utterance is translated and the resulting utterance is synthesized in the target language. The "Paraphrase or Spell" module may prompt the user with something like "I did not understand ⟨*audio corresponding to OOV word*⟩." It is important that the OOVs be detected as precisely as possible, i.e. with accurate time boundaries. If the audio playback includes words that are already correctly decoded by the ASR, it represents wasted effort to ask the user for correction. On the other hand, if the audio did not cover enough of the OOV word to be intelligible, the user will become confused and cannot correct the error. The act of asking for clarification is already likely to negatively impact the user's perception of the system quality [6], so we must proceed in such a way as to make this negativity have as little impact as possible.

This paper is concerned with pinpointing OOV words uttered during free dialogue with a speech-to-speech translation system in both source and target languages. Previous work in the area has often focused on detecting OOVs loosely, in the sense that a small overlap in time between the hypothesized and reference region of an OOV counts as a successfully detected OOV. In certain situations, however, such as a dialogue system with confirmation or error correction, especially when playing back audio snippets to the user, accurate boundary information is key because the ability to clearly indicate which portion of the user's original utterance is failing, rather than asking for a paraphrase of the entire utterance, is important for users to successfully make progress [3]. In light of that observation, we explore various metrics for evaluating our OOV detection framework and discuss the strengths and weaknesses of each.

In this paper, we report progress we have made in the OOV detection component of a two-way speech-to-speech translation system between English and Iraqi Arabic which we have built for the Phase 2 DARPA BOLT-BC Evaluation. In Section 2, we describe our OOV detector which follows a strategy of first identifying a rough location of the OOV and then merging adjacent decoded words to cover the true OOV word. Section 3 describes why commonly used metrics for OOV detection do not meet our needs and explores a few metrics for evaluating our system. We report in Section 4 a real-time implementation of an acoustic model based on Convolutional Neural Networks (CNN), which results in improved OOV detection. Section 5 briefly relates to prior work, and we finish with discussion and conclusions in Section 6.

2. OOV DETECTOR

Our OOV detector is based on a maximum entropy (maxent) classifier similar to that described in [7, 8]. The speech recognizer uses a hybrid language model that contains a vocabulary of both word and sub-word (fragment) units. The fragment units are variable length phone sequences that are intended to be filler models to absorb OOV words, especially when they are acoustically different from in-vocabulary words. They can be selected automatically using statistical methods [9].

The recognizer decodes a speech utterance to produce a confusion network structure [10] that compactly encodes the likely hypotheses and their posterior probabilities. An example is shown in Figure 1. Each confusion bin contains a set of competing hypotheses with their posterior probabilities (not shown.) The basic idea behind our OOV detector is that an OOV word does not match well with in-vocabulary words and is more likely to activate fragment hypotheses (e.g. IX_I_IY in the figure); in addition, there is likely to be more confusion in the bin, and the best hypothesis is likely to have lower posterior probability.

Following these intuitions, the following features can be extracted from each bin of the confusion network and used in a maxent model to detect OOVs:

$$\text{Fragment\_Posterior} = \sum_{f \in t_j} p(f|t_j), \tag{1}$$

where f are the fragments in the current confusion bin t_j,

$$\text{Entropy} = -\sum_{w \in t_j} p(w|t_j) \log p(w|t_j), \tag{2}$$

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

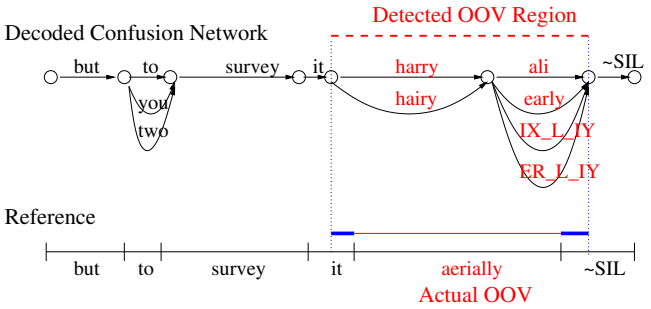

Fig. 1. Confusion network example.

where w is any word or fragment in the bin, and

$$\text{Posterior} = \max_{w \in t_j} p(w|t_j), \tag{3}$$

which is the posterior of the best hypothesis in the bin.

The time boundaries (start and end times) of each bin are expected times computed by weighting the times of the alternatives by the word posterior probabilities. Given time boundaries, another feature we can use is the duration of the bin. We do not use w, the identity of the word corresponding to the best hypothesis in the bin due to data sparseness, but we use the word frequency rank as a feature. All features were quantized using uniform-occupancy partitioning [11], and we used about ten partitions for most features. Features associated with neighboring words can also be used as context features. Empirically we found left context features to be useful.

We use a maxent classifier with three categories: OOV, in-vocabulary error (WErr), and in-vocabulary correctly decoded word (WCorr). To train the classifier, we use a development set containing OOV instances. The audio is decoded to produce confusion networks and also force aligned with reference transcripts. Each decoded word is labeled OOV, WErr, or WCorr based on the reference aligned word that overlaps the most with it.

The maxent classifier assigns category probabilities to each decoded word. However, our goal is to find the true OOV word and its time boundaries; often the true OOV word encompasses more than one decoded word. Our strategy is to find the decoded word that is most likely to be part of an OOV and to grow a region around that word. The merging is based on the category probabilities coming out of the maxent classifier and the durations of the word hypotheses. The merging algorithm strives to join contiguous words with high likelihood of representing OOV tokens into a single region, and agglomeratively joins additional neighbors, raising the threshold for merging as the duration of the merged segments increases. If desired, after the top hypothesized region has been found, we can iterate to find the next OOV word by considering the next most likely decoded word that does not overlap with previous region(s).

The OOV detector was trained on a relatively small development set of 575 sentences containing about 4500 words, with about 380 OOV words. The test set includes 493 sentences containing OOVs provided by SRI and 1138 sentences extracted from the BOLT test set which do not contain any OOVs. In total, the test set has about 19K words, with 501 OOV words.

3. METRICS

Recall that we need to cut out the audio containing the true OOV word in order to play back to the user to seek clarification/correction. The metrics commonly used to evaluate OOV detectors do not match our needs. In [12], an OOV instance is considered correctly detected if there is any (even tiny) overlap between the detected region and the true OOV word. In [7], frame level detection is considered. In [8], scoring is done at the level of individual decoded words. None of these metrics fit our needs. For example, detecting half of an OOV word would lead to a garbled play-back that can confuse the user. Consider the example in Figure 1. The metric used in [8] would give credit of 2 (for decoded words "harry" and "ali"); but we really want to give only a credit of at most 1 (for reference OOV word "aerially"). If only "ali" had been detected, we want to give it little or no credit because playing back that segment would confuse the user.

3.1. Overlap Metric

To address these issues, we define a metric based on the amount of overlap between the reference OOV word and the hypothesized OOV region. We will give credit only if the amount of overlap is 95% of the duration of the OOV word. Given a set of sentences s_n, in-vocabulary words $w_i \in I_{s_n}$, OOV words $w_j \in Q_{s_n}$, and hypothesized/predicted OOV words $h_k \in H_{s_n}$, the number of true positives is then defined to be:

$$\text{TP}_{\text{oov}} = \sum_{s_n} \sum_{w_j \in Q_{s_n}} \left\lceil \max_{h_k \in H_{s_n}} \frac{\text{overlap}(w_j, h_k)}{\text{len}(w_j)} \right\rceil_{0.95}, \tag{4}$$

where

$$\lceil x \rceil_T = \begin{cases} 1 & \text{if } x \geq T; \\ 0 & \text{if } x < T. \end{cases} \tag{5}$$

The number of false positives is defined to be:

$$\text{FP} = \sum_{s_n} \sum_{w_i \in I_{s_n}} \left\lceil \sum_{h_k} \frac{\text{overlap}(w_i, h_k)}{\text{len}(w_i)} \right\rceil_{0.95} \tag{6}$$

In our experiments, we loosened the definition in Equation 6 so that in-vocabulary words incorrectly recognized by the ASR are not counted as false positives. $w_i \in I_{s_n}$ is redefined to be in-vocabulary words that are correctly decoded by the ASR. We do not give any credit or impose any penalty for detecting a non-OOV ASR error. In this way, Equation 6 represents truly wasted conversational effort to clarify/correct words that are already correct.

One shortcoming of Equation 4 is that the system can hypothesize a region much larger than the OOV word and not be penalized, except in Equation 6. We will propose a metric in Section 3.3 that penalizes hypothesized regions that are either too short or too long. Nevertheless, the 95% overlap metrics make sense for our task: TP_{oov} measures how many OOVs we are potentially able to play back to the user and correct, while FP measures wasted effort.

Another metric that we found useful is a 5% overlap metric, where we replace $T = 0.95$ in Equations 4 and 6 with $T = 0.05$. If we constrain the detector to return at most one word per sentence, this metric gives us a idea of whether the detector is finding the approximate locations of the OOVs. This metric helps decouple the classifier performance from that of the merging algorithm, as we will see in the next section.

3.2. Using Overlap Metrics

Figure 2 shows the use of overlap metrics. In the first experiment, we use maxent classifier scores to label zero or more decoded words as a detected OOV region. A word is labeled OOV only if the OOV category score wins compared to WErr and WCorr, and if it exceeds a score threshold. Sweeping the score threshold produces the lowest

ROC curve, which at 6% FPR (False Positive Rate) has about 40% OOV recall (true detection rate), using the 95% overlap metric.

We next perform a diagnostic test using the 5% overlap metric. We ask the maxent classifier to label at most one decoded word as OOV per sentence, with rejection based on a score threshold. Then we compute the recall based on 5% overlap, i.e. give credit if the detected word overlaps at least 5% of the true OOV duration. We see that the OOV recall by this metric peaks at about 80% at very low FPR (leftmost curve). Intuitively, this means that the maxent classifier is getting the rough location of the OOV, but is not getting the correct boundaries of the whole word; this motivated us to develop the strategy of first finding the rough location of the OOV and using a merging algorithm to grow the region to cover the OOV word. By doing so, we end up with much better OOV region hypotheses; in Figure 2, we see that at 6% FPR, we can now achieve an OOV recall of 78% (compared to 40%), using the 95% overlap metric.

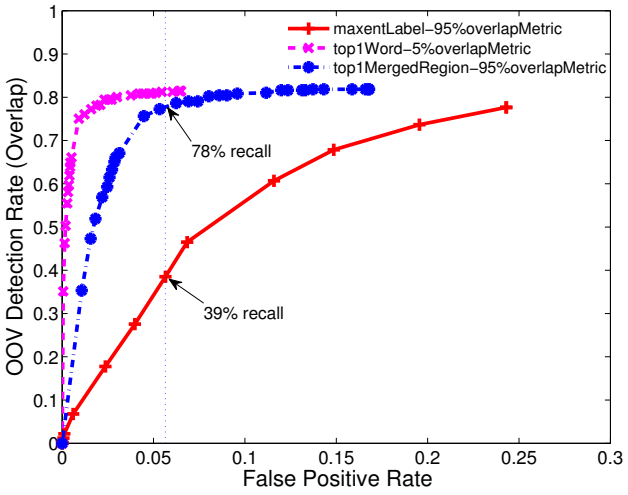

Fig. 2. Overlap Metrics.

3.3. Soft Counts and Jaccard Metric

Since our requirement is that the detected region covers the OOV word without being too long or too short, the Jaccard index [13] can be used to provide soft counts to the metric. The number of true positives is defined to be:

$$\mathrm{TP}_{\mathrm{oov}} = \sum_{s_n} \sum_{w_j \in Q_{s_n}} \max_{h_k \in H_{s_n}} \frac{\mathrm{overlap}(w_j, h_k)}{\mathrm{union}(w_j, h_k)}, \qquad (7)$$

where $\mathrm{TP}_{\mathrm{oov}}$ includes the Jaccard index. The sum is over all the true OOV instances. For each OOV instance, the Jaccard index gives a credit between 0 and 1: 0 if there is no overlap and 1 if there is exact overlap. If the hypothesized region grows larger than the maximum overlap, there is a penalty in the denominator, leading to a smaller value for the metric.

Also in the formula, we see that for each true OOV word, we allow only one of the hypothesized OOV words to contribute to the metric. If there are multiple OOV words covering the true OOV word, only the best matching one will count. We do not want multiple OOV hypothesized words to get the same credit as a single correct hypothesis, because each hypothesized word is intended to be played back separately.

The OOV detection rate according to the 95% overlap metric is insensitive to excessive length in the hypothesized OOV regions, and the Jaccard metric is more suitable for tuning parameters used in the OOV merging algorithm. The algorithm uses thresholds for "low," "medium," and "high" confidence levels. As we increase these thresholds, the OOV detection rate based on the 95% overlap metric continues to increase, resulting in very long sections of the sentence labeled as "OOV." The only indication of the decreasing quality of the hypotheses is that the false positive rate increases for the ROC curve. In contrast, the OOV detection rate based on the Jaccard metric allows us to tune the merging parameters, as shown in Figure 3, which indicates that threshold setting of (0.2,0.5,0.9) outperforms other values of the thresholds, including very low values (0.05,0.07, 0.1) and very high values (0.86,0.973,0.9788).

Because our end goal is to produce an interactive system where audio snippets are played back to the user, a desirable property of a metric is that it correlates well with human judgment about the system's ability to isolate keywords in the audio stream. To measure that correlation, we devised a listening test in which subjects were shown the text corresponding to a keyword and then played an audio snippet excising that keyword according to the automatically determined endpoints. Eight users were asked to rate the quality of the segmentation on a six point scale for each of ten keywords. The users' ratings were then correlated with the Jaccard metric as well as the 95% overlap metric. The Jaccard metric resulted in a correlation of 0.21 versus 0.09 for the 95% overlap metric.

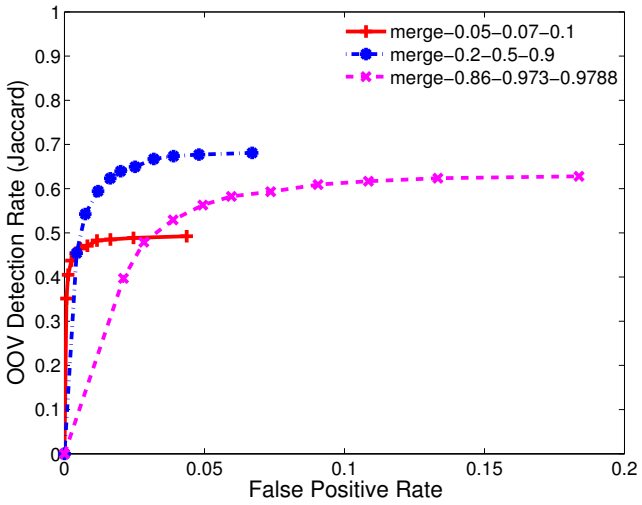

Fig. 3. Using Jaccard metric to tune "Low," "Medium," and "High" confidence threshold parameters of merging algorithm.

4. CNN ACOUSTIC MODELS

Our baseline acoustic models for speech-to-speech translation are regular speaker independent GMMs, trained with feature and model space discriminative training. The English model was trained on more than 200h of data from the DARPA Transtac speech-to-speech translation program. Like others [14], we have seen significant improvements switching to Neural Nets for various transcription tasks. However, the challenge for speech-to-speech translation compared with other transcription tasks is that the engine needs to run with low latency, limiting how much temporal context can be used. The low latency requirement makes it necessary to have only one decod-

ing pass without a second, speaker adaptive, pass. This makes it appealing to use Convolutional Neural Networks [15, 16] (CNNs), designed to achieve shift invariance in the feature domain as it was proposed in [17]. The advantage of this model is the capability of feature normalization through weight sharing and frequency shift, and it is therefore well suited for speaker independent decoding passes.

While variance normalization of the features is important for training Neural Nets, the low latency requirement makes it difficult for us to have robust variance estimates of the test data. Instead we opt for a two stage solution. We estimate global mean and variance statistics on the training data and update only the mean statistics in decoding which is less sensitive to short utterances.

For English, we reduced the error rate from 14.1% (our best GMM) to 9.0% for our CNN model. Similar error rate improvements were also observed for Iraqi. It is also worthwhile mentioning that the decoder speed improved by a factor of 3, partially because of better model, but also because the Neural Net model can use multiple cores more effectively.

The CNN also improved OOV detection. Figure 4 shows the OOV detection performance comparing the new CNN acoustic model with the GMM model. Just by substituting the CNN model, the false positive rate was substantially reduced, with a little improvement in recall. We also made improvements in the OOV detector training, including dealing with epsilon bins, posterior normalization, duration weighted scores, text normalization within consensus bins, and matched training. The improvements resulted in further improvement in OOV recall as shown in the figure.

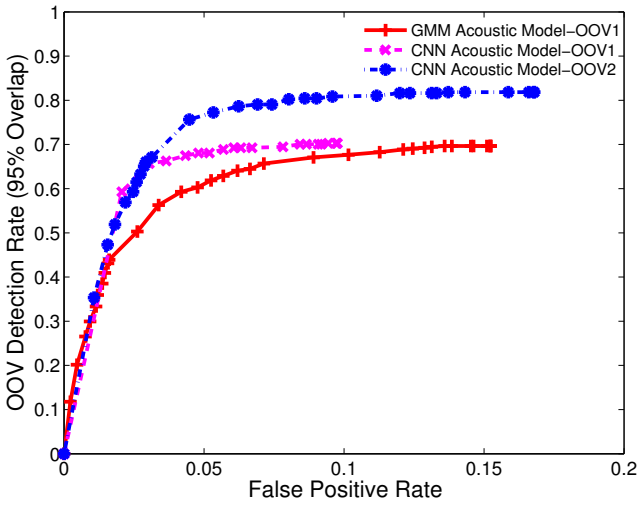

Fig. 4. OOV detection performance with CNN acoustic model and improved OOV model.

5. RELATION TO PRIOR WORK

Detecting out-of-vocabulary words in a speech transcription system has been previously explored for various applications, e.g. [18], [19], [8], and [20]. Most of the approaches can be characterized as either filler models or confidence-estimation models, or a combination of the two such as [21, 8]. Our system falls into the "combination" category, as we include word fragments in the vocabulary [7] and model features including word posterior probabilities.

In the context of a speech-to-speech translation system, Kumar et al. [1] focused on identifying OOV named entities using a maximum entropy model with word posterior, lexical and part-of-speech features. Our work differs from theirs in that we do not explicitly focus on named entities but rather model all OOVs, we model a variety of features in addition to word posterior, and we tune our system using various metrics tailored to the portion of the system in question. Some authors such as [5] have focused on grouping several hypothesized words together prior to modeling, for example by using conditional random fields [8]. Our merging algorithm is similar in spirit, aiming to group individual hypothesized words into a single OOV region. Our system requires accurate time-boundary information for out-of-vocabulary items because we engage the user in clarification dialogue in which audio snippets of the OOV region are played back, similar to the system described in [2].

Our performance metrics differ from those presented in prior art, in that we explicitly penalize discrepancies in the time boundaries between the reference and hypothesized OOVs.

6. DISCUSSION/CONCLUSIONS

In this paper we have looked at detecting out-of-vocabulary words in the context of a speech-to-speech translation system and measuring the accuracy of the OOV detection.

We have noted that accurately identifying the OOV word boundaries is important for error correction, and that most existing metrics do not suffice. We prefer to err on the side of including excess speech rather than losing information; in light of that preference, we have introduced a 95% overlap criterion for assessing the performance of our OOV detector. We have also introduced a Jaccard metric which penalizes discrepancies in the time boundaries of OOV estimates, whether they be too short or too long. We found that using CNNs in our ASR engine outperforms GMMs both in terms of word error rate and OOV detection.

In order to address the question of how accurate are the boundaries which we consider to be "truth," we looked at forced alignments using two different Gaussian mixture acoustic models, as well as a set of alignments using a Convolutional Neural Network model. We also synthesized an additional alignment by taking the median of the word start and end times from the set of available alignments. The recognition hypothesis of an ASR often turns out to be noisy or erroneous in terms of estimation of actual word boundaries, especially in the the presence of OOV items in speech [22]. We performed a hand alignment of the OOV words occurring in 20 utterances from our development test set. We found that the average deviation from the hand alignments was 0.049 seconds for the median and 0.054 seconds for the CNN alignment. Because hand-aligning the entire corpus would be very costly, we decided to use the median alignment as our reference alignment.

One question which we have not yet answered is "How inaccurate can boundaries of a hypothesized OOV word be before the user experience degrades when that OOV is played back to the user for correction or clarification?" Having an answer to this question would allow us to set the thresholds for the overlap metrics in a more principled way.

7. ACKNOWLEDGMENT

This work was partially funded by the DARPA BOLT program. The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense. We also thank Salim Roukos, Young-Suk Lee, Raimo Bakis, Leiming Qian, and Andy Sakrajda.

8. REFERENCES

[1] R. Kumar, R. Prasad, S. Ananthakrishnan, A. N. Vembu, D. Stallard, S. Tsakalidis, and P. Natarajan, "Detecting OOV named-entities in conversational speech," in *Proc. Interspeech*, 2012.

[2] R. Prasad, R. Kumar, S. Ananthakrishnan, W. Chen, S. Hewavitharana, M. Roy, F. Choi, A. Challenner, E. Kan, A. Neelakantan, and P. Natarajan, "Active error detection and resolution for speech-to-speech translation," in *IWSLT*, 2012.

[3] N. F. Ayan, A. Mandal, M. Frandsen, J. Zheng, P. Blasco, A. Kathol, F. Béchet, B. Favre, A. Marin, T. Kwiatkowski, M. Ostendorf, L. Zettlemoyer, P. Salletmayr, J. Hirschberg, and S. Stoyanchev, "Can you give me another word for hyperbaric?: Improving speech translation using targeted clarification questions," in *Proc. ICASSP*, Vancouver, 2013, IEEE.

[4] W. Chen, S. Ananthakrishnan, R. Kumar, R. Prasad, and P. Natarajan, "ASR error detection in a conversational spoken language translation system," in *Proc. ICASSP*, Vancouver, 2013, IEEE.

[5] W. Chen, S. Ananthakrishnan, R. Prasad, and P. Natarajan, "Variable-span out-of-vocabulary named entity detection," in *Proc. Interspeech*, Lyon, France, 2013.

[6] L. Dybkjr and N. Bernsen, "Usability evaluation in spoken language dialogue systems," in *Proc. ACL workshop on Evaluation Methodologies for Language and Dialogue Systems*, Toulouse, France, July 2001, pp. 9–18.

[7] A. Rastrow, A. Sethy, and B. Ramabhadran, "A new method for OOV detection using hybrid word/fragment system," in *Proc. ICASSP*, Taipei, 2009, IEEE, pp. 3953–3956.

[8] C. Parada, M. Dredze, D. Filimonov, and F. Jelinek, "Contextual information improves OOV detection in speech," in *Proc. NAACL*, 2010.

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

[10] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech and Language*, vol. 14, no. 4, pp. 373–400, 2000.

[11] C. White, J. Droppo, A. Acero, and J. Odell, "Maximum entropy confidence estimation for speech recognition," in *Proc. ICASSP*. IEEE, 2007.

[12] H. Lin, J. Bilmes, D. Vergyri, and K. Kirchhoff, "OOV detection by joint word/phone lattice alignment," in *Proc. ASRU*, Kyoto, Japan, 2007, IEEE, pp. 478–483.

[13] P. Jaccard, "Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines," *Bulletin de la Société Vaudoise des Sciences Naturelles*, vol. 37, pp. 241–272, 1901.

[14] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. Interspeech*, 2011.

[15] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing*, D. E. Rumelhart and J. L. McClelland, Eds., vol. 1, chapter 8, pp. 318–362. MIT Press, Cambridge, MA, 1986.

[16] Y. Le Cun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network," in *Proc. NIPS*, 1990.

[17] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural network concepts to hybrid NN-HMM model for speech recognition," in *Proc. ICASSP*. IEEE, 2012.

[18] I. Bazzi, *Modeling Out-of-Vocabulary Words for Robust Speech Recognition*, Ph.D. thesis, Massachusetts Institute of Technology, 2002.

[19] S. Hayamizu, K. Itou, and K. Tanaka, "Detection of unknown words in large vocabulary speech recognition," in *Proc. Eurospeech*, 1993.

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

[21] T. J. Hazen and I. Bazzi, "A comparison and combination of methods for OOV word detection and word confidence scoring," in *Proc. ICASSP*. IEEE, 2001, pp. 397–400.

[22] A. Tsiartas, P. K. Ghosh, P. G. Georgiou, and S. S. Narayanan, "Robust word boundary detection in spontaneous speech using acoustic and lexical cues," in *Proc. ICASSP*, Taipei, 2009, IEEE, pp. 4785–4788.