OOV DETECTION BY JOINT WORD/PHONE LATTICE ALIGNMENT

Hui Lin[1], Jeff Bilmes[1], Dimitra Vergyri[2], Katrin Kirchhoff[1]

[1]Department of Electrical Engineering, University of Washington, Seattle, WA 98195
[2]SRI International, Menlo Park, CA 94025

{hlin,bilmes,katrin}@ee.washington.edu, dverg@speech.sri.com

ABSTRACT

We propose a new method for detecting out-of-vocabulary (OOV) words for large vocabulary continuous speech recognition (LVCSR) systems. Our method is based on performing a joint alignment between independently generated word and phone lattices, where the word-lattice is aligned via a recognition lexicon. Based on a similarity measure between phones, we can locate highly mis-aligned regions of time, and then specify those regions as candidate OOVs. This novel approach is implemented using the framework of graphical models (GMs), which enable fast flexible integration of different scores from word lattices, phone lattices, and the similarity measures. We evaluate our method on switchboard data using RT-04 as test set. Experimental results show that our approach provides a promising and scalable new way to detect OOV for LVCSR.

Index Terms— out-of-vocabulary, OOV, lattices, graphical models, Bayesian networks, dynamic Bayesian networks

1. INTRODUCTION

Out-of-vocabulary (OOV) words are a well-known problem for large vocabulary speech recognition systems, especially for continuous speech where the presence of OOVs can often cause a mis-recognition of neighboring words due to the language model. Indefinitely increasing the vocabulary size to include still rarer words can indeed help alleviate the problem, but that will never solve it entirely since as languages evolve and new types enter the spoken lexicon, new sounds and thus OOVs constantly appear. Moreover, an increased vocabulary size can sometimes produce a higher word-error rate (i.e., additional substitutions), potentially leading to a tradeoff between the recognition accuracy of frequent words and not declaring rare words as OOVs. Reliable detection of at least the *presence* and *time location* of OOV words, therefore, is a viable long-term solution to improving real-word applications of automatic speech recognition (ASR) such as spoken term detection.

Many approaches have been proposed to detect OOVs [1, 2, 3, 4, 5, 6, 7, 8]. The most common methods focus on explicitly modeling OOVs using either filler or other generic word models. The goal is for the generic models to "absorb" the OOVs, either by specifically designed and trained structures, or by adding various phone loops to the HMM's state sequence to allow a wider variation of phonetic sequences. These approaches, however, need to be carefully (and ideally, discriminatively) tuned lest the generic models absorb non-OOV speech. Yet another category identifies OOVs without modeling OOVs explicitly. For example, in [6] confidence measures based on multiple and diverse knowledge sources are employed, in [7] word-level confidence scoring mechanism achieves higher accuracy for the in-vocabulary data with the same OOV detection rate compared to the filler-model based approach, and in [8] two recognition processes working in parallel compare the acoustic scores of a phonetic and a lexically constrained recognizer.

In this paper, we propose a new method for OOV detection based on the joint alignment of independently generated word and phone lattices, all expressed using graphical models. Lattices, as used in ASR, are a concise representation of a list of hypothesis strings (often an N-best list) along with their acoustic, language model, and possibly other (such as posterior) scores. The use of lattices has been crucial to almost all modern multi-pass LVCSR systems. We utilize separate lattices to represent, respectively, a set of word and a set of phone hypothesis strings for an unknown utterance. A joint alignment between the lattices therefore provides a phonetic sequence that best aligns the phone lattice and, via a pronunciation lexicon, the word lattice. Our general idea is similar to, but was not inspired by, [8] who used only the 1-best hypotheses. We express the above alignment using graphical models (GMs). Recent developments on the Graphical Model Tool Kit (GMTK) enables us to represent multiple heterogeneous lattices within the same model [9]. With the powerful modeling capability and rapid turnaround time of graphical models, we can easily incorporate both word and phone lattices together and jointly align them using GMTK's decoding algorithms. We note that DBNs have in the past been used to produce extensions to standard Levenshtein string-edit distance [10]. Here, we utilize such an approach, but for joint alignments. We evaluated our technique on switchboard data, using word and phone lattices generated from the

This work was supported by an ONR MURI grant, No. N000140510388.

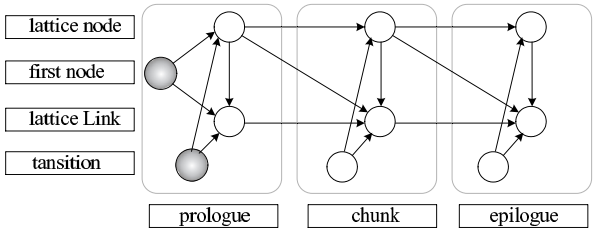


Fig. 1. A graphical model representation of a lattice [9]

SRI RT04 conversational telephone speech (CTS) system. We have found that our approach offers a promising new way of detecting OOVs for LVCSR.

2. GRAPHICAL MODELS OF LATTICES

A graphical model (GM) [11] is a visual representation of factorization properties of families of probability distributions. When instantiated, a GM also includes local score functions such as conditional probability tables (CPTs) for Bayesian networks. A dynamic Bayesian network (DBN) [12, 13], one form of graphical model, consists of a directed acyclic graph $G = (V, E)$ where V is a set of vertices (corresponding to random variables) and E is a set of directed edges, and where there is a fixed size template that can be unrolled to an unbounded length in order to represent any given length utterance. A lattice consists of a directed graph $\mathcal{D} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is a set of nodes, and $\mathcal{L} \subseteq \mathcal{N} \times \mathcal{N}$ is a set of directed links between two nodes, so that if $(n_1, n_2) \in \mathcal{L}$ then there is a link from n_1 to n_2. Nodes typically represent time points (we use the notation $\tau(n)$ to indicate the time point associated with node n); links represent tokens (words or phones) along with a number of possible scores (acoustic, language, posterior, etc).

Graphical model representations of lattices, and their implementation in GMTK, is described in [9] and GMTK-style DBNs are fully described in [13]. We give here only a brief outline and refer the reader to [9] for full details. To represent a lattice using a GM, we utilize two vertices at each time frame t, a lattice node N_t and a lattice link L_t (Figure 1). The two successive values $N_{t-1} = i, N_t = j$ determine a lattice link $L_t = l$ only if $(i, j) \in \mathcal{L}$. A third binary variable T_t indicates node transition. The purpose of T_t is to interface to the rest of a more complex DBN that uses a lattice (e.g., [9] and Figure 2).

The time information for each lattice node is represented by a time-inhomogeneous CPT, meaning the CPT $p(N_t = j | N_{t-1} = i, T_t = b) = f_t(j, i, b)$ is a function of time. The vertex N_t can only transit to value j when the current time is $\tau(j)$ (or within a time region around $\tau(j)$). We note here that this graphical construct can appear any number of times in a DBN that represents lattices, and in fact there is no need

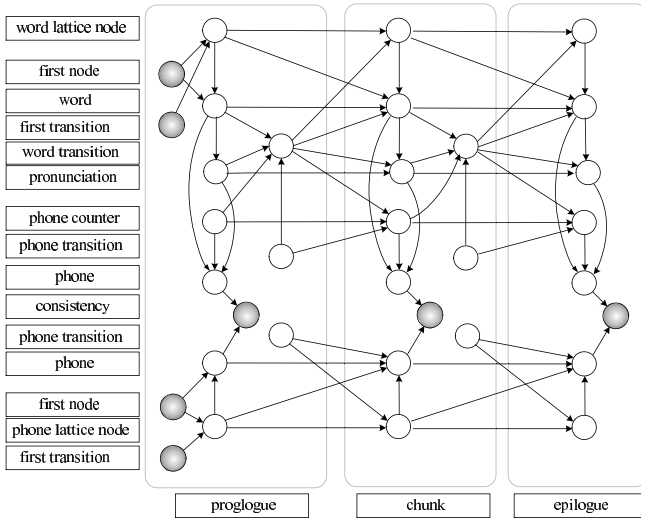


Fig. 2. Graphical model for word/phone lattice alignment

for the lattices to have the same meaning. This is precisely how we implement our joint word/phone lattice decoder, by the utilization of multiple separate lattices in a DBN.

3. JOINT WORD/PHONE LATTICE ALIGNMENT

We represent a joint set of word/phone lattices using the DBN in Figure 2. The word lattice is at the top where the link variable in this case represents a word. The variables "word transition", "pronunciation", "phone counter", "phone transition", and "phone" are used to explicitly express information about a pronunciation lexicon. Most of the CPTs between these variables are deterministic (see [13]). These variables and their associated CPTs yield the phone sequences for the word hypotheses contained in the word lattice. The bottom part of the figure shows the phone lattice, where the link variable is a phone from the phone lattice. To distinguish between the two phone variables, we denote the phone-lattice phone variable as H_t^p and the phone variable derived from word lattice via the pronunciation lexicon as H_t^w.

Given an utterance of length T, the "best alignment" between the word and phone lattices really means making the strings $(H_t^w)_{t=1}^T$ and $(H_t^p)_{t=1}^T$ as consistent as possible. Therefore, a binary consistency variable C_t is introduced to enable this functionality. We use the notion of a observed child [14, 15, 16] to link the two lattices together. This consistency variable is always observed with value unity. Its two parents are H_t^w and H_t^p, and the CPT $p(C_t = 1 | H_t^w, H_t^p) = f(H_t^w, H_t^p)$ is simply a function of H_t^w and H_t^p. If H_t^w is identical or similar to H_t^p, $f(H_t^w, H_t^p)$ should take larger values, and $f(\cdot)$ should take smaller values otherwise. In other words, the consistency variable will have high probability (or be 1) to be value unity if the phone variables from the word

and phone lattices are similar to each other. Therefore, better matched phone hypothesis string pairs will likely survive any pruning stage in decoding, while a hypothesis that produces less similar phone sequences will either get a low score or be pruned away.

The function $f(H_t^w, H_t^p)$ can be formed in a number of ways. The simplest approach utilizes a hard "step" function, i.e., something like $f(H_t^w, H_t^p) = 0.9$ if $H_t^w = H_t^p$ and $f(H_t^w, H_t^p) = 0.1$ otherwise. Using a $0/1$-valued function (e.g., using a value of 0 when $H_t^w \neq H_t^p$) would not allow insertions or deletions at all (such hypotheses would get pruned away). Another potentially more accurate approach utilizes linguistic/phonetic knowledge. For example, phonetic similarity (confusability) measures can be employed by mapping the distance or cost between phones into $[0, 1]$-valued probabilities. Also, $f(H_t^w, H_t^p)$ could be learnt given sufficient availability of both positive and negative training data [16].

4. OOV DETECTION

By combining word and phone lattices together and performing maximum-likelihood decoding, we can get the best jointly aligned phone strings $\bar{h}_{1:T}^w$ and $\bar{h}_{1:T}^p$, along with the word sequence $\bar{w}_{1:N}$, for a T frame utterance when N words have been decoded. For the OOV detection, our hypotheses is as follows: regions of time during which there is relatively little local misalignment between $\bar{h}_{1:T}^w$ and $\bar{h}_{1:T}^p$ correspond to the case where an OOV has not occurred, while regions where there is relatively high local misalignment indicate a likely OOV region. The reason is that during an OOV word, it is likely that even the best aligned word (and best pronunciation thereof) in the word-lattice and the best aligned sub-string in the phone lattice will require additional (or higher cost) edit operations. Our task, therefore, is to find localized misaligned regions within $\bar{h}_{1:T}^w$ and $\bar{h}_{1:T}^p$.

Let s_t denote a misalignment indicator at time frame t, with $s_t = \mathbf{1}\{\bar{h}_t^w \neq \bar{h}_t^p\}$. Then $(s_t)_{t=1}^T$ is a length-T vector which we smooth using a length-M Hamming window $(w_i)_{i=0}^{M-1}$ to produce $\ell_t = \sum_{i=-M/2}^{M/2} s_{t+i} w_{i+M/2}$. Finally, to detect an OOV in a time region, we introduce two real-valued thresholds: if regions within $\ell_{1:T}$ are above a *detection threshold* α for a time duration longer than the *duration threshold* β, we hypothesize an OOV region. If the hypothesized OOV region overlaps with a true OOV region, we count this as a (true positive) *OOV detection*, otherwise we count it as a (negative) *false detection*. This approach tends to favor recall at the cost of some precision. Figure 3 illustrates this process.

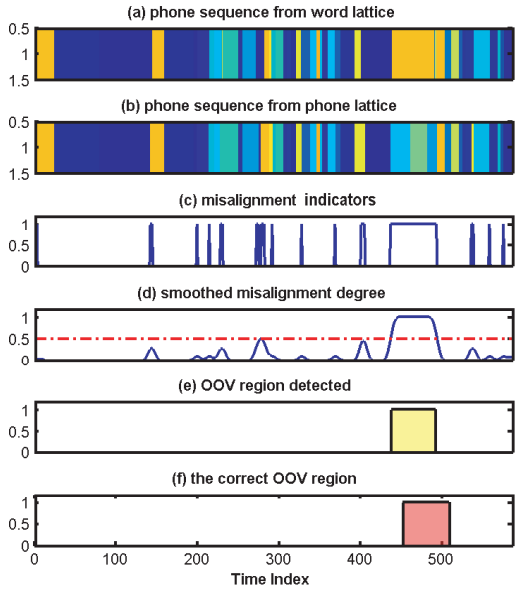
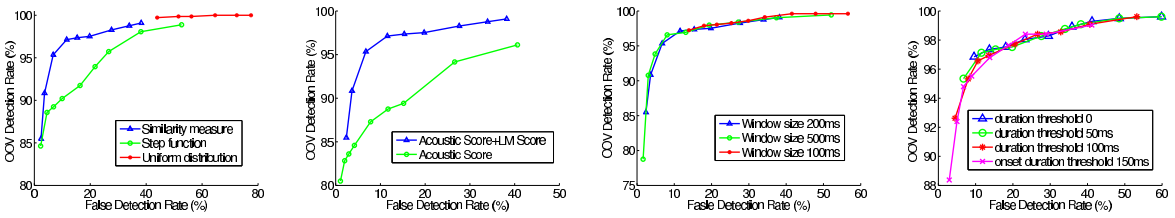


Fig. 3. *Our OOV detection procedure: (a) a phone sequence $\bar{h}_{1:T}^w$ is generated from the word lattice jointly with (b) a phone sequence $\bar{h}_{1:T}^p$ from the phone lattice (different colors designate different phones); (c) misalignment indicators $s_{1:T}$; (d) smoothed misalignment $\ell_{1:T}$ degree, where the dashed (red) line indicates α; (e) the detected OOV region; and lastly (f) the correct OOV region.*

5. EXPERIMENTS

5.1. Experimental Setup

We used 3 hours of conversational telephone speech (CTS) for our test lattices (both phone and word). The speech-to-text systems used are based on the SRIs RT04 systems for CTS [17]. It is using the SRI Decipher(TM) speaker-independent continuous speech recognition system, a continuous-density, state-clustered hidden Markov model (HMM). We used a 2-pass decoding scheme, where the first pass generated 2-gram lattices, which were then expanded with a bigger LM, and used for constrained decoding for the second pass lattice generation. Prosodic phone-in-word duration models were used for lattice rescoring. In the first pass, a phone-loop MLLR adapted within-word MFCC model is used to generate bigram lattices, which are rescored with 4-gram word LM, and consensus decoding is applied to generate one-best hypotheses for SAT and MLLR adaptation of the cross-word PLP model. The bigram lattices are then expanded to trigram lattices for constrained decoding using SAT-MLLR adapted cross-word PLP models to generate the 2nd pass lattices. These lattices are rescored using prosodic duration models and a 4-gram LM, and are then used as our "word lattices." For the phone lattices, a phone trigram was trained from an aligned phone transcription of the training data. The phone lattices were then obtained by this phone trigram with MFCC+ICSI features front-end and cross-word triphone models. Acoustic training uses all of Hub5 CTS plus CTRAN Switchboard 2 plus 2000 hours of Fisher data. MFCC and PLP models are trained each



(a) (b) (c) (d)

Fig. 4. *ROC Curves varying α: (a) Different $f(\cdot)$s, $\beta = 50ms$, $W = 200ms$, use of AC/LM weights; (b) Varying use of LM/AC weights, $\beta = 50ms$, $W = 200ms$, phone similarity for $f(\cdot)$; (c) Different window size, $\beta = 0ms$, use of AC/LM weights, phone similarity for $f(\cdot)$; (d) Different duration thresholds β, $W = 200ms$, use of AC/LM weights, phone similarity for $f(\cdot)$.*

using complementary halves of the Fisher corpus. LM training uses all CTS, plus UW web data, plus Broadcast News 96 transcripts, interpolated and entropy-pruned.

As the OOV rate for this data was relatively low (only about 0.5%), we performed two experiments on modified forms of these lattices which simulated the case where the OOV rate is much higher.

5.2. Experiment I

In this first case, we mainly tested how the parameter configurations affect the results of OOV detection. We simulated a larger OOV phenomenon by randomly choosing words to be OOVs, with longer words having a higher probability of being chosen as an OOV than shorter ones (since OOVs are less likely to be short). We used simple phone length (based on canonical pronunciations) to judge a word's length (so there are cases where actual short surface realizations of long canonical words have been removed). Once the set of "OOV words" were chosen, all associated word lattice links were removed. This procedure was randomly and independently repeated 20 times, in each case producing an OOV rate of 2.7%.

We utilize receiver operating characteristic (ROC) curves to display the tradeoff between OOV true detection and false detection rates (Figure 4). The standard deviations over the 20 trials are not explicitly shown, however, since they are almost the same for each parameter setting and would obfuscate the plots unnecessarily (for the detection rate it is about 1%, and for the false detection rate, it is about 0.5%). To produce each ROC curve in each figure, the parameter α is varied while keeping the other parameters fixed (most of the plots had alpha vary from 0.1 to 0.9 in steps of 0.1, while a few of the curves used an extra value or two to obtain extra points).

Figure 4(a) illustrates the ROC curves for three different choices of $f(\bar{h}_t^w, \bar{h}_t^p)$. The first (red) uses a uniform distribution as a sanity check, meaning $f(\bar{h}_t^w, \bar{h}_t^p) = 0.5$, which performs poorly as expected. The second (green) uses the step function $f(\bar{h}_t^w, \bar{h}_t^p) = 0.9$ when $\bar{h}_t^w = \bar{h}_t^p$ and $f(\bar{h}_t^w, \bar{h}_t^p) = 0.1$ otherwise. The third (blue) uses a simple phone similarity measure [18]. Here, $\beta = 50$ms, $W = 200$ms, and acoustic and language model scaling factors are obtained from the word and phone lattice. Clearly, the phone similarity measure, which achieves 95.3% detection rate at a 6.76% false detection rate, outperforms other two methods. These results show that the choice of $f(\bar{h}_t^w, \bar{h}_t^p)$ is crucial for our approach to perform well. Figure 4(b) shows the case with and without utilizing the language model (LM) weights, showing that this also is an important factor (see Section 6). Different window sizes W were also evaluated (Figure 4(c)) showing that this parameter did not influence the overall shape of the ROC curve, but rather where on this curve the various α samples would span. Finally, changing the duration threshold β can achieve an OOV detection rate of 92.6% with only a 4.48% false detection rate.

5.3. Experiment II

In this second case, test lattices were generated by reducing the recognizer vocabulary. The words were ordered based on the unigram frequency estimated on the LM training data, and a cutoff was chosen that resulted in a 3.5% OOV rate.

The ROC curve of our approach is shown in Figure 5, where $W = 200$ms, $\beta = 50$ms, and where we used the acoustic and language model scaling factors and the phone similarity measure. To further evaluate our method, other approaches for OOV detection were implemented in this experiment for comparison, and are described below.

5.3.1. Comparison with the 1-best Approach

Alignment between the 1-best hypotheses from the word and phone recognizers could also be used for OOV detection (similar to previous work [8]). In this part of experiment, the word/phone lattices were pruned to contain only the 1-best paths. The alignment between them was performed using the same graphical model (Figure 2) and the same parameter configurations (i.e., using language model scaling factors and the phone similarity measure) as the lattice alignment approach.

Different from the alignment between all alternative competing hypotheses, the 1-best approach could take into account only the 1-best hypotheses. Since the 1-best path is

likely to be inaccurate due to speech recognition errors, it is quite possible that no good alignment between the 1-best output of the word and phone recognizers exists even when there are perfectly aligned sub-optimal paths in the lattice alignment case. This might results a higher false detection rate for the 1-best approach. Experimental results support this claim. The ROC curve (by varying α) of the 1-best approach is shown in Figure 5. It can be seen that the false detection rate of the 1-best approach is higher than that of lattice approach for the same OOV detection rate. For a significant range of false detection values, our approach achieves 10-15% improvement in OOV detection rate over the 1-best approach.

5.3.2. *Comparison with the Confidence Measure Approach*

Another approach for OOV detection is done by using a confidence measure to predict whether a recognized word is actually a substitution of an OOV word. In this experiment, we tried two kinds of confidence measures both obtained from the lattices. One is the word log-likelihood score and the second is the word posterior probability. The word log-likelihood scores were calculated as the scaled combination of the acoustic and language model scores. The word posterior probabilities were estimated from the lattice using the forward-backward algorithm.

We used a threshold on the confidence measures to determine whether to characterize a hypothesized word as an OOV or not. The ROC curve can be obtained by varying the threshold, as shown in Figure 5. Generally, posterior word probabilities perform better as confidence measures than other numerous features that can be extracted from the lattices [19], but they still cannot achieve better performance in OOV detection than the lattice alignment approach. The reason mainly lies in the fact that although confidence measures may be a good indicator of mis-recognized words, they are unable to tease apart errors due to OOV words from those errors due to other phenomena such as degraded acoustic conditions.

5.3.3. *Comparison with Filler Model Approach*

The filler model based approach for OOV detection was implemented here as a comparison to the above approaches, by adding a lexical entry "@reject@" to the recognizer to represent the OOV word. This word contained a phone loop as pronunciation in order to match any sequence of phones. The presence of an OOV word in the output was based on the top recognizer hypothesis: the time region corresponding to the "@reject@" token in the 1-best output was marked as an OOV region. The main drawback of this implementation is the fact that the filler model can potentially absorb parts of the speech corresponding to in-vocabulary words, and thus cause an large increase in word deletions in the ASR output. In our experiment with the filler model we achieved 8% OOV detection rate, with about 10% increase in word deletions, while with 65.3% detection rate we had 63.8% word deletions, making the ASR output not very useful.

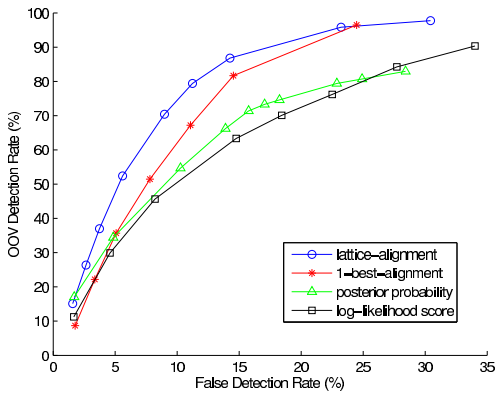


Fig. 5. *ROC plot for experiment II.*

6. DISCUSSION AND FUTURE WORK

A potential problem of our approach is that with a bushy enough word lattice, there might always be an alternate (and wrong) word path that is phonetically similar enough to the true phone sequence in the utterance to produce a good alignment with the phone lattice. In our experiments, however, we found that alternative word-lattice paths did not tend to be chosen, primarily due to the language model (evidence for this is seen in Figure 4(b) which shows that not using a language model scale significantly hurts performance) and also due to these alternative paths not aligning well enough.

In the future, we will focus on supervised training of the consistency CPT, possibly using the approach of [16]. We will also obtain lattices from "low-resource" languages and/or a broadcast news style corpus, both of which tend to have higher true OOV rates. We would like also to apply our approach to a named entity detection procedure. Ultimately, we wish this work to be integrated into a spoken term detection task, with the goal of finding all occurrences of a given sequence of words in a speech corpus. Proper OOV detection could significantly facilitate such a process.

7. REFERENCES

[1] Ayman Asadi, Richard Schwartz, and John Makhoul, "Automatic detecion of new words in large vocabulary continuous speech recognition system," *Proc. ICASSP*, 1990.

[2] Issam Bazzi and James R. Glass, "Modeling out-of-vocabulary words for roubust speech recognition," *Proc. ICSLP*, 2000.

[3] Thomas Schaaf, "Detection of OOV words using generalized word models and a semantic class language model," *Proc. Eurospeech*, 2001.

[4] Odette Scharenborg and Stephanie Seneff, "A two-pass for strategy handling OOVs in a large vocabulary recognition task," *Proc. Interspeech*, 2005.

[5] M. Bisani and H. Ney, "Open vocabulary speech recognition with flat hybrid models," in *Proc. of Interspeech*, 2005, pp. 725–728.

[6] Sheryl R. Young, "Recognition confidence measures: Detection of misrecognitions and out-of-vocabulary words," Tech. Rep., Carnegie Mellon University, 1994.

[7] Hui Sun, Guoliang Zhang, Fang Zheng, and Mingxing Xu, "Using word confidence measure for OOV words detection in a spontaneous spoken dialog system," *Proc. Eurospeech*, 2001.

[8] Satoru Hayamizu, Katunobu Itou, and Kazuyo Tanaka, "Detection of unknown words in large vocabulary speech recognition," *Proc. Eurospeech*, 1993.

[9] Gang Ji, Jeff Bilmes, Jeff Michels, Katrin Kirchhoff, and Chris Manning, "Graphical model representations of word lattices," *IEEE/ACL 2006 Workshop on Spoken Language Technology (SLT)*, 2006.

[10] K. Filali and J. Bilmes, "A dynamic Bayesian framework to model context and memory in edit distance learning: An application to pronunciation classification," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, Ann Arbor, Michigan, June 2005.

[11] S. L. Lauritzen, *Graphical Models*, Oxford Science Publications, 1996.

[12] T. Dean and K. Kanazawa, "Probabilistic temporal reasoning," *AAAI*, pp. 524–528, 1988.

[13] Jeff Bilmes and Chris Bartels, "Graphical model architectures for speech recognition," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 89–100, September 2005.

[14] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 2nd printing edition, 1988.

[15] J. Bilmes, "On soft evidence in Bayesian networks," Tech. Rep. UWEETR-2004-0016, University of Washington, Dept. of EE, 2004.

[16] S. Reynolds and J. Bilmes, "Part-of-speech tagging using virtual evidence and negative training," in *Human Language Technology Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-2005)*, Vancouver, CA, Oct 2005.

[17] A. Stolcke et al., "Recent innovations in speech-to-text transcription at SRI-ICSI-UW," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1729–1744, Sept. 2006.

[18] J. Clark and C. Yallop, *An Introduction to Phonetics and Phonology*, Blackwell, 1995.

[19] F. Wessel, R. Schluter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Trans. Speech and Audio Processing*, vol. 9, no. 3, Mar 2001.