IDIAP RESEARCH REPORT

Comparing Different Word Lattice Rescoring Approaches Towards Keyword Spotting

Joel Pinto [a b] Herve Bourlard [a b] Zacharie De Greve [a c]
Hynek Hermansky [a b]

IDIAP–RR 07-32

July 2007

soumis à publication

[a] IDIAP Research Institute, Martigny, Switzerland
[b] École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
[c] Faculté Polytechnique de Mons, Belgium

IDIAP Research Institute **www.idiap.ch**
Rue du Simplon 4 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

Rapport de recherche de l'IDIAP 07-32

Comparing Different Word Lattice Rescoring Approaches Towards Keyword Spotting

Joel Pinto Herve Bourlard Zacharie De Greve Hynek Hermansky

July 2007

soumis à publication

Résumé. In this paper, we further investigate the large vocabulary continuous speech recognition approach to keyword spotting. Given a speech utterance, recognition is performed to obtain a word lattice. The posterior probability of keyword hypotheses in the lattice is computed and used to derive a confidence measure to accept/reject the keyword. We extend this framework and replace the acoustic likelihoods in the lattice obtained from a Gaussian mixture model (GMM) with likelihoods derived from a multilayered perceptron (MLP). We compare the two rescoring techniques on the conversational telephone speech database distributed by NIST for the spoken term detection evaluation. Experimental results show that GMM lattices still perform better than the rescored lattices for short and medium length keywords, but on longer keywords, the MLP rescored word lattices perform slightly better.

1 Introduction

In large vocabulary continuous speech recognition (LVCSR) based approach to keyword spotting (KWS), keywords are spotted from the word lattice. LVCSR based keyword spotting is typically used in information retrieval applications like searching broadcast news etc, where it is shown to out perform the conventional methods based on garbage/filler modeling [1]. This is because in LVCSR approach, both the keywords as well as the non-keywords have detailed models (derived using pronunciation lexicon) as opposed to the conventional keyword spotting where, only the keyword has a detailed model. Moreover, language model information is also exploited in LVCSR based keyword spotting.

A word lattice is a compact representation of the multiple word hypothesis for a given speech utterance. The posterior probability of a word hypothesis conditioned on the entire utterance can be computed from the word lattice using the forward backward re-estimation algorithm. The word posterior probability was first proposed as a confidence measure in LVCSR [2][3]. Subsequently, this approach has been successfully applied in the state-of-the-art keyword spotting systems. Most of the systems (e.g. BBN, SRI, etc.) at the 2006 NIST spoken term detection evaluation were based on this approach [4][5].

In this work, we explore the LVCSR based approach to keyword spotting. We extend this work and rescore the word lattice by replacing the acoustic likelihoods obtained from the GMM in the ASR with the likelihoods from a multilayered perceptron (MLP). This work is motivated by our hypothesis that acoustic model likelihood obtained from an MLP trained to discriminate phonemes may be better than the likelihoods from GMM.

2 Word Posteriors : GMM Lattice

In this section, we describe the word lattice and the estimation of the posterior probability of a word hypothesis in the lattice. Here, the acoustic model likelihood is obtained from a Gaussian mixture model.

2.1 Word Lattice

In maximum likelihood approach to speech recognition, given a language model, a dictionary and the acoustic model, a search network called trellis is built which represents all possible word hypotheses that can be recognized. While decoding, given an observation vector sequence, the recognized word sequence is that which is most likely to have been produced by the trellis.

Instead of a single best hypothesis, multiple hypotheses can also be obtained while decoding and compactly represented in the form of a word lattice. A word lattice is a directed, acyclic, and weighted graph, where each node represents a time instance and each edge represents the word hypothesis along with its acoustic model likelihood and the language model probability. Confidence measure for a word can be computed from the word lattice using the forward-backward re-estimation algorithm [3][2].

2.2 Posterior Probability Estimation

The posterior probability of a word hypothesis $[w; t_s, t_e]$ conditioned on the entire observation is denoted by $P([w; t_s, t_e]|x_1^T)$. It is also refered to as edge posterior probability as the word $[w; t_s, t_e]$ correspond to an edge in the word lattice. This posterior probability is similar both in concept and computation to the state posterior probability in an HMM framework described in [6]. The posterior probability of a word hypothesis is given by

$$P([w; t_s, t_e]|x_1^T) = \sum_{q \in Q} P([w_m; t_{sm}, t_{em}]_{m=1}^{M} \mid x_1^T) \tag{1}$$

where Q is the set of all the word hypothesis sequences in the lattice that contain the word hypothesis $[w; t_s, t_e]$. t_{sm} and t_{em} are the start and end time frame for word the word w_m. The term inside the summation depends on q but is dropped here for notational simplicity. For path q, let $w_1^M = w_1, w_2, \ldots w_M$ be the word sequence in the path and $x_1^T = x_{t_{sm}}^{t_{em}}|_{m=1}^M$ be the corresponding partition of the observation sequence. Assuming that **(a)** acoustic observation $x_{t_{sm}}^{t_{em}}$ corresponding to a word w_m depends only on the word w_m and **(b)** an n-gram language model is considered, by using Bayes rule, the posterior probability of the word hypothesis can be rewritten as :

$$P([w; t_s, t_e]|x_1^T) = \sum_Q \frac{\prod_{m=1}^M [p(x_{t_{sm}}^{t_{em}}|w_m) P(w_m|w_{m-n}^{m-1})]}{p(x_1^T)}$$

We define two variables as given below :

$$\alpha([w; t_s, t_e]) = \sum_{\underline{w'x'} \in Q'} p(\underline{w'x'}, x_{ts}^{te}, w) \tag{2}$$

$$\beta([w; t_s, t_e]) = \sum_{\underline{w'x'} \in Q''} p(x_{ts}^{te}, w, \underline{w'x'}) \tag{3}$$

where, Q' is the set of all the sequences of word-observation vector pairs from the start of the word lattice that end before the hypothesis $[w, t_s, t_e]$ and Q'' is the set of all word-observation vector pairs after $[w, t_s, t_e]$ to the end of the lattice. The union of the two sets Q' and Q'' is the set Q defined above.

In further derivation, we denote the word hypothesis on the edge of the word lattice as e. Hence, $\alpha(e)$ is the probability of all the paths from the start of the word lattice to the edge e and $\beta(e)$ is the probability of all paths that begin with the edge e and reach the end of the lattice. Here, a path is a connected sequence of edges, where each edge represents the joint event of the observation vector and the word. Denoting the acoustic model likelihood as $A(e) = p(x_{t_s}^{t_e}|w)$ and the n-gram language model probability as $L(e) = P(w_m|w_{m-n}^{m-1})$, the posterior probability of the word hypothesis can be rewritten as follows :

$$P(e|x_1^T) = \frac{\alpha(e)\beta(e)}{A(e)L(e)p(x_1^T)} \tag{4}$$

Denoting $\mathcal{P}(e)$ and $\mathcal{F}(e)$ as set of edges preceding and following the edge e respectively, the *alpha* and *beta* variables can be recursively computed as follows :

$$\alpha(e) = A(e)L(e) \sum_{e' \in \mathcal{P}(e)} \alpha(e') \tag{5}$$

$$\beta(e) = A(e)L(e) \sum_{e' \in \mathcal{F}(e)} \beta(e') \tag{6}$$

The normalizing factor $p(x_1^T)$ in (4) is the unconditional likelihood of the observation sequence and can be computed using the following :

$$p(x_1^T) = \sum_{w, t_s} \alpha([w; t_s, T]) = \sum_{w, t_e} \beta([w; 1, t_e]) \tag{7}$$

The word lattice contains overlapping edges representing the same word but with slightly shifted start and end times. To derive a meaningful confidence measure for a word, the posterior probability all the word hypothesis (with different start and end times) should be appropriately merged. One way of deriving a word posterior probability is to generate a word confusion network [7]. In our experiments, we sum up all the edge posteriors of the word that overlap in time as shown below :

$$\mathcal{S}([w; t_s, t_e]) = \sum_{\substack{[w; t'_s, t'_e] | \\ [t_s t_e] \cap [t'_s t'_e] \neq \emptyset}} P([w; t'_s, t'_e]|x_1^T) \tag{8}$$

Given a keyword, the word hypothesis with the maximum confidence score $\mathcal{S}([w; t_s, t_e])$ is selected from the cluster of overlapping word hypotheses.

The performance of a keyword spotter depends on the tradeoff between detection rate and the false acceptance rate. This tradeoff depends on the overlap between the probability distribution function (pdf) of the word posterior probability under two conditions **(a)** the word hypothesis is correct and **(b)** word hypothesis is incorrect. The area of overlap determines the minimum Bayes error for the two-class classification problem of accepting/rejecting the keyword. Lesser the overlap between the two distributions, the better is the keyword spotting performance. The acoustic and language model likelihoods determine the minimum Bayes error. Language model probabilities may be improved to introducing higher order language models. Better acoustic models also lead to improved keyword spotting performance. In this paper, we explore the use of acoustic likelihood derived from an MLP. This is explained in the next section.

3 Word Posteriors : MLP lattice

In a conventional HMM framework, the observation in a state is modeled by parametric density function *e.g.* Gaussian mixture model (GMM). Alternatively, discriminative methods can be used to estimate the state emission probability. For example, neural networks like a multilayered perceptron (MLP) can be used to estimate the posterior probability of the phonemes also refered to as phoneme posteriors. The phoneme posteriors are estimated for every frame (typically 10 ms).

3.1 Phoneme posteriors

Neural network classifiers estimate the Bayesian *a posteriori* probability provided that the network is complex enough, trained on sufficient training data and classes are taken with the correct *a priori* probabilities [8][9]. In our experiments, we use an MLP to estimate the phoneme posteriors. To train the MLP, every frame of feature vector in the training data must be labeled as one of the phoneme class. This is generally done by either hand labeling or force alignment. Cross entropy error criteria is used while training and the MLP training termination is decided by the frame-level phoneme classification rate on the cross-validation data.

3.2 Hybrid Rescoring

The word lattices provide the word information, its start and end time and an acoustic model likelihood $p(x_{ts}^{te}|w, gmm)$, where gmm is the Gaussian mixture model used in ASR. We replace this acoustic model likelihood with the likelihood obtained from an MLP $p(x_{ts}^{te}|w, mlp)$. To estimate this, a model for the word is built by dictionary lookup and considering a 3 state HMM per phoneme. The phoneme posterior probability is taken as emission probability in the HMM states and Viterbi algorithm is applied to find the optimal alignment. Denoting the observation sequence as $x_{ts}^{te} = x_1^K$ and the corresponding state sequence as s_1^K, the acoustic model likelihood can be expressed as follows :

$$p(x_{ts}^{te}|w, mlp) = P(s_1)p(x_1|s_1)\prod_{k=2}^{K} P(s_k|s_{k-1})p(x_k|s_k)$$

where, $P(s_k|s_{k-1})$ is the state transition probabilities and $p(x_k|s_k)$ is the state emission likelihood. We assume equal self-transition and next-transition probabilities. Moreover, the state emission likelihood is the same in all three states of a phoneme. Furthermore, we assume that phonemes have equal prior distribution.

The acoustic likelihood obtained from the phoneme posteriors have a different dynamic range compared to that of GMM lattices. The optimal language model scaling factor for the posterior probability estimation is that which gives minimum 1-best word error rate.

4 Experimental Setup

In this section we describe the database, generation of word lattices and the MLP phoneme posteriors. The word lattices were generated at Speech@FIT group at Brno Institute of Technology and MLP training for phoneme posterior estimation was done at ICSI, Berkely, USA. Quicknet toolkit was used to train the MLP and the lattice-tool utility in SRILM [10] was used for posterior probability computation.

4.1 Database

Experiments were conducted on the dryrun set of the conversational telephone speech (CTS) database distributed by NIST for the 2006 spoken term detection evaluation [11]. The database consisted of 3 hours of two-channel speech and each channel was processed independently. We selected only single word keywords from the search list distributed. These keywords were divided into short (103 keywords, 3591 occurrences, *e.g.* "too"), medium (159 keywords, 2014 occurrences, *e.g.* "pretty") and long (73 keywords, 496 occurrences, *e.g.* "something") to study the affect of word duration on the performance.

4.2 Word Lattices

Speech was first segmented into 'speech' and 'non-speech' class using speech-silence segmentation algorithm which removed around 50% of the data. A multi-pass LVCSR system was used to obtain the word lattice [12]. Acoustic models were trained on 278 hours of the **ctstrain04** database. Features comprised of 13 dimensional PLP cepstral coefficients along with its delta and delta-delta components. A 3-state left-to-right phoneme HMM representing 46 phoneme classes were trained with 16 Gaussian mixtures per state. Context dependent phonemes were generated by decision tree clustering. The dictionary contained 50K words. Bigram lattices were first generated by keeping 48 tokens per state and subsequently expanded with a 4-gram language model.

4.3 MLP Posteriors

Phoneme posteriors were estimated using an MLP trained on 2000 hours of Fishers CTS database. Features comprised of 13 dimensional PLP cepstral coefficients along with its delta and delta-delta components. After speaker segmentation and vocal tract length normalization, gender specific MLPs were trained. A context of 9 frames was presented at the input layer. The MLP comprised of 20800 hidden neurons and 46 output classes which included 41 phonemes, a silence class and 4 classes for speech artifacts.

5 Results

The performance of the keyword spotter is evaluated in terms of the figure-of-merit (FOM) measure, which is the average keyword detection rate for false alarm rates of $1, 2, \ldots 10$ false alarms per keyword per hour (FA/KW/h). In the case of long keyword set, the performance measure was redefined to detection rate at 1 FA/KW/h as false acceptance rates of 5 FA/KW/h and above was not reached on the receiver operator characteristics. We denote the word lattice with GMM acoustic model as GMM lattice and the word lattice with MLP acoustic model as MLP lattice. Table. 1 shows the FOM obtained by rescoring the GMM lattice as well as the MLP lattice. The results are shown for short, medium and long keywords. It is clear from the results that, the GMM lattice performs superior to the MLP lattice. The difference in the FOM is maximum for short keywords, followed by medium and long keywords.

Tab. 1 – *The FOM for keyword spotting on GMM word lattice and MLP word lattice for short, medium and long keywords.*

Keyword Type	GMM lattice	MLP lattice
short	82.3	76.7
medium	91.5	89.8
long	92.5	92.8

6 Discussion and Conclusion

In this work, we compare the performance of keyword spotting from a word lattice with GMM acoustic model and the word lattice with MLP acoustic model. Experimental results contradict our initial hypothesis that MLP lattice should perform better than GMM lattice. In the case of short and medium length keywords, GMM lattice give a higher FOM. However, in the case of longer keywords, MLP lattice is slightly better than the GMM lattice. We attribute the drop in the FOM to the mismatch between the phoneme posteriors and the dictionary. While the phoneme posteriors capture the identity of the acutal sound, the dictionary is based on expected way to pronounce the word. Due to the discriminative nature, lattice with MLP acoustic model may be more sensitive to mispronunciations than lattice with GMM acoustic model. The trend observed in the difference in the figure-of-merit for the two rescoring techniques for short, medium and long keywords seem to strengthen this argument. In the case of short keywords, a mismatch in one or two phonemes will affect the keyword detection performance, while longer keywords may be tolerate mismatch at a few places.

7 Acknowledgements

This work was supported by the European Union (EU) under the integrated project DIRAC, Detection and Identification of Rare Audio-visual Cues, contract number FP6-IST-027787 as well as the integrated project AMIDA, Augmented Multi-party Interaction with Distance Access, contract number IST-033812. The authors gratefully thank the EU for the financial support. More information on DIRAC and AMIDA is available from the project website www.diracproject.org and www.amiproject.org. The authors acknowledge Igor Szoke, Brno University of Technology and Chuck Wooters, ICSI, Berkeley, USA for their help.

Références

[1] J.G. Wilpon, Miller L.G., and P. Modi, "Improvements and Applications for Keyword Recognition using Hidden Markov Modeling Techniques," *Proc. of ICASSP*, pp. 309–312, 1991.

[2] Wessel F., Schlter R., Macherey K., and Ney H., "Confidence Measures in Large Vocabulary Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 3, March 2001.

[3] Evermann G. and Woodland P.C., "Large vocabulary decoding and confidence estimation using word posterior probabilities," *Proc. of ICASSP*, vol. 3, pp. 1655–1658, 2000.

[4] D. Vergyri et al., "The SRI/OGI 2006 Spoken Term Detection System," *Proc. of Interspeech*, 2007.

[5] D. Miller et al., "Rapid and Accurate Spoken Term Detection," *In Proc. of NIST Spoken Term Detection Workshop (STD 2006)*, Dec 2006.

[6] L.R Rabiner and B.H Juang, "An Introduction to Hidden Markov Models," *IEEE Signal Processing Magazine*, vol. 3, no. 1, pp. 4–16, Jan 1986.

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

[8] M.D Richard and R.P Lippmann, "Neural Network Classifiers Estimate Bayesian *a posteriori* Probabilities," *Neural Computation*, vol. 3, pp. 461–483, 1991.

[9] H. Bourlard and N. Morgan, *Connectionist Speech Recognition – A Hybrid Approach*, Kluwer Academic Publishers, 1994.

[10] "SRILM - The SRI Language Modeling Toolkit," *http ://www.speech.sri.com/projects/srilm*.

[11] "NIST Spoken Term Detection Evaluation," *http ://www.nist.gov/speech/tests/std*, 2006.

[12] Szoke Igor et al., "BUT System for NIST Spoken Term Detection 2006 - English," *In Proc. of NIST Spoken Term Detection Workshop (STD 2006)*, 2006.