AN EVALUATION OF LATTICE SCORING USING A SMOOTHED ESTIMATE OF WORD ACCURACY

Mohamed Kamal Omar and Lidia Mangu

IBM T. J. Watson Research Center
Yorktown Heights, NY 10598, USA
mkomar,mangu@us.ibm.com

ABSTRACT

This paper describes a novel approach for estimating the best hypothesis of a given word lattice, the hypothesis lattice, using another word lattice, the reference lattice, and its application to large vocabulary automatic speech recognition. This approach selects the word sequence in the hypothesis lattice which maximizes a smoothed estimate of the word accuracy with respect to the reference lattice. It is shown in the paper that two algorithms similar to the Viterbi and the forward-backward algorithms can be used to estimate the hypothesis which approximately maximizes this objective function. We present in this paper two setups to test the performance of our approach. In the first setup, only one lattice is used as both the reference and the hypothesis lattices. In the second setup, two lattices produced by different systems are used to calculate the best hypothesis. In each setup, we test our approach on two Arabic broadcast news speech recognition tasks. Compared to the baseline results, up to 2.1% relative improvement in the word error rate (WER) is obtained by using our approach.

Index Terms— Lattice scoring, confusion network, ASR decoding

1. INTRODUCTION

In automatic speech recognition systems, the maximum *a posteriori* probability (MAP) is the standard decoding criterion. The hypothesis selected using the MAP criterion minimizes an estimate of the sentence-level error. However, speech recognition systems are evaluated based on their word error rate (WER) not the sentence-level error. This motivates selecting a hypothesis which minimizes an estimate of the word error rate instead of the one maximizing the sentence-level posterior probability.

In large vocabulary speech recognition systems, it is commonly the case that word lattices are used as a compact representation of the alternative hypotheses produced by the decoder. Word lattices provide more accurate representation of the search space compared to other forms like N-best lists. However, calculating pair-wise word error rates for different hypotheses in the lattice is computationally infeasible and therefore many algorithms were developed to minimize an estimate of the word error rate in a computationally feasible way. This problem was addressed in [1], and an algorithm was described to carry out a practical approximate word error minimization on word lattices. This is achieved by finding an alignment of all the words in the lattice which identify mutually supporting and competing word hypotheses. Then a new sentence hypothesis is formed by concatenating the words with maximal posterior probability from different non-overlapping classes of words in the lattice. In [2], the problem is formulated as the problem of finding the hypothesis which minimizes a Bayesian risk function related to the word error rate. To avoid the computational problems associated with estimating this hypothesis, many approaches to segment the word lattice to non-overlapping regions are described. A frame-based error rate was introduced in [3], and it was shown that it is closely correlated with the word error rate. This was used to avoid the need of dynamic programming alignment to calculate an estimate of the pair-wise word error rate.

In this paper, we describe an approach to find the hypothesis in a lattice which maximizes an estimate of the expectation of the word accuracy of this hypothesis lattice with respect to the same lattice or another reference lattice. Contrary to most previous approaches, the approach proposed here uses a non-Bayesian approximation of the word error rate which allows using one lattice or two lattices generated by different systems to estimate the best hypothesis without making any changes to the algorithm. Our approach represents an intermediate choice between calculating an approximate word-based estimate of word errors [1], and [2], and calculating a frame-based estimate of word errors [3]. This is achieved by taking the phonetic similarity between words into consideration while estimating word errors. We approximate the word accuracy of one lattice with respect to another by an estimate of the expected value of the word accuracy using the joint posterior probability mass function of the hypothesis and the reference word sequences. Using this formulation, we propose an algorithm based on the Viterbi algorithm to estimate the best hypothesis. We propose also another algorithm similar to the forward-backward algorithm to estimate the conditional values of our estimate of the word accuracy given the word arc. These values are then used by an algorithm similar to the confusion network (CN) algorithm, [1], to generate the best hypothesis. It is interesting to note that if the smoothed approximation of the pair-wise word error is replaced by a zero-one function and the same lattice was used as both the reference and the hypothesis lattices, we get the same word sequence obtained by MAP scoring. Also if the hypothesis word sequences are assumed to have the same posterior probability, then the objective function is reduced to the objective function used by many previous approaches [1], and [2].

In the next section, we will formulate the problem and describe our objective criterion. In Section 3, the algorithms used in estimating the best hypothesis based on our objective criterion are described. The experiments performed to evaluate the performance of our approach are described in Section 4. Finally, Section 5 contains a discussion of the results and future research. We will use capital letters to represent random vectors and the corresponding small letters to represent their realizations.

2. PROBLEM FORMULATION

In this section, we will discuss how a computationally feasible approximation of the word error rate can be derived. This approximation is based on an estimate of the expected word accuracy of the hypothesis lattice with respect to the reference lattice over the joint posterior probability mass function of the reference and the hypothesis word sequences. Then we show how the problem can be reduced to estimating the word sequence which maximizes the product of the posterior probability of the hypothesis word sequence and a smoothed approximation of the average word accuracy of the hypothesis word sequence with respect to all possible reference sequences in the reference lattice. We describe also an alternative approximation by using the conditional value of the objective function given the word arc as a measure of the accuracy of the arc.

Given two lattices, a measure of the word accuracy of the hypothesis lattice with respect to the reference lattice can be approximated by

$$E_{P(H,R|Y)}[A(h,r)] \approx \sum_R \sum_H P(r|Y=y)Q(h|Y=y)\hat{A}(h,r), \quad (1)$$

where $E_{P(H,R|Y)}[.]$ is the expected value over the joint probability mass function (PMF) of the hypothesis word sequence, H, and the reference word sequence, R, given the observation vector Y, $A(h,r)$ is the word accuracy of h with respect to r, $P(r|Y=y)$ is the posterior probability of the reference string r estimated from the reference lattice, $Q(h|Y=y)$ is the posterior probability of the hypothesis string h estimated from the hypothesis lattice, and $\hat{A}(h,r)$ is a smoothed approximation of the word accuracy of h with respect to r which takes phonetic similarity into consideration.

Our goal is to select the word sequence in the hypothesis lattice which maximizes our estimate of the word accuracy with respect to the reference lattice as given in Equation 1. We call this approach the maximum smoothed word accuracy (MSWA) approach. To achieve this goal, the selection rule can be written as

$$h^* = \arg\max_h Q(h|Y=y) \sum_R P(r|Y=y)\hat{A}(h,r), \quad (2)$$

where h^* is the hypothesis word sequence which maximizes our objective function. This word sequence can be estimated using the Viterbi algorithm as will be discussed in the next section.

Alternatively we can assign to each word arc, w, in the hypothesis lattice, the conditional value of the objective function in Equation 1 given this word arc, i.e.

$$\tilde{\gamma_w} = \sum_R \sum_{H:w\in h} P(r|Y=y)Q(h|Y=y)\hat{A}(h,r). \quad (3)$$

These values can be estimated using an algorithm similar to the forward-backward algorithm. We will discuss in the next section how the confusion network (CN) algorithm, [1], can be used to find the best word sequence in the hypothesis lattice using these values in Equation 3 instead of the posterior probabilities of the word arcs.

3. IMPLEMENTATION

In this section, we present our implementation of the two approaches described in the previous section to use the objective function in Equation 1 to select the best word sequence from the hypothesis lattice. First we describe the MSWA algorithm to find the word sequence in the hypothesis lattice which maximizes our objective function according to Equation 2 by using the Viterbi algorithm. Then we will describe the MSWA-CN algorithm, which is based on the CN algorithm, to estimate the best word sequence.

The estimation of the word accuracy, $A(h,r)$, of the word sequence h with respect to the word sequence r involves the calculation of the Levenshtein distance between the two sequences. The Levenshtein distance is defined as the number of substitutions, deletions, and insertions in h with respect to r conditioned on an alignment of the two sequences that minimizes a weighted sum of these error types. To avoid the computational infeasibility of calculating the pair-wise Levenshtein distance between each two possible paths in the reference and the hypothesis lattices, we condition the alignment of the two paths and therefore the calculation of their Levenshtein distance on the segmentation of the hypothesis path. Instead of counting the different types of word-level errors, we use an approximate measure of word accuracy which takes phonetic similarity into consideration. The approximate measure of the accuracy of a word arc, w, in the hypothesis lattice with respect to a path, g, which may start or end in the middle of an arc in the reference lattice such that it coincides with w in time, i.e. both have the same starting and ending times, is

$$\hat{A}(w,g) = \max_{u\in g} \sum_{i=1}^{N} \hat{A}_p(w_i,u_i), \quad (4)$$

where $\hat{A}_p(w_i,u_i)$ is an estimate of the accuracy of the phone w_i with respect to u_i, w_i is the ith phone of the word w, u_i represents the part of the word arc u which coincides with w_i in time, and N is the number of phones in w.

The approximate phone-level accuracy, $\hat{A}_p(w_i,u_i)$, is given by

$$\hat{A}_p(w_i,u_i) = \max_{q\in u_i} d(w_i,q), \quad (5)$$

where q is one of the phones in u_i, and $d(w_i,q)$ is given by

$$d(w_i,q) = \begin{cases} -1 + \frac{4e_{iq}}{l_i+l_q} & \text{if } w_i = q \\ -1 + \frac{2e_{iq}}{l_i+l_q} & \text{if } w_i \neq q \end{cases} \quad (6)$$

where l_i is the length of w_i in frames, l_q is the length of q in frames, e_{iq} is the overlap between w_i and q in frames.

We used two approaches to estimate the hypothesis word sequence which will approximately maximize our objective function. For both approaches, the forward-backward algorithm has to be applied to the reference lattice and the state sequence for each arc in the reference lattice has to be known. These two requirements allow us to calculate the forward probabilities, α_g, the backward probabilities, β_g, and the posterior probabilities, γ_g, for any path g in the reference lattice which may start or end in the middle of an arc. In the first approach, we used the Viterbi algorithm to estimate the word sequence given by Equation 2. The steps of the Viterbi algorithm are

1. Initialization: For each starting arc in the hypothesis lattice, $w_s \in S$,

$$\alpha_{w_s} = P(w_s)^k, \quad (7)$$

$$\zeta_{w_s g} = \hat{A}(w_s,g) \quad \forall g \in G_{w_s}, \quad (8)$$

where $P(w_s)$ is the likelihood of w_s, k is the acoustic weight, G_{w_s} is the set of paths in the reference lattice which coincides with w_s in time.

2. Forward Propagation: The update equations of the Viterbi algorithm for each non-starting arc, $w \notin S$, is

$$\zeta_{wg} = \hat{A}(w,g) + \frac{\sum_{q\in Q_g} \alpha_q t_{qg} \zeta_{v^* q}}{\sum_{q\in Q_g} \alpha_q t_{qg}} \quad \forall g \in G_w, \tag{9}$$

$$v^* = \arg\max_{v\in V_w} \sum_{g\in G_w} \gamma_g \frac{\sum_{q\in Q_g} \alpha_q t_{qg} \alpha_v t_{vw} \zeta_{vq}}{\sum_{q\in Q_g} \alpha_q t_{qg}}, \tag{10}$$

$$Prec[w] = v^*, \tag{11}$$

$$\alpha_w = \alpha_{v^*} t_{v^* w} P(w)^k, \tag{12}$$

where Q_g is the set of paths in the reference lattice which precedes the path g, V_w is the set of arcs preceding w in the hypothesis lattice, and G_w is the set of paths in the reference lattice which coincides with w in time, $Prec[w]$ is the best preceding word of w, α_q is the forward probability for the path q, t_{qg} is the reference lattice transition probability derived from the language model, and if g starts in the middle of an arc in the reference lattice, $t_{qg} = 1$.

3. Backtracking:

   (a) Set $i = 0$,

$$w_i = \arg\max_{w_e\in E} \gamma_{w_e} \sum_{g\in G_{w_e}} \gamma_g \zeta_{w_e g}, \tag{13}$$

   where E is the set of ending arcs in the hypothesis lattice.

   (b) While w_i is not a starting arc, i.e. $w_i \notin S$
   - $w_{i+1} = Prec[w_i]$
   - $i = i + 1$

   End.

4. Set $L = i$ and exit with the output word sequence $\{w_L, w_{L-1}, \ldots w_0\}$.

The second approach is based on the idea of assigning to each arc in the hypothesis lattice the value of the objective function conditioned on this arc as given by Equation 3. To estimate these values, we use an algorithm similar to the forward backward algorithm. The algorithm is very similar to the previous MSWA algorithm used in the first approach. But it replaces the maximization over previous arcs in the hypothesis lattice by the sum over all previous arcs, and makes a backward propagation step as well as the forward propagation step. The steps of the algorithm are

1. Initialization:

   - For each starting arc in the hypothesis lattice, $w_s \in S$,

$$\alpha_{w_s} = P(w_s)^k, \tag{14}$$

$$\zeta_{w_s g} = \hat{A}(w_s, g) \quad \forall g \in G_{w_s}. \tag{15}$$

   - For each ending arc in the hypothesis lattice, $w_e \in E$,

$$\beta_{w_e} = 1, \tag{16}$$

$$\eta_{w_e g} = 0 \quad \forall g \in G_{w_e}. \tag{17}$$

2. Forward Propagation: The update equations of the forward propagation part of the algorithm for each non-starting arc, $w \notin S$, is

$$\zeta_{wg} = \hat{A}(w,g) + \frac{\sum_{q\in Q_g} \sum_{v\in V_w} \alpha_q t_{qg} \alpha_v t_{vw} \zeta_{vq}}{\sum_{q\in Q_g} \alpha_q t_{qr} \sum_{v\in V} \alpha_v t_{vw}} \quad \forall g \in G_w, \tag{18}$$

$$\zeta_w = \gamma_w \sum_{g\in G_w} \gamma_g \zeta_{wg}, \tag{19}$$

$$\alpha_w = \sum_{v\in V_w} \alpha_v t_{vw} P(w)^k, \tag{20}$$

3. Backward Propagation: The update equations of the backward propagation part of the algorithm for each non-ending arc, $w \notin E$, is

$$\eta_{wg} = \frac{\sum_{f\in F_g} \sum_{b\in B_w} \beta_f t_{gf} \beta_b t_{wb} (\eta_{bf} + \hat{A}(b,f))}{\sum_{f\in F_g} \beta_f t_{gf} \sum_{b\in B_w} \beta_b t_{wb}} \quad \forall g \in G_w, \tag{21}$$

$$\eta_w = \gamma_w \sum_{g\in G_w} \gamma_g \eta_{wg}, \tag{22}$$

$$\beta_w = \sum_{b\in B_w} \beta_b t_{wb} P(b)^k, \tag{23}$$

   where F_g is the set of paths in the reference lattice which follows the path g, B_w is the set of arcs following w in the hypothesis lattice, β_f is the backward probability for the path f, t_{gf} is the reference lattice transition probability from g to f derived from the language model, and if g ends in the middle of an arc in the reference lattice, $t_{gf} = 1$.

4. For each word arc w in the hypothesis lattice,

$$\tilde{\gamma_w} = \zeta_w + \eta_w \tag{24}$$

The confusion network algorithm [1] is then used to find the best path in the hypothesis lattice after replacing the word posterior probability in the original algorithm with these conditional values of the objective function, $\tilde{\gamma_w}$. Therefore we call this second approach the MSWA-CN approach.

System	RT04	BNAT05
Baseline	14.6	15.5
CN	14.5	15.4
MSWA	14.5	15.4
MSWA-CN	14.3	15.2

Table 1. Word error rates (%) on the Arabic RT04 and BNAT05 evaluation data using the unvowelized system.

System	RT04	BNAT05
Baseline	12.9	13.9
CN	12.8	13.9
MSWA	12.8	13.9
MSWA-CN	12.7	13.8

Table 2. Word error rates (%) on the Arabic RT04 and BNAT05 evaluation data using the vowelized system.

System	RT04	BNAT05
Baseline	12.9	13.9
MSWA	12.9	13.9
MSWA-CN	12.7	13.8

Table 3. Word error rates (%) on the Arabic RT04 and BNAT05 evaluation data using an unvowelized reference lattice and a vowelized hypothesis lattice.

4. EXPERIMENTS AND RESULTS

This section gives the experimental results of applying the two approaches described in the last section on the tasks of the Arabic DARPA 2004 Rich Transcription (RT04) evaluation data, which consists of 3 shows of 25 minutes each, and the 2005 broadcast news Arabic test set (BNAT05) which consists of 12 shows of 30 minutes each from 5 different sources provided by BBN. We use two systems in our experiments: an unvowelized system and a vowelized system. The main difference between the two systems is the explicit modeling by the vowelized system of the short vowels which are pronounced in Arabic but almost never transcribed. For both systems, each phoneme is represented by 3 HMM states with left-to-right topology with the exception of modeling short vowels with 2 states in the vowelized system. For both systems, the raw features are 13-dimensional PLP features computed every 10 ms. from 25-ms. frames. The recognition features are computed by splicing together 9 frames of raw features, projecting the spliced features to 40 dimensions using LDA, and then applying maximum likelihood linear transformation (MLLT) to the projected features. Both systems use a pentaphone acoustic context and comprise 5K context dependent states and 400K Gaussians. Both systems are trained with a combination of fMPE and MPE on 135 hours of supervised data and 1800 hours of unsupervised data.

In the context of speaker-adaptive training, both systems use vocal tract length normalization (VTLN), and feature-space MLLR. A single pass of MLLR adaptation, using a regression tree to generate transforms for different sets of mixture components, is also done. The language model is a 617K vocabulary 4-gram LM with 56M n-grams trained with modified Kneser-Ney smoothing. The vowelized system is a cross-adapted system as the transcripts generated by the unvowelized system were used to train the speaker-adaptive transforms for the vowelized system.

We tested our Viterbi-based MSWA algorithm and the MSWA-CN algorithm using three different setups. In the first setup the lattices produced by the unvowelized system were used as both the reference lattice and the hypothesis lattice. In the second setup, the lattices produced by the vowelized system were used as both the reference lattice and the hypothesis lattice. In the third setup, the lattices produced by the vowelized system were used as the hypothesis lattice, while the lattices produced by the unvowelized system were used as the reference lattice. Due to the different phoneme set of the unvowelized system, we did not score phonemes which exist in the vowelized system but not in the unvowelized system in the third setup during the estimation of the approximate word accuracy.

As shown in Table 1, the WER results improved by 2.1% relative compared to the baseline by using the MSWA-CN algorithm. The MSWA-CN algorithm outperforms using either the Viterbi-based MSWA algorithm or the CN algorithm by itself. Although the gain is small but it is consistent over the two test databases.

The results in Table 2 show that the gain, if any, compared to the baseline of any of the three algorithms is very small on both the Arabic RT04 and BNAT05 test data. However, still the gain from the MSWA-CN algorithm outperforms using either the Viterbi-based MSWA algorithm or the CN algorithm by itself.

Finally, the results in Table 3 show that there is no gain obtained from using both the unvowelized and the vowelized lattices compared to using the vowelized lattices alone. This can be attributed to the facts that the difference in performance between the vowelized and the unvowelized systems is large, the vowelized system is a cross-adapted system which uses the output of the unvowelized system, and that phonemes like short vowels are modeled in the vowelized system but not modeled in the unvowelized system.

5. DISCUSSION

In this paper, we examined using a new smoothed word accuracy (SWA) objective function for lattice scoring. We described two algorithms which use this objective function to estimate the best hypothesis in a given lattice: the Viterbi-based MSWA algorithm and the MSWA-CN algorithm. The results reported in the paper show that small consistent improvements are achieved by using the MSWA-CN algorithm compared to the MSWA and the CN algorithms.

We plan to investigate the performance of our approach on systems based on significantly different models of comparable performance. We also intend to assess the usefulness of the conditional score in Eq. 3 for confidence annotation.

6. REFERENCES

[1] 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*, 14(4):373–400, 2000.

[2] V. Goel, S. Kumar, and W. Byrne, Segmental Minimum Bayes-Risk Decoding for Automatic Speech Recognition Systems, *IEEE Trans. Speech and Audio Proc.*, 14(1):356–357, January 2006.

[3] F. Wessel, R. Schluter, and H. Ney, Explicit Word Error Minimization Using Word Hypothesis Posterior Probabilities, *Proc. of ICASSP*, Salt Lake City, pp. 33–36, 2001.