ACOUSTIC AND WORD LATTICE BASED ALGORITHMS FOR CONFIDENCE SCORES

Daniele Falavigna, Roberto Gretter [*]

ITC-irst, Povo di Trento, ITALY
{falavi,gretter}@itc.it

Giuseppe Riccardi

AT&T Labs-Research, Florham Park, NJ, USA
dsp3@research.att.com

ABSTRACT

Word confidence scores are crucial for unsupervised learning in automatic speech recognition. In the last decade there has been a flourish of work on two fundamentally different approaches to compute confidence scores. The first paradigm is acoustic and the second is based on word lattices. The first approach is data-intensive and it requires to explicitly model the acoustic channel. The second approach is suitable for on-line (unsupervised) learning and requires no training. In this paper we present a comparative analysis of off-the-shelf and new algorithms for computing confidence scores, following the acoustic and lattice-based paradigms. We compare the performance of these algorithms across three tasks for small, medium and large vocabulary speech recognition tasks and for two languages (Italian and English). We show that word-lattice based algorithm provides consistent and effective performance across automatic speech recognition tasks.

1. INTRODUCTION

Word confidence scores are crucial for unsupervised learning in automatic speech recognition. In the last decade work to compute confidence scores has been flourishing while proceeding along two fundamentally different approaches. The first paradigm is acoustic and the second is based on word lattices. The first approach is data-intensive and it aims at modeling the acoustic channel. The second approach is suitable for on-line (unsupervised) learning and requires no training.

The acoustic paradigm for computing word confidence score is based on acoustic measurements [1, 5]. In this case the processing of the spoken utterance is a two-pass algorithm. In the first step the best word hypotheses is computed and then it is rescored to compute the confidence scores for each word in the best hypotheses. The first pass uses standard acoustic models, while the second has a different set of acoustic models that normalize log-likelihood functions [1, 5].

In the lattice based approach the confidence scores are computed, on-line, with a single pass. On-line computation requires no ad-hoc models for rescoring word hypotheses and is easily portable across tasks and acoustic channels. While there have been many algorithm proposed for this approach, in this paper we will analyze the performance of two lattice based algorithms. The first is an off-the-shelf algorithm to compute word posteriors for a lattice structure called *sausage* [2]. While such topology is not necessary for computing posterior probabilities we have shown that they are effective predictors for word accuracy [4]. The second algorithm is a new algorithm which is not constrained to specific graph topology and is based on clustering word alignments into time slots.

In section 2 we will describe the confidence score algorithms both for the acoustic and lattice based approach. In section 3 a short description of the tasks and databases will be presented, while in section 4 the results of the various experiments will be given. Finally, in section 5 we will discuss the results.

2. ALGORITHMS FOR CONFIDENCE SCORES

The algorithms we are going to compare belong to two families. The first one uses the log-likelihoods, computed during the Viterbi decoding step, for each one of the phones of the best sentence. Then, in a second step, the corresponding log-likelihood ratios, using some *anti-models* are computed. Confidence scores result by summing the log-likelihoods of the phones of each word. The second family computes the confidence scores starting from a word lattice, which is one of the possible outputs of a speech recognizer. A word lattice is a connected graph, where each state has a time information and each arc represents a word that has been hypothesized during the decoding. Each arc has a score, which comes from the combination of the acoustic and language models; the topology of the graph reflects the constraints of the language model.

2.1. Log-likelihood ratio

Given a word W in the best recognized string and the corresponding acoustic observation sequence of cepstral vectors O, we can perform a statistical hypothesis test by considering the ratio between two probabilities, i.e. $P[H_0 \mid O]$ and $P[H_1 \mid O]$. H_0 is called *null* hypothesis and represents, in our case, the probability of word W given O, H_1 is called *alternative* hypothesis and represents the probability of an event complementary to W (i.e. "all that is not W"). We can expand the hypothesis test ratio as follows:

$$\frac{P[H_0 \mid O]}{P[H_1 \mid O]} = \frac{P[O \mid H_0]}{P[O \mid H_1]} \times \frac{P[H_0]}{P[H_1]}$$

We call *likelihood ratio* the quantity $\frac{P[O|H_0]}{P[O|H_1]}$ in the equation above. For the given word W the corresponding likelihood ratio $S[O \mid W]$ is evaluated as follows:

$$S[O \mid W] = \frac{P[O \mid H_0]}{P[O \mid H_1]} = \frac{P[O \mid W]}{P[O \mid \widetilde{W}]}$$

where $P[O \mid W]$ is the word likelihood estimated during the forward step, $P[O \mid \widetilde{W}]$ is estimated in a successive step, by using an appropriate complementary model $\widetilde{W}$ of W. The work in [5, 1] proposes to use phone *anti-models* for estimating $P[O \mid$

[*] This work was partly performed while the second author was visitin AT&T Labs Research

$\widetilde{W}$]. Basically, for each phone HMM, ph_i, a corresponding anti-HMM, $\widetilde{ph_i}$, is estimated that should account for all the acoustic observations not generated by ph_i. Hence, the likelihood ratio $S[O \mid W]$ is evaluated by summing the phone likelihood ratios $S[O \mid ph_i] = \frac{P[O|ph_i]}{P[O|\widetilde{ph_i}]}$ of the phones in W. In practice, the phone log-likelihood ratios are taken, i.e. $log(S[O \mid ph_i]) = log(P[O \mid ph_i]) - log(P[O \mid \widetilde{ph_i}])$, while several measures (i.e. not only the phone log-likelihhod ratios summation) could be used for estimating the whole word likelihood ratio, as proposed in [6].

For estimating phone log-likelihood ratios, we carried out several experiments with different sets of phone anti-models. More specifically, given phone HMM ph_i, the corresponding anti-model, $\widetilde{ph_i}$, can been constructed as follows.

- $\widetilde{ph_i}$ is trained directly on our training databases.
- Transition probabilities of $\widetilde{ph_i}$ are the same of ph_i; output density probabilitiy functions are evaluated as a linear combination of all the output probability densities of all the available phone unit HMMs, except the ones of ph_i:

$$\begin{aligned} b_k^{\widetilde{ph_i}}(\cdot) \quad &= \textstyle\sum_{ph_j \neq ph_i} \sum_{q=1}^{Q} c_{kq}^{ph_j}\, g_{kq}^{ph_j} \quad (\mu_{kq};\sigma_{kq}) \\ & \qquad\qquad\qquad\qquad 1 \leq j \leq J \end{aligned}$$

  where, $b_k^{\widetilde{ph_i}}(\cdot)$ is the Gaussian mixture output density associated to state k of $\widetilde{ph_i}$, J is the total number of HMMs, $g_{kq}(\cdot;\cdot)$ is the q^{th} gaussian density component in state k, Q is the total number of gaussian mixture components for state k. In this case the training phase consists in estimating the mixture coefficients c_{kq}.

- $\widetilde{ph_i}$ is represented by a grammar defined with parallel transitions corresponding to all the available phone unit HMMs, except the one corresponding to ph_i:

$$\widetilde{ph_i} = (ph_1| \dots |ph_{i-1}|ph_{i+1}| \dots |ph_I)$$

In the experiments reported below we have used the third type of anti-models described above.

2.2. Sausages posterior/entropy

Recently, an algorithm has been proposed [2] for converting a word graph to a compact format, called a *sausage*. A sausage is a simplified graph with a particular topology: it turns out to be a sequence of confusion sets, each one being a group of words, which may include a null word (*eps*), competing in the same (with some approximation) time interval. Each word has a posterior probability, which is the sum of the probabilities of all the paths of that word occurrence in the graph. In each confusion set, the sum of all posteriors equals 1. In a sausage the time order is preserved, but time information is lost. The main motivation of this algorithm is that of minimizing the Word Error Rate (WER), instead of the Sentence Error Rate (SER).

We briefly review Mangu's algorithm. It takes a word graph as input and goes through the following steps:

• low probability links of the graph are pruned;

• a posterior probability for each link of the graph is computed;

• a "temporal" order over the states of the graph is found;

• different occurrences of the same word in the "same" time interval are merged (intra - word clustering stage) and their posteriors summed;

• words which compete in the "same" time interval are grouped together to form a confusion set (inter - word clustering stage).

It is straightforward to extract from a sausage what is called the *Consensus Hypothesis*, which is the word sequence obtained by picking up from each confusion set the word with the highest posterior. This sequence can differ from the best path hypothesis, i.e. the optimal word sequence inside the graph. The Consensus Hypothesis is said to have a better WER of the best path hypothesis, and experiments on the HMIHY task ([3, 4]) confirmed this. As previously said, each word of the sausage (and in particular the words of the Consensus Hypothesis) has an associated posterior probability.

Another quantity that can be used as a confidence score is a local entropy, computed on each confusion set:

$H = -\sum_{i=1}^{N} post(w_i) \times log(post(w_i))$

where N is the number of competing hypotheses and $post(w_i)$ is the posterior of word w_i. This measure includes more information than the previous one, because it takes into account not only the posterior of the winning word, but also the distribution of the posteriors of the competing words.

A previous work [4] showed the effectiveness of these two quantities, which behave similarly as confidence scores. Both posterior and local entropy give a measure of the acoustic / linguistic confusion found during the decoding stage; in this sense, they can identify speech segments with problematic speech recognition.

In the following experiments, only posteriors computed on sausages will be considered confidence scores.

2.3. Lattice posterior/entropy

The algorithm proposed here tries to get a confidence score directly from the word graph, without transforming it into a sausage. A word graph is shown in figure 1, and reflects the syntactic constraints of the grammar used during recognition. It is made of states and transitions connecting them: each transition corresponds to a recognized word (@BG means silence) and has the following information associated with it:

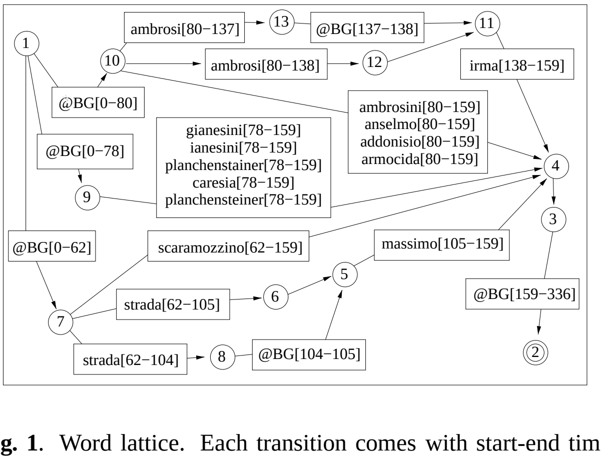

Fig. 1. Word lattice. Each transition comes with start-end time frames.

• first and last time frame, as resulting from Viterbi alignment (appended to each word and surrounded by square brackets in figure 1),

• the local likelihood, i.e. the sum of the acoustic and language model contribution for that word in that temporal segment, scaled down by a factor that approaches the language model weigth.

By applying the forward-backward algorithm to the word graph and combining the local likelihoods, the posterior probability for each transition is computed. At this point, the posterior of a transition could be used as a confidence score by itself, but some improvement is possible by taking into account the competing hypotheses in the same time slot.

We define time slot $ts(T)$ of transition T the speech interval between the starting and ending time frames of T, regardless of graph topology. Each transition overlapping $ts(T)$ is a *competitor* ot T, but competitors having the same word label as T are *allies*. Figure 2 shows all the competitors / allies for a word of the best path. We sum all the posteriors of the allies of transition T and we obtain what we call the posterior of word w. The rationale of this operation is to try to avoid to miss the important contribution of the same word, often due to very small differences in alignment. To illustrate this, have a look at the two transitions labeled *ambrosi*, on the top of figure 1, and notice that they differ only because they belong to two paths, one of which has the insertion of a small silence (@BG) before the following word. It is clear that in this case, considering them as allied and summing their contributions will result in a better estimate of the posterior of the *word* ambrosi. Note that a similar approximation is contained in Mangu's algorithm, where transitions having the same word label belonging to the same competitor set are merged.

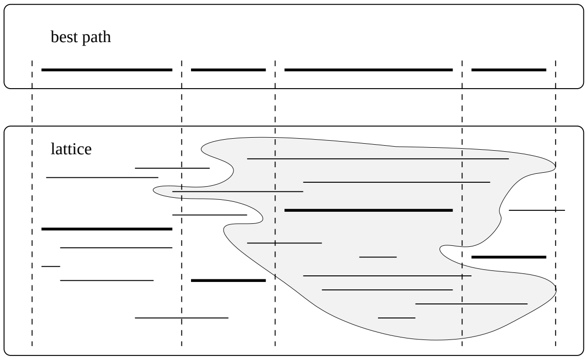

Fig. 2. Transitions belonging to the same time slot (competitors / allies).

Another quantity that can be used as a confidence score is a local entropy, computed on the list of competitors in a time slot:

$$H = -\sum_{w_i \in ts(T)} post(w_i) \times log(post(w_i))$$

where $ts(T)$ is the time slot corresponding to a transition of the best path and $post(w_i)$ is the sum of the posterior of all the competitors / allies having the same word label (each transition is counted only once inside a $ts(T)$).

In the following, both Word Posterior and Word Entropies computed on word graphs will be considered as confidence scores.

3. TASKS AND DATABASES

To test and compare the various approaches described above we have used speech databases collected, on the field, during the interactions of users with some automatic services. The Italian databases have been acquired by means of two different services developed by ITC-irst: the first one is an automatic switchboard service, the second one is an automatic service to access tourism information. The American English database has been acquired within the "How May I Help You" project, hereinafter called *HMIHY* [3].

The first Italian database, hereinafter called *CCC*, consists of 1781 speech files, each containing an isolated utterance of a person name or of a city name. The second Italian database has been collected by means of a mixed-initiative dialog service [7] and consists of 3635 files containing requests for accessing tourism information, expressed in natural language (e.g. "I want to know the addresss of a three star hotel in Val di Fassa", etc.). Hereinafter we will call this database *APT*.

The *HMIHY* database is a collection of human-human interactions within a customer case task. The user utterances are responses to the *How May I Help You?* prompt. The average length of the speech transcriptions is 39. For our experiments we have used this corpus for testing purpose only (word lattice algorithms).

4. EXPERIMENTS

As explained above, we have compared four different methods for evaluating confidence scores, i.e. Log-likelihood Ratios (LR), Word Posterior (WP) probabilities, Word Entropies (WE) and Sausages posterior (Saus). For the Italian tasks, *CCC* and *APT*, we have used loop transition recognition grammars. For task *CCC* the number of transitions in the grammar is 7136, for task *APT* the number of transitions is 1000. Word recognition accuracies obtained on these tasks are 85.6% for *CCC* and 92.1% for *APT*. For the English task *HMIHY*, test set has 1K utterances and the baseline large vocabulary word accuracy is 61.2%. In the reported experiments we compare the following quantities: Receiving Operating Curves (*ROC*), Equal Error Rate (*EER*) and Minimum Error (*ME*). *EER* and *ME* are defined as:

$$\begin{aligned} EER &= \frac{\%FR + \%FA}{2} \qquad if \quad \%FA = \%FR \\ ME &= min_{\{thr\}}(\%FR + \%FA) \end{aligned}$$

where %FA is the percentage of False Acceptances (given a threshold *thr*) and %FR is the related percentage of False Rejections.

	CCC	APT	HMIHY
LR	29.77%	34.28%	
WP	20.85%		30.28%
WE	27.20%		38.60%
Saus	21.06%		25.1%

Table 1. *Equal Error Rates of confidence scores obtained on the various tasks using the four different approaches.*

For lattice based confidence scores (i.e. WP, WE and Saus) we run several experiments in order to find the best value to scale word likelihoods (this value accounts for the language model probabilities, as explained above).

	CCC	APT	HMIHY
LR	58.96%	65.97%	
WP	38.85%		57.73
WE	49.49%		38.60
Saus	40.96%		49.65%

Table 2. *Minimum Errors obtained on the various tasks.*

Table 1 shows EER otained on the various tasks using the four different approaches for evaluating confidence scores. Similarly, Table 2 shows Minimum Errors for the same tasks and approaches. ROC curves, related to the different approaches, are shown for task *CCC* in Figure 3. In the figure, the horizontal line corresponds to the percentage of False Rejections while the vertical line corresponds to False Acceptances.

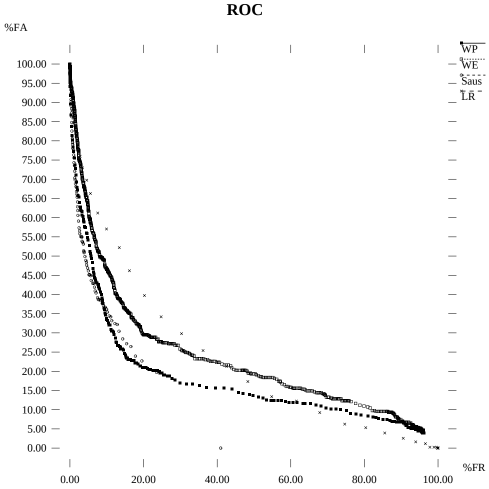

Fig. 3. ROC curve for the posterior (WP), entropy (WE), sausage (Saus) and log-likelihood ratio (LR) algorithms on the CCC task

Similar results have been obtained for task *HMIHY* as reported in Figure 4.

Best performance are obtained using word posterior probabilities, even if similar performance can be achieved with sausages (sausages themselves make use of posterior probabilities). LR scores provide worse ROC curves and they are strictly dependent of the set of anti-models trained for each task. Word entropies provide worse performance than Word posteriors. These results carry over onto the large vocabulary task such as HMIHY, where word accuracy are lower than the previous two and confidence scores are crucial for unsupervised learning or rejection mechanism.

5. CONCLUSION

In this paper we have presented a comparative analysis of off-the-shelf and new algorithms for computing confidence scores following the acoustic and lattice-based paradigm. We have compared the performances of these algorithms across three tasks for small, medium and large vocabulary speech recognition tasks and for two languages (Italian and English). When compared to acoustic algorithms, word lattice algorithms are robust toward changes in the speech recognizer and do not need channel-dependent or task-dependent training. Overall, word-lattice based algorithm provides consistent and effective performance across automatic speech recognition tasks.

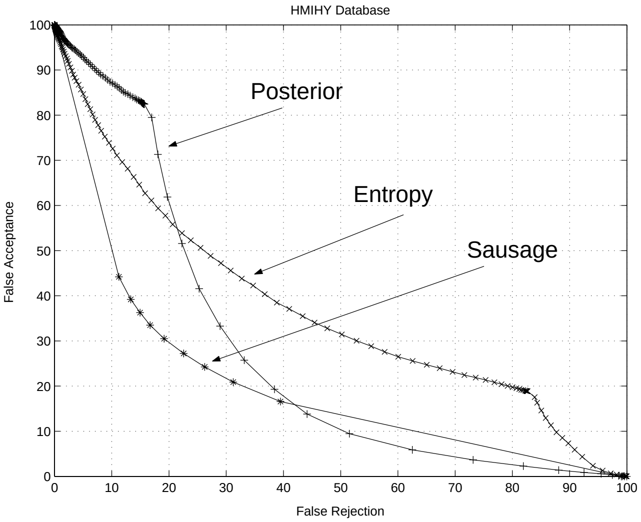

Fig. 4. ROC curve for the posterior, entropy and sausage algorithms on the HMIHY task

6. REFERENCES

[1] R. C. Rose, B. H. Juang, and C. H. Lee, "A Training Procedure for Verifying String Hypotheses in Continuous Speech Recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Detroit, USA, 1995, pp. 281–284.

[2] L. Mangu, E. Brill, and A. Stolcke, "Finding Consensus Among Words: Lattice-Based Word Error Minimization," in *Proceedings of the European Conference on Speech Communication and Technology*, Budapest, Hungary, 1999.

[3] G. Riccardi and A. L. Gorin, "Stochastic Language Adaptation Over Time and State in a Natural Spoken Dialog System," in *IEEE Trans. on Speech and Audio Proc*, vol.8, no. 1, January 2000.

[4] R. Gretter and G. Riccardi, "On-line Learning of Language Models With Word Error Probability Distributions," in *Proceedings of ICASSP*, Salt Lake City, Utah, May 2001.

[5] R.A. Sukkar and C.H. Lee, "Vocabulary Independent Discriminative Utterance Verification for Nonkeyword Rejection in Subword Based Speech Recognition," in *IEEE Trans. on SAP*, Vol. 4, No. 6, pp. 420–429, November 1996.

[6] T. Kawahara, C.H. Lee and B.H. Juang, "Flexible Speech Understanding Based on Combined Key-Phrase Detection and Verification," in *IEEE Trans. on SAP*, Vol. 6, No. 6, pp. 558–562, November 1998.

[7] C. Barbero, D. Falavigna, R. Gretter, M. Orlandi, E. Pianta " Some Improvements on the IRST Mixed Initiative Dialogue Technology ," in *Proceedings of TSD 2000 Workshop*, Brno (Czech Republic), pp. 351–356, September 2000.