

NEURAL - NETWORK BASED MEASURES OF CONFIDENCE FOR WORD RECOGNITION

Mitch Weintraub, Françoise Beaufays, Ze'ev Rivlin, Yochai Konig, Andreas Stolcke

Speech Technology and Research Laboratory
SRI International, Menlo Park, CA.

e-mail: mw, francois, zev, konig, stolcke@speech.sri.com

ABSTRACT

This paper proposes a probabilistic framework to define and evaluate confidence measures for word recognition. We describe a novel method to combine different knowledge sources and estimate the confidence in a word hypothesis, via a neural network. We also propose a measure of the joint performance of the recognition and confidence systems. The definitions and algorithms are illustrated with results on the Switchboard Corpus.

1. INTRODUCTION

In the last few years, a lot of research has been devoted to the development of *confidence scores* associated with the outputs of automatic speech recognition (ASR) systems. These scores were used mostly to help spot keywords in spontaneous or read texts, and to provide a basis for the rejection of out-of-vocabulary words (*e.g.* [4-11]). Many other ASR applications could also benefit from knowing the level of confidence in correct recognition. For example, text-dependent speaker recognition systems could put more emphasis on words recognized with higher confidence; unsupervised adaptation algorithms could adapt the acoustic models only when the confidence is high; human-made transcriptions could be verified by ASR systems outputting their confidence in the transcribed word sequence, etc. In addition, a measure of the global confidence in recognition, *i.e.* the recognition confidence for the entire database, could be a useful tool to compare different recognizers that have similar recognition performance.

We propose a probabilistic framework for defining and evaluating confidence measures, as well as a novel method for estimating confidence. More specifically, we describe:

- A definition of word correctness based on time-alignments.
- A definition of confidence in a word hypothesis.
- A method for combining different knowledge sources and estimating the confidence in a word hypothesis via a neural network.
- Three alternative metrics to measure the performance of confidence estimators.
- A joint performance criterion that combines the word error-rate (WER) and the confidence in word hypotheses.

The definitions and algorithms proposed in the paper are illustrated with experiments on the Switchboard Corpus [1].

2. WORD CORRECTNESS

Given a reference and a hypothesis word string, each word in the hypothesis can be labeled as correct or incorrect. Many applications (*e.g.* speaker identification, adaptation of acoustic models, ...) require a definition of *word correctness* that involves time-alignments. We provide such a definition. Namely, we say that a hypothesized word, h_i , is correct iff there exists a word in the reference string, r_j , such that (1) h_i and r_j are identical and (2) h_i and r_j are correctly time-aligned. For these two words to be correctly time-aligned, we require that (a) more than 50% of h_i overlap with r_j , (b) more than 50% of r_j overlap with h_i , and (c) no other reference word overlap by 50% or more with h_j . This last condition makes it possible to identify deletions. Examples are given in Fig.1.

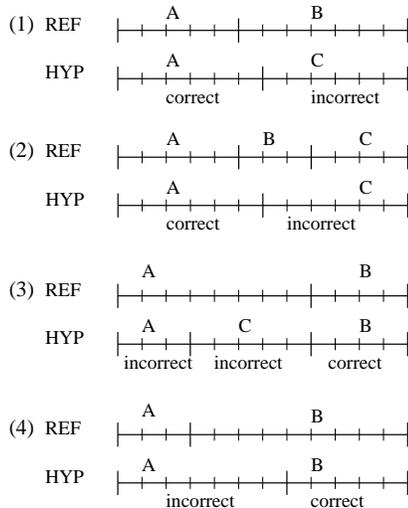


Figure 1. Examples of words labelled as correct or incorrect, according to our definition of word correctness ((1) substitution, (2) deletion, (3) insertion, (4) bad time-alignment). The small/large vertical bars indicate frame/word boundaries.

3. WORD CONFIDENCE

Given the above definition for the correctness of a word, we define the confidence in a word hypothesis as the posterior probability that the word is correct, given a set of observations relative to this word.

4. WORD CONFIDENCE METRICS

4.1. Unnormalized Confidence Metrics

The same way that a metric, the word error-rate, is traditionally used to evaluate the word recognition performance of a recognizer, a metric must be defined to evaluate the word confidence performance of a recognizer. We propose three such metrics:

- the *Mean Square Error* (MSE):

$$\frac{1}{N} \sum_w [\delta_w (1 - c_w)^2 + (1 - \delta_w) (c_w)^2],$$

- the *Cross-Entropy* (CREP):

$$\frac{1}{N} \sum_w [\delta_w \log(c_w) + (1 - \delta_w) \log(1 - c_w)],$$

- the *Classification Error-Rate* (CER):

$$\frac{1}{N} \sum_w [\delta_w (1 - t(c_w)) + (1 - \delta_w) t(c_w)],$$

where, c_w denotes the confidence in word w , δ_w denotes the word correctness (0 or 1), and $t(c_w)$ is 1 when c_w is greater than some threshold, τ , and 0 otherwise. For N large, the CER is the probability of error obtained when hard-thresholding the confidences, c_w , and classifying the words as correct or incorrect. Assuming that the costs for misclassifying correct and incorrect words are equal, we set τ to 0.5.

Accordingly, confidence measure algorithms can be developed to minimize the MSE, maximize the CREP, or minimize the CER.

4.2. Normalized Confidence Metrics

The above metrics could potentially reflect artificially good performance if the recognizer outputs incorrect hypotheses with high confidence. We address this issue by normalizing the metrics by the values they assume when estimated with the prior probability of correctness as the sole knowledge source. These normalization factors are obtained by replacing c_w with $P_c = (1 - \text{WER})$ in the above definitions, where P_c is the a-priori probability that any word is correct and WER is the word error-rate. For example, $\text{MSE}(\text{priors}) = P_c(1 - P_c)$, and $\text{CER}(\text{priors}) = 1 - P_c$ (assuming $P_c \geq (1 - P_c)$). The metrics estimated from the priors only correspond to the best performance a confidence algorithm can achieve without extra word-dependent information.

The normalized confidence metric based on the MSE can then be expressed as

$$\text{Normalized MSE} = \frac{\text{MSE}(\text{posteriors}) - \text{MSE}(\text{priors})}{\text{MSE}(\text{priors})}.$$

The normalized CREP and CER are defined similarly.

5. A JOINT PERFORMANCE METRIC

The above metrics evaluate the performance of the confidence estimator only. However, applications involving the comparison of several recognizers performing at different error-rates require a metric that characterizes the *overall* performance of the recognizer, that is a metric that combines word error-rate and word confidence.

To address this issue, we propose a new criterion that weights the word correctness by the word confidence and averages over the words:

- the *NEt Recognition Performance* (NERP):

$$\frac{1}{N} \sum_w [\delta_w c_w + (1 - \delta_w) (-c_w)].$$

With its penalization term for incorrect words ($-c_w$), the NERP intrinsically encodes the word error-rate and defeats gamesmanship scenarios.

6. COMPUTATION OF CONFIDENCE

The features that indicate the correctness of a word hypothesis are numerous and different in nature. For example, some might take on real values, some might be integers. To take advantage of this diversity, we propose to combine different features with a multi-layer neural network, as illustrated in Fig.2.

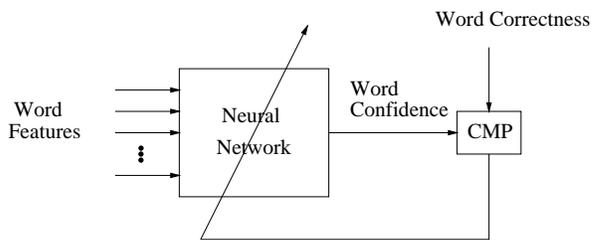


Figure 2. Neural network architecture for word confidence estimation.

The inputs to the net are the (normalized) features relative to a word, and the desired output is the word correctness: 0 or 1. Under these conditions, and assuming that the net is trained to minimize the MSE or maximize the CREP, the output of the net estimates the confidence in the word hypothesis (according to our definition) [2, 3].

6.1. Features

The performance of the neural network clearly depends on the quality of the features used as knowledge sources. In our experiments, we implemented 13 features. These are by no means exhaustive.

6.1.1. Acoustic Features

The acoustic features we implemented measure the normalized log-likelihood (LL) of the acoustic realization of each word. They differ by the models used to normalize the LLs and by the way the frame-level LLs are combined. Normalization was done either with context-independent HMMs (CLHMMs) [4] or with a Gaussian mixture model (GMM). The frame-level LLs were combined at the word level, phone level, or phone-state level. This gives six possible features of which five were implemented. For example, the phone-averaged, GMM-normalized feature is defined as:

$$\frac{1}{N_\varphi} \sum_{j=1}^{N_\varphi} \left(\frac{1}{N_{f_j}} \sum_{i=1}^{N_{f_j}} (\log P(x_i | \lambda_{H_i}) - \log P(x_i | \lambda_G)) \right)$$

where N_φ is the number of phones in the word, N_{f_j} is the number of frames in phone j , x_i is the acoustic vector corresponding to frame i of the word, and λ_{H_i} and λ_G represent, respectively, the HMM state Viterbi-decoded for frame i and the GMM.

6.1.2. Language Model Features

We implemented four features based on the language model (LM):

- Trigram LM log-probability.
- Order of the n-gram used in the LM.
- Reverse trigram LM log-probability.
- Order of the n-gram used in the reverse LM.

6.1.3. N-best List Posterior Probabilities

Based on the recognition N-best list, we computed the log posterior probability of each word given the acoustic and language models, or given only one of the two. This generated three features. The first one is defined below; the other two are similar.

$$\log P(W|X, LM) = \log \frac{\sum_{\text{HYPs} | W \in \text{HYP}} P(\text{HYP}|X, LM)}{\sum_{\text{all HYPs}} P(\text{HYP}|X, LM)},$$

where X is the acoustic realization of the word W [10].

6.1.4. Other Features

We also used the number of phones in the word as a feature, with the intuition that shorter words tend to be more often incorrect than longer words.

7. EXPERIMENTAL RESULTS

A multi-layer neural network was trained with the back-propagation algorithm to minimize the MSE of a set of training sentences (alternatively, we could have trained the network to maximize the CREP). We chose a two-hidden layer sigmoidal architecture, with 50 nodes in each hidden layer. The database used for these experiments was the NIST'95 Evaluations subset of the Switchboard Corpus. The training and testing set consisted of about 20,500 and 2,400 words, respectively.

7.1. Global Performance

In Fig.3, we compare the posteriors estimated by the neural net with the "true" posteriors estimated as a frequency interpretation of the data set. The figure shows that the data points don't depart much from the diagonal, which indicates that the true posteriors are well-estimated by the neural network. The figure also shows that the neural net made use of most of the available dynamic range (0 to 1).

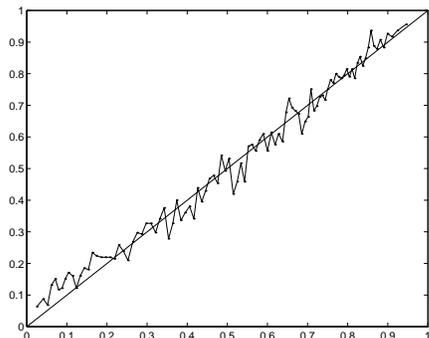


Figure 3. Posteriors estimated from the data vs. posteriors estimated by the neural network.

Table 1 compares the MSE, CREP, and CER computed from the priors only and from the posteriors outputted by the neural network. The decrease in MSE from priors to posteriors indicates that the average estimation of the word confidence improved by $(\sqrt{\text{MSE from priors}} - \sqrt{\text{MSE from posteriors}})$, which is roughly a 14% relative improvement on the test data. For the same data set, the CER decreased by 43%.

Training set ($P_c = 51.62\%$)

	MSE	CREP	CER
from priors	0.2497	-0.6926	48.38%
from posteriors	0.1838	-0.5484	27.50%

Testing set ($P_c = 51.11\%$)

	MSE	CREP	CER
from priors	0.2499	-0.6929	48.89%
from posteriors	0.1852	-0.5516	27.70%

Table 1. MSE, CREP, and CER for priors only and for the neural net outputs on the NIST'95 Evaluation subset of Switchboard.

7.2. Feature Performance

The performance of the neural network depends (1) on how well the individual features can classify words as correct or incorrect, (2) on how uncorrelated the features are.

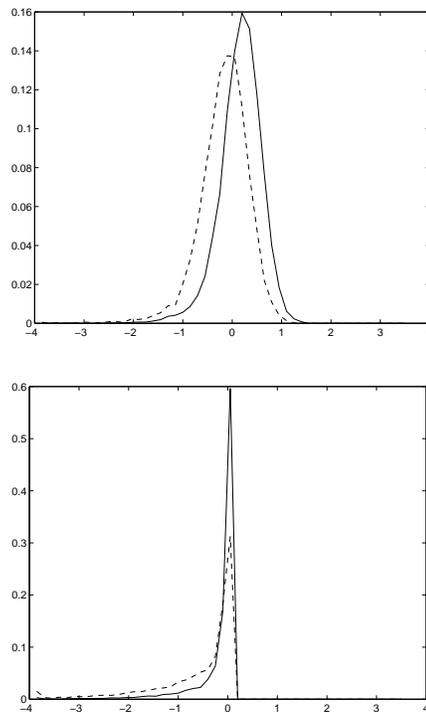


Figure 4. Pdfs of the CI-HMM-normalized, phone-averaged, LLs (first figure) and of the N-best posterior log-probabilities with acoustic and language models (second figure), for correct (solid) and incorrect (dash) words.

To illustrate the performance of the individual features, we show in Fig. 4 the probability density functions (pdf) of the scores associated with correct and incorrect words by the two best features we implemented: the CI-HMM-normalized, phone-averaged, log-likelihoods and the N-best posterior log-probability based on the acoustic and language models. Although there is a clear separation between the pdfs of correct and incorrect words, the overlap is large, meaning that even our best features are not very discriminative.

Table 2 summarizes the performance of the features on the test set. The first column indicates the feature. The second column specifies the CER if only the current feature and the prior, P_c , are used. The last 2 columns give the MSE and CER of a neural network trained with all the features *but* the current one. Thus, the second column shows the “quality” of the feature, while the last two columns show how correlated the feature is with the other features. These numbers should be compared to the global performance of the system on the same data set (second portion of Table 1).

Feature	Feature alone	NN without this feature	
	CER (%)	MSE	CER (%)
CI-norm word-LL	35.33	0.1860	27.75
CI-norm phone-LL	34.73	0.1862	27.87
GMM-norm word-LL	36.55	0.1857	27.89
GMM-norm phone-LL	36.16	0.1857	27.88
GMM-norm state-LL	37.25	0.1860	27.88
trigram LM log-P	46.16	0.1878	28.16
n-gram order in LM	46.92	0.1863	27.97
rev.-trigram LM log-P	47.15	0.1874	28.15
n-gram order rev.LM	46.83	0.1863	27.87
N-best log-P(Ac,LM)	31.38	0.1938	29.55
N-best log-P(Ac)	37.07	0.1866	27.92
N-best log-P(LM)	36.63	0.1862	27.89
num. phones	45.84	0.1883	28.15

Table 2. Feature performance on the test set of the NIST’95 Evaluation subset of Switchboard.

From Table 1 and 2, we can conclude that:

1. Any feature taken alone gave a probability of classification error lower than that obtained with the priors only.
2. The combined features (neural net output) gave better performance than any feature alone.
3. The best features according to the CER are the N-best list posterior log-probability based on the acoustic and language model, and the CI-HMM normalized acoustic log-likelihoods computed and averaged at the phone level.
4. Some features have a high CER if taken alone but contain information that is uncorrelated with the other features: *e.g.* the number of phones in the word and the trigram LM log-probabilities.

7.3. Maximizing the NERP

So far, we discussed the neural net performance in terms of MSE, CREP, and CER. As the neural network was trained to maximize the posterior probability that a word is correct (minimizing the MSE), it does not maximize the NERP. In

order to improve the NERP, we performed a preliminary experiment where we passed the confidence measure outputted by the neural network through an adjustable non-linearity (sigmoid), and we optimized the non-linearity to maximize the NERP. We found that the NERP could be improved from 0.1532 to 0.4550 by making the non-linearity a hard threshold.

8. SUMMARY

We defined the confidence in a word hypothesis as the posterior probability that the word is correct. We proposed three criteria to measure the performance of a word confidence algorithm: the mean square error, the cross-entropy, and the classification error-rate. We estimated word confidences with a neural network that combines various knowledge sources relative to the words and to the hypotheses. We showed that the combination of several features significantly improved our confidence estimates. We also proposed a joint criterion that combines the performance of the recognition and confidence algorithms.

REFERENCES

- [1] J.J. Godfrey and E. C. Holliman and J. McDaniel, “SWITCHBOARD: Telephone Speech Corpus for Research and Development”, *ICASSP’92*, vol. I, pp. 517-520.
- [2] E.A.Wan, “Neural Network Classification: A Bayesian Interpretation”, *IEEE Trans. Neural Networks*, vol. 1, no. 4, pp. 303-305.
- [3] M. D. Richard and R. P. Lippmann, “Neural Network Classifiers Estimate Bayesian *a posteriori* Probabilities”, *Neural Computation*, vol. 3, no. 4, pp. 461-483, 1991.
- [4] Z. Rivlin, M. Cohen, V. Abrash, Th. Chung, “A Phone-Dependent Confidence Measure for Utterance Rejection”, *it ICASSP’96*, vol. I, pp. 515-519.
- [5] R. C. Rose and D. B. Paul, “A Hidden Markov Model Based Keyword Recognition System”, *ICASSP’90*, pp. 129-132.
- [6] R. A. Sukkar and J. G. Wilpon, “A Two Pass Classifier Utterance Rejection in Keyword Spotting”, *ICASSP’93*, vol. II, pp. 451-454.
- [7] H. Gish and K. Ng, “A Segmental Speech Model with Applications to Word Spotting”, *ICASSP’93*, vol. II, pp. 447-450.
- [8] Sh. R. Young, “Detecting Misrecognitions and Out-of-Vocabulary Words”, *ICASSP’94*, vol. II, pp. 21-24.
- [9] E. Eide, H. Gish, Ph. Jeanrenaud, A. Mielke, “Understanding and Improving Speech Recognition Performance Through the Use of Diagnostic Tools”, vol. I, pp. 221-224.
- [10] M. Weintraub, “LVCSR Log-Likelihood Ratio Scoring for Keyword Spotting”, *ICASSP’95*, vol. I, pp. 297-300.
- [11] E. Lleida and R.C.Rose, “Efficient Decoding and Training Procedures for Utterance Verification in Continuous Speech Recognition”, *ICASSP’96*, vol. 1, pp. 507-510.