Improved Word Confidence Estimation using Long Range Features

David D. Palmer†‡ and Mari Ostendorf†

†Electrical Engineering Dept., University of Washington, Seattle, WA 98195
‡The MITRE Corporation, Bedford, MA 01730
{palmer,mo}@ee.washington.edu

Abstract

This paper describes experiments in improving word confidence estimation using document- and task-level features of the hypothesized word sequence from a recognizer. The improved confidence estimates are shown to improve information extraction performance, specifically named entity (NE) recognition. The detected names can then be used to further improve confidence estimation in a multi-pass NE recognition framework.

1. Introduction

Many automatic speech recognition (ASR) systems now provide word confidence scores – an estimate of the probability that the word is correct – with each word output from the system. Confidence measures associated with ASR output are useful both for rejecting errorful utterances or parts of utterances in dialog systems, as well as for unsupervised adaptation in transcription applications where confidences allow the system to target regions of speech that are more likely to be correct. A relatively new application is in explicit error modeling to improve the performance of an information extraction system [7]. Since word confidences are an essential part of the uncertainty model for information extraction, we would naturally expect that better confidence scores would further improve the results. In this paper we describe experiments in improving word confidence estimation and demonstrate that this improvement results in improved information extraction performance. We focus specifically on several features that are not available to a standard ASR confidence predictor and that might complement previous results in confidence estimation. In particular, we investigate **document-level features** that encode information from an extended window around a word that can encompass the entire document, and **task-dependent features** that are derived from the entire training corpus.

The goal of word confidence estimation is to combine a set of features derived from the speech recognition process and compute the posterior probability that a hypothesized word is correct. The features typically come from three main sources: posterior probability scores directly from the recognizer, language model information (such as the n-gram back-off for the candidate word), and acoustic information (including signal-to-noise ratio, word duration, and speaking rate). For example, in the baseline confidence predictor we have worked with (from Dragon), six features are used: the word duration, the ASR language model score, a normalized acoustic score, the average number of HMM states from the ASR system active during recognition of the word, the fraction of times the word occurs in the top 100 hypotheses for the utterance, and the log of the number of recognized words in the utterance [4].

Since confidence estimation methods use feature sets that include both numeric features (continuous or discrete values) and categorical features, the model used to estimate confidences must efficiently combine all types of possible features. Some estimation techniques commonly used are decision trees, neural networks, generalized linear models, and generalized additive models (see, e.g., [8, 3, 5, 10, 2]). Siu and Gish [9] provide a thorough discussion of the prediction and evaluation of word confidence scores in speech recognition. They report that methods accommodating all three sources of possible features perform better than those methods that produce confidence scores derived solely from the recognizer's posterior probabilities.

In the following sections, we describe the features we used to improve the confidence estimates for our data and provide the motivation behind our selection. We then describe three methods used to estimate confidences using the features. We report the results of the confidence estimation itself and the impact on information extraction performance.

2. Long-range Features

In this work, detailed output in terms of acoustic scores from the recognizer was not available, and our focus was on improving confidence prediction by introducing features derived only from the hypothesized text string H. We assume that the original confidence scores, $\gamma_t = p(K_t = 0|A)$ (where K_t is an error indicator) encode information related to the acoustic and language models of the recognizer. The new text features were combined with the baseline confidence score γ_t for each hypothesized word h_t. Several features were tried; the most useful are described here.

Original confidence scores: We use a short window of the original confidence scores: γ_t, γ_{t-1} and γ_{t+1}. Note that the post-processing paradigm allows us to use non-causal features such as γ_{t+1}. Table 1 shows the average confidence scores for hypotheses within a window of the target h_t for correct output words and word errors. As expected, there is a clear difference between the confidence distributions for output errors and for correct words. We conjecture that several low confidence scores in a row, surrounded by high confidence scores on either side, can be a strong indication of a sequence of errors. The fact that the average confidence scores of the neighboring words, γ_{t-1} and γ_{t+1}, are similar to those for γ_t supports our use of these features by providing evidence that errors often occur in sequences. However, it is interesting to note that the average γ_t for word errors is still larger than 0.5, so that a threshold applied at 0.5 would catch fewer than half the word errors.

Ratio of original confidence scores to the average score for the document: A low confidence score for a word is less likely to indicate a word error if the average confidence for the entire document is also low. We define three features, based on the ratios of γ_{t-1}, γ_t, and γ_{t+1} to the average confidence for the

Output type	Average γ_{t-1}	Average γ_t	Average γ_{t+1}
Correct	0.78	0.81	0.79
Error	0.65	0.55	0.62

Table 1: *Average confidence scores for a window about the target word h_t for correct words and for ASR output errors.*

document in which h_t appears.

Frequency of occurrence of the word within a window of n words to the left and right in the document: An ASR system may not consistently produce the same errors for multiple occurrences of the same input word, and we can derive features from this information. In general, we would expect that words occurring frequently in a small window would be more likely to be errors, while words occurring frequently in a larger window would be more likely to be correct. We define features based on how many times the hypothesis word h_t occurs in a window $(h_{t-n}, ..., h_t, ..., h_{t+n})$ for $n = 5$, 10, 25, 50, and 100 words.

Relative frequency of words occurring as an error in the training corpus: When observing a word in the ASR output, it is useful to know how often the word is correctly recognized by the ASR system, on average, and how often it is output as an error. Considering the entire corpus of output for a particular recognizer can also reveal particular biases in its performance. In particular, the rate at which many words are produced in error differs dramatically from the overall error rate of all words combined. For example, in our data, the word "yet" is more likely to be observed as an error than as correct: it occurs 120 times, 56 times correct (46.7%), 64 times as an error (53.3%). In contrast, the word "nichols" occurs 28 times, and is correct all 28 times (100%). We define three features, the relative frequencies of h_t and its neighbors occurring as errors in the training corpus: $p(error, t-1)$, $p(error, t)$, and $p(error, t+1)$. To allow for the possibility of words occurring in the test data that were not present in the training data, we also train a "back-off" model. Words that occur fewer than five times in the training data are collapsed into a single token, and a single "rare word" error probability is calculated using all instances of these infrequent words. Words in the test data that were unseen in training are then assigned this probability as a feature.

Additional features that we experimented with and that did not improve confidence estimation included: the number of syllables in h_{t-1}, h_t, and h_{t+1}; whether h_{t-1}, h_t, and h_{t+1} are function words; and the frequency of occurrence of h_{t-1}, h_t, and h_{t+1} in the entire training corpus.

3. Estimation Techniques

We investigated three different methods for using the above features to improve confidence scores: decision trees, generalized linear models, and linear interpolation of the outputs of the decision tree and generalized linear model.

Decision Tree (DT)

A very common machine learning tool is the decision tree, or classification and regression tree [1]. The tree can be used to classify a set of examples into categories, or alternatively to generate probability distributions.

In order to train, or induce, a decision tree using a set of features $f_1 ... f_i$, each feature is individually evaluated based on its ability to distinguish the desired categories (in our case, *correct* or *error*) in the training data. At each step in the learning procedure, the evolving tree is branched on the feature which best divides the data items, according to a minimum entropy criterion. Branches are added to the tree until a stopping criterion such as maximum number of leaf nodes is met. To reduce the effects of overfitting, the learning algorithm prunes the tree after the entire decision tree has been constructed. It recursively examines each subtree to determine whether replacing it with a leaf node would reduce the number of errors on a held-out data set. This pruning produces a decision tree which is better able to generalize to data that is different from the training data.

The relative frequencies of the training item categories that fall through to each leaf node in the tree provide the probability distribution for that node. This distribution can also be viewed as the confidence score for each category for each leaf node. For example, if the features in the training data cause seven "correct" training items and three "error" training items to reach a particular leaf node, any test item reaching that leaf node would be assigned a confidence $\gamma_{DT} = 0.7$. Smoothing is carried out by replacing all zero probabilities with the smallest non-zero probability of any of the leaf nodes in the tree and normalizing to maintain consistent probability distributions.

Generalized Linear Model (GLM)

The generalized linear model (GLM) is a log-linear model [6] that provides a means for linearly combining a diverse set of features, such as those used for confidence estimation. Training a GLM using a set of features $f_1 ... f_i$ involves estimating (from training data) a weight for each feature, $\beta_1 ... \beta_i$, plus a normalizing constant β_0. The feature values are then linearly combined using the weights:

$$\alpha = \beta_0 + \beta_1 f_1 + \beta_2 f_2 + ... + \beta_i f_i. \quad (1)$$

When f_i is a numerical feature, β_i is a scalar weight. When f_i is a categorical feature, we can consider it an indicator vector of dimensionality equal to the number of categories, with vector elements equal either 1 or 0; the weight β_i is then actually a vector with dimensionality equal to that of the indicator vector. A single probability value can then be calculated from the linearly combined value α:

$$\gamma_{GLM} = \frac{e^{\alpha}}{1 - e^{\alpha}}. \quad (2)$$

Interpolation of DT and GLM

Since the decision tree and the generalized linear model combine the input features using very different statistical techniques, the resulting predictions are likely to be different. It has been shown that significant performance improvements can be achieved on a wide variety of classification and regression problems by interpolating the outputs of diverse models. We can also take advantage of this diversity in word confidence estimation by interpolating the confidence values produced by the decision tree and the GLM, γ_{DT} and γ_{GLM}, respectively:

$$\gamma' = \lambda * \gamma_{DT} + (1 - \lambda) * \gamma_{GLM}, \quad (3)$$

where $\lambda = 0.4$, determined empirically using development test data.

4. Improved Confidence Prediction Results

In this section we present the results of our experiments in using the features and techniques described in the previous section to improve confidence estimation. For our experiments, we used

the decision tree package in S-plus and the *glm* function in the public-domain statistical software package SAS. We report results using the most common metric for reporting confidence estimation performance, normalized cross entropy (NCE) [9]. Increased NCE scores (max is 1.0) indicate that the confidence estimates provide a significant amount of information about the correctness of the words; a low NCE score indicates the confidences do not provide a useful measure of correctness. The baseline confidences we use in our experiments have an NCE of 0.195 [4], which was one of the best reported confidence estimation results at the time the data set was produced.

4.1. Relative Importance of Features

This section presents results of experiments aimed at determining which features contribute most to improved confidence estimation. Table 2 presents the results of progressively adding features in decision tree learning. The first column gives the features that were used to train the tree; the second column shows the total number of leaf nodes in the pruned decision tree; and the third column shows the NCE for that tree measured on the development test data. The first row represents the baseline that uses only the word confidence γ_t. The resulting tree consists of just two leaf nodes, with the single question ("$\gamma_t > 0.725$?"). The second row shows the improvement when training on all features derived directly from the confidence values, including the three ratio features, which interestingly gives a similar NCE to that of the baseline confidence scores. The third row shows a significant increase in performance when adding the $p(error)$ features, and the fourth row shows a small improvement when adding the window features.

Features	# Leaves	NCE
baseline (state-of-the-art)	n/a	0.195
γ_t	2	0.160
γ_{t-1},γ_t,γ_{t+1}, ratio	34	0.194
γ_{t-1},γ_t,γ_{t+1}, ratio, $p(error)$	122	0.242
γ_{t-1},γ_t,γ_{t+1}, ratio, $p(error)$, window	212	0.252

Table 2: *Confidence estimation performance of the decision tree for various feature sets.*

Another method for determining the importance of individual features is to examine both the feature weights, $\beta_1, \ldots, \beta_i$, learned by the GLM together with the average values of each feature. As we see from Equations 1 and 2, when all feature values are non-negative, a positive β weight indicates that the feature provides evidence of the correctness of h_t, and a negative β indicates evidence of a word error. The magnitude of the product of β and the average feature value represents the strength of the evidence either way. Table 3 shows the β weights learned for each of the features. The intercept, β_0, was -1.77, which is intuitive since this would result in a very low confidence score whenever all the input features equal zero.

For the most part, the values in Table 3 are consistent with the relative importance of the features used in the decision tree: the products for the features γ_t and γ_{t-1} have high positive values (evidence of correct words), and the products for the $p(error)$ features have large negative values (evidence of word errors). Somewhat surprising is the low products for all the window features, since these features figured prominently in the decision tree induction. However, the relative values of the window features are consistent with our hypothesis: words occurring frequently in a small window are more likely to be errors (negative GLM β), while words occurring frequently in a larger window are more likely to be correct (positive GLM β).

Training Feature (f_i)	GLM Weight (β_i)	Average Value ($\bar{f}_i$)	$\beta_i \bar{f}_i$
γ_{t-1}	0.77	0.75	0.58
γ_t	4.15	0.75	3.11
γ_{t+1}	1.64	0.75	1.23
ratio(t-1)	0.23	1.00	0.23
ratio(t)	0.77	1.00	0.77
ratio(t+1)	0.49	1.00	0.49
p(error,t-1)	-1.17	0.23	-0.27
p(error,t)	-5.16	0.25	-1.29
p(error,t+1)	-1.16	0.23	-0.27
window5	-0.11	1.11	-0.12
window10	-0.05	1.30	-0.07
window25	0.04	1.59	0.06
window50	0.02	2.16	0.04
window100	0.03	3.28	0.10

Table 3: *Relative importance of document-level and task-dependent features in GLM training.*

Table 4 shows the test set NCE together with the corresponding error detection performance[1] for a single operating point (threshold T=0.7). The NCE value increases significantly over the state-of-the-art baseline (0.195) for all models, with the interpolated confidences showing the largest improvement.

Training method	NCE	Error detection F-Measure
Baseline (state-of-the-art)	0.195	56.1
Decision Tree	0.252	58.4
GLM	0.255	58.6
Interpolated	0.262	60.2

Table 4: *Performance of confidence estimation methods, using confidence thresholding ($T = 0.7$).*

5. Word Confidence and Information Extraction

The motivation for our experiments in improving confidence prediction stems from uncertainty modeling in an information extraction system, specifically for a named entity (NE) recognition task, where we saw that effective error modeling greatly improved name finding performance [7]. As anticipated, Table 5 shows improvement in NE recognition performance using the new confidence predictors. For comparison, the final row shows the performance of the model with "perfect" confidence scores, in which each correct word has a confidence of 1 and each word error has a confidence of 0; this can be thought of as the upper bound for improved confidences. Note that significant improvement is still possible, as the performance with improved confidences is still well below this upper bound.

Due to the close correlation between names and errors, as demonstrated in [7], we would expect to see improvement in the error modeling performance by including information about

[1] Performance is given in terms of the F-measure, which is the harmonic mean of precision and recall.

Confidence Source	NE F-Measure	NE SER
Baseline	71.1	46.4
Decision Tree	71.8	45.5
GLM	71.6	45.6
Interpolated	72.0	45.3
Perfect	80.1	27.3

Table 5: *Impact of improved confidences in NE recognition (SER = slot error rate).*

which words are names, as determined by the NE system. In addition to the set of document-level and task-dependent features used in the experiments in the previous section, we can also define a new feature: *whether the hypothesis word h_t is part of a location, organization, or person phrase*. We can determine the value of this feature directly from the output of the NE system. Given this additional feature, we can define a multi-pass processing cycle consisting of two steps: confidence re-estimation and information extraction. To obtain the name information for the first pass, the confidence scores are re-estimated using just the features from Section 2, and these confidences are used in a joint NE and error decoding system. The resulting name information is then used, in addition to all the features used in the previous pass, to improve the word confidence estimates. The improved confidences are in turn used to further improve the performance of the NE system.

Processing Pass	Confidence NCE	NE System F-Measure	NE System SER
Baseline	0.195	68.4	50.9
0	0.252	71.8	45.5
1	0.274	72.6	45.1
2	0.282	73.0	44.5
3	0.287	73.1	44.3

Table 6: *Results of multi-pass experiments in confidence estimation and information extraction (SER = slot error rate).*

The results are shown in Table 6. Note that the confidence estimation result for Pass 1 (NCE = 0.274) is better than the best result in Table 4 (NCE=0.262). This indicates that the name information improves confidence estimation, even when the name phrase features are provided by an information extraction system with less-than-perfect performance. The remaining rows in Table 6 show the results of Passes 2 and 3, in which the improved information extraction output was again used to reestimate the confidences, which in turn are used to again improve the information extraction performance. The performance appears to reach a maximum very quickly, as there was marginal improvement between Passes 2 and 3, and repeating the cycle further produced even smaller improvements.

6. Discussion

In this paper we presented results in word confidence estimation, in which we showed that long range features can be used to improve estimation. We investigated features that encode information from an extended window around a word that can encompass the entire document, and task-dependent features that are derived from the entire training corpus. These new features improved confidence estimation from the baseline NCE of 0.195 to an improved value of 0.262 (row 2, Table 7). We also showed that name information can further improve estimation, giving an NCE of 0.287 (row 3, Table 7). In turn, the improved confidence estimation leads to a significant improvement in named entity detection performance, reducing the slot error rate by 10%.

Features	NCE
Baseline (state-of-the-art)	0.195
Document-level + Task-dependent	0.262
Document-level + Task-dependent + Names	0.287

Table 7: *Summary of best word confidence estimation results.*

7. References

[1] L. Breiman, J. H. Friedman, R. Olshen, and C.J. Stone, *Classification and Regression Trees*, Wadsworth International Group, Belmont, CA, 1984.

[2] G. Evermann and P. C. Woodland, "Posterior Probability Decoding, Confidence Estimation and System Combination," *Proceedings of the 2000 Speech Transcription Workshop*, University of Maryland, May 16-19, 2000.

[3] L. Gillick, Y. Ito and J. Young, "A Probabilistic Approach to Confidence Estimation and Evaluation," *Proc. International Conference on Acoustic, Speech and Signal Processing*, vol. 2, pp. 879-882, 1997.

[4] L. Gillick, Y. Ito, L. Manganaro, M. Newman, F. Scattone, S. Wegmann, J. Yamron and P. Zhan, "Dragon Systems' Automatic Transcription of New TDT Corpus," *Proceedings of the 1998 Topic Detection and Tracking (TDT) Evaluation.*

[5] T. Kemp and T. Schaaf, "Estimating Confidence Using Word Lattices," *Proc. European Conference on Speech Comm. and Tech.*, pp. 827-830, 1997.

[6] P. McCullagh and J.A. Nelder, *Generalized Linear Models* (second edition), London: Chapman and Hall, 1989.

[7] D. Palmer, M. Ostendorf and J. Burger, "Robust Information Extraction from Automatically Generated Speech Transcriptions," *Speech Communication*, vol. 32, pp. 95-109, 2000.

[8] M. Siu, H. Gish and F. Richardson, "Improved Estimation, Evaluation, and Applications of Confidence Measures for Speech Recognition," *Proc. European Conference on Speech Comm. and Tech.*, pp. 831-834, 1997.

[9] M. Siu and H. Gish, "Evaluation of word confidence for speech recognition systems," *Computer Speech & Language*, vol. 13, No. 4, Oct 1999, pp. 299-319

[10] M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig and A. Stolcke, "Neural-network based measures of confidence for word recognition," *Proc. International Conference on Acoustic, Speech and Signal Processing*, vol. 2, pp. 887-890, 1997.