COMPENSATING FOR WORD POSTERIOR ESTIMATION BIAS IN CONFUSION NETWORKS

Dustin Hillard and Mari Ostendorf

Department of Electrical Engineering, University of Washington, Seattle, WA
{hillard,mo}@ee.washington.edu

ABSTRACT

This paper looks at the problem of confidence estimation at the word network level, where multiple hypotheses from a recognizer are represented in a confusion network. Given features of the network, an SVM is used to estimate the probability that the correct word is missing from a candidate slot and then other word probabilities are normalized accordingly. The result is a reduction in overall bias of the estimated word posteriors and an improvement in the confidence estimate for the top word hypothesis in particular.

1 Introduction

There is increasing interest in applying language processing technology, such as information extraction and machine translation, to spoken documents (voicemail, meeting recordings, broadcast news, etc.). However, input from speech recognition systems typically contains errors, so it is useful to have some measure of the recognizer's confidence that the word is correct in order to allow systems to deweight the importance of certain words and thereby reduce the adverse affects of errors.

Most speech recognition systems provide a confidence estimate for the best word hypothesis. Usually the word posteriors resulting from the recognition process are over confident, because (in order to be computationally tractable) recognizers prune the set of hypotheses that are considered and the word likelihoods are thus normalized by some subset rather than the total hypothesis space. When the pruned hypothesis space is richer, then better results are obtained, as illustrated by the improvements from using word graphs rather than N-best lists in [1]. To account for this bias, previous work has trained models to warp the confidence of the top word hypothesis to obtain a more accurate measure [2, 3]. Alternatively, the use of a background model to cover the pruned space is proposed in [4].

This work departs from previous work by addressing the issue of bias for multiple word hypotheses generated by the recognizer, not just the top hypothesis. By improving the confidence estimate for all word hypotheses we improve the ability of downstream systems to make use of alternative words and have accurate confidence measures for those words. Our method directly models the probability that the recognizer did not hypothesize the correct word, and uses this predicted probability to adjust the hypothesized word posteriors. Experimental results show that this simple predictor reduces the problem of bias in word posterior estimates for the whole network and improves the 1-best confidence estimate significantly as well. In addition, we can use this probability of a missed word for other tasks such as out-of-vocabulary (OOV) word detection or other types of error handling.

Our approach builds on the confusion network representation of word uncertainty, which is described in Section 2 to present the framework and illustrate the problem of bias. The method for predicting the missed word probability and thereby adjusting the network posteriors is described in Section 3. The experimental paradigm and results are described in Section 4 and 5, respectively. Contributions and open questions are summarized in Section 6.

2 Confusion Network Posteriors

This work assumes a confusion network representation of recognizer uncertainty. A confusion network (CN) is a compact representation of a word lattice or N-best list, where the complexity of the lattice or list representation is reduced to a series of slots that each have word hypotheses (and null arcs) and associated posterior probabilities [5]. The posterior probabilities of all hypotheses in a slot (including the null arc, if present) are chosen such that they sum to one. This effectively assigns zero probability to the event that the word is not in the lattice (or list), which results in an optimistic bias of the word posteriors since such events do occur, particularly for OOV words.

To illustrate the problem of bias, Figure 1 shows a plot of the relative frequency that hypothesized words are correct as a function of their predicted confidence, using data from a conversational speech recognition task described further below. The relative frequencies are computed by binning over different confidence intervals. The distance of the curve from the diagonal line reflects the bias of the estimate. Where the curve falls below the diagonal, the estimates are "over confident", e.g. words predicted with a posterior of 0.8 are correct in less than 60% of the cases. Similarly, when the curve is

above the diagonal, as for the low posterior cases, the estimate is lower than it should be.

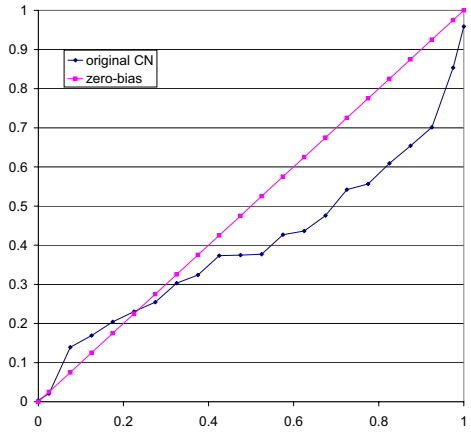

Fig. 1. Relative frequency that a hypothesized word is correct as a function of the predicted posterior in a CN.

Were it not for the bias, the posterior probability of the top word in the CN would be a reasonable confidence estimate for that word. To obtain better confidence estimates, it is common practice to use a secondary classifier, such as a neural network (NN) [2], that takes as input this posterior probability (or a posterior calculated in some other way from the lattice) as well as other features of the lattice or CN. This is a useful approach if one only wants the confidence of a single hypothesis, but it is not practical if one wants to adjust the posteriors of all arcs in the lattice.

3 Network Posterior Adjustment

An obvious solution to the problem of over-confidence is to introduce an entry in each slot to account for the event that the correct word is not in the list. The probability of a missed word is predicted independently at each slot in a manner somewhat similar to confidence prediction techniques that combine recognizer posteriors with other features. (An advantage of this approach over using a background model is that the additional features can provide cues beyond what is captured in the recognizer acoustic and language models.) Then, the word and null-arc probabilities are simply renormalized to account for the added probability mass. Thus, the simple prediction of a series of probabilities of missed words has the effect of adjusting the posteriors of the full network. Also, the addition of a missed word probability may be useful for other tasks, such as OOV detection.

Many different approaches are possible for predicting the missed word probability. Details regarding the features and prediction models used in the experiments conducted here are described next. In order to extract features and train the model, we align reference transcriptions with CNs for a collection of recognizer test data. A separate recognizer test set is used for evaluating the prediction and its impact on the CN posteriors as a whole.

3.1 Features

As the "baseline" set, we adopt the features used in previous work [2], which include: the length-normalized position of the word in the sentence; the log length of the sentence; two Boolean features indicating whether the adjacent slot (to the left and right) has the null arc as the most probable word; the posterior probability of the mostly likely word in the current, left and right slots; and the unigram word probability from the recognizer language model. In the "extended" set, we include additional feature types: the mean and variance of the posteriors in the current slot, length (in characters) of the top word, and the Boolean features from above extended to the slots two to the left and right of the current slot. In addition, we explore also adding the lexical identity of the top word in the current slot, referred to as the "full" feature set. Boolean features are added for each word, either for the top 1000 most common words, or for all words in the dictionary.

3.2 Models

Given a set of features, the problem is to predict the posterior probability of a binary event (whether or not the list is missing the correct word), which could be based on a statistical binary classifier or a regression model. Initially, we trained a NN with the same setup as used in previous work on confidence prediction [2]. However, although the NN approach is effective for improving the confidence estimate of the top hypothesis, its performance was poor in predicting missing word probabilities. We then adopted support vector machine (SVM) regression as an approach that was more flexible (i.e. allowed easily for a larger feature set, such as the lexical word feature). SVMs have been used with success in other work on word error detection [6]. The SVM with a linear kernel was not a successful approach, but a Gaussian kernel improved results to a reasonable level. In both cases, the models are trained by using a target of 1 when the reference word is not present in the slot hypothesis list, and 0 when it is.

Analysis of preliminary experiments indicated some regions that were not being handled well by the classifiers. An SVM trained on all words performed poorly for words that received very high recognizer posteriors (.999 and greater). Since these comprised more than half of the slots and the actual accuracy for these words was .95, a simple heuristic solution imposed a minimum probability of miss equal to .05. This provided improved results, but did not completely address the issue. An alternative solution was to train two classifiers, one for slots with confidence greater than .95, and one for those less than .95 (before clipping). As shown in the experiments, combining these approaches gave the best results.

4 Experimental Paradigm

Experiments are conducted on a conversational telephone speech recognition task. The recognizer used for our experiments is the SRI 20 times real time Decipher system with small mod-

ifications from the system used for the 2004 DARPA evaluations. This is a state-of-the-art system, which combines three systems (MFCC cross-word, MFCC non-cross-word, and PLP cross-word) with cross adaptation and a final Rover step. The data sets used for training and testing the missed word probability predictor are the from the NIST RT evaluations. Training is performed on the development test set from 2004, and testing on the evaluation set from 2003.

The results are evaluated in three ways, associated with the different tasks that might benefit from this approach. First, to evaluate the impact of posterior correction over the whole CN, we present a plot of the word accuracy versus predicted confidence that reflects the percent of words that are correct over multiple confidence intervals, as in Figure 1. To evaluate the impact on the confidence estimate of the 1-best word hypothesis, we use the standard normalized cross entropy (NCE) measure [7]:

$$NCE = (H_{max} - H_{conf})/H_{max}$$

where

$$\begin{aligned} H_{max} &= -p_c \log_2 p_c - (1-p_c)\log_2(1-p_c) \\ H_{conf} &= -1/n[\sum_{w_i\ corr} \log_2 p_i + \sum_{w_i\ err} \log_2(1-p_i)] \end{aligned}$$

and where $p_c = n_c/n$ is the average probability that a hypothesized word is correct, p_i is the predicted confidence that w_i is correct, and the sum is over all n hypothesized words in the test set. Finally, to evaluate the prediction of the probability of a missing word (i.e. the correct word is not in that slot in the confusion network), we use a decision-error trade-off (DET) curve. The DET curve is a standard method for showing system performance on detection tasks, illustrating the trade-off in the percentage of misses and false alarms.

5 Experiments

The primary aim of our experiments is to improve the confusion network as a whole. A series of experiments evaluating different feature sets and model configurations for that purpose are described in Section 5.1. In addition, we examine the impact of the resulting system on 1-best confidence estimation and missing word detection, in Sections 5.2 and 5.3, respectively.

5.1 Confusion Network Posterior Correction

Our first series of experiments explored the different feature sets (baseline, extended and full) with a single regression SVM. Recall, the baseline feature set is that used in standard NN-based confidence prediction; the extended set adds five related features from a larger window, and the full set also adds lexical identity. As shown in Figure 2, the extended features improve results, although the full feature results do not.

The majority of the CN slots had a top word with a posterior (from the recognizer) greater than .95, so to investigate further modeling improvements we experimented with

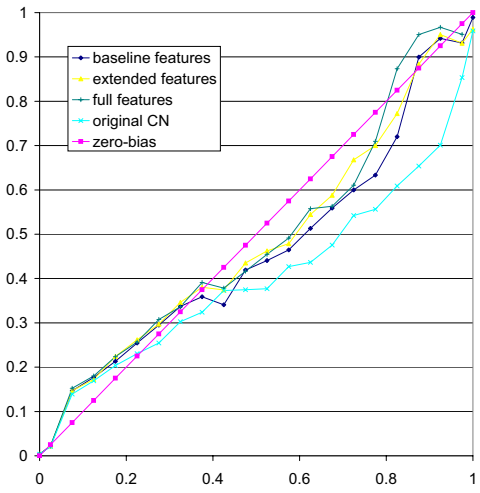

Fig. 2. Various feature sets using one SVM

using two SVMs to separate out the large portion of the data which had very high "original" posteriors: one SVM for slots with a top word having a posterior greater than .95, and the other SVM for those with a top word having confidence less than .95. The performance for mid-range confidences was improved, and the best results were obtained with all lexical identity features, but the results for the very highest confidence words were still severely biased. The poor performance was in part attributable to sparseness, because very few words remained with high confidence after applying the two SVMs. This problem is addressed by introducing a heuristic cap of .95 on the posteriors, chosen because the words with confidence of 1 output by the recognizer are only 95% accurate (this threshold would likely need to be tuned for other tasks).

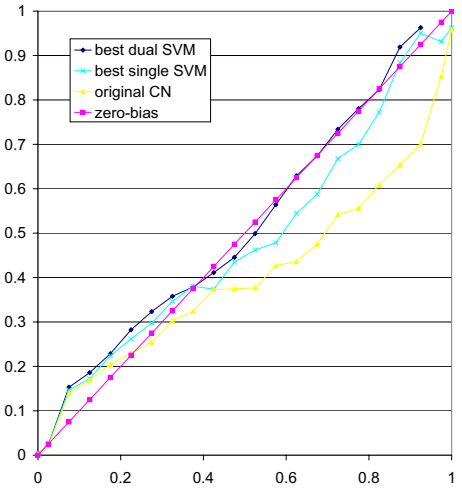

Fig. 3. Comparison of bias plots for original posteriors, compensation using the best single SVM, and compensation using the best dual SVM with thresholding.

Figure 3 shows a comparison of the bias of the original confidence output by the recognizer, the best single SVM

(with extended features), and the full feature dual SVM with thresholding. With the best case system, almost all of the bias has been eliminated.

5.2 1-Best Word Confidence Prediction

As described earlier, the highest ranking word in a slot is the best recognizer output, and its posterior probability can be used as a confidence estimate. While the goal of this work is to improve the posteriors in the network as whole, it is of interest to assess the impact on the 1-best hypothesis because of its particular importance. The original unwarped CN confidences give a NCE of .161 (with confidences clipped at .05 and .95 to avoid negative NCE values). Using the predicted missed word probability to normalize the network with the baseline feature set increases NCE to .187. Using the same features in a NN trained to predict 1-best word confidence explicitly results in an NCE of .222. The NCE of the 1-best word confidence using our best network compensation system (two SVMs, the full feature set, and thresholding) is .259.

5.3 Detecting Missed Words

The probability of missed word output by our system could also be used to detect slots where the recognizer has not hypothesized the correct word, which would indicate regions where the lattice (or lexicon) should be expanded and the hypothesis rescored. The DET curve in Figure 4 shows curves for our system with the baseline features, and with the full feature set. The full feature set is almost always better than the baseline features, but still does not have great success. When false alarms are limited to 10%, 60% of the slots with missing words are not detected.

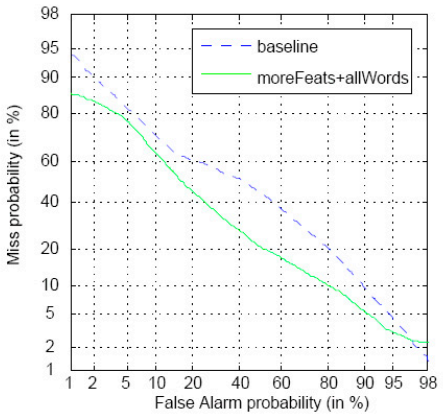

Fig. 4. DET curve for detecting slots with missing words.

6 Conclusion

In summary, this work addresses the problem of accurate word posterior probability estimation at the network level, using prediction of the probability of missed words to adjust for biases introduced by using confusion networks. Using SVMs and simple features of the local context in the confusion network, very good performance is achieved, removing most of the bias in the network estimates. Using the resulting 1-best word posterior as a confidence estimate is an improvement over the original network, though not quite as good as predicting the corrected confidence for that word directly using the same features (using a NN). However, the SVM can make use of additional features that lead to an overall improvement in 1-best word confidence. A by-product of the method is the availability of a probability of missed words, which might be used for OOV or more general error detection. Unfortunately, the performance of that detector on its own is still quite poor. One direction for future work is to improve this component. In addition, we plan to assess the impact of the network improvements on system combination and model adaptation.

Acknowledgments

This work was supported by DARPA under contract HR0011-06-C0023. Distribution is unlimited.

7 References

[1] F. Wessel, R. Schluter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Trans. Speech and Audio Processing*, vol. 9, no. 3, pp. 288–298, 2001.

[2] M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig, and A. Stolcke, "Neural-network based measures of confidence for word recognition," *Proc. ICASSP*, pp. 887–890, 1997.

[3] F. Soong, W.K. Lo, and S. Nakamura, "Optimal acoustic and language model weights for minimizing word verification error," in *Proc. ICSLP*, 2004, pp. 441–444.

[4] P. Liu, J.-L. Zhou, and F. Soong, "Background model based posterior probability for measuring confidence," in *Proc. Eurospeech*, 2005, pp. 1465–1468.

[5] 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*, pp. 373–400, 2000.

[6] Z.-Y. Zhou and H. Meng, "A two-level schema for detecting recognition errors," in *Proc. ICSLP*, 2004, pp. 449–452.

[7] NIST, "The 2001 NIST Evaluation Plan for Recognition of Conversational Speech over the Telephone," http://www.nist.gov/speech/tests/ctr/h5_2001/h5-01v1.1.pdf, 2000.