New Features for Confidence Annotation

D. Bansal and M. K. Ravishankar

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
$dbansal@cs.cmu.edu$, $rkm@cs.cmu.edu$

ABSTRACT

In this paper we describe two new confidence measures for estimating the reliability of speech-to-text output: *Likelihood Dependence* and *Neighborhood Dependence*. Each word in the speech-to-text output for a given utterance is annotated with these two measures. Likelihood dependence for a given word occurrence indicates how critical that word is to the overall utterance likelihood; i.e., how much worse is the likelihood of the next best utterance if that word is eliminated from the recognition. Neighborhood dependence measures how stable a given word is when neighboring words are changed in the recognition. We show that correct and incorrect words in the recognition behave significantly differently with respect to these measures. We also show that on the broadcast news task they perform better than some of the existing, commonly used confidence measures.

1. Introduction

Detecting regions of high and low reliability or $confidence$ in the output of an automatic speech recognizer is an important task. Many practical applications of speech recognition systems can benefit from such information. For example, if a certain recognition result is deemed to be unreliable, the application may prompt the user for clarification. Thus, the availability of confidence information can add to the sophistication of speech-based applications.

Confidence annotation can be done in principle at various levels: e.g., sentence, phrase, word, or phone level. In this paper, as with most current implementations, we are concerned with confidence annotation at the word level. That is, each word output by the speech recognizer is annotated with a confidence value.

A perfect confidence annotator would assign a reliability of 100% to correctly recognized words, and 0% to incorrect ones. In practice, no such thing exists, of course. Instead, typical confidence annotators assign any value between 0 and 100%. The quality of a confidence annotator can be evaluated by seeing how close it performs to the perfect annotator. (Chase's Ph.D. thesis [1] addresses this question in depth.) Another desirable property of a confidence estimator is that it should be computationally inexpensive; for example, it should allow real-time recognition performance.

Several confidence measures have been proposed in the past [1, 2, 4, 7]. Two of the more successful ones are: *N-best List Homogeneity* (NBH) and *Language Model Jitter* (LMJ) [1, 3]. NBH is the fraction of an N-Best list containing a given word within a given time segment. The closer this fraction to 1, the more reliable is the word. LMJ is the fraction of times a given word remains present in the recognition hypothesis under varying language-model parameters, such as the language weight[1]. Again, the higher this fraction, the more likely it is correct.

In this paper, we introduce two new features for confidence annotation. The first is *Likelihood Dependence* (LD). The LD value for a given word instance indicates how much the overall hypothesis likelihood depends on that instance. It is obtained by comparing the original utterance likelihood to that of the next best recognition hypothesis that is constrained to exclude that word instance.

The second feature is *Neighborhood Dependence*. It defines how stable a given word instance is, even when other words in the recognition are forced to be different. Briefly, for each word instance in the main recognition hypothesis, another hypothesis is obtained that is forced to exclude that word instance. This may cause other nearby words to be changed as well. The stability of a word instance, in spite of neighboring ones being excluded, indicates its correctness.

Confidence measures can also be applied jointly. Chase [1] used a decision tree procedure to combine an arbitrary number of them. We have used a straightforward generalization of the one-dimensional case to combine the two proposed features.

We first describe the two features in greater detail in Section 2. We then describe their use in actual confidence annotation of a broadcast news test set in Sections 3 and 4. These include measurements of how accurately the features can be used to identify correct and incorrect recognitions, as well comparisons with the other measures NBH and LMJ. Section 5 concludes this paper.

2. The Proposed Features

2.1. Likelihood Dependence (LD)

For each word instance in the recognition hypothesis, we wish to estimate its contribution to the total hypothesis likelihood. We get this information as follows: For a given word instance in the original recognition, we obtain the next best recognition in which that instance does not appear, and compare the likelihoods of the two.

Let P be the log-likelihood of the original hypothesis consisting of n word instances w_1,w_2,w_3,...,w_n. (The same actual word may appear as more than one instance, of course.) Let us denote the time segmentation of the i-th word instance w_i to be (s_i, e_i); i.e., start time s_i and end time e_i, obtained as part of the recognition process. For each w_i, we prevent it from occurring anywhere $around$ the time segment (s_i, e_i), and obtain a new recognition.

[1] An exponent applied to the language model probability in obtaining the overall likelihood for a recognition hypothesis.

Let P_i be the log-likelihood of this hypothesis. The log-likelihood difference $P - P_i$ is a measure of the relevance of w_i to the original recognition hypothesis. The larger this difference, the more likely that w_i is a correctly recognized word. (We will henceforth simply say "likelihood" to mean "log-likelihood".)

A number of questions arise. First, how do we obtain a recognition hypothesis that does not contain a given word instance. (Let us refer to such hypotheses as *constrained hypotheses*.) Second, in obtaining P_i, it is not sufficient to prohibit w_i from occurring $exactly$ within (s_i, e_i), since time segmentations are never known perfectly. Finally, raw likelihood difference values cannot be directly used as confidence measures, for obvious reasons. We must derive a usable confidence measure, or probability of correctness, from the raw likelihood differences. We discuss these issues below.

Generating Constrained Hypotheses. We first provide some background on the CMU Sphinx-3 recognizer [6] that was used for this research. We use two passes to get the initial recognition. The first pass is a conventional beam search using the Viterbi algorithm. It produces a word lattice that includes word segmentations and acoustic likelihoods. The second pass is an A* search through a word graph constructed from the word lattice. The top of the N-best list from this search is the final recognition hypothesis.

To generate a constrained hypothesis, we repeat the second (A*) pass over a suitably modified word lattice. Specifically, for a given word instance w_i with time segmentation (s_i, e_i) in the original hypothesis, we create a modified word lattice from the original that excludes w_i as well as other nearby segmentations of the word. The "slop" at segment boundaries are determined empirically. In our case, if the same word occurs within 1 frame (10msec) of s_i or within 4 frames of e_i, we eliminate it from the word lattice. We use a larger slop for the end time since the word lattice produced by the Viterbi search has greater uncertainty in its word end times.

Occasionally, removing a word w_i splits the lattice into two unconnected parts. In this case, no recognition result is available. We call them $critical$ word instances. The raw likelihood difference for them is essentially infinity, and they are very likely to be correct recognitions. Second, since our A* search contains pruning and is not an optimal search, the likelihood difference $P - P_i$ may be negative. In this case, the word instance is likely to be incorrect.

Deriving the LD Confidence Measure. As mentioned earlier, given a word instance w_i and its raw likelihood difference $P - P_i$, we ultimately need to derive a probability that w_i is correct. Such a mapping function is obtained through a straightforward training process, described below.

The training data is a set of utterances for which words in the recognition hypotheses have been labeled according to their actual class: correct or incorrect. For each utterance hypothesis in the training set, likelihood difference values are computed for its constituent word instances, as described above. The entire range of these values is divided into discrete bins and the fraction of correct words in each bin determined. This is the desired LD confidence measure.

We ran this training process on data consisting of about 14000 word instances from the broadcast news (BN) recognition task [6]. (The word error rate on this test set was about 27%.) Figure 1 shows the distribution of raw likelihood difference values for the two classes of correct and incorrect word instances. There is a clear distinction in the behavior of the two classes.

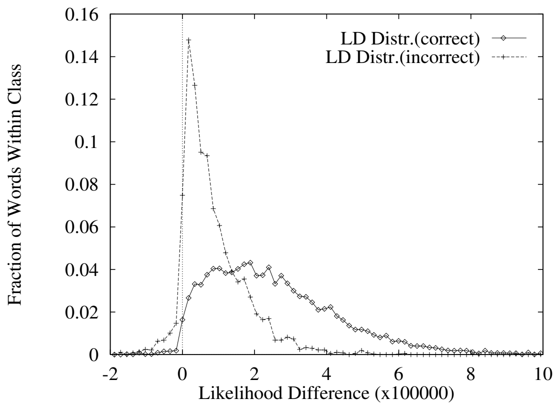

Figure 1: Distribution of raw likelihood difference values for correct and incorrect word classes.

Figure 2 shows the probability that a word instance is correct (i.e., its LD confidence score), given its likelihood difference. We see a clear correlation between the two, especially in the region where most of the data is concentrated. Note that in regions of sparse training data, the curve is quite uneven, as shown by the occasional spikes. The unevenness should be smoothed using neighboring bins to provide a more reliable curve.

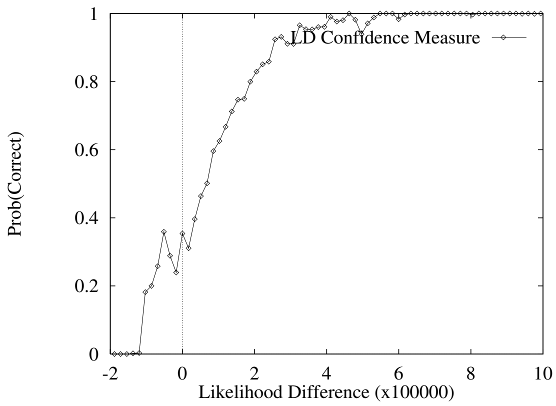

Figure 2: Probability of a word instance being correct, based on its likelihood difference.

We should also note that the two figures exclude critical words that split a lattice when removed. There were about 1000 such words in the training set, of which 98.8% were correct.

2.2. Neighborhood Dependence (ND)

Neighborhood dependence represents the number of neighbors that can affect a word instance in the recognition hypothesis. Once again, let the original utterance consist n word instances w_1, w_2, w_3,...,w_n. As with LD, we generate n constrained recognitions forcing one of the word instances to be excluded at a time. Consider two word instances w_i and w_j in the original recognition hypothesis. When a constrained recognition is produced by excluding w_i, instance w_j may or may not be present in it. (To determine its presence we look for w_j around its original segmentation, using boundary tolerances as described earlier.) We count the number of times w_j is absent in the n constrained recognitions. This is the raw *Neighborhood Dependence Count*. The larger this count, the more likely that w_j is an incorrect word. (It is important to count the number of times w_j is $absent$, rather than $present$. The reason is that whenever the excluded word w_i is far removed from the subject w_j, the latter usually remains unchanged, whether it is correct or not.)

The details of obtaining constrained recognitions are identical to the case of the LD measure (Section 2.1). Conversion of the raw neighborhood dependence count to an ND confidence score is similar: Given all the word instances in the training set with a specific raw count, we compute the fraction of them that are correct.

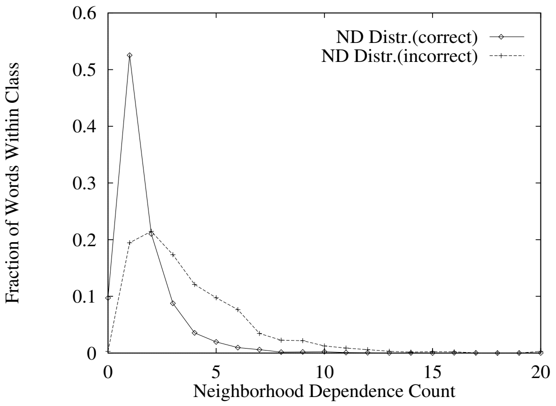

Figure 3: Distribution of neighborhood dependence counts for correct and incorrect word classes.

Figure 3 shows the distribution of the neighborhood dependence counts for the classes of correct and incorrect word instances. As with the LD measure, there is a noticeable though smaller difference between the behavior of the two classes.

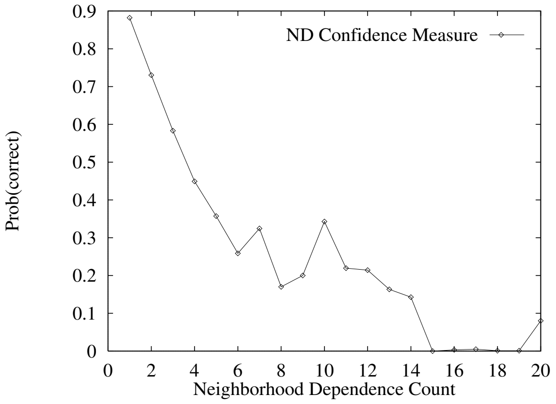

Figure 4: Probability of a word instance being correct, based on its neighborhood dependence count.

Figure 4 shows the probability of correctness derived from the ND count. As with LD, there is a marked correlation between the two. There is also a similar unevenness in regions of sparse training data that needs to be overcome using smoothing.

2.3. Joint Measures

It is straightforward to combine the two measures (or any two, for that matter) to obtain a joint one. We create a 2-dimensional array of bins, covering the space of the two measures, and count the fraction of words correct in each bin. However, one has to be much more aware of the possible sparseness of the training data. The number and granularity of bins must be chosen accordingly. It is also much more critical to smooth the distributions so obtained, to avoid over-fitting to the training data.

2.4. Algorithmic Complexity

The process described above for obtaining the confidence measures appears to be computationally expensive. For an utterance with n word instances, n new (constrained) recognitions have to be obtained. In practice, this overhead need not be inordinately high. The solution we propose for our real-time experiments is to use the *global best path search* [5] pass of the Sphinx-3 or Sphinx-II decoders. This algorithm finds the globally optimum path through a word lattice such as the one described in Section 2.1. It is an efficient algorithm that usually runs about 10-20 times faster than real time on large vocabulary tasks on modern computers. Therefore, for short utterances where n is about 10 words, the additional computation is likely to be within a real time. For the longer sentences in the BN task, we have seen that the computation required was about 2 times real time.

3. Experiments

We have just shown that the LD and ND statistics are significantly different for the classes of correct and incorrect words. We also evaluated the two features by using them for tagging the recognition on a separate test set as correct or incorrect. In addition, we compared their performance to similar tests using the N-best List Homogeneity (NBH) and Language Model Jitter (LMJ) features [1]. We briefly outline the latter two below.

Like LD and ND, NBH is computed for each word instance w_i in the original recognition hypothesis. Basically, one searches for instances of w_i near its original segmentation in the N-best list. The ratio of matches found to the size of the N-best list is the NBH measure. (The original implementation computed the ratio by weighting each N-best entry by its total likelihood. We have not done so in our experiments.)

LMJ is computed as the fraction of times a word instance remains present in the recognition hypothesis under varying language weight and word insertion penalties. (In our experience, word insertion penalty has played a minor part; we have not varied this parameter.) We trained NBH and LMJ on the same data as LD and ND.

The experiments performed involved using the confidence measures to tag recognition on a test set as correct or incorrect. The test set included about 6000 words from the Broadcast News domain. Briefly, each word in the recognition was annotated with a confidence score; i.e., probability of being correct as determined by the training set statistics. (Actually, each word had four different annotations for the four measures.) A word was tagged as being correct if its confidence score exceeded a chosen threshold.

The performance of each measure was evaluated by repeating the tagging at several different thresholds.

Based on the tagging, we computed the following two figures for the four confidence measures:

1. *Contamination Rate:* the ratio of the number of incorrect words tagged as correct to the total number of words tagged as correct.
2. *False Alarm Rate:* the ratio of the number of correct words tagged as incorrect to the total number of words tagged as incorrect.

An ideal tagger would have a contamination rate and false alarm rate of zero. The confidence measures can be evaluated by how close they get to the ideal tagger.

4. Results

Figures 5 and 6 show the contamination and false alarm rates for the four confidence measures, at different levels of tagging. As the threshold is varied, the fraction of words tagged as correct changes. From Figure 5 we can see that as the fraction of words tagged as

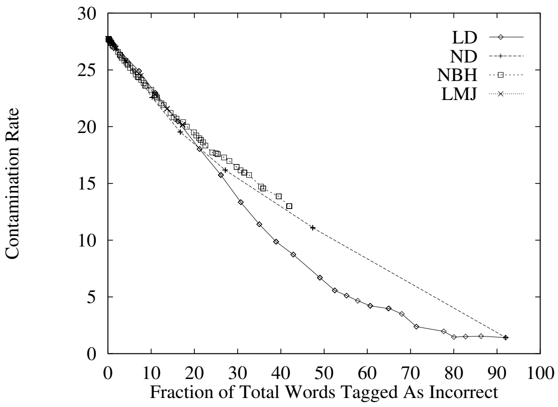

Figure 5: Contamination rate vs the fraction of words tagged as incorrect, for each confidence measure.

incorrect becomes greater, LD performs the best. That is, the remaining words that are tagged as correct are less contaminated with actually incorrect words.

From Figure 6 we see that as the threshold is set to tag incorrect words more aggressively, LD agains performs best; it has the lowest false alarm rate. At somewhat lower rates of tagging, ND seems to be the best. Overall, LD or ND dominates over a wide range of the graph.

A second problem with both NBH and LMJ is that they can be evaluated for only a portion of the entire graph. The reason is that a large fraction of the data, whether correct or incorrect, falls in exactly one point. This is especially true of LMJ; for over 60% of the words, the LMJ score is 1; their recognition is unaffected as the language weight is changed. Thus, discrimination between correct and incorrect words is impossible for that fraction of data. This problem is less severe for ND, and not at all for LD.

We have also used LD and ND jointly, as described in Section 2.3. The main benefit we have observed in this case is that the joint

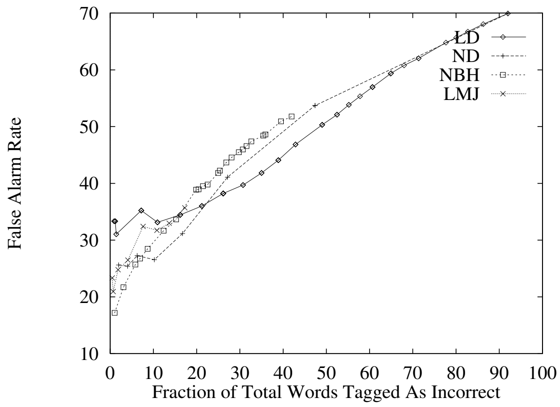

Figure 6: False alarm rate vs the fraction of words tagged as incorrect, for each confidence measure.

measure achieves the better performance of LD at the higher end of the graphs, and that of ND at the lower end.

5. Conclusion

We have introduced *Likelihood Dependence* and *Neighborhood Dependence* as two new features for use in confidence annotation. We have seen that together the two outperform other established measures over a wide range of operation. LD, in particular, seems to be significantly better than any of the others individually. We are also evaluating the application of these features in other areas, including confidence annotation for a medium-vocabulary, real-time system interactive system, and in improving the word error rate of speech recognition based on the additional confidence information.

Acknowledgements. This research was sponsored in part by the Department of the Navy, Naval Research Laboratory under Grant No. N00014-93-1-2005. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

References

1. Chase, L., *Error-Responsive Feedback Mechanisms for Speech Recognizers*, Ph.D thesis, Carnegie Mellon University, Tech. Report CMU-RI-TR-97-18, April 1997.
2. Cox, S. and Rose, R., *Confidence Measures for the Switchboard Database*, ICASSP, pp 511-514, May 1996.
3. Jeanrenaud, P. *et al.*, *Large Vocabulary Word Scoring as a Basis for Transcription Generation*, Proc. Eurospeech, pp 2149-2152, 1995.
4. Koo, M-W. *et al.*, *A New Decoder Based on a Generalized Confidence Score*, ICASSP, Vol. 1., pp 213-216, May 1998.
5. Ravishankar, M., *Some Results on Search Complexity vs Accuracy*, DARPA Speech Recognition Workshop, pp 104-107, Feb. 1997.
6. Seymore, K. *et al*, *The 1997 CMU Sphinx-3 English Broadcast News Transcription System*, ARPA SLT Workshop, pp 55-59, Feb. 1998.
7. Wessel, F. *et al.*, *Using Word Probabilities as Confidence Measures*, ICASSP, Vol. 1., pp 225-228, May 1998.