KEYWORD-BASED DISCRIMINATIVE TRAINING OF ACOUSTIC MODELS[1]

Eric D. Sandness and I. Lee Hetherington

Spoken Language Systems Group
MIT Laboratory for Computer Science
200 Technology Square, Cambridge, Massachusetts 02139, USA
eric.sandness@speechworks.com, ilh@sls.lcs.mit.edu

ABSTRACT

In this paper, we investigate a new discriminative training technique which focuses on optimizing a keyword error rate, rather than the error rate on all words. We hypothesize that improvements in keyword error rate correlate with improvements in understanding error rates. Keyword-based discriminative training is accomplished by modifying a standard minimum classification error (MCE) training algorithm so that only segments of speech relevant to keyword errors are used in the acoustic model training. When both the standard and keyword-based techniques are used to adjust mixture weights, we find that keyword error rate reduction compared to baseline maximum likelihood (ML) trained models is nearly twice as large for the keyword-based approach. The overall word accuracy is also found to be improved for keyword-based training, and we run several experiments to investigate this phenomenon.

1. INTRODUCTION

Discriminative training has often been shown to improve the accuracy of speech recognition systems which use ML estimation. Many variants of discriminative training have been studied, using different objective criteria and parameters to be optimized in a wide variety of recognition contexts. Most studies have observed similar gains from the various algorithms, depending on the difficulty of the recognition task.

The goal of discriminative training has usually been to optimize word error rates for a recognizer. But word error rate is not necessarily the most meaningful metric in a conversational system. Ultimately, it is only important that an utterance is properly understood, even if some words in the utterance are not recognized correctly. In most utterances, certain keywords must be recognized for correct understanding to take place, while other unimportant words may be confused without affecting the sentence's meaning. By focusing on recognizing the keywords, the utterance understanding rate can be maximized. The discriminative training algorithm presented in this work aims to adjust the acoustic models to focus on keyword recognition.

Several techniques, such as keyword spotting, already exist for focusing the recognizer on a set of keywords. A disadvantage of these techniques is that they require the recognition algorithm to be modified. With our discriminative training technique we hope to train up a set of models that can be used in the recognizer exactly as before, while still improving keyword recognition.

We begin using a discriminative training algorithm based on an utterance-level MCE criterion [2]. In this work the acoustic model parameters to be optimized are the mixture weights in a set of Gaussian mixture models. After measuring accuracy gains using this standard algorithm we introduce a modification to focus the training on proper keyword recognition. This modification leads to significantly better keyword recognition than when either ML-trained models or the previous discriminative training algorithm are used. Surprisingly, we also find that the new algorithm also improves *overall* word error rates more than the previous algorithm. We propose an explanation for this phenemenon and run several experiments to test our hypothesis.

2. EXPERIMENTAL FRAMEWORK

Recognition experiments for this work are conducted using the JUPITER corpus [8]. This corpus consists of spontaneous speech data from a live telephone-based weather information system with a vocabulary of about 2,000 words. The data is divided into several training and test sets. Two training sets are used containing 12,000 and 18,000 utterances, named *train_12000* and *train_18000*, respectively. We use a test set containing 500 in-vocabulary utterances named *test_500*, and another set with 2,500 in- and out-of-vocabulary utterances named *test_2500*.

Recognition is performed using the SUMMIT segment-based speech recognizer [3, 4]. Boundary-based diphone models, centered at hypothesized phonetic boundaries, are used exclusively for acoustic modeling in this work. The models are mixtures of diagonal Gaussians in a 50-dimensional feature space. There can be up to 50 Gaussians per mixture, depending on the amount of training data available for the model. For this paper, we used only a word-class bigram in order to concentrate on acoustic modeling gains.

3. DISCRIMINATIVE TRAINING

This section describes our basic discriminative training algorithm. We use an utterance-level scoring criterion similar to that of [1]. For each training utterance, complete recognizer scores are computed for the correct word sequence and an N-best list of competing hypotheses obtained during statistics collection. These scores are a sum of acoustic and non-acoustic (i.e., lexical and language model) scores. The acoustic scores change

[1] This research was supported by DARPA under contract DAAN02-98-K-003, monitored through U.S. Army Natick Research, Development and Engineering Center; and contract N66001-99-1-8904, monitored through Naval Command, Control and Ocean Surveillance Center.

and are recomputed at each iteration of the training process as the mixture weights are altered, while the non-acoustic scores of course remain the same.

The recognizer scores are used to compute an MCE objective function similar to the one in [5]. The form of the cost function for each utterance is:

$$f_s(X_s, \Lambda) = \frac{1}{N_{h,s}} \sum_{h=1}^{N_{h,s}} \frac{1}{1 + e^{\rho(g_{c,s}(X_s,\Lambda) - g_{h,s}(X_s,\Lambda))}}$$

where X_s denotes the sequence of acoustic observation vectors for the sentence, Λ represents the classifier parameters, $g_{c,s}$ is the log recognizer score for the sentence's correct word string, $g_{h,s}$ is a log recognizer score for a competing hypothesis, $N_{h,s}$ is the number of competing hypotheses in the sentence's N-best list, and ρ is a rolloff which determines how sharply the function transitions. This function differs from the basic form in [5] only in that the summation is done outside the sigmoid instead of inside the exponential. The rolloff ρ is set to a moderate value of 4.0 throughout this work. The complete objective function is just the average of all of the cost functions:

$$\mathcal{F}_{MCE}(\Lambda) = \frac{1}{N_s} \sum_{s=1}^{N_s} \frac{1}{N_{h,s}} \sum_{h=1}^{N_{h,s}} \frac{1}{1 + e^{\rho(g_{c,s}(X_s,\Lambda) - g_{h,s}(X_s,\Lambda))}}$$

where N_s is the total number of training utterances.

Using the objective function, only the mixture weights are updated according to:

$$\hat{w}_{i,j} = w_{i,j} + \epsilon \frac{\partial \mathcal{F}}{\partial w_{i,j}}$$

where $w_{i,j}$ is the j^{th} Gaussian of the i^{th} mixture, and ϵ is a step size. The derivatives are calculated numerically, and all derivative calculations are done before any alterations are done.

The baseline set of ML-trained models is used to provide the initial mixture weights, and then the above steps are iterated until convergence is observed. Twenty percent of the training data is set aside for measuring training progress.

The first two rows of Table 4 provide word error rates for the baseline models and a set of models trained on *train_12000* using the above algorithm. The discriminative training produces relative error rate reductions of 6.7% for *test_500* and 2.7% for *test_2500*.

4. USING HOT BOUNDARIES

To this point, we have not addressed our goal of focusing the training on the proper handling of keywords. The discriminative training algorithm in the previous section treats all parts of a training utterance equally. Now we wish to place more emphasis on the parts of an utterance that may be relevant to keyword recognition and ultimately understanding. We will need a way to distinguish between the potentially important and unimportant segments of the training utterances. We call the boundaries lying in the important segments the *hot boundaries.*

The hot boundaries are chosen as follows. For each training utterance, a list of the keywords contained in the correct hypothesis is compared to lists of keywords in each of the competing hypotheses. When a competing hypothesis has the same keywords as the correct hypothesis, regardless of their positions in the sentence, no hot boundaries are generated. When there is a mismatch in the keyword lists, all of the boundaries spanning all of the mismatched keywords are flagged as hot boundaries for the utterance. In this way, we hope to capture all of the regions of the utterance that have a high risk of causing a keyword recognition error.

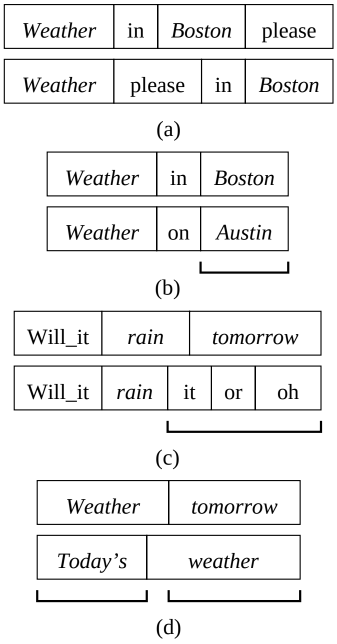

Figure 1: For each example, the top row represents the time alignment of the correct hypothesis and the bottom row an incorrect hypothesis. The bracket below indicates the region of the utterance that takes part in keyword-based discriminative training, with keywords indicated by *italics.*

The process of choosing the hot boundaries is best illustrated using several synthetic examples. These examples are shown in Figure 1. Notice in example (a) that even though the keywords do not appear in the same positions in both hypotheses, the keyword lists for the hypotheses are the same, and thus this pair does not produce any hot boundaries. Example (b) is straightforward, and shows that hot boundaries are only produced where there are mismatched keywords, not mismatches in other words. Example (c) shows that insertions or deletions of keywords, not just keyword exchanges, produce hot boundaries. Finally, example (d) shows that hot boundaries are produced at all positions spanned by a mismatched keyword in any hypothesis—in this case this includes almost the entire utterance.

The hot boundaries are the key to focusing the training on the keywords. To accomplish this, we simply change the way scores are computed on the training data. Previously, the scores were the sum of the acoustic and non-acoustic scores for each utterance hypothesis. The acoustic scores were the sums of the scores at each boundary. Now, we modify this score to be the sum of the scores at each *hot* boundary. Other boundaries do not contribute to the acoustic score. The data associated with those bound-

Place Names	*Weather Terms*	*Dates/ Times*
Boston	snow	tomorrow
India	humidity	o'clock
MIT	weather	January
Asia	advisories	sixth
Beijing	extended	Saturday

Table 1: Typical keywords.

aries will then have no effect on the objective function, hence it can have no effect on the parameter alterations. Training is only being influenced by the important segments of the training utterances.

We also scale down the non-acoustic score to account for the fact that less boundaries are contributing to the acoustic score. This keeps the relative importance of both types of scores roughly the same. The true non-acoustic score is multiplied by the fraction of the utterance's boundaries that are hot.

5. EXPERIMENTS

We now wish to determine how much the method of hot boundaries will improve keyword accuracies. The first issue to consider is how to obtain the list of keywords. For this experiment we manually choose a list of 1066 words out of JUPITER's 1957-word vocabulary. Table 1 illustrates typical keywords; they are mostly place names, weather-specific terms, and date/time words.

Hot boundaries are determined using the keyword list. Training is performed using both the *train_12000* and *train_18000* sets with N-best lists of 20 hypotheses. The reason for including the *train_18000* set is that much of the training data goes unused. Any competing hypothesis with the same keywords as the correct hypothesis is thrown away during statistics collection, since it cannot result in any parameter optimizations. If every competing hypothesis gets thrown away, the entire training utterance gets thrown away. For this reason the number of training utterances used at training time will be lower than the number in the training set. In this experiment, it turns out that only about 70% of the utterances survive the statistics collection phase. Thus, compared with non-keyword training on *train_12000*, using *train_12000* measures the effect of keyword training when the same amount of training data is *available*, while using *train_18000* measures the effect when similar amounts of training data are actually *used*.

Improvements in keyword error rate on *test_500* and *test_2500* relative to the baseline models are shown in Table 2 for three training runs. The keyword error rate barely decreases from the baseline when using standard MCE training; it appears most of the previously observed overall error rate reduction comes from other words. A much larger improvement in keyword error rate is observed when keyword-based training is performed. Almost three times as many keyword errors are corrected with keyword-based training compared to standard training when *train_12000* is used for both training runs. The keyword error rate increases still more when *train_18000* is used. It is apparent that our training algorithm is indeed resulting in better keyword recognition.

Training	*test_500*	*test_2500*
ML	6.0	13.9
MCE, *train_12000*	5.7	13.6
KB, *train_12000*	5.2	13.1
KB, *train_18000*	5.0	12.9

Table 2: Keyword error rates for baseline ML-trained models, standard MCE training, and keyword-based (KB) training.

Training	*test_500*
ML	19.44
MCE, *train_12000*	18.56
KB, *train_12000*	18.18

Table 3: Understanding error rates for baseline ML-trained models, standard MCE training, and keyword-based (KB) training.

Because our eventual goal is to improve understanding error rates more so than word error rates, we compute a measure of understanding error rate that is equal to the sum of substitutions, insertions, and deletions of semantic frame entries following the method described in [6]. Table 3 summarizes the results as measured on *test_500*, which we see correlate with the keyword error rate results of Table 2. We find that the keyword-based discriminative training results in a larger decrease in understanding error rate as compared to full discriminative training.

It is also informative to observe the effect of keyword-based discriminative training on overall word error rate. We expect this error rate to be roughly the same as the ML word error rate. Since non-keywords are ignored during the training, we expect errors on these words to increase, offsetting the gains achieved on the keywords. This would result in little net change in the overall word accuracy. The overall word error rates actually observed for the two keyword-based training runs are listed in Table 4 underneath the accuracies for ML-trained models and models trained using the standard MCE algorithm. We immediately see that, contrary to our expectations, using keyword-based training on *train_12000* decreases the overall word error rate considerably more than standard MCE training, using either *test_500* or *test_2500*. We seem to be improving the modeling not only for the keywords, but for all the words!

Our hypothesis is that with standard MCE training, the training data associated with a subset of the vocabulary can actually *hurt* model accuracy. For example, most function words are usually unstressed, poorly articulated, and not necessarily important to the meaning of an utterance. The acoustic features derived from within these words are likely to be erratic since the speaker may be sloppy in articulating these words. Thus the observation vectors for the various subword units may not tend to fall into neat clusters. When Gaussian means and variances are trained, they may be skewed away from their "correct" locations by the presence of these somewhat noisy observation vectors. Thus modeling of more precisely articulated realizations of the subword units may be harmed in an attempt to model realizations which are highly variable.

According to this hypothesis, the reason why keyword-based

Training	*test_500*	*test_2500*
ML	10.4	18.3
MCE, *train_12000*	9.7	17.8
KB, *train_12000*	9.3	17.4
KB, *train_18000*	9.1	17.3
OFW, *train_12000*	9.4	17.4
OFW, *train_18000*	9.3	17.4

Table 4: Overall word error rates for baseline ML-trained models, standard MCE training, keyword-based (KB) training, and omitted-function-words (OFW) training.

training would improve overall word accuracy is that it does not use data from these function words when updating the model parameters. As a result, it should focus the training effort on improving the classification of the more predicatable (i.e., well-enunciated) speech. This would not have much effect on the classification of feature vectors from the unimportant words, since these are already spread across a large region of the feature space.

We set up two experiments to test this hypothesis. In the first experiment, we use the same training process as in the previous keyword-based experiments, except that a different list of keywords is used. This time, the list consists of all words in the vocabulary *except* a manually chosen set of 148 words. We placed any words in this set that we thought were unimportant to sentence understanding and were likely to be poorly articulated (e.g., function words). If our hypothesis is correct, we expect the overall word accuracies on the test sets to approach the levels achieved by the previous keyword-based training experiments, since the poor quality data is excluded from both training runs.

Referring again to Table 4, the overall word accuracies using both *train_12000* and *train_18000* can be found on the last two rows, labeled as OFW (omitted function words). We see that they are indeed nearly as high as those for the keyword-based training experiments, providing support for our hypothesis. We might expect beforehand that the accuracies would not go quite as high in this experiment, since there is a chance that some function words were missed when the list of 148 words was chosen. In any case, the most important factor in increasing overall word accuracies seems to be the absence of function word data, not the precise choice of the keyword list.

In the second experiment, we again use the same keyword-based training process, but this time the set of 148 function words is treated as the list of keywords. We expect that if including the function word data reduces accuracy gains in training, using *only* the function word data should result in an accuracy decrease compared to the ML-trained models. Models are trained using this strategy with *train_12000*, and word accuracies are measured on *test_2500*. Indeed, we find that the final overall word acuracy is 81.1%, down from 81.7% for the ML-trained models. The accuracy measured on only the function words is 83.3% for the ML models, and 83.6% for the new models, showing that the training is at least able to improve accuracy on the target words a bit. It is clear, though, that the effect of training on these function words is to greatly lower the accuracy on the other words in the vocabulary.

6. CONCLUSION

The experiments in this work have indicated that it is possible to improve the recognizer's accuracy on a subset of words in the vocabulary using discriminative training. This is useful for ensuring that the most important words in an utterance have the best chance of being recognized correctly, thus improving the understanding error rate. It is possible, however, that the benefits of this technique may be limited to domains similar to JUPITER: moderate-sized vocabulary, spoken dialog systems. Naturally, focusing on certain keywords seems less appealing in dictation systems, where it is equally important to recognize every word correctly. Also, as vocabularies get larger and larger, it becomes harder to identify a small set of keywords that are much more important than other words. For systems like JUPITER, though, keyword-based discriminative training seems to have the potential to offer substantial increases in utterance understanding rates. Since conversational systems like this are becoming more and more common, the technique might be applicable in many circumstances.

In our pursuit of lower keyword error rates we have also discovered an insight into improving the acoustic models overall. It seems that preselecting certain higher quality parts of the training data can increase the general usefulness of the resulting models. This insight could have applications in types of training other than discriminative training. For example, if the training tokens derived from function words are discarded prior to ML training, it is possible that accuracy gains will still be observed. The experiments in this paper support the idea that realizations of subword units can vary widely depending on their positions in an utterance, with some realizations being much more precise than others. Perhaps even greater accuracy gains are possible if this phenomenon is studied more closely and exploited to a greater degree.

7. REFERENCES

1. F. Beaufays, M. Weintraub, and Y. Konig, "Discriminative mixture weight estimation for large gaussian mixture models," in *Proc. ICASSP*, Phoenix, AZ, pp. 337–340, Mar. 1999.
2. W. Chou, B.-H. Juang, and C.-H. Lee, "Segmental GPD training of HMM based speech recognizer," in *Proc. ICASSP*, San Francisco, CA, pp. 473–476, Mar. 1992.
3. J. Glass, J. Chang, and M. McCandless, "A probabilistic framework for feature-based speech recognition," in *Proc. ICSLP*, Philadelphia, PA, pp. 2277–2280, Sept. 1996.
4. J. R. Glass, T. J. Hazen, and I. L. Hetherington, "Real-time telephone-based speech recognition in the jupiter domain," in *Proc. ICASSP*, Phoenix, AZ, pp. 61–64, Mar. 1999.
5. B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Transactions on Signal Processing*, vol. 40, no. 12, pp. 3043–3054, Dec. 1992.
6. J. Polifroni, S. Seneff, J. Glass, and T. J. Hazen, "Evaluation methodology for a telephone-based conversational system," in *Proc. LREC*, Granada, Spain, pp. 43–50, May 1998.
7. E. D. Sandness. Ph.D. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, June 2000.
8. V. Zue, S. Seneff, J. Glass, J. Polifroni, C. Pao, T. Hazen, and L. Hetherington, "JUPITER: A telephone-based conversational interface for weather information," *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 1, pp. 85–96, Jan. 2000.