DISCRIMINATIVE TRAINING OF LANGUAGE MODELS FOR SPEECH RECOGNITION

*Hong-Kwang Jeff Kuo, Eric Fosler-Lussier, Hui Jiang, Chin-Hui Lee**

Bell Labs, Lucent Technologies, 600 Mountain Ave., Murray Hill, NJ 07974-0636, U.S.A.
{kuo,fosler,hui,chl}@research.bell-labs.com

ABSTRACT

In this paper we describe how discriminative training can be applied to language models for speech recognition. Language models are important to guide the speech recognition search, particularly in compensating for mistakes in acoustic decoding. A frequently used measure of the quality of language models is the perplexity; however, what is more important for accurate decoding is not necessarily having the maximum likelihood hypothesis, but rather the best separation of the correct string from the competing, acoustically confusible hypotheses. Discriminative training can help to improve language models for the purpose of speech recognition by improving the separation of the correct hypothesis from the competing hypotheses. We describe the algorithm and demonstrate modest improvements in word and sentence error rates on the DARPA Communicator task without any increase in language model complexity.

1. INTRODUCTION

Statistical language models are often used in speech recognition based on the maximum *a posteriori* decision rule to find the optimal word sequence $\bar{W}$ for a given speech signal X:

$$\bar{W} = \underset{W}{\operatorname{argmax}} P(W|X) = \underset{W}{\operatorname{argmax}} \frac{P(X|W)P(W)}{P(X)}, \quad (1)$$

where $P(X|W)$ is the acoustic model, and $P(W)$ is the language model. Traditionally, language models have been trained to have maximum likelihood (P(W)) or minimum perplexity. Although a lower perplexity is often correlated with a better recognition performance, the correlation is not perfect. In fact, one could sometimes have a lower perplexity, yet not see a lower error rate. The traditional argument that what we really want is minimum classification error, not maximum likelihood, also holds here. For example, a measure of language model based on competing hypotheses has been proposed [2], and it was argued that this measure has a higher correlation with the word error rate than the traditional measure of perplexity. Furthermore, when assumptions made about the model are incorrect and the amount of data used to train the model not sufficient, maximum likelihood training yields a suboptimal solution. In such cases discriminative training based on minimum classification error yields a better solution [4].

Discriminative training of the language model has recently been suggested to improve Chinese speech recognition [1]. In that paper, the training corpus was manipulated to increase or decrease the counts of word sequences to compensate for errors that were observed during speech recognition. In addition, new words and word sequences from a tuning set were added to the language model. It is unclear how much of the reported improvements were due to the reduction of out-of-vocabulary words and how much to

the manipulation of word counts. The amount by which the word counts should be adjusted at each step was also not given.

In this paper, we propose a model-based approach for discriminative training of language models. The language model is trained using the generalized probabilistic descent (GPD) algorithm [4, 3] to minimize the string error rate. The goal is to adjust the language model to overcome acoustic confusion and achieve minimum recognition error. Details of this framework are given next.

2. DISCRIMINATIVE TRAINING

In this section, we describe how the parameters of an n-gram language model, originally trained using the conventional maximum likelihood criterion, are adjusted to achieve minimum sentence error through improving the separation of the correct word sequence from competing word sequence hypotheses.

Given an observation sequence X_i representing the speech signal and a word sequence $W = w_1, w_2, \ldots, w_n$, we define a discriminant function that is a weighted combination of acoustic and language model scores:

$$g(X_i, W; \Lambda, \Gamma) = \alpha \log P(X_i|W, \Lambda) + \log P(W|\Gamma), \quad (2)$$

where Λ is the acoustic model, Γ is the language model, and α is the inverse of the language model weight. A common strategy for a speech recognizer is to select the word sequence W_1 with the largest value for this function:

$$W_1 = \underset{W}{\operatorname{argmax}} g(X_i, W; \Lambda, \Gamma). \quad (3)$$

A list of N-best word sequence hypotheses is defined as follows:

$$W_r = \underset{W \neq W_1, \ldots, W_{r-1}}{\operatorname{argmax}} g(X_i, W; \Lambda, \Gamma), \quad (4)$$

where W_r is the rth best hypothesized word sequence.

Let W_0 be the known correct word sequence. We would like to compare the discriminant function for W_0 and that for N competing word sequences $\{W_1, W_2, \ldots, W_N\}$ hypothesized by the recognizer that are different from W_0. For this purpose, the misclassification function is defined to be:

$$d(X_i; \Lambda, \Gamma) = \begin{array}{l} -g(X_i, W_0; \Lambda, \Gamma) \\ +G(X_i, W_1, \ldots, W_N; \Lambda, \Gamma), \end{array} \quad (5)$$

where the anti-discriminant function representing the competitors is defined as:

$$G(X_i, W_1, \ldots, W_N; \Lambda, \Gamma) = \log \left(\frac{1}{N} \sum_{r=1}^{N} \exp[g(X_i, W_r; \Lambda, \Gamma)\eta] \right)^{\frac{1}{\eta}}. \quad (6)$$

η is a positive parameter that controls how the different hypotheses are weighted. In the limit as $\eta \rightarrow \infty$, the anti-discriminant function is dominated by the biggest competing discriminant function: $G(X_i, W_1, \ldots, W_N; \Lambda, \Gamma) \rightarrow g(X_i, W_1; \Lambda, \Gamma)$. Therefore

*Chin-Hui Lee is currently a visiting professor at the Department of Computer Science, National University of Singapore, Singapore 117543. Email: chl@comp.nus.edu.sg

in this limit, the misclassification function is the difference between the scores of the best competitor and the correct string.

The interpretation of the misclassification function is that if it is positive, a sentence error is made, if negative, the hypothesized string is correct. That is, $d(X_i; \Lambda, \Gamma) > 0$ implies an error in sentence recognition: the discriminant function for the correct word sequence is less than the anti-discriminant function of competing word sequences. To formulate an error function appropriate for gradient descent optimization, a smooth, differentiable function ranging from 0 to 1 such as the sigmoid function is chosen to be the *class loss function*:

$$l(X_i) = l(d(X_i)) = \frac{1}{1 + \exp(-\gamma d(X_i) + \theta)}, \qquad (7)$$

where γ and θ are constants which control the slope and the shift of the sigmoid function, respectively.

Using the GPD algorithm, the parameters of the language model can be adjusted iteratively (with step size ϵ) using the following update equation to minimize the recognition error:

$$\Gamma_{t+1} = \Gamma_t - \epsilon \nabla l(X_i; \Lambda_t, \Gamma_t). \qquad (8)$$

The problem of jointly training the acoustic and language model discriminatively is important but complicated. For simplicity, we here focus on training only the language model discriminatively, keeping the acoustic model constant. The gradient of the loss function becomes

$$\nabla l = \frac{\partial l_i}{\partial d_i} \frac{\partial d(X_i; \Lambda, \Gamma)}{\partial \Gamma}, \qquad (9)$$

where the first term is the slope associated with the sigmoid class-loss function and is given by:

$$\frac{\partial l_i}{\partial d_i} = \gamma l(d_i)(1 - l(d_i)). \qquad (10)$$

If we regard Γ as being parametrized by the n-gram log probabilities, to compute $\frac{\partial d(X_i; \Lambda, \Gamma)}{\partial \Gamma}$, we can take the partial derivatives with respect to each of the log probability parameters. Without loss of generality, we use a bigram language model as an example. Let $p_{w_x w_y} = \log P(w_2|w_1)$ be the bigram log probability for the bigram $w_x w_y$. Using the definition of d in Equation 5 and after working out the mathematics, we get:

$$\frac{\partial d(X_i; \Lambda, \Gamma)}{\partial p_{w_x w_y}} = \left[-I(W_0, w_x w_y) + \sum_{r=1}^{N} C_r I(W_r, w_x w_y) \right], \qquad (11)$$

where

$$C_r = \frac{\exp[g(X_i, W_r; \Lambda, \Gamma)\eta]}{\sum_{j=1}^{N} \exp[g(X_i, W_j; \Lambda, \Gamma)\eta]} \qquad (12)$$

and $I(W, w_x w_y)$ denotes the number of times the bigram $w_x w_y$ appears in word sequence W. These two equations hold for any n-gram, not just bigram.

Intuitively, if we focus only on the top competing hypothesis, we see that what the algorithm is doing is counting the number of bigram terms in the correct string and the competing hypothesis. All the bigrams common to both sentences cancel out and the parameters associated with such bigrams are left unchanged. For the bigrams in the correct string but not in the hypothesized string, we add a little to the log probability parameter associated with these bigrams. For the bigrams in the competing string but not in the correct string, we decrease the corresponding parameters. The

amount of increase or decrease is proportional to the step size ϵ, the value of the slope of the sigmoid function and the difference in the number of times the bigram appears. The slope of the sigmoid function is 0 for very large positive or negative d, so that no adjustments are done for a sentence whose total score of the correct string is much better (already correct) or much worse (hopeless to correct) than those of the competing strings. This effective decision boundary is defined by the parameters of the sigmoid function γ and θ.

Notice that the only dependence on the acoustic scores (more specifically, the total acoustic and language model score) in the equations are in C_r and in the slope $\frac{\partial l_i}{\partial d_i}$. In fact, if only the single best competitor is used, $C_r = 1$ and does not depend on the acoustic score at all. Thus the major influence of the acoustic scores is through the value of the sigmoid slope, which determines how much weight a particular training sample has in updating the parameters.

3. ISSUES IN LANGUAGE MODEL ADJUSTMENTS

In this section, we discuss issues that complicate the simple formulation given in the previous section. Again for simplicity and without loss of generality, we discuss only a bigram language model.

In the previous section, we showed that the bigram probabilities are adjusted during discriminative training. If a particular bigram does not exist in the language model, its probability is computed via back-off to the unigram probability according to:

$$P(w_2|w_1) = b(w_1)P(w_2), \qquad (13)$$

where $b(w_1)$ is the back-off weight for word w_1, and $P(w_2)$ is the unigram probability of word w_2.

What does it mean then to adjust $P(w_2|w_1)$, or more precisely, $p_{w_x w_y} = \log P(w_2|w_1)$? There are different choices: keep the back-off weight constant while adjusting the unigram probability,[1] keep the unigram probability constant while adjusting the back-off weight, adjust both, or create a new bigram with the backed off probability and adjust this bigram probability. In our initial experiments, we chose the last approach which creates new bigrams. However, there are good arguments for the other choices as well, since these techniques will generalize the changes made to the language model to help other instances which are not exactly the same bigram. For example, if the unigram probability is adjusted, $P(w_2|w_i)$ will be adjusted for every $w_i w_2$ that is not an extant bigram in the language model. We will also try this method and compare these two back-off strategies when a bigram does not exist: creating new bigrams or adjusting the unigrams. Finally, if only the back-off weight is adjusted, we can think of that as changing the mass allocated for unseen events following word w_1.[2] How well such changes will generalize to new data is not well understood.

A related issue arises in class-based language models, where the bigram probability is given by:

$$P(w_2|w_1) = P(\text{class}(w_2)|w_1)P(w_2|\text{class}(w_2)). \qquad (14)$$

An example of a class could be a city name, and a member of this class "Boston." Thus the adjustment to the bigram probability can be assigned to either the class bigram or to the probability of membership of word w_2 in its class.

---

[1] Strictly speaking, the back-off weight has to be re-calculated after the unigram probability has been changed.

[2] The back-off parameter $b(w_1)$ is not a free parameter, but rather a function of the assigned mass in an n-gram and its corresponding mass in the $(n-1)$-gram. One can therefore adjust the parameter by changing the proportion of assigned mass in the n-gram, distributing or removing probability mass proportionally among the extant n-grams.

Another interesting issue is whether certain bigram probabilities should be adjusted if they do not have the same contexts. As an example, suppose the correct string is "A B C D" and the competing string is "A X Y D." Potential bigrams for adjustments include $P(B|A)$, $P(X|A)$, $P(C|B)$, $P(Y|X)$, $P(D|C)$, and $P(D|Y)$. It is obvious that $P(B|A)$ should be increased and $P(X|A)$ should be decreased – what we are doing is to steal mass from $P(X|A)$ and give it to $P(B|A)$, and this is correct because they are conditioned on the same event: "left context is A." But should $P(C|B)$ and $P(D|C)$ be increased and $P(Y|X)$ and $P(D|Y)$ be decreased? These probabilities are conditioned on different events. If "Y" was an unfixable error, then we may not want to decrease $P(D|Y)$ since we want to get "D" correct. If "X" was an unfixable error, then we may even want to adjust a new bigram $P(C|X)$.

Such analysis has potential to improve the overall word error rate. However, in this paper we focus on the sentence error rate and make the simplifying assumption that all of the divergent bigrams should be adjusted: $P(B|A)$, $P(X|A)$, $P(C|B)$, $P(Y|X)$, $P(D|C)$, and $P(D|Y)$. For example, by decreasing $P(D|Y)$, we will in fact lower the total path likelihood of the competing string relative to the correct string, even though "D" was correct. In other words, $P(D|Y)$ not only affects the recognition of "D" but potentially of "Y" as well.

4. EXPERIMENTAL SETUP

The data used for experiments were collected as part of the DARPA Communicator Project (http://fofoca.mitre.org), a project with many academic and industrial participants aimed at improving natural-language mixed-initiative dialogue systems with a real back-end. The specific task was a travel reservation system allowing users to make travel arrangements for airplane flights, cars, and hotels. Naive subjects recruited by NIST were instructed to call systems built by the different participating sites.

The initial language model was built based on the transcriptions of calls collected by Colorado University prior to the NIST data collection. This set contained about 8.5K sentences. Because the callers were probably in Colorado, there was a strong bias for the departing airport Denver in the data. Therefore, to generalize the language model, the transcriptions were processed in order to build a class-based language model with a "city" concept. About 375 city/airport names were included, with initial probabilities estimated based on the airport traffic volume and weighted towards U.S. domestic cities. In addition, date, time, and other phrases were added to the data. The classes in the class-based language model were then expanded to its individual members to obtain the language model which we will refer to as the baseline in this paper.

In the summer of year 2000, subjects recruited by NIST called 9 different participating sites. Of these calls, we divided the data into two sets, those made to the Bell Labs system and those made to other sites. Our initial experiments were on a set of data collected by Bell Labs, consisting of 1395 sentences, which included NIST subjects as well as some data from other subjects.

The baseline language model has about 900 unigrams and 41K bigrams. The language model bigram perplexity of this language model with respect to the test set of 1395 sentences is 34. Most of the perplexity comes from the members of classes such as cities. Without expanding the classes, the perplexity would be below 20.

5. RESULTS

Word graphs [6] were generated for the 1395 sentences, and N-best sentences with their associated total and component acoustic and language model scores were extracted. The scores for the correct sentence were extracted from the N-best list if the correct sentence was in the list. Otherwise, the correct sentence was force-aligned to the speech to derive an acoustic score. For simplicity, during discriminative training, we ignored any sentence in which there was an out-of-vocabulary (OOV) word, including partial words

which were not in the dictionary, or if there were no competing hypotheses. Out of the 1395 sentences, 1077 had no unknown words and at least one competing hypothesis.

These 1077 sentences were first used as training data for performing discriminative training on the baseline language model. The following parameters were used: $\eta = 0.1$, $\gamma = 0.5$, $\theta = 0$, $\epsilon = 0.5$, $N = 50(max)$. During several iterations of the GPD algorithm, we observed a steady reduction in the sentence error rate, as measured by the selection of the best hypothesis out of the N-best list for each sentence. After ten iterations, the final language model was tested on the entire set of 1395 sentences by decoding using the speech recognizer. The purpose of this first experiment was to demonstrate that the algorithm is working correctly on the training data. Compared to the baseline language model, the word error rate was reduced from 19.7% to 17.8%, a relative improvement of 10%. The sentence error rate went from 30.9% to 27.0%, a relative improvement of 13%. Interestingly, the bigram perplexity increased from 34 to 35, despite the reduction in word error rate; this observation is consistent with the argument that perplexity and word error rate are not perfectly correlated.

Figure 1 shows the histogram of the misclassification function for the training data before and after discriminative training. Recall that the misclassification function is more negative when the separation between the correct and competing hypotheses is larger. The figure shows that the histogram is shifted left, demonstrating that this separation is indeed increased after discriminative training. This increase in separation can improve the robustness of the language model.
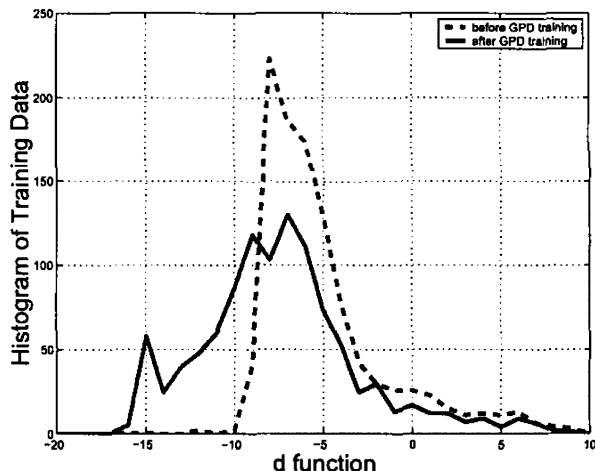

Fig. 1. Histogram of misclassification function before (dashed line) and after (solid line) discriminative training.

The next experiment is a fair experiment that involves round-robin training and testing. The set of sentences were divided into four roughly equal subsets. Three subsets were used for discriminative training and the fourth used for testing the resulting model. Then the subsets were rotated so that four experiments on unseen data were performed, and all the results were averaged.

Table 1 shows the results of the round-robin experiment. The improvement on the training sentences is larger than on the test sentences, as expected. The relative improvement for the training sentences is on the order of 11-15% while that for the test sentences is on the order of 4-6%. About 3K new bigrams were added to the language model through promotion (cf. Section 3). Using a different back-off strategy of adjusting the unigram when a bigram does not exist, almost identical results were obtained. Therefore the improvements are clearly due to discriminative training

| | Baseline | After Discriminative Training | |
Back-off strategy →		bigrams added	unigrams adjusted
No. bigrams	41K	44K	41K
Word Error Rate		abs (%diff)	abs (%diff)
Training Sentences	19.7%	17.5% (11%)	17.5% (11%)
Test Sentences	19.7%	19.0% (4%)	18.9% (4%)
Sentence Error Rate		abs (%diff)	abs (%diff)
Training Sentences	30.9%	26.4% (15%)	26.8% (13%)
Test Sentences	30.9%	29.0% (6%)	29.0% (6%)

Table 1. Round robin experiments on the baseline and discriminatively trained language models.

and not to adding new bigrams. The improvements were achieved by merely adjusting the existing parameters without increasing the complexity of the language model.

In an entirely different experiment, we used a new class-based language model that we had later built with more data (about 9K sentences collected by other sites during the NIST data collection) and additional semantic classes. The data used for discriminative training included the 9K sentences, whereas testing was done on the 1395 sentences collected by Bell Labs. Only the class probabilities were adjusted in this experiment. This has the advantage that the changes would be more generalizable. For example, the probability of any city name following a particular word could be increased or decreased. On the other hand, it neglects the actual confusion of a particular city name with other words in the vocabulary. The baseline class-based bigram language model had a word error rate of 15.9% on the 1395 sentence test set, with a sentence error rate of 25.4%. Preliminary results with discriminative training gave a relative improvement of about 3%: the word error rate dropped to 15.5% and sentence error rate to 24.7%.

6. DISCUSSION AND CONCLUSION

We have formulated a new method for discriminative training of language models using a list of N-best sentence hypotheses from the recognizer. The goal is to improve the separation between the total acoustic and language model scores of the correct string from the competing sentences. We showed that after working out the equations for GPD training, the equations simplify into terms that include weighted counts of n-gram sequences that appear exclusively in either the correct or competing sentences. From our experiments on a spontaneous speech database associated with a flight reservation dialogue system, we achieved modest improvements in word and sentence error rates of 4-6%, without any increase in the language model size.

In addition, even with simple discriminative training of class bigrams, we observed an improvement. Further improvements may be possible with tuning of the GPD parameters. Moreover, only the class probabilities were adjusted; it is conceivable that adjusting the member probabilities or working with the word n-grams directly can improve the performance.

However, there seems to be a big mismatch between the training and test errors, perhaps due to different channel conditions of different sites. If the confusible/error-prone sets are different in training and testing, one would not expect discriminative training to improve the language model for the test set. In addition, just because we know why something is wrong and how to fix it manually does not mean the algorithm can do so without sufficient data. As an example, the recognizer made a mistake: "no i said .." was recognized as "um i that .." Looking at the language model and training data, we discovered that both "i said" and "i that" were not in the training corpus, so both bigram probabilities were computed using back-off. Because the unigram probability of "said" was much smaller than for "that," the recognizer made the above error. Using the limited data set in the first experiment, discrimina-

tive training was not able to correct for this error because "i said" appeared only once in that set.

The idea of discriminative training on language models for speech recognition is an appealing one because probability mass can be stolen from acoustically distinct words with high probabilities and distributed to acoustically confusible words. Thus the goal of discriminative training of the language model is to achieve minimum recognition error, which need not imply maximum likelihood or minimum perplexity. For simplicity, we had kept the acoustic model constant while adjusting the language model. However, it has been shown that discriminative training of acoustic models depends on the language model [7]. It is therefore of considerable interest to formulate a method for such joint training of acoustic and language models.

The use of an N-best list to capture the competing hypotheses is simple but perhaps inadequate. One problem is that after the language model has been changed, certain hypotheses may emerge which were not in the N-best list. This leads to an overly optimistic estimate of the error while doing the discriminative training which may not bear out when the language model is re-inserted into the recognizer. An N-best list also often includes sentences with only minor differences, for example in only one or two words. As a result, only very few parameters out of the large number of parameters in the language model will end up getting adjusted.

An improvement would be to use the word graph directly for discriminative training. However, the formulation for the word graph is more difficult because it is not clear how to formulate the correct and competing hypotheses because of overlapping and varying durations of words. Perhaps an interesting experiment would be to convert the word graph into a word sausage [5] which explicitly lays out the competing words in order. Using word sausages can also lead to formulation of minimizing the word error rate rather than the sentence error rate.

7. REFERENCES

[1] Z. Chen, K.-F. Lee, and M.-J. Li, "Discriminative training on language model," in *Proc. ICSLP 2000*, (Beijing, China), Oct. 2000.

[2] A. Ito, M. Kohda, and M. Ostendorf, "A new metric for stochastic language model evaluation," in *Proc. Eurospeech 1999*, (Budapest, Hungary), Sept. 1999.

[3] S. Katagiri, B.-H. Juang, and C.-H. Lee, "Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method," *Proceedings of the IEEE*, vol. 86, pp. 2345–2373, Nov. 1998.

[4] S. Katagiri, C.-H. Lee, and B.-H. Juang, "New discriminative algorithm based on the generalized probabilistic descent method," in *Proc. IEEE Workshop on Neural Network for Signal Processing*, (Princeton), pp. 299–309, Sept. 1991.

[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*, vol. 14, no. 4, pp. 373–400, 2000.

[6] S. Ortmanns, H. Ney, and X. Aubert, "A word graph algorithm for large vocabulary continuous speech recognition," *Computer, Speech and Language*, vol. 11, no. 1, pp. 43–72, 1997.

[7] R. Schlüter, B. Müller, F. Wessel, and H. Ney, "Interdependence of language models and discriminative training," in *Proc. 1999 Automatic Speech Recognition and Understanding Workshop*, vol. 1, (Keystone, CO), pp. 119–122, Dec. 1999.