DISCRIMINATIVE TRAINING ON LANGUAGE MODEL

Zheng Chen, Kai-Fu Lee, Ming-jing Li

Microsoft Research China
5F. Beijing Sigma Center, No. 49. Zhichun Road, Haidian District, Beijing 100080, P.R.C.
{zhengc, kfl, mjli}@microsoft.com

ABSTRACT
Statistical language models have been successfully applied to a lot of problems, including speech recognition, handwriting, Chinese pinyin-input etc. In recognition, a statistical language model, such as the trigram model, is used to predict the probabilities of hypothesized word sequences. The traditional method that relies on distribution estimation is sub-optimal when the assumed distribution form is inapplicable, and that “optimality” in distribution estimation does not automatically translate into “optimality” in classifier design. This paper proposed a discriminative training method to minimize the error rate of recognizer rather than estimate the distribution of training data. Furthermore, lexicon is also optimized to minimize the error rate of the decoder through discriminative training. Compared to the traditional LM building method, our system achieves a 5%-25% reduction in error rate.

1. INTRODUCTION
Statistical language models have been successfully applied to a lot of problems, including speech recognition, handwriting, Chinese pinyin-input etc. In recognition, statistical language model, such as a trigram model is used to predict the probabilities of hypothesized word sequences.

Usually, Maximum Likelihood Estimation (MLE) is used in language model training. MLE, relying on distribution estimation, is probably optimal if the underlying models are correct. But the trigram model is not the “true model” of the language. Furthermore, trigram models try to separate the likely from the unlikely, without considering their actual confusability. However, for recognition, the relative scores of candidates are more important than the absolute scores.

Moreover, during training, only a small fraction of the data all over the world is selected as the training data. The inadequate training data make it difficult to obtain complete knowledge of the form of the data distribution. So the traditional method relying on distribution estimation is sub-optimal when the assumed distribution form is not the true one, and that “optimality” in distribution estimation does not automatically translate into “optimality” in classifier design. [1]

In this paper, we apply “discriminative training”[1,2,3,4] into language model building. Different from the traditional method, “discriminative training” aims to minimize the error rate of recognizer, while “traditional statistical training” aims to optimize the estimation of the distribution. A key to the development of the discriminative method is to build an error function which can be evaluated and minimized by the system. Our approach is to first train an MLE model, and then iteratively improve it using discriminative training. There are some similar approaches, such as “Corrective Training in Computer Speech & Language” from IBM. But to our knowledge, this is the first application of discriminative training to language modeling.

In the next section, we will brief introduce the basic theory about discriminative training. Section 3 details the implementation of language model optimization through discriminative training. In the section 4, we evaluate the proposed approaches by some experiments. Finally, we give some conclusions.

2. THEORY
A statistical language model plays an important role in speech recognition. Combined with an acoustic model, it can help the system to find the most possible word strings in speech recognition. Let X be the observation of the input; in the speech recognition, X is the acoustic data; in the pinyin input method, X is the stream of the roman characters, etc. What we want to find the Chinese characters (H), which maximize the value of Pr(H | X). We can use the Bayes rule to decompose it into two problems as in equation 2.1.

\[
Pr(H | X) = \frac{Pr(X | H)Pr(H)}{Pr(X)} \quad (2.1)
\]

For any given X, Pr(X) is constant. We therefore only need to consider Pr(X | H) and Pr(H). Pr(X | H) is the acoustic model or typing model, and Pr(H) is the language model [5]. In the traditional method, we will choose H = H\textsubscript{1} if Pr(H\textsubscript{1} | X) is maximum. We call it “maximum a posteriori” (MAP) decision [1,6]. Unfortunately, lack of training data will cause it to be sub-optimal, and it cannot minimize the error rate of recognizer. Because it only aims to maximize the probability of the correct model instead of minimizing the probability of the competing incorrect models. Consider the equation 2.1, traditional maximal likelihood method does not consider the influence of Pr(X), in real, Pr(X) can be rewritten as equation 2.2.

\[
Pr(X) = \sum_{i}^{M} Pr(X | H\textsubscript{i})Pr(H\textsubscript{i}) \quad (2.2)
\]

For recognition, the relative score is more important than the absolute score. The performance of the system depends on the ability to differentiate correct and incorrect answer. We therefore use the discriminative training to minimize the sum of the
probability of incorrect models. We define a misrecognition measure \([1]\) as equation 2.3.

\[
d_i(X) = -Pr(X_i, H) + \left[ \frac{1}{M-1} \sum_{j \neq i} Pr(X_j, H) \right]^{-\eta},
\]

(2.3)

where \(\eta\) is a positive number. \(d_i(X) > 0\) implies misrecognition and \(d_i(X) \leq 0\) means a correct decision. When \(\eta\) approaches \(\infty\), the bracket becomes \(\max_{j \neq i} Pr(X_j, H)\). By varying the value of \(\eta\), we can take all the competing models into consideration.

We can use sigmoid function to define the loss function \([1]\) as in equation,

\[
l_i(X) = l(d_i(X)) = \frac{1}{1 + \exp(-d_i(X) + \theta)}
\]

(2.4)

So we try to find suitable parameters to minimize the loss \(l(X)\) \([1]\).

\[
l(X) = \sum_{i=1}^{M} l_i(X) = \sum_{i=1}^{M} l(d_i(X))
\]

(2.5)

### 3. Optimization by Discriminative Training

The training process is based on the recognition results on original language model. Traditional maximum likelihood estimation can be used to build the original language model, such as a trigram model. For each sentence in training corpus, the corresponding recognition results would be obtained with the statistical language model. The correct answer and the hypothesis can be aligned through dynamic programming. For each word pair, we try to simultaneously enhance the correct word pair and weaken the error word pair. All these modifications can be done on count file of word pairs directly. After discriminative training, we can train another language model from the updated count file. After several iterations, we could reduce the error rate of recognizer.

For example, supposing we are training a trigram language model and \(S_i\) is one sentence in the training corpus. Based on original language model, \(S_i\) can be segmented into \((w_1, w_2, L, w_n)\). After recognition, new segmentation results \((w'_1, w'_2, L, w'_n)\) will be obtained. We can align these two results, and tag the error words on the word sequence. Let’s suppose \(w_j\) is aligned with \(w'_j\), and both of these two words contain error character. Then we can modify the count file as equation 3.1.

\[
C(w_{j-2}, w_{j-1}, w_j) = C(w_{j-2}, w_{j-1}, w_j) + \alpha
\]

\[
C(w_{j-2}, w_{j-1}, w'_j) = C(w_{j-2}, w_{j-1}, w'_j) - \beta
\]

\[
C(w_{j-1}, w_j, w_{j+1}) = C(w_{j-1}, w_j, w_{j+1}) + \alpha
\]

\[
C(w_{j-1}, w'_j, w_{j+1}) = C(w_{j-1}, w'_j, w_{j+1}) - \beta
\]

\[
C(w_j, w_{j+1}, w_{j+2}) = C(w_j, w_{j+1}, w_{j+2}) + \alpha
\]

\[
C(w'_j, w_{j+1}, w'_{j+2}) = C(w'_j, w_{j+1}, w'_{j+2}) - \beta
\]

(3.1)

Discriminative training also optimizes the lexicon. In the discriminative training, some new words, which are frequently wrongly decoded, are selected as word candidates. Through training, important new words are added into the lexicon so that recognitions errors due to these words are eliminated. There are three kinds of new words which are frequently wrongly decoded.

- Words that have not been considered by the linguists
- Domain specific words
- Proper nouns, such as personal names, place name, date, number, etc.

These new words are not included in the traditional dictionary. The probabilities of these new words are estimated by the trigram of single characters. In the process of recognition, the discrimination between these new words and other similar characters is little. So they are frequently decoded wrong. Through counting the error number of decoded strings, some new words are chosen and added into dictionary to increase the discrimination.
4. EXPERIMENTS

In our experiments, we applied discriminative training to speech recognition. In order to simplify the decoder, we only considered the effect of language model instead of acoustic model. More than 600 mega bytes of newspapers were collected as training data, and 2 mega bytes balanced corpus are collected as testing data. A baseline LM is built with the CMU LM Toolkit, and the language model is tuned to the proper size with entropy-based cutoff method [7].

The next step is to optimize the LM with discriminative training. We use the held-out data [5] selected from training data as the tuning data. The training corpus is divided into n parts. Each time, n-1 parts are selected as the training data and the other one as tuning data. There will be n parallel sub processes to do the discriminative training. The iteration continues until there is no significant error rate reduction on the tuning data. Then we evaluate the language model on the test-set.

From the experiments, we show that the discriminative training can obtain a 5%-25% rate of error reduction on varying sizes of language model. The results are shown as table 4.1. For 10 MB LM, the error rate reduced from 6.56% to 6.25%, the error reduction is 4.7%. While for 100 MB LM, the error rate reduced from 4.01% to 3.0%, the error reduction is 25.2%. For a large LM, more discriminative pairs can be added into the LM to discriminate the confusion between the words. On the other hand, little space is left for LM to store the confusion pairs with a small language model size. Although large model size can get better performance, appropriate size will be selected according to specific applications.

<table>
<thead>
<tr>
<th>LM Size (M)</th>
<th>Error Rate (MLE)</th>
<th>Error Rate (Discriminative Training)</th>
<th>Error Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>6.56%</td>
<td>6.25%</td>
<td>4.7%</td>
</tr>
<tr>
<td>100</td>
<td>4.01%</td>
<td>3.0%</td>
<td>25.2%</td>
</tr>
</tbody>
</table>

Table 4.1 Comparison between MLE and Discriminative Training

We also evaluated the language model on different test-sets. The results are shown as table 4.2. We found that the effect of discriminative training varies with the style of the test-sets. If the training data similar to testing data, then the effect will be significantly better. However, if the training data is different from testing data, the effect will becomes lower. The theme of many_news test-set is same as training data, so the effect is greater than other test-set. Opentest and People’s daily are also similar to the training data, so the improvement is good. However the webdata is unlike the training data, so little improvement is realized. From the experiment, we can infer that if some information about application domain can be gathered, the language model can be optimized to fit the needs of the specific domain.

<table>
<thead>
<tr>
<th>Test-Set</th>
<th>Error rate (MLE)</th>
<th>Error Rate (Discriminative Training)</th>
<th>Error Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Many_news</td>
<td>3.50%</td>
<td>3.29%</td>
<td>6%</td>
</tr>
</tbody>
</table>

Another experiment optimized the lexicon. Some word pairs which are frequently decoded wrong are chosen as the candidates of new words. We classified these new words into three categories:

- **Word consists with high frequency co-occurrences characters**, for example, 我的( mine), 你的( your), 他的( his), 她的( her), a one( one), etc. From the point of the linguists, these words cannot be included in the lexicon. But adding these words will result in approximately 1%-2% error reduction on different test-sets.

- **Domain specific words**. Lacking the information of one specific domain, the performance of system is dramatically dropped compared to the general domain. Without sufficient training data on these specific words, the probabilities of these new words cannot be estimated correctly. In our experiment, some new words about Internet are frequently decode wrong, e.g. 域名( domain name), 网页( Web page), 网址( net address), 超链接( hyperlink), etc.

- **Proper nouns**. There are many kinds of proper nouns in the corpus. Unfortunately, only a small fraction of proper nouns are included in the lexicon. Furthermore, the distribution of proper noun is sparse. So it is impossible to gather all of the proper nouns into the lexicon. Through discriminative training, some proper noun are detected and added into the dynamic lexicon [8] to improve the performance of the recognizer. In our experiment, some proper nouns are detected, e.g. 唐学通( a Chinese personal name), 葛罗夫( a foreign personal name), 安纳( a Chinese organization), 二十九岁( twenties), etc.

All these candidates are sorted by their frequency and added into the lexicon to optimize the new language model.

5. CONCLUSION

In this paper, we proposed a new approach to training LM to improve the recognition performance. But the models are not generalizable, e.g., LM for speech recognition may not be good for handwriting or spelling correction. Compared to the traditional LM construction method, our systems gets approximately 5%-25% recognition error reduction with discriminative training during language model building.

6. ACKNOWLEDGEMENT

Our thanks to DongFeng Cai, Hai-Feng Wang, JianFeng Gao for their helps on the data collection and model training.
7. REFERENCE


