DISCRIMINATIVE TRAINING IN NATURAL LANGUAGE CALL ROUTING

Hong-Kwang Jeff Kuo and Chin-Hui Lee

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

ABSTRACT

In this paper, we show how discriminative training can be used to improve classifiers used in natural language processing, using as an example the task of natural language call routing. In natural language call routing, callers are routed to desired departments based on natural spoken responses to an open-ended “How may I direct your call?” prompt. With vector-based natural language call routing, callers can be transferred using a routing matrix that is trained based on statistics of occurrence of words and word sequences in a training corpus after morphological and stop-word filtering. New user requests are represented as feature vectors and are routed based on the cosine similarity score with the model destination vectors encoded in the routing matrix. The present paper proposes the use of discriminative training on the routing matrix to improve routing accuracy and robustness. By retraining the routing matrix, a relative error rate reduction of 13-19% was achieved. Increased robustness was demonstrated in that with 10% rejection, there was a relative error rate reduction of 40%.

This is the first study of using discriminative training in an information retrieval problem, and we believe the proposed formulation is equally applicable to algorithms addressing a broad range of speech understanding, information retrieval, and topic identification problems.

In the following sections, details of the algorithm and results of experiments will be presented.

1. INTRODUCTION

Touch-tone menus for routing callers are typically cumbersome to use when there are many destinations. Call routing based on spoken utterances have been proposed as an alternative that is natural and easier to use. However, callers often do not know the specific department they need (e.g. "new car loan department"), but they do know what they want to do (e.g. "My car broke down last weekend and I need some money to get a new car"). In fact, a previous study [1] has shown that callers prefer to specify the activity they need to accomplish rather than the name of the department, by a factor of more than three to one.

In natural language call routing, callers are routed to desired departments based on natural spoken responses to an open-ended “How may I direct your call?” prompt. In designing a voice response system to handle these calls, it is not sufficient to include just the names of the departments in the vocabulary, and what the callers may say cannot be fully anticipated. Instead, requests from real callers have to be collected for training the system. Data-driven techniques are essential in the design of such systems.

In previous work [1], a vector-based information retrieval technique was introduced for performing call routing. A routing matrix was trained based on statistics of occurrence of words and word sequences in a training corpus after morphological and stop-word filtering. New user requests were represented as feature vectors and were routed based on the cosine similarity score with the model destination vectors encoded in the routing matrix.

In this paper we propose the use of discriminative training on the routing matrix to improve routing accuracy and robustness.
cosine similarity score:

\[
\text{destination } j = \arg \max_j \cos \phi_j = \arg \max_j \frac{\vec{r}_j \cdot \vec{z}}{\|\vec{r}_j\| \|\vec{z}\|}. \tag{1}
\]

A classification error occurs when the score of the correct class is less than the maximum score.

To solve the nonlinear optimization problem of achieving minimum classification error, we adopt the generalized probabilistic descent (GPD) algorithm [5, 4, 7]. In the current problem, the \( n \times m \) elements of the routing matrix are regarded as the classifier parameters to be adjusted to achieve minimum classification error by improving the separation of the correct class from competing classes. The dot product of normalized query and destination vectors is used as the discriminant function. In the training algorithm, the model destination vectors are normalized after each adjustment step in order to maintain the equivalence between the measures of dot product and cosine score used in computing the classification error rate. Intuitively, this algorithm looks at each training example and adjusts the model parameters of the correct class relative to the other classes.

Specifically, let \( \vec{y} \) be the observation vector and \( \vec{z}_j \) be the model document vector for destination \( j \). We define the discriminant function for class \( j \) and observation vector \( \vec{y} \) to be the dot product of the model vector and the observation vector:

\[
g_j(\vec{y}, R) = \vec{z}_j \cdot \vec{y} = \sum_i z_{ji} y_i. \tag{2}
\]

Note that this function is identical to the cosine score if the two vectors have been normalized to unit length.

Given that the correct target destination for \( \vec{z} \) is \( k \), we define the misclassification function as

\[
d_k(\vec{z}, R) = -g_k(\vec{z}, R) + G_k(\vec{z}, R), \tag{3}
\]

where

\[
G_k(\vec{z}, R) = \left[ \frac{1}{K-1} \sum_{j \neq k, 1 \leq j \leq K} g_j(\vec{z}, R) \right]^+ \tag{4}
\]

is the anti-discriminant function of the input \( \vec{z} \) in class \( k \) and \( K - 1 \) is the number of competing classes. Note that in the limit as the positive parameter \( \eta \to \infty \), the anti-discriminant function is dominated by the biggest competing discriminant function: \( G_k(\vec{z}, R) \to \max_{j \neq k} g_j(\vec{z}, R) \). Notice also that \( d_k(\vec{z}, R) > 0 \) implies misclassification, i.e., the discriminant function for the correct class is less than the anti-discriminant function.

A smooth differentiable 0-1 function such as the sigmoid function is chosen to be the classification error:

\[
l_k(\vec{y}, R) = l(d_k) = \frac{1}{1 + \exp^{-\gamma \eta d_k}}, \tag{5}
\]

where \( \gamma \) and \( \eta \) are constants which control the slope and the shift of the sigmoid function, respectively.

The parameter set \( R \) is adjusted iteratively according to

\[
R_{t+1} = R_t + \delta R_t, \tag{6}
\]

where \( R_t \) is the parameter set at the \( t \)-th iteration. The correction term \( \delta R_t \) is solved using the training sample \( \vec{z}_t \) given for that iteration, whose true destination is \( k \):

\[
\delta R_t = -\eta l_k(\vec{z}_t, R_t), \tag{7}
\]

where \( \eta \) is the learning step size.

Once this essential framework for discriminative training has been laid out, what is left is to work out the algebra for this particular problem. Let \( r_{uw} \) be elements of the routing matrix \( R \). Then at iteration \( t \):

\[
\nabla l_k(\vec{z}_t, R_t) = \frac{\partial l_k(\vec{z}_t, R_t)}{\partial R_t} = \frac{\partial l_k}{\partial d_k} \frac{\partial d_k}{\partial r_{uw}}. \tag{8}
\]

Note that for the \( l_k \) we have chosen,

\[
\frac{\partial l_k}{\partial d_k} = \gamma l_k (d_k) (1 - l_k (d_k)). \tag{9}
\]

From equations 2, 3, and 4, the following can be shown:

\[
\frac{\partial d_k(\vec{z}_t, R_t)}{\partial r_{uw}} = \begin{cases} -x_{uw} & \text{if } u = k \\ 0 & \text{otherwise} \end{cases} \tag{10}
\]

Therefore, given the observation vector \( \vec{z}_t \) at each iteration, each element of the routing matrix is adjusted according to:

\[
r_{uw}(t+1) = \begin{cases} r_{uw}(t) + \frac{\partial}{\partial r_{uw}} l_k(\vec{z}_t, R(t+1)) & \text{if } u = k \\ r_{uw}(t) - \frac{\partial}{\partial r_{uw}} l_k(\vec{z}_t, R(t+1)) & \text{if } u \neq k \end{cases} \tag{11}
\]

Equation 11 shows that the model vector for the correct class is adjusted differently from those of the competing classes; notice in particular the difference in sign of the adjustment. Intuitively, the score of the correct class is improved relative to the scores of the competitors by the incremental adjustments. The adjustment to the \( u \)-th component of each model vector is proportional to the learning step size \( \eta \), the size of the \( u \)-th component in the observation vector \( \vec{z}_t \), and the slope of the sigmoid function \( \frac{\partial}{\partial d_k} \) at \( d_k \). This slope is zero for very large or small values of \( d_k \) and positive in a certain region: the decision boundary which depends on \( \eta \) and \( \gamma \). Only the training data whose \( d_k \) values fall within the decision boundary will affect the model parameters significantly.

After each adjustment step, the affected models \( \vec{z}_t \) are normalized to unit length in order that the discriminant function be identical to the cosine similarity score used in classification. The training vectors are normalized once before the discriminative training.

4. EXPERIMENTAL SETUP

Experiments were performed using the training and test sets collected for the USAA call routing task as reported in [1], consisting of a total of about 4000 calls. The same set of training data was used both to construct the initial routing matrix and for performing discriminative training. Each training vector is composed of the information provided by all the customer utterances within each call session, including disambiguating follow-up utterances. Each call session has been manually routed to a destination, representing the ground truth of the correct class. There are a total
of $n = 23$ destinations and $m = 756$ terms. In the discriminative training, multiple passes are made through the entire training set. Within each pass, the order in which each training vector is processed is randomized. For testing, only the 307 unambiguous initial utterances were used from the unseen test set. The baseline system as reported in [1] has a classification rate of 92.2% or error rate of 7.8% for this same set of 307 test utterances.

5. PARAMETER SELECTION

We see from the above equations that a number of parameters for GPD training have to be chosen. $\eta$ controls the relative importance among the competitors – a larger value emphasizes the strongest competitors only. $\gamma$ and $\theta$ controls the decision boundary through modifying the shape and location of the sigmoid function. $\epsilon_{t}$ controls the step size of the gradient descent. It is reduced gradually in order to achieve stable convergence; specifically, the step size is chosen to be a function like $1/t$, but chosen so that it changes only once every 25 passes. Note that $K-1$ is the total number of competitors to the correct class. In practice, the discriminative training can be focused on just the top $M$ competitors instead of all $n-1$ classes. Another parameter is the number of passes through the training set which can be expressed as a stopping criterion, for example, when the change in the empirical loss function is less than a certain threshold.

Several experiments were run to find appropriate values of parameters to use in the discriminative training. The results do not seem to be sensitive to most of the parameters. Because the computational cost is low, we were able to run over a hundred passes through the training set and the choice of a relatively small $\epsilon_{t}$ may have reduced the sensitivity of the results to the other parameters. In the following results, we chose the following parameter values: $\eta = 2$, $\gamma = 8.0$, $\theta = 0.0$, $\epsilon_{t} = 3 \times 10^{-6}$ (initial), and $M = 4$.

6. RESULTS

Figure 1 shows the (average) empirical loss from equation 5 of the training set as a function of the number of discriminative training passes through the entire training set. The discriminative training appears to be doing the right thing in progressively decreasing the empirical loss. It can be seen from the figure that the $\epsilon_{t}$ step size is reduced over time, specifically once every 25 passes, as seen in the decrease in spacing between the data points and in the level-off of the slope of the empirical loss.

Figure 2 shows the classification error rate for the training set as a function of the number of passes through the training data. The error rate follows a generally downward trend as a function of the number of passes through the training data. After about 80 passes, the error rate has been decreased from about 9% to 7%, a relative reduction of about 20%.

The GPD algorithm is an iterative algorithm that improves the empirical loss in the training data. One issue is how to define a stopping criterion. In our experiments, we initially chose a threshold in the change of the empirical loss, i.e. we stop if the change in empirical loss after the pass through the training set is less than a certain threshold. However, it appears that the classification error rate of the training set may be a better criterion: the curve is seen to plateau even when the empirical loss is still decreasing.

Figure 3 shows the error rate curve for the unseen test set as a function of the number of discriminative training passes through the entire test set. After discriminative training over about 50 passes, the error rate has been decreased about 13%, from about 7.8% to 6.8%. This result is achieved using the transcription of the customer utterances. We want to see if the improvements also carry over when we use the actual recognized strings from the automatic speech recognizer which has a word error rate of around 30%. Table 1 shows that when the recognized strings are used for routing instead of the transcribed strings, the relative error reduction is even larger, around 19%, from 10.4% to 8.5%.

Figure 3 also shows that at some point the test error rate increases with additional passes through the data. This could be caused by the differences between the training and test conditions. For example, in the training phase, we group, for convenience, all the utterances within a session into one vector (in order to assign it unambiguously to a “ground truth” destination), but the test vectors are created from only an initial utterance.

Figure 4 shows the classification error rate of the test data when a subset is rejected using the difference in scores of the top two candidates. At 0% rejection, as seen earlier, the error rate is reduced from 7.8% to 6.8% after discriminative training. At 10% utterance rejection for the test data, there is a relative error reduction of about 40% (from 5% to 3%). At all levels of rejection, the discriminative training consistently does better than the baseline because the separation of the correct class from the
In summary, this is the first study of its kind to apply discriminative training in natural language information retrieval problems. Discriminative training is a powerful technique to improve classifiers to outperform maximum likelihood (or other counting based) classifiers by reducing the classification error rate and by increasing robustness. The specific formulation outlined in this paper can be directly applied to any existing algorithm using a routing matrix. We believe that discriminative training, which has been successful with acoustic model training and joint feature HMM design [8], will play an increasingly important role in areas of speech understanding, topic identification, and information retrieval.

8. REFERENCES


<table>
<thead>
<tr>
<th>transcription recognized strings</th>
<th>baseline 7.82%</th>
<th>after DT 6.84%</th>
<th>%change 13%</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.42%</td>
<td>8.47%</td>
<td>19%</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Classification error rate before and after discriminative training.