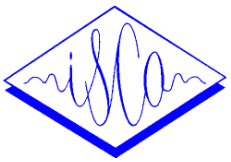

International Symposium on Chinese Spoken Language Processing (ISCSLP 2002)
Taipei, Taiwan
August 23-24, 2002

IMPROVEMENT OF THE POST-PROCESSING METHOD FOR ISOLATED WORD OOV REJECTION

Yifei Zhu,Chengrong Li,Bo Xu

National Laboratory of Pattern Recognition Institute of Automation,

Chinese Academy of Science,Bejing
{yfzhu,lich,xubo}@nlpr.ia.ac.cn

ABSTRACT

For many practical speech recognition applications, the rejection of out-of-vocabulary(OOV) words is an important issue. To make the rejection decision, confidence measures are computed in many systems. In this paper we focus on the problem of isolated word rejection and our strategy is based on a post-classifier. First of all, by using a linear classifier, we combine several promising features presented by others to obtain a confidence measure. By comparing the confidence value with the decision threshold, we can reject the OOV words. After that we present two novel features which are proved effective in our experiment. Finally, because many data sets are linearly non-separable, we present a framework based on Support Vector Machine(SVM). The experiments show that a considerable improvement(about 74%) of the equal-error-rate(EER) is achieved after all improvement strategies are integrated into the system.

1. INTRODUCTION

For many practical speech recognition applications, the rejection of out-of-vocabulary (OOV) words is an important issue, especially in dialogue systems, where acting upon a misunderstood input may incur a high cost to the user. To avoid this problem, the rejection strategy of OOV is necessary. In many systems, confidence measures are computed to make the rejection decision. To compute confidence measures, a lot of approaches are presented in the literature: the first method is to use additional models: in [1], the author used the garbage model to normalize the word scores; in [2,3], anti-models are used to compute confidence measure. The usage of additional models has the disadvantage that the training process is complex, especially in the sub word based systems.

The second important method to compute confidence measure is using the information that can be derived from the recognizer directly, which we call a "post-processing" method. In [4,5,6,7], the features from different information source: acoustic, language model, N-BEST list or word lattice, are gathered and combined to a single confidence measure.

The rejection strategy presented here is also based on the post-processing method. Because there are two important issues in this method: the first one is the selection of features and the second is the way to combine features, in this article we present some new ideas to these two problems: several new heuristic features which are promising especially in the isolated word systems and a novel SVM-based framework to combine features.

The rest of the paper is organized as follows: in the next section we introduce the post-processing method, including the selection of features and the design of the classifiers. In section 3 the experiment results are reported and analyzed. Finally in section 4, we present the conclusion.

2. THE POST-PROCESSING METHOD

The basic idea of the post-classifier is to estimate the probability that a hypothesis is correct using some features of that hypothesis after it has been made. There are two important issues in this method: the first is the selection of enough discriminative features, the second one is the design of the classifier which is used to combine all the features into a confidence measure.

2.1 Features

The key problem in the post-processing method is the selection of effective features and a variety of features have been presented in the literature[4,5,6,7], which can be assigned into one of the four categories depending on the information source: acoustic, language model, N-BEST list or word lattice. Here we investigate several promising features presented in the literature.

Duration Normalized Score(C_{DNS})[6]: Because the scores of the error results tend to be lower than the right ones, when an OOV utterance occurs, the score is probably lower than the scores of in-vocabulary utterance, which can be used as a discriminative feature.

$$C_{DNS} = \frac{acoustic\ \ score\ \ of\ \ the\ \ best\ \ hypothesis}{the\ \ duration\ \ of\ \ the\ \ word} \qquad (1)$$

C_{DNS} is used in our baseline system as the confidence measure.

Acoustic Stability(C_{ACU})[7]: The consideration of this feature is that a word is most probably correct if it is contained in the majority of the N-BEST list. In this paper, the times of the best hypothesis occurring in the N-BEST list, normalized by the number of N-BEST, is taken as the feature.

Hypothesis Density[7]: In the decoder of a speech recognizer for large vocabularies, unlikely hypotheses must be pruned from the search space. In time segments where the probability for a word is very much higher than the probability of all other words, most of the competing other words are pruned. If, on the other hand, a great number of words has similar likelihood in a time segment, no effective pruning can take place. As a high number of hypotheses with similar likelihood implies a higher probability of error, the number of links that span the time segment of a word in the most likely hypothesis should be

strongly correlated with the word error. For each word, we computed three numbers of competing links: at the word beginning, at the word end, and the average number averaged over the time segment. The resulting features are named C_{TA}, C_{TE} and $C_{AVERAGE}$, respectively.

For each word of the training set, a feature vector is computed, in which the last component is the correct/false tag and the rest are the described features. To evaluate the effectiveness of the features, we can analyze the correlation of the features with the correct/false tag. If the absolute value of the correlation coefficient is high, the corresponding feature can be regarded as 'good'. Therefore by the correlation coefficients, we can select suitable features. Table 1 shows the correlation coefficients of each feature.

Feature	Correlation
C_{DNS}	0.528
C_{ACU}	0.494
C_{TE}	-0.554
$C_{AVERAGE}$	-0.598
C_{TA}	-0.031

Table1: Correlation coefficient list

From the table, we can see that the correlation coefficient of C_{TA} is almost zero, which means it is not a useful feature in our system. Therefore we select the rest four features to conduct experiments.

Because the selection of the discriminative features is a key problem in the post-classifier method, we investigate many other features. Among them, two features are proved to be effective, which are listed below:

Normalized Score Drop(C_{NSD}): In the experiment, we find that the score of the second best hypothesis is useful to the classifier. We find that if a recognized word is a right one, there is a great difference between the score of the best hypothesis and that of the second one. While if the result is wrong, the difference is much smaller. We define it as:

$$C_{NSD} = \frac{S_1}{T_1} - \frac{S_2}{T_2} \qquad (2)$$

where S_1, S_2 are the scores of the best and second hypotheses, and T_1, T_2 are the duration of them.

Average Phone Duration(C_{APD}): We observed the recognition results of our isolated word system and find that in most cases of wrong recognition, the word length of the result is shorter than the actual result. Therefore the average phone duration of the wrong result is longer than that of the actual result.

We define C_{APD} as:

$$C_{APD} = \frac{the\ \ duration\ \ of\ \ the\ \ word}{the\ \ phone\ \ number\ \ of\ \ the\ \ result} \qquad (3)$$

The corresponding correlation coefficients of these two features with the c/f tag is listed in table 2.

Feature	Correlation
C_{NSD}	0.702
C_{APD}	0.406

Table 2: Correlation coefficient list

From the table, we can see that the two features have high correlation with the c/f tag which means they are probably useful to the system.

2.2 Design of the classifiers

Another important issue in the post-classifier method is the design of the classifier. In the literature, several different approaches are used, such as the linear classifier, Neural Network and Decision Tree. Meanwhile, in recent years, Support Vector Machine(SVM) has been shown to achieve better performance for long-standing problems than the traditional approaches. Therefore, in this article, we design a SVM-based classifier and compare it with a traditional linear classifier.

2.2.1 The linear classifier

The most commonly used classifier is the linear classifier. With a linear transformation function $g(X)=AX+a_0$, the input feature space X is transformed into a target feature space Y. By the transformation, we can get a target feature y, which is a one-dimension value in our system. Therefore the classifier problem can be solved by choosing a threshold y_{thresh}, and comparing the target feature y with y_{thresh} to decide which class the sample belongs to. Here we use fisher discriminative analysis to obtain the transformation matrix A. The disadvantage of the linear classifier is that it can only work well for the linearly separable classes.

2.2.2 The SVM-based classifier

The linear classifier can work very well for the linearly separable classes, however, many data sets are actually linearly non-separable. A recent technique, Support Vector Machine(SVM) [8], has been shown to achieve better performance than the traditional approaches under such conditions. The basic idea of the SVM method is mapping the vector into a higher dimension space using a kernel function and seeking the hyperplane in that space. In this article, we design a classifying framework based on SVM and compare it with the traditional linear classifier.

3. EXPERIMENT

3.1 Database

The experiments have been conducted on the database DB863(Mandarin speech).We selected the isolate word subset of the DB863 for our task. The selected utterances are divided into two parts: a training set and a test set. Table 3 show the composition of the data used for the training and evaluation of the classifiers. Meanwhile we use a speaker independent

Mandarin isolated word recognizer which produces the recognition results and the multiple features used in our system.

Set	Speaker	Utterance	Total length(min)
Training	55	2011	80
Test	28	1250	35
Total	83	3261	115

Table3: Database Composition

3.2 Evaluation of confidence measure

After the confidence has been computed, each recognized word is classified as either correct or false, depending on whether its confidence exceeds a certain threshold or not. Here, two different types of errors can occur. The first is a false acceptance, i.e., a false word is classified as correct, and the second is a false rejection, i.e., a correct word is classified as false. Obviously, there is a trade-off between the two types of errors, depending on the choice of the threshold. By adjusting the threshold, we can get an operation point where the false acceptance rate and the false rejection rate are equal. The false acceptance rate and the false rejection rate at this point is called equal-error-rate(**EER**). A small EER means that the system can keep the false rejection rate and the false acceptance rate small at the same time, which indicates that the classifier has a good performance. In this article, we use EER as the criterion to evaluate the performance of different methods.

3.3 Experiment results

The experiments include two aspects: the different combination of features and the selection of different classifiers. Firstly, we conduct experiments on a linear classifier, then we test the same data set on a SVM classifier.

3.3.1 The performance of the linear classifier

First of all, we use a linear classifier to investigate the effect of the features presented by other authors. In our baseline system, we use C_{DNS} as confidence measure because it is easy to obtain from the decoding results. In table 4, we can see EER in the baseline system is 29.1%. The rest of the table is the results of the combination of different features in the linear classifier.

Feature	EER	Error reduction
Baseline(C_{DNS})	23.1%	
C_{DNS} + C_{ACU}	20.2%	12.6%
C_{DNS} + $C_{AVERAGE}$	18.5%	19.9%
C_{DNS} + C_{TE}	15.1%	34.6%
C_{DNS} + $C_{AVERAGE}$ + C_{TE}	14.0%	39.4%
C_{DNS} + C_{ACU} + $C_{AVERAGE}$ + C_{TE}	13.5%	41.6%

Table 4: Performance of different feature sets

From the table, we can see that EER is reduced when more features are added into the classifier. The best result is obtained when all the features are combined. In this case, the EER is 41.6% less than that of the baseline system. From the results, we find that the combination of different features can improve the performance of the classifier. Under such a consideration, we try to add other new features into the classifier. Experiments results show that two features, which we call *Normalized Score Drop* and *Average Phone Duration*, are useful to our system. After adding them into the linear classifier, the performance is improved furthermore. The results are listed in Table 5.

Feature	EER	Error reduction
Baseline(C_{DNS})	23.1%	
All features(including novel ones)	11.0%	52.4%

Table 5: Performance of new features

Table5 shows that after two new features are added, the performance of the classifier can be improved. But the improvement is not exciting. What's more, even when we add more features into the classifier, the result can not change better. The main reason is that the data set is linearly non-separable and adding more features can not avoid this problem. Therefore, we try to improve the performance from the design of the classifier. Here we present a SVM-based framework to combine features.

3.3.2 The performance of SVM

The classifying process can be described as below: the feature vector X is computed and then it will be mapped to {-1,+1}, the process can be defined as $C:\{X\}\rightarrow\{+1,-1\}$.

It is obvious that mapping C works as a classifier and in our system SVM classifier is applied to run the task of C. In the proposed system, SVM is applied to search an optimal hyperplane in the feature space to separate positive examples (right recognition) from negative examples (error recognition). The function of resulting hyperplane of SVM can be represented as:

$$Y_{SVM}(\tilde{X}) = \sum_{i=1}^{S} y_i \alpha_i K(\tilde{X}, X_i) + b \qquad (4)$$

where $\tilde{X}$ is the decision vector. S is the number of the training "access examples", and X_i is the feature-vector for i^{th} access example. The value y_i is +1 for right recognition and –1 for error recognition. Both a_i and b are parameters of the classifier which can be learned from the training examples. $K(\tilde{X}, X_i)$ is a kernel function which represents the 'distance' between the decision-vector and those support vectors. In our system we use Gaussian kernel

$$K(s,t) = \exp(-\frac{\|s-t\|^2}{2\delta^2}) \qquad (5)$$

After we obtain $Y_{SVM}(\tilde{X})$, we can make the decision as:

$$V_{SVM}:(X,I,K)=\begin{cases}+1, & Y_{SVM}(\tilde{X})>0 \\ -1, & otherwise\end{cases} \quad (6)$$

By the means of Eq.(3), the framework of SVM is involved into the combination system.

We use the described features in the previous part and combine them in the SVM framework. Table 6 is the comparison results of the linear and SVM classifiers.

Feature	EER	Error reduction
Baseline(C_{DNS})	23.1%	
4 features in experiment1(linear)	13.5%	41.6%
6 features in experiment2(linear)	11.0%	52.4%
4 features in experiment1 (SVM)	8.1%	62.4%
6 features in experiment2 (SVM)	6.0%	74.0%

Table 6: Performance of the SVM classifier

From the table we can see that the performance of the combination system is improved when the framework of SVM is involved and the best result is obtained when all the features are combined under the SVM framework. EER in the best performance is 6.0%, which is 74% less than that of the baseline system.

4. CONCLUSION

In this paper we investigate the OOV rejection strategy based on post-classifiers. First of all, we select several promising features presented in the literature and combine them by a linear classifier to get a confidence measure. The experiment results show that the combination of multiple features can achieve better rejection performance than using individual ones. After that we add into the classifier two novel features, which are proved effective in our system. Furthermore, we present a new classifier based on SVM. The experiments prove that the new classifier performs better than the linear classifier because the data set is linearly non-separable in many cases. When all the proposed methods is used, a considerable improvement(about 74%) of the rejection performance is achieved.

5. ACKNOWLEDGEMENT

The work is supported by National Key Fundamental Research Program(the 973 Program) of China under the grant G1998030504-02 and the Science Foundation of Beijing under the grant 4022010.

6. REFERENCES

[1] E.Tsiporkova, F. Vanpoucke, H. Van hamme, " Evaluation of various confidence-based strategies for isolated word rejection", *Proc. ICASSP*, 2000.

[2] Lleida, E. and Rose, R. C., ``Efficient decoding and training procedures for utterance verification in continuous speech recognition," *Proc. ICASSP*, May 1996.

[3] R. A. Sukkar and C.-H. Lee, ``Vocabulary Independent Discriminative Utterance Verification for Non-Keyword in Subword Based Speech Recognition," *IEEE Trans. on Audio and Speech Processing*, Vol. 4, No. 6, pp. 420-429, Nov. 1996.

[4] Benoit Maison and Ramesh Gopinath "Robust confidence annotation and rejection for continuous speech recognition," in *Proc. ICASSP*,2001

[5] Burges, C. J. C. "A tutorial on support vector machines for pattern recognition", *Data Mining and Knowledge Discovery*, 2(2), pp.121-167, 1998

[6] R. San-Segundo, B. Pellom, K. Hacioglu, and W. Ward, "Confidence measures for spoken dialogue systems", *Proc. ICASSP*, Salt Lake City, 2001

[7] Thomas Kemp, Thomas Schaaf, "Estimating confidence using word lattice", *Proc. EuroSpeech*,Rhodes,1997

[8] Burges, C. J. C. "A tutorial on support vector machines for pattern recognition", *Data Mining and Knowledge Discovery*, 2(2), pp.121-167, 1998