Improving End-to-End Performance of Call Classification Through Data Confusion Reduction and Model Tolerance Enhancement

Cheng Wu, Xiang Li, Hong-Kwang Jeff Kuo, E.E. Jan, Vaibhava Goel, David Lubensky

Human Language Technology Department, IBM T. J. Watson Research Center,
P. O. Box 218, Yorktown Heights, New York 10549, USA

{chengwu, xiangli, hkuo, ejan, vgoel, davidlu,}@us.ibm.com

ABSTRACT

Two major challenges in the rapid deployment of automated natural language call routing system are the minimization of the manual effort in tagging data and reducing the impact of speech recognition errors on the call classification. In this paper we explore some novel approaches which target at these two challenges. One of our approaches enriches the training set with additional speech recognition hypotheses, automatically splits the neighboring data with its original classes, retags the training data based on a similarity measure, and elects the final result among multiple classifiers which are trained from the split data; the other approach incorporates the acoustical confusable information into call classifier to reduce the impact of the speech recognition error on the call classification accuracy. The experimental results show that our new approaches can reduce the classification error rate of an automated natural language call routing system by relative 10% in the end to end performance, using the live data collected from an enterprise call center.

1. INTRODUCTION

As the voice activated self-service, such as automated natural language call routing, plays an important role in today's call center optimization, more attentions and efforts have been absorbed in natural language call routing field [Kuo, et al, 2003]. Despite the great success of the existing approaches, there is an obvious gap in the call classification performance between using clean text input (*i.e.* the transcribed speech sentence) and speech recognizer decoded sentence, and this gap could vary between absolute 3-7% in terms of call classification accuracy based on our previous experience. More and more peoples are focusing their attentions on bridging this gap (e.g. [Goel et al 2005] [Saraclar et al 2005]). In this paper we will report some of our initial efforts in this area.

Most of the conventional systems use the single best hypothesis from the speech recognizer as input to the call classifier, which is obviously suboptimal due to recognition errors. An intuitively improvement is to use a word lattice, sausage, or N-best list generated from the speech recognizer as input to the call classifier. Another general approach is to optimize the call classification model under "end-to-end" system performance criteria (*e.g.* [Goel et al 2005] and [Saraclar et al 2005]). These approaches achieved some performance gain, but so far their improvements are still below the expectation.

In addition to the performance degradation, there is another challenge in building a call classification model, in the sense that it requires a huge amount of human effort to process the data. So another primary goal of our algorithms described in this paper is to explore some automatic procedures of processing field data for rapidly building of a call classification model.

In this paper we propose two new approaches to improve the "end-to-end" performance of large scale natural language call routing system. One of them first selects a set of N-best decoding results based on their distance to the transcription, then it augments the training data with the selected N-best list and the decoding result to create a new, larger training data set. A vector distance based data confliction analysis is then performed on this newly generated data set to construct a confusion matrix and class mapping matrix. Based on these matrices we can automatically split the across classes neighboring data with its original classes into different training sets and retag data based on a similarity measure. We then apply Maximum Entropy (ME) training on each training set to generate a series of call classification models. These classification models have been evaluated by text input and speech input respectively, and a classifier score based voting strategy are employed. The second approach we proposed is to incorporate the acoustical confusion information into the call classification process. This approach first rank all the words according to their acoustical confusable measures (based on a heuristic measure), then it shrinks the testing and/or training sentences and keep only those words which are not acoustical confusable. New call classification models can be trained based on the shrunk training sentences.

In section 2 we describe how to select and use N-best list from a speech recognizer. In section 3 we explore how to split across classes neighboring data as well as retagging a cluster based on confusion matrix and class mapping matrix. In section 4 we discuss in details our approach of incorporating acoustic confusion information into call classification. In section 5 we present our experiment with some encouraging results. Finally in section 6 we conclude this work with some further discussion.

2. N-BEST HYPOTHESES SELECTION

The N-best list of the speech recognition output provides a good description of speech recognition characters and can be used to predict the decoding output distortions imposed by the speech recognizer for the future testing sentence. As our discussions are all based on the resources provided by IBM WebSphere Voice Server (WVS), the algorithm described in this paper can be easily implemented into enterprise voice activated applications.

WVS engine generates N-best hypotheses based on the segments (*i.e.* part of the sentence) of a sentence, so the total number of N-best hypotheses in the sentence level is very large. It is not a wise decision to blindly use all these N-best hypotheses and we

need to make selection. We choose not to select the N-best hypotheses based on their decoding scores (*i.e.* acoustic and language model scores), as those acoustically (or linguistically) different decoding results may be very similar to each other in the call classification model. We want to select the decoded sentences which are difference from each other in the call classification model, and we choose to use the Cosine distance between each decoded sentence A_i and the transcription C as the criterion of ranking as illustrated in Equation (2.1):

$$D_{A_i,C} = \sum_{k=1}^{V} \frac{a_{i,k}}{|A_i|} \cdot \frac{c_k}{|C|} \qquad (2.1)$$

Where A_i and C are the bag of words representations of the corresponding decoding result and the transcription. V is the total number of words in the vocabulary, and $a_{i,k}$ and c_k are the indicators of whether word k exists in the sentence A_i or C. For each sentence in the training set, we select the top N decoding results (in addition to the single best hypothesis) with the minimum cosine distance to the corresponding transcription. We add these N decoding results and the single best hypothesis to each transcription to generate a new, augmented training set for the call classifier.

3. AUTOMATIC DATA SPLITTING AND RETAGGING

3.1 Data Splitting

In order to continuously reduce the performance gap and manual effort required for data tagging we consider to introduce a confusion matrix measure and class mapping matrix. Unlike conventional boosting technique which aims at combining some weak classifiers to get a strong classifier, our goal here is to statistically identify the neighboring data sets and split each conflicting data pair with its original class, which is trapped into confusion matrix measure and easily causes ambiguity for involved classes, into different models and even to automatically re-classify the identified clusters based on the class mapping matrix.

The training set now becomes bigger after including N-best hypotheses data described in Section 2, and we consider to proceed the confusion measure to entire training set.

Assume: current training set can be described as

$$(y_1, x_1) \ldots\ldots (y_j, x_j) \ldots\ldots (y_n, x_n) \subset Y \cdot X \qquad (3.1)$$

Where Y is a set of classifications and X is a set of pattern sentences.

The classification problem can be considered: to classify an unseen input sentence x_j, one takes into account a notion of similarity between already classified y_j and X, so the similarity measure can be formalized as

$$X \cdot X' \rightarrow R \quad (x, x_i') \rightarrow k(x, x_i') \qquad (3.2)$$

One of the simplest kernels k( ) is dot product

$$\langle x, x' \rangle = \sum_{i=1}^{M} x_i x_i' \qquad (3.3)$$

Assume: there are N classes in training set, for each class to cluster all the sentences by using dot product.

For a cluster T_j $0 < j < k$ in Class W_i $0 < i < N$, it constructs a central vector V_j which mathematically represents cluster T_j.

For each V_j of a cluster T_j in $0 < j < k$ in Class W_j $0 < j < N$, to compute the similarity distance D_{jh} again every central vector V_h of a cluster T_h $0 < h < k$ in a class W_h $0 < h < N$, $h \neq j$;

If $D_{jh} < \lambda$ (pre-set threshold) Cluster T_j and T_h are identified as neighboring data pair, and D_{jh} will be registered in confusion matrix at node n_{jh}.

Since the cluster T_j is the cross class neighboring data pair of a cluster T_h it will be split from its original class W_j and moved into a split new training set $S_{compete}$ where the cluster T_h will be also moved in based on the confusion matrix, be aware that the class names for both clusters T_j and T_h will be determined by the class mapping matrix. Based on the confusion matrix, if a cluster T_l in class W_l doesn't get the corresponding node in confusion matrix set, it will be moved to a separated new training set $S_{non-compete}$ but it's class name will depend on the class mapping matrix.

3.2 Automatic re-tagging of a cluster

A central vector V_j represents the meaning of the cluster T_j $0 < j < k$ in terms of statistics matter under Class W_j $0 < j < N$. By applying the same rule as to a cluster each class W_j $0 < j < N$. we construct a central vector VW_j which statistically represents the meaning of its class.

For each V_j of a cluster T_j in $0 < j < k$ under Class W_j $0 < j < N$, we compute the similarity distance DW_{jh} again each VW_h in a class W_h $0 < h < N$. We will register a new class name W_h for T_j in a class mapping matrix, if $DW_{jh} <$

DW_{jj}, where DW_{jj} is the similarity distance between a cluster T_j and its original tagged class W_j. Based on this class mapping matrix no matter how the cluster T_j will be moved to $S_{compete}$ or $S_{NONcompete}$ it will be reclassified under class W_h instead of original class W_j.

3.3 Voting Strategy

After data split processing now there are two more new training sets , $S_{compete}$ and $S_{non-compete}$. It's designed to have the third training set $S_{combined}$ by combining $S_{compete}$ and $S_{non-compete}$. By applying Maximum Entropy training to these three training sets individually there are total three classification models created $M_{combined}$, $M_{compete}$, and $M_{non-compete}$. We analyzed several kinds of voting strategies, and finally the confidence score of a classifier based method comes out and is used as voting strategy for the final system. This voting strategy is to trust the output of the model with best performance unless the majority voting generates a different output, and the average score of majority voting is greater than the score of that individual model.

4. INCORPORATING ACOUSTIC CONFUSABLE INFORMATION INTO CALL CLASSIFICATION

Speech recognizers make mistakes in the decoding result, and these mistakes will degrade the call classification accuracy. But instead of making random errors, speech recognizers can accurately predict certain words and phrases, and this might help us in reducing the gap between the call classification accuracy of using transcribed sentences and the decoding results.

Among all the words that will be used to build the call classification model, some are different from the other in the call classification perspective, meaning they provide different information or confidence about different call types, but they are also different from each other in the acoustic point of view, meaning some of them can be easily detected by the speech recognizer, while some of them are more likely to be confused with something else. If we can put more emphasis on those acoustically easy detecting words in developing the call classification model, we might be able to reduce the gap between the call classification using the decoded speech and the transcript.

The approach of building call classification model using acoustically easy detecting words consists of two parts: finding the acoustical confusable words, and incorporating acoustical confusable information into call classification model. To detect those acoustical confusable words, we first generate an N-best list for each transcribed speech sentences *S*, then for each word W in the transcript of sentence *S*, we define a confusable measure $T_{w,s}$ as the following:

$$T_{w,s} = \frac{\sum_{i=1}^{M} \delta_{w,i}}{LogM} \tag{4.1}$$

where *M* is the total number of N-best list generated for the sentence S^1, $\delta_{w,i}$ is a delta function that indicates whether the word *W* appears in the *ith* decoding result of the N best list. $\delta_{w,i}$ is 1 if word *W* appears in the ith N-best list, and 0 otherwise. As indicated by Equation 4.1, the T measure of a word W will be higher if it appears more frequently in the N-best list or if there is less total number of N-best listed generated for the sentence *S*. We then take the average of the $T_{w,s}$ measure for each word *W* across all the training set sentences *S* as in Equation 4.2, and ranked the words according to their T_w measures. We then assume that the lower T_w measure of a word *W,* the more confusable of our speech recognizer in detecting the word *W*.

$$T = \underset{s}{Average}\{T_{w,s}\} \tag{4.2}$$

Once we rank the words according to their acoustical confusable measure, we then incorporate their confusable information into the testing and training of our call classifier. We use two different approaches. One is to train the call classifier using every possible word in the vocabulary, but only to use those words that are acoustically un-confusable in the testing sentence (decoding result) in call classification process. The other is to modify both the training sentences and testing sentences, keep only those words that are acoustically un-confusable, train the call classifier and test it.

5. EXPERIMENTAL RESULTS

As mentioned before the goal is to look for an automatic approach of data modeling, our experiments were carried out on a live field data collected from an enterprise call center application. The original call classification labels of the training and testing data were automatically made by a Wizard of Oz system with poor quality, and none of the 30,000 sentence training set nor 5000 sentence testing set have been checked by human.

Our first experiment was to train one single call classifier using the augmented new training set (*i.e.* the transcription + single best hypothesis + N-best lists based on the cosine distance). Table 1 shows the results (in terms of Call routing Error Rate(CER)) of using different numbers of N-best hypotheses based on the cosine distance , where CTT means the clean training text data that is transcribed by human, RTO represents the single best hypothesis output from speech recognizer, and N=x of CM are the top x N-best hypotheses determined by the cosine distance measurement as described in Equation (2.1).

[1] The speech recognition engine we used will generate different number of N-best lists for the different sentences.

Training set & components	Relative CER % (from speech input)
CTT	28.54
CTT+RTO	31.20
CTT+RTO+N=10 of CM	28.98
CTT+RTO+N=5 of CM	27.58
CTT+RTO+N=3 of CM	26.58
CTT+RTO+N=2 of CM	27.40

Table 1: The call classification performance of training one single classifier using the augmented new training set. N is the number of n-best lists that were incorporated in the new training set. (e.g. N=5 means to use top 5 N-best hypotheses of speech recognition outputs).

It's clear from Table 1 that adding N-best data leads to a small but promising gain for the end-to-end performance of call routing system. It also proves that speech recognition errors can be modeled and complemented by N-best hypotheses selected by cosine distance measure.

The result for using clean training text (CTT) model against clean testing set input is 24.94%, so there is a performance gap between speech input and text input to an action classifier, Table 2 showed the data on these gaps. The results clearly reveal that cosine distance measure based N-best hypotheses complement reduced the performance gap in classification accuracy.

	Before N-best complement	After N-best complement
Text Input	24.94%	24.98%
Speech Input	28.54%	26.58%
Performance Gap	Absolute 3.6%	Absolute 1.6%

Table 2 Performance gap in CER between speech and text input

In section 3 we proposed a new approach to reduce the ambiguity and perform automatically data retagging based similarity measure. After training data splitting and retagging as discussed in section 3, we generate three new training sets as $S_{compete}$, which contains all the data clusters that are positive to confusion matrix, $S_{non-compete}$, which is the union of all the data clusters that are negative to confusion matrix, and $S_{combined} = S_{compete} + S_{non-compete}$, which is the combination of the previous two sets.

Maximum Entropy training is applied to above models, and testing results are showed in Table 3:

	$M_{compete}$	$M_{non-compete}$	$M_{combined}$	M_{voted}
Text Input	29.74%	24.16%	25.62%	24.20%
Speech Input	30.82%	25.8%	26.58%	25.74%

Table 3: Testing results for final combined system

$M_{compete}$, $M_{non-compete}$ and $M_{combined}$ are the models for training sets $S_{compete}$, $S_{non-compete}$, and $S_{combined}$ respectively, and M_{voted} is final model which uses the voting strategy in Section 3 to combine $M_{compete}$, $M_{non-compete}$ and $M_{combined}$ The result is encourage, if comparing between Table 1 and Table3, the best CER 25.74% from the final system outperforms the baseline (28.54%) by a 10% relative for speech input of a live data set. Table 3 also shows that the difference in the CER from speech input (end-to-end) and the CER from text input (human transcribed sentences), has been reduced from the previous 3.6% absolute difference (as in Table 2) into current 1.54% absolute as depicted in the Table 3. We believe one of reasons is that the N-best hypotheses added in provides the tolerance to speech recognition errors; the others reasons could be the reduced ambiguity and improved classification consistency resulted from data splitting and retagging.

In our experiments of incorporating the acoustic confusion information into call classification, we first ranked and selected the top *L* words based on their acoustic confusable measure as discussed in the section 4, then we proceed in two different directions, one was to generate the call classification models using the original training data but used the processed testing data (*i.e.* the modified decoding result which keeps only those words that are in the top *L* list), and the second was to modify both the training and testing data based on the top *L* word list, re-train and re-test the call classifier. The specific value of *L* was generated from the validation set, which was 1100 in the first case and around 2000 in the second case. We got similar improvements (around absolute 3%) in both situations.

6. CONCLUSIONS

In this paper we propose some novel approaches that aim at improving the end-to-end performance of automatic call routing system and reduce the manual effort in tagging the speech data. Our proposed approaches improve the call routing accuracy of the live field data with 10% relatively through the data confusion reduction and model tolerance enhancement. They also reduce the human effort in the tagging through an automatic splitting and re-tagging process.

7. REFERENCES

[1] Goel, V., Kuo, H. K. (Jeff), Deligence, S., Wu, C. 2005 "Language model estimation for optimizing End-to-End performance of a Natural language call routing system", *Proc. ICASSP 2005*, Philadelphia, U.S.

[2] Saraclar, M. and Roark, B. 2005 "Joint discriminative language modeling and utterance classification", *Proc. ICASSP 2005,* Philadelphia, U.S.

[3] Kuo, H. K. (Jeff) and Lee, C. H. 2003 "Discriminative Training of Natural Language Call Routers", *IEEE Transactions on speech and audio processing, Vol. 11 No. 1* January, 2003.