# IMPROVING AUTOMATIC CALL CLASSIFICATION USING MACHINE TRANSLATION

*Tanveer A. Faruquie, Nitendra Rajput*

IBM India Research Lab,
IIT, Hauz Khas, New Delhi 110016

*Vimal Raj*

Department of Electrical Engineering
IIT, Chennai, 600036

## ABSTRACT

Utterance classification is an important task in spoken-dialog systems. The response of the system is dependent on category assigned to the speaker's utterance by the classifier. However often the input speech is spontaneous and noisy which results in high word error rates. This results in unsatisfactory system performance. In this paper we describe a method to improve the natural language call classification task using statistical machine translation (SMT). We utilize the translation model in SMT to capture the relation between truth and the ASR transcribed text. The model is trained using the human transcribed text and the ASR transcribed text. During deployment SMT is used to sanitize the ASR transcribed text. Our experiments with IBM model 2 shows significant improvement in call classification accuracy.

***Index Terms***— call classification, call routing, ASR, statistical machine translation.

## 1. INTRODUCTION

A lot of real life applications are based on correct classification of user utterance. Call centers are one such example. The first task facing any call center is to direct the user to the appropriate department depending on his request. Natural language call classification systems [1] mimic the capability of human routing agents to provide natural human like interface for call routing. The main component of the call classification system is the classifier. The classifier is trained using supervised learning on a set of labeled data. Labeling is done manually on a human transcribed text of the actual audio recordings of the calls. It assigns labels from the set of possible classes to which the calls can be classified. Additionally automatic speech recognizer (ASR) transcribed text can be used instead or augmented with the human transcribed text for training. For satisfactory performance, such systems need to have a high level of classification accuracy. This in turn needs an accurate ASR which should transcribe the audio to as close as possible to the human transcribed text irrespective of factors like ambient noise, the accent of the caller or the calling medium. However ASRs often produce recognition errors. The mis-recognized text when fed to the classifier results in poor classification accuracy even if the classifier is of high accuracy.

Several methods [2] have tried to improve the classifier to make it more robust to recognition errors. The methods include boosting, discriminative training and constrained minimization. The most successful approach among these is boosting [3]. Instead of using a single classifier a combination of classifiers are also used. Niyogi et. al. [4] uses three classifiers; the decision is made by the first two classifiers if they agree, and arbitrated by the third when they disagree. The third classifier may be explicitly trained on disagreements of the first two using minimum error training and can make a choice only on a subset of topics.

Instead of improving the classifier several methods have been proposed which either try to sanitize the ASR output before feeding it to the classifier or provide some additional information to the classifier to aid classification. Cheng et. al. [5] reduce the impact of speech recognition errors on call classification by selecting a list of ASR transcribed text for training the classifier. They select the list of ASR transcribed text by looking at the distance of the generated N-best sentences from the human transcribed text. Paulik et. al. [6] uses machine translation techniques to improve target language ASR performance by using the source language resources. Matula et. al. [7] use the confidence scores generated with the N-best list for improving call classification. The query vector is weighed using ASR confidence scores for each unigram and bigram. Hence words with high confidence scores influence the final selection more than words with low confidence scores. Relevance Feedback [8] technique tries to sanitize the input of the classifier by aiding the user to reformulate their queries so that the reformulated query results in better classification.

It is seen that the best performance obtained by a call classification system is when the system is trained on human transcribed text and tested on human transcribed text. However obtaining a human transcribed text at deployment time is not possible. Here we propose a method to improve the natural language call classification task using statistical machine translation (SMT). The ASR transcribed text is sanitized before feeding the classifier. The sanitization process is thought of as a translation process in which the

source language is the erroneous ASR output and the target language is the clean human transcribed text. At training time the SMT system is trained using the human transcribed text and N-best list of the ASR transcribed text. At deployment time the SMT is used to sanitize the N-best list by translating the ASR N-best list to estimate the possible human transcription of the input utterance.

In next section we describe the Statistical machine translation and the IBM Translation Model 2. Next we present an architectural overview of our system. In section 4 we present some experimental results on the automatic call routing task.

## 2. TRANSLATION MODEL

In order to clean the ASR transcribed text we model the cleaning process as a translation process. We use the IBM statistical models for our task. The IBM Statistical Translation models are based on the source-channel paradigm of communication theory. Consider the problem of translating a noisy sentence *n* (source language) to a clean sentence *c* (target language). We imagine that the originally clean utterance *c* when transmitted over the noisy communication channel gets corrupted by ASR errors and become a noisy sentence *n*. The goal is to estimate the original clean utterance from the noisy sentence generated by the channel by modeling the noise characteristics of the channel mathematically and determining the parameters of the model experimentally. This can be expressed as

$$\hat{c} = \arg\max_{c} \Pr(c/n)$$

By Bayes' Theorem

$$\hat{c} = \arg\max_{c} \Pr(n/c)\Pr(c)$$

The above equation is known as the *Fundamental Equation of Statistical Machine Translation*. The computation tasks in a SMT are therefore

- Estimating the translation model probability $\Pr(n/c)$
- Estimating the language model probability $\Pr(c)$
- Searching for the utterance c that maximizes the product $\Pr(n/c)\Pr(c)$

### 2.1. IBM Translation model 2

The conditional distribution $\Pr(n/c)$ is expressed in terms of a set of parameters and these parameters are estimated at training time. The input to the training process is a corpus of aligned bilingual sentences and the training process is essentially an iterative application of the EM algorithm. Brown et al. [9] proposed a series of five translation models of increasing complexity and provided algorithms for estimating the parameters of the models. We have used IBM model 2 as the model to learn the relationship between the clean utterances and the ASR transcribed text.

IBM Model 2 is a generative model and it works as follows

- For a clean sentence *c* of length *l*, choose the length *m* of the noisy sentence from distribution $\varepsilon(m/l)$.
- For each position *j = 1, 2, ..., m* in the noisy sentence choose a position $a_j$ in the clean sentence from a distribution $a(a_j / j,l,m)$. This distribution tells which word of *c* is associated with which word of *n*.
- For each word at *j = 1, 2, ...., m* in the noisy utterance choose a word $c_j$ from the manual transcription according to the distribution $t(c_j / n_{a_j})$.

The probability of generating a clean sentence $c = c_1 c_2 c_3 ....c_m$ given a noisy input $n = n_1 n_2 n_3 ...n_l$ is given by

$$\Pr(n/c) = \varepsilon(m/l)\prod_{j=1}^{m}\sum_{i=0}^{l} t(c_j / n_i).a(i/j,m,l)$$

IBM Model 1 is a special case of Model 2 in which a uniform distribution is assumed for $a(a_j / j,l,m)$ and is usually kept fixed at $(l+1)^{-1}$.

## 3. SYSTEM DESCRIPTION

As shown in Figure 1 we used the statistical machine translation to capture the relationship between the N-best sentences and the human transcribed sentences. At training time we have a set of human transcribed truth sentences, $S_T$, and the corresponding N-best sentences, $S_N$, transcribed by the ASR. We train a model that probabilistically determines the relation of a truth sentence with the N-best sentence.

The classifier is usually trained using supervised learning on the labeled data set; $S_T$. Labeling is done manually on the human transcribed text of the actual audio recordings of the calls. Additionally automatic speech recognizer (ASR) transcribed text, $S_N$, can be augmented with or used in place of the human transcribed text. The transcribed text along with the class labels are used to train the classifier. When the system is actually deployed to classify calls of live users, no human transcribed text is available. An automatic speech recognizer (ASR) is used to transcribe the text from user's speech. This automatically

transcribed text is used as input to the classifier to classify the calls. Poor recognition accuracy of the ASR systems generates a noisy transcribed text and is one of the major problem areas to be dealt with in automatic call classification systems.

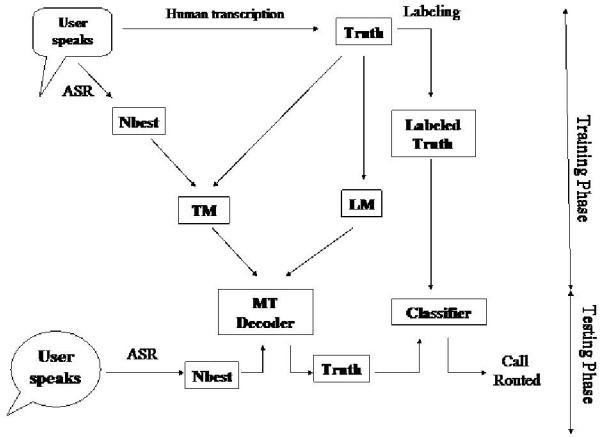


*Figure 1: The SMT based classification system*

During training, the N-best sentences, $S_N$, of ASR transcribed text along with the corresponding human transcribed text, $S_T$, form the parallel corpora. Each N-best hypothesis for every utterance used in training is a source sentence and the corresponding human transcribed text forms the target sentence, i.e. ASR generated sentences belong to the source language and the manually transcribed sentence belong to the target language. This parallel corpus is used to train the SMT system. This includes the translation model and the language model. The human transcribed sentences, optionally augmented with N-best sentences, are used for training the classifier as well.

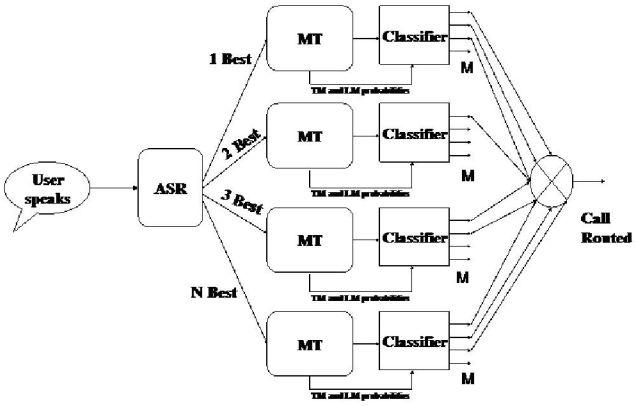


*Figure 2: Call classification using SMT*

As shown in figure 2, at deployment time only the ASR transcribed text is available. We essentially get a list of top N best ASR transcribed sentences for a given input utterance. Each of the N ASR transcribed text is fed to the machine translator to predict the human transcribed text. The classifier gives the scores for top M possible classes in which the utterance can be classified. Hence for N sentences we have N x M scores available. For our experiments we choose the top class of the 1 best in the N-best sentences. Alternatively one can use the associated confidence scores of the N-best sentences and the class scores of the classifier to make a decision. Apart from using the N-best and class confidence scores one can weight the choices using the LM and TM probabilities given by the SMT system. Those with high LM and TM probabilities get higher weights than those with low probabilities.

## 4. EXPERIMENTAL EVALUATION

Our experiments were carried out on live data collected from an enterprise call centre application. The data was collected for two tasks. There were 7636 training samples for task one and 7848 training samples for task two. There were 1300 training samples for each task used for testing purpose. Both the testing and the training samples were manually transcribed and labeled by the agents of the call centre. Additionally audio of each utterance was transcribed by the ASR and the N-best sentences were stored. For our experimentations N was set to four. The number of target classes for task one is 36 and the number of target classes for task two is 28.

For each utterance the manually transcribed text and the N-best list produced by ASR form a set of N sentence parallel corpora. Hence for task one and task two we have a total of 30544 and 31392 sentence pair parallel corpora respectively. This parallel corpus is used to train an IBM translation model 2. The manually transcribed corpus is also used to train the language model for the machine translation. We used an alternating optimization decoder for translating the N best sentence generated by the ASR system.

The classifier used for our experiments is a TF-IDF based vector classifier. The classifier was trained on the manually transcribed corpus for each of the tasks separately. The testing was done on the manually transcribed text, the 1 best ASR transcribed text and the output of the machine translated text. For comparison purpose the classifier was also trained on the 1 best and N best ASR transcribed text and was tested on the manually transcribed sentence, 1 best sentence and the N best sentences.

The quality of the audio input was quite poor and the ASR word error rates obtained for task one and task two were 28% and 21% respectively. Table 1 shows the classification accuracy for task 1 with 36 classes and table 2 shows the classification accuracy for task 2 with 28 classes. It can be seen from the table 1 and table 2 that manual training vs manual testing gives the best performance. Unfortunately manual transcription cannot be obtained at

runtime and this accuracy only serves as a benchmark to test other configurations.

*Table 1: Classification accuracy for task 1*

| Training | Testing | Classification Accuracy |
|---|---|---|
| Manual | Manual | 77.2% |
| 1 best | 1 best | 55.4% |
| N best | N best | 57% |
| Manual | SMT | 65.1% |

*Table 2: Classification accuracy for task 2*

| Training | Testing | Classification Accuracy |
|---|---|---|
| Manual | Manual | 80.2% |
| 1 best | 1 best | 52.9% |
| N best | N best | 54.3% |
| Manual | SMT | 62.1% |

We get an improvement of 8.1% and 7.8% for task one and task two respectively over N-best vs N-best performance. However the SMT based method still has to catch up with the performance of the manual vs manual classification accuracy.

*Table 3: Example of SMT corrected sentences*

| Truth | N Best | MT output |
|---|---|---|
| I forgot my email password | I forgot my attic password | I forgot my email password |
| | I forgot my attic password | I forgot my email password |
| | I forgot my attic password | I forgot my email password |
| | I forgot my headache password | I forgot my email password |
| No | Well | well |
| | Will | No |
| | Mill | No |
| I'm having problems logging in | I'm having problems audience | I'm having problems logging |
| | I have problems audience | I have problems login |
| | 9 having problems audience | I having problems login |
| | I'd having problems audience | I'd having problems login |

Table 3 shows few examples of the manually transcribed sentence, the ASR generated N-best sentences and the sentences produced by the SMT system. It can be seen that in most cases the SMT is able to sanitize the sentence produced by the ASR system quite satisfactorily.

## 5. CONCLUSION

In this paper we present a method for improving the call classification accuracy using the statistical machine translation system. The human transcribed text and the corresponding ASR transcribed text form a parallel corpus which is used to train the SMT system. It was shown that using SMT translated sentences as input instead of N-best sentence significantly improves the call classification accuracy. This method can be easily extended for different type of classifiers. In future we plan to incorporate the SMT translation scores along with the N-best scores to further improve the classification accuracy. The current translation model 2 uses a generalized alignment model however the alignment between the human transcribed sentence and the ASR transcribed sentence is monotonic. Imposing monotonicity constraints on the translation model can further improve the translation accuracy and hence the classification accuracy.

## 6. REFERENCES


[1] Chu-Carroll, and R. Carpenter, "Vector based natural language call routing," *Computation Linguistics*, 25(3), pp-361-388, 1999.

[2] Zitouni, I., H.K.J. Kuo, and C.H. Lee, "Natural Language call routing: towards combination and boosting of classifiers", *Proc. of IEEE ASRU*, 2001.

[3] R.E., Schapire, M. Rochery, M., Rahim, and N. Gupta, "Boosting with Prior Knowledge for Call Classification", *IEEE Transaction on Speech and Audio Processing, Vol 3, No 2*, 2005.

[4] P. Niyogi, J. B., Pierrot, and O. Siohan, "Multiple classifiers by constrained minimization", *Proc. of ICASSP*, 2000.

[5] C. Wu, M. X., Li, H. K. J., Kuo, E. E., Jan, V., Goel and D. Lubensky, "Improving end-to-end performance of call classification through data confusion reduction and model tolerance enhancement ", *Proc of INTERSPEECH*, 2005.

[6] M. Paulik, C. Fugen, S. Stuker, T. Schultz, T. Schaaf, and A. Waibel, "Document Driven Machine Translation Enhanced ASR", *Proc of INTERSPEECH*, 2005.

[7] V.C., Matula, and N. Tyson, "Improved LSI-Based Natural Language Call Routing Using Speech Confidence Scores", *Proc of EMNLP*, 2004.

[8] H.K.J., Kuo, and C.H., Lee, "Discriminative training in Natural Language Call Routing", *Proc. of ICSLP*, 2000.

[9] P.F., Brown, S. A. Della Pietra, V. J. Della Pietra and R. L., Mercer, "The Mathematics of Statistical Machine Translation: Parameter Estimation", *Computation Linguistics, 19(2), pp 263-311*, June 1993.