A Misclassification Reduction Approach for Automatic Call Routing

Fernando Uceda-Ponga[1], Luis Villaseñor-Pineda[1],
Manuel Montes-y-Gómez[1], Alejandro Barbosa[2]

[1]Laboratorio de Tecnologías del Lenguaje, INAOE, México.
{fuceda, villasen, mmontesg}@inaoep.mx
[2]Nuance Technologies, Mexico.
Alejandro.Barbosa@nuance.com

Abstract. Automatic call routing is one of the most important issues in the call center domain. It can be modeled –once performed the speech recognition of utterances– as a text classification task. Nevertheless, in this case, texts are extremely small (just a few words) and there are a great number of narrow call-type classes. In this paper, we propose a text classification method specially suited to work on this scenario. This method considers a new weighting scheme of terms and uses a multiple stage classification approach with the aim of balance the rate of rejected calls (directed to a human operator) and the classification accuracy. The proposed method was evaluated on a Spanish corpus consisting of 24,638 call utterances achieving outstanding results: 95.5% of classification accuracy with a rejection rate of just 8.2%.

1 Introduction

Call routing (CR) is one of the most important issues in the call center domain. It can be defined as the process of associating user requests with their desire destinations [1]. In other words, it involves forwarding incoming phone calls to someone qualified to handle them (e.g., to the most appropriate customer representative).

Traditionally, this task has been performed using a touch-tone approach, which allows users to navigate through different options by a hierarchical menu. In spite of its conceptual simplicity, this approach has several disadvantages, such as the time the users need for listening all options as well as the difficulty for matching existing options with their specific needs. These inconveniences are evident even for simple requests such as "I want to know my account balance", which may require users to navigate as many as four or five nested menus with four or five options each [2].

The inconveniences of the tone-touch approach leaded to the emergence of *automatic call routing systems*, which allow users to interact in natural spoken language [1], and therefore, help companies to significantly reduce their telephone time costs. Typically, this kind of systems considers two main steps. In the first one, phone calls are transcribed to text by means of a speech recognition process. Then, in a second step, text transcriptions are used to predict the correct destination for the phone calls.

In particular, this paper focuses on the second step, which, in fact, can be considered as a *text classification* problem.

Text classification, the assignment of free text documents to one or more predefined categories based on their content, is a widely studied problem that has recently achieved significant advances [3]. Nevertheless, it is important to point out that automatic call routing is a more difficult problem than ordinary text classification, since phone call transcriptions are very short and noisy texts. That is, they are texts consisting of just a few words and containing many errors such as word insertions, deletions and substitutions.

In addition, automatic call routing differs from conventional text classification in that, in the former, there is always the possibility of appealing for human assistance. In particular, call routing systems tend to *reject* all ambiguous or incomprehensible transcriptions, transferring their resolution to a human operator. Evidently, this circumstance is of great relevance, since having a high rejection rate implies a high telephone time cost, whereas having a low classification accuracy (due to the misclassification of unclear transcriptions) causes a strong discomfort among users. Therefore, one of the main research topics on automatic call routing is to provide a good balance between these two factors, the classification accuracy and the rejection rate.

In this paper, we propose a classification method specially suited to the task of automatic call routing. On the one hand, this method considers a new *weighting scheme* that is especially appropriate for short texts. On the other hand, it uses a *two-step classification approach* that allows achieving an adequate balance between the rejection rate and the classification accuracy. In a first step, this approach guarantees high classification accuracy by tolerating a high rejection rate. Subsequently, in a second step, it reclassify rejected transcriptions with the intention of recovering some instances and, therefore, reducing the final rejection rate.

Experimental results on a Spanish corpus consisting of 24,638 call utterances are encouraging; the proposed method could correctly classified 95.5% of the utterances with a rejection rate of just 8.2%, outperforming the application of other text classification approaches.

The rest of the paper is organized as follows. Section 2 describes some previous work on automatic call routing. Section 3 describes the proposed method for automatic call routing. Section 4 presents the evaluation results. Finally, Section 5 gives our conclusions and describes some ideas for future work.

2 Related work

There exist different approaches for automatic call routing; among them we can mention the following three ones:

1. A straightforward approach based on the use of *keywords* for triggering different destinations [4].
2. An approach based on the use of *language models* that describe the main characteristics (words sequences) of each destination [5].
3. An approach based on the application of traditional *text classification* methods [2,6].

From these approaches, the first one is the simplest to understand and implement; nevertheless, it is very sensible to transcription errors. The second approach is more robust than the former, but it requires more "training data" as well as clearly differentiable destination vocabularies. Finally, the third approach is less sensible to transcription errors, but it is damaged by the short length of transcriptions as well as by the great number of narrow destinations (categories).

Different works have evaluated the effectiveness of conventional text classification methods for automatic call routing. For instance, [7] shows a comparison of several learning algorithms on a corpus of 4000 manual transcriptions corresponding to nine –balanced– categories. It reports a result of 81.6% of accuracy using a simple recurrent network. In the same way, [8] reports an accuracy of 75.9%; nevertheless, in this case they used a small corpus of 743 automatic transcriptions from six different categories. Both works do not report the rejection rate.

More recently, some works have proposed different adaptations to the traditional text classification approach. These adaptations consider the use of different utterance characterizations, weighting schemes, feature selection methods and learning strategies. For instance, [9] presents a learning approach that uses a cascade of binary classifiers (of Support Vector Machines) with the aim of reducing the classification errors. In this approach, those instances that could not be classified (i.e., that pass through all classifiers) are rejected. Reported results on a set of automatic transcriptions from 19 categories are very relevant; on the one hand, it reports a best accuracy of 99% corresponding to a rejection rate of 75%, and on the other hand, a best rejection rate of 15% for a classification accuracy of 95%.

Similar to these works, our automatic call routing method is also based on a supervised learning approach. Its main difference relays on the use of a novel two-step classification approach that allows maintaining a good balance between the accuracy and rejection rates. In addition, it applies a weighting scheme that is less sensible to the short length of texts. The following section introduces the proposed method.

3 Proposed Method

As we previously mentioned, our method includes two major modifications to the traditional text classification approach. On the one hand, it uses a new term weighting scheme that allows improving the discrimination among narrow classes formed by very short texts. On the other hand, it considers a two-step classification approach that helps maintaining an adequate balance between the accuracy and rejection rates. In the following subsections, we describe in detail these two modifications.

3.1 Term Weighting Scheme

Typically, term relevance is determined using the well-known *tf-idf* weighting scheme. This scheme, however, presents some drawbacks for its application to automatic call routing. First, given the short length of texts, the *tf*-value is quite similar

for most of the terms from a category[1]. Second, given the presence of several related narrow categories, the *idf*-value is also very similar for most terms. Therefore, this weighting scheme does not allow achieving a good discrimination among categories.

Based on the previous observations, we propose a weighting scheme that computes the weight of terms in relation to their occurrence in the whole categories instead than in individual utterances. This scheme considers that the probability of occurrence of a term across distinct categories is different. In particular, we compute the weight of a term t in a category $c_i \in C$ as follows:

$$w_t^{c_i} = \frac{p_t^{c_i}}{\max_{\forall k \in c_i}\left(p_k^{c_i}\right)} \qquad (1)$$

$$p_t^{c_i} = P(c_i \mid t) = \frac{f_t^{c_i}}{\sum_{\forall c_k \in C} f_t^{c_k}}$$

where C is the set of categories, $f_t^{c_i}$ is the frequency of occurrence of the term t in the category c_i, and $p_t^{c_i}$ indicates the conditional probability $P(c_i|t)$, that is, the probability of having the category c_i given the presence of the term t.

It is interesting to notice that the final weight of a term in a category is obtained by applying a kind of normalization with respect to the probabilities of other terms from the same category. Roughly speaking, this normalization allows giving greater weight to terms clearly related to the category and reducing the value of terms uniformly distributed among several classes.

3.2 Two-Step Classification Approach

Figure 1 shows the general architecture of the proposed classification approach, which consists of two main steps: a high-precision classification step and a misclassification reduction step. The main purpose of this new approach is to achieve a *good balance between the classification accuracy and the rejection rate*.

The first step achieves an initial classification of the input utterances (i.e., the phone call transcriptions). This classification considers all categories including an "unknown category" that gathers all ambiguous and incomprehensible transcriptions. This category functions as a rejection class, since its elements are transferred to a human operator for their resolution. This way, the purpose of this step is to guarantee a high classification accuracy by tolerating a high rejection rate. In other words, in this step only the most confident utterances are sent to content categories, whereas the rest of them are classified as unknown.

Subsequently, in a second step, the idea is to reclassify the set of rejected transcriptions with the intention of recovering some utterances and, therefore, reducing the final rejection rate. This step combines two additional classifiers. One of them considers only two categories (unknown and the rest), and it is especially suited to distinguish clear ambiguous and incomprehensible transcriptions. The other one

[1] This is because users tend to use a simple and direct language (more restricted vocabulary) when they interact with a machine (an automatic system).

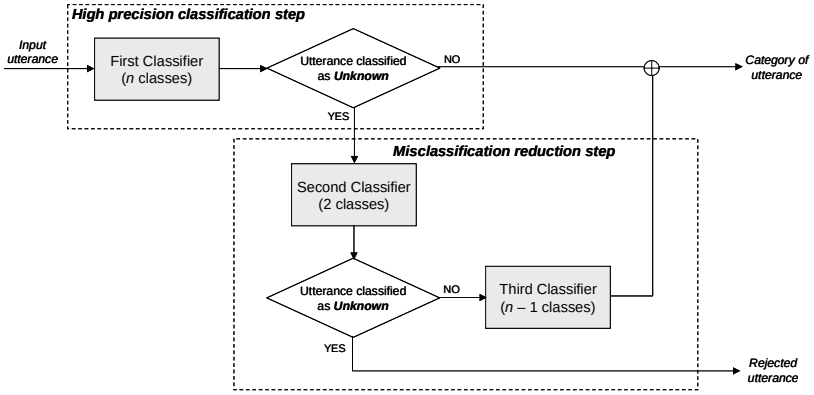

Figure 1. General architecture of the proposed method

considers all categories except unknown and its purpose is to reallocate rescue utterances within content categories.

It is important to point out that the weight of terms (defined in accordance to formula 1) varies from one classifier to another because they consider different sets of categories. This characteristic of our approach is very relevant, since it allows giving the terms a convenient (highly discriminative) weight at each step.

4 Experimental Evaluation

4.1 Train and Test Corpora

In order to evaluate the proposed method we used a corpus consisting of 24,638 Spanish automatic transcriptions[2]. These transcriptions came from a company that provides TV, internet and telephone services, and were manually classified in 24 different categories. The first 23 categories correspond to content categories (i.e., to possible destinations for user requests), whereas the last one is an "unknown category", which gathers all utterances that could not be classified and that must be rejected.

For evaluation purposes, we divided the corpus in two non-overlapped subsets: a train set formed by 80% of the instances from each category (in total, 19,694 utterances), and a test set including the remaining 20% of the instances (4,944 utterances). Figure 2 shows the distribution of instances per category; as it can be observed, this is a very imbalanced corpus.

[2] It was assembled by Nuance technologies (*www.nuance.com*) using the Wizard of Oz technique.

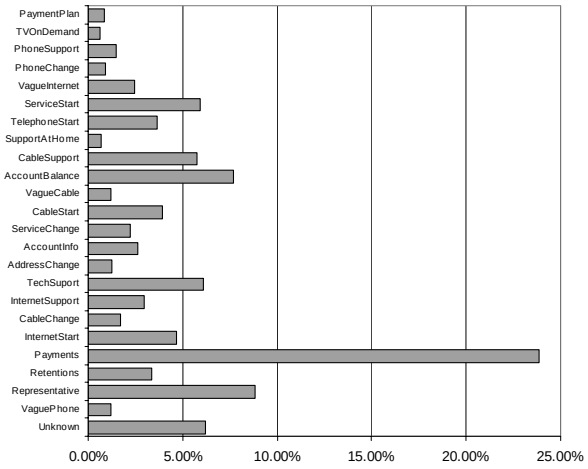

Figure 2. Percentage of instances per category in the used corpus

We decided using this evaluation scheme because we wished to simulate a real situation, where it is possible to find utterances containing unspecified words for the classifier. In particular, the vocabularies from the train and test sets showed important differences: 13.5% of the words from the test instances were not included in the train set.

4.2 Results

In order to obtain a general vision from the proposed method we performed several experiments. Initially, we studied the usefulness of the weighting scheme for tuning the accuracy and rejection rates. Then, we evaluated the effectiveness of our two-step classification approach by comparing its results against others from conventional one-step classifiers.

Tuning the accuracy and rejection rates

For this first experiment, we used the weight of terms as the main criterion for *feature selection*. In particular, we eliminated terms with weight –in all categories– less than a given specified threshold. Table 1 presents the results corresponding to different thresholds. It mainly shows the number of training features, the classification accuracy[3] as well as the rejection rate[4]. It is important to mention that in all cases we

[3] Defined as the percentage of utterances correctly classified within the set of 23 content categories.

[4] Defined as the percentage of test utterances classified as unknown.

used a single (*one-step*) multi-class Naïve Bayes classifier and applied a Boolean weighting for the test utterances[5].

Table 1. Feature selection using the proposed weighting scheme

Weight threshold	Number of features	Classification accuracy	Rejection rate
0	2474	64.9%	1.0%
0.1	2473	71.6%	0.9%
0.2	2431	87.5%	24.2%
0.3	2290	97.4%	31.5%
0.4	2074	99.5%	40.7%
0.5	1712	99.9%	50.9%

Results from this initial experiment showed that using different thresholds it was possible to obtain different combinations of classification accuracies and rejection rates. As we expected, the use of the most discriminative terms allowed achieving better classification accuracy, but also incremented the rejection rate since several instances could not be represented. These results are of great relevance to our method since it considers three different classifiers having different purposes.

Evaluating the effectiveness of our two-step classification method

The goal of the proposed two-step classification method is to achieve a good balance between the classification accuracy and the rejection rate. In order to evaluate the effectiveness of this method, we compared its results against those from a conventional one-step classifier (refer to Table 1).

Our two-step classification method, as shown in Figure 1, considers the combination of three different classifiers. All of them, as in the previous experiment, were implemented using a Naïve Bayes classifier and Boolean weights for representing test instances. Nevertheless, in this case, each classifier considers a different weight threshold[6]: the first two classifiers used a threshold of 0.3, whereas the last one used a threshold of 0.2.

Table 2. Results from the proposed two-step classification method

Correctly classified Utterances		Classification accuracy	Rejection rate
First step	***Second step***		
3396	938	95.5%	8.2%

Table 2 shows the results achieved by this method. In this case, the accuracy was calculated taking into consideration the correctly classified instances from the first step as well as the correctly rescued instances from the second step. On the other hand, the rejection rate was obtained from the set of instances classified as unknown

[5] In contrast, training instances were represented using the proposed weighting scheme.

[6] These thresholds were empirically determined; for each classifier we evaluated several thresholds and selected the combination that achieved the best overall accuracy.

by the second classifier. As can be observed, the achieved results are encouraging; the proposed method reached an adequate balance between the classification accuracy and rejection rate, better than the results from all previous one-step classifiers (refer to Table 1).

Finally, in order to have more information to judge the effectiveness of the proposed method, we compared its results against those from the AdaBoost algorithm (commonly used in this task [10]). Table 3 shows the results from this comparison. The first row presents the results of our method, and the second and the third rows show the results from AdaBoost using Naïve Bayes as base classifier and appliying six iterations. The only difference from these two runs was the used weighting scheme. The second row reports the result using the proposed weighting scheme, whereas the last row shows the result corresponding to the *tf-idf* weighting scheme.

Table 3. The proposed method against the AdaBoost classifier

Classification scheme	**Classification accuracy**	**Rejection rate**	**Recall of *unknown* class**
Our method	95.5%	8.2%	100%
AdaBoost (using our weighting scheme)	98.9%	29.7%	100%
AdaBoost (using *tf-idf* weights)	78.2%	6.8%	60%

The results from this experiment are very interesting; they show that our method allowed achieving a better balance between the accuracy and rejection rates than the traditional AdaBoost method. On the other hand, they also show that the proposed weighting scheme lead to better results than the *tf-idf* weighting scheme. Moreover, using the *tf-idf* weighting scheme it was possible to obtain a small rejection rate, however, it could not identify (and therefore, transfer to a human operator) all ambiguous or incomprehensible utterances. This last factor is very relevant since the misclassification of unclear transcriptions tend to cause a strong discomfort among users.

5 Conclusions

This paper proposed a new text classification method that is specially suited to the task of automatic call routing. This method differs from previous works in two main concerns: the application of a new term weighting scheme, and the use of a two-step classification approach.

Experimental results on a Spanish corpus consisting of 24,638 call utterances showed, on the one hand, that the proposed weighting scheme is especially appropriate for short texts, and on the other hand, that the two-step classification approach allows maintaining an adequate balance between the classification accuracy and the rejection rate. In other words, these results indicated that the proposed method is pertinent for the automatic call routing task, which commonly considers a great number of narrow categories, imbalanced training data sets as well as very short text instances.

Another important aspect is that the method itself does not rely on language dependent rules, and therefore, it would be easily used in other languages.

Finally, it is important to mention that the method can be adjusted (moving the weight thresholds) for each particular situation, in order to reduce the rejection rate or improve the classification accuracy. In addition, it is also important to comment that the performance of this method greatly depends on the corpus size. It requires of a large train corpora to calculate confident weights for the terms. For this reason, one of the main directions for future work is the application of semi-supervised learning strategies.

Acknowledgements: This work was done under partial support of CONACYT project grant 61335.

References

1. Lee, C.-H., Carpenter, B., Chou, W., Chu-Carroll, J., Reichl, W., Saad, A. & Zhou, Q. *On Natural Languge Call Routing*. Speech Communication, Vol. 31, Issue 4, August 2000, pp. 309-320.
2. Carpenter B. & Chu-Carroll, J. Natural Language Call Routing: A Robust, Self-organizing Approach, Proc. ICSLP 1998.
3. Sebastiani F., *Machine learning in automated text categorization*, ACM Computing Surveys, 34(1):1–47, 2002.
4. Gorin, A.L., Riccardi, G. & Wright, J.H. *How may I help you?* Speech Communication 23, 1997.113–127. Elsevier.
5. Huang Q. & Cox, S.J. *Automatic Call Routing with Multiple Language Models*. Proc. Workshop on Spoken Language Understanding for Conversational Systems, Boston, May 2004.
6. Kuo, H.-K., J., Lee, C.-H., Zitouni, I., Fosler-Lussier, E. & Ammicht, E. *Discriminative training for call classification and routing*. In ICSLP-2002, 1145-1148.
7. Garfield S., Wermter S. & Devlin S., *Spoken Language Classification using Hybrid Classifier Combination.* International Journal of Hybrid Intelligent Systems, Vol. 2, Issue 1, pp 13 – 33, January 2005.
8. Ming Tang, Bryan Pellom, Kadri Hacioglu. *Call-Type Classification And Unsupervised Training For The Call Center Domain*. Automatic Speech Recognition and Understanding IEEE Workshop, 204- 208, ISBN 0-7803-7980-2.ASRU 2003.
9. Haffner, P., Tur, G., Wright, J.H. *Optimizing SVMs for complex call classification.* International Conference on Acoustics, Speech, and Signal Processing. ICASSP '03 IEEE. April 2003 Vol. 1, pp 632- 635, 2003.
10. Schapire, R. E. & Singer, Y. *BoosTexter: A Boosting-based System for Text Categorization*. Machine Learning, 39, pp. 135–168, 2000. Kluwer.