

Mining call center dialog data

A. Gilman¹, B. Narayanan² & S. Paul²

¹*GE Global Research, USA*

²*GE India Technology Center, India*

Abstract

We consider the problem of mining conversations between customers and call center representatives for automatically classifying calls into predefined categories. We analyze the conversations for speaker dependent information content using several multi-class classification technologies. The data consists of 539 manually transcribed conversations belonging to 15 categories. Classifiers were built using Support Vector Machines, Naïve Bayes, Latent Semantic Analysis, Vector Space and K-Nearest Neighbor technologies. SVM classifiers were found to perform consistently well giving an accuracy of about 74% on the entire data and about 92% when considering only the 4 largest classes. It is observed that very high weightage to either the customer part of the dialog or that of the agent results in poor accuracy. Nearly equal weightage to the customer and agent provides the best results consistently. This approach has potential to identify cross sell and up-sell opportunities in real-time.

Keywords: text mining, classification, support vector machines, dialog mining, customer relationship management.

1 Introduction

Customer Relationship Management (CRM) strategy and implementation are essential for a business to retain its most valuable customers and boost overall profitability. Information related to the products, customers and transactions is aggregated in a CRM data warehouse. This data is then analyzed using traditional analytics to build models for customer segmentation, propensity for products, and cross sell opportunities. It is also recognized that a lot of information about customer behavior and propensities may be gleaned from the conversations between call center service representatives and customers. Dialog Mining of these conversations – that may be transcribed to text automatically



using Speech to Text Translators – may hold the key in discovery of this information that can lead to better models and identify additional cross-sell or up-sell opportunities. The technology may also be used for automated quality control at the call center. In this paper, we focus on an essential step in this process – that of automatically categorizing the call. Over the last few years, there has been a heightened interest in textual data mining as a result of the revolution of World Wide Web and rapid growth of digital records. For good surveys on text mining, see Chakrabarti [1], Sebastiani [2].

In recent years, owing to the advancement of automatic speech recognition and human-computer interaction systems, analysis of dialog data also started receiving attention from the text mining community. Researchers have used both the manually transcribed data and automatic speech-to-text translators and evaluated the performance of different classification methods. There has been some attention paid to topic identification for call routing. See Carpenter and Chu-Carroll [3], Chu-Carroll and Carpenter [4], Golden et al. [5], Rochery et al. [6]. Myers et al. [7] has used a corpus of conversational data to test the effect of sub-dialogs for topic detection.

In this paper, we study the problem of automatically categorizing manually transcribed call center dialogs into predefined Classes. We have implemented the Support Vector Machines (SVM), Naïve Bayes, Latent Semantic Analysis (LSA), Vector Space Model, and k-Nearest Neighbor (k -NN) based multi-class classification techniques and compared their effectiveness in categorizing dialog data. We also study whether the Agent part of the conversation can help discrimination better than that of the Customer. An intuition here is that an extreme weight to either the Agent or Customer will result in loss of information. It is intuitively unclear if equal weight will always provide the best results. This paper organized as follows. Section 2 presents the methodology for feature extraction. In section 3, we briefly describe the various classification techniques studied. Section 4 describes the data set and experimental setup. Section 5 presents the results. Conclusions and future work is discussed in section 6.

2 Dialog classification

2.1 Feature representation

Text mining algorithms have commonly used the bag of words concept, see Joachims [8]. Each document is represented as a term frequency vector, a term corresponding to a word. The order of the words is not relevant. We adopt a well known variation of this concept in the representation of dialog. Specifically, a term corresponds to the stem of a word that is not a stopword. The stopword list was assembled semi-automatically to include all commonly occurring words that had little discriminating power. Additionally, we also consider terms that are pairs of consecutively occurring word stems. For the rest of this paper, we will refer to such terms as digrams. Let us assume that we have collection of N training dialogs $\{d_1, \dots, d_N\}$. Let the vocabulary $V = \{W_1, \dots, W_M\}$ be the collection



of all terms in the corpus. Any given dialog d can be represented as: $d = \{w_1, \dots, w_M\}$, where w_i indicates number of times the i^{th} term appears in the dialog. Let us denote by T , the term-document matrix $T_{M \times N}$, where t_{ij} is the number of times the term w_i appears in the dialog d_j .

2.2 Agent and Customer specific features

We separate the data from the Agent and the Customer into two term-document matrices A and C respectively. Here A_{ij} is the number of times term w_i appears in the part of the dialog d_j that is uttered by the agent and similarly C_{ij} is the number of times term w_i appears in the customer part of the dialog d_j . We then consider a weighted combination of the two matrices A and C to compute the overall T as follows

$$T = 2 \times (p \times A + (1 - p) \times C); p \in [0,1] \quad (1)$$

We let p vary from 0 to 1 to find the optimal p that results in the least misclassification.

3 Classification technologies

3.1 Vector Space Model

The Vector Space Model is a popular model in the Information Retrieval (IR) domain, see Raghavan and Wong [9]. The documents and queries are represented by a set of index terms in a space where the similarity between them can be computed using an inner product. The term document matrix is replaced by a weighted version using the using *tf-idf* (term frequency - inverse document frequency) formula, see Salton and Buckley [10]. The similarity between training dialog d_j and query q is calculated using the weighted vectors d_j and q using the cosine of the angle between the two vectors, and the class of the nearest neighbor in the training set is used as the class label for the test document.

3.2 Latent Semantic Analysis (LSA)

LSA is a technique similar to Principal Components Analysis and is used in text mining to reduce the dimensionality of the feature space. It has been argued that LSA attempts to capture the morphological relationship between similar features and hence extracts some semantic information LSA uses singular value decomposition to extract some linear combinations of the initial set of features as new features to work with. Please see Berry et al. [11] for details. We chose the dimensionality of the new feature space to be 200, after sufficient experimentation. Classification was done using nearest neighbor with the cosine distance measure.



3.3 *k*-Nearest neighbors (*k*-NN)

The two approaches described above assign the class label of the nearest neighbor in the training set to the test document. In *k*-NN, the majority class label of the nearest *k* training instances is assigned to the test document. Ties are broken randomly. We implemented *k*-NN for both the Vector Space and LSA paradigms. We chose *k* to be 3 and the cosine distance measure after sufficient experimentation

3.4 Multinomial Naive Bayes

The Naive Bayes classification technique is very popular in the text mining community, see Agrawal Srikant [12]. A successful adaptation of the Naive Bayes has been to assume that the words are not independent, but are generated through a multinomial distribution, see Vaithyanathan et al. [13], McCallum and Nigam [14], Chakrabarti et al. [15]. We have used Lidstone's law of succession to estimate class conditional term probabilities as in Raghavan and Wong [9].

3.5 Support Vector Machines (SVM)

Support Vector Machines (SVM) introduced by Vapnik, is one of the most exciting classification paradigms for its elegance, theory and successful applications in text mining and other areas. Please see Cristianini Shawe-Taylor [16], Joachims [17] for a comprehensive treatment of SVM technology. We have considered Kernels with radial basis functions as well as polynomials for our experiments. Multi-class classification was carried out using the one-against-one method implemented in OSU-SVM [18] and LIBSVM [19]. The penalty parameter *C* was always set to 1. We chose polynomials of degree 4 for the Polynomial kernels, All the parameters were chosen optimally based on experimental results.

4 Experimental setup

4.1 Data description and pre-processing

Our corpus consists of 621 calls obtained from the Leeds call center of General Electric's Consumer Finance division. These calls were obtained, manually transcribed in XML format. The transcribed files were processed through a Document Object Model (DOM) based XML parser into plain text, while also separating the dialog into an Agent part and Customer part. The text was then processed through Connexor's Functional Dependency Parser (now known as Machine Syntax) [20] to generate detailed morphological and syntactical information in XML format. The XML output was then parsed to identify the morphological tags associated with the set of grammatically correct features. The tags correspond to the stemmed version of the word. Stopwords were semi-automatically selected and eliminated. In particular, we eliminated words that were too common (e.g.: the, a, hello) and had little discriminating power. The



terms consisted of the stemmed versions of the words as well as digrams, as described earlier. The term document matrix was then built for each dialog. The calls were categorized into 15 classes manually. Some of the calls were assigned to more than one class. We then considered a subset of 539 conversations that were assigned to only one class. The distribution of dialogs into classes is shown in Table 1. The number of terms was equal to 1152 when only stemmed words were used, and there were 6603 terms in the case of using both words and digrams.

Table 1: Class frequency data.

Class	Number
Address Change	75
Name Change	8
Late Payment	16
Debit Payment	123
Account Premium	33
Lost of Credit Card	30
Technical Difficulties	7
Credit Line	9
Account Balance	60
Statement Clarification	27
New Account	96
Interest Free	11
Card Cancellation	23
Unauthorized person	3
Miscellaneous	18
Total	539

4.2 Implementation

The implementation was carried out in Matlab on a Xeon® powered CDC computer running Windows 2000. The Support Vector Machine software we worked with is the OSU SVM Matlab interface [18] with the LIBSVM software's C-SVC version [19]. Several types of experiments were conducted using all combinations of the following three cases.

- 1) 15 Class vs 4 Class: The data considered for the experiment consisted of either all the 539 calls for the 15 classes or the subset of data corresponding to the 4 largest classes- *Address change*, *Debit payment*, *Account balance* and *New account* that accounted for 66% of the data shown in Table 1.
- 2) Words vs Words and Digrams: The terms used in the experiment were either stemmed words or both stemmed words and digrams.
- 3) Agent vs Customer weight: The weight for the Agent was varied from 0 to 1 in steps of 0.1.

The data for each experiment was divided into training and test with 75% being training. 10-fold random cross-validation was used for robust validation.



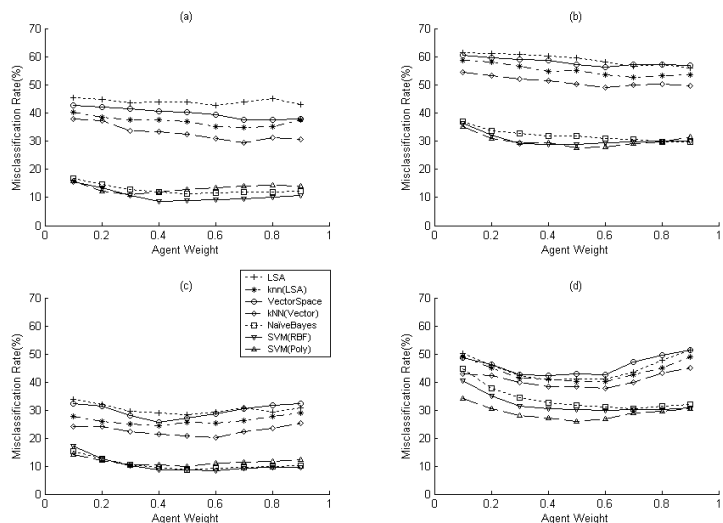


Figure 1: Results (a) 4 classes with words, (b) 15 classes with words, (c) 4 classes with both digrams and words (d) 15 classes with both digrams and words.

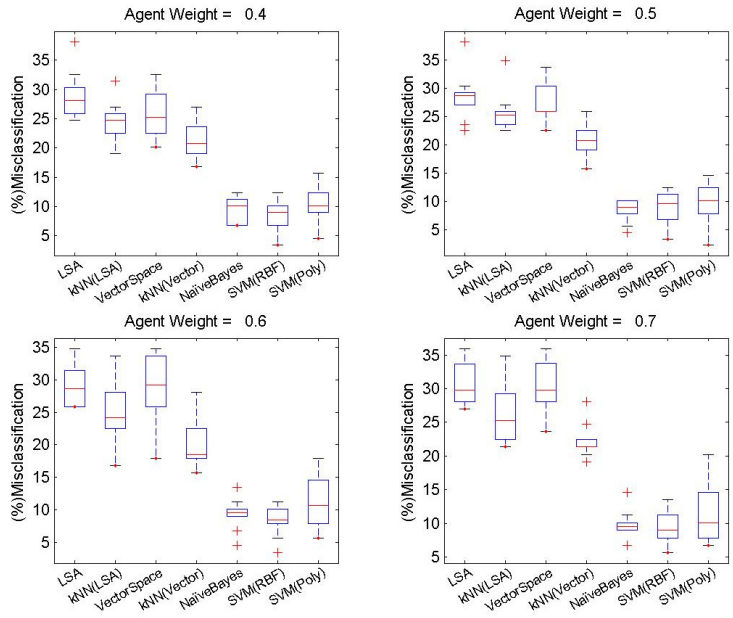


Figure 2: Box plots for each classifier for 4 classes considering both words and digrams.

5 Results

The results for all the experiments are presented in Figure 1. The points on the charts correspond to average misclassification rate over the 10 iterations of 10-fold cross-validation. The SVM and Naïve Bayes approaches performed significantly better than the Vector Space, LSA and k-NN approaches in all the experiments. Overall, SVM and Naïve Bayes classifiers performed the best. Incorporating both the words and digrams as features, SVM with RBF Kernel achieved the best accuracy with misclassification rate of 8.4% with an agent weight of 0.6 ($p = 0.6$). This is in comparison to SVM with polynomial kernel with almost 9.9% misclassification rate (with $p = 0.5$) and Naïve Bayes at 8.5% (with $p = 0.5$). In the general 15class problem, SVM polynomial kernel attained the best misclassification rate at 26% (with $p = 0.5$). SVM with RBF kernel was at 30% for $p = 0.6$ and Naïve Bayes was at 30.6% misclassification for $p = 0.7$.

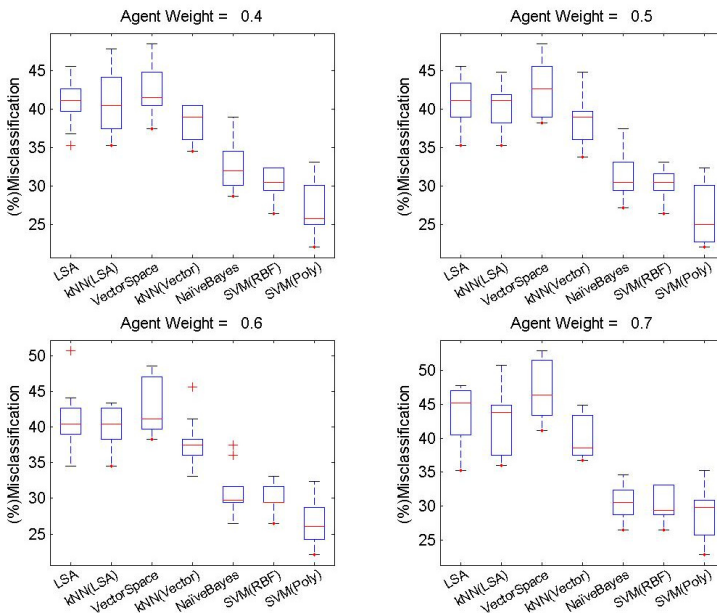


Figure 3: Box plots for each classifier for 15 classes considering both words and digrams.

Upon considering only words as features, the performance of the classifiers was consistently somewhat poorer for the 4 class problem for all agent weights. For the 15 class problem also, the same was generally true except for the Naïve Bayes classifier which performed a little better using only words as terms. In order to ensure the robustness of the results, we performed 10 fold cross validation of every experiment. Figure 2 shows the box plots for some of the

experiments for the 4 class problem, while Figure 3 shows the box plots for the 15 class problem.

6 Conclusion and future work

Our experiments provide evidence that very high bias in weightage towards customer or agent results in significantly poorer results. This is especially true for high weightage for the customer part of the dialog. Overall, there is no clear evidence to operate at unequal weights other than $p = 0.5$. Additionally, we find that using words and digrams gives some improvement over using only words as features. Overall, SVM classifiers along with Naïve Bayes perform the best. The results on the 4 class problem are significantly better than the general 15 class problem. This is easily explained by the lack of adequate training set data for most of the remaining 11 classes. We believe that with adequate amount of training data, the multi-class classification problem for call center dialogs can be solved to a high degree of accuracy. This may serve as a good starting point for business intelligence applications involving cross-sell and up-sell. In this paper, we have addressed only the single label multi-class classification problem. We would also like to consider the problem of solving the multi-label multi-class problem. Other interesting problems are emotion mining of conversations and dialog mining for quality control.

References

- [1] Chakrabarti, S., *Mining the Web Discovering Knowledge from Hypertext Data*. Morgan-Kaufmann Publishers, 2002.
- [2] Sebastiani, F., Machine learning in automated text categorization. *ACM Computing Surveys*, **34(1)**, 2002.
- [3] Carpenter, B. & Chu-Carroll, J., Natural language call routing: A robust self-organizing approach, 1998.
- [4] Chu-Carroll, J. & Carpenter, B., Vector-based natural language call routing. *Computational Linguistics*, **25(3)**, pp. 361–388, 1999.
- [5] Golden, J.A., Kimball, O., Siu, M.H. & Gish, H., Automatic topic identification for two-level call routing. *Proc. ICASSP '99*, Phoenix, Arizona, US, pp. 509–512, April 1999.
- [6] Rochery, M., Schapire, R., Rahim, M., Gupta, N., Riccardi, G., Bangalore, S., Alshawi, H. & Douglas, S., Combining prior knowledge and boosting for call classification in spoken language dialogue. *Proc. ICASSP '2002*, Orlando, Florida, US, 2002.
- [7] Myers, K., Kearns, M., Singh, S., & Walker, M.A., A boosting approach to topic spotting on sub-dialogues. *Proceedings of ICML-00, 17th International Conference on Machine Learning*, ed. P. Langley, Morgan Kaufmann Publishers, San Francisco, US: Stanford, US, pp. 655–662.
- [8] Joachims, T., A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. *Proceedings of ICML-97, 14th International*



- Conference on Machine Learning*, ed. D.H. Fisher, Morgan Kaufmann Publishers, San Francisco, US, pp. 143–151, 1997.
- [9] Raghavan, V. V. & Wong, S. K. M. A critical analysis of vector space model for information retrieval. *Journal of the American Society for Information Science*, **37(5)**, p. 279–87, 1986.
 - [10] Salton, G. & Buckley, C. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, **24(5)**, pp. 513–523, 1988.
 - [11] Berry, M. W., Dumais, S. T. & O'Brien, G. W., Using linear algebra for intelligent information retrieval. *SIAM Review*, **37(4)**, pp. 177–196, 1995.
 - [12] Agrawal, R., Jr., R.J.B. & Srikant, R., Athena: Mining-based interactive management of text databases. *Extending Database Technology*, pp. 365–379, 2000.
 - [13] Vaithyanathan, S., Mao, J.C. & Dom, B., Hierarchical bayes for text classification. *PRICAI Workshop on Text and Web Mining*, pp. 36–43, 2000.
 - [14] McCallum, A. & Nigam, K., A comparison of event models for naive bayes text classification. *AAAI/ICML-98 Workshop on Learning for Text Categorization*, AAAI Press, pp. 41–48, 1998.
 - [15] Chakrabarti, S., Dom, B., Agrawal, R. & Raghavan, P., Using taxonomy, discriminants, and signatures for navigating in text databases. *Proceedings of the 23rd VLDB Conference*, pp. 446–455, 1997.
 - [16] Cristianini, N. and Shawe-Taylor, J., *An Introduction to Support Vector Machines (and other kernel-based learning methods)*. Cambridge University Press, 2000.
 - [17] Joachims, T., *Learning to Classify Text using Support Vector Machines*. Kluwer Academic Publishers, 2002.
 - [18] Ma, J. & Ahalt, S., *OSU SVM classifier Matlab Toolbox* (Version 3.00). Ohio State University, Columbus. URL: http://www.ece.osu.edu/~maj/osu_svm/.
 - [19] Chang, C., -C. & Lin, C., -J., *LIBSVM: a library for support vector machines*. 2001. URL: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
 - [20] Connexor Machine Syntax. URL: http://www.connexor.com/m_syntax.html.

