Automatic Call Quality Monitoring Using Cost-Sensitive Classification

Youngja Park

IBM T.J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598, USA
young_park@us.ibm.com

Abstract

In this paper, we propose advanced text analytics and cost-sensitive classification-based approaches for call quality monitoring and show that automatic quality monitoring with ASR transcripts can be achieved with a high accuracy. Our system analyzes ASR transcripts and determines if a call is a *good* call or a *bad* call. The set of features were identified through analysis of a large number of human monitoring results, which aim to estimate agent's attitude and customer's sentiment during the call. To enhance the accuracy of feature extraction, we apply various techniques to improve the quality of transcribed calls, such as sentence boundary detection and disfluency removal. We further note that quality monitoring has skewed class distribution and unequal classification error costs, and thus apply cost sensitive classification algorithms. Validation on 386 customer calls confirms the benefits of our approach. A SVM-based method produces a classification accuracy of 83.16% and 67.66% in $F_1 Score$ for identifying *bad* calls, which is promising. This system can therefore be used to conduct initial monitoring of all the calls in a contact center and to select calls that require human monitoring.

Index Terms: Speech Analytics, Contact center calls, Call Quality Monitoring, Call Classification

1. Introduction

Customer and agent conversations are a valuable source of insights into the contact center operations and also the company's overall business. Recently, many speech applications for contact center calls have gained much attention from researchers, such as automatic call routing through an interactive voice response system or word spotting [1, 2], call topic classification based on a predefined domain taxonomy [3, 4, 5, 6, 7], information retrieval on contact center conversations [8, 9], and customer satisfaction measurement [10]. In this paper, we focus on automated call quality monitoring. Call quality monitoring is a mandatory operation in call centers, and is typically conducted by call center managers by listening to a very small fraction of randomly selected calls. While manual monitoring is very costly, it provides a limited value to the contact center. First, only a very small fraction of calls can be monitored due to the high cost of listening to recorded calls. Second, most of the monitored calls are ordinary, as the calls are usually randomly selected. Automatic call monitoring can thus provide a significant benefit to call centers by enabling monitoring of all calls in real time.

We propose advanced text analytics and cost sensitive classification algorithms for call quality monitoring. We identified various features, including lexical and semantic features, through analysis of a large number of human monitoring results. These features aim to estimate the agent's attitude toward the caller, the agent's call handling skills and the customer's sentiment during the call. Automatically transcribed calls (hereafter, ASR transcripts) pose many challenges for text analysis systems including many disfluencies, misrecognized words, and ill-formed sentences. To enhance feature extraction accuracy, we apply various text analytics techniques such as removal of disfluency and interjection and sentence boundary detection to improve the quality of ASR transcripts. Furthermore, we note that the distribution of calls in most contact centers is biased, i.e., much more calls are rated as *good*; and misclassifying *bad* calls as *good* calls results in a higher penalty. Cost sensitive learning techniques have been proved to be very effective to address these issues [11, 12].

Validation on 386 customer calls to an automotive company shows very promising results. We conducted a 10-fold cross validation using cost sensitive versions of Support Vector Machines (SVM) and Logistic Regression (LR) for classification. Both SVM and LR perform very well: an accuracy of 83.16% and 82.12% and AUC of 0.7888 and 0.758 respectively. Furthermore, the systems produce promising results for identifying *bad* calls. Therefore, the systems can be used to do monitoring of all calls in a contact center and identify calls that may require human monitoring.

2. Quality Monitoring in Contact Centers

Most call centers conduct manual monitoring of a very small percentage of calls for general quality and training purposes. Typically, human monitors listen to a random sample of calls and score the calls with respect to the company's quality monitoring (QM) questionnaire. In this work, we use the QM questionnaire used in an automotive company. The 23 questions in the QM questionnaire are *yes-no* questions, and each question has an associated score. A call is rated as a *good* (or *bad*) call, if the total score is equal to or above (or below) a threshold predefined by the company.

The questions concern various aspects of call quality ranging from the agent's call handling skills and attitude to call accuracy. Some sample QM questions include:

- Treated customer courteously and showed genuine concern
- Followed current call scripts for opening, hold/transfer process, and closing
- Understood customer request
- Documentation is accurate and complete
- Provided accurate information and correct resolution

- Maintained confidentiality

As we can observe, these questions have different levels of complexity to answer. Some questions can be easily answered by listening to the call (e.g., "treated customer courteously"), while other questions require human-level domain knowledge (e.g., "provided accurate information") and cross-examination of the call content with information in external resources (e.g., "documentation is accurate"). The latter questions can not yet be answered by an automated system. We found that 10 questions out of the 23 questions can be estimated using information found in call transcripts. Thus, in this work, we try to simulate a quality monitoring process by estimating the quality rating of a call based on the 10 questions. We define call monitoring as a binary classification problem, which categorizes calls into *meet or exceed expectation* (*good*) calls and *below expectation* (*bad*) calls.

3. Automatic Call Monitoring Algorithms

In this section, we describe the technical details of the proposed quality monitoring systems. The input to the system is an ASR transcript, and the output is a binary decision on if the call is a *good* call or a *bad* call.

3.1. ASR Transcript Preprocessing

Automatically generated call transcripts pose many challenges for content analysis including many disfluencies, idiosyncratic expressions and disconnected sentences due to frequent interjections and overlapped speeches. We apply several text analysis tools to improve the quality of ASR transcripts and to subsequently enhance the accuracy of feature extraction.

Filler word removal: Fillers are words or sounds that people often say unconsciously that add no meaning to the communication, for example, "ah", "uh", "umm", etc. We compiled 24 filler words and remove all occurrences of the filler words from transcripts. If the removal of fillers produces an empty speaker turn (i.e, the original speaker turn contains only filler words), we remove the speaker turn from the transcript.

Interjection removal: An interjection is a short expression spoken by a speaker between two utterances of the other speaker. Most interjections are used to show the listener is engaged in the conversation, for example, "okay", "yep", "right", etc. We remove these interjections and combine the the two speaker turns interrupted by the interjection.

Expression Normalization: Acronyms, alphanumerics (e.g., case number) and numbers (e.g., phone number) are typically spelled out in ASR transcripts. For instance, a telephone number is transcribed as "one eight hundred one two three four five six seven". For acronyms, we concatenate the initials into a token; for alphanumerics and numbers, we convert the numbers to the corresponding decimal representation, and merge them into a single token (e.g., 18001234567).

Utterance Boundary Detection: ASR transcripts contain a sequence of words together with the speaker id, the start time and the duration. We apply an utterance boundary detector to divide the continuous stream of words into sentences [13]. The system uses linguistic and prosodic features such as the probabilities of unigrams and bi-grams occurring as the first or last unigram (or bi-gram) in utterances and the length of pauses between two words.

Call Segmentation: Most contact center calls follow a standard call flow such as "greeting section", "question section", "resolution section", and "closing section". Call segmentation is a process to automatically divide a call into a sequence of call sections. We identified 12 section types for the automotive company, including both domain independent (e.g., "greeting") and domain dependent (e.g., "VIN identification section"). We apply SVM-based classification to each utterance, and merge adjacent utterances of the same section into a section [13].

3.2. Features

We analyzed the QM questionnaire used in the automotive company's contact center to learn which aspects of calls and agents are examined for quality monitoring. As discussed in Section 2, not all questions can be directly answered with information from call transcripts, but, call transcripts provide valuable clues for judging the call quality. We identified the following 9 features that can estimate important aspects for call quality and can be extracted from call transcripts. Note that the features are domain independent, including timing information, lexical information and semantic information. The values for class-specific word features are normalized into a range of [0, 1], and all other features have a binary value (1 or 0) based on a threshold. The threshold for each feature is the average value obtained from about 1,700 call transcripts.

Long silences by the agent: The feature indicates if the agent made many long pauses during the call. Long pauses includes any silence periods initiated by the agent which lasted longer than 5 seconds. If the number of long pauses is below the average number, the feature value is set to 1, otherwise to 0.

Agent talk speed: The talk speed of a speaker is calculated by the number of words spoken by the speaker divided by the total talk time by the same speaker during the call. Note that the total speech time does not include silences. If the agent's talk speed is slower than the average speed, then the feature value is 1, else 0.

Cue words denoting a follow up call scheduling: This feature aims to judge if the caller's problem was resolved during the call. When the agent is not able to solve the problem, she usually schedules a follow-up call at the end of the call. The scheduling action can be identified with common cue phrases such as "call you back", "give a call", "contact you', "schedule", "follow up", and various time expressions such as "eastern time", and "ten o'clock on Monday", etc. We count the number of the occurrences of the cue phrases in the "Closing" section, and set the feature value to 1 if the count is greater than the average, and 0 otherwise.

Filler words: The frequency of fillers in a conversation is often reflective of a speaker's emotional state (e.g., lack of confidence). Contact centers encourage the agents to minimize the use of fillers. We count the number of fillers in the agent's utterances during the transcript preprocessing step, and when the number is below the average, the feature value is set to 1.

Sentiment words: We count the use of sentiment words by the agent to measure the agent's attitude, and the sentiment words by the customer to estimate customer satisfaction. To identify words with sentiment polarity, we use the subjectivity lexicon described in [14, 15]. The lexicon contains a list of words with a priori polarity (*positive*, *negative*, *neutral* and *both*) and the strength of the polarity (*strongsubj* vs. *weaksubj*). In this work, we use only words of which prior polarity is either *positive* or *negative*, and the strength of the polarity is *strongsubj*. We then removed a few words which are frequently used non-subjectively in conversational text such as "okay", "kind", "right", and "yes".

We analyze a local context analysis to decide the polarity of a sentiment word in the given context. If a sentiment word has a polarity shifter within a three word window in the left, the polarity of the word is changed based on the shifter [16]. For instance, if a positive sentiment word appears with a negation word, the polarity of word in the context is negative. We then count the number of positive sentiment words and negative sentiment words spoken by each speaker, and set the feature value to 1 if the count is below the average.

Class-specific words: Some set of words tend to appear more frequently in a class than the other classes. We call these words category-specific words, and automatically extract category-specific words based on Shannon's entropy [17]. We calculate the probabilities of a word, w, appearing in each class, $p_c(w)$, and compute the entropy of the word, $H(w)$, as defined in Equation 1.

$$H(w) = -\sum_{c \in \{good, bad\}} p_c(w) \cdot log p_c(w) \quad (1)$$

where $p_c(w) = \frac{f_c(w)}{f(w)}$, $f_c(w)$ denotes the counts of word w in class c and $f(w)$ is the total count of w. If the entropy of a word is greater than a threshold, then the word is regarded as category-specific. Furthermore, a category-specific word w is regarded as a *good* call word, if $p_{good}(w) > p_{bad}(w)$, and a *bad* call word if $p_{good}(w) < p_{bad}(w)$. We count the number of *good* call words and *bad* call words in a transcript, and normalize the counts into a range of [0,1].

3.3. Machine Learning Systems

We validate the analytics with Logistic Regression and Support Vector Machines, which have been successfully used in many natural language processing and speech applications. As discussed earlier, the distribution of calls in contact centers is biased. Furthermore, the penalty of missing a *bad* call is greater than mis-identifying a *good* call as a *bad* call. To address these characteristics, we apply cost-sensitive versions of SVM and Logistic Regression classification. We use *RapidMiner* [18], a machine learning toolkit offering a wide range of methods for data pre-processing, machine learning and validation for all experiments. Specifically, we use the C-support vector classification (C-SVC) with a linear kernel for SVM, and a radial kernel for Logistic Regression. The cost for misclassification of *good* calls is set to 1, and the cost for misclassification of *bad* calls is set to 3. The 1:3 cost ratio was determined based on the call distribution in the experimental data set (see Section 4.1), and it can be tuned for a different data set.

4. Experiments

4.1. Experimental Data

We obtained 386 customer in-bound calls to the automotive company that have both recordings and human monitoring results. These calls concern a wide range of customer issues including inquiries on recalls or reimbursements, questions related to car defects, and complaints about vehicles or dealerships. To build supervised learning models and conduct performance evaluation, we use the manual monitoring results for the calls as the ground truth. 269 calls (69.7%) out of the 386 calls are labeled as *good* calls by human monitors and 117 calls (30.3%) are labeled as *bad* calls. Therefore, a simple baseline system that classifies all calls as *good* would produce 69.7% accuracy.

4.2. Performance Evaluation

To validate our algorithms, we conducted a 10-fold cross validation on the data set. In this setting, we randomly shuffle the entire data set, D, and divide it into 10 disjoint subgroups, $D = \{D_1, D_2, \cdots, D_{10}\}$. At each validation step $i, 1 \leq i \leq 10$, a subset D_i is held out for evaluation, and the remaining data sets, $D \setminus D_i$, are used for training a model. This process is repeated 10 times with each D_i used exactly once as the validation data. The results from the 10 validation processes can then be averaged to produce a single estimation for the algorithm.

Figure 1 depicts the average of overall classification accuracy, AUC (the area under the ROC curve), and F_1 scores for both *good* calls and *bad* calls from 10-fold cross validation. As

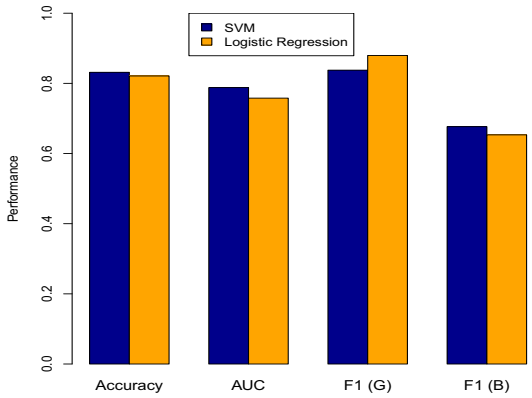


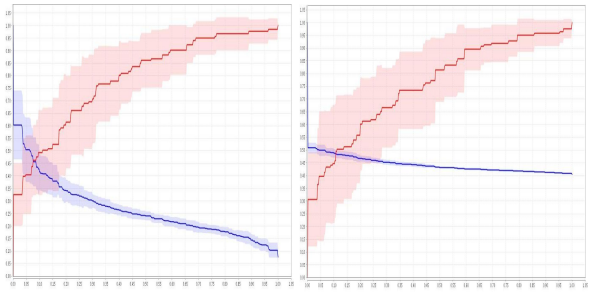
Figure 1: Comparison of accuracy, AUC, F_1 Scores of the two classification algorithms

we can see from the chart, both SVM and Logistic Regression perform very well. SVM and LR produce an overall classification accuracy of 83.16% and 82.12% and AUC of 0.788 and 0.758 respectively. Both SVM and LR outperform the baseline by 19.3% and 17.8% respectively. Overall, SVM outperforms Logistic Regression resulting in higher classification accuracy, AUC and F_1 for *bad* calls.

Note that the classifiers produce probabilities of a call belonging to each class (i.e., *good* and *bad*), and the final binary decision is made based on a threshold. A ROC curve depicts a classifier's performance across all possible threshold points in terms of true positive rate (sensitivity) and false positive rate (1-specificity). Figure 2 shows the ROC (Receiver Operating Characteristic) curves of SVM and Logistic Regression, in which the values are the average of the 10-fold cross validation results. In these charts, the plots on top are the ROC curves, and the plots below the ROC curves show the thresholds used to generate the averaged ROC.

5. Related Work

Since human monitors can only listen to a very small fraction of calls, there is a strong demand for automatic call monitoring systems. However, little work has been done in this area. Zweig *et al.* developed a pioneering system for automatically assigning quality scores to call center conversations [19]. To our knowledge, this system is the only attempt made to-



(a) SVM (b) Logistic Regression

Figure 2: ROC Curves resulted from 10-fold cross validation using SVM and Logistic Regression

ward automated quality monitoring of human-human conversations. The work has explored two different approaches; a pattern matching-based question answering approach and maximum entropy classification approach. The question answering method uses a small number (typically one or two) of precompiled phrases for each question in the call monitoring questionnaire, and looks for phrases in the entire call transcript. If the phrase is present in the call transcript, the question is regarded as satisfied. The maximum entropy-based approach determines if a call is *bad* based on a set of prosodic features such as the number of silences and a list of precompiled n-gram word sequences. However, both approaches are based on a very small set of simple features, and do not take into account more advanced features used in our work. They reported that the system produced the best result when both approaches were combined, but the accuracy was quite low. The combined approach showed 53% accuracy for identifying the best 20% of calls and 44% accuracy for identifying the bottom 20% calls in a test set.

6. Conclusions

In this paper, we proposed advanced text analytics and cost sensitive classification-based approaches for automated call quality monitoring. Even though quality monitoring is very difficult to automate, we demonstrated that our algorithms achieved over 82% classification accuracy. The results confirm that the system can be used to monitor all the calls in a contact center in real time and to sample the calls requiring thorough human monitoring.

7. References

[1] J. Chu-Carroll and B. Carpenter, "Vector-based natural language call routing," *Computational Linguistics*, 1999.

[2] G. Riccardi, A. Gorin, A. Ljolje, and M. Riley, "A spoken language system for automated call routing," in *Proceedings of Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP'97)*, 1997, pp. 1143–1146.

[3] S. Busemann, S. Schmeier, and R. G. Arens, "Message classification in the call center," in *Proceedings of the sixth conference on Applied natural language processing*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 158–165.

[4] P. Haffner, G. Tur, and J. Wright, "Optimizing svms for complex call classification," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, 2003.

[5] S. Roy and L. Subramaniam, "Automatic generation of domain models for call-centers from noisy transcriptions," in *Proceedings of COLING-ACL 2006*.

[6] M. Tang, B. Pellom, and K. Hacioglu, "Call-type classification and unsupervised training for the call center domain," in *IEEE Automatic Speech Recognition and Understanding (ASRU) Workshop*, 2003.

[7] G. Tur, D. Hakkani-Tur, A. Stolcke, and E. Shriberg, "Integrating prosodic and lexical cues for automatic topic segmentation," *Computational Linguistics*, vol. 1, no. 27, pp. 31–57, 2001.

[8] J. Mamou, D. Carmel, and R. Hoory, "Spoken document retrieval from call-center conversations," in *Proceedings of SIGIR'06*.

[9] G. Mishne, D. Carmel, and R. Hoory, "Automatic analysis of call-center conversations," in *Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM)*, 2005.

[10] Y. Park and S. C. Gates, "Towards real-time measurement of customer satisfaction using automatically generated call transcripts," in *CIKM '09: Proceedings of the 18th ACM Conference on Information and Knowledge Management*, 2009, pp. 1387–1396.

[11] P. Domingos, "Metacost: A general method for making classifiers cost-sensitive," in *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, 1999, pp. 155–164.

[12] C. Elkan, "The foundations of cost-sensitive learning," in *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)*, 2001.

[13] Y. Park, "Automatic call section segmentation for contact-center calls," in *CIKM'07: Proceedings of the 16th ACM Conference on Information and Knowledge Management*, 2007, pp. 117–126.

[14] T. Wilson, P. Hoffmann, S. Somasundaran, J. Kessler, J. Wiebe, Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan, "OpinionFinder: A system for subjectivity analysis," in *Proc. Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-2005) Companion Volume (software demonstration)*, 2005.

[15] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Vancouver, Canada, 2005, pp. 347–354.

[16] L. Polanyi and A. Zaenen, "Contextual valence shifters," in *Working Notes — Exploring Attitude and Affect in Text: Theories and Applications (AAAI Spring Symposium Series)*, 2004.

[17] C. E. Shannon, "A mathematical theory of communication," 1948.

[18] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler, "Yale: Rapid prototyping for complex data mining tasks," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006, pp. 935–940.

[19] G. Zweig, O. Siohan, B. Ramabhadran, D. Povey, L. Mangu, and B. Kingsbury, "Automated quality monitoring in the call center with ASR and maximum entropy," in *Proceedings of ICASSP*, 2006.