RC24574 (W0806-018) June 5, 2008
Computer Science

IBM Research Report

Your Call May Be Recorded for *Automatic* Quality-Control

Youngja Park
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Your call may be recorded for *automatic* quality-control

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

ABSTRACT

A call center quality control process typically relies on human labor to evaluate sample conversations according to a quality monitoring (QM) questionnaire. Due to the effort involved, the sample of calls evaluated is often very small and likely to miss problematic calls.

This paper presents an automatic call quality monitoring (QM) system for contact centers, which applies natural language processing (NLP) and machine learning techniques. Specifically, the system aims at categorizing contact center calls into *good* calls which meet or exceed a company's quality expectation and *bad* calls which are below the expectation.

In this work, we first transcribe a call using an automatic speech recognition (ASR) system, and extract features from the call transcript using various text mining techniques. The features include timing features, lexical features and structural features that indicate various aspects of call quality. We then apply maximum entropy classification to decide if a question in a company's QM questionnaire is satisfied or not resulting in as many maximum entropy classifiers as the number of questions in the QM questionnaire. The system produces a score for each question depending on the classification result. All scores are then combined to generate a quality score for the call. If the total quality score is above a predetermined threshold, the call is regarded as a *good* call.

We have conducted experiments with 387 customer calls to an automotive company. The system was trained using 310 calls with associated manual monitoring results and tested on the remaining 77 calls. 70% calls in the training data were rated as *good* calls by human monitors. The experimental result shows 72.7% classification accuracy, which is very promising given the fact that the system was trained with a very small and highly biased data set.

Categories and Subject Descriptors

H.4 [**Knowledge Management**]: Mining and representing text, Classification

General Terms

Algorithm, Design, Experimentation

Keywords

Call Quality Monitoring, Maximum Entropy Classification, Natural Language Processing, Text Mining, Contact Center Analytics, Speech Analytics

1. INTRODUCTION

Most companies that use call centers sample a small percentage of conversations to make sure agents are following scripts and 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. Quality monitoring questions are usually *yes-no* questions, and each question has an associated score. If a question is satisfied (*yes*) in a call, then the call receives the score associated with the question. A call is regarded as a *good* or *bad* call if the total score is above (or below) a threshold value predefined by the company.

QM questions concern various aspects of call quality ranging from the agent's attitude to call accuracy and to compliance with the company's confidentiality. Furthermore, a QM questionnaire consists of questions with different levels of difficulties. Some questions can be easily answered by listening to the call, while other questions require the human monitors to cross-examine external resources or to be very knowledgable on the customer's issue and the company's policy. Some examples of QM questions are listed in Table 1.

1.	Treated customer courteously and showed genuine concern
2.	Followed current call scripts for opening, hold/transfer process, and closing
3.	Understood customer request
4.	All customer information is documented accurately in the service database
5.	Provided correct resolution
6.	Maintained confidentiality

Table 1: Sample quality monitoring questions. Question 1–3 can be answered by human monitors while listening to the calls. However, question 4–5 require external information or deep knowledge on the customer's problem and the company's policy.

Manual call quality monitoring process, however, has a limited value due to the following problems. First, only a very small fraction of calls can be monitored due to high cost of listening to recorded calls. Second, most of the monitored calls are ordinary as the calls are usually randomly selected. Therefore, any automatic or semi-automatic QM system which can monitor 100% calls in a contact center and can identify calls worthy of human monitoring would be very valuable.

Recently, text analytics on contact center calls (also called speech analytics) have gained much attention from researchers and businesses alike. Much of the effort applying natural language processing (NLP) and speech recognition technologies in this domain has focused on automatic call routing [7, 18], call topic classification [6, 10, 19, 21, 22], and information retrieval from contact center conversations [13, 14]. Up to now, however, there has been little effort by both speech and natural language processing (NLP) communities on building automatic quality monitoring systems. The main reason would be technical difficulties in answering questions which seem only possible by domain experts. We, however, believe that continuing advances in automatic speech recognition and natural language processing make automatic quality monitoring process feasible.

In this paper, we propose an automatic quality monitoring method based on natural language processing and machine learning technologies. More specifically, we first identified a set of features which can be automatically extractable from speech transcripts using text mining techniques and can correlate a given call with the quality score with high accuracy. We then built a maximum entropy-based classifier for each question in a QM questionnaire, which estimates if the question is satisfied or not in the call. This method therefore comprises as many maximum entropy classifiers as the number of questions in a company's QM questionnaire. Each classifier generates a score for the question based on the classification result. When the question is satisfied, the score is the probability of the call being *good* when the question is satisfied. When the question is not satisfied, the score is set to the probability of the call being *good* when the question is not satisfied. If the total score of the call is above the company's threshold, the call is regarded as a *good* call. Otherwise, it is regarded as a *bad* call.

Initial experiments were conducted with 387 calls to an automotive company. In this work, we use a state-of-the-art automatic speech recognition (ASR) system for transcribing the calls, which is an updated version of the ASR system described in [20]. We used 310 calls (80%) for development and training, and 77 calls (20%) for evaluation. The system classified the test calls with 72.7% accuracy. Furthermore, the system identified 2.7 times more bad calls than random selection when the bottom 20% of calls was sampled from a sorted list of the calls by their quality scores. Even though we used a very small size of data for development and training, the experimental results are promising. The system can help contact centers sample more bad calls for human monitoring and thus can reduce the number of calls to be manually monitored which results in huge cost saving.

The remainder of the paper is structured as follows. We first introduce the background system, Contact-center Agent Buddies system, and the ASR system used in this work in Section 2. Our proposed approach is described in detail in Section 3. Section 3.1 outlines the overall structure of the proposed system. Section 3.2 describe the feature set used for automatic call monitoring. Section 3.3 explains how to estimate if a question in a QM questionnaire is satisfied or not, and Section 3.4 describes how scores for the individual questions are computed. We outline experimental setup in Section 4, and discuss the experimental results and performance evaluation results in Section 5. The comparison of our approach with previous approaches is done in Section 6. Finally, we discuss the results and future work in Section 7.

2. BACKGROUND SYSTEM

In this section, we describe a high-level overview of the background systems, Contact-center Agent Buddies (CAB) in which the automatic quality monitoring system is built, and the automatic speech recognition system used for the CAB system.

2.1 Contact-center Agent Buddies

The presented quality monitoring system was developed as a part of the Contact-center Agent Buddies (CAB) project. The goal of the CAB project is to develop speech analysis systems for improving agent productivity and call center operations. The CAB system consists of two components: a suite of agent assistant systems (i.e., Agent Buddies) and a Transformational Diagnostic Tool (TDT). The Agent Buddies intend to process (or listen in) an on-going call in real time and to help agents reduce the call handling time. Some examples of Agent Buddies include a system which proposes solution documents to the customer's inquiry, a call log generation system which produces a summary of the call, and a call quality monitoring system. The TDT is an off-line system which analyzes a large amount of heterogeneous data including call transcripts, call logs and structured data from a contact center to extract business insights and to identify opportunities for improvement in call center operation or the company's business. Most subcomponents of the CAB system were implemented using the Unstructured Information Management Architecture (UIMA) [8]. UIMA is an architecture and software framework for creating, discovering, composing and deploying a broad range of multi-modal analysis capabilities [1].

Figure 1 depicts the high-level architecture of the CAB system. When an agent is taking a call, the ASR system transcribes the call, and the Agent Buddies analyze the call transcript in real time and generate recommendations such as candidate solutions or a call log. The call transcript is then stored and analyzed later by the TDT together with other information such as call logs and structured information for extracting business insights.

The call quality monitoring system is used both for the real-time component as an Agent Buddy and for the TDT. As an Agent Buddy, the tool can provide real-time feedback to the agent by showing the estimated quality score of the call. Furthermore, the system can provide comparative analysis results between the current call and previous calls handled by the agent and between the call and all calls belonging to a same topic. The TDT adds quality score as a dimension (or a property) for call analysis. For instance, we can examine if there is any correlation between quality scores and call topics. If calls on a certain topic tend to have

[1] The Apache UIMA open source can be downloaded from http://incubator.apache.org/uima/

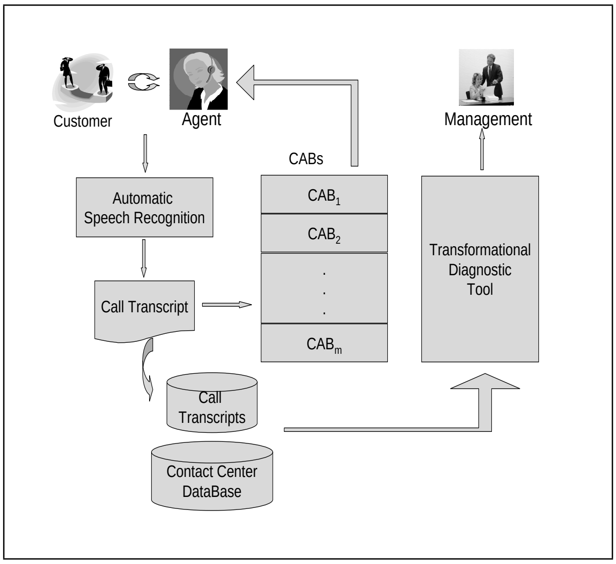

Figure 1: High-level architecture of the Contact-center Agent Buddies (CAB) system. The CAB system is a speech analytics application comprising one or more Agent Buddies and a TDT (Transformational Diagnostic Tool).

lower (or higher) scores than average quality score, then the contact center management can look into the issue and provide a solution to improve call quality on the issues.

2.2 Automatic Speech Transcription

We used the IBM Research Attila Speech recognition toolkit to transcribe the contact center calls [16, 20]. The ASR system uses a large US English vocabulary and works in speaker-independent mode. The acoustic model was built applying various state-of-the-art techniques such as VTLN (vocal tract length normalization), FMLLR (Feature-space Maximum Likelihood Linear Regression) and MLLR for handling mismatch between training data and test data with linear transforms, and fMPE (Feature-space Minimum Phone Error) and MPE for training the speaker adaptive models. The language model is a mixture of a general language model and a domain-specific language model.

An ASR system typically needs to be retrained on domain-specific conversational data to produce good quality transcriptions. In this work, we selected approximately 340 hours of customer calls to the automotive company and generated manual transcriptions for the calls for retraining the ASR system for the domain. The acoustic model were trained on the customer calls in addition to approximately 2,000 hours of general conversational telephony speech data. The general language model was trained on data from various sources including conversational telephony speech and broadcast news. The domain-specific language model was trained on the manual transcriptions of the customer calls.

Call transcripts generated by the ASR system contain speaker turns, words, and the start times and the durations for the recognized words. The speech system shows an overall word error rate of 26.4%, which is reasonably accurate for telephony conversations. Furthermore, the transcripts have very accurate speaker turns as the contact center uses VoIP (Voice-over-Internet Protocol) telephony which provides separate channels for the agent and the caller. Accurate speaker turn information is important for automatic call monitoring because a word can have different implications depending on who spoke the word. For instance, “appreciate” at the end of a call may indicate the customer’s problem was resolved if it was spoken by the customer. The start time and the duration of each word are used to compute the length of the gap between two words, and to identify silences during a call.

3. THE APPROACH

In this section, we present the technical details of the call quality monitoring system including the feature set, the maximum entropy classifiers for judging if individual questions were satisfied, and the learning method for estimating the scores for the quality questions.

3.1 System Overview

The presented system aims at estimating quality scores of contact center calls and at classifying the calls into *good* calls and *bad* calls with respect to a company’s quality monitoring guideline. Answering the questions in a company’s QM questionnaire requires human-level intelligence and knowledge as well as good language skills. An intelligent system may be able to judge the agent’s attitude such as ‘did the agent treat the customer courteously?’ by looking for words and phrases expressing the attitude (i.e., courteousness) in the call. However, a question like ‘did the agent demonstrate authority to handle the request?’ certainly requires more than lexical knowledge. Also, some questions are inter-connected. For example, ‘was the agent knowledgeable to handle the request’ and ‘was the request resolved in a timely manner’ are related each other. Therefore, some information can be used for answering multiple questions. Furthermore, some questions can not be directly answered only with information extracted from the call transcript. For instance, some questions check if the agent created correct documentations on the call in the contact center’s database. Some other questions examine if the agent kept the company’s confidentiality. For these reasons, we decided not to take a direct question answering approach.

In this work, we first identify a set of features which indicate some aspects of call quality and can be automatically extracted from call transcripts. Then, we apply maximum entropy classification to estimate if each individual question is satisfied or not in a given call based on the feature set. In other words, we build a maximum entropy classifier for each question resulting in as many classifiers as the number of questions in the QM questionnaire. The system also generates a score for each question depending on the classification result. In this work, each question is assigned with two scores: a score when the question is satisfied and the other score when the question is not satisfied. The scores are defined as the probability of the call being *good* when the given question is satisfied and the probability of the call being *good* when the question is not satisfied respectively. The scores for questions are learned from manual quality monitoring results in a separate step. Finally, we normalize the individual scores to generate a total quality score for the call. When the score is above the threshold for “meet or exceed expectation”, the call is regarded as *good.* Otherwise, the call is regarded as *bad.* The overall system is depicted in Figure 2.

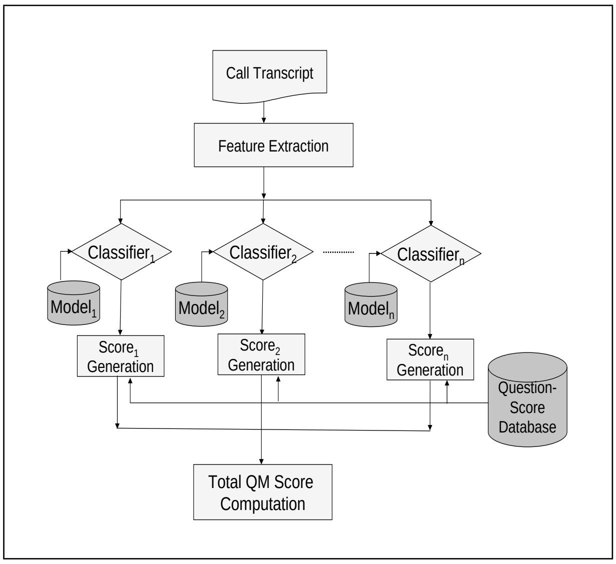

Figure 2: System diagram for the call monitoring system. 'n' denotes the number of questions in a QM questionnaire. Classification models and the question-score database are created at the training phase separately.

3.2 Features

The features should allow the models to learn various aspects of call quality. We identified 20 different features based on our analysis on the automotive company's quality monitoring questionnaire. Note that different companies employ different set of questions for quality monitoring. Therefore, the feature set needs to be determined based on the target company's QM questionnaire. In this work, the feature set is primarily designed for the automotive company, but we believe that the features are very general and applicable to other industries and companies.

In this work, the feature set includes lexical features to capture specific words and phrases, timing features to assess if the call was proceeded smoothly, and structural features to learn contexts that can not be directly captured by linguistic patterns. Lexical features are words or phrases often used in certain situations (e.g., greeting and closing phases) in contact center calls. In this work, we apply a cascade of finite state machines over call transcripts to recognize the words and phrases. Finite state machines have been efficiently used in many rule-based information extraction systems [5, 11]. Timing features are non-lexical conversational information such as the number of long silences and the total length of silences during a call. These features can be extracted from call transcripts, as modern automatic speech recognition systems typically generate the start time and the ending time (or the duration) for the recognized words.

Structural features include high level contextual or topical information which can not be easily captured by lexical features. In this work, we use call segment information as structural features. A call segment consists of a set of consecutive utterances in a call which belong to a same activity or phase during a call. Some examples of the call segments are "greeting section", "transfer section", "question section", "resolution section", "follow-up scheduling section", "out-of-topic section" and "closing section". Information on the presence of a call section or the length of a call section can be a valuable feature for estimating call quality. For instance, if the agent scheduled a follow-up call at the end of a call, that can indicate the customer's request was not resolved during the call. We have built a support vector machine-based approach to recognize different call segments in a call transcript (see [15] for more details on call segmentation).

The lexical, timing and structural features are all intended to estimate one or more aspects of call quality as listed in the following.

- Call Handling Skills
  Features in the category measure the agent's overall call handling skills. The features include the number of long silences and the total length of the long silences. In this work, a silence longer than 5 seconds is considered long. Another interesting feature in this category is the number of filler words [2] spoken by the agent. Many uses of filler words during a call can hamper smooth progress of the conversation, and also imply that the agent had difficulties to understand the customer's problem or to provide a solution.

- Compliance with Quality Monitoring Scripts
  The features in this category are used to asses if the agent followed the company's scripts and used appropriate verbiage in certain situations such as when greeting the customer, closing the conversation, putting the customer on-hold or transferring the customer to a different agent. Therefore, the features are mostly lexical patterns such as "how can I can help/assist you" for the greeting script, "anything else I can do for you" for the closing script, and "do you mind holding" for the transfer script. "thank you for calling AAA (a company name)" can be used both for the greeting and the closing scripts.

- Agent Attitudes
  The features in this category measure the agent's attitude shown during a call. Some examples of measurable attitudes are if the agent was courteous; if the agent was willing to help the customer; and if the agent showed genuine concern on the customer's problem. Agent attitudes are estimated by looking for certain lexical patterns in the calls such as "see what I can do to resolve", and also by finding cues suggesting that the agent has talked to other people, such as a supervisor or a dealership manager in the automotive company's case, to obtain helps.

- Information Gathering and Sharing
  The features check if the agent has collected required information to handle the request from the customer such as customer's name and address, and the product or service name of interest in the interaction. This category also includes features to check if the agent provided certain information to the customer such as the agent's contact information and the case number

[2] Filler words are words that people often say unconsciously that add no meaning to the communication. Examples of filler words include "umm", "uh", "ah", etc.

for the call. Therefore, the information on the presence of person names, telephone numbers, postal addresses, and product names is used to measure this aspect.

- Problem Resolution
  The features in this category aim at estimating if the customer's request was resolved during the interaction. We exploit linguistic patterns indicating that the problem was resolved or the problem was not resolved. If the customer said "appreciate" at the end of the call, that may indicate the issue was resolved. On the other hand, if the agent and the customer scheduled a "Follow-up" at the end of the interaction, the request was not resolved.

Note that we exploit different types of features, and the features have different ranges of values. For instance, the total length of silences during a call is measured in seconds and often has a large number (hundreds or thousands), but the number of occurrences of a certain expression is usually much smaller. When features have different scales of values, features in greater ranges often dominate those in smaller ranges in machine learning approaches [1]. In this work, we apply a liner scaling method to normalize all feature values into integer values in the [0,10] range using Equation 1.

$$\tilde{x} = \frac{x-l}{u-l} \times 10 \qquad (1)$$

where u is the upper bound and l is the lower bound for a feature x. The equation results in $\tilde{x}$ being in the [0, 10] range.

3.3 Question Answering Estimation

In this work, we apply maximum entropy classification to estimate if a question is satisfied or not during a call. Maximum entropy classification has been successfully used in many natural language processing applications such as topic categorization and named entity recognition [4, 17]. A maximum entropy classifier computes the probabilities of the various outcomes which the model has assigned based on the features using Equation 2, and selects the class with the highest probability.

$$p(c|\vec{x}) = \frac{1}{Z} exp(\sum_{i=1}^{n} \lambda_i f_i(\vec{x}, c)) \qquad (2)$$

where c is a class in the target class set $\mathcal{C}$, $\vec{x}$ is a feature vector representing a call x , Z is a normalizing constant used to ensure that a probability distribution results, λ_i are the parameters of the model, and $f_i(c, \vec{x})$ are indicator functions.

We build a maximum entropy classifier for each question, which makes a binary decision; i.e., if the associated question is satisfied or not-satisfied in a given call. In this work, the class set $\mathcal{C}$ is defined as $\{$*satisfied*, *not-satisfied*$\}$ and $\vec{x}$ is the 20-dimensional feature vector representing a contact center call x as described in Section 3.2. The maximum entropy classifiers were implemented by using the OpenNLP Maxent machine learning package [3].

3.4 Score Estimation

In addition to judging if a question is satisfied, the system generates a score for each question depending on the classification decision. The scores are not the absolute scores assigned to the questions in the company's QM questionnaire. Since we can only estimate if a question is satisfied or not with a certain level of certainty (not at human monitors' level), the scores also represent the likelihood of the call being *good* (or *bad*). In this work, the score for a question is defined as the probability of the question making the given call a *good* call.

Each question is associated with two scores; one score for calls where the question is satisfied and the other score for calls where the question is not satisfied. More specifically, the scores for a questions are defined as the probability of the call being a *good* call when the question is satisfied, $p(good|satisfied)$, and the probability of the call being a *good* call when the question is not satisfied, $p(good|\textit{not-satisfied})$.

The scores are computed in advance by analyzing manual quality monitoring reports. A manual monitoring report typically contains the answers (*yes* or *no*) for all questions and the total score of the call. In this work, we analyzed human-generated QM reports and calculated the two probability value for each question in the QM questionnaire.

For a new call, the QA system first decides if a question is satisfied or not by running the Maxent classifier corresponding to the question. If the question is satisfied, $p(good|satisfied)$ is returned as the score for the question. Otherwise, $p(good|\textit{not-satisfied})$ is used. Finally, the scores for all questions are combined to generate the total score for the call which are in the range of [0, 100]. If the total score is above the predefined threshold for "meet or exceed expectation", the call is regarded as a *good* call. If the total score is smaller or equal to the threshold, the call is regarded as a *bad* call.

4. EXPERIMENTAL SETUP

The call quality monitoring system was developed with a small set of customer calls to four contact centers of an automotive company, which were made available for the research. In this section, we present the characteristics of the experimental data and also briefly describe a Proof of Concept test conducted in one of the four contact centers.

4.1 Experimental Data

4.1.1 Experimental data for question answering estimation

Call monitoring process primarily concerns in-bound customer calls [3]. Our experiments were thus conducted with customer in-bound calls to the automotive company. For the development and testing of the CAB system, we recorded customer calls to four different contact centers located in the United States for about two month time period, resulting in 76,460 calls. These calls concern a wide range of customer issues including inquiries on recalls or reimbursements, questions related to car defects, and complaints on vehicles or dealerships.

For learning the models for question answering estimation, we need associated manual monitoring results in addition to call transcripts. As a separate step, we extracted the manual monitoring reports for the calls to the same contact centers during the same period of time, resulting in 3,989 QM reports. The manual monitoring reports contain the scores for individual questions as well as the total quality score of the calls. We then matched the manual monitoring reports with the call recordings based on the time the call was taken and

[3] calls from customers to a contact center

the agent name who handled the call, and finally identified 387 call transcripts that have manual monitoring reports.

Detailed characteristics of the experimental data are shown in Table 2.

Number of Calls	Total Call Length	Number of Utterances	Number of Tokens
387	60.1 hours	29,731	392,000

Table 2: Detailed size information of the experimental data

4.1.2 *Experimental data for score estimation*

Score estimation intends to compute the degree of a question's contribution for a call being *good*. For this purpose, we don't need to have matching manual monitoring reports and call transcripts. It can be done only with manual monitoring reports if the reports contain the total quality scores (to judge if the call is a *good* call or a *bad* call), and answers for individual questions (to judge if the question was satisfied). In this work, we obtained 700 manual quality monitoring reports, which is a superset of the manual monitoring reports included in the experimental data set described in Table 2.

As described in Section 3.4, we generated two probability values as the score for each question by analyzing the 700 reports. More formally, for a question q_i, we compute the probability of the call being *good* when the question is satisfied, $p_i(good|satisfied)$, and the probability of the call being *good* when the question is not satisfied, $p_i(good|not-satisfied)$.

4.2 Proof of Concept Test

We conducted a Proof of Concept (PoC) test for the CAB system at one of the automotive company's contact centers. Eight contact center agents and two supervisors participated in the PoC to evaluate various pieces of the CAB system including the call quality monitoring system. For the PoC test, we used a different set of calls from the experimental data set described in Section 4.1 to judge the feasibility of the CAB system in a more objective way. The calls were recorded from the same four contact centers but during a different period of time. Therefore, the test calls are likely more dissimilar to the training data than the experimental data set.

The evaluation of the call monitoring system was done by the two supervisors. The supervisors listened to 108 recorded calls in total, and evaluated them with respect to the company's QM questionnaire. Due to time and technical restrictions during the PoC test, the supervisors were not able to evaluate 9 questions out of the 24 questions in the company's questionnaire. The 9 questions check if the agents documented required information accurately in the contact center's database. During the PoC, the supervisors didn't have access to the database and thus these questions were not evaluated. The 9 questions amount to 15 points in total, and therefore the supervisors' scores range from 0 to 85 points. For the purpose of performance evaluation, we normalized the supervisors's scores into 0 to 100 points.

5. EXPERIMENTAL RESULTS AND PERFORMANCE EVALUATION

We have conducted several experiments including formal performance evaluation and a Proof of Concept (PoC) test by real customers in a contact center. In this section, we discuss the experimental results and show the performance of the proposed system in several different perspectives.

5.1 Classification Accuracy

In this experiment, we divided the experimental data set into a training set and a test set containing 310 (80%) calls and 77 (20%) calls respectively, and measured overall accuracy of the system in classifying calls into *good* calls and *bad* calls. The system shows 72.7% classification accuracy.

In the test data set, 54 out of 77 calls were rated as "meet or exceed expectation" (i.e., *good*), and 23 calls were evaluated as "below expectation" (i.e., *bad*) by human monitors. A baseline system which classifies all calls into the dominant class (i.e., *good*) would have 70% classification accuracy. Note that the experimental data is very small and highly biased, which are major obstacles of a machine learning-based approach. Nonetheless, the experiment shows that our system outperforms the baseline system by 2.7 points which is very promising.

The primary purpose of quality monitoring process is to identify calls which may need agent retraining or additional coaching to enhance customer satisfaction. In addition to overall classification accuracy, we measured the effectiveness of this system in selecting *bad* calls. For this purpose, we sorted the 77 test call set by the order of total score, and counted *bad* calls in the bottom 20% of the call set (i.e., 15 calls). The experiment showed that 12 calls out of the 15 calls were rated as *bad* calls by human monitors. The result shows that the system achieved 80% accuracy and generated 2.7 times more *bad* calls in the bottom 20% than random sampling.

5.2 Comparison with a Global Classification Approach

An alternative approach would be using a binary classification method which estimates if a given call is a *good* call or a *bad* call without looking into individual questions. In this experiment, we compare our proposed method with the global classification approach.

We built a maximum entropy classifier using the same set of training data and the same feature set as used in building the presented system. At the training phase, if a call's total quality score is above the predefined threshold, the call is regarded as a *good* call. If the total score is below or equal to the threshold, the call is included in the *bad* call set. In this experiment, therefore, $\mathcal{C} = \{good, bad\}$ and $\vec{x}$ is the 20-dimensional feature vector representing a contact center call x as described in Section 3.2. The global classifier achieved 71.5% classification accuracy when tested with the 77 test data set showing that the individual question answering estimation outperforms the global estimation method by 1.2 points.

Other differences to note between the two approaches are the followings. First, the global classification approach does not know which questions are satisfied in the given call. Note that, QM questionnaires are not a list of independent questions, but consist of a set of categories of questions which are important metrics for judging specific aspects of call quality such as "customer satisfaction", "documentation requirement" and "accuracy of answers". Knowing which questions were satisfied (or not-satisfied) can help contact center management identify the areas where agents often fail and enable them to plan more appropriate coaching for agents. Second, a binary classification approach does not generate scores for monitored calls. The scores can provide more detailed picture on the call quality of the contact center operation.

5.3 Comparison with the Results of a Proof-of-Concept Test

Due to the limitations during the PoC test mentioned in Section 4.1, we are not able to directly compare the supervisors' scores with the scores generated by our system. Instead, we normalized the supervisors' scores from an 85-point scale to a 100-point scale, and categorized the calls into *good* calls and *bad* calls. We then compared the classification accuracy of our system with respect to the human monitoring results. Our system showed 60% classification accuracy on average for the 108 calls, which is much lower than 72.7% achieved with the initial experimental data. The reasons for the performance degradation would be (1) the arbitrary normalization of the human monitoring scores; and (2) the increased dissimilarity of the test calls with the training data.

We learned that the two supervisors have different levels of experience, so it would be interesting to see how the accuracy of the automatic quality monitoring results differ from human monitors with different levels of experience. We conducted performance comparison with each human monitor to find out if human monitors' experience level makes any difference. The results are summarized in Table 3. As we can see from the table, the automatic system shows higher degree of agreement with the more experienced human monitor.

	Less Experienced Evaluator	More Experienced Evaluator
Number of calls	84	24
Classification Accuracy	56%	67%

Table 3: Classification accuracy with respect to the human monitoring results by two human monitors with different levels of experience in call monitoring. The automatic call monitoring system shows a higher degree of agreement with the more experienced human monitor.

5.4 Comparison of Automatic Scores and Manual Scores

Performance measurements in the previous sections focused on the accuracy of the proposed method for relatively small size of test data sets. Note that one of the main advantages of an automatic QM system is that it can monitor all calls in a contact center with no additional cost. Therefore, it would be interesting to see how well automatic scores simulate manual scores. In this experiment, we measure the relationship between manually generated QM scores and automatically generated QM scores by analyzing a large number of call transcripts and manual monitoring reports. The call transcripts and manual monitoring reports are not matched for this study.

The target data set for this study is the 76,460 automatically transcribed calls and 3,989 manual quality monitoring results that we obtained for the development of the CAB system as described in Section 4.1. The automatic quality monitoring system first processed all 76,460 call transcripts and generated QM scores. We then computed the mean and standard deviation values of the two sets of QM scores, and also examined the distribution of the QM scores between the two call categories.

Table 4 shows detailed characteristics of the two data sets and also summarizes the experimental results. As we can see from the table, the automatically generated QM scores and manual QM scores have very similar mean score and score distribution. It is worth noting that both the automatic quality monitoring and manual monitoring have the mean quality scores of around 85, which is the company's threshold value for separating *good* calls and *bad* calls. In addition, the categorization results by the automatic monitoring and the manual monitoring exhibit almost same call distribution: 73% calls being *good* and 27% calls beng *bad*. Based on these observations, we can conclude that the system simulates human monitoring very well.

6. RELATED WORK

Customer and agent conversations are a valuable source of insights into the contact center operations and also the company's overall business. For instance, deep analysis of such conversations can enable estimating customer satisfaction, identifying up-sell/cross-sell opportunities and monitoring contact center performance. Recently, text analytics on contact center calls have gained much attention from researchers. However, until now, much of the effort applying natural language processing (NLP) and speech recognition technologies in this domain has focused on automatic call routing through an interactive voice response system or word spotting [7, 18], call topic classification based on a predefined domain taxonomy [6, 10, 19, 21, 22], and information retrieval from contact center conversations [13, 14].

Recently, there have been strong demand for automatic tools for quality control because call center managers are only able to listen to a small number of calls. Several tools for quality assurance and business intelligence were developed using word spotting technique [2]. Word spotting is a speech recognition technique which recognizes certain words in a predefined vocabulary list in an unconstrained speech [23, 12]. Word spotting has been widely used for information retrieval and topic identification of speech or video data [9, 24]. However, word spotting-based tools can identify only the words in the keyword list, and don't take the contexts into account. While those tools can identify calls where a certain keyword was mentioned, the tools are not able to judge the quality of calls.

Zweig *et al.* presented an automated system for assigning quality scores to call center conversations [25]. To our knowledge, this system is the first fully automated tool for estimating call quality. The work has explored two different approaches for automated quality monitoring; simple pattern matching-based question answering and maximum entropy classification. The question answering approach is an extended word spotting. They pre-compiled one or two words and phrases for each question, and looked for the words and phrases in a call transcript. If a word is present in the call transcript, the question is regarded as satisfied. For instance, if "thank you for calling" or "anything else" is present in a call transcript, the system judges the associ-

		Automatic QM Score	Manual QM Score
Number of calls		76,460	3,989
Number of agents who handled the calls		453	623
Number of human monitors participated		0	91
Average number of calls per agent		168.8	6.4
Mean quality score		86.3	83.5
Standard deviation		17.6	22.9
Score distribution	*Good* calls	55,572 (72.7%)	2,929 (73.4%)
	Bad calls	20,888 (27.3%)	1,060 (26.6%)

Table 4: Comparison results between automatically generated QM scores and human generated QM scores. The automatic QM scores and manual QM scores have very similar mean value and score distribution showing that about 73% calls are judged *good* both by the machine and by human monitors. The experiment reveals that the automatic system simulates human monitoring very well.

ated question "did the agent follow the appropriate closing script?" is satisfied. The maximum entropy-based approach, on the other hand, determines if a call is "bad" based on a set of ASR-derived features and precompiled n-gram word sequences. The system then interpolates the score generated by the question-answering method and the probability of a call being "bad" as determined by the maximum entropy classifier and generates a quality score for a call.

Similarly to [25], we apply maximum entropy classification and use ASR-derived features such as "number of long silences" and the occurrence of certain expressions in a call. However, our system is different from [25] for the followings. First, in addition to the ASR-derived features and lexical features, we exploit high level contextual and topical features which go beyond lexical pattern matching. Second, in [25], calls are regarded as bad if they belong to the bottom 20% in the sorted list of the calls. Therefore, the definition of being "bad" changes depending on the training data set. In our work, on the other hand, a call is regarded as bad if the call's quality score is below the company's predefined threshold as done in human monitoring.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we presented an automatic call quality monitoring system for contact centers, which analyzes call transcripts generated by an automatic speech recognition system and decides if a given call is a *bad* call or a *good* call, depending on how well the call complies with the company's quality control guidelines. Specifically, the system applies various natural language processing techniques for feature extraction and machine learning methods for call categorization.

Quality monitoring requires human-level intelligence and knowledge, and not all information needed to answer the questions are present in call transcripts. We selected the features which are good indicators for call quality and can automatically extractable from call transcripts. The features include general conversational features such as the the number of long pauses and the total length of pauses during a call, lexical features such as words or phrases which are commonly used in certain situations, and high-level structural features such as call segments. We then apply maximum entropy classification to estimate if questions in a QM questionnaire are satisfied or not based on the feature set. The probability of the call being *good* when the question is satisfied (or not-satisfied) is used as the score for the question. The scores for all questions are combined to generate a total quality score for the call. If the total score is above a predetermined threshold, the call is regarded as a *good* call. If the score is equal to or below the threshold, the call is considered as a *bad* call.

We have conducted several experiments with customer calls to an automotive company. First, we obtained 387 call recordings with associated manual quality monitoring results. The system was trained on 310 calls (80% of the experimental data) and tested on the remaining 77 calls (20% of data). The experimental result shows 72.7% classification accuracy, which is very promising given the fact that the system was trained with a very small and highly biased data set. It also showed that the automatic system identified 2.7 times more bad calls than random selection in the bottom 20%. This indicates that the system can sample more appropriate calls for human monitoring and thus can reduce the number of calls requiring human monitoring, which subsequently results in cost reduction for contact centers.

Furthermore, the comparison of about 76,500 automatic quality scores and 4,000 manual scores showed that the automatic system simulates the human monitoring very closely exhibiting similar mean value and class distribution. This confirms that the automatic call quality monitoring system can predict overall call quality trend in a contact center, which is very beneficial for contact center operation.

A limitation of the work is that the system was developed with a very small size of training data. In the future, we plan to obtain a bigger and more balanced set of annotated calls to improve the performance of the system. Also, we would like to apply the system in a different domain and measure if the system is widely applicable across different industries without much customization.

8. ACKNOWLEDGEMENT

The author is very grateful to the following colleagues: Roy J Byrd, Keh-Shin Cheng, Stephen Gates, Mary Neff and Wilfried Teiken for their collaboration in building the CAB system, and especially to Wilfried Teiken for producing a call recording solution; Karthik Visweswariah for retraining the ASR system and for generating call transcripts; and Scott Butcher for providing detailed information on the automotive company's quality monitoring process. The author also would like to thank Olivier Siohan and Bhuvana Ramabhadran, co-authors of [25], for many helpful discussions on their work.

9. REFERENCES

[1] S. Aksoy and R. M. Haralick. Feature normalization and likelihood-based similarity measures for image retreval, 2001.

[2] G. Alon. Key-word spotting–the base technology for speech analytics. Natural Speech Communication Ltd., White Paper, 2005.

[3] J. Baldridge. The opennlp project. http://opennlp.sourceforge.net/, 2002.

[4] A. Berger, S. D. Pietra, and V. D. Pietra. A maximum entropy approach to natural language processing. *Computational Linguisitics*, 22(1), 1996.

[5] B. Boguraev. Annotations-based finite state processing in a large scale nlp architecture. In N. Nicolov, K. Boncheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing III, Selected papers from RANLP 2003*, Current Issues in Linguistic Theory (CILT). John Benjamins, Amsterdam/Philadelphia, 2004.

[6] 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*, pages 158–165, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

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

[8] D. Ferrucci and A. Lally. Uima: An architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3):476–489, 2004.

[9] J. Foote, G. Jones, K. Sparck, and S. Young. Talker-independent keyword spotting for information retrieval. In *Proceedings of Eurospeech 95*, volume 3, pages 2145–2148, 1995.

[10] 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.

[11] J. Hobbs, D. Appelt, J. Bear, D. Israel, M. Kameyama, M. Stickel, and M. Tyson. Fastus:a cascaded finite-state transducer for extracting information from natural-language text. In E. Roche and Y. Schabes, editors, *Finite-State Language Processing*, pages 383–406. The MIT Press, 1997.

[12] J. Junkawitsch, L. Neubauer, H. Höge, and G. Ruske. A new keyword spotting algorithm with pre-calculated optimal thresholds. In *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, volume 4, pages 2067–2070, 1996.

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

[14] 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.

[15] Y. Park. Automatic call section segmentation for contact-center calls. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 2007.

[16] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig. fmpe: Discriminatively trained features for speech recognition. In *Proceedings of ICASSP*, pages 961–964, 2005.

[17] A. Ratnaparkhi. Maximum entropy models for natural language ambiguity resolution. In *Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.*, 1998.

[18] 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.

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

[20] H. Soltau, B. Kingsbury, L. Mangu, D. Povey, G. Saon, and G. Zweig. The ibm 2004 conversational telephony system for rich transcription. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2004.

[21] 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.

[22] G. Tur, D. Hakkani-Tur, A. Stolcke, and E. Shriberg. Integrating prosodic and lexical cues for automatic topic segmentation. *Computational Linguistics*, (27):31–57, 2001.

[23] J. G. Wilpon, L. R. Rabiner, C.-H. Lee, and E. R. Goldman. Automatic recognition of keywords in uncontrained speech using hidden markov models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1990.

[24] Y. Yamashita, T. Tsunekawa, and R. Mizoguchi. Topic recognition for news speech based on keyword spotting. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP)*, 1998.

[25] 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.