Extracting, Ranking, and Evaluating Quality Features of Web Services through User Review Sentiment Analysis

Xumin Liu[1], Arpeet Kale[1], Javed Wasani[1], Chen Ding[2], and Qi Yu[1]
[1] Golisano College of Computing and Information Science, Rochester Institute of Technology, USA
[2] Department of Computer Science, Ryerson University, Canada

***Abstract*—Quality of Service (QoS) has become a standard way of evaluating web services and selecting the one that suites user interests the best. Traditional methods adopt a fixed set of QoS parameters and typical ones include response time, fee, and availability. There currently lacks an effective way of identifying quality features that users are actually interested in when choosing a service. Meanwhile, the traditional way of collecting QoS values relies on either public information released by service providers or test results from repeatedly invoking a service. Therefore, the values can be heavily affected by authenticity of the provider offered information or the quality/configuration of the test code/environment. As a result, existing QoS evaluation methods are not applicable to subject features, such as usability and affordability, where the values depend on user personal judgement. In this paper, we propose a novel approach to extracting domain-related QoS features, ranking those features based on their interestingness, evaluating the value of these features through sentiment analysis on user reviews. More specifically, we leverage natural language processing techniques and machine learning approaches to identify top QoS features that users are interested in and simultaneously learn their sentiment orientation towards those features. We model the problem as sentiment classification, where relevant terms in a review are modeled as features that determine whether a review is positive or negative. Logistic regression is used so that the impact of these terms are learned simultaneously when the classifier is learned through a supervised learning process. The nontrivial terms are selected as the candidate QoS featured. A comprehensive experiment has been conducted on a real-world dataset and the result demonstrates the effectiveness of our approach.**

***Keywords*-QoS; web services; sentiment analysis; logistic regression; natural language processing; supervised learning**

I. Introduction

Quality of Service (QoS) has been widely used as a standard way to model and evaluate the non-functional features of a web service. Typical QoS features include reliability, response time, security, and invocation fee. QoS plays an essential role in various web service management tasks, such as selecting a service that fulfills both the functional and non-functional requirement specified by a user. It also serves as the key criterion to differentiate web services that provide similar functionality. As a result, many QoS-aware or QoS-based approaches have been proposed in the field of service computing, such as QoS-aware service discovery and selection [16], composition [2], recommendation [15], and provisioning [8].

Due to the importance of QoS, many research efforts have been conducted over these years centering around QoS collection [19], QoS monitoring [13], [11], QoS prediction [1], [12], QoS evaluation for composite services [18], [6], and QoS management [17], [10]. QoS collection efforts focus on testing web service QoS features in a large number of web services under various testing environments and configurations from a third party. QoS monitoring efforts focus on providing declarative methods for users to specify monitoring requirement and automating the QoS testing process. QoS prediction efforts focus on predicting the QoS values based on various information, such as time, location, system throughput, and historical data. QoS evaluation for composite services focus on an effective way to integrate QoS values of the component services. QoS management efforts focus on detecting the inconsistency between the delivered QoS and the ones defined in Service Level Agreement (SLA). The limitations of these efforts are explained below.

Current QoS approaches use the QoS features (or parameters), which were primarily determined by domain experts. Those parameters can be domain-independent, such as availability, security, cost, and reliability, as well as domain specific, such as latency for weather services and accuracy for traffic services. The evaluation of QoS features mainly rely on two resources. First, some web service providers make related information, such as the security level and invocation fee, available to users. Second, some users or third party agents may run tests and collect QoS values, such as for availability and reliability. Although these approaches have been increasingly adopted, they still have their limitations. First, the predefined QoS features may not always reflect what users are interested in. For example, users may care about if a service is always compatible to the previous version when updated, so that no versioning issue will occur once the service is included in a software package. However, this concern may not be foreseen by domain experts when determining the QoS list. Second, it is limited to rely on the QoS information published by service providers, which may be misleading or unauthentic, or rely on the testing result in a particular time period or

978-1-4673-7272-5/15 $31.00 © 2015 IEEE
DOI 10.1109/ICWS.2015.30

in a particular geographic area, as the QoS values can vary under different temporal and/or spatial settings.

To address the above limitations, we propose to analyze the reviews made on web services by their users and extract QoS features that users are truly interested in. Different from traditional methods, where the QoS values are quantified, we will learn users' sentimental orientation towards a QoS feature, i.e., whether the review is positive or negative. For example, instead of getting the actual value of response time, we will learn whether users feel that the service responds fast or slow to a request. As another example, instead of getting the actual invocation fee, we will learn whether user feel that it is expensive, cheap, or free to invoke a service. In this way, the proposed approach provides more interpretable evaluation result than actual numeric values to users.

Combining QoS feature learning and sentiment analysis together offers a number of key benefits. First, it is not always that the frequently mentioned features in user reviews are the ones that are related to the quality of services. For example, users may first describe a web service before commenting it, such as *"It is an online service that provides hotel reservation. It offers good price"* and *"You can buy apple products through this API. The payment is convenient"*. Here, *service* and *product* may frequent terms in users reviews but they are irrelevant to evaluate a web service. Through analyzing user sentiment orientation towards a web service, we can extract those features that play a key role in the sentiment orientation, such as *price* and *security level*, which can be used as the candidates of QoS features. Second, we can evaluate how much these features contribute to user's sentiment orientation, which reflects how much users take interest in those features when evaluating a web service, i.e., interestingness of a QoS feature. For example, users may comment their positive or negative opinions more on price than on security level. This means that price is a more important or concerned feature of that web service than security. Third, through sentiment analysis, we can learn user sentiment orientation toward a QoS feature of a web service. The evaluation of a web service on this feature can be conducted by integrating the reviews of all users.

In this paper, we propose a novel approach for QoS feature extraction and evaluation through sentiment classification over user reviews on web services. Each review will be classified into either *positive* and *negative*. We exploit an augmented logistic regression, referred to l_1 regularized logistic regression, to perform user review classification. Meanwhile, l_1 regularized logistic regression has the effect of forcing some coefficients of the model to be exactly zero. As a result, QoS feature selection can be conveniently conducted by directly choosing the QoS related features with nonzero coefficients. Using this method, learning, ranking, and evaluating QoS features can be conducted simultaneously. To our best knowledge, this is the first attempt to combine QoS feature learning and sentimental analysis together to identify, rank, and evaluate a web service's QoS feature. It is worth to know that our approach is not supposed to replace, but instead complement the current QoS modeling, monitoring, and prediction methods. It takes a different perspective and utilizes the rich information resources available on online API discussion forums, where actual users' opinions and experiences are shared.

The remainder of this paper is organized as follows. In Section II, we give an overview of the process of learning QoS features of web services from user reviews. In Section III, we present a learning method where we model the problem as a sentiment classification and use l_1 regularized logistic regression to extract QoS related features. In Section IV, we present our experimental study performed on a real world dataset. In Section V, we review some representative work that is related to ours and highlight the difference. In Section VI, we conclude our work and briefly discuss the further work.

II. Learning QoS features from User Reviews: An Overview

In this section, we give an overview of the process about learning QoS features through sentiment analysis on user reviews.

As shown in Figure 1, there are many web portals that solicit feedback from end users on the usage of web services and other online services. The input from users forms a large user review repository. Each review is a free text description, which not only reveals user opinion on the service but also implies the sentiment orientation: positive or negative. Some examples of positive reviews are: *"Communication is great! High quality products at a very fair price. You can find the 800 customer service number on the shipping slip."* An example of negative reviews can be: *"None of the coupon codes would work!"*

The entire learning process consists of two phases: data preprocessing and sentiment analysis. During the data preprocessing phase, the reviews are extracted from the repository and processed using the following steps:

1) *Review cleaning*: Each review is evaluated and will be filtered out if it is not valid. Invalid reviews refer to those that are empty, have too few words, or mainly provide commercial links. The validation criteria include the number of words in each review and the ratio of texts to URLs. Furthermore, irrelevant information, such as URLs, the tagged usernames, and hashtags (if the review is a tweet), is also removed. Regular expression can be used to identify this type of information.
2) *Word tokenizing and stemming:* Each review will be tokenized into a bag of words, which will then be stemmed to their root forms, such as from *stopped* to *stop*.

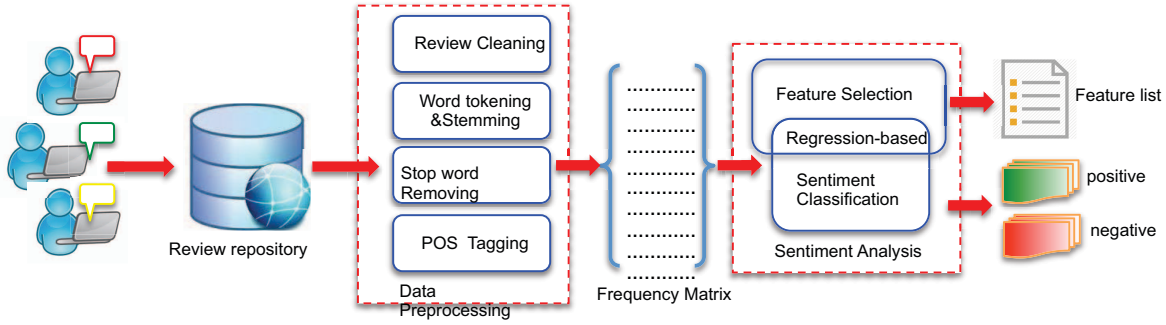

Figure 1. The Process of Learning QoS Features from User Reviews Through Sentiment Analysis

3) *Stop word removing:* Stop words, such as *and* and *the*, are irrelevant to feature extraction and sentiment classification. Therefore, they will be removed from each review to improve the efficiency and accuracy of the learning result.
4) *POS tagging:* Part Of Speech Tagging (POS Tagging), which is a natural language processing (NLP) technique, will be exploited to parse a review and assign part of speech to each word, such as verb, noun, adjective, and so on. For example, given the review *"Google maps APIs are comprehensive and amazing"*, the POS tagging result is *"(NNP Google) (NNS maps) (NN APIs) (VBP are) (JJ comprehensive) (CC and) (JJ amazing)*. The parts of speech in this reviews are: NNP (proper noun), NNS (plural noun), NN (Noun), VBP(verb, past participle), JJ (adjective), and CC (conjunction). After applying POS tagging, only nouns and adjectives are kept since they are potential QoS features and their descriptors, respectively. They both play a key role in the sentiment orientation of the review. By removing other terms, we further improve the efficiency of the learning result.

After the data preprocessing phase, suppose there are M reviews and N distinct terms left. The terms are the mixture of nouns and adjectives. We can generate a $M \times N$ frequency matrix, F, where each row represents a review and each column represents a term. $F_{i,j}$ is the normalized frequency of the j-th term in the i-th review, given by

$$F_{i,j} = \frac{count(t_j)}{\max_{1 \le k \le N} count(t_k)}$$

The frequency matrix will be used as the input of the second phase: sentiment analysis. The sentiment analysis is an integrated process that combines feature selection as well as sentiment orientation classification. The result of this phase is twofold. First, a learning model will be generated to determine if a review is positive or negative. Second, a list of features will be generated, where each feature represents a factor that users consider when evaluating the quality of the service. The features are ranked based on the impact they have on user evaluation. The higher a feature is ranked, the more important or interesting it is for users. For this purpose, we propose to model the sentiment analysis problem using a logistic regression model. The terms, including the nouns and adjectives, are treated as the features of reviews, which determine the sentimental orientation of the review. The coefficients indicate the importance of each feature on the sentiment classification result and consequently important QoS related features can be conveniently identified. The process of sentiment analysis is elaborated in Section III.

III. l_1 Regularized Logistic Regression for QoS Feature Extraction

We present our l_1 regularized logistic regression (or l_1-RLR for short) based approach for QoS feature extraction. Logistic regression is one type of probabilistic discriminative model that is widely used to solve classification problems. In this section, we start by describing the basic logistic regression model for QoS related sentiment analysis and how it can used to extract QoS features based on the p-values of the coefficients in the model. We then introduce a non-trivial extension of logistic regression, which is achieved through l_1 regularization. The resultant l_1 regularized logistic regression (or l_1-RLR) can not only improve the accuracy in sentiment analysis but also offers an elegant way to directly extract important QoS features.

A. Logistic Regression for QoS based Sentiment Analysis

Consider a set of comments with positive and negative sentiment orientations $\{\mathbf{f}_m, t_m\}$, where $\mathbf{f}_m \in R^N$ represents a term vector, $t_m \in \{0, 1\}$, and $m = 1, ..., M$. Let 1 and 0 represent positive and negative sentiment orientations, respectively. The posterior probability of the positive class $\mathcal{P}$ is given by applying a logistic sigmoid function on a linear function of the term vector $\mathbf{f}$

$$p(\mathcal{P}|\mathbf{f}) = \sigma(\boldsymbol{\beta}^T \mathbf{f}) = \frac{1}{1 + \exp(-\boldsymbol{\beta}^T \mathbf{f})} \quad (1)$$

where $\boldsymbol{\beta} \in R^N$ is the coefficient vector of the logistic regression model and $\sigma(\cdot)$ is the sigmoid function, given by the

second equality. It is easy to see that the posterior probability of the negative class $\mathcal{N}$ is given by $p(\mathcal{N}|\mathbf{f}) = 1 - p(\mathcal{P}|\mathbf{f})$.

The logistic regression model is essentially specified by the coefficient vector $\boldsymbol{\beta}$, which is typically determined through maximum likelihood. Specifically, given the training set of comments $\{\mathbf{f}_m, t_m\}$, the likelihood function is given by

$$\begin{aligned} p(\mathbf{t}|\boldsymbol{\beta}) &= \prod_{m:t_m=1} y_m \prod_{m':t_{m'}=0} (1 - y_{m'}) \\ &= \prod_{m=1}^{M} y_m^{t_m} (1 - y_m)^{1-t_m} \end{aligned} \tag{2}$$

where $y_m = p(\mathcal{P}|\mathbf{f}_m)$ and $\mathbf{t} = \{t_1, ..., t_M\}^T$. Instead of directly maximizing $p(\mathbf{t}|\boldsymbol{\beta})$, it is usually more convenient to minimize an error function, which is achieved by taking the negative logarithm of the likelihood:

$$-\log p(\mathbf{t}|\boldsymbol{\beta}) = \sum_{m=1}^{M} \{t_m \log y_m + (1 - t_m) \log(1 - y_m)\} \tag{3}$$

where $-\log p(\mathbf{t}|\boldsymbol{\beta})$ is commonly known as the *cross-entropy* error. So the logistic regression model is to seek a coefficient vector $\boldsymbol{\beta}$ that minimize the *cross-entropy* error. To ensure that such a $\boldsymbol{\beta}$ exists, we first compute the second order derivative, i.e., the Hessian, of $-\log p(\mathbf{t}|\boldsymbol{\beta})$, which gives

$$\begin{aligned} \mathbf{H} = \nabla\nabla(-\log p(\mathbf{t}|\boldsymbol{\beta})) &= \sum_{m=1}^{M} y_m(1 - y_m)\mathbf{f}_m\mathbf{f}_m^T \\ &= F^T R F \end{aligned} \tag{4}$$

where

$$R = \text{diag}\{y_1(1 - y_1), ..., y_m(1 - y_m)\} \tag{5}$$

Since $0 \leq y_m \leq 1$, it can be shown that $\mathbf{x}H\mathbf{x}^T > 0$ for any nonzero vector $\mathbf{x}$. In another word, $\mathbf{H}$ is positive definite and hence it is guaranteed that there exists a vector $\boldsymbol{\beta}$ that is a unique minimum of $-\log p(\mathbf{t}|\boldsymbol{\beta})$.

The optimal coefficient vector $\boldsymbol{\beta}$ can be efficiently computed using the Newton-Raphson method, which is an iterative optimization scheme that uses a local quadratic approximation to the cross-entropy error to find a step direction, given by

$$d^{(k)} = \boldsymbol{\beta}^{(k)} - \mathbf{H}^{-1}\nabla(-\log p(\mathbf{t}|\boldsymbol{\beta})) \tag{6}$$

where $\boldsymbol{\beta}^{(k)}$ is the coefficient vector computed in the k-th iteration. Once the step direction is determined, the coefficient vector will be updated as

$$\boldsymbol{\beta}^{(k+1)} = (1 - s)\boldsymbol{\beta}^{(k)} + sd^{(k)} \tag{7}$$

where s is the step size parameter.

B. QoS Feature Extraction using Logistic Regression

One key reason of choosing logistic regression over other classification algorithms (e.g., Naive Bayes and SVM) to perform QoS based sentiment analysis is that the model also provides key information for QoS feature extraction. This can be made clear by checking the *log-odds* or *logit* of logistic regression:

$$\log\left(\frac{p(\mathcal{P}|\mathbf{f})}{1 - p(\mathcal{P}|\mathbf{f})}\right) = \boldsymbol{\beta}^T\mathbf{f} \tag{8}$$

Since the logit of logistic regression is a linear function of the term vector $\mathbf{f}$, the importance of the terms (especially QoS related ones) can be evaluated by checking the coefficient vector $\boldsymbol{\beta}$. Intuitively, important QoS features should be the ones that play an important role in determining the sentiment orientations of comments. Therefore, their corresponding coefficient should not be zero (because otherwise these features will not help determine the sentiment orientations). Consequently, we perform the test of null hypothesis, which can be achieved by computing the *standard score* of the coefficients, given by $z = (\beta_j - 0)/SE(\beta_j)$, where $SE(\beta_j)$ is the standard error of the j-th coefficient. In essence, a large z-score implies that the coefficient is far away from 0. A corresponding probability value, referred to as *p-value*, can be computed indicating the probability of the coefficient being zero. Therefore, QoS feature extraction can be achieved by selecting terms whose coefficients have small p-values.

C. l_1-RLR for QoS Feature Extraction

When performing sentiment analysis over reviews, each distinct term in the reviews becomes a feature. This will result in a logistic regression model with a high dimensional feature space where there may be a lot of variability in the maximum likelihood fit. As a result, the model is more likely to overfit the training data, leading to a poor predictive power, which is further confirmed by our experimental study. Besides, a large feature space may cause another less obvious but equally important issue, the selection of irrelevant features. This is because, as the number of features increase, it is more likely to observe some features which take a small p-value by chance. As a concrete example, given 100 features, around 5% p-values associated with the features will be smaller than 0.05 by chance.

A commonly used approach to reducing the high variance of statistical models is regularization, which constrains or shrinks the coefficients so that the variance of the model can be significantly reduced and consequently the overall error rate can be reduced. We propose to apply l_1 regularization to logistic regression, leading to l_1 regularized logistic regression model or l_1-RLR, given by

$$-\log p(\mathbf{t}|\boldsymbol{\beta}) + \lambda||\boldsymbol{\beta}||_1 \tag{9}$$

where $||\boldsymbol{\beta}||_1 = \sum_{j=1}^{N} |\beta_j|$ is the l_1 norm of the coefficient vector $\boldsymbol{\beta}$, and $\lambda > 0$ is a regularization parameter. It is worth to note that when $\lambda = 0$, l_1-RLR reduces to the basic logistic regression. The reason that we choose l_1 regularization over other regularization strategies (e.g., l_2 regularization) is that the l_1 penalty has the effect of forcing some coefficients to be exactly zero. As a result, QoS feature selection can be conveniently conducted with the l_1-RLR by directly choosing the features with nonzero coefficients.

Computing the coefficient vector $\boldsymbol{\beta}$ in l_1-RLR requires minimizing the objective function in (9). This is more involved than minimizing the cross-entropy error itself in logistic regression because of nonsmooth l_1 term. However, efficient algorithms (e.g., LARS [3]) exist that can solve a slightly different problem than (9), where the cross-entropy error term is replaced by a quadratic function of $\boldsymbol{\beta}$. Furthermore, it has been observed in [7] that while minimizing the cross-entropy error in logistic regression, the step direction used by the Newton-Raphson method in (6) can also be found by minimizing a quadratic function, given by

$$||(R^{\frac{1}{2}}F^T)\boldsymbol{\beta} - R^{\frac{1}{2}}\mathbf{z}||_2^2 \tag{10}$$

where

$$\mathbf{z} = F\boldsymbol{\beta}^{(k)} - R^{-1}(\mathbf{y} - \mathbf{t}) \tag{11}$$

Therefore, the coefficient vector $\boldsymbol{\beta}$ of l_1-RLR can be efficiently computed by integrating LARS into the iterative optimization process, where LARS is used to minimize the following function

$$||(R^{\frac{1}{2}}F^T)\boldsymbol{\beta} - R^{\frac{1}{2}}\mathbf{z}||_2^2 + \lambda||\boldsymbol{\beta}||_1 \tag{12}$$

to find the step direction $d^{(k)}$. Algorithm 1 details the QoS feature selection process using l_1-RLR.

Algorithm 1 l_1-RLR for QoS Feature Extraction

Require: Frequency matrix F
Ensure: a set $\mathcal{S}$ of QoS related features
1: Initialize $\boldsymbol{\beta}^{(0)}$
2: **for all** $k \in [0, maxIter]$ **do**
3: compute R using $\boldsymbol{\beta}^{(k)}$ based on Eq. (5)
4: compute $\mathbf{z}$ using R and $\boldsymbol{\beta}^{(k)}$ based on Eq. (11)
5: use LARS to compute the minimum of Eq. (10), which gives $d^{(k)}$
6: set $\boldsymbol{\beta}^{(k+1)} = (1-s)\boldsymbol{\beta}^{(k)} + sd^{(k)}$
7: **end for**
8: **for all** $j \in [1, N]$ **do**
9: Include feature j where $\boldsymbol{\beta}_j \neq 0$ into the feature set $\mathcal{S}$
10: **end for**

IV. Experimental Study

In this section, we conduct a set of experiments to evaluate the effectiveness of the proposed QoS feature learning process. We collect the experimental dataset by crawling a real-world website for reviewing online services. All experiments have been carried out on a Macbook Pro with 2.6 GHz Core processor and 8GB DDR3 memory under Mac OS X 10.9.5 operating system. The evaluation covers two major aspects. First, we evaluate the accuracy of feature extraction result. We compare the features resulting from our approach with those from the frequency-based feature selection. Second, we evaluate the accuracy of sentiment classification. The classification result of our approach is compared with a number of competitive classification algorithms.

A. Experimental Data Set

We crawled an online service review website, *www.sitejabber.com*, to collect the dataset for our experiments. This website allows users to rate and review online businesses in 18 different domains. Figure 2 shows a review from the website, where the user left the rating and the textural comment on the business.

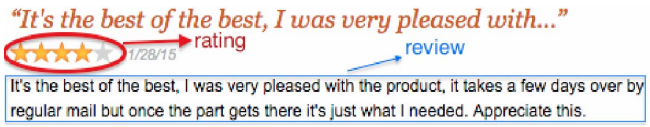

Figure 2. A sample review on Sitejabber

We crawled the user input for three different domains, including *computer*, *business*, *mobile-phones* from the website. We performed data preprocessing and cleaning to filter out invalid input. We noticed that user reviews are not always consistent with the given ratings due to the discrepancy among users on the rating standard. Therefore, instead of using the given ratings, we manually labelled the sentiment orientation for each review. We removed terms that appear no more than 5 times in all the reviews, which results in 698 features (i.e., distinct noun and adjective terms). The detailed information about the dataset is shown in Table I.

Table I
Experimental Data Set

parameter	value
# of services	28
# of reviews	855
# of positive reviews	491
# of negative reviews	364
# of features	698

B. Feature Extraction Evaluation

In order to evaluate the effectiveness of the proposed feature extraction approach, we manually read every review and chose the major quality features mentioned in the reviews as the ground truth. Some top ranked features are described in Table II along with an example review where these features are mentioned.

We use recall@K as the measure to evaluate the accuracy of feature extraction result. That is, given a threshold K, the

Table II
MAJOR QUALITY FEATURES COMMENTED IN USER REVIEWS

Rank	Feature Category	Feature Description	Example Review
1	Attention to customers	How much a company cares about its customers	*"They don't seem to care that millions are coming of age and switching to Vontage, Consumer Cellular, etc., or other AARP rec. wireless.."*
2	Cost	The affordability of the web service or the products they sell.	*"I have purchased a watch and shoes from this website. The price seemed reasonable for me and I ordered for the products."*
3	Delivery time	Whether the delivery is in time or being delayed.	*"Just ordered and got it fast! I would certainly recommend the site.*
4	Handling	Whether the products are handled properly or damaged on the delivery.	*"I received my item with damage already in place."*
5	Security	Whether the online transactions are secured enough.	*''I noticed a mysterious customer service this morning trying to see who used my card and where the purchase was going to."*

top K features are extracted and compared to the ground truth feature set. Recall is calculated as the ratio of the number of collect features in the extraction result (N_E) to the size of the ground truth feature set. (N_G):

$$recall@K = \frac{N_E}{N_G} \tag{13}$$

To demonstrate the effectiveness of using l_1-RLR for QoS feature extract, we include two other approach for comparison: logistic regression and frequency based method. As discussed in Section III, in Logistic regression, we may compute the p-values of the feature coefficients. A significant feature should have a coefficient with a small p-value, implying that probability of the coefficient being zero is small. However, when the feature space becomes large, the accuracy of using p-values to determine the importance of features will decrease (see the discussion in Section III for details). The frequency based method relies on the frequency of the features (i.e., terms) mentioned in the reviews to rank the features [14].

Table III
RECALL COMPARISON AMONG FEATURE EXTRACTION METHODS

Methods	K=20	K=30	K=40	K=50
l_1-RLR	0.355	0.509	0.586	0.913
Logistic Regression	0.125	0.125	0.125	0.125
Frequency based method	0.125	0.125	0.175	0.278

Table III compares the recall@K performance of different approaches by varying K from 20 to 50. l_1-RLR clearly outperforms the other two approaches. The reason why logistic regression does not perform well is mainly due to the large size of the feature space. While a detailed comparison with the frequency based method reveals some interesting findings that further justify the advantage of using l_1-RLR for QoS feature extraction.

More specifically, Table IV shows some top features extracted by both approaches. It can be seen that many features extracted by the frequency based method, such as site, business, and product, are neutral terms that are not directly relevant to the quality aspects of a service. But these terms are frequently mentioned in user reviews and hence extracted. This can be further verified as some terms (e.g., service) are assigned into both positive and negative categories meaning that they are frequently mentioned in both positive and negative reviews. It is also interesting to note that some terms, such as customer service and phone, are identified as negative features by the frequency based method. It may appear that customer service is of low quality but when checking the reviews, we found this is actually not true. In fact, the major reason for this is that customer service and phone are commonly mentioned when users complain other aspects of a service, such as shipping/handling or delivery. But users usually also mention that they have to call customer service to resolve the problem or ask the phone of customer service in the same review. In contrast, the features extracted by l_1-RLR, such as price, cost, damage, and fraud, are more relevant to the quality aspects of services while clearly indicating their sentiment orientations.

Table IV
EXTRACTED FEATURES

Frequency based method				**l_1-RLR**	
Positive		Negative		Positive	Negative
frequency	feature	frequency	feature	feature	feature
114	site	149	service	price	damage
83	service	125	customer	cost	fraud
60	business	78	phone	gift	return
60	product	70	company	deal	credit

C. Sentiment Classification Evaluation

We first evaluated the accuracy of sentiment classification over user reviews. Given a collection of reviews, we compute the accuracy as the percentage of correctly labeled reviews, i.e., the identified label, R_{i_d}, either positive or negative, is the same as the actual label, R_{i_a}:

$$AC = \frac{P_C}{P_A} = \frac{\sum_i I(R_{i_d} = R_{i_a})}{|R|} \tag{14}$$

We also compute the F_1-score since it is widely used for evaluating classification algorithms:

$$F = 2 \times \frac{Precision * Recall}{Precision + Recall} \tag{15}$$

To demonstrate the effectiveness of l_1-RLR for sentiment classification, we compare it to several other classification algorithms, including Logistic Regression, l_2-RLR (l_2 Regularized Logistic Regression), and Support Vector Machine

(SVM) classifier. A 10-fold cross validation is used to compare the classification performance of different algorithms.

Table V
ACCURACY COMPARISON FOR SENTIMENT CLASSIFICATION

Methods	**Accuracy**	F_1**-score**
l_1-RLR	0.753	0.764
l_2-RLR	0.765	0.756
Logistic Regression	0.503	0.695
SVM	0.709	0.725

The results are summarized in Table V. It can be seen that the performance of l_1-RLR is similar to that l_2-RLR and both outperform the other two algorithms. This clearly justifies the effectiveness of using regularization to deal with large feature spaces. l_1-RLR is chosen over l_2-RLR as it performs automatic feature selection by forcing some coefficients to be exactly zero.

V. RELATED WORK

In this section, we give an overview of some representative work that is related to modeling, evaluating, and monitoring QoS features. We differentiate these works from our approach.

A. QoS Collection and Evaluation

In [19], the importance of QoS in web service selection, discovery, recommendation, and composition is acknowledged. A comprehensive QoS evaluation on existing web services was conducted to generate a QoS dataset to benefit the research community of service computing. Over 20 thousands real world web services were tested and the evolution focused on respond time and throughput. In [13], a language was proposed to express QoS monitoring requirement from user perspective. A monitoring request is processed as data steams to cater for its continuous nature. It is evaluated continuously as new testing result is generated. A sliding window is used and the request is responded with the statistical observation during the sliding window. In [11], an approach was proposed to automatically learn and monitor QoS values of web services through Aspect Oriented Programming (AOP). Under AOP, a service stub is generated for each evaluated service and an invocation was performed for the evaluation. This approach focuses on a subset of common QoS features that can be measured through invocation, such as response time, execution time, availability, and accuracy. In [1], a forecasting approach was proposed to predict dynamic QoS values. It leverages both Auto Regressive Integrated Moving Average (ARIMA) and Generalized Auto Regressive Conditional Heteroscedastic (GARCH) to capture the volatility of QoS data and predict the future values.

In sum, these methods focus on evaluating the values of predefined QoS parameters. Our approach is different since we focus on extracting features that are relate to a service's quality and assessing them based on the analysis of user feedback.

B. QoS Evaluation for Composite Services

In [18], an approach was proposed to compute QoS values of a composite service based on the workflow of the service and QoS values of each component service. Instead of using fixed values, this approach models each QoS parameter as a probabilistic distribution to better capture the real world scenarios. In [6], an approach was proposed to integrate QoS values and analyze QoS requirement in reconfigurable web services choreographies. It reconfigures a composite service given a QoS goal, with a major focus on latency, throughput, accuracy, and data quality. These approaches focus on measuring and analyzing QoS for composite services, based on the QoS values of individual services. This is different from our focus.

C. QoS Management

In [17], an approach was proposed to identify those component services that do not deliver the expected QoS values in a business process. It integrates dependency matrix based and Bayesian network based diagnosis to reduce diagnose cost and improve the accuracy. In [10], an approach was proposed to realize a soft probabilistic contract for QoS management in a composite web service. Instead of enforcing a hard constraint on QoS values in a contract, this work acknowledged the uncertain nature of QoS values and provided flexibility for reaching agreement between users and services. The work in [17] and [10] both focused on manage QoS values of web services and their conformance to service level agreement, which is different from our focus.

D. Quality feature extraction

In [14], Quality of Experience (QoE) parameters were extracted from analyzing user reviews. It uses POS tagging to identify frequent nouns in reviews as potential QoE features. Similar nouns are aggregated and grouped into clusters using semantic lexicon, such as SentiWordNet. Representative nouns in each cluster are selected as QoE parameters. This work is most close to our work as it also exploits user reviews as the input for quality feature extraction from services. The difference lies in the way of extracting the features and the extraction result. Instead of choosing frequent nouns, our approach extracts features that are associated with user sentiment orientation towards a service. By seamlessly integrating feature extraction with sentiment analysis, our approach is able to extract better features that are more relevant to the quality aspects of services while being more indicative of users' positive or negative opinions. Our experimental results clearly justify the effectiveness of our approach.

Feature extraction from user reviews has also been investigated in natural language processing and machine learning [5], [9], [4]. Most of these approaches rely on POS

tagging [5], association rule mining [5], unsupervised [9], or semi-supervised learning over unlabeled data [4] for feature extraction. In contrast, our approach adopts a supervised learning strategy that extracts quality attributes, performs sentiment analysis, and assigns sentiment orientation to the quality attributes using a single integrated model.

VI. Conclusion

We proposed a novel approach to learning quality features of web services from analyzing user reviews. We leveraged natural language processing techniques and machine leaning methods. We modeled the problem as sentiment classification and used l_1 logistic regularized logistic regression to simultaneously extract important terms for sentiment classification as QoS features, rank them based on their importance, and evaluate their values with the identified sentiment labels. Experimental results showed that this approach can effectively identify quality features and outperforms current frequency based feature selection method. It can also achieve a higher classification accuracy compared to the current SVM based method.

The further work of this paper will follow two directions. First, we will investigate more real world dataset and perform more experiments to evaluate our approach. Second, we will consider the quality of reviews and the credential of users when learning QoS features in order to improve learning result.

References

[1] A. Amin, A. Colman, and L. Grunske. An approach to forecasting qos attributes of web services based on arima and garch models. In *IEEE International Conference on Web Services*, 2012.

[2] Ying Chen, Jiwei Huang, and Chuang Lin. Partial selection: An efficient approach for qos-aware web service composition. In *2014 IEEE International Conference on Web Services, ICWS, 2014, Anchorage, AK, USA, June 27 - July 2, 2014*, pages 1–8, 2014.

[3] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[4] Rayid Ghani, Katharina Probst, Yan Liu, Marko Krema, and Andrew Fano. Text mining for product attribute extraction. *SIGKDD Explor. Newsl.*, 8(1):41–48, June 2006.

[5] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA, 2004. ACM.

[6] A. Kattepur, N. Georgantas, and V. Issarny. Qos composition and analysis in reconfigurable web services choreographies. In *IEEE International Conference on Web Services*, 2012.

[7] Su-In Lee, Honglak Lee, Pieter Abbeel, and Andrew Y. Ng. Efficient L1 regularized logistic regression. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 401–408, 2006.

[8] Ahmed Moustafa and Minjie Zhang. Learning efficient compositions for qos-aware service provisioning. In *2014 IEEE International Conference on Web Services, ICWS, 2014, Anchorage, AK, USA, June 27 - July 2, 2014*, pages 185–192, 2014.

[9] Ana-Maria Popescu and Oren Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 339–346, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[10] S. Rosario, A. Benveniste, and C. Jard. Flexible probabilistic QoS management of transaction based web services orchestrations. In *IEEE International Conference on Web Services*, 2009.

[11] F. Rosenberg, C. Platzer, and S. Dustdar. Bootstrapping performance and dependability attributes of web services. In *IEEE International Conference on Web Services*, 2006.

[12] Yuanhong Shen, Jianke Zhu, Xinyu Wang, Liang Cai, Xiaohu Yang, and Bo Zhou. Geographic location-based network-aware qos prediction for service composition. In *2013 IEEE 20th International Conference on Web Services, Santa Clara, CA, USA, June 28 - July 3, 2013*, pages 66–74, 2013.

[13] P. J. Stockreisser, J. Shao, W. Alex Gray, and N. J. Fiddian. Supporting QoS monitoring in virtual organizations. In *International Conference on Service Oriented Computing*, 2006.

[14] Bipin Upadhyaya, Ying Zou, Iman Keivanloo, and Joanna W. Ng. Quality of experience: What end-users say about web services? In *2014 IEEE International Conference on Web Services, ICWS, 2014, Anchorage, AK, USA, June 27 - July 2, 2014*, pages 57–64, 2014.

[15] Qi Yu. Decision tree learning from incomplete qos to bootstrap service recommendation. In *2012 IEEE 19th International Conference on Web Services, Honolulu, HI, USA, June 24-29, 2012*, pages 194–201, 2012.

[16] Qi Yu. Qos-aware service selection via collaborative qos evaluation. *World Wide Web*, 17(1):33–57, 2014.

[17] J. Zhang, Z. Huang, and K. J. Lin. A hybrid diagnosis approach for qos management in service-oriented architecture. In *IEEE International Conference on Web Services*, 2012.

[18] H. Zheng, J. Yang, W. Zhao, and A. Bouguettaya. QoS analysis for web service compositions based on probabilistic QoS. In *International Conference on Service Oriented Computing*, 2011.

[19] Z. Zheng, Y. Zhang, and M. R. Lyu. Investigating qos of real-world web services. *IEEE Transactions on Service Computing*, 7(1), 2012.