

Quality of Experience: User's Perception about Web Services

Bipin Upadhyaya, Ying Zou, Iman Keivanloo, and Joanna Ng, *IEEE Member*

Abstract— Web service composition enables seamless and dynamic integration of Web services. The behavior of participant Web services determines the overall performance of a composition. Therefore, it is important to choose high quality services for service composition. Existing Web service selection and discovery approaches rely on non-functional aspects (also known as quality of service or QoS) *e.g.*, response time and availability. Though these parameters are crucial for selecting Web services, they may not reflect the user's perspective of quality. In this paper, we explore the feasibility of incorporating perceived quality from user's perspective for service selection and composition. We name such quality attributes as quality of experience (QoE). First, we propose a solution that automatically mines and identifies QoE attributes from the Web. Second, we study the application of such dynamically extracted QoE attributes for service selection. For the evaluation purpose, we collected more than 34,000 reviews from 58 different services in six domains. Our findings show that it is possible to automatically identify QoE attributes with an average precision and recall of 92% and 80% respectively. Our study shows that there is a strong positive correlation between QoS and QoE. Hence QoE can be used during service selection especially when QoS data are not available. Furthermore, we found 70% of service discovery queries indeed contain QoE attributes showing the importance of QoE attributes during the service discovery phase. Our study also finds 80% of service selection is based on QoE attributes of a service.

Index Terms— service composition, service selection, quality of service, quality of experience

◆

1 INTRODUCTION

Service oriented architecture (SOA) provides a mechanism to publish and receive various forms of information through standard protocols. A common technology for SOA implementation is Web services. Al-Masri *et al.* [22] report that there is more than 130% growth in the number of published Web services in the last couple of years. Similar observation can be made by reviewing the statistics from the Web service search engines such as Seekda [25]. In particular, Programmable Web directory [24] indicates an exponential increase in the number of Web services over the last three years. Such rapid growth in the number of services increases the importance of the service selection task due to the presence of low quality services. Non-functional attributes are exploited as the key decision making criteria in the state of the art approaches for service selection, *e.g.*, [20]. As a result, quality of service (QoS) becomes a significant concept for service selection since QoS properties describe non-functional attributes of services.

Most research in QoS-based service selection [1, 2, 3, 6 and 16] focus on proposing a comprehensive pre-defined QoS schema to represent service requests and offers, or implementing a selection algorithm to achieve an optimized composition. However, the process of obtaining QoS information is largely overlooked. There are mainly two ways to obtain QoS information: static release, and runtime monitoring [16]. Service providers publish static release of QoS information. The static release is not frequently updated, and the QoS attribute are measured in a specific environment and platform.

The published QoS information may be different if the same service invoked from a different geographical location or through different devices. Hence the static information is less reliable. Runtime monitoring is the main way to collect objective and effective QoS information. Runtime monitoring approaches require analysis of Web service quality at client side. Client side evaluation of real world services are resource intensive, time consuming and expensive [22]. These issues threaten the applicability of QoS-based service selection approaches, *e.g.*, [1, 2, 3, 6, and 16].

An alternative source of information about the quality of Web services is online reviews available on the Web. Web 2.0 user-oriented content generation approach has enabled people to broadcast their knowledge and experience to the mass. Online user reviews are one example of such phenomenon. Users express their experience via online reviews to reveal their satisfactions and disappointments about services. In this paper, we explore the possibility of exploiting user reviews for service selection applications. We propose the concept of quality of experience (QoE) which captures and quantifies customer feedback on a service. In this approach, QoE attributes are extracted from online reviews reflecting user experience feedback on Web services. However, the first challenge towards the proposed approach is extracting QoE attributes from user reviews. User reviews are written in natural language and presented as unstructured data. Therefore, it is not trivial for computers to understand, analyze, and aggregate QoE from the Web.

In our paper, we present the result of our study on the possibility of automatically extracting QoE attributes from user reviews. We explore the relationship between QoS and QoE attributes. We study if QoE can replace QoS for service selection in a case of insufficient QoS information. Finally, we examine the behavior of users to see the presence of quality attributes in Web search query and the effect of quality during service selection. This paper extends earlier work, published in IEEE 20th International Conference in Web Services (ICWS) [30]. We enhance the earlier work in the following aspects:

1. We extend the existing case study to include two additional domains: Financial and Food/Nutrition. We further validate the benefits of our approach through an extended case study.
2. We perform a user study to study the presence of quality attributes in Web service search queries. We study the users' behaviors in selecting Web services in the presence of Web services with either QoE attributes or QoS attributes.

We present the result of our study in the following four research questions:

RQ1: Can our approach extract QoE from online reviews?

Our study shows that it is possible to automatically extract QoE attributes from reviews. The proposed approach achieves an average precision of 90% and an average recall of 79%. Our approach identified more than twice as many quality attributes as those that present in traditional QoS attributes.

RQ2: How well QoE reflects QoS of a service?

Our study finds most of the QoE and QoS attributes are strongly correlated. Thus, QoE attributes can be safely used for service selection if QoS is not available.

RQ3: Are QoE attributes used in service search queries for service discovery?

We study the use of quality attributes in service search queries used for service discovery. In our user study, we found that 70% of Web service queries are expressed using QoE attributes.

RQ4: Do QoE attributes affect service selection?

To identify the services preferred and selected by users, we provided a set of services with either QoS attributes or QoE attributes. Our study reveals that service selection is influenced by the QoE attributes. We observe that users tend to select services with description expressed by QoE attributes than QoS attributes.

The remainder of this paper is organized as follows: Section 2 gives an overview of quality in Web services.

Section 3 presents an overview of our approach. Section 4 discusses the case study. Section 5 reports the related work and finally Section 6 concludes the paper and explores the future work.

2 QoS AND QoE FOR WEB SERVICES

Service providers report quality aspects of services as non-functional attributes. These attributes are mainly provided for service composers and developers. An alternative resource for such information is online reviews where users provide their feedback in form of reviews or comments. The experience with a service is called quality of experience (QoE).

2.1 Quality of Service

QoS refers to non-functional properties of Web services such as performance, reliability, and security. Maintaining the promised QoS on the Internet is a critical and significant challenge because of its dynamic and unpredictable nature. Therefore, a wide range of techniques exist that match the needs of service requestors with those of the service provider's based on network resources available and QoS information. Table 1 shows a set of common QoS attributes for Web services [22].

Table 1: QoS of Web Services

QoS	Definition
Response time	Execution time (s) + waiting time(s) + network latency(s)
Reliability	1-Failure Rate(s)
Availability	Uptime (s) / (uptime (s) + downtime(s))
Price	Execution fee for a request.
Usage Limits	Number of requests per day
Security	Authentication Model, SSL support

2.2 Quality of Experience

Quality of Experience (QoE) is a subjective measurement which reflects user's experience with a service. A user can provide her opinion on any aspect of a service, e.g., cost and performance. Each aspect of a service is called QoE attribute. Contrary to QoS, QoE reflects quality from the user's point of view. The primary source of QoE is online reviews. Since reviews come from large number of users with diverse platforms and different geographical locations, QoE becomes a credible source of information. Figure 1 shows, reviews from three different users taken from <http://expertreviews.co.uk>. The reviews contain valuable information provided by people who used Dropbox service. The first user tells his experience with synchronization and folder sharing capability. The second user expresses her dissatisfaction with cross platform support and security. As pointed out by this example, the QoE

attributes can be mapped to QoS attributes such as performance and security. Hence, in this paper, we explore the possibility of extracting and using QoE attributes for service selection purposes where QoS information is not available.

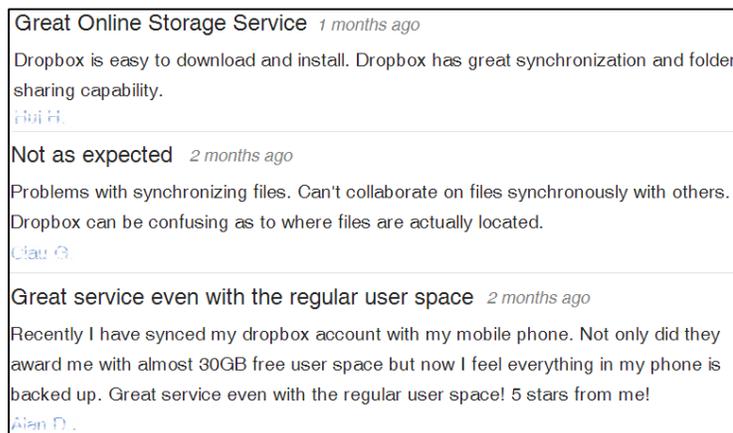


Figure 1: Sample reviews of an online storage provider

3 OUR APPROACH TO EXTRACT QOE ATTRIBUTES

Users use natural language to provide their feedback in reviews or comments. Furthermore, a user may mention more than one quality attribute of a service in a review. Therefore, the first step is to automatically extract and aggregate QoE attributes from reviews since finding and going through a large number of reviews to manually find QoE information for service selection and composition is not feasible. Ideally, the automatic

approach analyzes the natural language content, identifies QoE attributes, and represent them in a structured way that can be used by service selection algorithms.

In this section, we describe our approach to extract quality of experience information for Web services. Figure 2 shows an overview of our approach. Our QoE extraction approach mainly consists of four steps. First, we crawl the Web for user reviews. Second, we use natural language processing techniques to automatically and dynamically extract QoE attributes. Finally, we store QoE attributes in a database and provide an interface to query the extracted QoE attributes for service selection.

3.1 Crawling online reviews

Given an unseen Web service, we crawl reviews and put them in a review database. We form a Web search query to get the reviews posted within the last 2 years on the Internet. The downloaded reviews are locally stored as HTML Web pages. Malformed HTML files are quite common in the Web. For example, an HTML file may contain mismatched HTML tags. To generate the DOM tree structure from an HTML file, we use the HTML syntax checker [15] to correct the malformed HTML tags. We then extract reviews from the stored pages in a text format without HTML tags.

3.2 Processing reviews

A QoE has two key data fields which are attributes (e.g., streaming) and opinion (e.g., unreliable) associated with an attribute. For an unseen service, neither its QoE attributes nor are opinions known in advance. In this section, we describe our approach to dynamically identify both QoE attributes and opinions from user reviews.

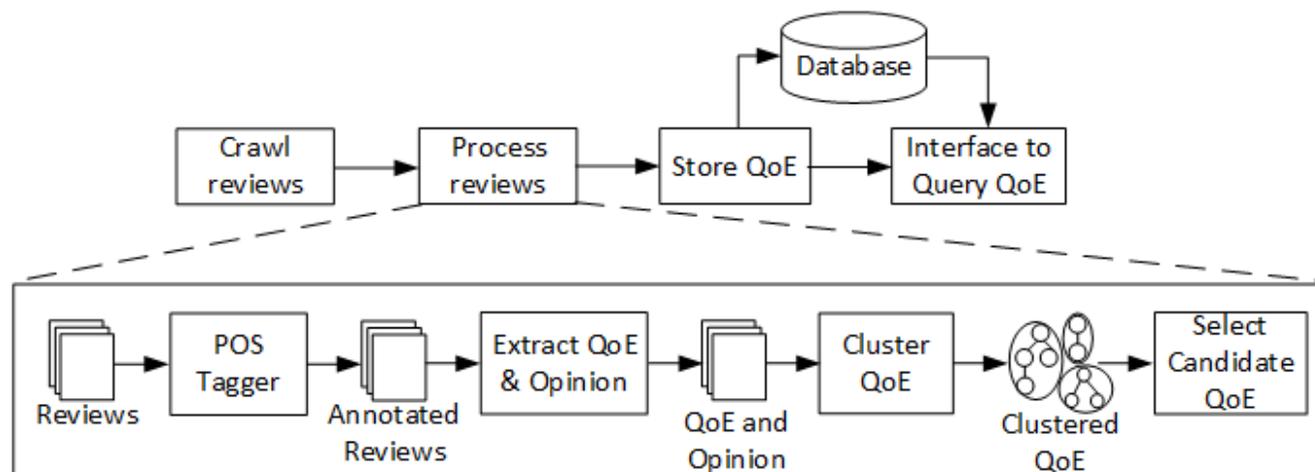


Figure 2: Overview of our approach

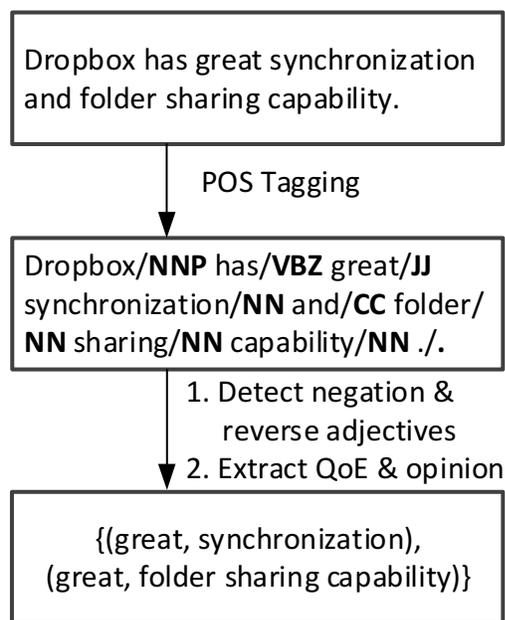


Figure 3: Extracted QoE attributes and opinion

3.2.1 Identifying POS tags in reviews

A review typically has several sentences. Usually, a single review by a user expresses multiple positive and negative opinions. For example, a Dropbox reviewer may use a couple of sentences to praise its performance, but use other sentences to belittle its cost and media streaming capability. For each review, we identify the target attribute and opinion. It is not trivial to determine the opinion orientation of such review without distinguishing different segments of a review. To overcome this problem, we split a review into sentences. This approach makes it possible to assign positive or negative opinions on different aspects of an experience.

Natural language processing helps us determine the part of speech (POS) of each word in a sentence. POS is used to define a syntactic or morphological behavior of a word. The English language grammar classifies parts-of-speech in the following categories: verb, noun, adjective, adverb, pronoun, preposition, conjunction and interjection. Each above mentioned category plays a specific role in a sentence. For example, nouns give names to objects, beings or entities and an adjective qualifies a noun. As a result, POS identifies the behavior of each word which in turn helps us understand a reviewer's experience. We use a well-known POS (part-of-speech) tagger [12] to identify the syntactic structure of a sentence. Second box in Figure 3 shows a review sentence with POS tags. We post-process the generated tags to resolve object names consisting of multiple words (e.g., "Folder sharing capability"), phrasal verbs (e.g., "go to"), and pronominal referrals (pronouns e.g., "it"). We assume words like "it" always refer to the last mentioned

object, which proved to be a sensible heuristic in most of the cases.

3.2.2 Extracting QoE attributes and opinion

We represent the extracted reviews as shown in equation (1) and transform extracted QoE information to a schema shown in equation (2). For a review, quality attributes (i.e., QA) and its opinion value (i.e., R) are stored as QoE and OScore in equation (2). We extract quality attributes from the body of a review by analyzing its POS (i.e., the tagged review after POS analysis in Figure 3).

$$Review = (service, user, date, body, (QA, R), TV) \quad (1)$$

where, *body* is the textual content of a review of a user on a specific date. *QA* and *R* is the quality attribute and its rank provided by a user. *TV* is the overall quality rank for a service.

In the outcome of POS tagger, adjectives and adverbs reflect the opinion about nouns. Opinions encode an emotional state, which can be desirable or undesirable. Opinions that encode desirable states (e.g., beautiful, nice, and happy) have positive orientation while the ones that encode undesirable state (e.g., bad, terrible and disappointing) have a negative orientation. Often the opinion information in a sentence is expressed as "not", "no" and "barely". In such case, the sentiment about the QoE attribute is the opposite of the corresponding opinion phrases. For example, two consecutive negative terms reflect a positive opinion (e.g., no problem). The overall idea is to apply such rules to infer the final value (i.e., opinion) for each mentioned QoE attribute. We employ the idea proposed by Turney *et al.* [23], where two consecutive words are extracted from a review if their tags conform to predefined patterns. The first pattern means that two consecutive words are extracted if the first word is an adjective and the second is a noun. For example, "The maps support multiple destinations", the "multiple destinations" phrase is the quality. The second pattern means that two consecutive words are extracted if the first word is an adverb, and the second word is an adjective, but the third word is a noun. The third pattern means that two consecutive words are extracted if they are all adjectives, but the following word is not a noun. Singular and plural proper nouns are avoided so that the names of the items in the review cannot influence the classification. At this stage, we have extracted QoE attributes and opinion of each review. The extracted information is stored as a tuple shown in equation (2).

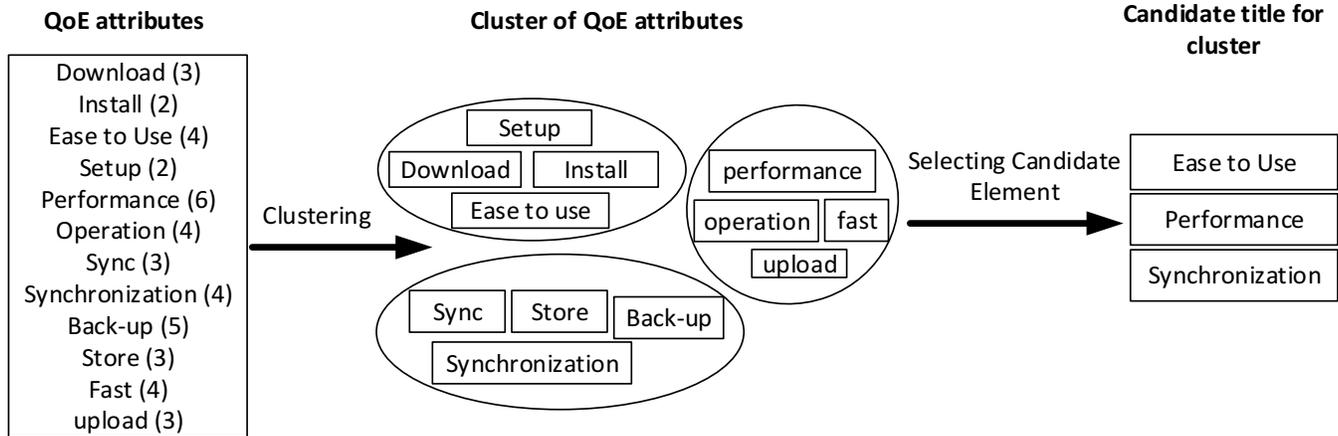


Figure 4: The process of clustering QoE attributes and selecting a representative title

$$Extract_{Review} = \{QoE, Opinion, OScore, Date\} \quad (2)$$

where *QoE* is a quality of experience attribute; *Opinion* is the opinion about *QoE*; *OScore* is the polarity score of *Opinion* and *date* is the time when the review was posted.

We quantify the QoE attributes based on the opinion provided by users. In this paper, QoE can be quantified as score between [0, 1]. 1 represents the highest positive opinion for a service, and 0 relates the lowest negative feedback. We used SentiwordNet [4] to calculate the positive and negative effect of opinion in a QoE attribute.

3.2.3 Clustering QoE attributes

An extensive list of QoE attributes and opinions of QoE attributes can be extracted using the process defined in section 3.2.2. QoE attributes are not predefined since they depend on the nature of target Web services and users experience. At this stage, our goal is to find related attributes, represented with different phrases, and find a representative title for each group of similar candidates. This step clusters similar QoE attributes together and summarizes the mentioned opinions for the finalized QoEs. Figure 4 presents an illustrative example of the input and output of this step. To automatically create the clusters, we use k-means [26][27] which is an unsupervised clustering algorithm. K-means algorithm divides the data into a set of disjoint groups.

$$wordSim(x, y) = 1 - \frac{mcp(cp)}{mcp(cp) + dcp(cp, root)} \quad (3)$$

where *cp* is the common parent of the two QoE attributes *x*, *y*; *root* is the root of the WordNet ontology; *minimum common parent length* (i.e., *mcp(cp)*) is the shortest path from either *x* or *y* to *cp*, and *dcp(cp, root)* is the length of the path from *cp* to *root*.

The main challenge in applying the clustering algorithm is to identify the expected number of clusters [28]. In case of k-means, this parameter is called *k*. One possible solution is to ask domain experts to identify the proper value for *k* empirically. However, since we need to automate the process completely, we use a clustering validation approach proposed by Rousseeuw [28]. Using this approach, we can measure the success of any possible value for *k* in generating a set of coherent clusters. To find the proper value for *k* automatically, we create clusters with all possible values of *k* where the maximum value is the number of distinct data points. Then, we measure the success of each experiment using Rousseeuw [28] approach. Finally, we select the *k* value with the highest success rate for the clustering step in Figure 4.

We use semantic similarity as shown in equation (3) to find the distance between words. We use WordNet [14] to find the similarity between the QoE attributes. In WordNet, all words are connected as a graph. The two words can be directly or indirectly connected through many intermediate relations. The distance in our approach is defined as the number of intermediate words of the shortest path between two words. The similarity between two QoE attributes, *x* and *y*, is measured by the path length (i.e., *dcp* in equation (3)) between words to reach their common parent in WordNet ontology as used in [15]. The value of the similarity is shown in equation (3) ranges from 0 to 1. 0 represents unrelated words and 1 signifies synonymous words. Figure 4 shows the extracted QoE and the corresponding clusters based on the word similarity. In this example (Figure 4), in total, our approach identified 3 final QoEs (shown as clusters) from the 8 initial QoE attributes.

$$R(x) = \left\{ \sum_{y \in C; y \neq x} WordSim(x, y) f(y) \right\} + f(x) \quad (4)$$

where $R(x)$ denotes the rank of the QoE x in the cluster C ; $WordSim(x, y)$ is the similarity between QoE x and y , and $f(x)$ is the frequency of the QoE x .

3.2.4 Selecting representative titles

In this step, we identify a representative title for the QoE attribute of each cluster. The candidate element represents the whole cluster. The final sentiment associated with the representative QoE attribute is an average of all the sentiments of the elements in the cluster. Our approach to select a candidate element from a cluster is similar to our previous work [15]. Equation (4) shows how we compute the rank of a QoE attribute x in cluster C . Ranking QoE attributes signifies the frequency of a QoE attribute with respect to the other QoE attributes in a cluster. The computed rank is then normalized between 0 to 1 by dividing the raw value by the sum of all QoE rank values in a cluster. 1 signifies the most dominant QoE and a QoE with the largest normalized rank value represents the cluster. For example, as shown in Figure 4, the similarity between sync and synchronization is 1 as one is the abbreviation of another; synchronization and backup is 0.7; synchronize and store is 0.6. Using these similarity values, we compute the rank of the QoE candidates {synchronization, backup, and store}, the QoE attributes rank as described in Equation (4) is {synchronization (0.3+0.6+7=7.9), backup (0.3+0.4+5=5.7), and store (0.6+0.2+3=3.8)}. Hence, we select Synchronization as the representative title for the QoE attribute of the cluster {Sync, Store, Synchronization and back-up}.

3.3 Store and query QoE attributes

Once we have ranked and indexed services based on the user’s quality of experience. We store QoE attributes in a database. We provide a user interface (UI) on top of a database. A user has an ability to query for QoE attributes for a service. The result shows information about a service such as the name of a service, service category and QoE attributes and its score as shown in Figure 5. A user can query about the trend for each QoE attribute. QoE attributes and opinions are recalculated and updated as new reviews are downloaded by the crawler.

4 CASE STUDY

We conduct a case study to evaluate the effectiveness of our approach. The objectives of the case study are: 1) to evaluate our approach in terms of precision and recall for automatic QoE attributes extraction, 2) to measure the correlation between QoS and QoE attributes, 3) to

evaluate the use of quality attributes in Web search queries for service discovery , and 4) to observe if QoE attributes affect users in service selection.

4.1 Case study setup

In this sub-section we discuss of the case study set-up.

4.1.1 Data collection and processing

We collect reviews for Web services from six different domains: 1) trip (e.g., CleanTrip and Ebookers), 2) shopping (e.g., Amazon and eBay), 3) storage (e.g., Dropbox), 4) mapping service (e.g., Google maps), 5) Financial (e.g., PayPal and Square), and 6) food/nutrition (e.g., Zomanto) as shown in Table 2. Services in the first two domains are aggregator in table 2. A service aggregator is a type of broker that packages and integrates multiple Web services into one or more composite services. To avoid skewness in the data, we crawled similar number of reviews for each category. We crawled reviews from different sites such as pcmag.com, sitejabber.com, and expertreviews.co.uk. For each service, we crawled and downloaded reviews. We clean these reviews by removing html tags and store the review in the format as discussed in equation (1) in section 3.2. Table 2 shows the services that are considered for our case study. The table also describes the number of sentences extracted from the reviews and the number of sentences directly expressing an opinion about the quality of experience. We used the gathered raw data as the input of our case study.

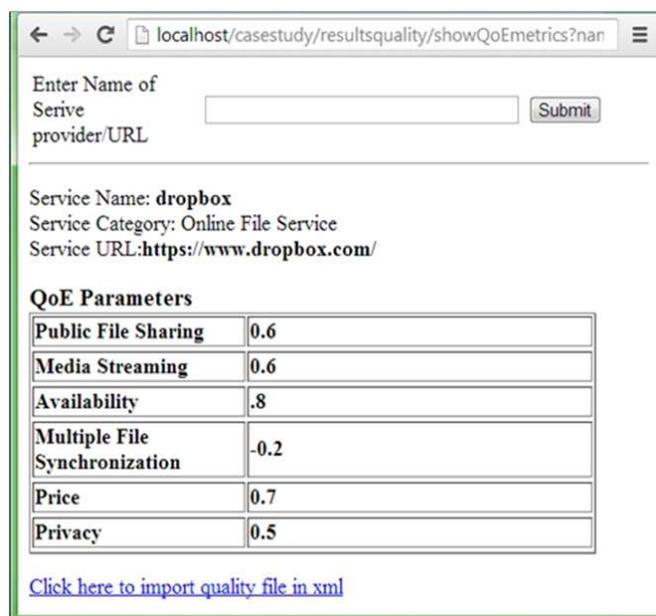


Figure 5: Interface showing a user selecting different quality metrics

Table 2. Services and their review sentences used in our case study

Domain	Agg.	Number of services used in the study	Number of sentences in reviews	Number of sentences with QoE & Opinion	Example Services
Trip	Yes	10	7428	6980	Expedia, Yahoo Travel, Tripit
Shopping	Yes	15	6306	5866	Amazon, Ebay, Best buy
Storage	No	8	7033	6611	Dropbox, Box, Google Drive
Mapping Service	No	5	4529	4110	Google Maps, Bing Maps, Openstreet
Financial	No	10	5359	4890	PayPal, BrainTree, Square
Food/Nutrition	No	10	4010	3850	Zomanto, Order.In, ReciPal

As part of our study, we require QoS information of the subject services. During the preparation phase, we gathered the required QoS data. We implemented the service invoker using JDK 7.0, Eclipse 3.6, Axis2 and HTTPClient4.3. Axis2 is employed to generate the Web service invocation and test cases for SOAP-based services. HTTPClient4.3 is used to invoke RESTful services. We used an automated agent to measure the average response time by considering a period of two months. We extracted the availability of services data as posted by the service providers. We extracted the service cost and usage limits from service providers' documentation. The information regarding price and usage limits were not readily available, and we gathered them manually

proper sentence structure. For the rest of the data, we also manually created a gold standard dataset. Gold standard dataset includes all the QoE attributes available in our review dataset. We use this dataset in order to evaluate the performance of our proposed approach discussed in Section 3. In our case study, the first author, as the evaluator, inspected all the data to create the gold dataset for QoE attributes. The evaluator has more than two years of experience in developing services and service oriented systems. To create such gold dataset, we manually read all the reviews. For each sentence in a review, we tag QoE related attributes and opinions. Whether the opinion is positive or negative (*i.e.*, orientation) is also identified. If the user gives no opinion in a sentence, the sentence is not tagged as we are only interested in sentences expressing an opinion in this work.

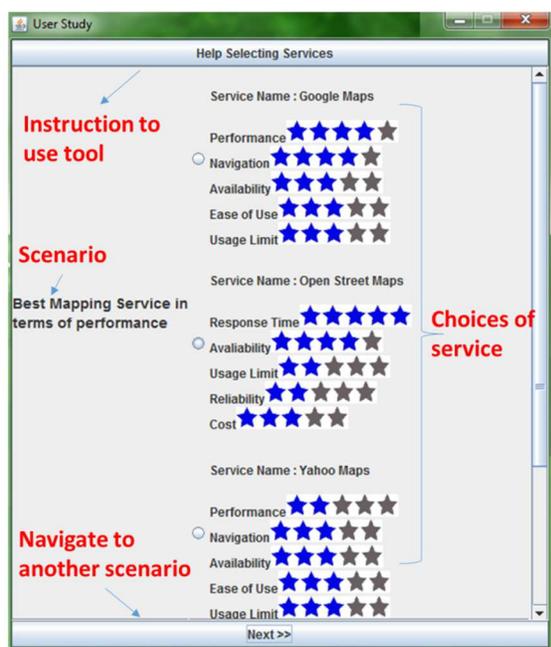


Figure 6: Annotated screenshot used in the user study

4.1.2 Gold standard dataset preparation

From the crawled review dataset, we removed reviews that are not in English language and do not form the

4.1.3 User Study

We setup a user study to evaluate the use of QoE attributes in Web service queries for service discovery and examine the importance of QoE attributes used by users for service selection. More specifically, given a set of services with either QoS or QoE attributes, we want to study which service is likely to be selected by a user. We recruited ten participants to participate in our user study. All the participants are graduate students and have the software engineering background. Moreover, each all the participants have more than one year experience in using Web services. We want to know how a user formulates a query given a set of scenarios related to discovering services from the domains listed in Table 2. We gave ten different scenarios, e.g. "Select the best service to book a flight ticket with lowest price", to each participant. All the participants are given the same scenarios. We record each service query to identify the presence of quality attributes (*i.e.*, QoE or QoS attributes) in queries. For each query in the scenario, we list a set of functionally equivalent services based on our previous work [15]. For each scenario, we presented the

services with either QoE attributes or QoS attributes but not both. We record participant service selection to assess the importance of QoS or QoE in service selection. Figure 6 shows a screenshot showing one of the scenarios and the available services. A participant selects one of the services from the list.

4.2 Results of our study

In this section, we outline the motivation, approach and findings of our research questions.

RQ1. Can our approach effectively extract QoE attributes from reviews?

Motivation. While QoS attributes are predefined and documented (*e.g.*, [1, 5, and 29]), QoE attributes are dynamic and domain dependent. Therefore, we extract QoE attributes automatically from the Web. In this research question, we measure effectiveness of our approach introduced in Section 3 to extract QoE attributes from reviews.

$$P = \frac{\{relevant\ attributes\} \cap \{retrieved\ attributes\}}{\{retrieved\ attributes\}} \quad (5)$$

$$R = \frac{\{relevant\ attributes\} \cap \{retrieved\ attributes\}}{\{relevant\ attributes\}} \quad (6)$$

Approach. We use precision and recall in order to measure the effectiveness of our approach to identifying quality of experience (QoE) attributes. As shown in Equation (5), the precision is the ratio of the total number of QoE attributes correctly extracted by our approach to the total number of QoE attributes. Recall is the ratio of the total number of QoE attributes correctly extracted by our approach to the total number of QoE attributes exist in the reviews as shown in Equation (6). However, to successfully calculate the precision and recall, we need an oracle covering the relevant QoE attributes that are required by Equation (5) and (6). We use the QoE gold standard dataset for services in Table 2 which is created manually as part of our case study setup.

Finding. Table 3 summarizes the result of the performance evaluation study on the effectiveness of our proposed approach for extracting QoEs automatically. We compared the extracted QoE attributes with the manually made oracle that covers all QoE attributes from six selected domains. The effectiveness is measured via precision and recall as described in Equation (5) and (6).

Our approach extracted all QoS-related QoE attributes with 100% precision and recall. The additional new domain specific QoE attributes extracted by our approach have the precision above 92%, meaning that our approach can correctly identify the QoE attributes. The recall is above 80%, meaning that coverage of our approach is acceptable. Our manual investigation revealed that the missing cases that affect our recall negatively are due to implicit expressions. In such cases, QoE attributes may not appear in sentences explicitly. We call such QoE attribute implicit QoE. For example, in one of the reviews in the mapping service, the reviewer expressed her unsatisfactory opinion about the latency time by saying “you can go for a cup of tea after requesting...”. In overall, considering the limitations in opinion mining techniques and comparing to the performance observed in the other successful opinion mining techniques of other domains, *e.g.*, [23], we can conclude the precision and recall of our approach is acceptable.

As shown in Table 3, our approach identified more than twice as many quality attributes (*i.e.*, total #QoE attributes) as those that present in traditional QoS attributes (*i.e.*, #overlapped attributes). Furthermore, to show the dynamic nature of QoE attributes, we plotted eight most frequent QoE attributes and plotted the values for the past 13 months in Figure 7. Figure 7 shows each service has different scores for quality metrics. Some quality metric may be strong while others may be weak. Therefore, it is reasonable to let a user to select a service based on a different quality aspect to achieve higher satisfaction. Figure 7 shows that QoE can also capture the changes in the quality. We see six different patterns: increasing, decreasing, constant, decreasing and then increasing, increasing and then decreasing, and zigzag. The presence of these different trends, shows the need to continually collect reviews and extract QoE attributes to know or predict the quality of services.

Table 3: Evaluation on automatic extraction of QoE attributes

Domain	Overlapped QoE and QoS Attributes		New QoE Attributes		Total #QoE Attributes	#Overlapped Attributes	#New QoE Extracted
	Precision	Recall	Precision	Recall			
Travel	100%	100%	0.93	0.72	18	5	8
Shopping	100%	100%	0.92	0.87	16	5	9
Storage	100%	100%	0.93	0.76	17	5	8
Mapping Service	100%	100%	0.90	0.82	17	4	10
Food/Nutrition	100%	100%	0.89	0.80	10	3	6
Finance	100%	100%	0.95	0.85	12	3	8

Our approach finds all QoS-based attributes from reviews. Our approach identifies new domain specific QoE attributes with high precision and recall. We showed QoE changes over the period of time as service providers improves or degrades their service.

RQ 2. How do QoE attributes relate with QoS attributes?

Motivation. QoE and QoS come from different sources. QoS is provided by service providers or recorded by a client, whereas QoE is directly reflected in the user's feedback. The process of collecting QoS related information is tedious, time consuming and difficult to collect on client side [22]. In this research question, we explore the possibility of using QoE during the service selection process. Since our approach can be automated, and it is independent of service providers, it can be replicated across different domains. A strong correlation between QoE and QoS attributes indicates the possibility of using QoE attributes for service selection.

Approach. To evaluate the relation between the QoS and QoE attributes, we collected QoS attributes for all the services based on [5 and 22]. We measured and collected each of the quality metrics described in which we have QoE attributes. QoS attributes, such as cost and security, extract from the service provider's Web page, whereas QoS attributes, such as upload speed is measured by writing a client program that calls a service API. We map the QoE attributes to their corresponding QoS attributes. For example, QoE synchronization, QoE upload speed and QoE media streaming in the storage domain are mapped to QoS performance.

To study if the opinions expressed for QoE attributes are in agreement with QoS data, we use the Pearson correlation coefficient. The Pearson population correlation coefficient of QoS attributes and QoE attributes is defined as the ratio of the covariance of QoS attributes and QoE attributes and the product of their standard deviation.

Findings. Our approach discovers five QoS attributes from reviews: Performance, Availability, Usage Limit, Security and Cost. Security is excluded because QoE attribute security corresponds to the number of times a user felt the system or software were hacked or broken. However, the similar information for Web services is not freely available. The security QoS attribute available is the encryption and secure socket layer used

by the service provider. Since we cannot measure security, we decided not to use the metric in our study. Similarly, we did not find QoS attribute cost for travel, shopping, mapping services and food/nutrition, as all of the services in those domains are free.

Table 4 lists the absolute values of correlation, fitness and p -values of related QoS and QoE attributes. Our study shows a high correlation between QoS and QoE attributes except in the case of cost QoS attribute in the storage domain. Performance and usage limit attributes are highly correlated. For performance attribute, all the service statistically significant as their p -value is less than 0.05. Similarly, QoS and QoE attributes for the availability of services in the storage and mapping services are highly correlated with p -value less than 0.05. We found the availability of shopping services and the availability of travel services do not have the same level of correlation as other QoS attributes. We went and re-analyzed the sentiment related to availability for shopping services. We found the sentiment of availability was mixed with product availability and service availability. Similarly, for travel services, sentiment for the availability is mixed with the hotel and flight availability.

The cost QoS attribute listed in Table 4 is available for services in the online storage and financial domains. For services in the storage domain, there was almost no correlation between the cost QoS attribute collected and the sentiment of cost QoE attribute. When we analyzed the reviews related to cost QoE attribute for the online storage, we find most of the reviews sentiment was related to the free storage space rather than the commercial plan of storage. We then try to find the correlation between free space provided by the service provider and the sentiment of QoE cost. Our analysis shows correlation between cost QoE and Free space is 0.946, and the fitness value is 0.895. Hence reviews and comments on the online storage are based on free storage rather than the average cost for using the storage service. However, for the services in the financial domain, there is a positive correlation between QoS and QoE attributes related to cost.

When there exist equivalent QoS and QoE attributes, the attributes are correlated. In addition, we have observed that there are more QoE attributes in comparison to QoS attributes. For example, security related reviews provide information about the total hacks which are not available in QoS. Similarly, the availability of multiple domain related QoE attributes, such as product quality, can be useful for service selection.

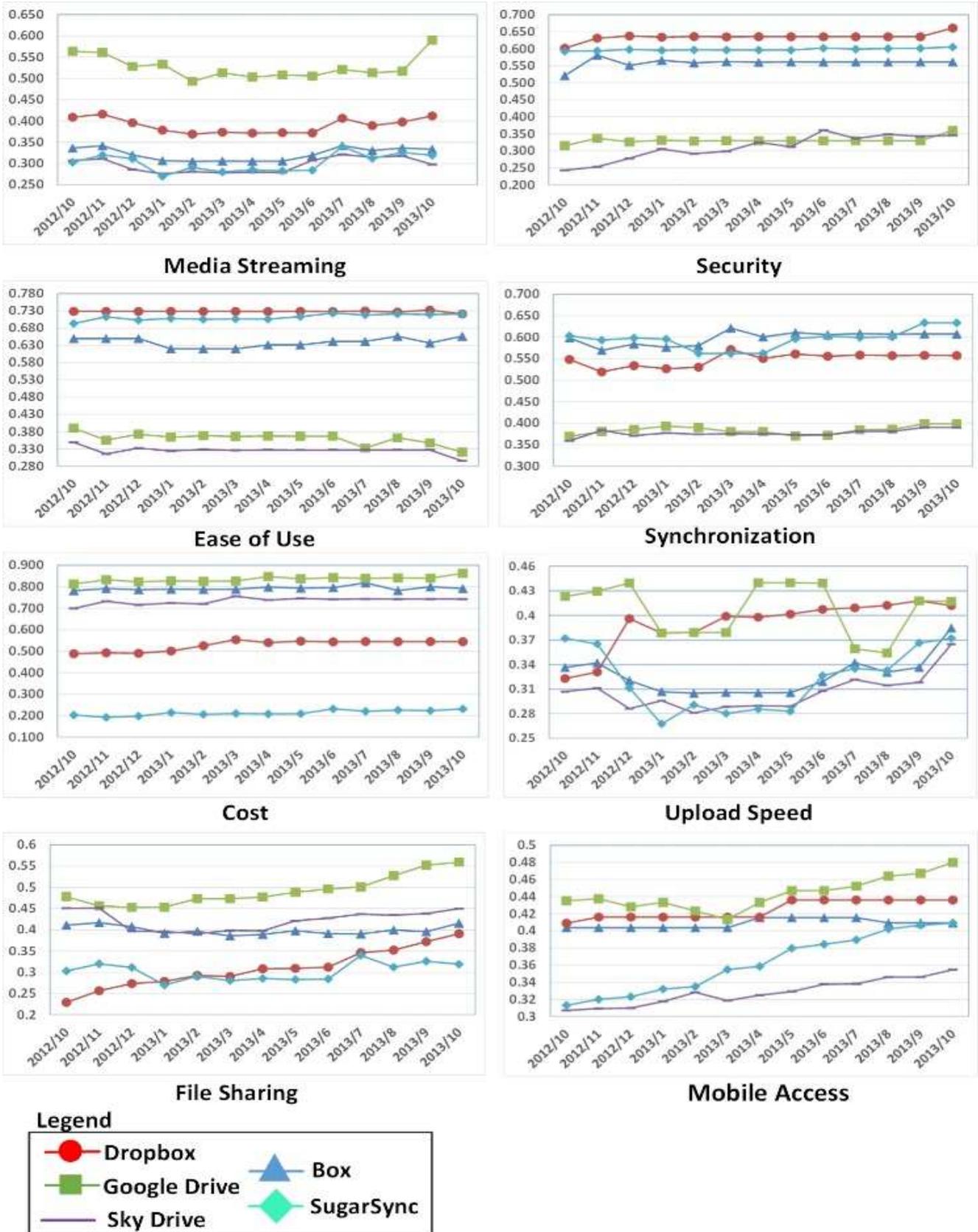


Figure 7: Eight most frequent QoE attributes of online storage providers over a period of 13 months

Table 4: Summary of our study on the relation between QoE and QoS attributes

Domain	Performance			Availability			Usage Limit			Cost		
	cor.	r ²	p-value	cor.	r ²	p-value	cor.	r ²	p-value	cor.	r ²	p-value
Travel	0.948	0.900	0.001	0.475	0.226	0.280	0.994	0.989	3.6e-6	-	-	-
Shopping	0.939	0.883	0.005	0.333	0.111	0.518	0.986	0.973	0.001	-	-	-
Storage	0.950	0.904	0.012	0.968	0.937	0.006	0.998	0.996	3.7e-6	0.017	0.0002	0.978
Mapping Service	0.953	0.908	0.046	0.994	0.988	0.005	0.713	0.508	0.286	-	-	-
Financial	0.944	0.892	0.004	0.980	0.961	0.0005	-	-	-	0.987	0.975	0.0002
Food/ Nutrition	0.981	0.963	0.0004	0.972	0.945	0.001	0.972	0.946	0.001	-	-	-

High correlation between QoS and QoE attributes suggest the applicability QoE for service selection when QoS is not available.

RQ3: Are QoE attributes used in service search queries for service discovery?

Motivation. With so many Web services available in the Web, it is a common approach to discover [30] Web services by querying the Web service repositories, *e.g.*, Programmable Web [24]. A user expresses his or her expectation of the desired service as a service search query. Two types of information can appear in such queries: (1) the query includes the main functionality of the service, *e.g.*, hotel booking; (2) the query can contain some non-functional quality features about the desired service or product, such as price range. A service search engine is responsible to find the best matching services by taking into consideration of quality constraints specified in both the query and service descriptions. In this research question, we investigate whether QoE attributes can be useful for service discovery via search. Specifically, we explore if QoE attributes appear in service search queries when a user wants to express the quality constraints.

Table 5. Results of Web service query formulation from our user study

Observation	Percentage of web queries with observations
Web search queries with either QoS or QoE attributes	80%
Web search queries with QoS attributes only	10%
Web service queries with QoE attributes only	70%

Approach. To evaluate the presence of QoE attributes in Web service search queries, we perform a user study. For each participant, we give ten different scenarios for which they have to find the services. We ask them to write a query for each search scenario. We record and analyze their queries to identify the presence of quality attributes. We measure the number of times that a participant expresses the quality attributes using QoE attributes or QoS attributes.

Findings. Our study shows that quality attributes appear in Web service search queries. Table 5 outlines our findings. We found around 80% of Web service queries have either QoS or QoE attributes embedded in them. More specifically, around 10% of Web service queries have QoS attributes and approximately 70% of Web search queries contain domain specific QoE attributes. The presence of quality attributes in web service queries signifies the importance of quality for users. The substantial gap between the QoS and QoE attributes in queries is due to the fact that QoE attributes are closer to end-users' vocabulary over QoS attributes.

Quality attributes appear in Web service query. Around 70% of Web service queries have QoE attributes.

RQ4: Do QoE attributes affect service selection?

Motivation. Typical web service selection mechanisms are based on the prediction of services' performance from the quality advertised by providers. Since QoS and QoE attributes are correlated, we want to know how comfortable a user is to select a service with QoE attributes. We also want to see if there is a higher chance for a service to be selected based on QoE attributes because QoE attributes are domain specific and easy to understand.

Approach. We want to examine how a participant selects the services with QoS attributes or QoE attributes description. For each scenario, we returned 6 to 10 services. There is equal number of services with QoE attributes as services with QoS attributes. Each service is described by either QoS attributes or QoE attributes. But the participants are not aware of the types of the attributes in the description. For each scenario, the participants make a service selection from the list of functionally equivalent services. It is totally up to the participants to decide on which service to be selected. We evaluate the distribution of users' selection of services and count the number of times that a participant has selected a service with QoE attributes over a service with QoS attributes.

Table 6. Result of service selection from our user study

Observation	Percentage of users observed
Selected service with QoS attributes	20%
Selected service with QoE attributes	80%

Findings. Table 6 outlines the result of our findings. We find that given a set of equivalent services with either QoS attributes or QoE attributes, it is 80% more likely a user selects a Web service described with QoE attributes. We observe that around 20% of times a participant selects a service with QoS attributes instead of a service with QoE attributes. We think the expressiveness of QoE attributes helps a participant to relate with QoE attributes than QoS attributes. Hence more participants choose services with QoE attributes.

Services with QoE attributes are 80% likely to be selected by users.

4.5 Threats to validity

In this section, we discuss the limitations of our approach and the different types of threats which may affect the validity of the results of our case study. The main threat of our case study that could affect the generalization of the presented results relates to the number of service description documents analyzed. We have analyzed more than 34,000 reviews of different services from different domains. Nevertheless, further validation of our approach requires an analysis of a larger set of reviews. Our dataset was limited to 2 years results from a Web search query and hence does not give the whole picture of all the comments by a user.

The QoE is manually checked by the authors and is arguable whether a particular attribute is a QoE attribute or not.

5 RELATED WORKS

The problem of QoS-based Web service selection and composition has received a lot of attention by many researchers.

QoS Model and Metrics. Sabata *et al.* [31] sketched a QoS taxonomy, mostly in the context of Web applications. Most of the existing approaches use the generic QoS attributes for Web service discovery such as response time, reliability, availability and cost [2, 4, 6, 7, and 8]. Ran [5] extend the traditional service discovery model with a new role called a Certifier, in addition to the existing three roles of Service Provider, Service Consumer and UDDI Registry. The Certifier verifies the advertised QoS of a Web service before its registration. The consumer can also verify the advertised QoS with the Certifier before binding to a Web service. This approach prevents publishing invalid QoS claims during the registration phase, and help consumers to verify the QoS claims. Although this model incorporates QoS into the UDDI, it does not provide a matching and ranking algorithm, nor does it integrate consumer feedback into service discovery process.

Reputation based systems. Building trust and reputation for web service providers is beneficial for web service selection, and has been neglected in current trust and reputation approaches for web services. A good reputation of a service provider can enhance a consumer's confidence in its services. More importantly, for the service for which the trust and reputation has not been established, e.g. a new service or a service that has not been selected by consumers, the trust and reputation of the service provider, accumulated by the provider from providing other services, can be used for the selection since if a provider has a good reputation for providing good quality services, a consumer would like to believe that its new service has good quality too.

Service quality can be determined collaboratively by participating service consumers and agents via the agent framework. Xu *et al.* [10] incorporated QoS with customer feedback to enhance the service selection approach. Kalepu *et al.* [17] evaluated the reputation of a service as a function of three factors: ratings made by users, service quality compliance, and the changes of service quality conformance over time. Liu *et al.* [18] suggested an approach for rates services computationally in terms of their quality performance from QoS information provided by monitoring services and users. All the above mentioned approaches do not explain sources of the user feedback and the ranking methods

for the feedback. Our work is based on extracting QoE from user feedback in the Web and using it to select services. Our work also finds the correlation between traditional QoS attributes and QoE attributes extracted. We use feature extraction and sentiment mining to find the meaning embedded in the service reviews that are expressed in natural language.

Quality driven Discovery and Selection. Brokers can enable dynamic selection of services using QoS [5, 33]. The brokers use third party certifiers to collect QoS data on the services. The main difference with our work is that we use users experience to select services that comes directly from diverse geographical location and diverse platforms.

Zeng *et al.* [32] discuss a global planning approach for selecting composed services. They propose a QoS model using the examples of price, availability, reliability, and reputation. They apply linear programming for solving the optimization QoS matrix formed by all of the possible execution plans that result in the plan with the maximum QoS values. The major differences with our work is that we extend reputation to encompass all qualities and our model for reputation has a dampening temporal characteristics. Poladian *et al.* [34] present a mathematical model of the problem of configuring user tasks. They assume that each configuration is based on selecting services from providers of differing QoS.

6 CONCLUSION

We presented an approach to identify and aggregate QoE attributes for a service. Our approach has shown significant precision and recall on the identification and grouping of QoE attributes in reviews. We also provide an approach to query the quality attributes for a service. Since all the steps were performed in a domain-independent way, the system is flexible enough to be equally applicable to any other domain. The recall of the QoE identification system is not 100%, however, in real life scenario, most of the services have a sizable amount of reviews, and hence even a moderate recall result could be representative and helpful to customers. Our study shows our approach can identify all the QoS information discussed in the reviews. Most of the QoE and QoS attributes are highly correlated, suggesting that we can use QoE attribute for service selection whenever QoS is not available.

Through a user study, we showed that 80% of Web service search queries have quality attributes specifically, 10% of queries have QoS attributes and 70% of queries have QoE attributes. Our study also shows that the services described with QoE attributes are 80% more likely to being selected. In the future, we will perform a

user study to show the effectiveness QoE attributes in a service composition process. We would like to extend our approach to address the bootstrapping problem for QoE attribute identification.

REFERENCES

- [1] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, Q.Z. Sheng, Quality Driven Web Services Composition. In World Wide Web (WWW) Conference 2003
- [2] G. Canfora, M. Di Penta, R. Esposito, M.L. Villani, An Approach for QoS-aware Service Composition based on Genetic Algorithms. In 7th annual conference on Genetic and evolutionary computation (GECCO), pages 1069-1075, 2005, ACM Press
- [3] G. Canfora, M. Di Penta, R. Esposito, F. Perfetto, M.L. Villani, Service Composition (re)Binding Driven by Application-Specific QoS. In International Conference in Service-Oriented Computing (ICSOC), 2006. LNCS, vol. 4294
- [4] A. Esuli, F. Sebastiani, "SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining", In 5th Conference on Language Resources and Evaluation, 2006
- [5] S. Ran, "A model for Web services discovery with QoS", ACM SIGecom Exchanges. 4, 2003,pp 1-10.
- [6] M. Rathore and U. Suman, "A Quality of Service Broker Based Process Model for Dynamic Web Service Composition," Science Publications, Journal of Computer Science, USA, Vol. 7, No. 8, pp. 1267-1274, 2011.
- [7] R. Khorsand, M. Esfahani "Reputation Improved Web Service Discovery based on QoS," Journal of Convergent Information Technology (JCIT), Vol. 5, No. 9, 2010.
- [8] S. A. Ludwig, "Memetic Algorithm for Web Service Selection," Proceedings of the 3rd workshop on Biologically inspired algorithms for distributed systems, 2011.
- [9] T. Rajendran and Dr. P. Balasubramanie, "An Efficient Architecture for Agent based Dynamic Web Services Discovery with QoS," Journal of Theoretical and Applied Information Technology, Pakistan, Vol. 15, No. 2, 2010.
- [10] Z. Xu, P. Martin, W. Powley, F. and Zulkernine, Reputation-Enhanced QoS-based Web Services Discovery, In Proceedings international Conference on Web Services, 2007
- [11] G. Ye, "A QoS Aware Model for Web Service Discovery," First International Workshop on ETCS, 2009.
- [12] K. Toutanova, D. Klein, C. Manning, and Y. Singer, Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network, In Proceedings of HLT-NAACL 2003
- [13] E. M. Maximilien and M.P. Singh, "A Framework and Ontology for Dynamic Web Services Selection". IEEE Internet Computing, Vol. 8(5), pp.84-93
- [14] G. A. Miller. WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11 (pp). 39-41.

[15] B. Upadhyaya, F. Khomh, Y. Zou, A. Lau, J. Ng, A concept analysis approach for guiding users in service discovery. In SOCA, 2012 5th IEEE International Conference on (pp. 1-8).

[16] D.A. Menasce, QoS Issues in Web Services. IEEE Internet Computing 6(6),2002

[17] S. Kalepu, S. Krishnaswamy, and S.W. Loke, "Reputation = f (User Ranking, Compliance, Verity)", In International Conference in Web Services, 2004

[18] Y. Liu, A. Ngu, and L. Zheng, "QoS Computation and Policing in Dynamic Web Service Selection", In World Wide Web, 2004

[19] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke, M. Xu, "Web Services Agreement Specification (WS-Agreement)", September 2005

[20] H. Q. Yu, and S. Reiff-Marganiec, Non-functional Property based service selection: A survey and classification of approaches. In: Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop co-located with The 6th IEEE European Conference on Web Services, 12 - 14 Nov 2008, Ireland, and Dublin.

[21] K. Nahrstedt, D. Xu, D. Wichadakul, B. Li, QoS-Aware Middleware for Ubiquitous and Heterogeneous Environments. IEEE Comm. Magazine 39(11), 2001

[22] W. Al-Masri and Q.H. Mahmoud, Investigating Web services on the World Wide Web. In International World Wide Web Conference, 2008.

[23] P. D. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews", In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL '02)

[24] ProgrammableWeb, [http:// www.programmableWeb.com](http://www.programmableWeb.com)

[25] Seekda, <http://www.seekda.com>

[26] E. W. Forgy, Cluster analysis of multivariate data: efficiency vs interpretability of classifications. Biometrics 21, 1965

[27] J. A. Hartigan, and M. A. Wong, A K-means clustering algorithm. Applied Statistics 28, 100-108, 1979

[28] P. J. Rousseeuw, Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis". Computational and Applied Mathematics 20: 53-65, 1987

[29] Quality of Service, <http://www.w3.org/Architecture/qos.html>

[30] B. Upadhyaya, Y. Zou, I. Keivanloo, and J. Ng, "Quality of Experience: What users Say About Web Services?", 21th IEEE International Conference on Web Services (ICWS), June 27 - July 2, 2014, Alaska, USA

[31] B. Sabata, S. Chatterjee, M. Davis, J. J. Sydir, and T. F. Lawrence. Taxonomy for QoS Specifications. In Workshop on Object-Oriented Real-Time Dependable Systems (WORDS), Newport Beach, CA, Feb. 1997. IEEE Computer Society

[32] L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang. QoS-Aware Middleware for Web Services Composition. IEEE Transactions on Software Engineering, 30(5):311-327, May 2004. 221

[33] M. Tian, A. Gramm, T. Naumowicz, H. Ritter, and J. Schiller. A Concept for QoS Integration in Web Services. In Proc. of the First Web Services Quality Workshop, Rome, Italy, Dec. 2003. IEEE Computer Society.

[34] V. Poladian, D. Garlan, M. Shaw, and J. P. Sousa. Dynamic Configuration of Resource-Aware Services. In Proc. 26th International Conference on Software Engineering (ICSE 2004), pp. 604-613, Edinburgh, Scotland, May 2004. IEEE Computer Society.



Bipin Upadhyaya received his PhD degree in 2014 from Queen's University. He is currently doing a post-doctoral fellow in Department of Electrical and Computer Engineering in Software Re-engineering lab in Queen's University. His research interest includes service-oriented architecture, user service composition and cloud computing.



Iman Keivanloo received the PhD degree in 2013 from Concordia University, Canada. He is currently a Post Doctoral Fellow in the Department of Electrical and Computer Engineering at Queen's University. His research interests include source code similarity search, clone detection and their applications for software evolution and maintenance.



Ying Zou is a Canada Research Chair in Software Evolution. She is an associate professor in the Department of Electrical and Computer Engineering and cross-appointed to the School of Computing at Queen's University in Canada. She is a visiting scientist of IBM Centers for Advanced Studies, IBM Canada. Her research interests include software engineering, software reengineering, software reverse engineering, software maintenance, and service-oriented architecture.



Joanna Ng is currently the Head of Research at IBM Canada Software Laboratories, Center for Advanced Studies. She is also a Senior Technical Staff Member of IBM Software Group. In her current position, she directs the organization to conduct industrial research in collaboration with academic researchers, IBM researchers and software technologists.