Quality of Experience: What End-users Say About Web Services?

Bipin Upadhyaya, Ying Zou, Iman Keivanloo
Queen's University
Kingston, Canada
(9bu, ying.zou, iman.keivanloo)@queensu.ca

Joanna Ng
IBM Toronto Lab
Markham, Canada
jwng@ca.ibm.com

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 the high quality participants for service composition. The state of the art in service discovery and selection rely on non-functional aspects also known as quality of service (QoS) *e.g.*, response time and availability. Though these parameters are crucial for selecting web services, they do not reflect the end-user's perspective on quality. In this paper, we explore the feasibility of adopting the perceived quality from end-user's perspective for service selection and composition. We name such quality parameters as quality of experience (QoE). First, we propose a solution that automatically mines and identifies QoE parameters 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 24,000 reviews from 22 different services from four domains. Our result shows the automated approach identifies QoE attributes with an average precision and recall 90% and 79% 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.

Keywords - service composition, quality of service, quality of experience

1. Introduction

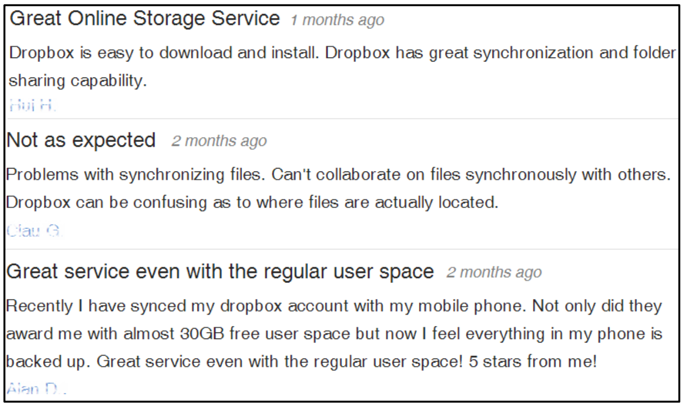
Service oriented architecture (SOA) provides a mechanism to publish and receive various forms of information through standard protocols. A common technology for SOA implementation is web services. Al-Masri *et al.* [22] report that there is more than 130% growth in the number of published web services.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

QoS-based service selection approaches [1, 2, 3, 6 and 16] focus on proposing comprehensive pre-defined QoS languages to describe service requests and offers, or implements 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. Static release of QoS information is conducted by service providers. 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 dominant 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]. This issue threatens the applicability of QoS-based service selection approaches [1, 2, 3, 6, and 16].

An alternative source of information about the quality of web services is online reviews available on the web. User oriented content generation approach of Web 2.0 has enabled people to broadcast their knowledge and experience to the mass. Online user review is an example of such a phenomenon. End-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 which measures customer satisfaction with a service. QoE attributes are extracted from online reviews reflecting user feedback on web services. Extracting QoE attributes from user reviews is challenging. 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 also explore the relationship between QoS and QoE attributes. Finally, we study if QoE can replace QoS for service selection in a case of insufficient QoS information. We present the result of our study in the following two 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 do QoE attributes relate with QoS attributes?
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 provided.



Great Online Storage Service *1 months ago*

Dropbox is easy to download and install. Dropbox has great synchronization and folder sharing capability.

Hui H.

Not as expected *2 months ago*

Problems with synchronizing files. Can't collaborate on files synchronously with others. Dropbox can be confusing as to where files are actually located.

Clau G.

Great service even with the regular user space *2 months ago*

Recently I have synced my dropbox account with my mobile phone. Not only did they award me with almost 30GB free user space but now I feel everything in my phone is backed up. Great service even with the regular user space! 5 stars from me!

Alan D.

Figure 1: Sample reviews of an online storage provider

The remainder of this paper is organized as follows: Section 2 gives an overview on the concept of QoE for 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. Quality of Experience

Quality of Experience (QoE) is a subjective measure of end-user's experience with a service. A service has different aspects, *e.g.,* cost and performance, for which an end-user can express her opinion. Each aspect of a service is called QoE attribute. Contrary to QoS, QoE reflects quality from the end-user point of view. The primary source of QoE is online reviews. Reviews come from users with diverse platforms and different geographical locations. Hence it is more 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 the service. The first user tells his experience with synchronization and folder sharing capability. The second user expresses her dissatisfaction with cross platform support. These attributes can be directly mapped to QoS attributes such as performance and cross platform support.

Users use natural language to provide their feedback. In a posted review, a user may mention more than one quality attribute of a service. Without an automatic aggregation and search tool, finding and going through a large number of reviews to manually find QoE information for service selection and composition is not feasible. Therefore as the first step an automatic approach for mining QoE from user reviews is required. Ideally, such approach analyzes the natural language content, identifies QoE attributes, and represents them in a structured way that can be used by service composition algorithms.

3. Our Approach to Extract QoE Attributes

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 three 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 store them in a 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 webpages. 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 review typically comprises of 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, our goal is to identify QoE attributes and a user's opinion about the QoE attributes. It is not trivial to determine the opinion orientation of such a review as a whole. 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 at a sentence level. A QoE has two key data fields which are attributes (*e.g.,* streaming) and opinion about the attributes (*e.g.,* unreliable). For an unseen service, neither its QoE attributes nor users' opinions is known in advance. In this section, we describe our approach for dynamically identify both QoE attributes and opinions from user reviews.

3.2.1 Tag POS in Reviews

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,

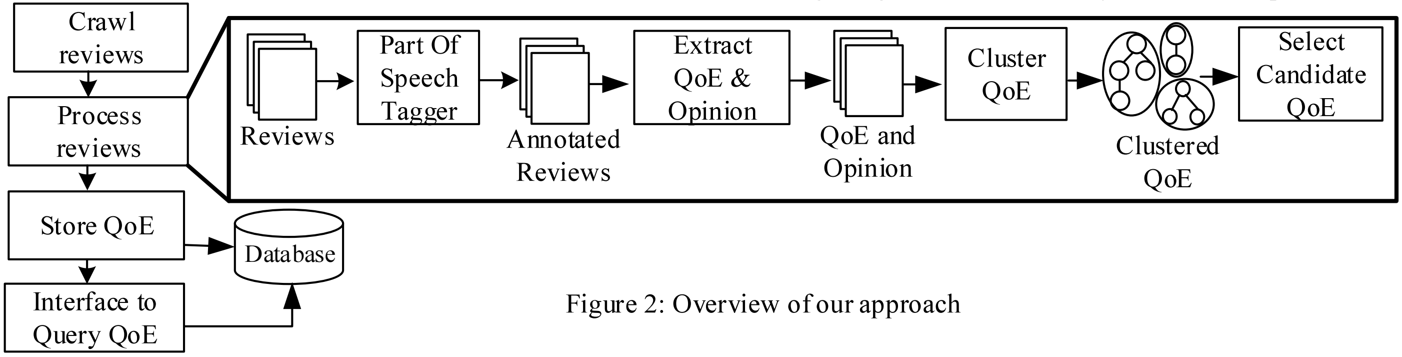


Figure 2: Overview of our approach

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.

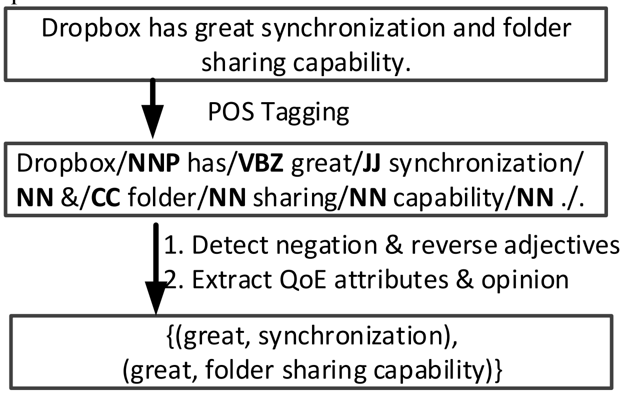

Dropbox has great synchronization and folder sharing capability.
POS Tagging
Dropbox/**NNP** has/**VBZ** great/**JJ** synchronization/ **NN** &/**CC** folder/**NN** sharing/**NN** capability/**NN** ./.
1. Detect negation & reverse adjectives
2. Extract QoE attributes & opinion
{(great, synchronization), (great, folder sharing capability)}

Figure 3: Extracted QoE attributes and opinion based on POS

3.2.2 *Extract QoE Attributes and Opinion*

We represent the extracted reviews as shown in equation (1) and transform extracted QoE information to a model 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.

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

Where, body is the text content of a review by a user on a specific date. QA and R is the quality attribute and its rank provided by the user. TV is the overall value 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 use Turney *et al.* [23] to extract two consecutive words from a sentence based on a predefined list of 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 noun. The third pattern means that two consecutive words are extracted if they are all adjectives, but the following word is not 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 extraction information is stored as a tuple shown in equation (2).

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

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 end-users. In this paper, QoE can be quantitatively scored in range [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 effects of opinion on a QoE attribute.

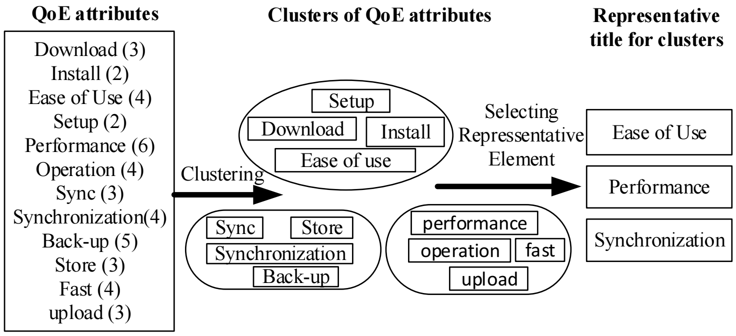


Figure 4: Process of clustering QoE attributes and selecting a representative element

3.2.3 *Clustering QoE Attributes*

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. An extensive list of QoE attributes and opinions of QoE attributes 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 end-users experience. Our aim in this step is to group similar QoE attributes together and summarize the opinion on 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, and 27] which is an unsupervised clustering algorithm. *K-means* algorithm divides the data into a set of disjoint groups.

The main challenge in using such a 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 for 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 measured success rate for our actual clustering step in Figure 4.

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

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.

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 Representatives

In this step, we identify a representative QoE attribute for each cluster of QoE attributes. 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 attribute 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; synchronization 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 the 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. 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 in automatic QoE extraction, and 2) to measure the correlation between QoS and QoE attributes, to observe if we can use QoE for service selection.

Table 1: Services used in the case study

Domain	Agg	#Services	#Sentences in Reviews	#Sentences with QoE & Opinion
Trip	Yes	Expedia, Tripit, Hotwire, Belair, Cleantrip, Ebookers, Yahoo Travel	7428	6980
Shopping	Yes	Amazon,eBay, Best Buy,Zappos, Checkout,Discfoo	6306	5866
Storage	No	Dropbox, Sugar Sync, Google Drive, Sky Drive Box	7033	6611
Mapping Service	No	Yahoo Maps, Google Maps, Bing Maps, Open Street Maps	4529	4110

4.1 Data Collection and Processing

We collect reviews for web services from four different domains: 1) trip (*e.g.,* CleanTrip and Ebookers), 2) shopping (*e.g.,* Amazon and eBay), 3) storage (*e.g.,* Dropbox) and 4) mapping service (*e.g.,* Google Maps) as shown in Table 1. Services in the first two domains are aggregator (*i.e.,* Agg in Table 1). 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 1 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.

We also manually created a gold dataset for the QoE attributes available in our dataset in order to evaluate the performance of our proposed approach in Section 3. In our case study, the first and third authors inspected all the data to create the gold dataset for QoE attributes. Our evaluators have two years of experience in developing services and composing services. To create such oracle, 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.

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 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.

4.2 Research Questions and Results

In this section, we outline our research questions and our approach to answer the research questions and the findings.

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

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

Approach. We use precision and recall in order to measure the effectiveness of our approach on identifying quality of experience (QoE) attributes. We compare the QoE attributes with the gold standard. 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 existed 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 oracle for services in Table 2 which is created manually as part of our case study setup.

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

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

Finding. Table 2 summarizes the result of the evaluation analysis step on the effectiveness of our proposed approach in extracting QoEs automatically. We compared the extracted QoE attributes with the manually made oracle that covers all QoE attributes from four 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 90% meaning that our approach can correctly identify the QoE attributes. The recall is above 79% 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 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 2, 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). Due to space limitation, we only show eight most frequent extracted attributes and plotted the values for past 13 months in Figure 5. Figure 5 shows each service has its own weak and strong aspects. Allowing a user to specify QoE attributes helps to filter a more suitable service. Figure 5 also shows another crucial aspect which is the change of quality over time. Our result shows four different types of trend. The first trend is the user's sentiment of a particular QoE attribute remains almost constant over a period of time as in the case of ease of use and cost QoE attributes. The second trend is a QoE attribute of a service is high during the initial phase, and it slowly demises and again gains the user's confidence as seen in case of Dropbox and Google Drive for media streaming quality attribute. The third kind of trend is slow and steady rise or fall in the user's confidence. The rise of the third kind of trend is seen in mobile access quality attribute of skydrive and sugarsync services. The fall is seen in file sharing attribute for Box service. The fourth trend is the oscillating trend, which is high for certain duration and low in certain duration as seen in the case of google drive in Figure 5. Interestingly, our further investigation revealed that the trends reported by our opinion mining approach are consistent with official QoS information. In summary, our result shows the feasibility to automatically extract QoE information, and such information capture dynamic and domain dependent aspects of a service.

Table 2: 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

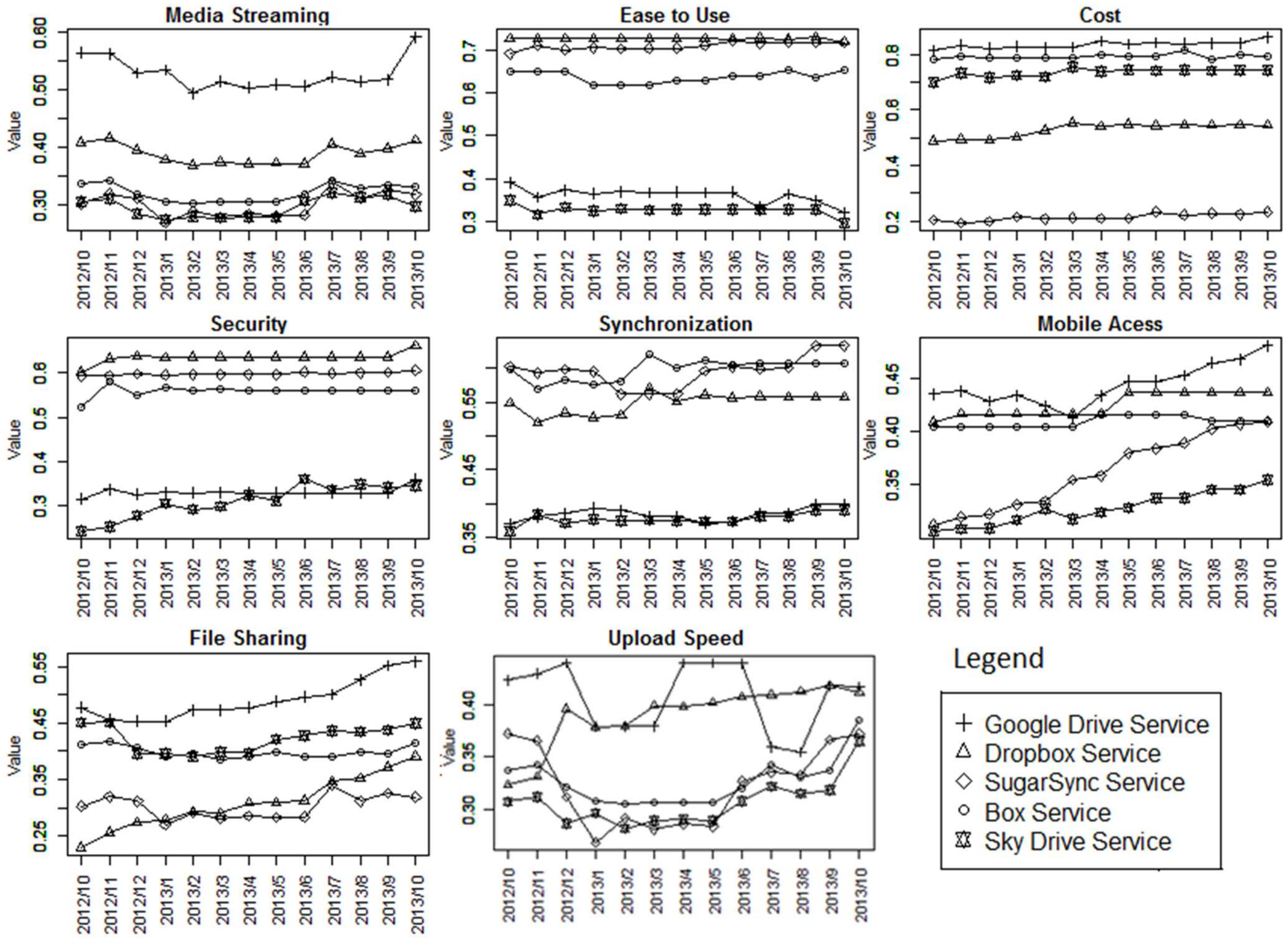


Figure 5: Eight most frequent QoE attributes of online storage providers over a period of 13 months

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 clients whereas QoE is directly based on user's feedback. The process of collecting QoS related information is tedious, time consuming and difficult to collect at client side [22]. In this research question, we explore the option of using QoE during service selection process. Since our approach can be automated, and it is independent from service providers. 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 for which we have QoE attributes. QoS attributes such as cost and security, is extracted from service providers web page, whereas QoS attributes such as upload speed is measured by writing a client program that calling service API. We manually mapped different QoE attributes to corresponding QoS attributes. For example, QoE synchronization, QoE upload speed and QoE media streaming in storage domain are mapped to QoS performance.

Table 3: 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	**0.01**	-	-	-
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	-	-	-

To study if the opinions expressed via 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 discovered five QoS attributes (Performance, Availability, Usage Limit, Security and Cost) in reviews. 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, similar kind of information for web services were not freely available. As security, the QoS information available were 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 and mapping services, as all of the services in those domains are free.

Table 3 lists the absolute value of correlation, fitness and p-value of related QoS and QoE attributes. Our study shows a high correlation between QoS and QoE attributes except in the case of QoS attribute cost in storage domain. Performance and usage limit attributes are highly correlated. For performance attribute, all the service statistically significant as their p-value is larger than 0.05. Similarly, QoS and QoE attributes for the availability for services in storage and mapping service are highly correlated with p-value less than 0.05. We found the availability of shopping service and the availability of travel services do not have the level of correlation as other QoS attributes. We went and re-analyzed the sentiment related to availability for shopping service. We found 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 only cost related QoS in Table 3 is online storage, and there was almost no correlation between the cost QoS collected and the sentiment of QoE attribute cost. When we analyzed the reviews related to cost for online storage, we found most of the sentiments were related to the free storage space rather than commercial plan of storage. We then try to find the correlation between free space by a service provider and the sentiment of QoE cost. Our analysis shows a correlation between QoE attributes cost and free space is 0.946, and the fitness value is 0.895. Hence reviews and comments on online storage were based on free storage rather than average storage. Our approach shows strong correlation between QoE and QoS attributes indicating the possibility of using QoE attributes for service selection.

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 24,000 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 two authors and is arguable whether a particular attribute is a QoE attribute or not.

6. Related Work

The problem of QoS-based web service selection and composition has received a lot of attention by many researchers. Nahrstedt *et al.* [10] proposed a QoS middleware infrastructure which required a build-in tool to monitor quality metrics automatically. Their approach needs to poll all web services to collect quality metrics and the willingness of service providers to surrender some of their autonomy. Most of the existing approaches use the generic QoS parameters for web service discovery such as response time, reliability, availability and cost [2, 4, 6, 7, and 8]. In [5] Ran extends 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 helps 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.

Maximilien *et al.* [13] propose an agent framework and ontology for dynamic web services selection. 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 for service selection. Our study also shows the correlation between traditional QoS attributes and QoE attributes extracted. Our approach uses the previous work in feature extraction and sentiment mining to find the meaning embedded in the service reviews that are expressed in natural language.

7. 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 QoE identification system are not high, in real life scenario, most of the services have a sizable amount of reviews, and hence even a moderate recall could result in a representative feedback. Our study shows our approach can identify all the QoS information discussed in 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. In the future, we will perform a user study to show the effectiveness of QoE attributes in a service composition process. We did not find enough review data available for new and unpopular services. We would like to extend our approach to address bootstrapping problem for QoE attribute identification.

8. References

[1] L.Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, Q.Z. Sheng. Quality Driven Web Services Composition. In Proceeding of WWW, 2003.

[2] G. Canfora, M. Di Penta, R. Esposito, M.L. Villani. An Approach for QoS-aware Service Composition based on Genetic Algorithms. In Proceeding of GECCO 2005.

[3] Canfora, M. Di Penta, R. Esposito, F. Perfetto, M.L. Villani. Service Composition (re)Binding Driven by Application-Specific QoS. In Proceeding of ICSOC, 2006.

[4] A. Esuli, F. Sebastiani. SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. In Proceeding of Conference on Language Resources and Evaluation, 2006.

[5] S. Ran. A model for web services discovery with QoS. In Proceeding of SIGecom Exch. 4, pp 1-10, 2003.

[6] M. Rathore and U. Suman, “A Quality of Service Broker Based Process Model for Dynamic Web Service Composition,” Journal of Computer Science, Vol. 7, 2011.

[7] R. Khorsand, M. Esfahani “Reputation Improved Web Service Discovery based on QoS,” Journal of Convergence Information Technology (JCIT), Vol. 5, No. 9, 2010.

[8] S. A. Ludwig, “Memetic Algorithm for Web Service Selection”, In Proceedings of 3rd workshop on biologically inspired algorithms for distributed systems, 2011.

[9] T. Rajendran and 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 ICWS,2007

[11] G. Ye, “A QoS Aware Model for Web Service Discovery,” In Proceeding 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 & M.P. Singh, “A Framework and Ontology for Dynamic Web Services Selection”. In Proceeding of 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 Proceeding of SOCA, 2012.

[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 Proceedings of ICWS, 2004

[18] Y. Liu, A. Ngu, and L. Zheng. QoS Computation and Policing in Dynamic Web Service Selection. In Proceedings of WWW, 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)", Open Grid Forum. Vol. 128. 2007.

[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 ECWS, 2008.

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

[22] W. Al-Masri and Q.H. Mahmoud, Investigating Web services on the World Wide Web. In Proceeding of WWW, 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, 2002

[24] ProgrammableWeb, http:// www.programmableweb.com

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

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

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

All URL were last accessed on 1st December 2013