Hindawi
Mobile Information Systems
Volume 2022, Article ID 2373596, 12 pages
https://doi.org/10.1155/2022/2373596

Research Article

An Effective Bootstrapping Framework for Web Services Discovery Using Trigram Approach

G. Senthil Kumar,[1] Kadiyala Ramana ,[2] Rasineni Madana Mohana ,[3] Rajanikanth Aluvalu,[2] Vinit Kumar Gunjan,[4] and Ninni Singh [4,5]

[1]*School of Computing, SRM University, Chennai, India*
[2]*Department of Information Technology, Chaitanya Bharathi Institute of Technology, Hyderabad 500075, India*
[3]*Department of Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, Hyderabad 500075, India*
[4]*Department of Computer Science and Engineering, CMR Institute of Technology, Hyderabad 501401, Telangana, India*
[5]*University of Science and Technology Chattogram, Chittagong, Chattogram, Bangladesh*

Correspondence should be addressed to Ninni Singh; ninni.singh@ustc.ac.bd

Received 9 April 2022; Revised 25 April 2022; Accepted 17 May 2022; Published 3 June 2022

Academic Editor: Saqib Hakak

Web services are progressively being used to comprehend service-oriented architectures. Web services facilitate the integration of applications and simplify interoperability. Additionally, it assists in wrapping accessible applications in order for developers to access them using standard languages and protocols. The user faces a difficult challenge in selecting the appropriate service in accordance with the user request as the behavior of the participating service affects the overall performance in discovery, selection, and composition. As a result, it is critical to select a high-quality service provider for these activities. Existing approaches rely on nonfunctional qualities for discovery and selection, but the user cannot always rely on these features, and these QoS values cannot be used to determine the user's or quality perspective. Additionally, the user indicates an interest in a high-quality service based on quality attributes or service with a good reputation throughout the selection process rather than a newly registered service. As a result, a proper bootstrapping mechanism is required to evaluate newly registered services prior to their use by service requestors. This paper proposes a novel bootstrapping mechanism. The contribution of this paper involves (a) a method for evaluating the quality of service (QoS) by focusing on performance-related indicators such as response time, execution time, throughput, latency, and dependability; (b) a methodology for evaluating the QoE attributes based on user reviews that take into account both attributes and opinions; (c) bootstrap the newly registered service based on quality of service and quality of experience; and (d) building a recommender system that suggests the top-rated service for composition. The evaluation results are used to augment currently available online services by providing up-to-date quality of service and quality of experience attributes for discovery, selection, and composition.

1. Introduction

Web services [1] offer a methodical and extensible framework for application-to-application communication, built on the existing web protocols and based on open XML standards. It simplifies the procedure by defining a consistent mechanism to describe, discover, and communicate with web applications. Figure 1 shows the three major entities in the Web services model: A provider is a person or organization that provides a Web service for a particular business purpose. A requestor is a person or organization that seeks to use a provider's Web service to meet business requirements. A broker, or discovery agency, acts as a matchmaker between the Web services provider [1] and requestor.

The web services framework is categorized into three areas: (i) communication protocols; (ii) service descriptions; (iii) service discovery. The simple object transport protocol (SOAP) [2] is used for communication between web services. SOAP works on existing transport

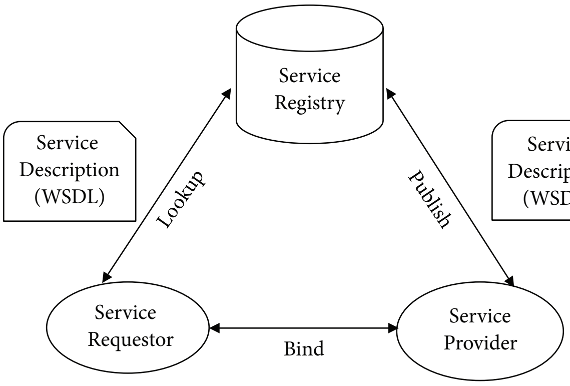


Figure 1: Publish-find-bind model (4).

protocols, such as Hypertext Transfer Protocol (HTTP) and Simple Mail Transfer Protocol (SMTP). The communications usually happen between the service provider, service requestor, and the registry. Web Services Description Language (WSDL) [3] provides a recognized, computer-readable description of Web services. WSDL follows XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information.

The Universal Description, Discovery, and Integration Directory (UDDI) [4], which is a registry of Web services descriptions, defines a standard method for various enterprises to dynamically discover and invoke web services. UDDI provides two basic specifications that define a service registry's structure and operation: (i) a definition of the information about each service and how to encode it; (ii) a query and update API for the registry that describes how this information can be accessed and updated. SOAP API is used for querying and updating the registry.

UDDI contains three types of information about web services: (i) White pages; (ii) Yellow pages; (iii) Green pages. White pages consist of the name and contact details of the service provider, whereas yellow pages consist of information on business and service types; the technical information about the services is described in Green pages.

UDDI models are Centralized [4] so that the performance will be lower if there are too many services to be registered or queried. UDDI stores all kinds of service information and is not classified into different types. It will affect search efficiency UDDI and thus can easily become passive; i.e., if the providers do not publish their Web services in registries, clients will not be able to find them. Universal Business registries (UBR) are incapable of providing quality-of-service (QoS) measurements for registered Web services.

2. Related Work

2.1. Need for Quality of Service and Quality of Experience in Web Services. Quality of service measures is essential for service selection. The service requestor searches the service through the registry. Since there are many services that address the same functionality, the process of selecting the correct service is a tedious task for the user [5]. In this scenario, either he has to compare every service and select the best service according to his requirement, which is time-consuming, or he has to rely on service provider claims.

The quality claims registered by the service provider are not always reliable [6]. The service providers could publish false and inaccurate information about the services, and if the web service information is not updated periodically by the service provider, the service will become passive and outdated. A service provider may also embellish by providing good QoS to fascinate customers to get additional profits. So, a proper validation mechanism is needed to assess the QoS values. Zou et al. proposed a ranking framework [7] based on service selection and the services are ranked based on the QoS values given by the service provider. Li et al. proposed an exponential function [8] to revise the quality of service data from the service providers since the past runtime data is not sufficient to judge the services. Ludwick et al. from IBM proposed a Web service level agreement [9] as a specification for SLA language to define and monitor QoS parameters. The specifications are termed as agreements that occur between a service requestor and provider and are used whenever there is a deviation in the service level agreements. WSLA includes the parties involved, obligations sections, and service definitions, which are too complex for the service providers to use in their applications. Degwaker et al. proposed a QoS broker [10], which is accountable for publishing and discovering the QoS information's at the time of communication happens between the service provider and the requestor.

Tian et al. [11] proposed a method of publishing the QoS information with the service descriptions in the UDDI. Zhang et al. proposed a computational model [12] based on QoS for web service selection in which the suitable service is recommended to the service requestor from the QoS registry. The process of getting the QoS information can assess through static release and runtime monitoring [13]. If the static release is not updated frequently, the information becomes passive or outdated. The runtime monitoring is done from the client side to evaluate the QoS information, which is time-consuming and expensive.

An alternate parameter used to measure the quality of web services is from the online reviews. A various number of users from different geographical locations expressed their satisfaction through the reviews. A satisfied user positively gives his views, and a dissatisfied user expresses the reviews negatively. Since the user experiences the product or service and expresses his views in the form of reviews, we can measure the quality of service through these reviews.

The majority of the services or products have a sufficient number of reviews expressed by the users, and based on these reviews, effective grading can be calculated and given to the users. The end users often prefer the service which has the highest grading rather than trusting the service provider's claims.

The quality of experience can be measured by (i) objective and (ii) subjective methods. The objective methods can be computed from the characteristics of the services, such as response time, throughput, etc., and it is determined at the time of invocation of the services, whereas the subjective methods come from the user perceptions, which are obtained by the monitoring process [14]. The subjective methods monitor subjective data and the nature of the service to be measured.

Omar Radwan proposed the quality of experience of web services (QoEWS) [15] which focuses on modeling, monitoring, measurements, prediction, and optimization. He proposed conceptual and technical artifacts for each aspect. Geerts et al. proposed a framework for predicting and measuring the quality of experience QoE information [16] in information and communication technology. Bipin et al. proposed an approach that identifies the QoE attributes and aggregates accordingly [17]. The authors have mined reviews and identified the QoE attributes for service selection. Zainab Alazzaf proposed a bootstrapping framework and QoS model [18], which monitors the available services and calculates the QoE values.

2.2. Quality of Service. Quality of Service (QoS) covers an ample variety of techniques that matches the needs of service requestors with those of the service provider's based on the network resources available.

Using QoS, nonfunctional properties of web services such as performance, reliability, availability, and security [19] can be assessed. For selecting a service, the performance attribute plays a vital role [20]. The service requestor searches his query by sending a request and waits for the response. The user will be satisfied if he receives the response in a shorter period of time. The performance is measured by execution time, throughput, response time, and latency [21]. Higher throughput, lower latency, lower execution, and faster transaction time represent good performance of web services [22].

2.3. Quality of Experience. From the end user perspective, accessing the service according to the QoS attributes alone is not sufficient to judge. The practical difficulty is (i) what happens if a service which has good performance and reliability fails to satisfy the expectations of the users. (ii) How to trust the QoS descriptions which are given by the service providers are reliable. (iii) How to validate the information given by the service providers. (iv) The service possesses good QoS values, but the cost is very expensive. Hence, if the service fails to meet the users' perspective, it is difficult to judge the services only by the QoS values.

In this paper, a combined approach is proposed for judging the top rated service using QoS and QoE. User reviews are collected from various domains, and QoE attributes are identified for selection and composition. The proposed framework monitors the performance of web services and processes the user reviews so that the service with the highest grading, with good performance, reliability, and low cost suggested to the user. In addition to that, a bootstrapping framework is proposed for the newcomer or the newly registered services and assesses and rates them accordingly.

3. Framework Using QoS and QoE

In this paper, we have proposed a framework using Quality of Service and Quality of Experience. Figure 2 represents the proposed recommender system which accepts a query from the user and fetches the services available from the registry. Since the QoS claims are not reliable [5], the services are monitored for a period of time, and the corresponding QoS values are calculated and maintained in the registry. Parallelly, the reviews are collected and processed, and the top rated service which possesses the high quality of service and with good service rating are suggested to the user. For this purpose, we have chosen web services domains such as weather services, map services, trip services, online calculator services, e-book services, education services, and product services. In these vertical domains, the following services are taken for analysis. Table 1shows the domains and their services used as datasets.

If a user wants to search the weather service, the services which are related to weather are collected, and the QoS values are calculated, such as response time, execution time, throughput, and latency calculated and the service which has the highest throughput, petite response time with high reliability service is offered to the user.

Additionally, the reviews given by the different users about the weather service are collected and processed, then the service which has the highest rating is suggested to the user. So that the QoS values with the user experience (QoE) about the services are calculated, and the best service is recommended to the user.

In some domains, the QoS values are not needed for the users, such as educational services, for the reviews processed and for top services categorized in the registry. For a few services, there are limited numbers of reviews or no reviews. In such cases, the services are monitored, and the QoS values are calculated. So our recommender system suggests the suitable service needed by the user.

3.1. QoS Framework. The proposed framework consists of the following steps to collect the QoS values.

(i) Preprocessing: The services fetched from the registry and invocation happens, such as service type, domain, operation, etc.

(ii) QoS Evaluation: From the fetched services, the QoS values are monitored, and corresponding values such as response time and reliability are calculated.

(iii) Each quality attribute is associated with a metric. Using these metrics, the QoS values are calculated (Table 2). We have used the SOAP UI tool for invoking the service parameters for the SOAP-based web services.

Table 3 provides the data of available web services in the Weather and Map domains. These services are preprocessed and evaluated. The evaluation includes service monitoring and assessing of service metrics. The next step is that according to the user request, the service should bootstrap the values, i.e., if the user searches the weather services from

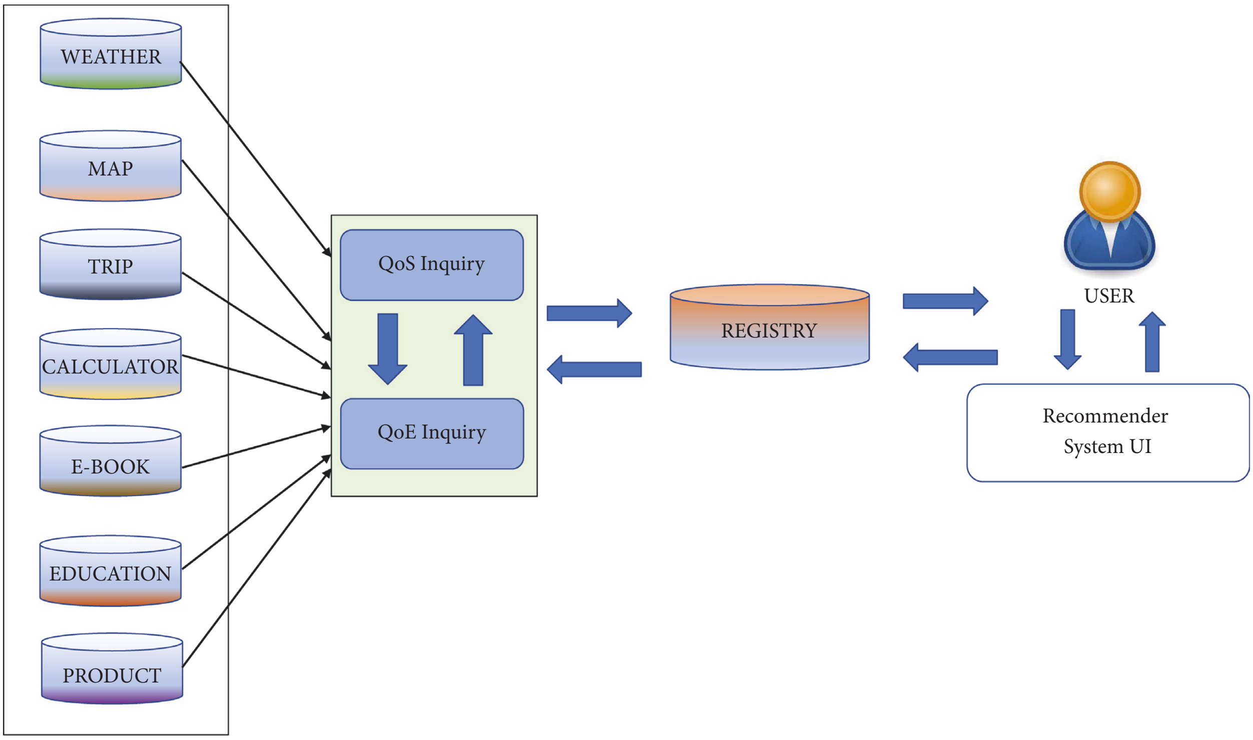


Figure 2: System overview.

Table 1: Domains and their services used as datasets.

Domain	Services
Weather services	Accuweather, BBCWeather, weather forecast
Map services	Google map, Yahoo map, bing map
Trip services	Yatra, MakeMyTrip
Calculator services	Online calculator, percentage calculator, web 2.0 calculator
E-Book services	Bookrix, Bookboon, Kobo Books
Education services	Coursera, Udemy, Edx
Product services	Paytm, Flipkart

Table 2: QoS attributes.

QoS attributes		Description	Measurement
Performance	Execution time	Time taken to execute and process the sequence of activities	Millisecond
	Throughput	No. of invocations in a given time	Invocation per seconds
	Response time	Time taken to send and receive the request and response	Millisecond
	Latency	Time taken (round trip time) between sending a request and receiving the response	Millisecond
Reliability		The ratio of the no. of error messages to the total messages	Percent
Accessibility		Successful service invocation at a point in time.	Percent

the recommender system, it compares with the available services and bootstraps the service, which has a shorter response time given in Table 4. Here, weather 5 service has a shorter response time of 119 ms when compared with all other services. The analysis is given in Table 4.

3.2. QoE Framework. Trust and reputation are to be considered for the newcomer services. The reputation [23] is the public opinion about the services, and collective evaluation is needed to judge the reputation, whereas the trust is a subjective reflection expressed by the user by his own experiences. Quality of experience is considered a subjective measurement that reflects user experience with the service [24–26].

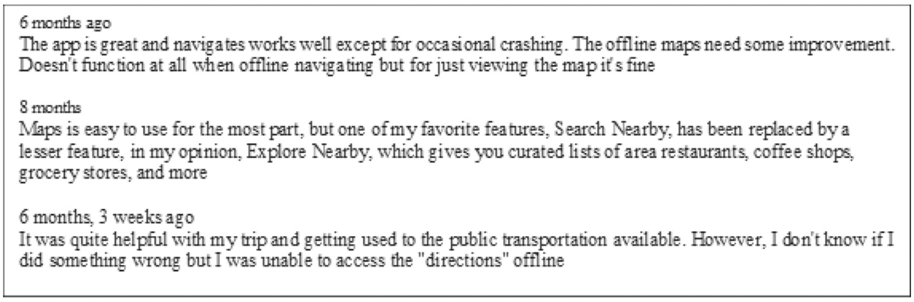
The user expresses his views on any feature of the service. Each feature expressed by the user is considered a Quality of Experience. Since many users are expressing their views from different geographical locations through reviews, it has been considered as the primary measure of assessing the QoE attributes.

Table 3: Observed QoS values for weather and map services.

Name	Service category	Latency (Ms)	Response time (Ms)	Execution time (Ms)	Throughput (Ms)	Reliability (%)
Weather 1	Weather service	49.4	342.8	293.4	1.111	94.111
Weather 2	Weather service	2.9	129.9	127	5.000	94.321
Weather 3	Weather service	1.8	162.7	160.9	1.000	95.482
Weather 4	Weather service	140.1	310.1	170	2.147	96.110
Weather 5	Weather service	1.7	119.8	118.1	2.147	95.515
Map 1	Map service	131.6	179.1	47.5	1.000	96.957
Map 2	Map service	12.6	132.9	120.3	2.147	94.062
Map 3	Map service	1.6	78.5	76.9	2.147	95.861
Map 4	Map service	33.3	173.3	140	1.000	94.663
Map 5	Map service	42.4	64.9	22.5	1.000	96.344

Table 4: Bootstrapped QoS values with respect to response time for weather services.

Name	Service_Category	Latency (Ms)	Response time (Ms)	Execution time (Ms)	Throughput (Ms)	Reliability (%)
Weather 5	Weather service	1.7	119.8	118.1	2.147	95.515
Weather 2	Weather service	2.9	129.9	127	5.000	94.321
Weather 3	Weather service	1.8	162.7	160.9	1.000	95.482
Weather 4	Weather service	140.1	310.1	170	2.147	96.110
Weather 1	Weather service	49.4	342.8	293.4	1.111	94.111



6 months ago

The app is great and navigates works well except for occasional crashing. The offline maps need some improvement. Doesn't function at all when offline navigating but for just viewing the map it's fine

8 months

Maps is easy to use for the most part, but one of my favorite features, Search Nearby, has been replaced by a lesser feature, in my opinion, Explore Nearby, which gives you curated lists of area restaurants, coffee shops, grocery stores, and more

6 months, 3 weeks ago

It was quite helpful with my trip and getting used to the public transportation available. However, I don't know if I did something wrong but I was unable to access the "directions" offline

Figure 3: Sample reviews of Google Maps.

Figure 3 shows the reviews given by the different users about Google Maps. The review contains the information provided by the people who have used the Google Maps application. They are expressing their satisfaction and dissatisfaction through these reviews. The QoE attributes can be identified from these reviews.

4. Extracting QoE Attributes

Users provide feedback through reviews and express more than one quality attribute in their reviews. So, the reviews should be extracted, and QoE attributes to be found and aggregated from the selected domains. Our approach consists of the following steps: (a) crawling huge amount of reviews from different portals for the different domains, (b) processing the reviews using natural language processing and trigram approach for identifying QoE attributes, (c) creating a database that stores all the relevant QoE attributes with representative titles, and (d) creating recommender system and providing an interface for web services selection and composition.

4.1. Collecting and Processing Online Reviews. Reviews are considered as a primary measure for analyzing the QoEattributes. We have crawled reviews from various portals, i.e., weather services, map services, trip services, online calculator services, E-book services, education services, and product services. From the collected reviews, the stop words and the special symbols are removed to make the sentence as structured. The QoE values are determined by two aspects as attributes and opinions. From the collected reviews, these fields should be identified and categorized.

4.1.1. Identifying Part of Speech (POS) and Tags in Reviews. A review is a collection of multiple words collectively called sentences. For example, in Figure 3, the user expressed his views about his satisfaction with the map application. But in the same review, he uses other sentences like “Poor navigation in offline” which relates to streaming and performance. So, for each review, target attributes and opinions are identified. The whole reviews are split into sentences, and positive and negative opinions based on user’s experience are assigned as follows:

Positive Words. Dazzling, brilliant, phenomenal, excellent, fantastic, gripping, riveting, spectacular.

Negative Words. Terrible, awful, hideous, bad, worst, stupid, waste, boring, Poor, Bad.

Neutral Words. Not bad, it is okay.

Natural language processing supports the definition of the part of speech (POS) of each word that occurred in a sentence. The English grammar classifies the POS into the following categories: conjunction (CC), preposition (IN), noun (NN), adjective (JJ), verb (VB), symbol (SYM), interjection (UH), and adverb (RB). We have used a part of speech tagger [27] to find the structure of the sentence.

4.1.2. Trigram Approach. A well-known trigram approach [28] was used to find the language similarity. A trigram is a method used for context-sensitive spelling error detection and correction based on three-word sequences. The main idea behind this is that the misspelling of a word often results in an unlikely sequence of (three) words. For example, someone misspells weather as whether in the sequence "Find weather information." Unlike weather, the term whether is not a noun or verb but rather is a conjunction which joins two words or phrases together. Here, the relative frequency of the trigram "Find whether information" is much smaller than the relative frequency of "Find weather information." We have used the trigram approach to correct an erroneous trigram by changing the relatively similar words into original words. The metrics are based on the notions of (i) word claims, (ii) character set claims, and (iii) sentence claims.

Moreover, when compared to the Unigram [29] and Bigram [30] approaches, trigrams produce more accurate results [31]. Collocations and phrases that express stronger [32]sentiment can be easily captured with trigram. More unique phrases can be formed by pairing and grouping with the trigrams approach, and hence in this work, the trigram approach is adopted to find the language similarity.

4.1.3. Extracting QoE Attributes and Opinion. The attributes and opinions were identified after performing the trigram approach. The next step is to perform occurrences of the attributes and opinions in the reviews because the user expresses similar kinds of words in their views. After finding the occurrence of attributes and opinions sentiment score has to be calculated to rate the services from the reviews.

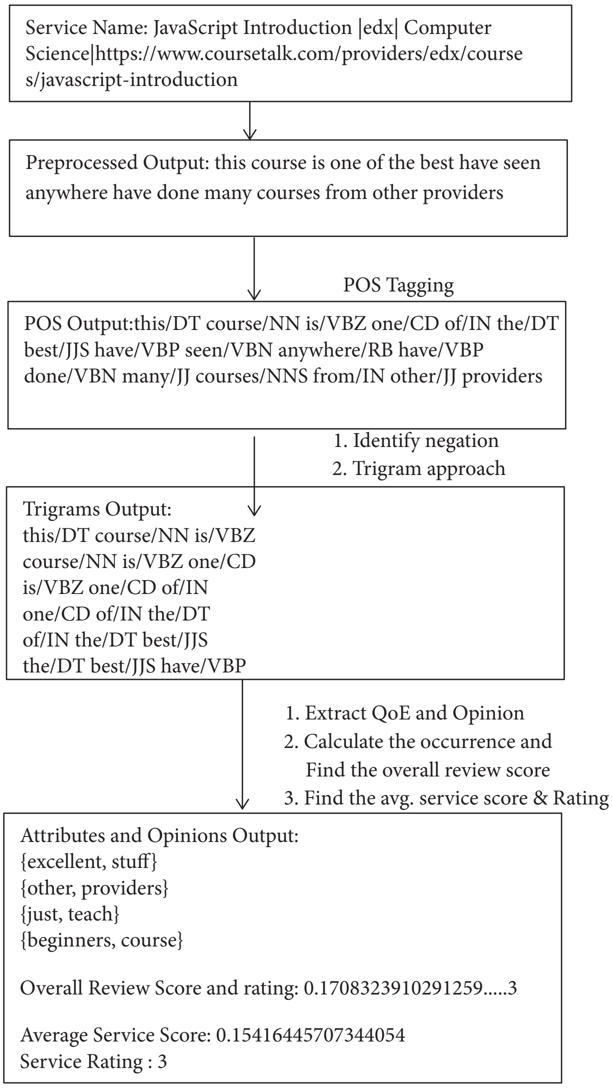
We have used Pointwise Mutual Information and Information Retrieval PMI-IR [33] algorithm for performing this approach. The pointwise Mutual Information is used to find the semantic orientation among two consecutive words, word_1 and word_2, which is defined [34] as follows:

p $(\text{word}_1$ & $\text{word}_2)$

PMI $(\text{word}_1,$ $\text{word}_2) = \log_2[-----------$

$p(\text{word}_1)\, p(\text{word}_2)]$

$(\text{word}_1$ & $\text{word}_2)$ is the probability that word_1 and word_2 cooccur, and if the words are statistically independent, then the probability of cooccurrence is given by the product p $(\text{word}_1$ &$\text{word}_2)$. Here, log is used for the amount of information acquired about the presence of one word with the observation of other.



Service Name: JavaScript Introduction |edx| Computer Science|https://www.coursetalk.com/providers/edx/courses/javascript-introduction

Preprocessed Output: this course is one of the best have seen anywhere have done many courses from other providers

POS Output:this/DT course/NN is/VBZ one/CD of/IN the/DT best/JJS have/VBP seen/VBN anywhere/RB have/VBP done/VBN many/JJ courses/NNS from/IN other/JJ providers

Trigrams Output:
this/DT course/NN is/VBZ
course/NN is/VBZ one/CD
is/VBZ one/CD of/IN
one/CD of/IN the/DT
of/IN the/DT best/JJS
the/DT best/JJS have/VBP

1. Extract QoE and Opinion
2. Calculate the occurrence and
Find the overall review score
3. Find the avg. service score & Rating

Attributes and Opinions Output:
{excellent, stuff}
{other, providers}
{just, teach}
{beginners, course}

Overall Review Score and rating: 0.1708323910291259.....3

Average Service Score: 0.15416445707344054
Service Rating : 3

Figure 4: Extracted QoE attributes with opinions.

4.1.4. Attribute Clustering. Once the attributes are found or mapped from the different phrases of reviews, sentiment scores are calculated, and a representative or suitable title is assigned to similar candidates in each group. For example, phrases in the reviews like operation, fast, response and upload related to performance.

These phrases from different reviews are grouped under the representative title called performance. Similarly, the related kinds of attributes are clustered, and representative titles are assigned from the reviews. Figure 4 shows the Extracted QoE attributes with opinions. The overall view is as follows:

(i) The QoE attributes mined from the body of the review by analyzing its POS.

(ii) Trigram approach used for finding the language similarity.

(iii) Based on the previous process, the attributes and opinions are identified.

(iv) The occurrence of attributes and opinions identified.

Table 5: Number of sentences with QoE and opinions.

Domain	Number of services used for analysis	Number of sentences in reviews	Number of QoE & opinion	Services used
Weather	10	2522	1800	Accuweather, BBC weather, weather forecast
Map	10	1446	1020	Google map, Yahoo map, bing map
Trip	8	2230	1733	Yatra, MakeMyTrip
Calculator	6	1180	840	Online calculator, percentage calculator, web 2.0
E-Book	8	1330	920	Bookrix, Bookboon, Kobo Books
Education	12	5600	4480	Coursera, Udemy, Edx
Product	10	2110	1600	Flipkart,Amazon

(v) Sentiment score and rating calculated using PMI-IR algorithm for each review.

(vi) Overall review score and service rating calculated.

(vii) Number of clusters and their representative titles are assigned

(viii) Total reconciliation (understanding) through Indigenous (original) attributes identified.

5. Experiments and Evaluation

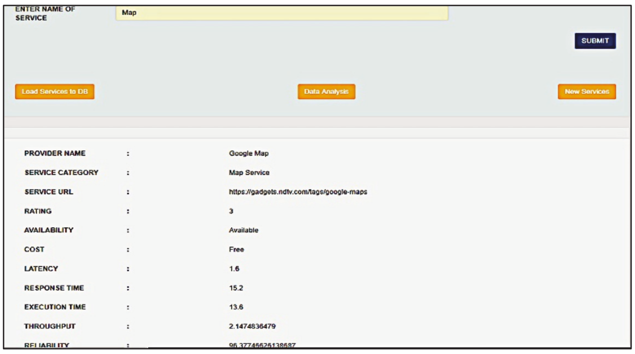
5.1. Data Collection and Processing. It is essential to gather and process information based on the stages listed below: (i) The recommender system gathers the query or service information from the user through the interface; (ii) The availability of the services is verified in the respective domains; (iii) The availability of the services is checked in the respective domains. (iv) It is necessary to analyze the reviews collected from each domain. (v) In order to discover the qualities and opinions contained within the reviews, specifically, we have gathered reviews from the following domains:

(vi) For each review, a sentiment score and rating are computed using the PMI-IR algorithm. (vii) An overall review score and service rating are calculated using the PMI-IR methodology. (viii) The number of clusters and the representative titles for each cluster are allocated. (IX) Total Reconciliation (understanding) is achieved by the identification of indigenous (original) characteristics.

Review data has been gathered from seven distinct domains, which are listed in Table 5. In order to call RESTful web services, a service invoker was developed utilizing the Java Development Kit (JDK), Eclipse 3.6, and an HTTP client.

We have tracked the response time, the execution time, the throughput, and the reliability of the services provided by the company. Details about the course and cost-related information are gathered from the service provider's description. The service information has been stored in a MySQL database, which we developed. This is accomplished by calculating service information, cost details, and ratings of the services, with the final ratings being kept in a database. We have developed a user interface to aid in the search process.

The user interface screen is depicted in Figure 5. Using the user interface, the user inputs his or her query, which is then processed by retrieving all available services from the relevant domains and listing them out as follows: (i) Service name with the highest rating (available services compared to the registry and the top-rated service given to the user), (ii) Service provider name, (iii) Domain Category, (iv) Service URI, (v) Overall rating, (vi) Availability, (v) Cost, (vi) Latency, (vii) Response time, (viii) Execution time, (ix) Throughput, and (x) Reliability of the service.



ENTER NAME OF SERVICE Map

PROVIDER NAME : Google Map
SERVICE CATEGORY : Map Service
SERVICE URL : https://gadgets.ndtv.com/tags/google-maps
RATING : 3
AVAILABILITY : Available
COST : Free
LATENCY : 1.6
RESPONSE TIME : 15.2
EXECUTION TIME : 13.6
THROUGHPUT : 2.1474836479
RELIABILITY : 96.37745625138687

Figure 5: User interface.

Table 6 shows the analysis of the monitored web services from the selected domains using our approach. The corresponding QoS values are identified and with the service ratings. We have found a correlation between QoS and QoE attributes. For example, Weather service 4 has the lowest response time and has scored three-star ratings from the user reviews. Preferably the service which has the highest response time will not be selected by the service requestor. Hence, the quality of experience attributes can be used as an alternative for the QoS attributes and vice versa.

6. Bootstrapping Framework

So far, in the above approach, the QoS values are monitored with their runtime performance with the service ratings. We have categorized these attributes in our registry according to the response time, throughput, availability, etc., because a random choice of selection of web services may not work and it is time-consuming for the user.

Hence, the service which has good QoS values such as low response time and high throughput with high reliability suggested to the user. We have also checked and measured the effectiveness of the Quality of Experience from the user

Table 6: Data analysis of the monitored web services using the proposed framework.

Name	Service category	Latency (Ms)	Response time (Ms)	Execution time (Ms)	Throughput (Ms)	Reliability (%)	Service rating
Weather 1	Weather service	49.4	342.8	293.4	1.111	94.111	2
Weather 2	Weather service	2.9	129.9	127	1.318	94.321	0*
Weather 4	Weather service	140.1	310.1	170	2.147	96.110	3#
Weather 5	Weather service	1.7	119.8	118.1	2.147	95.515	3
Map 2	Map service	12.6	132.9	120.3	2.147	94.062	2
Map 3	Map service	1.6	78.5	76.9	2.147	95.861	0*
Map 4	Map service	33.3	173.3	140	1.100	94.663	3
Map 5	Map service	42.4	64.9	22.5	1.100	96.344	3#
Calculator 1	Calculator service	0.3	189.8	189.5	1.100	95.624	3
Calculator 2	Calculator service	0.1	14.4	14.3	1.770	95.639	3
Calculator 3	Calculator service	73.2	84.6	11.4	2.147	94.929	3#
Calculator 5	Calculator service	77.2	101.4	24.2	2.323	96.586	0*
E_book 1	E_Book service	70.2	158.4	88.2	2.147	95.268	3
E_book 3	E_Book service	220.9	294.7	73.8	2.147	94.845	2*
E_book 4	E_Book service	185.3	431.6	246.3	2.147	96.475	0
Trip 1	Trip services	160	194.03	34.3	1.223	96.333	2*
Trip 2	Trip services	98	120.6	22.6	1.857	97.223	3
Trip 3	Trip services	174.23	208.05	33.82	1.524	93.226	0*
Product 1	Product services	188.3	247.6	59.3	1.121	96.66	2#
Product 2	Product services	164.3	627.5	163.2	1.356	98.33	0*

*New service; # Same service provider.

reviews and identified the QoE attributes and opinions and assigned the representative titles. The data analysis is given in Table 5. Based on the available services, our proposed approach monitored and calculated the QoS and QoE values. Figure 6 shows the bootstrapping framework.

But the real challenge depends on the new services because the new services have only the basic information like service description, cost, and the QoS given by the service provider. Hence, a proper validating mechanism is needed to assess the performance of the newcomer service. In our proposed approach, the new service is monitored, and the appropriate QoS values are calculated. For QoE, we are considering service provider reputation, availability of the service, and cost.

Here, the cost is considered as the key factor; a good quality service costs more than a low-quality service. In such cases, the consumer has to balance the cost with quality. Based on these parameters, the service is judged and compared with the available services in the registry and the top rating service recommended to the user. We have bootstrapped the values from Table 6, and the values are given in Table 7.

The framework will Bootstrap the QoS values and QoE ratings. For the newcomer services, we have bootstrapped using the service provider's reputation, cost, and accessibility of the services. The reason for selecting a service provider's reputation is a good service provider has accumulated trust and reputation from the previous services. The requestor or the consumer will believe the new service has good quality too. The newcomer service does not have the service score ratings. In such scenarios, the services are analyzed, and the service provider's reputation can be accounted (Algorithm 1).

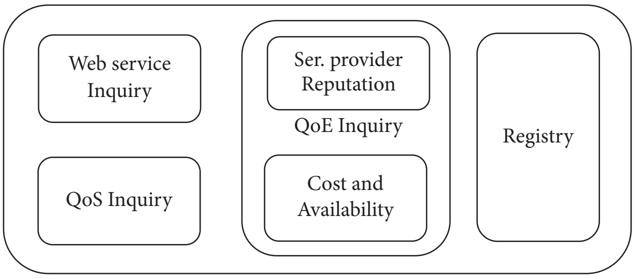


Figure 6: Bootstrapping framework.

Table 7 provides the bootstrapped values for the services. For each category, the topmost service is ranked and suggested by the recommender system. For the new services, the service provider reputations are assigned automatically, and the accessibility is calculated.

Figure 7 shows the bootstrapped values of the QoE services based on service providers' reputation and Figure 8 shows the accessibility rate of the bootstrapped services based on quality of experience. Accessibility can be measured through the number of successful invocations of the services. We have achieved a higher accessibility rate for the services. In Figures 9 and 10 the services with the shortest response time and latency are showcased. For example, if a user wants a service in the weather category, he will give the query through the user interface given in Figure 5. Based on the query, the various services are analyzed in the weather domain, and the service has a low response time with service ratings suggested to the user. In Table 7, the detailed bootstrapped values are presented.

Table 7: Data analysis of the Bootstrapped web services using the proposed framework.

Name	Latency (Ms)	Response time (Ms)	Execution time (Ms)	Throughput (Ms)	Reliability (%)	Service rating	Accessibility (%)	Cost
Weather 1	1.7	119.8	118.1	2.147	95.515	3	100	0
Weather 2	2.9	129.9	127	1.318	94.321	3*	98.89	0
Weather 3	140.1	310.1	170	2.147	96.11	3#	97.36	0
Weather 4	49.4	342.8	293.4	1.111	94.111	2	98.54	0
Map 1	42.4	64.9	22.5	1.1	96.344	3#	100	0
Map 2	1.6	78.5	76.9	2.147	95.861	3*	100	0
Map 3	12.6	132.9	120.3	2.147	94.062	2	98.63	0
Map 4	33.3	173.3	140	1.1	94.663	3	99.16	0
Calculator 1	0.1	14.4	14.3	1.77	95.639	3	100	0
Calculator 2	73.2	84.6	11.4	2.147	94.929	3#	88.6	0
Calculator 3	77.2	101.4	24.2	2.323	96.586	3*	99.54	0
Calculator 5	0.3	189.8	189.5	1.1	95.624	3	98.65	0
E-book 1	70.2	158.4	88.2	2.147	95.268	3	95.87	0
E-book 3	220.9	294.7	73.8	2.147	94.845	2#	97.36	0
E-book 4	185.3	431.6	246.3	2.147	96.475	2*	97.24	0
Trip 1	98	120.6	22.6	1.857	97.223	3	100	0
Trip 2	160	194.03	34.3	1.223	96.333	2#	94.98	0
Trip 3	174.23	208.05	33.82	1.524	93.226	2*	93.25	0
Product 1	188.3	247.6	59.3	1.121	96.66	2#	91.68	48
Product 2	164.3	627.5	163.2	1.356	98.33	2*	89.26	60

*New service; # Service provider reputation.

```
Input: User query
Output: Suitable service based on various parameter using service ratings
framework (web services, rt, tp, rel)
Boolean check availability (ws)//this will always
If (ws)
{
Calculate_rt; //Response time
Calculate_tp; //Throughput
Calculate_rel; //Reliability
Select (shortest rt & service reputation);
Choose_top rated service;
return best service
}
Else
{//new service or bootstrapping scenario
Monitor Qos
Where QoS = f (rt, tp, rel); --------- (1)
Compute QoS;
Compute QoE;//Quality of experience
Where QoE = g (spr, cost, acc); ------- (2)//service provider reputation, cost and accessibility
Using 1 and 2
Select best service;
Return best service;
}
```

Algorithm 1: Bootstrapping algorithm.

7. Web Services Composition

Web services composition [27] is the process of combining various services into a single service to perform one or more complex functions. Web service composition is categorized into two types: (i) static composition and (ii) dynamic composition. Static composition occurs during design time in which the user selects multiple services to do the transaction.

But one cannot always rely on static composition for the following reasons: (i) when frequent changes occur in the services, (ii) the selected services become unavailable. (iii)

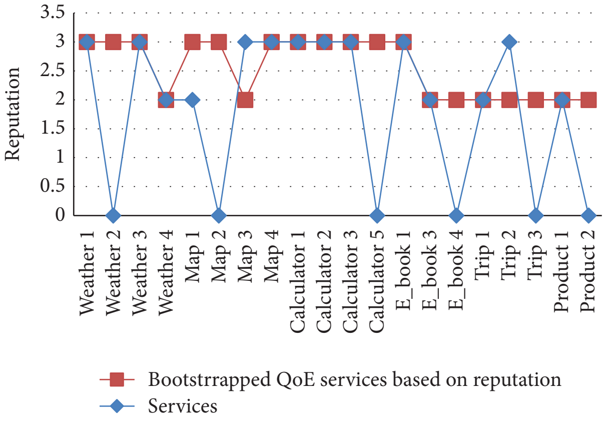


Figure 7: Bootstrapped QoE services based on reputation.

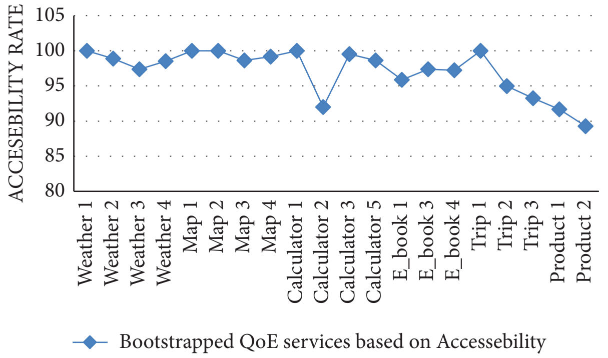


Figure 8: Bootstrapped QoE services based on accessibility.

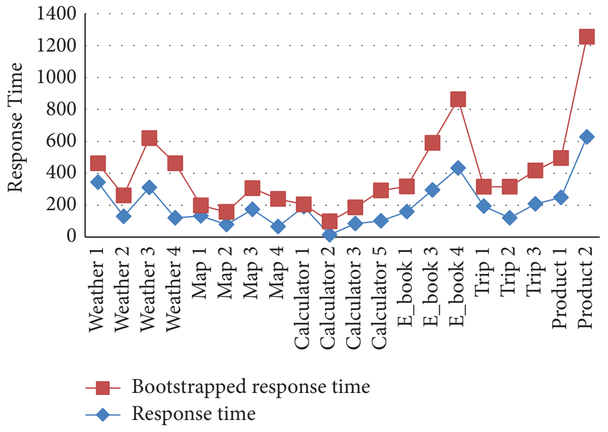


Figure 9: Bootstrapped QoS services based on response time.

When the services become passive or outdated in the registry, (iv) when the service moved to some other location whereas in dynamic composition the service selection and binding is automated.

Sheng et al. described the web service composition life cycle [28] which consists of four phases. The first phase is the definition phase, where the user specifies his requirement for composition and the second phase is the service selection phase, in which the appropriate service is fetched from the registry according to the user requirements. The third and fourth phases describe deployment and execution in which the service is invoked and executed.

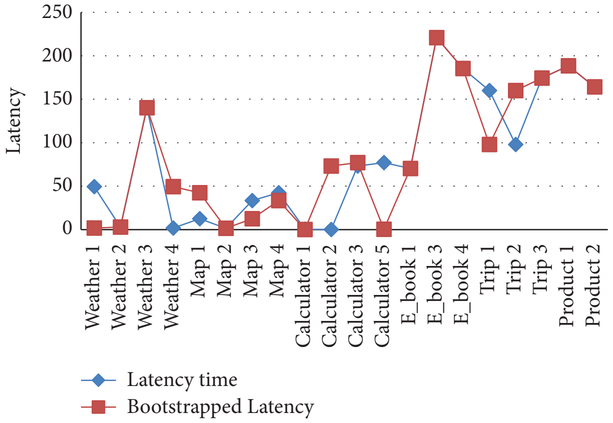


Figure 10: Bootstrapped QoS services based on latency.

Our approach mainly focuses on the definition and the service selection phase. Because the behavior of the participant service in service composition determines the overall performance of the transaction. Hence providing the right service with good Quality of service and Quality of Experience will be helpful to the user to do his transaction successfully, whereas selecting the services randomly will not help the user, and it is a time-consuming process. Our recommender system accepts a query from the user in the definition phase and parses the request. In the selection phase, it searches for the available services and provides the best suitable service which matches the user requirements.

8. Conclusion and Future Work

In this research, we proposed a recommender system that monitors and bootstraps the quality of experience and quality of service values. The evaluation results can be utilized to improve existing web services by giving an up-to-date quality of service and quality of experience attributes for discovery, selection, and composition. Our proposed approach includes a bootstrapping approach in which the newcomer service is assessed based on the QoS values, Service provider reputation and cost and accessibility. The proposed recommender system showed significant results in the process of identifying QoE attributes in the reviews.

In future, we will design a web service monitoring system that monitors the behavior of the services and their availability. The change in the requirement will cause different versioning of the web services and maintain all the versions of the services is a complex task. So in the future, we are going to develop an effective change management system that addresses this issue and smoothens the process of service selection to the user.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request (ninni.singh@ustc.ac.bd).

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

Authors' Contributions

G. Senthil Kumar conceptualized the study, performed data curation and formal analysis, proposed methodology, provided software, and wrote the original draft. Kadiyala Ramana supervised the study, reviewed and edited the manuscript, was responsible for project administration, and visualized the study. Rajanikanth Aluvalu supervised the study, reviewed and edited the manuscript, was responsible for project administration, and visualized the study. Rasinei Madana Mohana was responsible for data curation, investigated the study, was responsible for resources, and provided software. Vinit Kumar Gunjan provided software, validated the study, wrote the original draft, and proposed methodology. Ninni Singh supervised the study, reviewed and edited the manuscript, was responsible for funding acquisition, and visualized the study.

References

[1] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, M. Nirmal, and S. Weerawarana, "Unraveling the web services web an introduction to SOAP, WSDL, and UDDI," *IEEE Internet Computing*, vol. 6, no. 2, 2002.

[2] N. Mitra and Y. Lafon, "SOAP version 1.2 part 0: primer," 2007, 2nd edition, https://www.w3.org/TR/soap12-part0/.

[3] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web services description language (WSDL) 1.1," 2001, https://www.w3.org/TR/2001/NOTE-wsdl-20010315.

[4] "OASIS UDDI specification TC," https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uddi-spec.

[5] G. Senthil and C. Lakshmi, "A literature survey on web service discovery," *International Journal of Engineering and Technology*, vol. 8, 2016.

[6] E. Al-Masri and Q. H. Mahmoud, "Toward quality-driven web service discovery," *IT Professional*, vol. 10, no. 3, pp. 24–28, 2008.

[7] G. Zou, Y. Xiang, Y. Gan, D. Wang, and D. Liu, "An agent based web service selection and ranking framework with QoS," in *Proceedings of the International conference on Computer Science and Information Technology*, pp. 37–42, IEEE, Beijing, China, September 2009.

[8] Y. Li, M. H. Zhou, R. C. Li, D. G. Cao, and H. Mei, "Service selection approach considering the trustworthiness of QoS data," *Journal of Software*, vol. 19, no. 10, pp. 2620–2627, 2008.

[9] H. Ludwig, A. Keller, A. Dan, R. P. King, and R. Franck, "Web service level agreement (WSLA) language specification," http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf.

[10] S. Degwekar, S. Y. W. Su, and H. Lam, "Constraint specification and processing in web services publication and discovery," in *Proceedings of the 2004 IEEE International Conference on Web Services*, pp. 210–217, IEEE, San Diego, CA, USA, July 2004.

[11] M. Tian, A. Gramm, H. Ritter, and J. Schiller, "Efficient Selection and Monitoring of QoS-Aware Web Services with the WS-QoS Framework," in *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 152–158, Beijing, China, April 2005.

[12] J. Zhang, F. Zhong, Z. Yang, and H. Liu, "A QoS-aware computation model for dynamic web service selection," in *Proceedings of the 2012 IEEE 12th International Conference on Computer and Information Technology*, pp. 230–235, Chengdu, China, October 2012.

[13] D. A. Menasce, "QoS issues in Web services," *IEEE Internet Computing*, vol. 6, no. 6, pp. 72–75, 2002.

[14] R. Schatz, T. Hoßfeld, L. Janowski, and S. Egger, "From packets to people: quality of experience as a new measurement challenge," in *Data Traffic Monitoring and Analysis*, pp. 219–263, Springer, Berlin, Germany, 2013.

[15] O. Radwan, "An architectural model for managing the quality of experience of web services," in *Proceedings of the Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 513–518, Milan, Italy, July 2017.

[16] D. Geerts, K. Moor, and I. Ketyko, "Linking an integrated framework with appropriate methods for measuring QoE," in *Proceedings of the Quality of Multimedia Experience (QoMEX), 2010 Second International Workshop*, pp. 158–163, Norway, June 2010.

[17] B. Upadhyaya, Y. Zou, I. Keivanloo, and J. Ng, "Quality of experience: user's perception about web services," *IEEE Transactions on Services Computing*, vol. 8, no. 3, pp. 410–421, 2015.

[18] Z. Aljazzaf, "Bootstrapping quality of web services," *Journal of King Saud University-Computer and Information Sciences*, vol. 27, no. 3, pp. 323–333, 2015.

[19] A. Mani and A. Nagarajan, "Understanding quality of service for web services: improving the performance of your web services," https://www-128.ibm.com/developerworks/Webservices/library/ws-quality.html, 2002.

[20] K. W. Church and P. Hanks, "Word association norms, mutual information and lexicography," in *Proceedings of the 27th Annual Conference of the ACL*, pp. 76–83, ACL, Vancouver, BC, Canada, June 1989.

[21] K. Ramana and M. Ponnavaikko, "Web cluster load balancing techniques: a survey," *International Journal of Applied Engineering Research*, vol. 10, no. 19, Article ID 39983, 2015.

[22] K. Ramana and M. Ponnavaikko, "AWSQ: an approximated web server queuing algorithm for heterogeneous web server cluster," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 3, p. 2083, 2019.

[23] Ramana, M. Ponnavaikko, and A. Subramanyam, "A global dispatcher load balancing (GLDB) approach for a web server cluster," in *Proceedings of the International Conference on Communications and Cyber Physical Engineering*, pp. 341–357, Springer, Hyderabad, India, February 2018.

[24] A. Ahmed, A. R. Javed, Z. Jalil, G. Srivastava, and T. R. Gadekallu, "Privacy of web browsers: a challenge in digital forensics," in *Proceedings of the International Conference on Genetic and Evolutionary Computing*, pp. 493–504, Boston, MA, USA, July 2022.

[25] A. Alazab, M. Hobbs, J. Abawajy, A. Khraisat, and M. Alazab, "Using Response Action with Intelligent Intrusion Detection and Prevention System against Web Application Malware," *Information Management & Computer Security*, vol. 22, no. 5, pp. 431–449, 2014.

[26] M. Z. Asghar, F. Subhan, H. Ahmad et al., "Senti-eSystem : a sentiment-based eSystem -using hybridized fuzzy and deep neural network for measuring customer satisfaction," *Software: Practice and Experience*, vol. 51, no. 3, pp. 571–594, 2021.

[27] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology NAACL'03*, pp. 173–180, Edmonton, Canada, May 2003.

[28] A. Wilcox-O'Hearn, G. Hirst, and A. Budanitsky, "Real-word spelling correction with trigrams: a reconsideration of the mays, damerau, and mercer model," in *Computational Linguistics and Intelligent Text Processing. CICLing 2008*, A. Gelbukh, Ed., Springer, Berlin, Heidelberg, Gemany, 2008.

[29] L. P. Hung, R. Alfred, M. H. Ahmad, A. A. I. Ibrahim, and A. G. Ibrahim, "A review on the ensemble framework for sentiment analysis," *Advanced Science Letters*, vol. 21, no. 10, pp. 2957–2962, 2015.

[30] C. Liao, S. Alpha, and P. Dixon, "Feature Preparation in Text Categorization," in *Proceedings of the ADM03 workshop*, Geneva, Switzerland, May 2003.

[31] L. P. Hung and R. Alfred, "A performance comparison of feature extraction methods for sentiment analysis," in *Advanced Topics in Intelligent Information and Database Systems*, D. Król, N. Nguyen, and K. Shirai, Eds., Springer, Cham, Switzerland, pp. 379–390, 2017.

[32] Q. Z. Sheng, X. Qiao, A. V. Vasilakos, C. Szabo, S. Bourne, and X. Xu, "Web services composition: a decade's overview," *Information Sciences*, vol. 280, pp. 218–238, 2014.

[33] P. D. Turney, "Mining the web for synonyms:PMI-IR versus LSA on TOEFL," in *Proceedings of the Twelfth European Conference on Machine Learning*, pp. 491–502, Freiburg, Germany, September 2001.

[34] Y. Wang and J. Vassileva, "Toward trust and reputation based web service selection: a survey," *International Transactions on Systems Science and Applications*, vol. 3, no. 2, pp. 118–132, 2007.