Hindawi
Scientific Programming
Volume 2019, Article ID 5941096, 20 pages
https://doi.org/10.1155/2019/5941096

Research Article

An Intelligent Data Analysis for Recommendation Systems Using Machine Learning

Bushra Ramzan,[1] Imran Sarwar Bajwa,[1] Noreen Jamil,[2] Riaz Ul Amin,[3] Shabana Ramzan,[4] Farhan Mirza,[5] and Nadeem Sarwar[6]

[1]*Department of Computer Science & IT, The Islamia University, Bahawalpur 63100, Pakistan*
[2]*Department of Computer Science, FAST National University, Islamabad, Pakistan*
[3]*Faculty of Computing, BUITEMS, 83100 Quetta, Pakistan*
[4]*Department of Computer Science, Govt. Sadiq College Women University, Bahawalpur, Pakistan*
[5]*School of Engineering, Computer & Mathematical Sciences, Auckland University of Technology, Auckland, New Zealand*
[6]*Department of Computer Science, Bahria University, Lahore, Pakistan*

Correspondence should be addressed to Imran Sarwar Bajwa; imran.sarwar@iub.edu.pk

Received 27 May 2019; Revised 13 August 2019; Accepted 30 August 2019; Published 31 October 2019

Guest Editor: Aibo Song

In recent times, selection of a suitable hotel location and reservation of accommodation have become a critical issue for the travelers. The online hotel search has been increased at a very fast pace and became very time-consuming due to the presence of huge amount of online information. Recommender systems (RSs) are getting importance due to their significance in making decisions and providing detailed information about the required product or a service. To acquire the hotel recommendations while dealing with textual hotel reviews, numerical ranks, votes, ratings, and number of video views have become difficult. To generate true recommendations, we have proposed an intelligent approach which also deals with large-sized heterogeneous data to fulfill the needs of the potential customers. The collaborative filtering (CF) approach is one of the most popular techniques of the RS to generate recommendations. We have proposed a novel CF recommendation approach in which opinion-based sentiment analysis is used to achieve hotel feature matrix by polarity identification. Our approach combines lexical analysis, syntax analysis, and semantic analysis to understand sentiment towards hotel features and the profiling of guest type (solo, family, couple etc). The proposed system recommends hotels based on the hotel features and guest type for personalized recommendation. The developed system not only has the ability to handle heterogeneous data using big data Hadoop platform but it also recommends hotel class based on guest type using fuzzy rules. Different experiments are performed over the real-world datasets obtained from two hotel websites. Moreover, the values of precision and recall and F-measure have been calculated, and the results are discussed in terms of improved accuracy and response time, significantly better than the traditional approaches.

1. Introduction

In the modern era of advancing web technologies, the recommender systems (RSs) have turned the notice of the business society and the common man towards itself due to its significance and importance in the e-commerce and achievement of superior customer's approval. Nowadays e-commerce is believed to be strongly connected to the customer's satisfaction, and an ultimate success is always dependent on customer loyalty. The same is with the online booking and reservation systems being a main component of the tourism industry. Mariani et al. [1] discussed that the most powerful and popular industry which has a major impact on total GDP of world economy is tourism. Tourists around the world are always looking for best hotels for their residence during the tours which keeps the recommender systems as their primary choice to obtain best available hotel choices for online reservations well before reaching their

destinations in order to avoid any future residential trouble in hotels.

Liu et al. [2] have discussed that, in recent past, some recommender systems are made in order to facilitate tourists to get a list of hotel recommendations before making any booking. The nature of most of the data on Internet and web is heterogeneous becoming a hurdle for recommender systems because conventional recommender systems are dealing with homogenous data only which compromises the performance of hotel recommendation systems. Complex data in the multiple forms such as numeric, text, and visuals require developers' attention to develop recommender systems dealing with the heterogeneity of data. Li et al. [3] observed that recently few recommenders are available in the market having some capabilities to deal with heterogeneous data in which they have used ratings obtained from a customer feedback but do not include reviews, votes, ranks, and video views from the user feedback available on social media. In the proposed approach, we have also used multitypes of user feedbacks such as votes and YouTube video views. Our proposed recommender system gives two-fold novelty and advantage; first, it uses a hotel feature matrix to recommend a suitable hotel to a user on the basis of both quantitative (numerical) and qualitative (textual) features by using machine learning classification to achieve true recommendations; it mines user contextual information and extracts sentiments from reviews by analyzing the other travelers' reviews together with the ranks, votes, and YouTube video views to improve the recommendation accuracy. Second, a fuzzy module provides the recommendation of hotels in a particular type of user such as solo, family, business, friends, and couple because recommendations will be different based on type of user trip and user preference. Like for a family, "room," "food," and "cleanliness" facilities are the main preferences but for a single guest, facilities like "pool," "spa," and "gym" may have a greater preference. Similarly "WiFi" and "computer" can be an important feature for the user who is on a business trip.

Zhang and Mao [4] suggested that recommender systems are developed to achieve true and relevant recommendations. Relevant recommendation means the recommendation which is according to the customer's preference and choice. Usually, a recommender system uses customer ratings and reviews from the previous data considering the hotel's attributes or features. So the main challenge in this paper is to develop an intelligent approach which processes and analyzes large heterogeneous web data to achieve true hotel recommendations which are relevant to the customer's choice.

While dealing with diverse nature of data in our multifeature hotel recommendation system, the main challenge was the opinion mining/sentiment analysis of users' reviews to calculate a polarity score which represents the degree of likeness or dislikeness about a hotel by a user. A typical recommender usually banks on previous users' ratings about the hotel's attributes or features, but our proposed recommender also uses reviews, numerical rank votes, and video views to take true results of users' multitype feedback. The polarity score provides a textual side of user's opinions about a particular hotel. To handle the diversity of heterogeneous data as the presented approach uses both numeric data as well as textual data, a big data solution involving Hadoop was used in our approach because it efficiently handles data heterogeneity and data diversity in a better way. We have defined the guest type (solo, family, business, friends, and couple) as the main part of this research. We will not only consider different rating parameters but also apply feature based sentimental analysis on user's reviews. For example, Trip Advisor allows travelers to rate hotels on several options such as such as location, room, cleanliness, service, and staff. The process of extracting opinions from textual reviews is called as sentiment analysis/opinion mining.

There are number of studies present in the literature to perform sentiment analysis with the state-of-the-art methods to handle reviews to provide recommendations. Machine learning classification is the one of the most useful techniques for the sentiment classification of categorized text into positive, negative, or neutral categories. In machine learning technique, training and testing datasets are essential. A training dataset is used to learn the documents, and the test dataset is used to validate the performance. There are four main types of machine learning to classify reviews as shown in Table 1.

Within the field of data analytics, machine learning is part of a piece known as predictive analytics. Machine Learning algorithms are not series of processes serially executed to produce a predefined output. They are instead series of processes aiming to "learn" patterns from past events and build functions that can produce good predictions, within a degree of confidence.

Machine learning based-sentiment analysis or classification is used to classify and provide recommendations for the users. In supervised machine learning techniques, two types of data sets are required: training dataset and test data set. An automatic classifier learns the classification factors of the document from the training set, and the accuracy in classification can be evaluated using the test set. The key step in the supervised machine learning technique is feature selection. The classifier selection and feature selection determine the classification performance.

The main purpose of our recommender system is to provide suggestions and recommendations which are truly based on customer's preference and choice. Koren et al. [5] has focused on the quality of recommendations. It is suggested by the author that when a large number of user ratings and reviews are used to be processed to provide efficient and true recommendations, then quality of recommendations is significantly important [6]. Booking through online systems and recommenders has increased in recent years, and multinational organizations are working over this domain to take maximum advantage. A recommendation system (RS) helps customers not only finding appropriate hotels but also it is benefitting in all domains such as movies, books, and all sorts of other different products and items. Different types of data, i.e., hotels, movies, and music, can be processed by RS. The generic recommender architecture is presented in Figure 1. Hsieh et al. [6] have explained in detail that different recommender systems are built using different methods and

Table 1: Comparison of Machine learning approaches.

No.	Machine learning approaches	Description
1	Supervised learning	It uses previous data as input variable to predict the most probable output value for new data, depending upon those associations learned from the previous data sets. (i) *Regression* is a form of predictive modeling technique which examines the relationship between a dependent variable and an independent variable. (ii) *Classification* is the technique in which the algorithm learns from the data input given to it and then uses this learning to classify and produce new observation.
2	Unsupervised learning	It uses unlabeled data that have no historical labels to train the algorithm. The purpose is to find some structure within it by exploring the data. (i) Clustering groups a set of objects in such a way that objects in the same group are more similar to each other in some respect than to those in other groups. (ii) *Dimensionality reduction* removes useless data before analysis. This is used to remove redundant data and outliers.
3	Reinforcement learning	(i) With this approach, the algorithm discovers through trial and error which trials produce the best rewards. (ii) It is often used for gaming, navigation, and robotics.
4	Deep learning	(i) Deep Learning helps in training computers to deal with the problems that are not well defined. (ii) Deep learning and neural networks are often used in speech and image recognition applications.

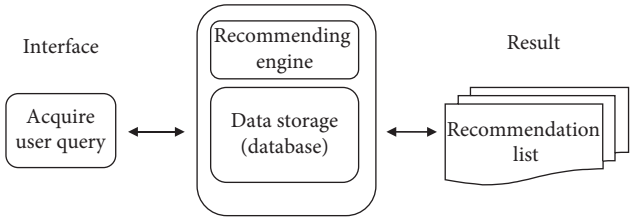

Figure 1: Generic architecture of a recommender system.

algorithms which uses customers' previous data consisting reviews and ratings about the different products to obtain true recommendations. Burke [7] has discussed about different types of recommendation algorithms. It explained that there are two recommendation techniques. First is collaborative filtering (CF), and the other is content-based filtering (CBF). Mixer of these two algorithms is called as hybrid filtering.

Lops et al. [8] have explained that the most widespread recommender filtering technique is collaborative filtering; however, users' preferences and choices are presented by their linked points in the content-based recommender system. CF works by collecting user ratings for items in a given domain and calculating similarities between users or items in order to provide relevant recommendations. Ekstrand et al. [9] elaborate that collaborative filtering uses a class of methods and utilizes preferences of other users which they have expressed for the same items to recommend items to the active user. This technique can also be beneficial in all other domains where the customer preferences can randomly change.

Collaborative filtering algorithms can be of two kinds, i.e., item-based recommender system and user-based recommender system. Item-based recommenders compare item similarities, and the user-based recommenders instead compare user similarities in the recommendation process.

According to Zhang et al. [10], collaborative filtering-(CF-) based approach is a very successful technology in all RSs. The papers [11, 12] haev reported that there are three fundamental challenges faced by CF approaches such as

(i) *Cold start* challenge rise where an item appears which has not been rated before, recommendations cannot be made for it or when a new user without any prerecorded profile appears [13, 14]

(ii) *Sparsity* challenge appears when there are numerous items but too less rating values available in the initial stage of recommendation [11, 14]

(iii) *Scalability* challenge appears when users' and items' data are very big to process [11].

Most of the recommender systems suffer from the cold start problem because users usually do not provide adequate ratings to hotels to enable collaborating filtering based recommendation, which can lead to an issue called as cold

start problem. We have proposed the hotel recommender system that mines contextual information and sentiments from reviews and recommend the travelers the name of the hotels based on their preferences, by analyzing the other travelers' reviews together with the rating value, ranks, votes, and YouTube video views to improve the recommendation accuracy. Opinion-based sentiment analysis resolves this issue by considering four types of contexts: (i) guest type, which can be "business," "couple," "solo," "group," and "family"; (ii) hotel name; (iii) location; and (iv) rating about different hotels. We have used the approach where collaborative filtering technique aggregates with the sentimental analysis, to provide personalized hotel recommendation. Opinion-based sentiment analysis calculates the polarity of each sentence of review to find its score effectively solving the problem of cold start and improving the accuracy.

Similarly, one cannot imagine manually sorting through thousands of comments, customer support conversations, or customer reviews, as there are too much data of hotels available to process. Sentiment analysis allows to process data at scale in an efficient and cost-effective way improving the scalability issue. Machine learning-based sentiment analysis or classification is used to classify and provide recommendations to the users. In the classification technique, the system learns from the data input given to it and then uses this learning to classify new recommendations. In supervised machine learning techniques, two types of data sets are required: training dataset and test data set. An automatic classifier learns the classification factors of the document from the training set, and the accuracy in classification can be evaluated using the test set. The key step in the supervised machine learning technique is feature selection. The classifier and feature selection determines the classification performance. Some researchers have introduced another approach known as cluster-based approach [6, 11, 15]. Collaborative filtering based on clustering reduces the computation time and focuses only on time efficiency improvement as the clustering phase is performed offline.

The main idea here is to develop a recommender system which helps users to find hotels according to their preference and choice using previous users' reviews and ratings. Due to the availability of extensive web data, the main problem while processing thousands of items and its related information is the storage and the time efficiency. In order to deal with this problem, an intelligent and efficient item-based collaborative filtering recommendation system is proposed which uses Hadoop platform along with NoSQL database to improve the performance and efficiency while dealing with a huge number of hotel data. We have performed various experiments to achieve performance gains of Hadoop with improved response time of the recommender resolving the scalability problem [11].

Previous researches lack accuracy of true recommendations. The reason for less accuracy is the use of only quantitative data such as likes and ratings which ignored the qualitative aspect of user feedback and challenge the reliability of the previous recommender's accuracy. Here, it is important to mention that both aspects of likeness of users can provide us with true and accurate recommendations. So there is a need to consider both quantitative as well as the qualitative aspects of multitype user feedback in the recommender. The both qualitative and quantitative aspects of likeness can be achieved by using not only ratings but also text reviews, votes, and video views that were not covered by the previous approaches. Some other gaps and deviations are also tabulated in Table 2. It signifies differences in the previous approaches and our approach.

The proposed system is capable of storing a large number of hotels data efficiently and provides improved time-efficient recommendations. This paper intends to provide four main contributions as stated in the following:

(i) The proposed recommender system helps in achieving true hotel recommendations through processing and analyzing of large heterogeneous web data, i.e., both ratings (numerical) and reviews (textual) using opinion mining approach and fuzzy approach to produce relevant recommendations according to customers' type and choice

(ii) Development of a web-based recommender for hotel recommendation which integrates linked data of external resources (hotel websites) containing hotel information available online

(iii) The proposed approach optimizes performance in the proposed hotel recommendation system using NoSQL Cassandra database in Hadoop environment

(iv) The dataset is obtained from two different sources (websites) such as TripAdvisor.com and Expedia.com

We are using opinion-based approach which is used to classify the text into three sentiment expressions such as "Positive," "Negative," and "Neutral" with the help of SentiWordnet Wordnet dictionaries. There were number of challenges that arise during the processing of reviews and extraction of the features from textual reviews. Some of the challenges are as follows:

(i) Dealing with the big data which consist of the textual reviews describing opinions given by the people for the hotels

(ii) Casual informal languages, abbreviation/emoji/slang, or use of emoticons

(iii) Spelling mistakes/typing errors

(iv) Ambiguous reviews given by a customer, e.g., I have never lived in a hotel quite like this one before! Ambiguity: we cannot understand whether the hotel is the best or the worst

(v) Reviews containing hashtags

(vi) Detecting polarities of hidden sentiments of a customer in a given review

The remaining paper is further organized and structured as follows. Related work and allied concepts are described in Section 2. Collaborative filtering and recommender system are discussed to cover the previous

Table 2: Deviation of different approaches.

Source	Multitype feedback	Ratings	Reviews	Votes	Video views	Polarity scores	Tf-Idf	Fuzzy logic	Multi data sources
[4]	✓	✓	✓	×	×	×	×	×	✓
[6]	×	✓	×	×	×	×	×	×	×
[16]	×	×	✓	×	×	×	×	×	×
[17]	✓	✓	✓	×	×	✓	×	×	×
[18]	×	×	×	×	×	×	×	✓	×
[19]	×	✓	×	×	×	×	×	×	×
Proposed approach	✓	✓	✓	✓	✓	✓	✓	✓	✓

work. A General recommender model is also explained. The methodology to design a hotel recommender is explained in Section 3. It explains the proposed hotel recommender components. The detailed description of preliminary experiments with testing and training datasets along with result is presented in Section 4. System overview is also provided in the same section. A comparative analysis of the proposed approach with the previous studies is presented in Section 5. Conclusion and future work are provided in Section 6.

2. Related Work

The previous work related to the recommender system is discussed in this section. Prior research describes the related concepts of recommenders such as information filtering and recommendation algorithms previously used to develop recommender systems [20] which help to understand and realize the need of recommenders in the modern era of web technologies.

2.1. Recommender Systems. Recommender systems help end users to help discover products and services that they are looking for. Tan and He [13] have proposed a physical resonance procedure, named resonance similarity (RES), as a novel approach. This novel similarity provides superior predictive accuracy in comparison to the traditional similarity measures used for users' evaluations. Fasahte et al. [21] have discussed that unrated items can be recommended and predicted by using different filtering techniques, and they have conducted experiments using Trip Advisor dataset. Additionally, they presented hybrid approach uses rating data and textual content to predict the user behavior. Crespo et al. [22] discuss that Sem-Fit uses the customers' experience point of view in order to apply fuzzy logic methods to relating customer and hotel characteristics, represented by means of domain ontologies and affect grids. Hu et al. [23] measured the fineness of rating predictions and evaluated the performance of the Context Aware Personalized Hotel (CAPH) recommender using ratings and reviews of Trip Advisor data.

The Hwang et al. [24] have used the Trip Advisor reviews in the semantic-based Latent Dirichlet Allocation (LDA) method to perform a hotel review for the hotel management systems and to obtain a distinguishable performance to the Term Frequency-Inverse Document Frequency (TF-IDF) method. The author has used all types of features of the hotels and concluded that the word-based LDA method has high precision compared to the LDA method. Sandeep et al. [25] performed twitter community sentiment analysis to obtain real-time sentiments of the common people to represent both existing and potential customers. The proposed method using monthwise sentiment score of twitter hash tags of Indian telecom operators successfully predicted their growth rate in terms of subscriber addition. Meng et al. [26] have proposed the method called KASR (Keyword Aware Service Recommendation) which is implemented in Hadoop and cloud for the big data analysis of reviews to improve the time efficiency and the scalability in big data projects [27, 28].

The text mining techniques combined with tracking and browsing are used to develop a personalized hotel recommendation by Lin et al. [29]. A useful stochastic programming model using multiple regression analysis was designed to lower the search cost of the customers by Rianthong et al. [30] It is concluded that the review rating, prices, and utility of the hotels are needed to be taken at the upper side of the sequence. Lal and Baghel [31] explore the most relevant and crucial features for sentiment classification and group them into seven categories, named as basic features, seed word features, TF-IDF, punctuation-based features, sentence-based features, N-grams, and POS lexicons. Sharma et al. [17] have used customer's reviews and preferences from booking.com to determine the hotel rating using previous users' data in a multicriteria review-based recommendation system approach and NLP technique. Chang et al. [16] hypothesized surrounding environments hotel recommendations. CATPAC (content analysis program) was used to analyze users' reviews and ratings, and SPSS (Statistical Package for Social Sciences) was used to analyze the users' ratings with regression analysis and analysis of variance (ANOVA) to check customer loyalty.

Jannach et al. [32] worked on the recommender system which utilized the regression-based methods and item-based models for accurate recommendations. Ibrahim et al. [33] presented a personalized intelligent information model to examine hotel services. Bouras and Tsogkas [34] have used the user clustering Word Net-enabled k-means algorithm to recommend improved news articles. Chen and Chuang [18] optimized the performance of a ubiquitous hotel recommender system by using a nonlinear and fuzzy programming approach over the hotels dataset. Fasahte et al. [21] have used a Lexicon-based approach to identify sentiments towards the hotel's aspects within the defined context, and they also explained how unrated items can be

used for recommendation using the item-based CF technique. Both rating data from the user's review and rating data of user's in a hybrid recommendation approach are used to analyze the user's behavior. Valcarce et al. [35] have used Cassandra as the platform of the distributed big data recommendation application and the MySQL Cluster for comparison.

User's previous data for the different products and items like hotels, books, and articles are gathered in the systems using collaborative filtering algorithms. Rankboost algorithm and cluster-based collaborative filtering are used to develop a hotel recommendation system proposed by Huming and Weili [11] to get recommendation according to users own choice. For the quantitative and qualitative analyses, the data were obtained from hoteltravel.com.

There are a number of recommender systems which are developed by researchers and developers who have used the collaborative filtering algorithms and techniques [36–42] to provide recommendation service. Collaborative and content-based filtering uses the knowledge of the users to calculate the correlation with other users and to perform certain deductions in the feature space [19, 43]. In this paper, the proposed hotel recommender system is highly efficient and somewhat bridges the gap using hotel feature extraction using natural language opinion mining analysis.

Nilashi et al. [44] have used PCA-ANFIS (Principal Component Analysis-Adaptive Neuro-Fuzzy Inference System) and EM (Expectation Maximization) to develop a recommender system and implemented them in the tourism domain based on multicriteria collaborative filtering technique. Data were obtained from TripAdvisor website to perform experiments and achieve high accuracy and high time efficiency for the recommendation of the hotel. Phorasim and Yu [45] have used k-means and collaborative filtering approach which results in more precise and less time-consuming recommendations as compared to the existing traditional one. Kögel [46] discussed that, to obtain relevant suggestions in real time, a model-driven software engineering approach is used which collects data from different sources and combines it. Do et al. [47] survey common techniques for implementing model-based approach so as to achieve high accuracy. Yibo et al. [48] have built a hybrid recommendation model for movie recommendation using sentiment analysis on spark platform which outperforms the traditional models in terms of various evaluation criteria.

3. Proposed Hotel Recommender System

The proposed system uses the heterogeneous nature of data (textual and numerical) crawled in from World Wide Web (www). Data are obtained from the selected hotel websites (data sources) containing the keywords present in the active user search query. A web crawler was used to download the requested data and store the obtained data in a NoSQL database Cassandra for further processing. The data are usually found in the form of numbers (such as votes, ranks, and number of video views) and text (such as reviews and comments). To get true recommendations, our system has used ranks, votes, and reviews data to extract hotel features from it.

The system works in two parallel ways. The numeric ranks and votes of hotels from each selected data source are normalized. On the other hand, review data are processed using natural language processing package for review mining, and features are extracted in the form of a hotel feature matrix. Further, numerical polarity scores are computed for these extracted features using SentiWordNet and average polarity score is calculated. Now weighted average polarity scores are calculated by aggregating normalized rank score, voting score, and polarity score. Finally, recommendations are computed by applying the fuzzy logic approach. We have defined a fuzzy set containing certain fuzzy rules to calculate the final score to find out the guest type (solo, family, business, friends, couple, etc.) for the hotel. The proposed hotel recommendation approach is shown in Figure 2. The final recommendations of the hotels based on a particular guest type in one of the five different classes are displayed.

3.1. Feature Extraction Process. First of all, before deriving reviewer's feature preferences, we first analyze raw textual reviews and convert them into structured form to extract opinion-based feature. Reviews from any website are usually extracted into a Json file which is then loaded into the system database. We have studied different types of methods for mining the feature-based opinions from textual reviews and found that NLTK package is most appropriate tool to extract features from hotel reviews. Natural language toolkit NLTK is a Python library to make programs that work with natural language. The library can perform different operations such as tokenizing, stemming, classification, parsing, tagging, and semantic reasoning. We have used NLTK 3.3 Version in this paper. Following steps have been performed for identifying numerous features from the hotel reviews:

(i) Extracting features from a review and grouping synonymous features

(ii) Finding and assigning value to the opinions that are associated with various features in the review

(iii) Assigning these features a value in the normalized range

The review data are converted in comma separated values to be available in easily readable form, i.e., natural language data format. The proposed system needs to perform a natural language processing to extract hotel features based on the previous guests' opinion. We have performed following four steps to process a natural language text review as follows:

(i) Lexical analysis

(ii) Syntax analysis

(iii) Semantic analysis

(iv) Feature extraction

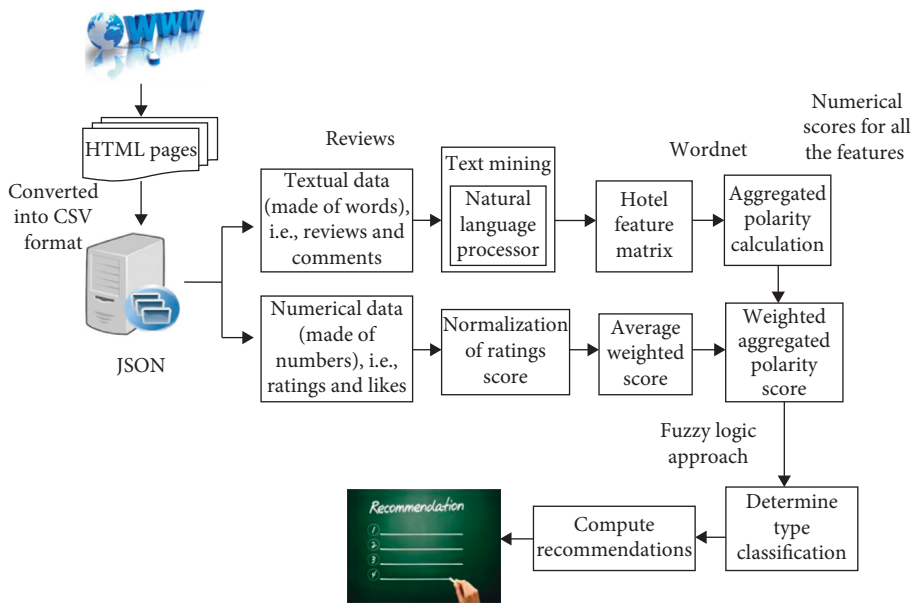

Figure 2: The proposed approach for hotel recommendation.

3.1.1. Lexical Analysis. In lexical analysis, the streams of characters are taken as input and streams of tokens are generated as an output.

(1) Tokenizing. The hotel reviews are available in the form of paragraph which contains a number of sentences or strings. These strings are tokenized into tokens or lexicons. These tokens are usually words but can also be numbers or symbols. Usually all whitespace characters are removed from the sentences, and alphabets or numbers are considered as a single token. These tokens further go through POS tagger to get different parts of speech called as morphemes. For example, a verb "feels" is stored as "feel + s" and a noun "vegetables" is stored as "vegetable + s." Afterwards, morphemes are lexically analyzed by a parse tree (Tables 3 and 4).

3.1.2. Syntax Analysis. In this step of analysis, all the sentences and the phrases of the paragraph of reviews are authenticated in consultation with the defined grammatical rules in the English language. This paper uses Google Spell Check to correct the grammatical errors and typos in the crawled reviews. The misspelled words in review text are corrected by using a statistical spell checker (http://norvig.com/spell-correct.html) and removing the duplicates and unnecessary punctuation marks in sentences, e.g., !, ?, etc. We have used the publicly recognized POS tagger to remove some noisy information contained in the review text such as syntactical errors and mistakes. The principal parts of all sentences are also identified in this phase, i.e., object part, subject part, and verb part. Parse tree and typed dependencies are also generated in this phase. Syntactic dependency parser (http://nlp.stanford.edu/software/lex-parser.shtml.) can also return the syntactic dependency relations between the words in a sentence.

Table 3: Lexical analysis of the review.

Review strings	Rooms of the hotel are big. Food is delicious. Hotel location is best. There is no Internet.
POS tagging	The/DT Room/NN of/IN the/DT hotel/NN is/VBZ big/NN./. Food/NN is/VBZ delicious/JJ./. Hotel/NN location/NN is/VBZ best/JJ./. There/EX is/VBZ no/DT internet/NN

Table 4: Parse tree of a review sentence.

```
(ROOT
    (S
      (NP
          (NP (DT the) (NN Room))
          (PP (IN of)
            (NP (DT the) (NN hotel))))
      (VP (VBZ is)
          (NP (NN big)))
      (. .)))
```

(a) Word stemming: there are many words which are derived from actual words called as derived words. Stemming is used to reduce derived words into their root forms. Lancaster Stemmer has been used in our work. Python NLTK provides WordNet Lemmatizer that uses the WordNet Database to lookup lemmas of words. The part of speech is first detected before getting the actual root word. In this process, the part of speech of a word is first determined and different normalization rules are applied for each part of speech.

(b) Extraneous word removal: reviews usually contain words which do not have significant meaning in extracting features for any product or item such as "the," "a,", "also," "about," "an," "at," "to," etc. These words are removed. There is not any globally approved library for list of stop words present in the

English language. To overcome this issue, we have developed library for such words in java of our own.

(c) Shorten exaggerated word.

(d) Words which have same letter repeating more than two times in a single word and not present in the lexicon are shortened to the meaningful word with the repeating letter occurring only once like exaggerated word "NOOOOOO" is reduced to "NO."

(e) Part of speech tagging: the words with similar grammatical properties are classified through part of speech tagging system. Each word in the review is separated and tagged according to part of speech it belongs to. The words are tagged as Singular Nouns, Plural Nouns, Verbs, Adjectives, Adverbs, etc. The NLP package returns tags like NN stands for singular common nouns, NP stands for singular proper nouns, etc.

3.1.3. Semantic Analysis. In the semantic analysis, all the tagged words among the sentences of the review are extracted in some sort of tabular form. It is decided in this particular stage that what actions are performed by the particular subject and a number of attributes related to every object are also identified. Output of the semantic analyzer as in Table 5 contains a semantic table which is generated by the input review text on the basis of the parse tree generated in the previous phase.

If there is a noun in the sentence, then we take it as a feature and store current sentence under this feature. The extracted information in the semantic table is used to express the features of the hotel. To find the polarity value for all the features from the user reviews using opinion mining approaches, a hotel feature matrix is obtained as in Table 6.

3.2. Polarity Detection. We identify the polarity of each review in the collection reviews using the NLTK library and calculate aggregated polarity score for each feature based on each review for every hotel from selected websites. We have started conducting feature-based opinion mining of every review, where opinion indicates positive, neutral, or negative sentiment that a reviewer expressed on a feature based on opinion words, as there are multiple opinion words (great, nice, and awesome) that are related with each feature (location, room, and food) in a review. We assess every opinion word's sentiment strength which is also called polarity value. In this analysis, the values of review features and their associated opinions in terms of polarity are derived and shown in Table 7.

3.2.1. TF-IDF Generation. After the preprocessing of textual reviews and removing all repetitive entries and unnecessary stop words, Tf-Idf (term frequency -inverse document frequency) needs to be generated for each review. We compute weights of each item in the review using term frequency-inverse document frequency (TF-IDF) technique to determine which terms might be the most representative and occur frequently in the collection of documents as well as which words are less representative and rarely occurring. TF-IDF is computed for each term word occurring in the collection of reviews. $\mathrm{TF}(t)$ is defined as follows:

$$\mathrm{TF}(t) = \frac{\text{number of times term } t \text{ appears in a document}}{\text{total number of terms in the document}}. \quad (1)$$

While IDF for a term (t) is given as follows:

$$\mathrm{IDF}(t) = \log \frac{\text{total number of documents}}{\text{number of documents consisting term } t}. \quad (2)$$

Now, we have to weigh down the frequent terms and find out the rare ones, by computing TF-IDF weight. The TF-IDF weight is the product of $\mathrm{TF}(t)$ and $\mathrm{IDF}(t)$:

$$\mathrm{TF-IDF\ weight} = \mathrm{TF}(t) * \mathrm{IDF}(t). \quad (3)$$

SentiWordNet is a dictionary that tells, rather than the meaning, the sentiment polarity of a review. For detecting the polarity and subjectivity of different hotel reviews and to get the polarity and subjectivity, we have used SentiWordNet, a publicly available analyzer of the English language that contains opinions extracted from a WordNet database. We separate our collection of reviews to extract words (hotel features) and assigned all representative occurring under the appropriate hotel features as explained in previous steps, find positive (pos), negative (neg), and neutral (neu) terms to

Table 5: Semantic analysis.

No.	Tagged words	Value
1	Room	Big
2	Food	Delicious
3	Location	Best
4	Internet	Not available

Table 6: Hotel x feature matrix after NL processing.

No.	Hotel-ID	Review-ID	Location	Price	Room	Food	Staff
1	Hotel-1	R-1	2	2	2	1	2
2	Hotel-1	R-2	1	1	1	2	1
3	Hotel-1	R-3	0	3	2	1	3
4	Hotel-2	R-1	2	0	2	1	2
5	Hotel-2	R-2	1	1	3	2	1
6	Hotel-2	R-3	2	2	2	1	2
7	Hotel-3	R-1	1	1	2	2	2
8	Hotel-3	R-2	0	1	1	2	1

Table 7: Polarity matrix.

No.	Review-ID	Worst	Great	Shame	Awesome	Nice	Label
1	R-1	2	1	2	1	1	Negative
2	R-2	1	1	1	2	1	Positive
3	R-3	0	3	2	1	3	Positive
4	R-1	2	0	2	1	2	Negative
5	R-2	1	1	3	2	1	Neutral

calculate the sentiment score. SentiWordNet is included with Python's NLTK package and provides WordNet synsets with sentiment polarity. WordNet gives different types of semantic associations between words, which are used to calculate sentiment polarities. In simple words, sentiment analysis is the process of quantifying something which is qualitative in nature such as textual reviews. The sentiment score of a term (pos or neg) is multiplied by TF-IDF weight to calculate overall sentiment score (polarity) of terms in the document and is given as follows:

$$\text{overall sentiment (polarity)} = \text{sentimentscore} * \text{TF} - \text{IDF weight}. \quad (4)$$

The overall sentiment polarity score (negative or positive) explains how many features are positively or negatively important in the hotel review. As there are multiple opinion words that are related with each feature in a review, a weighted average value is calculated which acts as the weight to represent the overall positive or negative polarization of the review. If the polarity score of a feature in the reviews of a hotel is greater than zero, then the feature is the positively polarized; if it is less than zero, then it is negatively polarized; and if it is equal to zero, then it represents the neutrality. We have calculated the polarities of all the reviews of the hotels taken from different data sources.

The polarity of the reviews Pr of a hotel from the selected website can be calculated by taking the difference of the aggregated polarity score of positive reviews posr and the aggregated polarity score of negative reviews negr of that particular hotel h_n from the particular selected website w_o:

$$\text{Polarity}_{rm}(P) = \text{sgn}\left[\left|\sum_{m=1}^{n}\left(\text{posr}_m\right)\right| - \left|\sum_{m=1}^{n}\left(\text{negr}_m\right)\right|\right], \quad (5)$$

where $\text{posr}_m \wedge \text{negr}_m \in h_n$ and $h_n \in w_o$.

Then, we will take aggregated polarity score of textual reviews of each hotel from each selected website. Aggregation is the process of combining things. That is, putting those things together so that we can refer to them collectively:

$$\text{aggregated polarity}_{hn}(A) = \sum_{m=1}^{n} P_{rm}. \quad (6)$$

The weighted average of the aggregated polarity by total reviews of the respective hotel from the selected website and the weight score of ranks and votes will be calculated as follows:

$$\text{weigted average polarity}_{hn}(B) = \frac{A_{hn}}{T} + \left(\text{aggregated ranks}_{wo}\right) + \left(\text{aggregated votes}_{wo}\right). \quad (7)$$

Here, T is the total number of reviews of hotel h_n from hotel website w_o:

$$\text{aggregated weighted average polarity}_{hn}(C) = \sum_{O=1}^{N}\left(B_{wo}\right) + \left(\text{Likes}_{hn}\right),$$

$$\text{average aggregated weighted average polarity}_{hn}(D) = \frac{C_{hn}}{N}. \quad (8)$$

where N is the total number of selected hotel websites containing large number of hotel reviews.

$$\text{Final score}_{hn}(F) = \frac{r_{hn} - \min\left(\text{r}_{\text{hn}}\right)}{\max\left(r_{hn}\right) - \min\left(r_{hn}\right)} * 10. \quad (9)$$

where F is the final value of the normalized average aggregated score r_{hn} of the hotel (h_n) from number N of selected websites.

3.3. Type Classification and Recommendation. The classification is done by calculating the final score. The reviews words are matched with the dictionary words, and if it is a positive word, then score will be +1; if negative word, then score will be −1, otherwise 0. The final recommendation is achieved by using the fuzzy logic approach. The fuzzy sets theory provides a framework for the representation of the uncertainty of many aspects of human knowledge. For a given element, fuzzy set theory presents the degrees of membership to a set. For example, if we have the set of solo guests, then we can consider that a person who likes to do gym or take massage in hotel belongs to such a set with a degree of 1 or a person who likes to have a ghazal night or cinema in the hotel must be a couple guest and belongs to a set in some other degree. The purpose of our recommender is to provide hotel recommendations based on some expert criteria using the fuzzy set. The first step of the recommendation process consists of representation of knowledge about how the hotels are selected. This knowledge is expressed using fuzzy sets.

In the first step, the fuzzy sets are defined based on the expert knowledge. The expert explains the characteristics of the hotels and the characteristics of the customers in terms of fuzzy sets. The second step consists of providing recommendations using a previously built hotel-feature-rating matrix which is usable by a recommender system based on a collaborative filtering technique.

3.3.1. Fuzzy Set. To represent the degree of membership of a certain hotel to a certain class, fuzzy set theory is used. The final recommendation is achieved by using the fuzzy logic approach based on the fuzzy rules by calculating the final score to provide the class of the hotel based on guest type (solo, couple, etc.) as shown in Figure 3. Fuzzy rules are defined as follows:

Rule 1: if $F > 8$ then Hotel Type is "R (Recommended)"

Rule 2: else if $F > 6$ and $F \leq 8$ then Hotel Type is "BR (Best Recommended)"

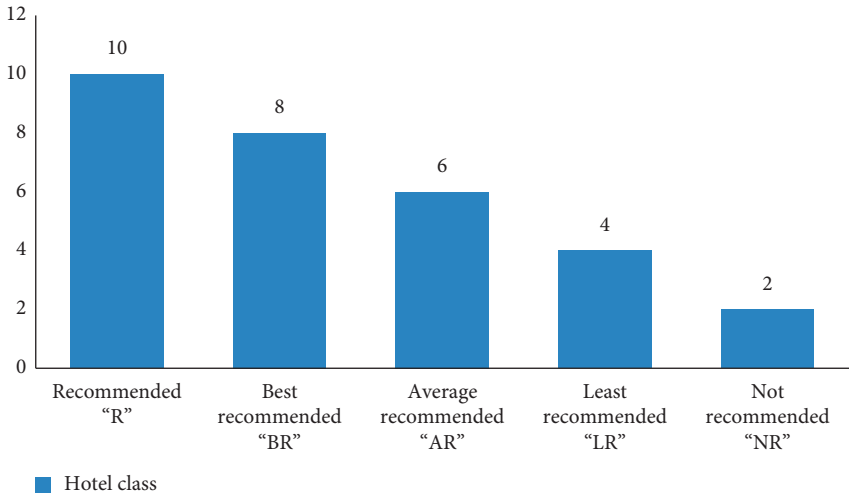

Figure 3: Hotel recommendation class.

Rule 3: else if $F > 4$ and $F \leq 6$ then Hotel Type is "AR (Average Recommended)"

Rule 4: else if $F > 2$ and $F \leq 4$ then Hotel Type is "LR (Least Recommended)"

Rule 5: else Hotel Type is "NR (Not recommended)"

4. Implementation Setup

4.1. NoSQL Storage. As our recommender is designed to deal with heterogeneous types of data, a database is required which can store this diverse nature of data. We have Cassandra database to store data used in the proposed recommender. Review pages which match the Query keywords are downloaded and are stored in the NoSQL database in Hadoop. The dataset used in this study is crawled in from the external resources such as hotel website of Trip Advisor and Expedia. The data of hotels are saved in comma separated value format (CSV). So, it is converted into the JSON format to increase its readability. The users' textual reviews and ratings assigned by existing users recorded as rating score, likes, or star ranks are stored in Cassandra. The ranks score can vary between the different scales of 1 to 5 or 1 to 10. Normalized ranks are calculated in this paper. The Cassandra in the designed approach will decrease the execution time represented in milliseconds (ms). The dataset contains hotel reviews and users feedbacks in the form of votes, ranks, and YouTube video views. The data are collected in the steps shown in Figure 4.

In our proposed application, the processing of the designed web recommender application is explained in the following steps:

(i) Start the process

(ii) The active user queries the system by inputting as per the search criteria and guest types such as solo, couple, and business

(iii) Then, the system checks for the previous users' data (ratings, ranks, and reviews) matching the query from the web in the system database

(iv) The system filters the query data by matching query in the external web sources available

(v) If match with the query, then collect metadata

(vi) Save the metadata in the NoSQL database

(vii) If it does not match, then discard it

(viii) Repeat until all matched metadata are found

(ix) End

4.2. System Overview. The proposed recommender system application is made up of three main components. The first is the external resources, then the front end, and the other is backend as shown in Figure 5. The dataset contains hotel reviews and ranks which are taken from the external hotel websites of Trip Advisor and Expedia. Reviews are divided into training and testing data sets to verify the improved performance of the proposed methodology using Hadoop plat form and Cassandra database. The complete data are stored into the proposed system database using web crawler written in java in the developed methodology.

In order to get best recommendations related to the users' choice and desire, our hotel recommendation system is developed with certain methods and techniques and also uses some open source tools such as Hadoop platform and Cassandra. The application is accessible online from any platform, and it uses development environment based on reliable open source tools. The computing environment is also discussed in Table 8. In the proposed application, the user can query providing search criteria to get their desire recommendations. The system provide recommendations by using the review polarity scores and ratings calculated using the reviews users data stored in Cassandra which matched the active user query.

4.3. Computing Resources. The resources required to test our proposed system also include some reliable open source tools, for example, Cassandra, Hadoop, amd PHP. The system specifications are given in Table 8.

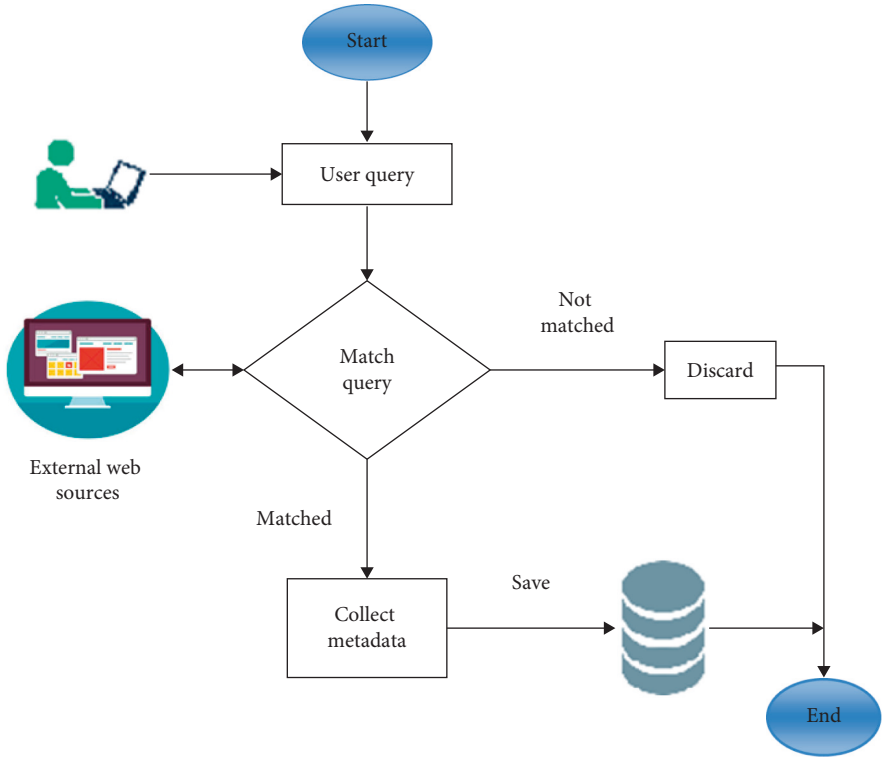

Figure 4: Data collection.

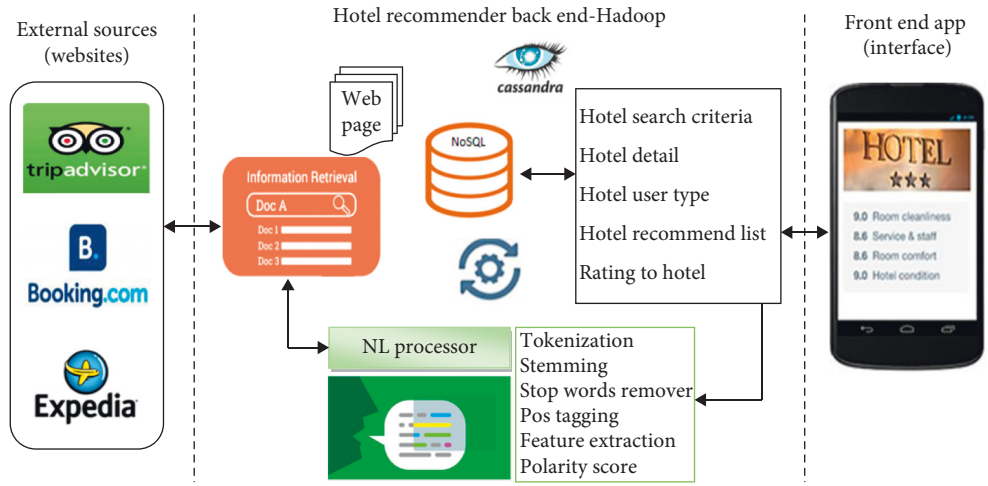

Figure 5: Proposed hotel recommender architecture.

Table 8: System specifications.

The system specifications are 4- Core. i7-3770, 3.40 GHz, 16 GB RAM, 500 GB disk
It is implemented with CentOS 7, Hadoop 2, Apache Cassandra 3.11, java 1.8, and PHP 7.1, PHP-Cassandra extension

4.4. Web Service and Methods. The designed hotel recommendation application will be accessible through a web page. HTTP connections are used to perform a number of web services using appropriate determinant URIs. The linked data are gathered by the application when a GET request is submitted by a URI, with the specified method providing user authentication and the required query parameters (QPs). The corresponding response message HTTP is formatted in JSON keeping the data uniformity and returned to App. When a user requires any data from the web page, certain specified URIs and HTTP connections are used to connect a user. The user made a request, and it is submitted through the appropriate HTTP method by using the necessary parameters required by that particular method, and

then after processing, the requested data are displayed on the front end page of the web application. All the methods used and their parameters in URIs are shown in Table 9.

The hotel data are converted into JSON format to provide information about the hotel to produce recommendation out of large number of hotels available. During the data transmission, the web services and methods are protected by SSL over the recommender web application.

4.5. Experiment and Results. We have used two reliable data repositories (Trip Advisor and Expedia) containing significant number of ranks, ratings, and reviews to represent heterogeneity of data, i.e., textual reviews and numerical ratings and ranks. These data scores contain data for 8000 of the most popular hotels and collection of hotel reviews and ratings which is useful in our experiments. After data preprocessing, TF-IDF generation, and polarity detection using SentiWordNet, we have computed the polarity scores for textual reviews obtained from each selected website. We have illustrated the data obtained from the selected hotels and the corresponding data source and its processing in the proposed methodology in Table 10.

In Table 8, the selected external data source used in our work such as Trip Advisor is represented by "D1" and Expedia is represented as "D2." Similarly, the corresponding selected hotels are represented as Mandarin Oriental, New York "H1," Amsterdam Court Hotel "H2," Hotel Metro "H3," Millennium Hilton "H4," and Belnord Hotel "H5." We have calculated polarity scores of the textual reviews for each hotel from the selected websites. Normalized rank score (scale 1–5) and the voting score are also calculated. Voting score represents the number of votes given by the customer to each hotel. Hotel names and data sources with their corresponding IDs are shown in the table.

The normalized rank scores of a selected hotel from two selected data sources such as Trip Advisor "D1" and Expedia "D2" are plotted in Figure 6, and it has been noted that the Expedia has high ranks comparatively in comparison with Trip Advisor.

Polarity scores of textual reviews of hotels taken from different selected data sources are calculated and aggregated in Table 11. Weighted average polarity is achieved by adding normalized ranks and votes in the average polarity (calculated by taking average of aggregated polarities).

The power of social media websites such as twitter, YouTube, and Facebook is also creating a shift in the way travelers seek out suggestions and tips before making any booking decision for certain hotel. Videos contain hotels pictures as well as present hotel services which may also affect the behavior of customers before selection of hotel and also have impact on the hotel rating. That is why we have used YouTube video views in our work to present the heterogeneous approach. We added the number of views in weighted average polarity to calculate the aggregated weighted average polarity for quality recommendations. The final scores along with hotel classes are shown in Table 12.

These heterogeneous data sources such as ranks, votes, textual reviews, and views are computed using the proposed approach, and final rank scores are obtained as shown in Figure 7. The hotels are classified based on the final ranking score. The same is explained in Table 12, and hotel class is identified based on the fuzzy set used in our work.

The final rank score "F" of H1 hotel "SpringHill Suites Denver Downtown" is greater than 8 that is why it is placed in class "R." The H3 hotel "Mandarin Oriental New York" lies in "BR"class because "F" score is greater than 6 and is less than 8. Whereas, the H2 hotel "Amsterdam Court Hotel," H4 hotel "Millennium Hilton," and H5 hotel "Belnord Hotel" scores are greater than 2 and less than 4 so it lies in class "LR."

Figure 8 represents the class of each hotel taken from selected data sources against the final rank score computed based on the guest types such as solo, family, and couple.

Figure 7 represents the each recommended criteria score against the selected data source and the proposed system using heterogeneous data.

This designed web application provides the customers the opportunity to obtain their desire hotel out of a large number of hotels available. The recommender system searches on the basis of system defined criteria depending upon the guest type. The list of recommended hotels generated by the proposed system is displayed in Figure 9.

5. System Performance Evaluation

5.1. Evaluation Metrics. Some evaluation metrics are used to evaluate the accuracy of the proposed system. These evaluation metrics convey that the results obtained by the proposed system are accepted by the targeted users. Mostly performance evaluation uncovers what needs to be improved before the product goes to market. Without performance analysis, software application is likely to suffer from issues such as running slow while several users use it simultaneously or if the system does not respond quickly. So we have performed performance analysis to demonstrate that the proposed system produces effective recommendations; there are number of metrics but we have used three metrics which are used in this study.

Precision is the first measure used to evaluate our system represented by the following equation:

$$\text{precision} = \frac{A}{C + A} \times 100\%. \tag{10}$$

According to the user feedbacks in the system, we assume that A represents the total number of recommended hotels liked by the user and C represents the number of hotels which are not liked by the user. "$C + A$" will be the total number of hotels recommended by the system to the same user. The precision can be calculated by taking the ratio of the number of the recommended hotels which are liked by the user over the total number of hotel recommended by the system. The Recall rate is defined as follows as per the same assumptions made above:

Table 9: Methods used in the web service.

Method	Description	QPs	HTTP method
Search	Searches by the given criteria to get a list of available hotels data	*Searchname*	GET
Topratings	Displays the specified hotels along with their ratings	*Ratings. Hotel id*	GET
Hoteldetail	Provides hotel details with name ID and region	*hotelId*	GET
Recommendhotel	Provides a list of recommended hotels	*Id/name*	GET
Getratings	Requires a list of hotels previously visited by users and replies by the ratings data	*numRating*	GET

Table 10: Polarity scores, rank scores, and voting scores.

Selected hotel	Data source					
	Trip advisor (D1)			Expedia (D2)		
	Polarity score	Normalized rank score	Voting score	Polarity score	Normalized rank score	Voting score
SpringHill Suites, Denver Downtown (H1)	31	3.9	209301	23	5	268231
Amsterdam Court Hotel (H2)	−5	3.4	38821	7	4	63420
Mandarin Oriental, New York (H3)	17	3.2	111620	24	5	127023
Millennium Hilton (H4)	−4	2.8	17023	−3	3.5	35622
Belnord Hotel (H5)	16	3.5	29441	22	5	41323

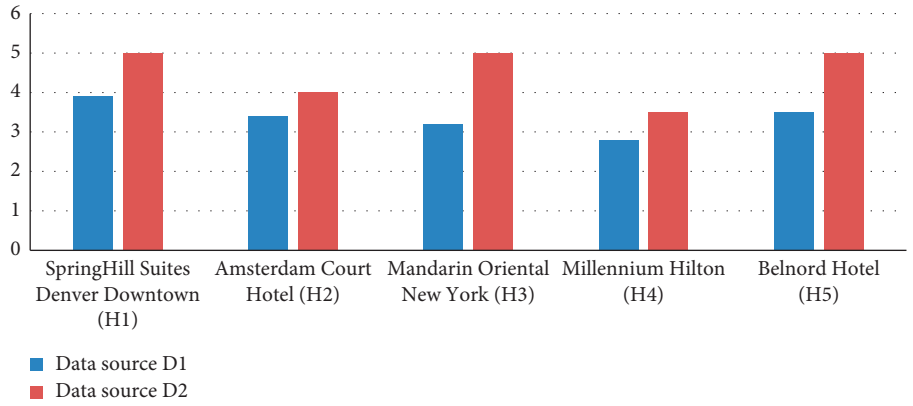

Figure 6: Difference in ranks scores.

Table 11: Weighted average polarity computation.

Hotel	Reviews		Aggregated polarity		Average polarity		Weighted average polarity	
	D1	D2	D1	D2	D1	D2	D1	D2
H1	1309	1519	31	23	0.023	0.015	209304.92	268236.01
H2	396	456	−5	7	0.012	0.015	38824.41	63424.01
H3	1189	998	17	24	0.014	0.024	111623.21	127028.02
H4	537	337	−4	−3	0.007	0.008	17025.80	35625.50
H5	971	1117	16	22	0.016	0.019	29460.50	41328.01

Table 12: Computation of final score.

Hotel ID	YouTube views	Aggregated weighted average polarity	Final score	Hotel class
H1	331025	569795.465	8.93234	R
H2	89023	140147.21	3.92215	LR
H3	284230	403555.615	7.63217	BR
H4	56056	82381.65	2.08245	LR
H5	78124	113518.255	3.12871	LR

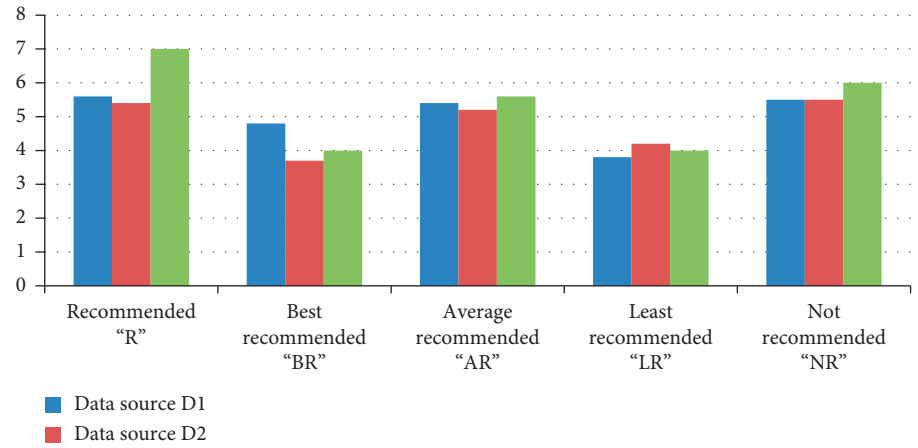

Figure 7: Comparison of hotel scores.

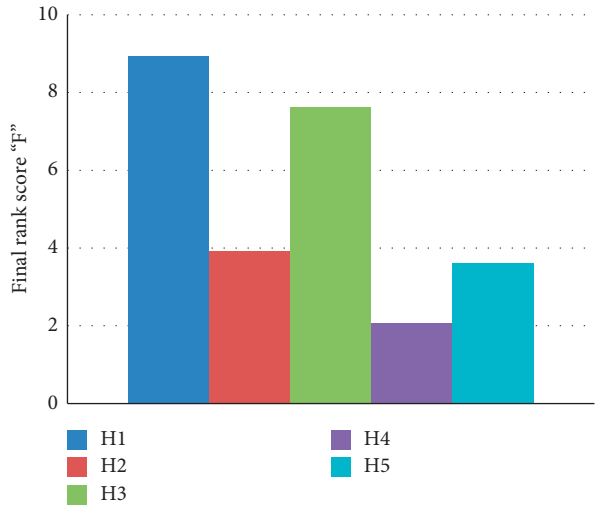

Figure 8: Proposed hotel recommender final ranking.

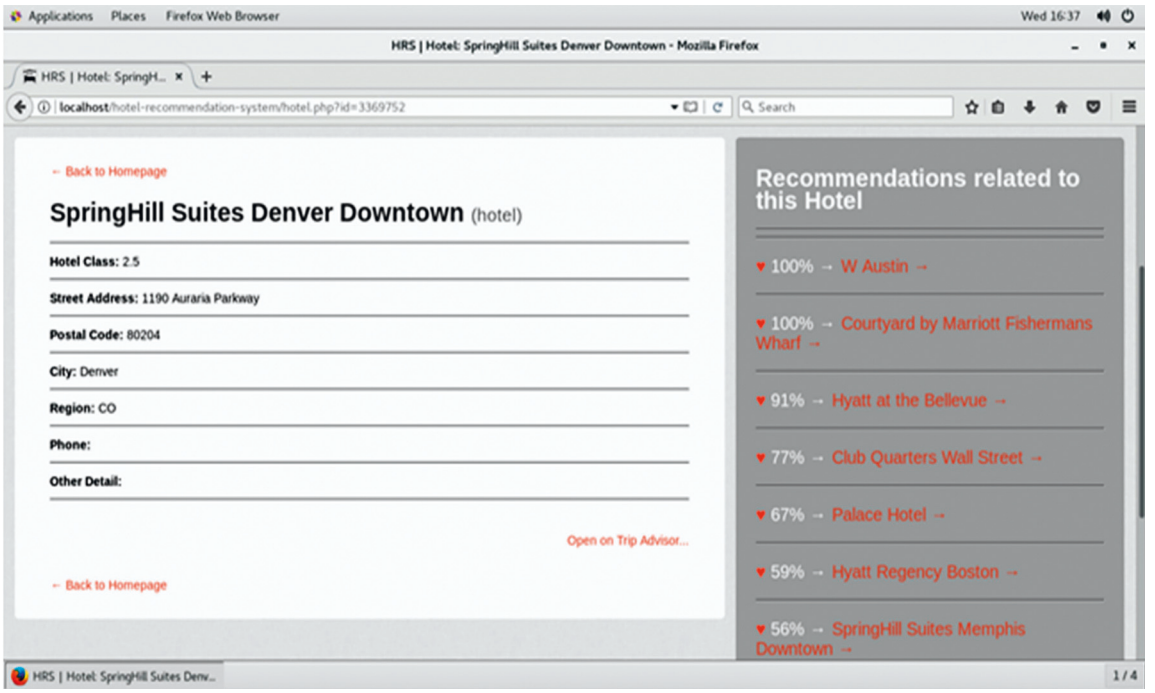

Figure 9: List of recommendations.

$$\text{Recall} = \frac{A}{B + A} \times 100\%. \quad (11)$$

where "B" represents the number of hotels targeted but are not recommended to the user and $B + A$ includes all the hotels which the user may possibly like. Recall and Precision ratios are opposite to each other. Perfect Recall rate of "1" means comparatively low Precision rate. High Precision but lower Recall gives extremely accurate recommendation. The third accuracy measure used is F-measure represented by the following equation:

$$\text{F} - \text{measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (12)$$

The F-measure is the Harmonic Mean of Precision and Recall and is used to evaluate the accuracy and efficiency of the results generated by the proposed recommender system.

We have used a subset of actual dataset which contains 100 hotels with 500 users from each of the selected websites as an exemplary dataset to test and train our proposed system. The metadata are divided into testing data (40%) and training data (60%). Depending on these data parts, we initially performed the calculations on the underlying 40% of the data of the exemplary metadata in the form of chunks of 20% of data for testing of the developed system in Table 13. We incrementally included the metadata of the remaining of the 60% dataset into the system for training the system. The accuracy metrics Precision, Recall, and F-measure with each increment are measured for exemplary data and recorded.

Figure 10 shows the F-measure values of recommendations under the proposed mechanisms with exemplary small data. The proposed recommender system supports computations to evaluate how it would influence the recommendation accuracy. The system with NoSQL dataset and proposed machine learning approach using sentiment analysis provides accurate recommendations, and its F-measure ratio value is 0.950 as the initial exemplary dataset used is very small containing 100 hotels with 500 users so such a huge improvement in terms of Precision, Recall, and F-measure is obtained. As the percentage of data increases, the values of performance measures decrease.

After testing the system with initial exemplary metadata, now the actual complete number of hotels in the used dataset obtained from Trip Advisor and Expedia contains 8000 hotels and 10 million users' data. As the actual complete dataset is large, the accuracy metrics values are comparatively decreased but still show promising results as recorded in Table 14. It shows that the proposed recommender using sentiment analysis approach provides accurate and quality recommendations to users. Graphical representation is also shown in Figure 11.

5.2. System Processing Time. Performance analysis is performed to ensure that software application will perform well under their expected workload. Most performance problems revolve around speed, response time, load time and poor scalability. Response time is often one of the most important attributes of an application. A slow running application will lose potential users. Performance testing is done to make sure an app runs fast enough to keep a user's attention and interest.

We have performed performance testing by varying patterns of workload (number of users). We have performed different experiments with different number of users. For the better understanding, we have taken data of 12 users from different professions and different age level to use and perform experiments over the proposed system. They have also used other recommenders, trivago.com, hotels.com and yatra.com to get a recommended list of hotels of their choice. Their satisfaction level is evaluated by taking their opinions about searches over each of these recommenders. Processing time over the developed system is also recorded during these searches. Participants have performed provide feedback which is recorded in Table 15. We have classified the satisfactory level into the 3 classis i.e. less satisfactory (L), satisfactory (S), and highly satisfactory (H). Feedback is taken against three capabilities i.e. time efficient (TE), relevance of recommended hotels with the user choice (RoR), cost effectiveness (CE).

Results presented in Table 15 shows satisfactory level of the participants in which most of users has selected best category to the proposed recommender system.

When the user entered the query in the proposed recommender to obtain best hotel recommendation providing a guest type according to their choice, the system performs computation using proposed machine learning sentiment analysis to collect the required recommendations. The proposed system has shown promising results in terms of improved response time comparing traditional recommenders. The system also calculates the time stamps as load time, search time, and execution time represented as milliseconds (ms). The outcomes generated in terms of response time are recorded in Table 16 and shown in Figure 12. The sum of load time and search time together is called as execution time. The proposed system outperforms and takes very less time giving the list of recommendations.

The total average query processing time taken by the proposed system with maximum workload is 2.6592 ms. Our approach along with Cassandra NoSQL database is efficient and helps to reduce the total processing time. Results indicate that the system is performing efficiently which also changes people opinions about using recommenders.

The system contributes efficiently to help users while searching hotels they like. The system is designed to achieve true recommendations and to impact customer behavior positively in terms of accuracy of recommendations with the need of the user. If true recommendations are provided, it will definitely increase the customer's satisfaction level helping them making their business decisions.

6. Comparative Analysis

The comparison in terms of time and evaluation metrics is performed in this section. It compares the proposed approach with the previous traditional approaches. A statistical analysis of the performance evaluation metrics such as the F-measure, Precision, and Recall is performed. To assess the

Table 13: Evaluation metrics with exemplary dataset.

Incremental data update using Euclidean similarity (%)	F-measure	Recall	Precision
20	0.966	0.954	0.978
40	0.956	0.941	0.972
60	0.951	0.936	0.968
80	0.938	0.921	0.956
100	0.924	0.898	0.951
Avg. ratio	0.950	0.930	0.965

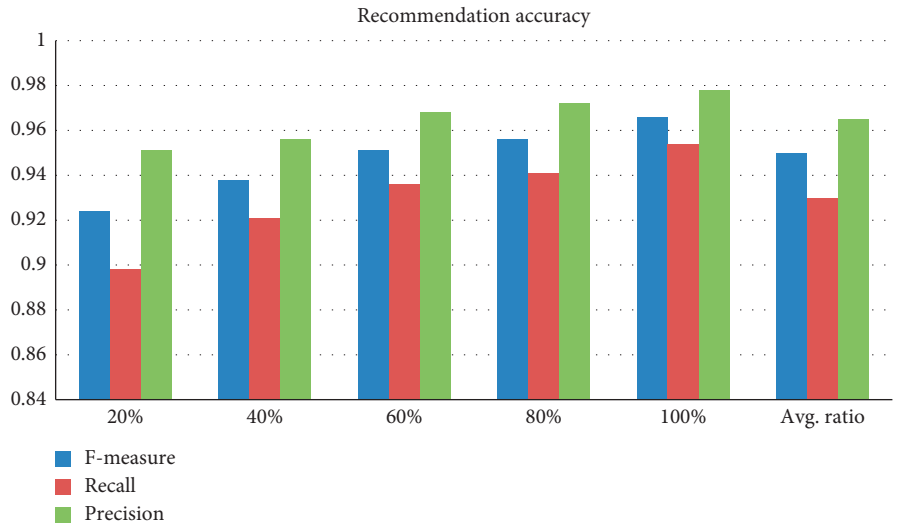

Figure 10: Performance metrics with exemplary dataset.

Table 14: Evaluation metrics with complete dataset.

Incremental data update using Euclidean similarity (%)	F-measure	Recall	Precision
20	0.825	0.791	0.877
40	0.789	0.728	0.861
60	0.771	0.726	0.821
80	0.729	0.679	0.788
100	0.688	0.631	0.756
Avg. ratio	0.76	0.711	0.821

recommendation accuracy of the proposed method, we have performed the comparison of the performance analysis metrics such as Recall, Precision, and F-measure of our hotel recommended system with a subset of actual dataset, i.e., with small exemplary dataset (100 hotels and 500 users), with complete dataset (8000 hotels 10 million users), and with the Precision, Recall, and F-measure of tradition-related recommenders and are provided in Table 17.

We have also compared the existing studies with our approach and found out promising improvement in terms of execution time of the proposed approach. The comparative analysis of the performance of the proposed hotel recommender approach with the existing related approaches found in the literature is shown in Table 18 which exhibits outcomes in terms of time improvement.

The above comparison of evaluation metric and recommendation time exhibits that the proposed approach has shown promising results. The recommendation time is reduced using the proposed approach when it is compared with the conventional approaches.

7. Conclusion and Future Work

In this paper, a novel CF recommendation approach is proposed which has the ability to handle heterogeneous data such as textual reviews, ranks, votes, and video views in a big data Hadoop environment with Cassandra database to guarantee the improved response time to generate recommendations. In the proposed system, opinion-based sentiment analysis is used to extract a hotel feature matrix and stored in a database. Our approach combines lexical analysis, syntax analysis, and semantic analysis to understand the sentiment towards hotel features. The NLTK library is used to identify the polarity of the textual reviews. The system makes use of fuzzy rules to determine the hotel class depending upon the guest type. Euclidean distance is used to

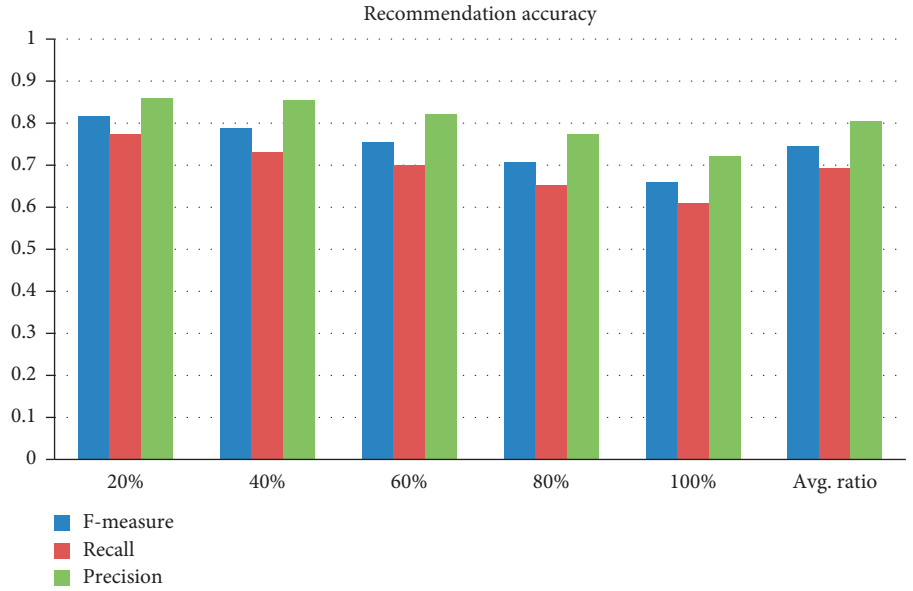

Figure 11: Performance metrics with complete dataset.

Table 15: An illustration of Participants Opinions.

No.	Profession	Age	kayak			Hotels.com			Booking.com			Proposed recommender		
			TE	RoR	CE	TE	RoR	CE	TE	RoR	CE	TE	RoR	CE
User1	Student	19	L	L	L	L	L	L	S	L	L	S	H	H
User2	Student	18	L	L	H	L	S	S	L	H	S	S	H	L
User3	Business	35	L	L	L	S	L	L	L	L	L	H	S	H
User4	Doctor	31	L	L	H	L	L	S	S	L	S	H	S	H
User5	Teacher	21	S	L	S	L	L	L	L	S	L	H	H	H
User6	Student	48	L	L	L	L	L	H	S	L	S	L	H	S
User7	Student	22	H	S	S	L	S	L	L	S	S	H	H	H
User8	Employee	41	L	S	L	L	H	L	L	L	H	S	H	H
User9	Student	29	L	S	L	S	S	L	S	L	S	H	S	S
User10	Doctor	39	S	L	S	L	L	L	L	S	L	H	H	L
User11	Teacher	46	L	L	S	L	L	S	S	S	S	H	H	S
User12	Student	28	S	L	S	L	L	L	L	S	L	S	H	H

Table 16: System response time.

No.	Loading time	Searching time	Execution time (ms)
1	3.0994	0.0461	3.5606
2	1.6212	0.0459	1.6712
3	1.0013	0.0488	1.0501
4	2.8610	0.0738	2.9348
5	3.0994	0.0455	3.1441
6	3.0994	0.0455	3.5544
7	2.8610	0.0727	2.9337
8	2.8610	0.0469	2.9079
9	3.0994	0.0024	3.1018
10	2.8610	0.0500	2.9069
11	1.9073	0.4583	1.9531
12	3.0994	0.0447	3.5544

calculate the similarities between the items and provide accurate recommendations based on the type of guest (solo, family, couple, etc.). The system takes 2.65 milliseconds to generate high-quality recommendations by reducing the system execution time. The resultant F-measure has resulted in 0.950 approximately 95% when we have run the system with exemplary data for training the system but when system is trained with the complete dataset obtained for websites, the

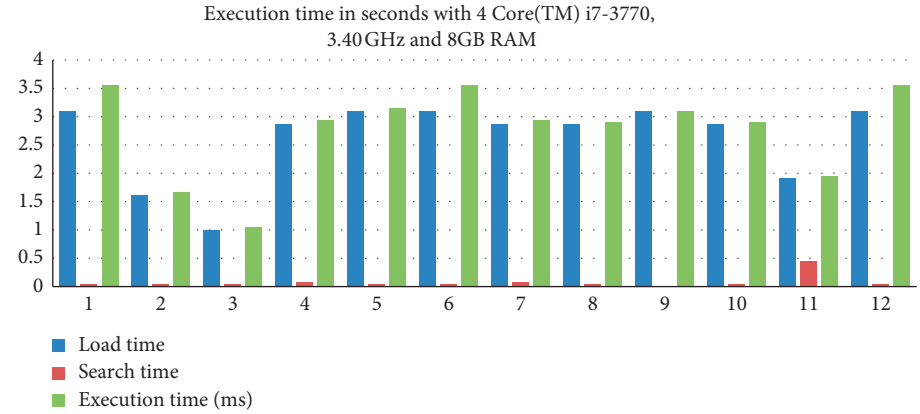

Figure 12: System response time.

Table 17: Comparison of accuracy metrics with related studies.

No.	Reference		Precision	Recall	F-measure
1	Liu et al. [2]		0.56	0.60	0.59
2	Hsieh et al. [6]		0.02	0.53	0.01
3	Zhang et al. [10]		0.25	0.34	0.35
4	Chang et al. [16]		0.43	0.29	—
5	Lin et al. [29]		0.62	0.51	—
6	Verma and Virk [19]		0.69	0.67	0.68
7	Proposed approach	With exemplary subset of dataset	0.96	0.93	0.95
		With complete dataset	0.80	0.69	0.74

Table 18: Comparison of execution time with related studies.

No.	Reference	Recommendation time
1	Bouras and Tsogkas [34]	12 sec
2	Jazayeriy et al. [37]	28 sec
3	Liu et al. [2]	27 sec
4	Proposed recommender	2.6 millisec

F- measure is somehow decreased. As the actual complete dataset is large, the accuracy metrics values are comparatively decreased to 0.745 approximately 74% but still showing promising results as recorded in Table 14. It shows that the proposed recommender using sentiment analysis approach provides accurate and quality recommendations to users.

In future, the recommender system is needed to be designed in a way that will utilize dynamic auto updated data containing the visual views, votes, and reviews online from the external websites to provide recommendations according to dynamic data found at the same time of active user query. To amplify the versatility of recommender services, this will be implemented by incorporating web cookies and using customer's navigational activities and getting feedbacks over the new recommended items.

Data Availability

The authors will provide the data used for the experiments, if requested.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] M. M. Mariani, D. Buhalis, C. Longhi, and O. Vitouladiti, "Managing change in tourism destinations: key issues and current trends," *Journal of Destination Marketing and Management*, vol. 2, no. 4, pp. 269–272, 2014.

[2] H. Liu, J. He, T. Wang, W. Song, and X. Du, "Combining user preferences and user opinions for accurate recommendation," *Electronic Commerce Research and Applications*, vol. 12, no. 1, pp. 14–23, 2013.

[3] M. Ibrahim and I. Bajwa, "Design and application of a multi-variant expert system using Apache Hadoop framework," *Sustainability*, vol. 10, no. 11, p. 4280, 2018.

[4] J. J. Zhang and Z. Mao, "Image of all hotel scales on travel blogs: its impact on customer loyalty," *Journal of Hospitality Marketing and Management*, vol. 21, no. 2, pp. 113–131, 2012.

[5] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, 2009.

[6] M.-Y. Hsieh, W.-K. Chou, and K.-C. Li, "Building a mobile movie recommendation service by user rating and APP usage with linked data on Hadoop," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 3383–3401, 2017.

[7] R. Burke, "Hybrid recommender systems: survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[8] P. Lops, M. De Gemmis, and G. Semeraro, "Content-based recommender systems: state of the art and trends," in *Recommender Systems Handbook*, pp. 73–105, Springer, Boston, MA, USA, 2011.

[9] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative filtering recommender systems," *Foundations and Trends in Human-Computer Interaction*, vol. 4, no. 2, pp. 81–173, 2011.

[10] J. Zhang, Q. Peng, S. Sun, and C. Liu, "Collaborative filtering recommendation algorithm based on user preference derived from item domain features," *Physica A: Statistical Mechanics and Its Applications*, vol. 396, pp. 66–76, 2014.

[11] G. Huming and L. Weili, "A hotel recommendation system based on collaborative filtering and rankboost algorithm," in *Proceedings of the 2010 Second International Conference on Multimedia and Information Technology (MMIT)*, vol. 1, pp. 317–320, IEEE, Kaifeng, China, April 2010.

[12] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International Conference on World Wide Web*, pp. 285–295, ACM, Hong Kong, May 2001.

[13] Z. Tan and L. He, "An efficient similarity measure for user-based collaborative filtering recommender systems inspired by the physical resonance principle," *IEEE Access*, vol. 5, pp. 27211–27228, 2017.

[14] K. Zhang, K. Wang, X. Wang, C. Jin, and A. Zhou, "Hotel recommendation based on user preference analysis," in *Proceedings of the 2015 31st IEEE International Conference on Data Engineering Workshops (ICDEW)*, pp. 134–138, IEEE, Seoul, South Korea, April 2015.

[15] L. Chen and F. Wang, "Preference-based clustering reviews for augmenting e-commerce recommendation," *Knowledge-Based Systems*, vol. 50, pp. 44–59, 2013.

[16] Z. Chang, M. S. Arefin, and Y. Morimoto, "Hotel recommendation based on surrounding environments," in *Proceedings of the 2013 IIAI International Conference on Advanced Applied Informatics (IIAIAAI)*, pp. 330–336, IEEE, Matsue, Japan, August-September 2013.

[17] Y. Sharma, J. Bhatt, and R. Magon, "A multi-criteria review-based hotel recommendation system," in *Proceedings of the 2015 IEEE International Conference on, Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM)*, pp. 687–691, IEEE, Liverpool, UK, October 2015.

[18] T. Chen and Y. H. Chuang, "Fuzzy and nonlinear programming approach for optimizing the performance of ubiquitous hotel recommendation," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 2, pp. 275–284, 2018.

[19] A. Verma and H. Virk, "A hybrid genre-based recommender system for movies using genetic algorithm and kNN approach," *International Journal of Innovations in Engineering and Technology*, vol. 5, no. 4, pp. 48–55, 2015.

[20] K. Kabassi, "Personalizing recommendations for tourists," *Telematics and Informatics*, vol. 27, no. 1, pp. 51–66, 2010.

[21] U. Fasahte, D. Gambhir, M. Merulingkar, and A. M. P. A. Pokhare, "Hotel recommendation system," *Imperial Journal of Interdisciplinary Research*, vol. 3, no. 11, 2017.

[22] G. Crespo, A. L. Cuadrado, J. L. C. Palacios, R. G. Carrasco, and I. R. Mezcua, "Sem-Fit: a semantic based expert system to provide recommendations in the tourism domain," *Expert Systems with Applications*, vol. 10, no. 38, pp. 13310–13319, 2011.

[23] Y. H. Hu, P. J. Lee, K. Chen, J. M. Tarn, and D.-V. Dang, "Hotel recommendation system based on review and context information: a collaborative filtering appro," in *Proceedings of the Pacific Asia Conference on Information Systems PACIS*, p. 221, Chiayi City, Taiwan, June-July 2016.

[24] S. Y. Hwang, C. Y. Lai, S. Chang, and J. J. Jiang, "The identification of noteworthy hotel reviews for hotel management," *Pacific Asia Journal of the Association for Information Systems*, vol. 6, no. 4, 2015.

[25] R. Sandeep, S. Sood, and V. Verma, "Twitter sentiment analysis of real-time customer experience feedback for predicting growth of Indian telecom companies," in *Proceedings of the 2018 4th International Conference on Computing Sciences (ICCS)*, pp. 166–174, IEEE, Phagwara, India, August 2018.

[26] S. Meng, W. Dou, X. Zhang, and J. Chen, "KASR: a Keyword-Aware Service Recommendation method on MapReduce for big data applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3221–3231, 2014.

[27] A. Ghose, P. G. Ipeirotis, and B. Li, "Designing ranking systems for hotels on travel search engines by mining user-generated and crowdsourced content," *Marketing Science*, vol. 31, no. 3, pp. 493–520, 2012.

[28] N. M. Almeida, J. A. Silva, J. Mendes, and P. Oom do Valle, "The effects of marketing communication on the tourist's hotel reservation process," *Anatolia*, vol. 23, no. 2, pp. 234–250, 2012.

[29] K. P. Lin, C. Y. Lai, P. C. Chen, and S. Y. Hwang, "Personalized hotel recommendation using text mining and mobile browsing tracking," in *Proceedings of the 2015 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 191–196, IEEE, Hong Kong, China, October 2015.

[30] N. Rianthong, A. Dumrongsiri, and Y. Kohda, "Improving the multidimensional sequencing of hotel rooms on an online travel agency web site," *Electronic Commerce Research and Applications*, vol. 17, pp. 74–86, 2016.

[31] G. S. Lal and A. S. Baghel, "Efficient feature extraction in sentiment classification for contrastive sentences," *International Journal of Modern Education and Computer Science*, vol. 10, no. 5, p. 54, 2018.

[32] D. Jannach, F. Gedikli, Z. Karakaya, and O. Juwig, "Recommending hotels based on multi-dimensional customer ratings," in *Information and Communication Technologies in Tourism*, Springer, Vienna, Austria, 2012.

[33] M. Ibrahim, I. S. Bajwa, R. Ul-Amin, and B. Kasi, "A neural network-inspired approach for improved and true movie recommendations," *Computational Intelligence and Neuroscience*, vol. 2019, no. 7, Article ID 4589060, 19 pages, 2019.

[34] C. Bouras and V. Tsogkas, "Improving news articles recommendations via user clustering," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 1, pp. 223–237, 2017.

[35] D. Valcarce, J. Parapar, and A. Barreiro, "A distributed recommendation platform for big data," *Journal of Universal Computer Science*, vol. 21, no. 13, pp. 1810–1829, 2015.

[36] S. Ishtiaq, N. Majeed, M. Maqsood, and A. Javed, "Improved scalable recommender system," *The Nucleus*, vol. 53, no. 3, pp. 200–207, 2016.

[37] H. Jazayeriy, S. Mohammadi, and S. Shamshirband, "A fast recommender system for cold user using categorized items," *Mathematical and Computational Applications*, vol. 23, no. 1, p. 1, 2018.

[38] K. Sudha, M. Lavanya, and A. Kanimozhi, "Item based collaborative filtering approach for big data application," *International Journal of Scientific Research Engineering & Technology (IJSRET)*, vol. 3, no. 8, pp. 1222–1224, 2014.

[39] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.

[40] M. Manu and B. Ramesh, "Single-criteria collaborative filter implementation using Apache Mahout in big data,"

International Journal of Computer Sciences and Engineering, vol. 5, no. 1, pp. 7–13, 2017.

[41] S. Morozov and X. Zhong, "The evaluation of similarity metrics in collaborative filtering recommenders," in *Proceedings of the 2013 Hawaii University International Conferences Education & Technology Math & Engineering Technology*, Honolulu, HI, USA, June 2013.

[42] Q. Shambour, M. A. Hourani, and S. Fraihat, "An item-based multi-criteria collaborative filtering algorithm for personalized recommender systems," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 8, pp. 275–279, 2016.

[43] F. Cacheda, V. Carneiro, D. Fernández, and V. Formoso, "Comparison of collaborative filtering algorithms: limitations of current techniques and proposals for scalable, high-performance recommender systems," *ACM Transactions on the Web (TWEB)*, vol. 5, no. 1, p. 2, 2011.

[44] M. Nilashi, O. bin Ibrahim, N. Ithnin, N. H. Sarmin, and N. H. Sarmin, "A multi-criteria collaborative filtering recommender system for the tourism domain using expectation maximization (EM) and PCA-ANFIS," *Electronic Commerce Research and Applications*, vol. 14, no. 6, pp. 542–562, 2015.

[45] P. Phorasim and L. Yu, "Movies recommendation system using collaborative filtering and *k*-means," *International Journal of Advanced Computer Research*, vol. 7, no. 29, pp. 52–59, 2017.

[46] S. Kögel, "Recommender system for model driven software development," in *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, pp. 1026–1029, Paderborn, Germany, September 2017.

[47] M.-P. T. Do, D. V. Nguyen, and L. Nguyen, "A model-based approach for collaborative filtering," in *Proceedings of the 6th International Conference on Information Technology for Education*, Ho Chi Minh City, Vietnam, August 2010.

[48] W. Yibo, M. Wang, and W. Xu, "A sentiment-enhanced hybrid recommender system for movie recommendation: a big data analytics framework," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 8263704, 9 pages, 2018.