

Semantic-Based Service Recommendation Method on MapReduce Using User-Generated Feedback

Ruchita Tatiya and Archana Vaidya

Abstract Service recommender systems provide same recommendations to different users based on ratings and rankings only, without considering the preference of an individual user. These ratings are based on the single criteria of a service ignoring its multiple aspects. Big data also affects these recommender systems with issues like scalability and inefficiency. Proposed system enhances existing recommendations systems and generates recommendations based on the categorical preferences of the present user by matching them with the feedback/comments of the past users. System semantically analyzes the users feedback and distinguishes it into positive and negative preferences to eliminate the unnecessary reviews of the users which boosts the system accuracy. Approximate and exact similarity between the preferences of present and past users is computed and thus the recommendations are generated using SBSR algorithm. To improve the performance, i.e., scalability and efficiency in big data environment, SBSR is ported on distributed computing platform, Hadoop.

Keywords Service recommender systems • Big data • Semantic analysis • Jaccard co-efficient • Cosine similarity • Hadoop • MapReduce

1 Introduction

The overabundance of data on the web drives away the focus of users, landing them to surf for the data that they were not searching for initially. Information filtering systems are used to overcome these problems and to eliminate the unnecessary information before presenting it to the user. The subclass of these systems, called as

Ruchita Tatiya (✉) • Archana Vaidya
Gokhale Education Society's R. H. Sapat College of Engineering, Management Studies
and Research, P T A Kulkarni Vidyanaagar, Nashik, Maharashtra, India
e-mail: ruchitatatiya@gmail.com

Archana Vaidya
e-mail: archana.s.vaidya@gmail.com

recommendation systems assist by predicting the services or items that the user would like. Service recommender systems [1] provide appropriate recommendations and have become popular in variety of practical applications like recommending the users about hotels, books, movies, music, travel, etc. [2, 3]. The enlarged number of Internet users is contributing to immense amount of data everyday [4]. Such immense data, known as Big data, is not only difficult to capture and store but also managing, processing, and analysing such data with the available current technology within the tolerable speed and time is a difficult task.

1.1 Motivation

The service recommender systems present the same ratings and rankings of the services to the different users and also provide the same recommendations to them without considering the user's personal likings and taste [1]. Also many recommendation systems provides single-criteria ratings i.e. just the overall rating of any service is being considered which makes them less accurate [5]. Due to the ever increasing amount of data, the Big-data management poses a heavy impact on service recommender systems with issues like scalability and inefficiency. The proposed system, Semantic-Based Service Recommendation (SBSR), considers the issues and drawbacks of the existing system and contributes to generate recommendations more accurate according to the user likings and the categorical preferences by considering the multiple aspects of the service and also tries to improve the efficiency and performance of the system in the big data environment.

2 Literature Survey

There are various recommendation methods based on the information or knowledge source they use for making the apt recommendations. Reference [6] describes various methods to generate recommendations and also focuses on the algorithmic methods like memory-based and model-based algorithms. Author Hiralall [7] has compared various methodologies which can be used to generate recommendations. Different pros and cons are also stated which helps the user to select the apt approach according to his/her application. Adomavicius and Tuzhilin [5], has described the current generation of the recommendation methods and have stated that they are based on rating and rankings only, without considering the taste and choices on an individual and just provide the recommendation based on the single criteria ratings. To overcome the drawback of single-criteria ratings, authors Adomavicius and Kwon [8] incorporated and leveraged multi-criteria rating which improved the accuracy of the system as compared with single-rating recommendations. The problem faced by many recommendation algorithms is its scalability, i.e. when the volume of the dataset is very large, the computation cost would be very high.

The development of cloud computing software tools such as Apache Hadoop, MapReduce, and Mahout, made possible to design and implement scalable recommender systems in Big data environment. Reference [9] implemented the collaborative filtering algorithm on the cloud computing platform, Hadoop which solves the scalability problem for large scale data by dividing the dataset. Meng et al. [1] proposed a keyword aware service recommendation method, which utilizes the reviews of previous users to get both, user preferences and the quality of multiple criteria of candidate services, and computes similarity with the preferences of active user which in turn makes the recommendations more accurate. Moreover, they implemented their approach on MapReduce which showed favorable scalability and efficiency. Turney [10] presented an algorithm for classifying the reviews as recommended (thumbs up) or not recommended (thumbs down). The classification of a review is predicted by the average semantic orientation of the phrases in the review that contain adjectives or adverbs. The algorithm presented has three steps: extract phrases containing adjectives or adverbs, estimate the semantic orientation of each phrase, and classify the review based on the average semantic orientation of the phrases.

3 Proposed System

3.1 Architecture

The proposed recommender system is specially designed for the large scale data processing. While recommending particular service, the system mainly considers the user preferences and uses the previous users' comments/reviews which accounts to the immense data on the web. In the following, Fig. 1 shows the architecture of the proposed system SBSR, which is specifically the information filtering architecture that uses the distributed computing platform to reduce the processing time. Here, the system needs to filter the previous users' comments according to the active user preferences and semantically analyze them for removing the negative reviews, to present a personalized service recommendation list. Hence, the system manages to deal with large scale data with the help of Hadoop (a distributed computing platform using the MapReduce parallel processing paradigm for big data). The processing of data can be distributed across various nodes by splitting the input into multiple Map() and Reduce() phases and the response time of the system can be decreased. To test the working of the system, test dataset regarding hotels is used that helps us to analyze the throughput of the system. Later a more generalized form of this system can be developed using precision of experiments.

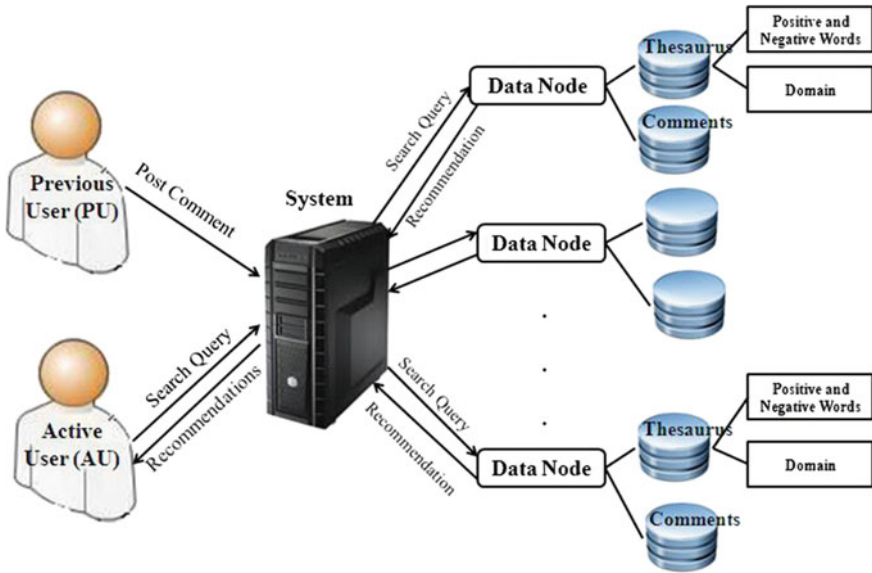


Fig. 1 System architecture

3.2 System Flow

Before mentioning the system flow, following are the descriptions of terminologies used.

1. Aspect keyword list (AKL): It is a keyword set related to the users' preferences searching for a particular aspect and also multiple criteria regarding that service are mentioned into it. For example, if the service is recommending the hotels then the aspect keyword list will contain all the main aspect keywords regarding the hotels like cleanliness, food, value, location, etc. [1].
2. Thesaurus: A thesaurus is the group of words collected according to their similarity of the meaning. Basically a domain thesaurus is associated with the aspect included in the AKL then all the related words of food like breakfast, tea, lunch, etc., are included in the thesaurus. Also the positive and negative words thesaurus consists of all the positive and negative words, phrases which are used in common natural language that are used for the semantic analysis.
3. Preference Weight Vector: The preference keyword sets of the active and previous users will be transformed into n -dimensional weight vectors, respectively, denoted as $W = [w_1; w_2; \dots; w_n]$ where ' n ' is the number of keywords and ' w_i ' is the weight of the keyword K_i in the AKL [1].

Figure 2 depicts the flow of the system diagrammatically and is explained below. The proposed system, SBSR, is divided into two processes as swing application and web application, respectively. The swing application mainly deals

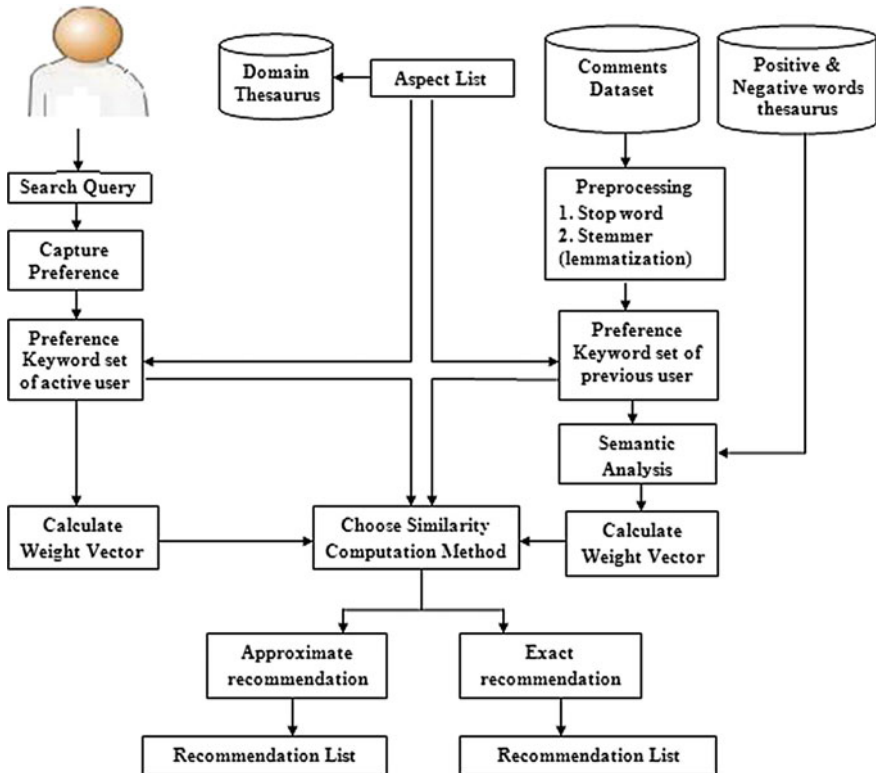


Fig. 2 System flow

with preprocessing the dataset which consists of the previous user comments. The administrator handles this swing application in which, the raw comments are pre-processed by applying the stop word removal and stemming algorithm and stored in the database. Then semantic analysis is performed on these processed comments and the positive and negative keywords in the comments are identified and categorized with respect to the aspects presented in the AKL. Later, the reviews for a particular hotel are amalgamated and the weight vector is calculated for every aspect and cached into the database. As the dataset is large and this processing is vast, the system is ported on hadoop which reduces the processing time. The web application is the recommendation generation system for the active users who receive the recommendations according to the personal likings and taste. In this, the active user can register and login to the system and choose approximate and exact recommendations of hotels according to his desires. The active user can search for hotels by using a natural language search query specifying his requirements. Following is the system flow which is divided into two parallel executable processes.

1. Process 1—Web Application (For active user) (Recommendation generation)
 - If the active user is registered on the system he can login and choose whether he wants an approximate or exact recommendation of hotels.
 - Active user gives his/her preferences about the aspect of the services in the form of natural language query, which reflects the quality criteria of the services he/she is concerned about.
 - Finally, after preprocessing the active user query, the preference keyword set of active user and its weight vector is calculated using the AKL and Domain thesaurus.
 - The recommendations are generated for the user.
2. Process 2—Swing Application (For administrator) (Pre-processing and Similarity computation)
 - Access the dataset having previous comments given by the past user and apply pre-processing like stop word removal, stemming.
 - Calculate preference keyword set of previous user using domain thesaurus and AKL.
 - Semantic analysis is performed on the preference keywords of previous user and the negative reviews are removed.
 - Calculate the weight vector for previous user preference keywords set.
 - Similarity computation—It identifies the comments of previous users whose taste matches to an active user by identifying the neighbors of the active user based on the similarity of their likings.
 - Calculate Approximate Similarity or Exact Similarity according to the user choice.

4 Implementation Details

4.1 Environment

The proposed system is designed for open source operating system Linux—Ubuntu 14.04. The implementation of this system is based on Java jdk-7 and Hadoop 2.3 platform using the MapReduce framework. MySQL 5.5.41 database is used for storing the datasets by configuring the LAMP server in Ubuntu. Also the configuration of phpMyAdmin in Ubuntu helps to perform various tasks such as creating, modifying or deleting databases with the use of a web browser. Eclipse (Luna) environment is being used for the system development. Initially for the testing purpose a single node Hadoop framework is being established. Also the Hadoop is configured with Eclipse to execute the hadoop programs in Eclipse environment.

4.2 Dataset

For the previous user comments or reviews regarding hotels, entity-ranking-dataset [11] is being used which is in the text format and contains: Full reviews of hotels in 10 different cities and there are about 80–700 hotels in each city which accounts to ~259,000 total number of reviews. The review format is: Date1 <tab> Full review1. For creating Domain Thesaurus related to aspect keyword list, the use of FeatureWords is done, downloaded from the Tripadvisor (<http://www.tripadvisor.com>) site and was in the text format having the following form of: #cat = <category or aspect>. For semantic analysis of comments there is a need of positive and negative words. It has been downloaded from [12]. These lists of words were downloaded in the .xls format.

4.2.1 Conversion of Raw Dataset

The datasets used for the system are in the raw format and immense in nature which therefore requires huge processing to convert it in the usable format. The text and .xls files of domain thesaurus and positive/negative words were converted into .csv format which were then imported into the MySQL database for further processing.

5 Results

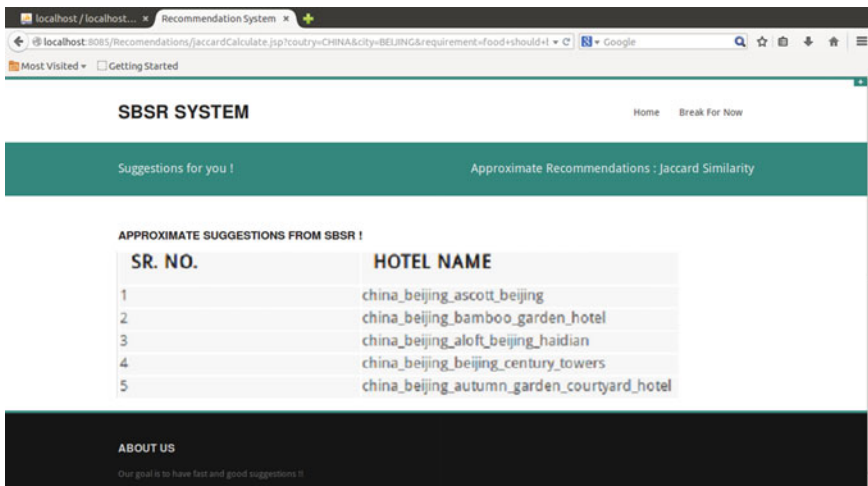
5.1 Pre-processing and Semantic Analysis of Previous User Comments

As the comments dataset is huge in size and in the raw format, the pre-processing of it is done on the Hadoop platform. The mapper() class of the hadoop is responsible for applying the stemming and stop word removal algorithms on the comments dataset & the intermediate pre-processed data (comment id, file name, date, original review, stemmed review, stop-word removed review, country and city) is stored in the database. Then the semantic analysis of the pre-processed comments is done with the help of positive and negative thesaurus. The domain of the preference word is found and the result is stored in the database (total domains occurring in the comment, negative words, domain related to negative words and domains related to positive words). After applying the semantic analysis phase the reducer() class of hadoop is responsible for the calculation of weight vector related to every aspect for a particular hotel, by considering all the reviews related to that one hotel. The positive & negative count for every aspect of that hotel is cached into the database so that it can be used while generating recommendation. If the positive count is high

then it means that, the hotel is good for that particular aspect and vice versa. The time required for pre-processing the data on hadoop is also noted. Likewise data is pre-processed & semantically analyzed for all the hotels in each city. After training raw data country, by country, completes the processing stage and the processed data can then be used by the web application while generating the recommendations.

5.2 Recommendations Generation

Using the web application, the active user can register into the system for approximate or exact recommendation generation. The active user queries the system regarding the hotels in natural language format, which generates the recommendation list of hotels for them according to their query and wish. A sample query fired for both approximate and exact recommendations was “Food should be tasty. Wifi should be there.” (Aspects mentioned in the query are food and business service) and the results generated are shown in the figures below. Figure 3 shows the approximate recommendations for the active user query using the Jaccard co-efficient. Even if any of the aspect mentioned in active user query is matched, the hotel is included in the approximate recommendation list. Figure 4 shows the exact recommendations using the cosine similarity function. If all the aspects mentioned in the active user query are matched then only that hotel is included in the recommendation list.



The screenshot shows a web browser window with the URL `localhost:8085/Recommendations/jaccardCalculate.jsp?country=CHINA&city=BEIJING&requirement=food+should+!+` . The page title is "SBSR SYSTEM" and it includes navigation links for "Home" and "Break For Now". A green banner indicates "Suggestions for you !" and "Approximate Recommendations : Jaccard Similarity". Below this, a section titled "APPROXIMATE SUGGESTIONS FROM SBSR !" contains a table with the following data:

SR. NO.	HOTEL NAME
1	china_beijing_ascott_beijing
2	china_beijing_bamboo_garden_hotel
3	china_beijing_aloft_beijing_haidian
4	china_beijing_beijing_century_towers
5	china_beijing_autumn_garden_courtyard_hotel

At the bottom, there is an "ABOUT US" section with the text: "Our goal is to have fast and good suggestions !!"

Fig. 3 Approximate recommendation

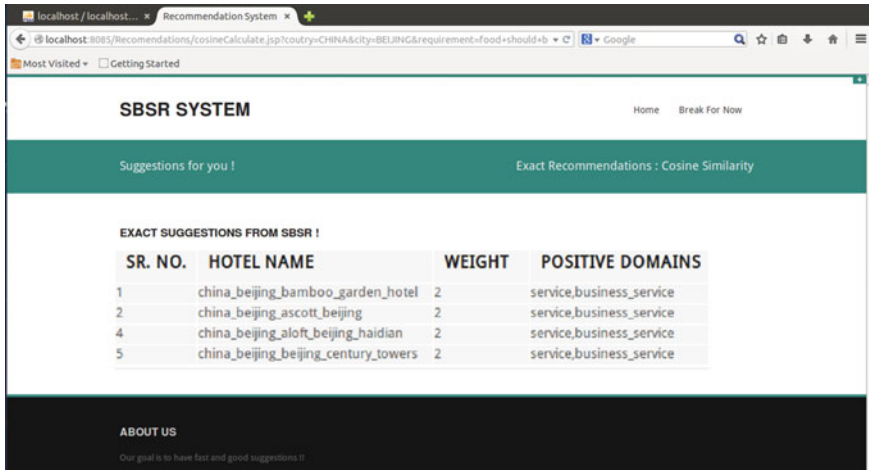


Fig. 4 Exact recommendation

6 Performance Evaluation

6.1 Comparison of the Recommendation System with and Without Semantic Analysis

For this comparison purpose, five hotels in the Beijing city and for each hotel 4 different aspects like rooms, service, location and business service were considered. Fig. 5 shows the weight vector table generated after processing the comments for these 5 hotels. For testing purpose the active user query was, “Rooms should be clean. Food should be tasty. Location should be pleasant and wifi should be there always.” After pre-processing this query the domains recognized were rooms, service, location and business-service. Based on this query and above weight vectors, the comparison results for recommendation generation with and without semantic analysis were noted as shown in Fig. 6. From the above drawn results following is the conclusion and description about how the recommendations vary from each other.

Sr. No.	Hotel Name	Categories/Aspects							
		Rooms		Services		Location		Business Service	
		Pcount	Ncount	Pcount	Ncount	Pcount	Ncount	Pcount	Ncount
1	china_beijing_autumn_garden_courtyard_hotel	3	5	1	2	0	3	3	4
2	china_beijing_ascott_beijing	28	15	9	11	26	10	15	6
3	china_beijing_bamboo_garden_hotel	42	20	27	10	42	23	22	3
4	china_beijing_aloft_beijing_haidian	6	1	0	0	2	3	4	0
5	china_beijing_beijing_century_towers	2	1	2	1	2	0	0	0

Fig. 5 Weight vector table for testing

Sr. No.	Hotel Name	Approximate Recommendation		Exact Recommendation	
		Without Semantic Analysis	With Semantic Analysis	Without Semantic Analysis	With Semantic Analysis
1	china beijing autumn garden courtyard hotel	Yes	No	Yes	No
2	china beijing ascott beijing	Yes	Yes	Yes	No
3	china beijing bamboo garden hotel	Yes	Yes	Yes	Yes
4	china beijing aloft beijing haidian	Yes	Yes	No	No
5	china beijing beijing century towers	Yes	Yes	No	No

Fig. 6 Comparison of approximate and exact recommendation with and without using semantic analysis

1. Approximate Recommendations

- Without Semantic Analysis—In this type of recommendation negative count is not considered. If any of the domains mentioned in the active user query matches, then the hotel is included in the recommendation list.
- With Semantic Analysis—Negative count about the hotel services is considered. If any of the domains mentioned in the active user query matches, then the hotel is included in the recommendation list. But if the positive count of that hotel in respective domain is less then that hotel is not included.

2. Exact Recommendations

- Without Semantic Analysis—In this type of recommendation negative count is not considered. The hotel is included in the recommendation list only if all the domains mentioned in the active user query are talked about in a particular hotel.
- With Semantic Analysis—Negative count about the hotel services is considered. If all of the domains mentioned in the active user query matches and if the positive count of all the domains is greater, then only the hotel is included in the recommendation list. But if the positive count of that hotel in respective domain is less then that hotel is not included in the recommendation list.

6.2 Comparison of Pre-processing Time with and Without Using Hadoop

To evaluate the system functioning, processing was carried out on the experimental dataset to test the working of the system on hadoop platform and without using it. The testing was carried out for five cities. Each city consisted of 5 hotels and multiple comments inside it. The processing of comments was done using hadoop platform and also without using hadoop platform and time required to complete the processing was noted down and accordingly the graph was plotted measuring the time in seconds on Y-axis and number of comments on X-axis as shown in Fig. 7. Also the speed-up of the system is calculated using the same results of processing time and it was concluded that if the processing is done on the Hadoop platform, the

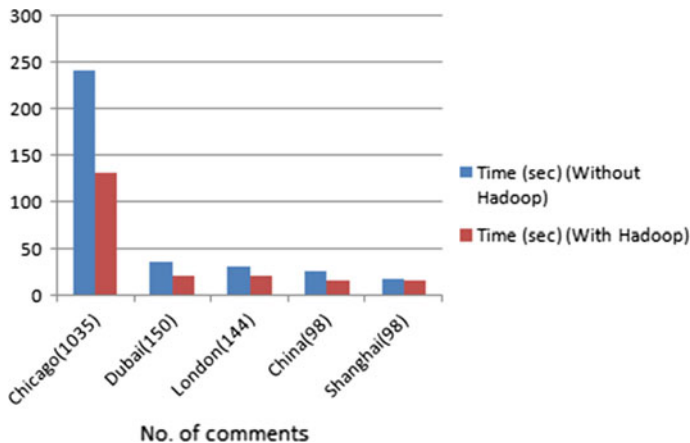


Fig. 7 Processing time required with and without using Hadoop

data processing is faster. The average speed-up of the system that was observed is 34 %. Also it is noticed that if the data size is larger, then the percentage speed-up was more, marking a favorable difference between processing on Hadoop platform and without it. The task of recommendation can also be divided among multiple nodes, which can decrease the processing and response time of the system and hence the efficiency of the system would increase. Also as the implementation of this system is on Hadoop platform which distributes its task across many map() and reduce() phases, the scalability of the system increases. This can lead to an increase in the overall performance of the system.

7 Conclusion

The SBSR system deals with generating the recommendations according to the personalized likings and taste of the users and by considering the multiple aspects of the service. The incorporation of semantic analysis of the previous user comments distinguishes the positive and negative preferences and avoids the negative comments to increase the accuracy of the recommendations. A comparative study is done to mark the difference between approximate and exact recommendation generation strategies with and without using semantic analysis. Thus, the results depict that the recommendations generated using semantic analysis are more accurate than without using it. As this accounts a large dataset, it is affected by the factors like scalability and inefficiency which is improved by 34 % by implementing the system in distributed platform known as Hadoop which uses MapReduce framework and can manage large amount of data in these service recommendation systems. The SBSR system shows a good accuracy, efficiency, and scalability when compared to other systems.

References

1. Shunmei Meng, Wanchun Dou, Xuyun Zhang and Jinjun 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, December 2014.
2. M. Bjelica, “Towards TV Recommender System Experiments with User Modeling”, *IEEE Trans. Consumer Electronics*, vol. 56, no. 3, pp. 1763–1769, Aug. 2010.
3. Y. Chen, A. Cheng, and W. Hsu, “Travel Recommendation by Mining People Attributes and Travel Group Types from Community Contributed Photos”, *IEEE Trans. Multimedia*, vol. 25, no. 6, pp. 1283–1295, Oct. 2013.
4. C. Lynch, “Big Data: How Do Your Data Grow?”, *Nature*, vol. 455, no. 7209, pp. 28–29, 2008.
5. G. Adomavicius and A. Tuzhilin, “Toward the Next Generation of Recommender Systems A Survey of the State of the Art & Possible Extensions”, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 6 pp. 734–749, 2005.
6. Ruchita V. Tatiya and Prof. Archana S. Vaidya, “A Survey of Recommendation Algorithms”, *International Organization of Scientific Research Journal of Computer Engineering (IOSR-JCE)*, Volume 16, Issue 6, Ver. V, PP 16–19, Nov–Dec. 2014.
7. ManishaHiralall, “Recommender systems for e-shops”, *VrijeUniversiteit*, 2011.
8. G. Adomavicius and Y. Kwon, “New Recommendation Techniques for Multicriteria Rating Systems”, *IEEE Intelligent Systems*, vol. 22, no. 3, pp. 48–55, May/June 2007.
9. Z. D. Zhao, and M. S. Shang, “User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop”, In the third International Workshop on Knowledge Discovery and Data Mining, pp. 478–481, 2010.
10. Peter Turney, “Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews”.
11. Kavita Ganesan and Cheng Xiang Zhai, “Opinion-Based Entity Ranking”, *Information Retrieval*, 2011. Comments dataset: <http://www.kavita-ganesan.com/entity-ranking-data>
12. For semantic analysis: <http://mpqa.cs.pitt.edu/lexicons/effectlexicon/>