*Journal of Scientific Research & Reports*
*10(4): 1-9, 2016; Article no.JSRR.24184*
*ISSN: 2320-0227*

**SCIENCEDOMAIN** *international*
www.sciencedomain.org

# Service Recommendation Based on Ranking Using Keywords in Hadoop

**V. Naveena[1*] and S. V. Kogilavani[1]**

*[1]Department of Computer Science and Engineering, Kongu Engineering College, Erode, Tamilnadu, India.*

***Authors' contributions***

*This work was carried out in collaboration between both authors. Both authors read and approved the final manuscript.*

***Article Information***

DOI: 10.9734/JSRR/2016/24184
Editor(s):
(1) Grigorios L. Kyriakopoulos, School of Electrical and Computer Engineering, National Technical University of Athens (NTUA), Greece.
(2) Martin J. Bohner, Missouri University of Science and Technology, Rolla, Missouri, USA.
Reviewers:
(1) Samer I. Mohamed, Msa University, Egypt.
(2) Shihan Yang, Guangxi University for Nationalities, China.
(3) Dominik Strzałka, Rzeszów University of Technology, Poland.
(4) Sivakumar Ramakrishnan, Universiti Sains Malaysia, Malaysia.
Complete Peer review History: http://sciencedomain.org/review-history/13700

***Original Research Article***

***Received 8th January 2016***
***Accepted 19th February 2016***
***Published 15th March 2016***

## ABSTRACT

Recommendation system acts as a tool in providing the most appropriate service to the user. Currently, information through online services is increased. This leads to the overhead of data in online and there is a possibility of getting less accurate prediction. In previous approaches, recommendation of services does not consider the suggestion of the user at a time, was in need of searching for the particular service. The proposed system deals with the implementation of personalized recommendation to provide services for hotel reservation system. Candidate service is created as the combination of keyword list and Domain Thesaurus which consist of semantically annotated words. Preferences are collected from the active user about particular service for each application. Similar user's opinions are taken from the reviews using keyword extraction method. Similarity is calculated between user preferences with reviews of the previous user using jaccard and cosine similarity measures. From this most similar keywords are provided to the user as a recommended service using MapReduce framework. It outperforms about 8% when compared with previous approaches, in providing the accurate prediction of relevant service to the active user.

**Corresponding author: E-mail: naveenavelumani@gmail.com;*

*Keywords: Keyword; preferences; recommendation system; Hadoop; MapReduce.*

## 1. INTRODUCTION

In Internet, amount of data increases day by day which leads to difficulty in analysis using data mining techniques. The sources of data can be a database, data warehouse, the web, other information repositories or data which are retrieved and stored in the system dynamically [1]. This causes inefficiency in retrieving vast amount of data and scalability problems. When datasets are humongous in size, a wide distribution of data is needed and complexity arises which leads to the development of parallel and distributed data-intensive mining algorithms [2]. Big Data Analytics is the process of computing such large dataset in parallel using MapReduce environment [3].

### 1.1 Opinion Mining

Opinion Mining (also refers to sentiment analysis) is the process of analyzing the text in the document and provides the suggestions to the people by extracting opinion through online [4]. Users post their opinion about the services or products in their respective blogs, shopping sites, or review sites. Reviews about hotel [5], automobiles, movies, restaurants are available on the websites. Text analysis in opinion mining is the process of getting high quality information from the text. Approximately, 90% of the world's data is available in unstructured format. By parsing this unstructured data, the patterns involved in it are identified and recommendations are provided.

### 1.2 Recommendation and Collaborative Filtering

Traditional system provides recommendation to particular application based upon the ranking given by the personalized user [6]. Now-a-days many applications use recommendation system which includes CDs, books, webpage, hotel reservation system, [7-9]. In hotel reservation system, if one user is concerned about particular service and another user is looking for different service in the same hotel. Then rating and ranking provided for the recommended service of both the users will be same. It is not a good recommendation system and people will not satisfy with the recommendation. Moreover, in hotel reservation system the rating of service and service recommendation list to the users are the same and do not consider the user preferences in recommending the service [10].

Recommendation system can be classified as content based, collaborative based and hybrid recommendation system. Content based recommendation provides recommendation system by taking the user preference from the previous user reviews. Collaborative Filtering (CF) recommendation service is based on the reviews of the previous user, by checking the similarity with the current user. CF is further classified as item-based CF and user-based CF. In item-based CF rating is provided based on the similar item rating by the same user and in user-based CF rating is predicted based on the same item rating provided by the similar user. Hybrid recommendation system combines recommendation of both content and CF based recommendation.

### 1.3 Big Data Framework

Cloud computing is an effective platform to facilitate parallel computing in a collaborative way to tackle large-scale data. Traditional data mining techniques are failed to process large or complex datasets. Big Data Analytics provides solution to this problem.

The main characteristics of Big Data are volume, variety, veracity and velocity. In Big Data, the large datasets are partitioned into small datasets. Each dataset is further processed in parallel, by searching the patterns. The parallel process may interact with one another. The patterns from each partition are eventually merged and produce the result. Most widely using Big Data Analytics tools is Hadoop [11]. It is the open source tool for MapReduce framework written in Java, originally developed by Yahoo. Nowadays everything acts as a service, so creating and recommending the service using big data analytics in the social networking will be more efficient and accurate. The File System used for storing large datasets are Hadoop Distributed File System (HDFS). In this by simply adding the servers can be achieved growth in storage capacity and computing power [12].

## 2. RELATED WORK

Recommendation is based on the people having similar preferences and interests (i.e. stable

ones) from past reviews [6]. It provides similarity computation using k-nearest neighbors. It uses user history profile as rows, their reviews as column and forms a rating matrix. Cosine similarity measure is used for representing the weight of the rank matrix, which is the number of interactions between rows and columns. Finally, calculate the item rating from the rank matrix of the neighbor user. The entire process is implemented in MapReduce framework to overcome scalability problem. It takes large computational time when dealing with huge amount of input data. So improvement must be done on Hadoop platform to reduce the computation time when dealing with these algorithms.

In item-based recommendation system using CF, rating is predicted based on similar items rating by the same user [7]. User-item matrixes are formed by finding relationship between different items and provide recommendation to the user. By considering the reviews of similar item the similarity between item-item is identified using cosine based similarity, correlation based similarity and adjusted cosine based similarity. Finally, predicted rating for the target user is calculated.

Keyword based service recommendation system [13] takes the preferences from the previous user keyword set and finds the similarity with the active user keyword set. Using CF, personalized rating for each service is considered and lists the top recommended services.

## 3. PROPOSED SYSTEM

The proposed system uses previous user reviews to find similarity with the active user and provide recommendation of service based on the active user needs [14]. First step is to form candidate service list for the application along with domain thesaurus i.e. semantic words [13]. Then collect the previous user posts in the form of reviews, which includes their opinion about the application. After the collection of reviews, a review sentence is given to data preprocessing stage. Data preprocessing consists of stop word removal and Part-Of-Speech (POS) tagging. The keywords obtained are taken as keyword set of previous user [15]. Meanwhile active user needs to provide the service as keywords. Next, the similarity between the active and previous user's preference keyword set is calculated. The similarity computation is done by jaccard and cosine similarity method [14]. Finally, personalized rating for each service of the active user is calculated as shown in Fig. 1 and recommend top-*k* rating is provided to the active user [6].

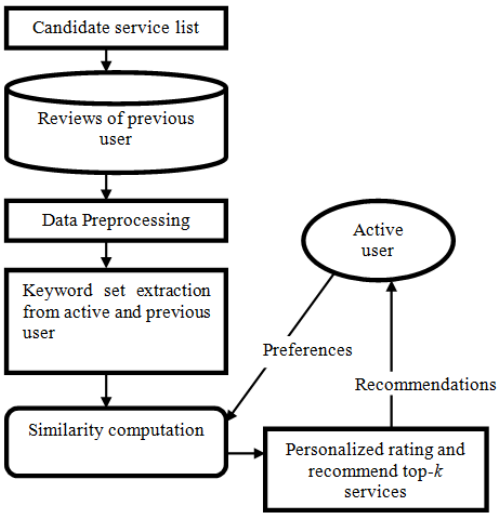


**Fig. 1. Architectural diagram**

The main steps for semantic based service recommendation system are described as follows**:**

**Step 1:**

Stop word removal involves removing of unwanted and low priority words in each review sentence. Reviews are stored in HDFS which is given as input to stop word removal process. Then each word is tagged using POS tagger.

**Step 2:**

Active users give their preferences of service as keywords by selecting from the candidate service list. From the active user preference services, keyword set is formed as Active Preference Keyword set (APK). Then correspondingly previous reviews will be transformed as Previous Preference Keyword (PPK) set along with semantic words. Keywords tagged as noun by POS tagger are extracted from the datasets [5]. The algorithm for keyword extraction is shown as follows:

```
keyword extraction (POS tagged input reviews)
    if word is a  noun then
        extract (word)
else
        remove (word)
    endif
```

### 3.1 Jaccard Similarity Measure

Jaccard similarity is an approximation method used for finding similarity between APK and PPK. It does not consider the repetition of keywords in the keyword set. It takes the extracted keyword set of different previous users and compares the similarity with the preference keyword set of active user. To calculate the similarity between APK and PPK, the jaccard similarity measure is given in algorithm is follows:

$$sim(APK,PPK) = \frac{|APK \bigcap PPK|}{|APK \bigcup PPK|} \quad (1)$$

*return sim (APK,PPK)*

In above equation (1), similarity between *APK* and *PPK* is given as, number of common keywords in *APK* and *PPK* divided by the number of all the keywords in *APK* and *PPK*.

### 3.2 Cosine Similarity Measure

It is an exact similarity method to find the highest similarity between active preference keyword set and previous preference keyword set. The number of times the particular keyword is repeated in the APK and PPK is taken as weight of the keyword. If the keyword is not available in the preference keyword set, then the weight of the keyword will be taken as zero (i.e. $w_{ij}$ = 0). The Term Frequency and Inverse Document Frequency (TF-IDF) is used for finding the number of times the particular term occurs in the document i.e. the frequency of the keywords. It can be taken as weight of the keyword in the keyword set. TF-IDF is calculated for both active preference keyword set and previous preference keyword set [6], [13].

TF-IDF in which Term Frequency (TF) takes the distinct keywords and number of times the particular keyword appears in the review and in the active keyword set is given by the following equation (2),

$$TF = \frac{N_{pk_i}}{\sum_g N_{pk_i}} \quad (2)$$

where, $N_{pki}$ number of times particular keyword appears in the keyword set, g is the number of keywords in the preference keyword set. Inverse Document Frequency (IDF) is computed by number of documents containing the keywords divided by the number of keywords present in that document. It is given by the following equation (3),

$$IDF = 1 + \log_e\left(\frac{N}{n_i}\right) \quad (3)$$

where, $N$ is the total number of reviews posted by the user, $n_i$ is the number of occurrence of the keywords in all reviews. TF-IDF scores for each keyword is calculated by the equation (5) as follows,

$$w_{pk_i} = TF * IDF \quad (4)$$

The weight of APK and PPK ( $w_{pk_i}$ ) is used to calculate the cosine similarity in the equation (5) defined as follows,

$$\text{sim(APK,PPK)} = \cos\left(\vec{W}_{AP}, \vec{W}_{PP}\right) \quad (5)$$

where, $\vec{W}_{AP}$ and $\vec{W}_{PP}$ be the weight of the keyword in the keyword set of the active preference and previous preference. The above equation can also be written as,

$$\text{sim(APK,PPK)} = \frac{\vec{W}_{AP} * \vec{W}_{PP}}{\| \vec{W}_{AP} \|_2 * \| \vec{W}_{PP} \|_2} \quad (6)$$

In cosine similarity method, similarity between *APK* and *PPK* is given as multiplication of weight vector of active preference with weight vector of previous preference divided by the square root of weight vector of active preference with the weight vector of previous preference.

### 3.3 Personalized Rating

Using CF algorithm [6], rating of each service is provided based on the cosine similarity value. The previous keyword set which is most similar to the active keyword set is filtered out from cosine similarity. Rating of each keyword using cosine similarity is calculated and it is used to provide the top-*k* rated service to the active user.

The personalized rating for each service of the active user is calculated as follows:

$$pr = \bar{r} + k \sum_{PPK_j \in R} sim(APK, PPK_j) * (r_j - \bar{r}) \quad (7)$$

where, $\bar{r}$ be the average rating of service, $r_j$ be the corresponding rating of the different previous user,

*sim(APK,PPK$_j$)* is the similarity between APK and PPK using cosine similarity measure. *k* is the normalizing factor and *R* is used to store the previous user after each filtration.

$$k = \frac{1}{\sum_{PPK_j \in R} sim(APK, PPK_J)} \quad (8)$$

## 4. IMPLEMENTATION ON MAPREDUCE

MapReduce framework used in [6,7,13] adopted to execute data in parallel manner. MapReduce can be used to implement keyword extraction, similarity method, raking of services in parallel. It reduces running time of the algorithm. The mapper and reducer function is implemented with the key and value pair as,

Mapper
Input: <k1, v1>
Output: list< k2, v2>
Reducer
Input: < k2,list< v2>>
Output: list< k3, v3>

where, k1,k2,k3 are keys and v1,v2,v3 are values. The map and reduce functions used in the proposed system is specified in Table 1 as below,

**Map-I and Reduce-I:** Each input review is taken as tuple. Similar user id tuples are allocated to the same node to calculating the average rating of each candidate service.

**Map-II and Reduce-II**: Compute the similarity between active and previous user.

**Map-III and Reduce-III:** To calculate the personalized rating for each candidate service and recommending the service list based on the rating value of each service.

**Table 1. Implementation of MapReduce**

| | |
|---|---|
| **Map-I&Reduce-I** | Map-I: Map<*i,j,r$_{ij}$,R$_{ij}$*> with same value of *i*, tuples are allocated to the same node and formed as <*j,r$_{ij}$,R$_{ij}$*>. Here *r$_{ij}$* is the rating of *R$_{ij}$*, *i*ε[1,N] and N is the number of candidate service, *R$_{ij}$* is the review commented on candidate service *i* by a past user *j*.<br>Reduce-I: Reduce<*i,j,r$_{ij}$,PPK$_{ij}$*, $\bar{r}_i$>, *i*∈ [1,N], PPK$_{ij}$ is the preference keyword set of the reviewer of *R$_{ij}$*, $\bar{r}_i$ is the average rating of candidate service *i*. |
| **Map-II&Reduce-II** | Map-II: Map<*i,j,r$_{ij}$,PPK$_{ij}$*, $\bar{r}_i$> on *i*ε[1,N] and tuples are allocated to the same node based on the value of i and formed as <*j,r$_{ij}$,PPK$_{ij}$*, $\bar{r}_i$ ><br>Reduce-II: It takes <*APK*> and <*j,r$_{ij}$,PPK$_{ij}$*, $\bar{r}_i$ > as input and forms output as *sim*=<*i,j,r$_{ij}$,s$^{ij}_{ASC}$*, $\bar{r}_i$ >, *i*ε[1,N], s$^{ij}_{ASC}$ is the similarity between active user *i* and *j* based on either jaccord or cosine similarity method. |
| **Map-III&Reduce-III** | Map-III: Map<*i,j,r$_{ij}$,s$^{ij}_{ASC}$*, $\bar{r}_i$> on i value allocated to the same node in the form as <*j,r$_{ij}$,s$^{ij}_{ASC}$*, $\bar{r}_i$><br>Reduce-III: It takes input of <*j,r$_{ij}$,s$^{ij}_{ASC}$*, $\bar{r}_i$ > and produce the output as ranking list<*pr$_i$,i*> where, *pr$_i$* is the personalized rating for each service *i, i*ε[1,N] and ranking is ordered based on service . |

## 5. EXPERIMENTAL EVALUATION

The dataset used in the experiment is real dataset [5] which consist of 4,35,666 reviews with 4,4676 user and 305 different hotels with overall rating of each hotel. The total input is split into 80% as training data with 20% as test data. The accuracy is measured by precision, recall and F-measures as shown below,

$$\text{Precision} = \frac{|ExtractedValues \cap TrueValues|}{|ExtractedValues|} \quad (9)$$

$$\text{Recall} = \frac{|ExtractedValues \cap TrueValues|}{|TrueValues|} \quad (10)$$

$$F-measure = \frac{2*\text{Recall}*\text{Precision}}{(\text{Recall}+\text{Precision})} \quad (11)$$

From the above equation (8), *Precision* is given by the intersection of number of values extracted with the number of true values obtained divided by the number of extracted elements. In equation (9), *Recall* is given by the intersection of number of extracted values with the number of true values obtained is divided by the number of true values. Fig. 2, represents accuracy in terms of precision, recall, F-measures values, which can be calculated for 20 sample reviews. Out of 20 review sentences, 7 are true value, 8 are extracted value and 6 are intersection of extracted vales and true values.

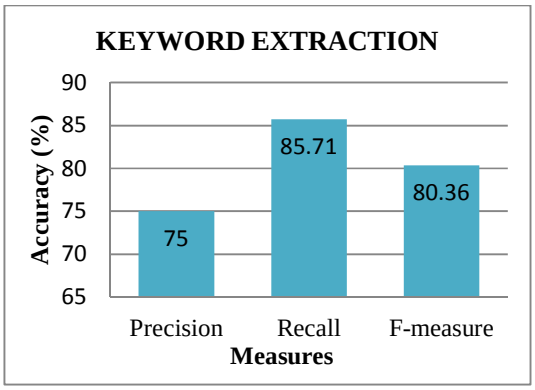


**Fig. 2. Keyword extraction**

The result is taken for similarity computation of APK with three different PPK keyword sets using jaccard and cosine similarity measures. Fig. 3 specifies the similarity value calculated using jaccard similarity measures and cosine similarity measures for 5 keywords (*furnished, large, spacious, room, apartment*). The result shows that cosine similarity measures provide the highest value for the keywords than jaccord similarity measures.

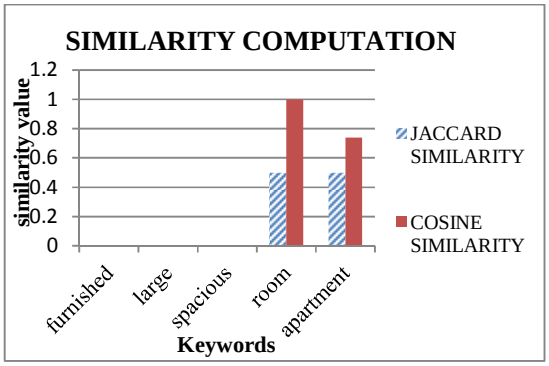


**Fig. 3. Similarity computation for PPK-1**

Rating of keyword for the most similar is rated using personalized rating, where the highest value gives the most needed keyword to the user. Semantic based service recommendation provides the most accurate rating as shown in Fig. 4,

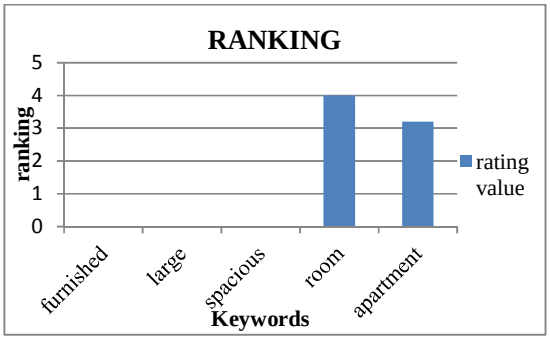


**Fig. 4. Ranking for PPK-1**

Fig. 4 shows the ranking for the keyword *room* is higher than *apartment* as needed by the active user preference. For second PPK set, similarity computation and ranking of keywords is shown in Fig. 5 and Fig. 6 as follows,

PPK-2 set finds most similar keywords when compared to PPK-1 and PPK-2 have three similar keywords according to the APK set as shown in Fig. 5. Ranking provided inPPK-2 for keyword *spacious* is higher, which is most similar keyword in APK set as shown in Fig. 6.

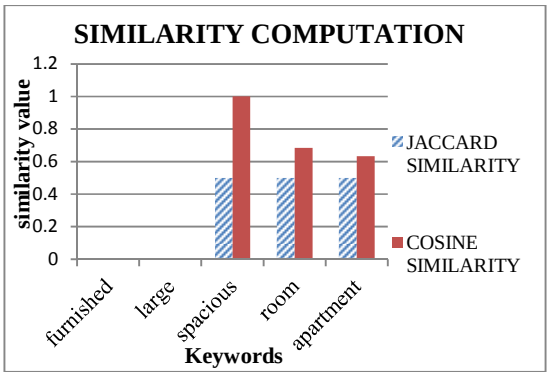


**Fig. 5. Similarity computation for PPK-2**

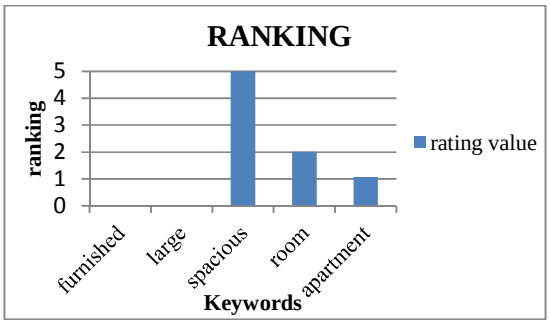


**Fig. 6. Ranking for PPK-2**

For PPK-3, the similarity computation and ranking of keywords is shown in Fig. 7 and Fig. 8 as follows,

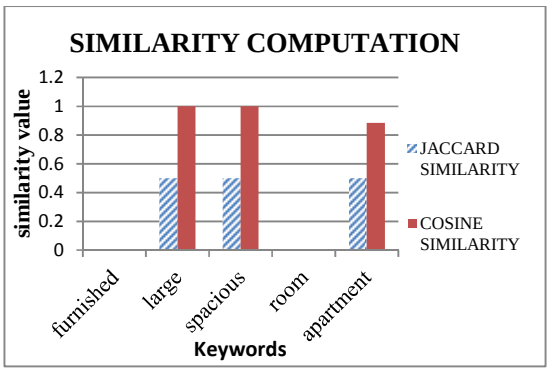


**Fig. 7. Similarity Computation for PPK-3**

In third PPK set, keywords extracted similar to APK contain three keywords (*large, spacious, apartment*) as shown in Fig. 7. Keywords matched exactly with APK set. When compared to PPK-1 and PPK-2, the keywords obtained in PPK-3 are different. Ranking for the keywords in PPK-3 is shown in Fig. 8, where keywords *large* and *spacious* contain the highest rating value as recommended by the active user. From the above result, cosine similarity measure provides highest rating than jaccard similarity measure and highest ranking for the keyword provide the needed result to the active user.

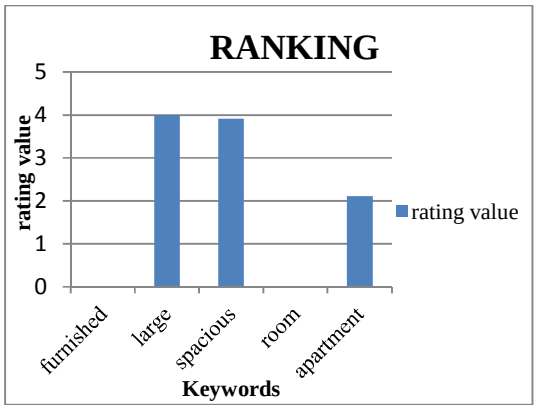


**Fig. 8. Ranking for PPK-3**

Execution time for a single mapper is higher for both similarity methods. By increasing the number of mapper, execution time is decreased in single node cluster as shown in Fig. 9,

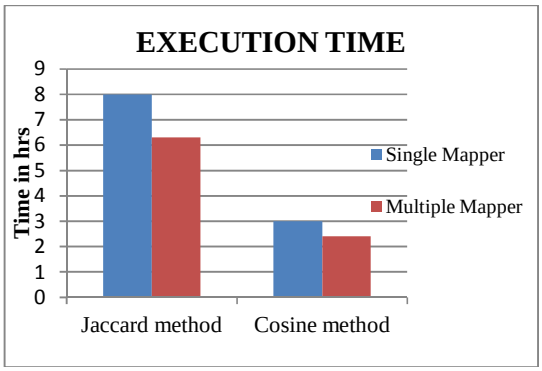


**Fig. 9. Execution time of similarity methods**

From the above Fig. 9, it reveals that jaccard takes maximum time of execution using single mapper but there is decrease in execution time of multiple mapper. Similarly, for cosine method the execution for single mapper is higher than multiple mapper. When comparing the execution of jaccard with cosine method, jaccard finds similar keyword from PPK with APK. So it takes longer time in execution than cosine method and cosine method takes the input of jaccard's. It leads to the decrease in execution time for cosine method.

Semantic based service recommendation system provides more accurate prediction than existing system (user-based pearson correlation) method. Semantic based service system uses cosine similarity value with user-based CF. Mean Average Precision (MAP) used as a parameter for predicting the accuracy of relevant item. It finds the Average Precision ($AveP_i$) by using precision values with the relevance keyword based on ranks where relevant items occur, which is further averaged over all queries to give MAP. MAP is explained in the equation (11) as follows,

$$MAP = \frac{\sum_{i=1}^{Q} AveP_i}{Q} \quad (11)$$

where, $AveP_i$ is the average precision of active user $i$, $Q$ is the number of active users. Average Precision of the top-$k$ recommendation list $AveP_K$ is given by the equation (12) as follows,

$$AveP_k = \frac{\sum_{k=1}^{K} (P_k * rel_k)}{number of relevant service} \quad (12)$$

where, $P_k$ is precision at cut-off $k$ in the predicted recommendation list, $rel_k$ is an indicator which is equal to 1 if service at $k$ in the predicted list contained in top-$k$ of real recommendation list, 0 otherwise. As shown in Fig. 10, recommended keywords at top-3 will be more matched to the keyword in APK set. MAP will decrease in relevance score based on increase in top recommendation value.

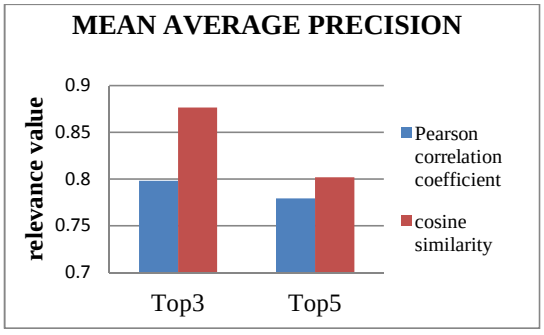


**Fig. 10. Mean average precision**

Cosine similarity using keywords outperforms 8% accuracy than pearson correlation method. Top recommended keyword will be more relevant to the active user keywords.

## 6. CONCLUSION

The proposed system recommends the most appropriate services to the active user based on the preference from the active user and previous user reviews. The PPK is extracted from the previous user reviews. Active user provides the services as APK. Similarity is computed between PPK with APK using jaccard and cosine similarity measures. Using personalized rating, rating for each service of the active user is calculated. From the results, cosine similarity measure provides the highest value than jaccard similarity measure. It overcomes issues of less accurate service from large dataset. It provides better results by implementing in Hadoop using MapReduce framework in single node.

In future, further implementation will be done by distinguishing positive and negative preferences separately by considering bigrams of words. It makes prediction more accurate to the user. Execution time can be still reduced by increasing the number of nodes.

## COMPETING INTERESTS

Authors have declared that no competing interests exist.

## REFERENCES

1. Manyika J, et al. Big data: The next frontier for innovation, competition, and productivity. McKinsey & Company Publications; 2011.
2. Lynch C. Big data: How do your data grow? CNI Publication. 2008;455(7209): 28-29.
3. Watkins, Andrew B. Exploiting immunological metaphors in the development of serial, parallel, and distributed learning algorithms. Diss. University of Kent at Canterbury; 2005.
4. Liu Bing. Opinion mining and sentiment analysis. Proc. Springer Berlin Heidelberg. 2011;2:459-526.
5. Available:https://archive.ics.uci.edu/ml/datasets/OpinRank+Review+Dataset
6. Zhao, Zhi-Dan, Ming-Sheng Shang. User-based collaborative-filtering recommendation algorithms on hadoop. Proc. IEEE 3rd International Conference on Knowledge Discovery and Data Mining. 2010;478-481.

7. Linden G, Smith B, York J. Amazon.com Recommendations: Item-to-Item collaborative filtering. IEEE Trans. Internet Computing. 2003;7(1):76-80.
8. Bjelica M. Towards TV recommender system experiments with user modelling. IEEE Trans. Consumer Electronics. 2010;56(3):1763-1769.
9. Alduan M, Alvarez F, Menendez J, Baez O. Recommender system for sport videos based on user audiovisual consumption. IEEE Trans. Multimedia. 2012;14(6):1546-1557.
10. Sikka R, Dhankhar A, Rana C. A survey paper on e-learning recommender system. International Journal of Computer Applications. 2012;47(9):27-30.
11. Lam Chuck. Hadoop in action. Manning Publications Co; 2010.
12. Ghemawat Sanjay, Howard Gobioff, Shun-Tak Leung. The Google file system. In ACM SIGOPS Operating Systems Review. 2003;37(5):29-43.
13. Meng S, Dou W, Zhang X, Chen J. KASR: A keyword-aware service recommendation method on MapReduce for big data applications. IEEE Trans. Parallel and Distributed Systems. 2014;25(12):3221-3231.
14. Turney, Peter D. Semantic orientation applied to unsupervised classification of reviews. In Proceedings of the 40th annual Meeting on Association for Computational Linguistics. 2002;417-424.
15. Singam J, Amaithi, Srinivasan S. Optimal keyword search for recommender system in big data application. ARPN Journal of Engineering and Applied Sciences. 2006;10(7).



*Peer-review history:*
*The peer review history for this paper can be accessed here:*
*http://sciencedomain.org/review-history/13700*