

**International Journal of**  
Engineering Research and Science & Technology



**ISSN : 2319-5991**

[www.ijerst.com](http://www.ijerst.com)

**Email: [editor@ijerst.com](mailto:editor@ijerst.com) or [editor.ijerst@gmail.com](mailto:editor.ijerst@gmail.com)**

This article can be downloaded from <http://www.ijerst.com/currentissue.php>

# Recommendation System for Hotels in Big Data Based on user Preferences

1M.SathyaDevi, 2Ch.Sarada

**Abstract:** The Recommendation System for Hotels is a valuable tool used to recommend hotels for users. In the existing system, hotel ratings and rankings are typically calculated based on reviews of previous users. Because of which the preferences of the current user are not considered anywhere. The current service recommender system presents the same ratings and services to different users without considering diverse users' preferences. Further, the present system will not work for a massive number of reviews. In this paper, a new computing approach is proposed to predict and recommend hotels based on user preferences, and it is scalable. In this paper, "User Preference based Comparison" algorithm is suggested to offer top hotels in services. MapReduce framework is used to implement it.

**Keywords:** User Preferences, Keywords, Recommendation system, MapReduce

## 1. INTRODUCTION

Hotel recommendation systems suggest the best hotels to the users.

It became an important research area because of a growing number of different services. Service recommendation frameworks appear as essential tool to assist and provide suitable service recommendations for the users. Examples of such recommendation systems include CDs, books, web pages, hotels and various other products [1]. The existing system does not consider user preferences while giving service recommendations. A novel approach is proposed to overcome this problem in hotel service recommendation. The proposed system provides service ratings based on user likings. In the new scenario, hotels' reviews are pruned, and a comparative analysis is done between pruned reviews and current user preferences to determine top hotels. A "Service Filter" algorithm is proposed to prune the assessments. Then

to accomplish a comparison, "User Preference based Comparison" algorithm is designed. In the real scenario, the proposed system needs to come across a huge number of reviews.

Hence, it is implemented in MapReduce framework. The entire work is accomplished in two phases. In the first phase, previous user reviews are trimmed using "Service Filter" algorithm. Service Filter algorithm filters only service/facility information and removes unnecessary information from reviews. In the Second phase of MapReduce, a personalized rating of each preference of the current user would be calculated using the comparison algorithm. Here the comparison algorithm compares calculated

1, 2 Department of Computer Science and Engineering, CVR College of Engineering Hyderabad, India  
satyamaranganti@gmail.com, sharada.ch@gmail.com

ratings with the current user preference ratings. Finally, it lists out top hotels based on user preferences. Further the paper is organized as follows: Section 2 provides a brief outline of the research work carried out in this area. Section 3 provides proposed system architecture. Section 4 gives details about the experimental evaluation. The conclusion and future work are briefly summarized in Section 5.

## 2. RELATED WORK

Service Recommendation system is a subdivision of information filtering system that predicts service ratings. Recommender frameworks have turned out to be prominent, and are used in different zones including hotels, movies, music, news, books, investigating articles, looking inquiries, political labels, and items in general [1]. Recommender framework characterizes a method for helping and expanding the typical procedure of utilizing proposals of other users to settle on decisions when there is no adequate individual information or experience of the options [2]. Various approaches exist to develop recommendation systems. Well-known procedures are Collaborative Filtering, Content-based Filtering or Hybrid Filtering [3]. Collaborative Filtering (CF) technique is the most developed and the most usually implemented one. Collaborative Filtering suggests things by recognizing different clients with comparable taste; it utilizes their sentiment to prescribe things to the dynamic client. Collaborative Filtering has two variants like user-based Collaborative Filtering and item-based Collaborative Filtering. In user-based Collaborative Filtering, the forecast of the rating of a thing for a client relies on the evaluations of a similar item appraised by a comparable client. Various similarity measures used in Collaborative Filtering [7] are Euclidean Distance, Pearson Correlation, Tanimoto Coefficient, Uncentered Cosine, City Block, Log Likelihood and Spearman Correlation. Item-based Collaborative Filtering is a type of Collaborative Filtering for recommender systems filtering

the likeness between things computed utilizing individuals' appraisals of those things.

Cold-Start problem is the major challenge with Collaborative Filtering. The Cold-Start problem says that the system may not know enough about the new users to decide who is similar [9].

Content-Based (CB) methodologies match content resource to customer qualities. Content-Based

dividing techniques ordinarily build the estimate in light of customer's data [4,5]. A framework that utilizes content-based separation to enable clients to discover data on the Internet include Letizia [6]. In hybrid Filtering, both CF and CB are combined to get the benefit of both collaborative and content-based filtering.

Despite the presence of many service recommender frameworks, that exhibit similar evaluations and rankings of services to various clients, they do not consider differing clients' inclinations, and consequently neglecting clients' customized necessities. Moreover, the existing system is not scalable. Hence, a novel approach is proposed for the Hotel Service Recommendation System wherein recommendations are made based on user preferences. Hence, this paper proposes the service system which is scalable.

Hadoop MapReduce is a framework for effectively composing distributed applications which can process a vast amount of data in parallel on large clusters of commodity hardware in a fault-tolerant way. HDFS (Hadoop Distributed File System) splits the input data-set into sovereign pieces which are processed by mappers in a parallel manner. The system sorts the yields of mappers, which are then inputted to the reducer. Usually, both the data and the yield of the job are put away in a file-system. The framework takes care of task scheduling, monitoring them and re-executes the failed ones.

## 3. PROPOSED SYSTEM ARCHITECTURE

The proposed framework prescribes hotels considering client inclinations. The

architecture of the proposed system is designed and shown in figure 1. Two-phase MapReduce structure is utilized to develop the system. The First phase of the scheme determines pruned reviews from reviews of previous customers. The Second phase of the MapReduce prepares a hotel service recommendation list.

In the first phase of MapReduce, reviews get split among mappers and each mapper prunes the reviews of its split. Subsequently, pruned reviews of mapper yield will go as an input to the reducer. Reducer composes all the pruned audits into a file.

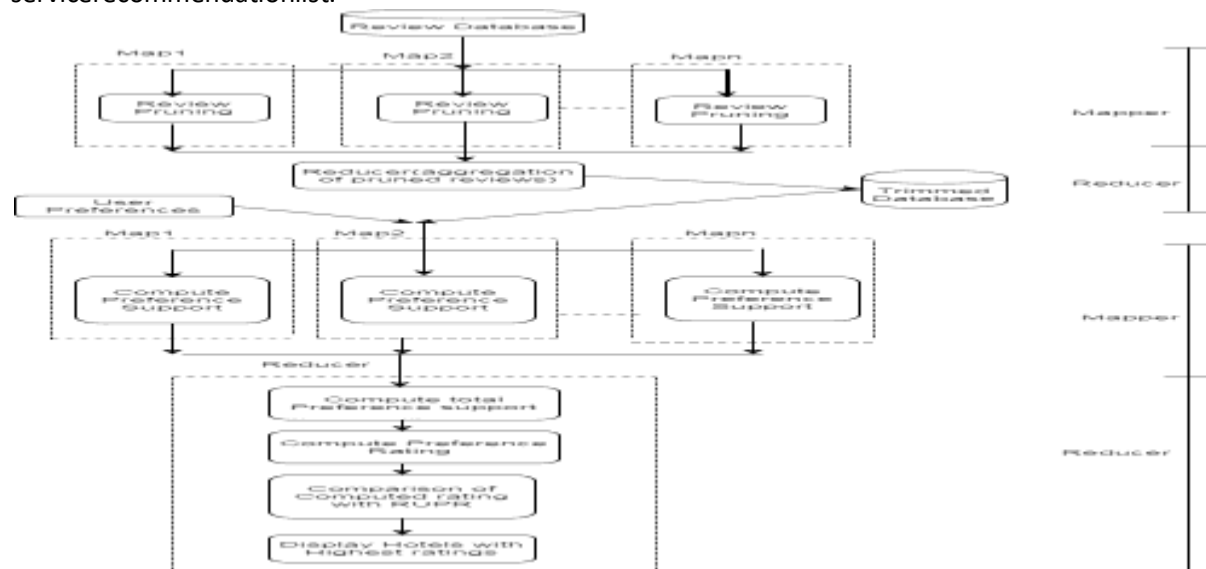


**Fig1: System Architecture**

In the second phase of MapReduce also data gets split among mappers. However, trimmed reviews are used as data. Support for each current user preference is determined at each mapper. Subsequently, the reducer aggregates the total support, calculates preferred service ratings, then compare them with the user

required preference ratings and consequently produce a final recommendation list.

In the proposed system two algorithms are used like "Service filter" for review pruning and "User Preference based comparison" for determining preference support from previous users. Table 1

illustrates various symbols used in this paper and their meanings.

**TABLE 1: Symbols and their Meanings**

| SYMBOL  | MEANING                                  |
|---------|------------------------------------------|
| PR      | Pruned Review                            |
| R       | Review                                   |
| PURS    | Previous User Reviews                    |
| SERVICE | Any Service or Facility of hotel         |
| PUR     | Previous User Review                     |
| PRS     | Pruned Reviews                           |
| CUPS    | Current user Preferences                 |
| PRGPS   | Pruned Review Groups based on Hotel Name |
| PRGP    | Pruned Review Group                      |
| P       | Preference                               |
| RPR     | Required Preference Rating               |

|     |                            |
|-----|----------------------------|
| HN  | HotelName                  |
| APR | Aggregate PreferenceRating |
| TNR | TotalnumberofReviews       |

Table2:SampleService\_Facility\_Listof

| S.No | Service_Facility | S.No. | Service_Facility |
|------|------------------|-------|------------------|
| 1    | Room             | 6     | Wi-Fi            |
| 2    | Shopping         | 7     | Beach            |
| 3    | Cleanness        | 8     | Transportation   |
| 4    | Airport          | 9     | Gym              |
| 5    | Environment      | 10    | Family           |

### 3.1 ReviewPruning

Table2:SampleService\_Facility\_Listof

| S.No | Service_Facility | S.No. | Service_Facility |
|------|------------------|-------|------------------|
| 1    | Room             | 6     | Wi-Fi            |
| 2    | Shopping         | 7     | Beach            |
| 3    | Cleanness        | 8     | Transportation   |
| 4    | Airport          | 9     | Gym              |
| 5    | Environment      | 10    | Family           |

The crucial work of the first phase of MapReduce is review pruning. In this phase, unnecessary information is wiped out from reviews. Service filter algorithm is used to do pruning & it is presented in algorithm 1.

#### Algorithm1:Servicefilter

7. EndFor

8. If PR not empty

then Write PR to the mapper output file

9. EndFor 10. End of Mapper

Mapper\_output: Pruned Reviews of an associated split Sorting\_and\_Shuffling:  
Pruned reviews are arranged in ascending order concerning the hotel name Reducer:

Reducer\_Input: Pruned Review (PRS)

1. For  $\forall PR \in PRS$

2. Write PR to the reducer output file

3. EndFor 4. End of Reducer

This algorithm extracts status information about each service/facility from the previous user reviews. Table 2 shows SampleService\_Facility\_List of hotels. Its input is previous user reviews (PURS) and the output is Pruned Reviews (PRS).



Input: Previous user reviews from websites like [www.tripadvisor.in](http://www.tripadvisor.in) or from other sources (PURS) Mapper:

1. Initialize PR = empty
2. For  $\forall PR \in PURS$
3. For  $\forall SERVICE \in SERVICE\_AND\_FACILITY\_LIST$
4. If SERVICE is found in the review R
5. PUR = Filter the user review on SERVICE
6. Append PUR to the PR

MapReduce approach is used here due to the need of processing millions of reviews. Every Mapper process prune reviews of previous users. It is an iterative process. In each iteration, one review is processed. Each review may contain feedback about multiple hotel services. The above algorithm finds information about every service written in the review. The resultant review can be called as pruned view. Every Mapper prune a set of

Reviews. Output of all Mappers is given as input to the Reducer. Sample review and the Fig.2 Sample review  
Output of above review:

Reducers simply take input from all Mappers and store them in the pruned reviews (PRS) file.

### 3.2 Calculation Of Service Ratings

In the second phase of MapReduce, for each current user preference, rating is calculated. Then the calculated ratings of preferred services of the current user are compared with his/her required preference ratings. At last, the hotel recommendation list will be out. To accomplish this, "User Preference based comparison" algorithm is presented in algorithm 2.

User Preference based comparison procedure prepare stop Hotel list based on ratings of hotel services against User Preference ratings. It uses MapReduce approach to do this comparison process. Every Mapper determines support metric of every User Preference of every hotel. At every Mapper, iterative procedure is applied on every review to determine support count of every User Preference of every hotel.

Reducer takes service count of every User Preference of every hotel as input from Mappers. It determines total support of every User preferred service of every hotel. Then it will calculate service rating of every hotel for every User Preference specified. Followed by it arranges calculated ratings in descending order and segregates them with respect to Hotel. Finally, it lists out top hotels that satisfy service ratings against User Preference ratings.

Algorithm 2: User Preference based comparison  
Input: Pruned Reviews-

PRS, Current User preferences-CUPS Mapper:  
3.

- PRGPS Divide Pruned Reviews into groups concerning Hotel names
4. For  $\forall PRGP \in PRGPS$
5. For  $\forall PC \in CUPS$
6.  $HN \leftarrow$  Store Hotel Name of  $PRGP$
6. For  $\forall PRE \in PRGP$
7. If  $P$  found in  $PR$  Then  $Support \leftarrow Support + 1$
8. End For  $PGRP$

concerned pruned review is shown here.

9.

$KEY \leftarrow$  Concatenate  $HN, P$ , and  $RPR$  with space as separator

11. Write( $KEY, Support$ ) into mapper output

12. End For

CUPS 13. End For  $PRGPS$  14. End of Mapper

Mapper\_Output: Support for all CUPS based on hotel name.

Sorting\_and\_Shuffling:

1.

$HNS, P, RPRS$  and the calculated Support are reshuffled and sorted so that the support for  $P$  from all mappers will form as one record. In this record  $HNS, P$ , and  $RPR$  forms the key and Support for the associated preference from mappers will become the value.

Reducer:

Reducer\_Input:  $KEY$  that contains  $HNS, P, RPRS$

:  $VALUE$  that holds Calculated Support for  $P$  from

all mappers 1. For  $\forall KEY \in KEYS$

2.  $Sum \leftarrow 0$

For  $\forall VALE \in VALUES$

$Sum \leftarrow Sum + VALE$  End For  $ASR \leftarrow$

$Sum / (TNR)$  If  $ASR \geq RPR$

$SR\_TABLE HN \leftarrow$  Store  $ASR$  into the next column of

$ST\_TABLE$  at the row corresponding to  $HN$  End For

3.

$SR\_TABLE$  is sorted in the descending order of  $ASR$  using Multi-

Columns sorting technique Print top three Hotel Names along with  $APR$  for each CUP

4. End of Reducer

Output: Top 3 hotels recommendation list

### 4. EXPERIMENTAL EVALUATION

The proposed system is portrayed under the MapReduce framework. In this section, the points of interest of sample hotel reviews and pruned reviews appear. At last, this section exhibits the result of the experimental study. Proposed system is portrayed under MapReduce framework. In this section, the points of interest of

sample hotel reviews and pruned reviews are appeared. At last, the result of experimental study is exhibited.

#### 4.1 Reviews

Hotel reviews are taken from the site [www.tripadvisor.in](http://www.tripadvisor.in). This analysis considers about 100,000 surveys for the study. Figure 3 shows some example reviews.

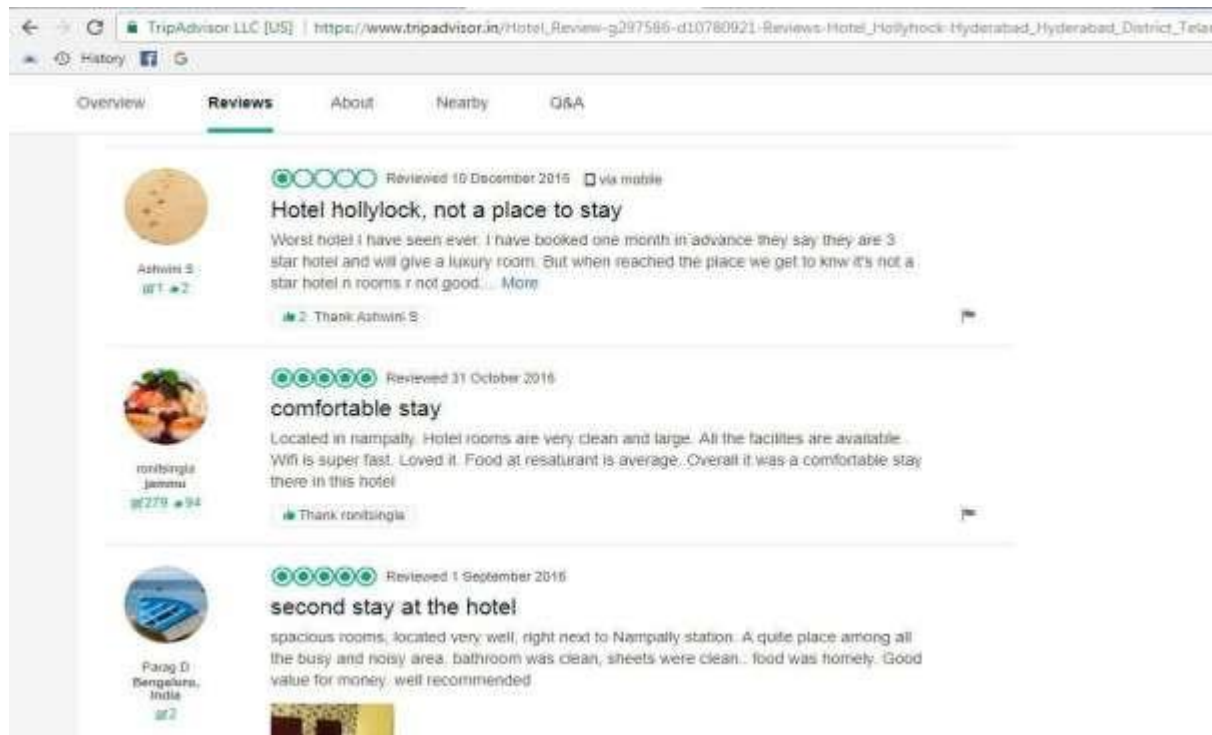


Fig3. Sample reviews

#### 4.2 Experimental Analysis

In the exploratory examination, reviews are pruned first then ratings for preferred services/facilities are calculated and compared with user required preference ratings.

##### 4.2.1 Pruned Reviews

Review pruning process extracts service/facility information from the original reviews. Pruned reviews are essential to determine service ratings. Table 3 shows pruned reviews of surveys listed in figure 3.

Table 3: Pruned reviews

|         |                                                                                               |
|---------|-----------------------------------------------------------------------------------------------|
| Review1 | Hotel hollylock, Hyderabad, worst hotel                                                       |
| Review2 | Hotel hollylock, Hotel rooms are very clean, wifi is super fast, food is average              |
| Review3 | Hotel hollylock, spacious rooms, located very well, next to Nampally station, food was homely |

##### 4.2.2. Final Recommendations

In this experiment, ratings of preferences of the current user are calculated based on pruned reviews

and compared them with their required preference ratings. At last, the recommendation list is displayed along with ratings of each preferred service/facility of the current user. Table 4, shows the hotel service recommendation list along with ratings for the following user preferences:

Table4:Userinputwithpreferencesalongwithrequiredratings

| ServicePreferred | Preferredrating |
|------------------|-----------------|
| GoodService      | 35%             |
| Friendlystaff    | 25%             |
| Cleanliness      | 20%             |

Table5:Finallistofhotelservicerecommendations

| HotelName         | Review        | Rating% |
|-------------------|---------------|---------|
| CentralCourtHotel | Clean lobby   | 45%     |
|                   | Friendlystaff | 28%     |
|                   | Serviceisgood | 25%     |
| TajDeccan         | Clean lobby   | 42%     |
|                   | Friendlystaff | 26%     |
|                   | Serviceisgood | 23%     |
| TajKrishna        | Clean lobby   | 39%     |
|                   | Friendlystaff | 25%     |
|                   | Serviceisgood | 22%     |

## 5. CONCLUSIONANDFUTUREWORK

Theproposedsystemrecommendshotelsbased onuserpreferences.Thewholesystemisimple mentedintheMapReduce environment, to make it scalable. In the proposed solution, “Service Filter” algorithm is the pivotalone,anditisusedtoprunethesurveys,yeti tisn'tperformingpruningflawlessly.Asapartoffu turework,wewouldlike toimprovethisalgorithm.

## REFERENCES

- [1] [https://en.wikipedia.org/wiki/Recommender\\_system](https://en.wikipedia.org/wiki/Recommender_system)
- [2] Resnick P, Varian HR. Recommender systems. Commun ACM 1997;40(3):56–8.<http://dx.doi.org/10.1145/245108.24512>.
- [3] AcilarAM,ArslanA.AcollaborativefilteringmethodbasedonArtificialImmuneNetwork.ExpSystAppl2009;36(4):8324–32.
- [4] MinSH,HanI.Detectionofthecustomer time-variantpatternforimprovingrecommendersystem.ExpSystAppl2010;37(4):2911–22.
- [5] CelmaO,SerraX.FOAFingtheMusic:bridgingthesemanticgapinmusicrecommendation. WebSemant:SciServAgentsWorldWideWeb2008;16(4):250–6.
- [6] LiebermanH.Letizia:anagentthatassistswebbrowsing.In:Proceedingsofthe1995internationaljointconferenceonartificialintelligence. Montreal,Canada;1995.p.924–9.
- [7] SaikatBagchi,PerformanceandQuality AssessmentofSimilarityMeasuresinCollaborativeFilteringUsingMahout.ProcediaComputerScience50 (2015)229– 234.
- [8] <https://www.Tripadvisor.com>.
- [9] Cold-startProblem in Collaborative Recommender Systems:EfficientMethodsBasedonAsk-to-rateTechnique.