ISSN (Print) : 0974-6846
**Indian Journal of Science and Technology,** *Vol 8(29), DOI: 10.17485/ijst/2015/v8i29/86257, November 2015* ISSN (Online) : 0974-5645

# Scalable Recommendation Engine for Optimized Product Discovery

**K. Harinath* and M. Kumaran**

Department of C.S.E, Jaya Engineering College, Thiruninravur, Chennai - 602024, Tamil Nadu, India;
hariganesh94@gmail.com, kumaran.ma@gmail.com

## Abstract

Recommending products to the user in e-commerce sites must be based on the user's taste and buying pattern. When a new user steps into the site, the system is unable to generate recommendations for that user, which is termed as cold start problem. The main objective is to introduce an approach that provides customized recommendations even to a new user thereby solving the cold start problem. Both the cold start problem and customized recommendation approach in the system can be addressed by combining two approaches. The first approach is building a graph relation which helps in knowing the taste of the user and thus overcomes the cold start problem and the second approach is harnessing the hidden potential from the review corpus of e-commerce sites. Review corpus consists of details such as rating, reviews of various users, each with a peculiar taste. When the user's taste and opinion polarity of the product is converged together it will lead to an optimized product recommendation to the user. The system uses hadoop, a scalable framework which enables to get recommendation in near real time. Orient DB is used for building graph relations. During testing this approach worked well with the cell phone accessories and clothing domain.

**Keywords:** Content based filtering, Opinion Mining, Recommendation Engine, Sentimental Analysis

## 1. Introduction

Recommender System is gaining more attention in the field of e-commerce. Lot of people have started buying products online. This field is a subset of information retrieval and natural language processing techniques. Recommendation stack is vital in e-commerce because it helps to bring more profit, to find relevant items in the list and to sell more items instantly. It is a win- win model for the both seller and buyer. Recommender system is not only used in the world of e-commerce but also in other fields like suggesting a movie to a user, providing relevant news feeds, recommending a new song to a user, etc. Despite garnering a lot of attention and research for the last two decades, this field is still filled with issues, like information overload and inability to provide personalized recommendation to the user. Thus to improve recommendation engine, the system must focus on user modeling and item characterization.

The latest trend is that, before buying from e-commerce sites people actively give and take opinions from web forums about the various products that they intend to buy. Example, they express their views about the product which they have bought. If a query about a particular product in any search engine is made, then the results are obtained from multiple sources and in each result there will be many reviews. Thus a systematic method is required to extract the summary from this large volume of unstructured text. Opinion mining is the one which gets the summary about a product based on the reviews expressed in various sites and web forums by various customers. It is a sub stream of web content mining and it involves information retrieval, text mining, and natural language processing techniques. It can be used by both companies and the individuals. From a user point of view, it is used to get summary about a particular product and to know what others think. From a company's point of view, they can find why a particular product of them is

* *Author for correspondence*

not moving well in the market and how to improvise their product's features.

## 2. Existing Algorithms and Their Limitations

### 2.1 Various Recommendation Algorithms

Most of the recommendations rely on the rating given by the users. In the system, the rating is obtained in an explicit way while user behavior and the buying history of the user are collected in an implicit way. Rating is not always done by the user who buys the product. It can also be done by user who just uses it. It can be a biased rating due to several reasons like no proper awareness about the product or the user being fanatic over the item. This makes the recommendation engine to perform inefficiently. The quality of data mining algorithm depends on the quality of the data[1]. Consider that the user has rated a set of movies out of five. The rating for the item by the user can be formulated in matrix format. The rating table or utility matrix is shown in Table 1.

**Table 1.** Utility matrix

| | Back to the future | Walk to Remember | October Sky |
|---|---|---|---|
| A | 5 | # | 3 |
| B | 2 | 4 | 3 |
| C | # | 2 | 4 |

The matrix contains null value # which represents the fact that the user has not rated. This problem may look small but in real time, the matrix may contain millions of rows and large number of columns. Sparse matrix is another problem in generating recommendation. The frequently used algorithm in recommendation is explained below.

#### *2.1.1 Content Based Algorithm*

Content based algorithm works on the principle that it recommends items based on the customer's buying history and his user profile. The recommendation is done using the attribute of the items which match the user's preference and taste that is stored in his profile.

This type of algorithm is habitually used in recommending news document in the feed. There can be news about various topics on news websites. When the system understands the preference of the user, it can recommend news that is relevant to the user. The relevance of the recommendation can be understood by positive and negative feedback. Positive feedback indicates that the user has interest with the item that he recommends and negative feedback means lack of interest on an item. There are several advantages of content based filtering. It is used to recommend an item which is new to the market, based on the item's attribute . The working principle is simple and can be implemented easily. The most important factor is that the designer of the system must have profound domain knowledge and expertise in choosing which attribute to consider, otherwise the system would recommend the item which does not fit the user behavior and would suffer from cold start problem.

While Recommending news documents, keywords are considered as attributes. All keywords in the document are indexed based on the frequency of occurrence and a new news document is recommended if it matches with same amount of frequency. This analysis suffers from

- Polysemy, the presence of multiple meanings for one word.
- Synonymy, multiple words with the same meaning.

For specifying keyword weights, Term Frequency - Inverse Document Frequency (TF-IDF) is used. Frequently occurring term is considered to be important.

**Term Frequency (TF)**, measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length as a way of normalization.

TF (t) = (Number of times term t appears in a document) / (Total number of terms in the document) (1)

**Inverse Document Frequency (IDF),** measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. IDF can be computed using (2).

IDF (t) = log [(Total Number of Documents / Number of Documents with term t)] (2)

#### *2.1.2 Collaborative Based Algorithm*

Collaborative filtering works with the help of other users in the system. They analyze the neighborhood for similar tastes. Now the system can generate recommendations with the help of the peer. This algorithm suffers from finding the center in the cluster and in deciding the

cluster size in the system. This cluster formation has to update over a period of time, this situation becomes more complicated when the number of user in the system increases. There can be user-user collaborative filtering and item-item collaborative filtering. Collaborative algorithm can be memory based or a model based one. Memory based technique uses the similarity of users or items. While model based technique uses machine learning algorithm and takes in the training set to find pattern.

### *2.1.3 Hybrid Algorithm*

Hybrid algorithm uses the technique of both content and collaborative algorithm. It overcomes the limitation of content based and collaborative algorithm but leads to an increase in the complexity of the system.

## 2.2 Opinion Mining

Opinion mining can be done at document level, word or phrase level. Mining in document level will help to find only the overall subjectivity. Word or phrase level is also not efficient. But there is another approach which is called feature based opinion mining in which aspects (features) of the product is considered and the opinion about each feature can be found[2]. This will help the customers to find the summary of a product in a better way as the user will buy the product for a particular set of features in which he/she is interested and not based on the overall opinion about the product. Feature based opinion mining will give fine-tuned results, hence it is chosen for the proposed approach. Feature based opinion mining basically consists of the following steps as shown in the Figure 1.

- Identifying features of the product from the unstructured review
- Finding polarity of the opinion i.e. positive or negative opinion
- Summarizing the product's features and their overall opinion

Basically the work of feature extraction can be categorized into two: Supervised and unsupervised approaches. In supervised approaches the problem is that it works well for the domain in which the system is trained. For other domains, the system needs to be retrained. Some of the previously used supervised approaches are based on CRF (conditional random field, HMM (Hidden Markov models)[3,4] and other techniques.

An approach that clusters product features and opinion words simultaneously and iteratively by fusing both their content information and sentiment link information was framed[5]. Under the same framework, based on the product feature categories and opinion word groups, sentiment association set between the two groups of data objects was constructed by identifying their strongest sentiment links. Thus the hidden links are considered to be the implicit features. There are various other issues involved while extracting features. Example, Non noun features are not addressed, rare but valid features are missed since most work is based on finding frequent features. Grammatical mistakes done by the user while writing reviews will not lead to efficient results after POS tagging.

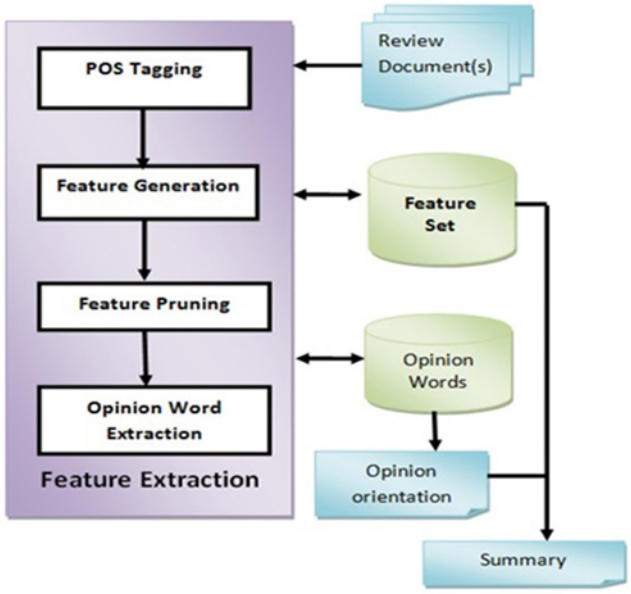


**Figure 1.** Feature based opinion mining.

PMI (Point wise Mutual Information) and SO (Semantic Orientation) are two important measures which are used in most of the works for opinion polarity extraction. Opinion words can be found by extracting adjectives, adverbs from the review sentences. After extracting opinion words, pruning is required to get valid features. Finally a summary is given based on the extracted set of opinion words and features. Topic modeling technique like LDA (Latent Drichlet Association) can also be used to solve feature based opinion mining tasks.

# 3. Proposed System

## 3.1 Domain Scoping

Our proposed system would offer recommendation in e-commerce sites. Accuracy rate will be the highest in the clothing and cell phone accessories system while the other domains will feature lesser accuracy as of now but will be made higher in the near future. The input to the system comes from the user review and rating. The user review can be analyzed through A/B testing.

## 3.2 User Profile Matching

As real time data is enormous, system schema must be flexible enough to store large number of user data and item data.

## 3.3 Graph Based Model

Enough ratings have to be collected before a recommender system can really understand user preferences and provide accurate recommendations[6]. To implement the graph based model in the product which can overcome new user problem and provide better recommendation, the multi model database Orient DB is used. It contains both Document and Graph based Database which makes it more powerful than other No Sql database. It can store up to 220,000 records per second on common hardware[7]. With the existing feature we can traverse the entire node on the graph in few milliseconds. In real world, data is very unpredictable yet it is important to capture all the details. So schema-less structure is required to store all the content into the database. The proposed approach have Document model for holding the review and rating but for storing the user and product details, Graph model in graph database is used. The nodes and edges contain the information. Table 2. Shows the difference between the models.

Figure 2 shows the overall architecture of the proposed recommendation engine.

**Table 2.** Comparison between relational, graph and document based model

| Relational model | Graph model | Document based model |
|---|---|---|
| Table | Class that extents 'v' or 'e" | Class or cluster |
| Row | Vertex | Document |
| Column | Vertex and edge property | Document field |
| Relationship | Edge | Link |

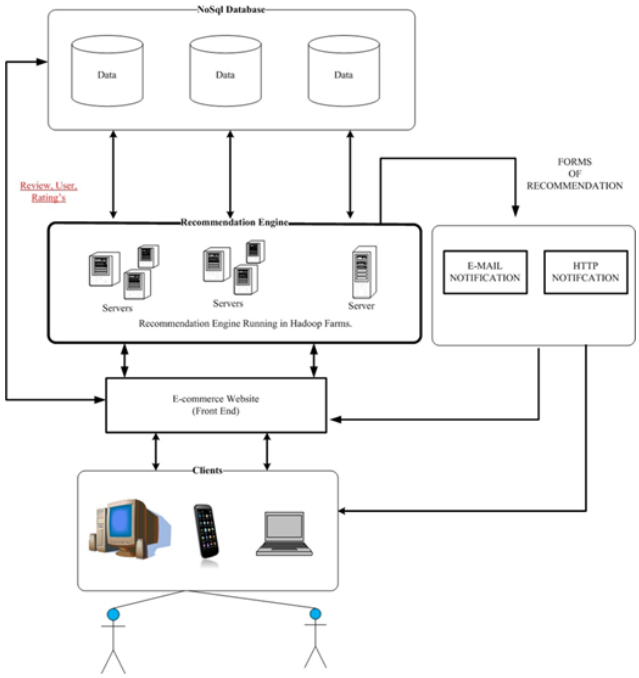


**Figure 2.** Overall architecture of proposed recommendation engine.

## 3.4 Opinion Mining

Feature extraction in the proposed approach for opinion mining is an extension of IEDR (Intrinsic and Extrinsic Domain Relevance) which was used in IEDR approach[8]. Opinion features are identified by exploiting their distribution disparities across different corpora. Domain Relevance (DR) of an opinion feature across two corpora (Domain dependent and domain independent corpus) is evaluated. DR is used to prune the valid features which are relevant to the domain and not over generic feature which is mentioned in a general topic. Example: "I don't have money to buy this", "I am a fan of this phone". Fan and money are generic terms and need not be considered as valid features of the product. First, several syntactic dependence rules are used to generate a list of Candidate Features (CF) from the given domain review corpus, for example, cell phone or hotel reviews. Next, for each recognized feature candidate, its domain relevance score with respect to the domain -specific and domain independent corpora is computed, which is termed as intrinsic-domain relevance (IDR) score, and the extrinsic domain relevance (EDR) score, respectively. In the final step, candidate features with low IDR scores and high EDR scores are pruned. This interval is called as the intrinsic and extrinsic domain relevance (IEDR) criterion, which is used to prune the valid opinions as shown in the Figure 3.

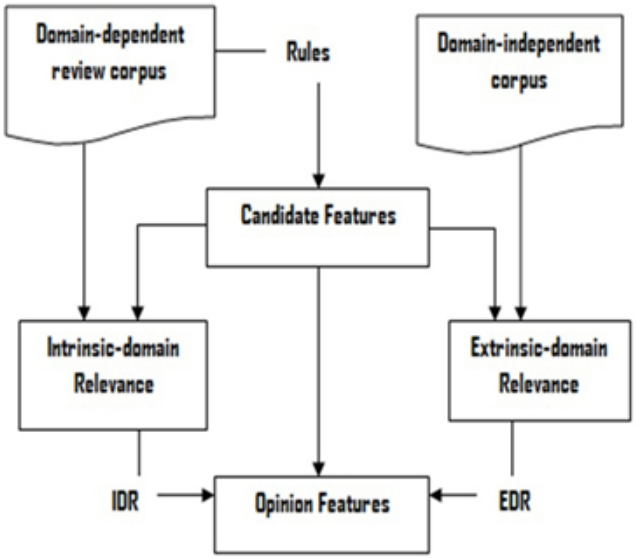


**Figure 3.** Workflow of the previous approach.

The drawback in this approach is that it extracts features based on frequency and hence rare features are missed and hence clustering is added to find Infrequent Features (IF). Clustering is based on TF (Term Frequency) the feature set is divided into 2 clusters, one with frequent features and the other with infrequent ones. Soft constraints are added to clustering in order to group similar features and also to find whether the feature is valid or not. The Threshold measure to group the features can be fixed through experimental analysis. Figure 4. Shows the Architecture of Feature extraction used in the proposed approach.

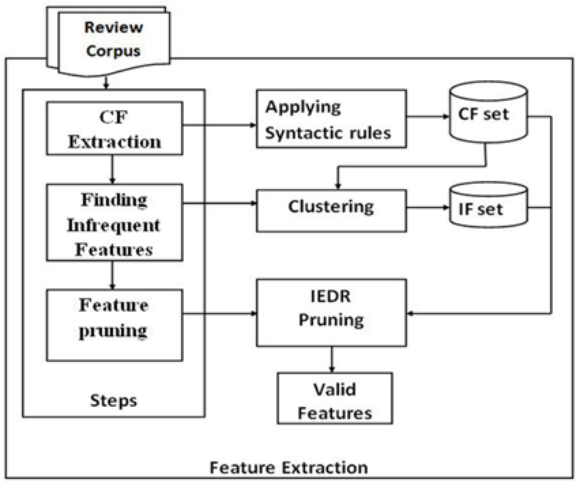


**Figure 4.** Architecture of feature extraction system.

After finding the valid features, opinion words can be found using semantic orientation which is found by using Equation (3). Semantic Orientation (SO) can be used to extract opinion polarity, which is based on Pointwise Mutual Information (PMI)[9]. PMI can be calculated using (4).

$$SO(\text{Phrase}) = PMI(\text{Phrase}, \text{“excellent”}) - PMI(\text{Phrase}, \text{“poor”}) \quad (3)$$

$$PMI(\text{word1}, \text{word2}) = \log_2 [P(\text{word1} \,\&\, \text{word2}) / P(\text{word1})\, P(\text{word2})] \quad (4)$$

Where, P (word 1 & word 2) is the probability that word1 and word2 Co-occur. If the words are statistically independent, then the probability that they Co-occur is given by P (word1) P (word 2). The ratio P (word 1 & word 2) / P (word 1) P (word 2) is the measure of the degree of statistical dependence between the words.

# 4. Implementation

## 4.1 Datasets

The data is obtained from the Stanford Large Network Data Collection, which consists review from popular e-commerce website www.amazon.com. The review spans over 18 years and contains more than a million reviews from the user. Following attributes are present in the review corpus[10]. The data sets consist of review summary, rating given by user, review usefulness, user id, product id etc.

## 4.2 Graph Model

In order for the system to be scalable and efficient we use queries which are ad-hoc in nature i.e. queries can be dynamic. The implementation of the system in orient db. uses the following commands. This is an ad-hoc query which creates a vertex of the type V and sets the name as User1. Now this query looks very weak in semantic understanding, in other terms very blurry. Now we see the other way to create vertex in the graph model.

> create vertex V set name = ‘User1’.
>create class Person extends V. (1)
>create vertex Person set name = ‘Harinath’ (2)
>create vertex Person set name = ‘Anjali’ (3)
>create vertex Product set name =’Nexus6’, type =

'Mobile' (4)

>create edge Buys from (select from Person where name = 'Harinath') to (select from Product where name ='Nexus6') (5)

>select name from (select expand (out ('Buys')) from Person where name = 'Harinath') (6)

>create edge Trust from (select from Person where name = 'Harinath') to (select from Person where name = 'Anjali ') (7)

The query in (2), (3) create a vertex of the type person and sets the name as Harinath and Anjali Respectively. This can be also modeled for the product in the system with all attributes. Query (4) Creates Vertex called product and sets name as Nexus6 and the type as Mobile. Query (5) creates an edge or connection between the user 'Harinath' and product 'Nexus6' with 'buys' as the relation. So system creates the same links to the product whenever a customer buys the product. So the number of products and customers in the database will be within the boundary.

$n \in N$ (*Total number of Customers*)

$P \in P$ *(Total number of products in the database)*

Now we can find the list of products that the customer has bought in the past using the query (6). So a graph like structure is created in the system, so that people with a similar buying pattern can be recommended the new item, based on the people's buying pattern belonging to the same group. We can serve the user a defined recommendation with the help of this. He can specify top three products in the particular group. We can drill down and drill up further based on the request provided by the user. With the support of the multidimensional query, he can find the best selling product during festival seasons like Diwali and Pongal. In case the user wants to inherit the property of a friend who is an expert in the field, he can create a link between them. So this soft link can serve as pathway to traverse and recommend items of the people who are similar (who are linked in the system). Query (7) creates a trust based recommendation. Figure 5. Shows the screenshot taken from OrientDB

## 4.3 Forms of Recommendation

Recommendation engine offers the recommendation through services such as HTTP notification, SMS notification and Email notification. Relevance in the e-mail notification using the web bugs which are attached to the mail. Recommendation is given in form of Top K product from the item catalog. Two types of selling, Cross selling and Top Selling of the product. Cross selling is providing complementary items of the selected product, while choosing the cross selling two constraints are imposed on the system i.e. the items which provide more profit to the company and the items which is of good quality. Top selling is another business model, where suggesting high cost product to the user than the specified search amount.

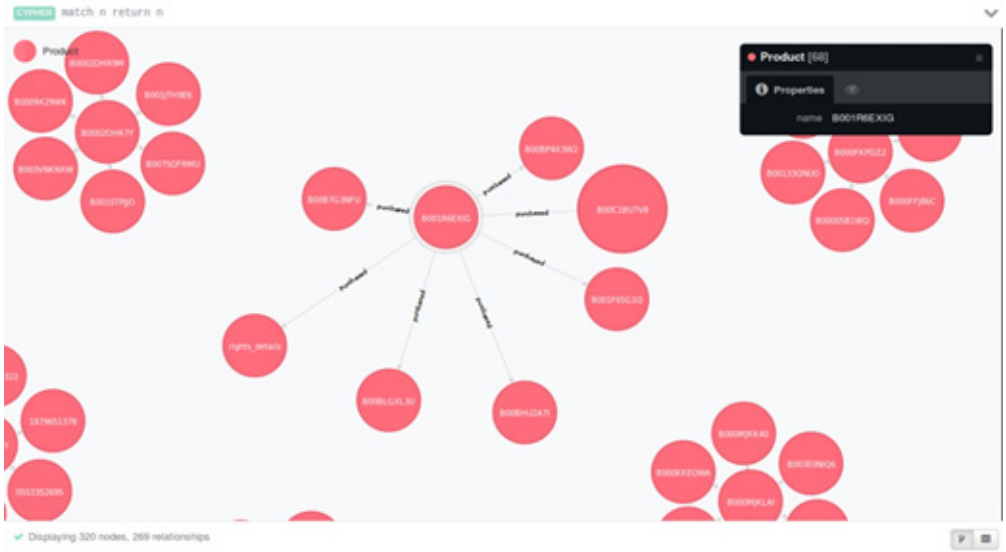


**Figure 5.** Screenshot from OrientDB.

### 4.4 Evaluation Metrics

After building the recommendation engine, an important task is to analyze how well the recommendation is provided to the user, it is done mathematically. Usual metrics for evaluating a recommendation engine is Root Mean Squared Error. A/B site testing is used, which focuses on the user interaction with the website and understanding the crucial components in the website with a parameter such as Click Through Rate (CTR).

### 4.5 Hardware Configuration

To store all the user data, product details and user feedback about the product such as rating and review are stored in NoSql OrientDB. OrientDB offers both document based structure as well as graph based structure. It contains a SQL layer through which user can take advantage and power Structured Query Language. Recommendation Engine runs in hadoop cluster with replication factor 3, to offer more request at the same time. The proposed approach uses Natural Language Toolkit. Hadoop version 1.2.1 with multi node configuration, Orient DB2.0, System memory is 150 GB, Hard Disk 2 TB.

### 4.6 Review Consideration

The sentiment analysis or opinion mining is done by the following ways. There are several types of users in the system who give reviews

- Actual user
- Person who did not buy the product
- Expert user ,who analyses a lot of details about the product

Actual user is the person who has bought the particular item from the site. This detail can be obtained from the past transactions of the website. They know how the product really works and their review must be given more weight age than the others. A person who did not buy the product doesn't have enough knowledge about the product, he/she might have seen their friends or relatives using the product. So their review may be biased. An expert, will see the finest of details. They would worry about the performance benchmark and other details. If my mom wishes to buy the product she may not require expert review. She may just need the normal user review about the product.

We can consider the reviews of the

- Expert user and actual user,
- Actual user and Person who did not buy,
- Expert and person who did not buy.

This approach gives more confidence to buy the product. Review weight age can be increased with a simple button like how many people found the review to be useful. If more people found a particular review to be useful, the weight age of the user's review can be increased. Review weight age is considered because people who use android give negative review about the iPhone and vice versa. Other small reviews which spans to two to three words can be neglected.

### 4.7 Overspecialization Problem

This problem occurs when the person who likes the particular thing for e.g. south Indian cuisine is shown only the south Indian hotels, the best Chinese hotel in the city is not recommended. This is because all the recommendation is tailored for them. So by removing the option called tailoring, cross recommendation can be provided, which gives a person much more choices and a different experience.

## 5. Conclusion

The proposed approach addressed the challenges to cold start problem and made the system scalable through hadoop framework. Accuracy rate is the highest in the clothing and cell phone accessories system while the other domains will feature lesser accuracy as of now but will be made higher in the near future

## 6. References

1. Data Mining: Concepts, Techniques T. 2nd ed. Gray J, editor. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers; 2006 Mar. ISBN: 1-55860-901-68].
2. Bakhtawar S and Farouque A. Opinion mining: Issues and challenges (a survey). International Journal of Computer Applications. 2012; 49:42-51.
3. Wei J, Hung HH, and Rohini K. Srihari. A novel lexicalized HMM-based learning framework for web opinion mining. Proceedings of the 26th Annual International Conference on Machine Learning; 2009.
4. Li, Fangtao et al. Structure-aware review mining and summarization. Proceedings of the 23rd international conference on computational linguistics. Association for Computational Linguistics; 2010.

5. Su, Qi et al. Hidden sentiment association in Chinese web opinion mining. Proceedings of the 17th international conference on World Wide Web. ACM; 2008.
6. Pasquale L, De Gemmis M, Semeraro G. Content-based recommender systems: State of the art and trends. Recommender systems handbook. US: Springer; 2011. p. 73–105.
7. Degemmis M, Pasquale L, and Pierpaolo B. An intelligent personalized service for conference participants. Foundations of Intelligent Systems. Berlin Heidelberg: Springer; 2006. p. 707–12.
8. Hai, Zhen et al. Identifying features in opinion mining via intrinsic and extrinsic domain relevance. Knowledge and-Data Engineering. IEEE Transactions. 2014; 26.3:623–34.
9. Turney PD. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics; 2002.
10. Fd e Leskovec, Jure, Krevl A. {SNAP Datasets}:{Stanford}. Large Network Dataset Collection; 2014.