Recommendation System Based on Content Filtering for Specific Commodity

Niranjan C Kundur[1], Praveen M Dhulavvagol[2], Prasad M R[3]

[1]*Dept. of CSE, JSSATE, VTU, Belagavi.*

[2]*Dept. of ISE, BVBCET,*

[3]*Dept. of CSE, JSSATE, VTU, Belagavi.*

***Abstract*— Internet Content-based recommendation systems may be used in a variety of domains ranging from recommending web sites, news items, restaurants, television programs, and commodities for sale. Content-based recommendation systems share a common means for describing the items that may be recommended. In this paper, we propose Recommendation System that uses Keywords as input query from user for extracting specific items that match user query from the list. User keywords may consists of keywords words from name of the item, brand and popularity. Here we are calculating the similarity between user given item names and collected item name in the database by using vector space model which in turn uses TF-IDF, Cosine Similarity and finally re-rank top recommended items. We measured satisfaction and accuracy for each system-recommended item to test and evaluated performances of the suggested system. Finally Recommendation System for item based represents high level of satisfaction and accuracy.**

***Keywords*: item based, recommendation, vector space model, hash-map**

I. INTRODUCTION

Recommender systems have become extremely common in recent years, and are applied in a variety of applications. The most popular ones are probably movies, music, news, books, research articles, search queries, social tags, and products in general. However, there are also recommender systems for experts, jokes, restaurants, financial services, life insurance, persons (online dating), and Twitter followers.

Recommender systems are software tools and techniques providing suggestions for items to be of use to a user. The term item here is generic. It may represent many concepts. For instance recommender systems may recommend news on a news portal, or products in an online shop, or even services. The recommendations are usually tailored to a given type of user or a given type of user group. Since recommendations are personalized, they may vary from one user to another or from one user group to another. Due to the development of technology of Internet, Web Programming and Web environment in recent years, the huge amount of data extremely increases in the Web[6], then following new exceed information overload problems occur. So, newly high technology search engines are developed and made to solve these problems and to provide user-wanted information quickly and accurately. Content-based recommendation systems analyse item descriptions to identify items that are of particular interest to the user. Recommender system is an active research area in the data mining and machine learning areas. [1]

II. RELATED WORKS

Collaborative Filtering Technique:

Collaborative filtering systems work by collecting user feedback in the form of ratings for items in a given domain and exploit similarities and differences among profiles of several users in determining how to recommend an item. The limitations of using collaborative filtering are

i. Most users do not rate most items and hence the user-item rating matrix is typically very sparse. Therefore the probability of finding a set of users with significantly similar ratings is usually low[2]. This is often the case when systems have a very high item-to-user ratio. This problem is also very significant when the system is in the initial stage of use[4].

ii. First-rater Problem*:* An item cannot be recommended unless a user has rated it before. This problem applies to new items and also obscure items and is particularly detrimental to users with eclectic tastes.

Click-Through Rate:

It is used in recommendation of papers, it's mainly online. The click-through rate is the number of times a click is made on the advertisement divided by the total impressions (the number of times an advertisement was served). The limitations of click-through rate are:

i. It does not help you with conversions. A high CTR might actually have a low conversion rate (and often does). Some Internet users just have a higher propensity to click, which does not actually mean that they want to buy anything. Usually, these people can be found in higher proportion at less popular sites (which is probably how they got there in the first place)[5].

ii. What about the people who don't click? High CTR's are rarely much bigger than a few percent. What about the other 90?

iii. It doesn't tell you about coverage. You want to reach as many people as possible and the best spaces usually can't be bought with PPC.

Challenges:

Finding research papers on different web site can be a difficult and time-consuming process. Recommender systems can help users to find relevant papers by providing them with personalized suggestions based on user interest (domain area). If there is a provision to find papers based on the content that would rather be leisure for a user.

III. PROPOSED METHOD

The following objectives are defined for the recommendation system:

- To recommend research paper based on content filtering.
- To store user published papers with keywords from title, author name, year of publication, keywords and abstract.
- To extract keyword from user query by removing stop words.
- To calculate similarity between user query and document using vector space model.
- To display Top -N ranked results.

The below figure 1 shows the block diagram of the system. A system comprises multiple views such as Query Processing, which involves stop word removal, stemming and building the dictionary. In second phase similarity computation is carried out using vector space model. A system model is required to describe and represent all these multiple views.

approach.

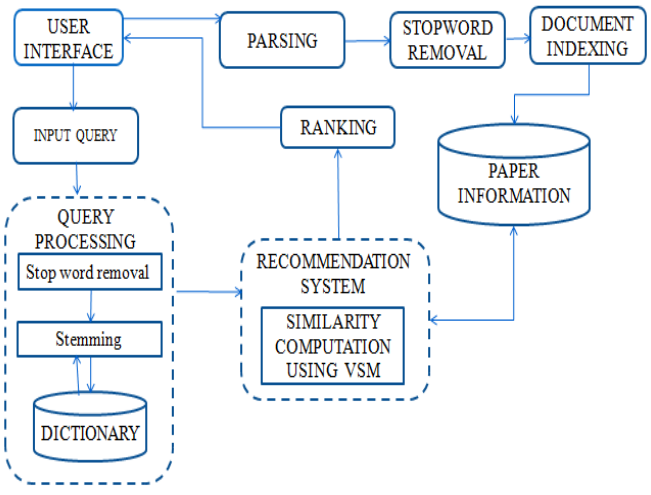


Figure 1: Block diagram of recommendation system.

The above block diagram consists of different components or modules:

1. Document Processing:

In this module user who has published standard research paper will add his papers to the database. Initially papers will be in pdf file format. Once those papers are selected internally those pdf files will be converted into text file format and will be given unique name and then stored in a directory. Once files are converted and stored they will be passed to a function where in each files title, author name, year of publication, abstract and keywords mentioned in paper are extracted. Then these file containing Metadata of papers will be passed to a function where in most repeating words in a file will be removed (stop words). There is a list of stop words that has been declared by Word Net dictionary we are incorporating the same. The input for this module is standard IEEE format papers and the output generated will be the parsed documents with keywords, title, abstract, author name in text files[9].

2. Query Processing:

In this module through user interface (GUI), user needs to give content description of paper that he is looking for. Description must be keywords from title, author name, year of publication, abstract and keywords mentioned in paper. Frequently repeating words i.e. stop words will be removed from user query, and query will be expanded using synonyms, hyponyms stored in a dictionary, so as to give user a broader option of interest. The input for this module is the given user query i.e. the key words and the output generated will be expanded user query with hyponyms and synonyms of keywords. Query expansion is preferred so that we get Recall rate high and more number of relevant documents are retrieved.

3. Similarity Computation:

Similarity computation is done using vector space model.

Step1: Term Frequency (Tf): Initially we will calculate the frequency of the terms which are there in the document, here frequency means the number of times a particular term appearing in that document, and we keep the count of those terms.

Step2: Inverse Document Frequency (Idf): In this step we try to avoid frequently appearing terms because rare terms are more informative than frequently appearing terms so that more number of relevant documents is retrieved. To calculate the IDF we use (log10 N/dft). N defines the number of documents in the collections and dft defines the number of documents in the collection that contain a term t.

Step 3: Calculation of weight: This is done by considering the dot product of term frequency and inverse document frequency wt= tf*Idf. This weight wt is used to rank the document based on the relevance of the user query. The input

for this module is the expanded user query with hyponyms and synonyms of keywords and the output generated is the ranked documents[8].

4. Document Indexing:

It is obvious that many of the words in a document do not describe the content, words like the, is, and, all etc such words are called as stop words. By using automatic document indexing those stop words are removed from the document vector, so the document will contain only those words which has some relevant meaning we call them as keywords. This indexing can be based on term frequency, where terms that have both high and low frequency within a document are considered to be function words. There are many indexing techniques available such as inverted index, zone based indexing and so on, in inverted index technique there are mainly 4 steps: Tokenization, Sorting, Merging and Calculation of term frequency. In tokenization each tokens are generated from the given document, Here tokens are nothing but key words, in second step these tokens are sorted in alphabetical order and in next step the tokens which are similar are merged, Grouped together along with the document id, and finally term frequency is calculated for these tokens and linked list data structure is used to link these tokens to the appropriate documents along with its frequency scores.

5. Term Weighting:

Term weighting has been explained by controlling the exhaustively and specificity of the search, where the exhaustively is related to recall and specificity to precision. The term weighting for the vector space model has entirely been based on single term statistics. There are three main factors for term weighting: term frequency factor, collection frequency factor and length normalization factor. These three factor are multiplied together to make the resulting term weight. A common weighting scheme for terms within a document is to use the frequency of occurrence. The inverse document frequency, assume that the importance of a term is proportional with the number of document the term appears in. Experimentally it has been shown that these document discrimination factors lead to a more effective retrieval, i.e., an improvement in precision and recall. The third possible weighting factor is a document length normalization factor. Long documents have usually a much larger term set than short documents, which makes long documents more likely to be retrieved than short documents[10].

6. Calculating Similarity Coefficients:

The similarity in vector space models is determined by using associative coefficients based on the inner product of the document vector and query vector, where word overlap indicates similarity[11]. The inner product is usually normalized. The most popular similarity measure is the cosine coefficient, which measures the angle between the document vector and the query vector.

IV. EXPERIMENTAL RESULTS

The evaluation of the proposed recommendation system is done by considering the following performance evaluation parameters: precision, recall and F score.

Precision is the fraction of retrieved documents that are relevant to the query

Precision P = (Number of relevant documents retrieved) / (total number of retrieved documents)

Precision considers all the retrieved documents into account, but it can also be evaluated at a given cut-off rank, considering only the topmost results returned by the system.

Recall in information retrieval is the fraction of the documents that are relevant to the query that are successfully retrieved.

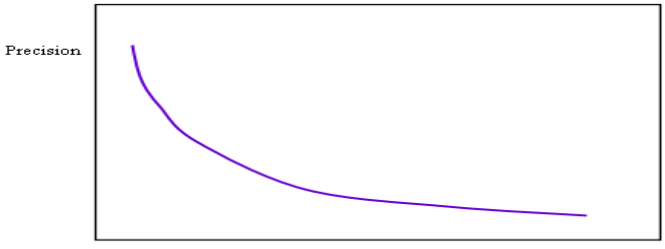
Recall R = (Number of relevant documents retrieved) / (total number of relevant documents)

For example for text search on a set of documents recall is the number of correct results divided by the number of results that should have been returned. In binary classification, recall is called sensitivity. So it can be looked at as the probability that a relevant document is retrieved by the query. It is trivial to achieve recall of 100 by returning all documents in response to any query. Therefore, recall alone is not enough but one needs to measure the number of non-relevant documents also, for example by computing the precision.

A measure that combines both precision and recall is the harmonic mean of precision and recall, the traditional F-measure or balanced F-score:

$$F= (2PR) / (P+R)$$

Precision and Recall Relationship: they are inversely proportional to each other, i.e as precision increases recall decreases this can be show using confusion matrix



	Relevant	Nonrelevant
Retrieved	true positives (tp)	false positives (fp)
Not Retrieved	false negatives (fn)	true negatives (tn)

Precision P = tp / (tp + fp)
Recall R = tp / (tp + fn)

V. RESULTS AND DISCUSSIONS

Initially for analyzing the performance of the recommendation system we considered a sample of 24 papers in the database. For this we calculated precision, recall and F score as follows:

	Relevant	Non relevant	Total
Retrieved	17	1	18
Not Retrieved	6	0	6
Total	23	1	24

Precision P = 17/23 = 0.7391
Recall = 17/18 = 0.9444

F-Measure F = 2PR/P+R
= 2*(0.7391*0.9444)/(0.7391+0.9444)
= 0.82981

For a sample of 24 documents we can observe that the harmonic mean value is 0.829

The size of the database is increased by adding more number of papers or documents and performance evaluation was done. Total papers considered were 40. For this precision, recall and F score values are:

	Relevant	Non relevant	Total
Retrieved	26	0	26
Not Retrieved	14	0	14
Total	40	0	40

Precision P = 26/40 = 0.65
Recall = 26/26 = 1

F-Measure F = 2PR/P+R
= 2*(0.65*1)/(0.65+1)
= 0.7787

For a sample of 40 documents/papers we got the harmonic mean value is 0.7787

Finally system performance was evaluated with 70 papers in database. In this case precision, recall, F score values are:

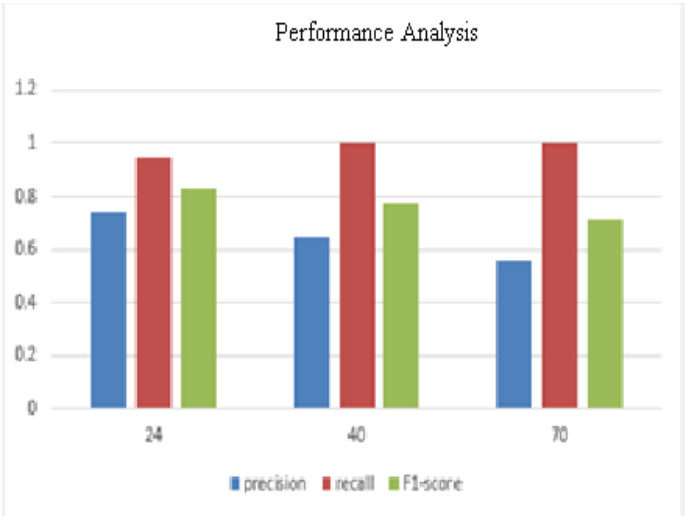
	Relevant	Non relevant	Total
Retrieved	39	0	39
Not Retrieved	31	0	31
Total	70	0	70

Precision P = 39/70 = 0.5571
Recall = 39/39 = 1

F-Measure F = 2PR/P+R
= 2*(0.5571*1)/(0.5571+1)
= 0.7156

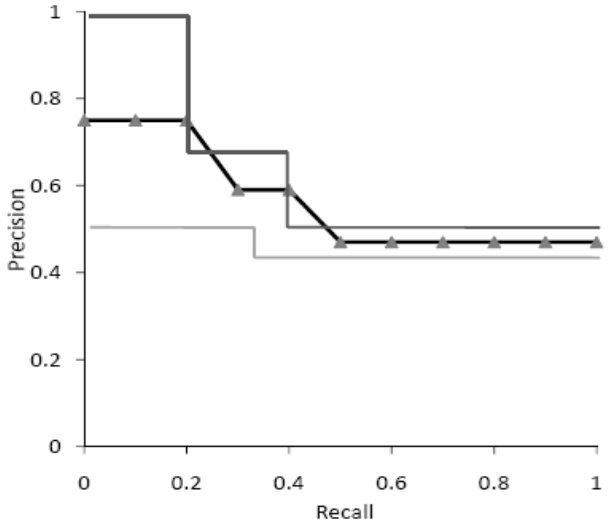
For a sample of 70 documents/papers we got the harmonic mean value is 0.7156

By comparing all the above cases with 24, 40, 70 papers we get performance graph for a given query:



In above test cases we can observe that as the number of papers in the database increases the searching time also increases. The precision and recall value also vary as size of database increases. Here we can observe through the graph, that precision and recall are inversely proportional, as the value of recall increases the value of precision decreases and visa verse. And the harmonic mean of both precision and recall gives us accuracy which is an average 77%.

Average Precision Recall graph for queries:



VI. CONCLUSION AND FUTURE WORK

When academician and researchers publish their research paper, they had to spend a lot of time and efforts to retrieve the relevant paper matching the given input query. Therefore, active research paper search and re-search are needed related to specific topic. Then experiments verify that Research Paper Recommendation System has a high level of satisfaction and accuracy. Future work will be implemented about grouping of research papers related to specific subject and active recognition of research trends continuously.

REFERENCES

[1]. Antal van den Bosch Toine Bogers, Maxim Gurevich. "Collaborative and Content based Filtering for Item Recommendation on Social Bookmarking Websites". ILK / Tilburg centre for Creative Computing, Tilburg University, pages 1–8, 26th August 2013.

[2]. From Lucene Apache. Lucene search engine. http://lucene.apache.org.

[3]. Chenguang Pan, Wenxin Li, "Research Paper Recommendation with Topic Analysis", 2010 International Conference On Computer Design And Applications (ICCDA 2010), vol.4, pp.264-268, 2010.

[4]. G. Shani and A. Gunawardana, "Evaluating recommendation systems," Recommender systems handbook, Springer, 2011, pp. 257–297

[5]. N.F. Matsatsinis, K. Lakiotaki, and P. Delia, "A system based on multiple criteria analysis for scientific paper recommendation," Proceedings of the 11th Panhellenic Conference on Informatics, 2007, pp. 135–149

[6]. T. Strohman, W.B. Croft, and D. Jensen, "Recommending citations for academic papers," Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2007, pp. 705 706.

[7]. Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, pages 94–105. ACM Press, 1998.

[8]. G. Adomavicius, and A. Tuzhilin. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions." Knowledge and Data Engineering, IEEE Transactions on 17, no. 6 (2005): 734-749.

[9]. M.J. Pazzani and D. Billsus, "Content-based recommendation systems", in: P. Brusilovsky, A. Kobsa, W. Nejdl (Eds.), The Adaptive Web, Lecture Notes in Computer Science, vol. 4321, Springer-Verlag, 2007, pp. 325–341.

[10]. Belkin, N., Croft, B.: Information Filtering and Information Retrieval: Two Sides of the Same Coin? Communications of the ACM 35(12) (1992) 29-38

[11]. Basu, C., Hirsh, H., Cohen W.: Recommendation as Classification: Using Social and Content-Based Information in Recommendation. In: Proceedings of the 15th National Conference on Artificial Intelligence, Madison, WI (1998) 714-720

[12]. Balabanovic, M., Shoham Y.: FAB: Content-based, Collaborative Recommendation. Communications of the Association for Computing Machinery 40(3) (1997) 66-72