REVIEW ON RESEARCH PAPER RECOMMENDATION USING CLUSTERING

[1]Pranali Manekar, [2]Anjali Raut
[1]Student, [2]professor
[1]Computer Science And Engineering,
[1]H.V.P.Mandal's College Of Engineering And Technology,
Sant Gadge Baba Amravati University, Amravati, India

Abstract : Research papers are increasing rapidly as the number of researchers' increases in academics field. As the research papers are present in huge amount, it is very difficult to search appropriate research paper according to search query. To solve this issue, the propose work includes citation search engine and recommendation system in which any student/researcher can search research paper and get recommended reference papers along with matching papers. To improve performance of the searching the propose work includes k-means clustering algorithm to form documents clusters. It also considers reference paper recommendation by using user defined algorithm. Along with papers searching and reference papers recommendation, the paper will be recommended in the model in which we will consider collaborative, profile wise and preferences wise citation recommendation techniques will be consider.

IndexTerms **– Clustering, citation recommendation, Collaborative Recommendation, Tokenization, K-means clustering, Data Mining**

I. INTRODUCTION

Ranking scientific publications is an important task, which helps researchers to find related works of high quality. However, the dynamic nature of the evolving publication network makes the task very challenging. For newly published articles, little citations can be found so that they are hard to be recommended by citation-based systems. The model reduces the bias of time to some extent, because recent articles will be promoted to higher scores. With the rapid growth of scientific literature, it is impossible for researchers to go through and digest all the available literature. Traditional approaches perform keyword-based searches to retrieve a list of relevant papers, and require the researchers to manually review them and thus select appropriate papers as reference papers.

II. LITERATURE REVIEW

Literature review is the study of previous work of existing system. Here it will study the previous work of Research paper recommender system.

Pan et al. [1] suggest an academic paper recommendation approach based on heterogeneous graph containing various kinds of features. Jiang et al. [2] suggest a chronological citation recommendation approach that considers chronological nature.

Wang et al. [3] expand a novel entity class dependent discriminative mixture model for cumulative citation recommendation, avoiding the insufficient training data of less popular entities in a chronological stream corpus.

Chakraborty et al. [4] Introduce diversified citation recommendation framework that balanced the prestige, relevance and diversity of reference papers. M. McNee *et al* [5] proposed a system in which a citation represents a research paper for which we only have a reference (i.e., a paper has cited this citation, but we do not possess the paper which corresponds to the citation). A paper is a citation for which we have access to the full text, including the paper's citation list. Thus, for a paper we have a listing of all the citations that it references, some of which *may* also be papers in our dataset but all of which *must* be citations in our dataset. It should be noted that a paper can exist without a citation.

Z. Kang, C. Peng, and Q [6] and Q. Cheng introduced that an alternative approach is to make paper authors the 'users' and keep citations as 'items.' In this ratings matrix, each author would "vote" for the papers that she has cited. By using the citation web to populate the matrix, this mapping does not suffer from the startup problem.

III. PROPOSED WORK

The proposed work includes citation recommendation system in which any researcher can search research paper and get recommended reference papers along with matching papers. Here first user uploads the research papers with details. And then when another user or admin search the papers then it gives the list of related papers to it. Here in this project it also gives the recommendation of reference papers which is related to the researcher's data. But for that first researchers have to provide the area of their interest. So according to that, the system will recommend the related reference papers list to the researchers.

3.1 System includes two users are as follows:

3.1.1 Admin:- Admin will view researchers detail and search papers

3.1.2 User:- User will upload research paper with details and search research paper

3.2 The proposed system contain the modules which are as follows

3.2.1 Admin panel: - Admin is the very first module in the system. It has all the authority of project. It manages users, project data, modules and maintains the database of the system. In this project, Admin first login into the system. Then as our work is on research paper recommendation system it allows the user for registration and research paper. It tracks all the user's activity and search the appropriate paper. It has authority to view the researcher's details and maintain their data. In this way admin have the most important responsibility of maintain all the users and their work.

3.2.2 User management:- It is the another module in the system. Firstly user has to register into the system. After the registration user can login into the system. For login user have to enter the username and password. If user can forgot password then he can either change the password or can recover the password. After successful login user has main work is upload the research papers. At the time of uploading research papers user also have to add all the details of those papers. After the uploading papers user also has authority to search the various research papers. In this system, the users get the recommendation of reference papers related to their search. As the user can search the paper on a particular topic for example data mining, so after searching, user can get the recommendation of the reference or related papers of data mining. In this way the user can manage the uploading and searching of papers.

3.2.3 Document searching:- Document is easy and fast for searching. Here the effective way for document searching is query based searching. To search any document user have to specify search query. Here the search engine is based on query searching. Where user can specify any search query to satisfy their need. After uploading the document user extract text and remove the special symbol from it after that by using tokenization technique it add some spaces in words and create tokens. After that with the help of stop word removal it removes meaningless words from the text and then keyword extraction is used to extract meaningful keywords from it. After that we find the synonym of the given keyword and search that keyword in database so that we recommend the related paper to it.

3.2.4 Document clustering:- Document clustering is the organization of a large amount of text documents into small number of meaningful clusters, where each cluster represent a specific topic. In clustering When user specify any search query, system will modify keywords set which is extracted from specified query using their synonyms and with the help of modified keyword set, system will fetch matching research papers. After fetching the result set system will perform K means clustering algorithm to form clusters of searched documents and show documents clusters to user

3.2.5 Reference Document Recommendation: - when user starts to search a paper and when select any paper for reading then system will automatically recommend the related reference paper to the user. Related paper recommendation is based on some criteria. For example if user consider a base paper of any topic then the related paper recommendation is depend on the author set, venue set and query text. But practically it is impossible that the reference papers are always written by the same author. So to improve the quality of reference paper recommendation here user proposed the TF-IDF technique which is used to calculate the weight of every keyword extracted from the document.

3.3 Working Diagram:-

In this system there are two users' admin and user. To access the system researcher need to login into the system. For login first researcher has to register. After completing registration researcher can login. After login researchers can either search the paper or can upload the paper. First consider that user is uploading the paper. In this first user upload the papers and then extract key phrases from the paper and then calculate the TF-IDF score for every key phrase. Then it store key phrases with score in database. Another function of user is to search the papers. For searching the paper researchers first specify the query. After specifying the query, the keyword will extracted from the query and then find the synonyms. After finding the synonyms the system will find out the matching papers related to it. The k means clustering is used to show the cluster of papers. Along with this system will recommend the reference papers related to papers present in result set to the users.

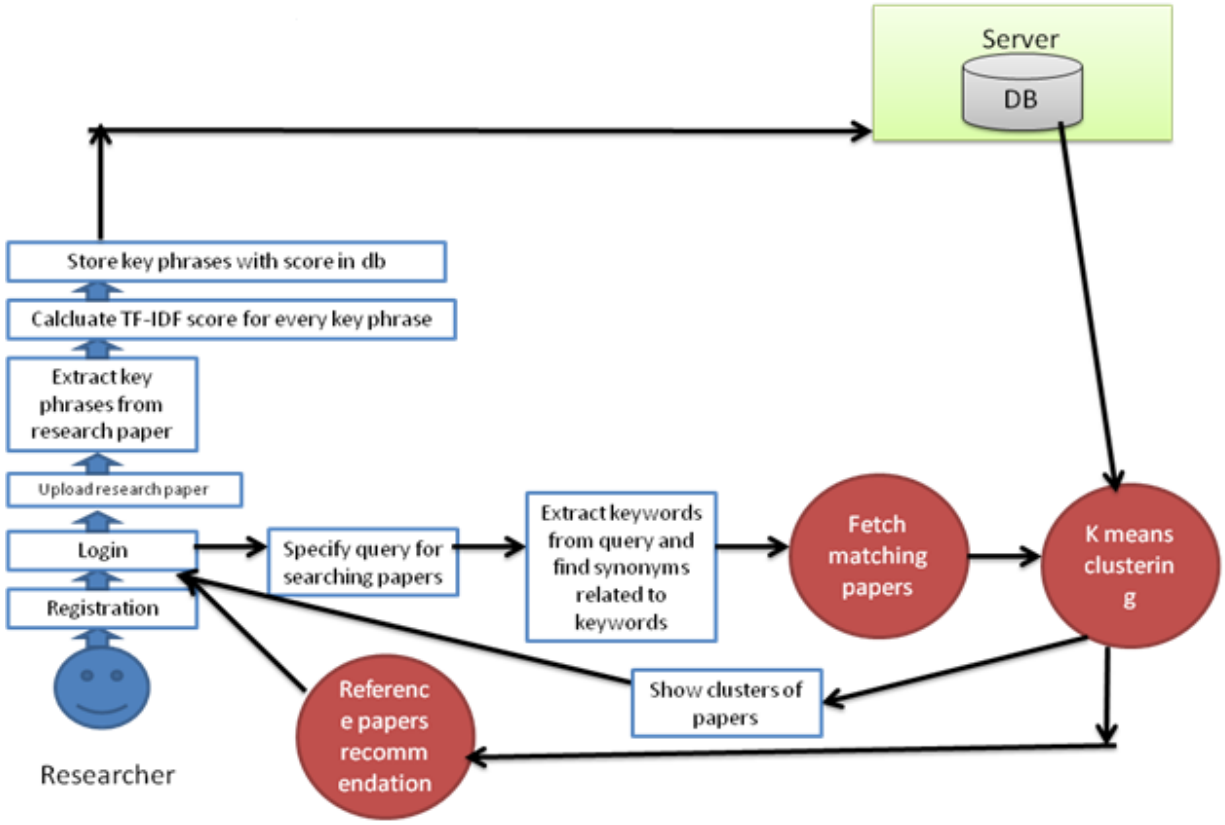

Fig 1. System Flow Diagram

3.4 Algorithm:-

3.4.1 K means algorithm

K means is one of the simplest learning algorithms which solve the clustering problem. It will classify the data sets into number of clusters (assume k clusters). Here k centers are defining one for each cluster. These centers need to place in a tricky way because different locations cause different different results. So best choice is to place them is far away from each other. Then take each point from the data sets and connect it to nearest center. When there is no point is remaining then first step is completed and an early group age is done.

Here k new centroid needs to recalculate as barycenter of the clusters resulting from the previous step. When k new centroid is found, a new binding has to be done between the same data set points and the nearest new center. A loop has been generated. At the end of this loop, some observation are found which are k centers change their location step by step until no more changes are done in other words centers do not move any more. After we have these k new centroid, a new binding has to be done between the same data set points and the nearest new center. A loop has been generated. As a result of this loop we may notice that the k centers change their location step by step until no more changes are done or in other words centers do not move any more. Finally, this algorithm aims at minimizing an objective function knows as squared error function.

3.4.1.1 Basic Steps of K-means Algorithm:-

Step1: Choose k number of clusters to be fixed

Step2: Choose k objects randomly as the initial cluster center

Step3: Repeat

Step4: Assign each object to their closest cluster

Compute new clusters, i.e. Calculate mean points.

Step5: Until

No changes on cluster centers (i.e. Centroid do not change location any more) OR
No object changes its cluster (We may define stopping criteria as well)

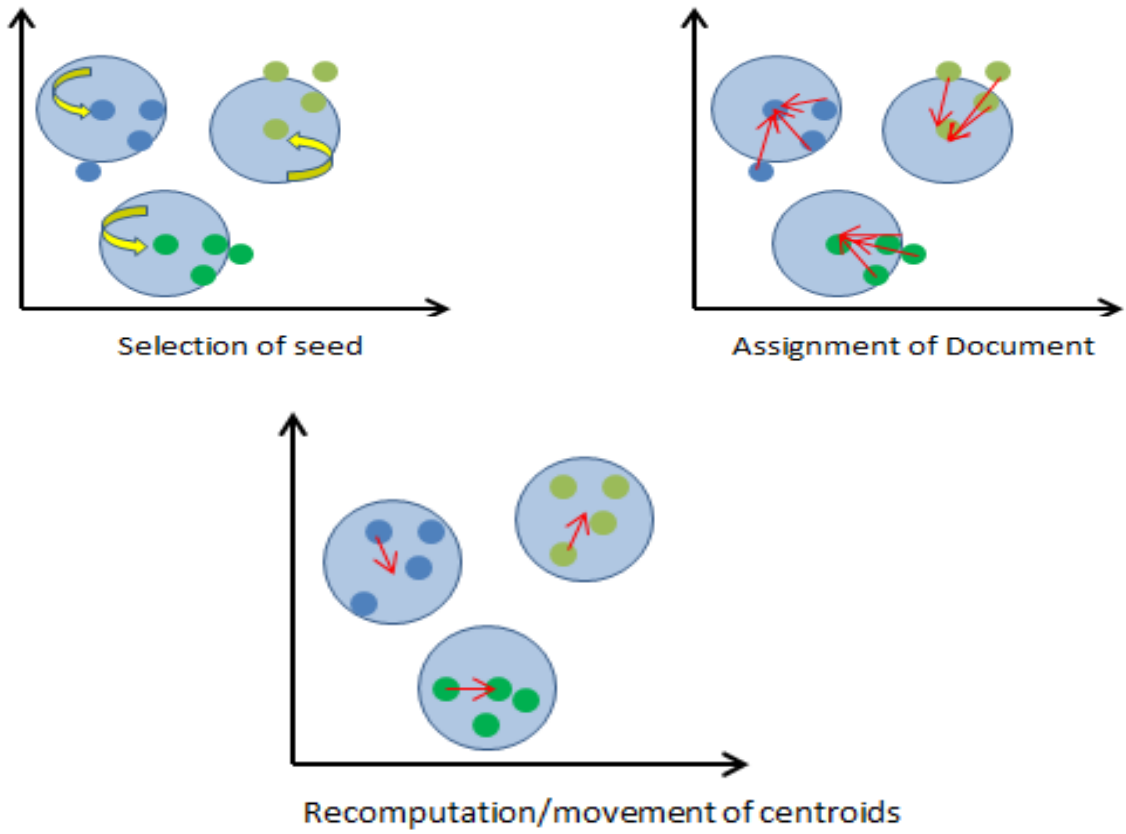

Fig. 2 Centroid K- Means clustering

3.4.1.2 Working of algorithm:- To implement K-Means algorithm I have defined a class Centroid in which documents are assigned during the clustering process.

Step1: Initializing cluster canter

Cluster canter is initialized for the next iteration, here the count variable holds the value of user defined initial cluster canter.

Step2: Finding closest cluster center

This function returns the index of closest cluster center for each document; I have used cosine similarity to identify the closeness of document. The array similarity Measure holds the similarity score for the document object

with each cluster center, the index which has maximum score is taken as the closest cluster center of the given document.

Step3: Identifying the new position of the cluster center

After each document being assigned to its closest cluster center we recalculate the mean of each cluster center which indicates the new position of cluster center (centroid).

3.4.2 Residual sum of squares:- RSS is a measure of how well the centroid represent the members of their clusters, the squared distance of each vector from its centroid summed over all vectors. The algorithm then moves the cluster centers around in space in order to minimize RSS.

$$\mathrm{RSS}_k = \sum_{\vec{x} \in \omega_k} |\vec{x} - \vec{\mu}(\omega_k)|^2$$

$$\mathrm{RSS} = \sum_{k=1}^{K} \mathrm{RSS}_k$$

Where

ω_k Document cluster k

$\vec{\mu}$ Mean or centroid of the documents in cluster ω_k

$\vec{x}$ Document vector in cluster k

3.4.3 TF-IDF Technique:-

The TF-IDF weighting, stands for term frequency (tf) × inverse document frequency (idf). TF-IDF weighting is commonly used in text mining and information retrieval to evaluate the important term in a studied corpus. Term importance (weight) increases with the term's frequency in the text. Given a collection of terms t ∈ T that appear in a set of N documents d ∈ D, each of length n_d, TF-IDF weighting is computed as follows:

$$tf_{t,d} = = f_{t,d} / n_d$$
$$idf_t = \log(N/df_t)$$
$$W_{t,d} = tf_{t,d} \times idf_t,$$

Where,

$f_{t,d}$: The frequency of term t in document d,

df_t: The document frequency of term t, that is, the number of documents in which term t appears.

For the task of comment classification, 'document' is replaced by 'class', e.g. sentiment class which is further divided as positive and negative in sentiment analysis. Term frequency (tf) is then computed per class. Inverse to document (idf) becomes "inverse to comments", means that N is size of set of comments, and the document frequency for a term (dft) is computed on that set. Birmingham and Smeaton define this method as sentiment tf-idf. Comments are then classified into classes using probabilistic (e.g., Naive Bayes) or discriminative models (e.g., SVMs). A common variant of the classic tf-idf is delta idf weighting, in which idf is calculated for each class separately, and then the difference between the values is used for sentiment classification. This variant is proved to be efficient for classification at the sentence level.

3.4.4 Collaborative Recommendation:- Collaborative filtering (**CF**) is a technique used by recommender systems. Collaborative filtering has two senses, a narrow one and a more general one. In the narrower sense, collaborative filtering is a method of making automatic guess (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating). The underlying supposition of the collaborative filtering approach is that if a person *A* has the same opinion as a person *B* on an issue, A is more likely to have B's opinion on a different issue than that of a randomly chosen person. These predictions are particular to the user, but use information obtain from many users. This varies from the simpler approach of giving an average (non-specific) score for each item of interest, for example based on its number of votes.

IV. Conclusion

The recommender system has a great future in recommending research paper. It helps the use to find the relevant paper related to their information. In this paper, the system will recommend the reference paper based on the users searching query. To get the more paper which is connected to their searching data this system will help. Therefore user can quickly access the paper.

Reference

[1]L. Pan, X. Dai, S. Huang, and J. Chen, "Academic paper recommendation based on heterogeneous graph," in Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data. Berlin, Germany: Springer, 2015, pp. 381–392.

[2] Z. Jiang, X. Liu, and L. Gao, "Chronological citation recommendation with information-need shifting," in Proc. 25th ACM Conf. Inf. Knowl. Manag., 2016, pp. 1291–1300.

[3] J. Wang, D. Song, Q. Wang, Z. Zhang, L. Si, and L. Liao, "An entity class-dependent discriminative mixture model for cumulative citation recommendation," in Proc. 38th SIGIR Conf., 2015, pp. 635–644.

[4] T. Chakraborty, N. Modani, R. Narayanam, and S. Nagar, "DiSCern: A diversified citation recommendation system for scientific queries," in Proc. 31st ICDE Conf., 2015, pp. 555–566.

[5] S. M. McNee *et al.*, "On the recommending of citations for research papers," in *Proc. ACM Conf. Comput.*, 2002, pp. 116–125.

[6] Z. Kang, C. Peng, and Q. Cheng, "Top-N recommender system via matrix completion," in *Proc. 30th AAAI Conf.*, Phoenix, AZ, USA, 2016, pp. 179–185.

[7]A. Hernando, J. Bobadilla, and F. Ortega, "A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model," Knowl. Based Syst., vol. 97, pp. 188–202, Apr. 2016.

[8] S. M. McNeeet al., "On the recommending of citations for research papers," in Proc. ACM Conf. Comput., 2002, pp. 116–125. [5] C. Yang, B. Wei, J. Wu, Y. Zhang, and L. Zhang, "CARES: A rankingoriented CADAL recommender system," in Proc. 9th Joint Conf. Dig. Libraries, Austin, TX, USA, 2009, pp. 203–212.

[9] Z. Kang, C. Peng, and Q. Cheng, "Top-N recommender system via matrix completion," in Proc. 30th AAAI Conf., Phoenix, AZ, USA, 2016, pp. 179–185.

[10] K. Chandrasekaran, S. Gauch, P. Lakkaraju, and H. Luong, "Conceptbased document recommendations forCiteSeer authors," in Proc. Int. Conf. Adap Hypermedia Adap Web Syst., Hannover, Germany, 2008, pp. 83–92.

[11] R. Torres, S. M. McNee, M. Abel, J. A. Konstan, and J. Riedl, "Enhancing digital libraries with TechLens+," in Proc. 4th Joint Conf. Dig. Libraries, Tucson, AZ, USA, 2004, pp. 228–236.

[12] M. Balabanovi´c and Y. Shoham, "Fab: Content-based, collaborative recommendation," Commun. ACM, vol. 40, no. 3, pp. 66–72, 1997.

[13] P. Lops, M. Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in Recommender Systems Handbook. Berlin, Germany: Springer, 2011.