Similarity Measures of Research Papers and Patents using Adaptive and Parameter Free Threshold

Gourav Bathla
Department of Information Technology
HCTM Kaithal, India

Rajni Jindal
Department of Computer Engineering
Delhi Technological University, India
(Formerly Delhi College of Engineering)

ABSTRACT
Patents and Research papers are published in various fields. These are stored in various conferences and journals database. If a user (researcher or any general user) want to search for any patent or research paper in any particular field, then there is lack of search criteria available for this. In this paper, we have used nearest neighbor algorithm with cosine similarity to categorize patents and research papers. In this paper, experimental results show that if a user want to search for the patent or research paper in any particular field or category, then user would get better results. The advantage of the approach presented in this paper is that the search area becomes very small and so waiting time of user to get answer of query reduces to a large extent. To take decision about category of particular research paper or patent, there have been a lot of research work but categorizing was not that much accurate. In this paper, we have calculated threshold based on the similarity of terms between query and research paper or patent. This proposed calculation of threshold value is not based on numerical values. So, this novel approach of threshold calculation categorize more accurately than previous research work.

General Terms
Data Mining, Search Engine, Patent and Research Papers ranking and classification and Information Retrieval.

Keywords
Search Engine, Term Frequency, Inverse Document Frequency, Vector Space Model, Nearest Neighbor, S-Cut Threshold

1. INTRODUCTION
World Wide Web (WWW) has large database of research papers and patents. Information Retrieval System is used for effective and efficient extraction of information from these research papers and patents. But a user who wants information about research papers and patents, only wants based on the relevance or importance. Research paper ranking is used for sorting the results based on relevance for the user [1]. These research papers are ranked by different algorithms. This ranking is based on static and dynamic properties of research paper on web [2]. We have used static algorithm to rank and classify these research papers and patents.

Only ranking of research papers in not efficient for a user. If a user wants to search research paper only in 'Data Mining' category, then there is no need of searching all categories research paper. So, research paper and patent classification is used for making efficient search. Nearest Neighbor algorithm is used for this classification. Novel approach for calculating threshold value in research paper and patent classification is proposed in this paper.

2. RESEARCH PAPERS AND PATENTS RANKING AND CLASSIFICATION
Different models and algorithms are used for ranking and classification of research papers and patents. Approximate all of ranking algorithms use TF/IDF and Vector Space Model [3]. For classification purpose, Nearest Neighbor Algorithm is used. These algorithms are explained in brief in following subsections.

2.1 Term Frequency/Inverse Document Frequency
When a user asks for some research paper or patent, then search engine should return results according to relevance [4]. This relevance is calculated by different methods. Static and Dynamic methods are used for this relevance calculation [5]. Term Frequency/Inverse Document Frequency (TF/IDF) is used to calculate relevance based on static method. Term Frequency is the number; a term is having in a research paper. Only simple Term Frequency gives more relevance to that research paper which is having more terms of query [6]. For example, if a user asks for information about "Search Engine Algorithm", then simple Term Frequency model gives more relevance to research papers having more frequency of terms "Search" and "Engine".

Only Term Frequency is not sufficient to give proper relevance to the user. There are many stop words also in a research paper and these stop words also come in the calculation of term frequency. Moreover only local information is checked by TF. Inverse Document Frequency (IDF) is used to get global information about corpus of research papers [6].

$$IDF_i = \log(\frac{D}{di}) \qquad (1),$$

So, IDF determines terms which are more discriminative than other terms. By checking whole corpus, global information is checked by IDF.

The weight of a term in TF/IDF model is calculated by:-

$$wi = tfi * \log\left(\frac{D}{dfi}\right) \qquad (2),$$

where tf_i is the number of occurrence of term i in a document, D is total number of documents and df_i is the number of documents containing term i.

2.2 Vector Space Model

After having weights of different terms using TF/IDF, checking of similarity between query and research paper contents is required by some method. Vector Space Model is used for calculating this similarity. Research paper and query is represented in form of vectors. Cosine angle is used to calculate similarity between these vectors [7]. Cosine similarity or Cosine distance is calculated by:-

$$Co\sin e\theta di = \frac{Q.Di}{|Q|.|Di|} \qquad (3),$$

Where Q.Di is the dot product of query weights with research paper weights. $|Q|$ is length of query vector and $|Di|$ is length of research paper vector. Similarity is same as the cosine angle between research paper vector and query vector.

$$Sim(Q, D_i) = Co\sin e\theta_{Di} \qquad (4).$$

2.3 Classification

Ranking shows the relevance of a research paper or patent based on the query of user [8]. But if a user wants research paper only in the category of 'search engine', then there is no use of searching all corpus of research papers. The efficient way is to first categorize the research papers and then only search in category of query by user [9].

Nearest Neighbor Algorithm is used for research paper classification when research paper is represented in form of Vectors using Vector Space Model. Research papers are first classified into different classes and then query terms are matched only within that class [10]. This reduces the search space and results are more efficient and effective for the user. Threshold is set on the similarity score between predefined class label and testing research paper. If similarity score is above that threshold then only that testing research paper is assigned that class. A novel approach for making this threshold calculation parameter free is given in this paper and explained in detail in Section 4.

3. EXPERIMENT

We have implemented this proposed technique in Java. In the first step, index is created using terms which have more discriminative power and which are rare in corpus of research papers. Stop words are not added in the index. We have taken some research papers for checking the relevance based on query by user. Steps are explained in following subsections which are followed for implementation in Java.

3.1 Indexing

Index is the first step of any ranking and classification algorithm. When any research paper is added in the corpus, then terms of that research paper is automatically added to the index and so index is updated regularly.

Suppose following are some part of seven research papers:

D1 = "search engine is used to rank web pages based on relevance according to query of user"

D2 = "google altavista bing are examples of search engine"

D3 = "data mining and data warehousing are used to make data efficient and effective for the user"

D4= "nearest neighbor decision tree are examples of data mining algorithm"

D5= "google search engine uses pagerank algorithm"

D6= "alpha and beta testing are examples of software testing"

D7= "spiral model is effective software engineering design model"

The query is Q = "search engine"

Index is created of these terms of research papers. Stop words are removed from the index. "pages", "search", "engine", "rank", "ranking", "user", "relevance", "web", "google", "altavista", "bing", "data", "mining" ,"nearest", "neighbor", "decision", "tree", "algorithm", "algorithms", "pagerank", "software", "testing", "engineering", "alpha", "beta", "spiral".

3.2 Term Frequency

Different terms are counted in different research papers to calculate Term Frequency.

Table 1

Terms	q	D1	D2	D3	D4	D5	D6	D7
pages	0	0	0	0	0	0	0	0
search	1	1	1	0	0	1	0	0
engine	1	1	1	0	0	1	0	0
rank	0	1	0	0	0	0	0	0
ranking	0	0	0	0	0	0	0	0
user	1	0	0	0	0	0	0	0
relevance	0	0	0	0	0	0	1	1
web	1	0	0	0	0	0	0	0
google	0	1	0	0	1	0	0	0
altavista	0	1	0	0	0	0	0	0

bing	0	1	0	0	0	0	0	0
Data	0	0	0	1	1	0	0	0
mining	0	0	0	1	1	0	0	0
nearest	0	0	0	0	1	0	0	0
neighbor	0	0	0	0	1	0	0	0
decision	0	0	0	0	1	0	0	0
tree	0	0	0	0	1	0	0	0
algorithm	0	0	0	0	1	1	0	0
algorithms	0	0	0	0	0	0	0	0
pagerank	0	0	0	0	0	1	0	0
software	0	0	0	0	0	0	1	1
testing	0	0	0	0	0	0	1	0
engineering	0	0	0	0	0	0	0	1
alpha	0	0	0	0	0	0	1	0
beta	0	0	0	0	0	0	1	0
spiral	0	0	0	0	0	0	0	1

3.3 Inverse Document Frequency

Inverse Document Frequency (IDF) is used to increase the weight of terms which have more discriminative power. IDF calculates the relevance of rare terms in the corpus of documents.

Table 2

Terms	**IDFi=log(D/dfi)**
Pages	1
Search	0.3979
Engine	0.3979
Rank	1.0
Ranking	1.0
User	0.5229
Relevance	0.699
Web	1.0
Google	0.699
Altavista	1.0
Bing	1.0
Data	0.301
Mining	0.5229
Nearest	0.699
Neighbor	0.699
Decision	0.699
Tree	0.699
Algorithm	0.5299
Algorithms	1.0
Pagerank	1.0
Software	0.5229
Testing	0.699
Engineering	0.699
Alpha	1.0
beta	1.0
spiral	0.699

3.4 Research Papers and Patents Vectors lengths

Research papers and query is represented in the form of vectors. Similarity is calculated by cosine angle between research paper and query. Vector length is calculated by:

$$\left|Di\right| = \sqrt{wd^2} \quad (5),$$

$$\left|Q\right| = \sqrt{wq^2} \quad (6),$$

So, by using (4) and (5), research papers and query vector length is calculated. In our experiment putting (4) and (5), the result is:

$$\left|D1\right| = 1.8183$$

$$\left|D2\right| = 1.6749$$

$$\left|D3\right| = 0.7984$$

$$\left|D4\right| = 1.6099$$

$$|D5| = 1.4418$$

$$|D6| = 1.6619$$

$$|D7| = 1.1183$$

$$|Q| = 0.5628$$

3.5 Cosine Similarity

Similarity is calculated by the cosine angle between two vectors i.e. research paper vector and query vector. Using (3), cosine similarity can be calculated.

Sim (Q, D1) =0.3095

Sim (Q, D2) =0.336

Sim (Q, D3) =0.0

Sim (Q, D4) =0.0

Sim (Q, D5) =0.3903

Sim (Q, D6) =0.0

Sim (Q, D7) =0.0

3.6 Ranking and Classification

After calculating similarity between research paper vector and query vector, research papers are displayed to user in order of relevance. In our experiment research paper 5 is having first rank, research paper 2 is having second rank and research paper 1 is having third rank.

In our experiment, similarity of research papers with Search Engine class is as follows:

Sim (D1, Search Engine) =0.4115

Sim (D2, Search Engine) =0.1312

Sim (D3, Search Engine) =0.2375

Sim (D4, Search Engine) =0.0

Sim (D5, Search Engine) =0.1524

Sim (D6, Search Engine) =0.0

Sim (D7, Search Engine) =0.0

So, research paper 1 and research paper 3 are classified in Search Engine class.

Similarity of research papers with Data Mining class is as follows:

Sim (D1, Data Mining) =0.0

Sim (D2, Data Mining) =0.0

Sim (D3, Data Mining) =0.2406

Sim (D4, Data Mining) =0.8495

Sim (D5, Data Mining) =0.1001

Sim (D6, Data Mining) =0.0

Sim (D7, Data Mining) =0.0

So, research paper 3 and research paper 4 are classified in Data Mining class.

Similarity of research papers with Software Engineering class is as follows:

Sim (D1, Software Engineering) =0.0

Sim (D2, Software Engineering) =0.0

Sim (D3, Software Engineering) =0.0

Sim (D4, Software Engineering) =0.0

Sim (D5, Software Engineering) =0.0

Sim (D6, Software Engineering) =0.1471

Sim (D7, Software Engineering) =1.0

So, research paper 7 is classified in Software Engineering class.

4. PROPOSED TECHNIQUE FOR THRESHOLD CALCULATION

Research papers are classified based on similarity score between that testing research paper and predefined class label [15]. This similarity score is calculated in our experiment. But every research paper can not be assigned in that particular class even if it is having some similarity. This similarity can be due to matching of stop words or some terms which are synonyms.

There must be some threshold value which is deciding factor about assigning research paper to a particular class. This threshold calculation is based on numeric score which is preset before making decision about assignment of research paper to a particular class. But this numerical value always leads to wrong assignment of class to research paper. If the value of threshold is set too high then many research papers which should be assigned a class, are not assigned the class. If the value of threshold is set too low then many research papers which should not be assigned a class, are assigned the class. This is all because value of threshold is static and based on numeric score.

In this paper, threshold value is calculated dynamically. Threshold value is calculated based on the number of terms present in research paper and then decided that how many terms should be matched to assign a class to that testing research paper. This novel approach to calculate threshold value removes the drawback of numeric calculation of threshold value. Threshold value from this approach is adaptive and parameter free.

5. CONCLUSION

Term Frequency/Inverse Document Frequency is very strong model for indexing and ranking when used along with Vector Space Model. In our paper we have shown that when query is matched in all search space then the result is not that much effective. This search space becomes very small when first research papers are classified. We have shown this in our experiment by using some research paper and then classifying these research papers and then calculating similarity with query. Calculation of similarity of query is limited only in the class of query. By using this approach, result is effective and efficient. Classification is not accurate if threshold value is based on only numeric score. We have given novel approach for calculating threshold value based on number of terms match.

6. FUTURE WORK

In this paper, research papers are ranked and classified based on TF/IDF with Vector Space Model. Nearest neighbor algorithm is used for research paper classification. In future, other algorithms are used for classification like BM25, Decision Tree etc. Results of these algorithms can be compared with nearest neighbor algorithm in future.

Novel approach proposed in this paper can be improved in future to make it more adaptive. This approach can be used to make classification more effective. Other similarity measures can be used instead of cosine similarity and then results can be compared.

7. REFERENCES

[1] Juan Ramos, Department of Computer Science, ICML 2005.Using TF-IDF to determine Word Relevance in Document Queries.

[2] Peter D. Turney, Patric Pantel, Journal of Artificial Intelligence Research, 141-188, 2010. From frequency to Meaning: Vector Space Models of Semantics.

[3] Christian Platzer, Schahram Dustdar ECOWS, IEEE 2005. A Vector Space Search Engine for Web Services.

[4] Stephan Robertson. Journal of Documentation, Volume 60, Number 5, pp. 503-520,2004.Understanding Inverse Document Frequency: On theoretical arguments for IDF, Microsoft Research.

[5] Sergey Brin, Lawrence Page. CNISDNS, Volume 30, Issue 1-7, pp.101-117, ACM 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine.

[6] S.Suseela. Periyar Maniammai University 2009. Document Clustering Based on Term Frequency and Inverse Document Frequency.

[7] Gang Qian, Shamik Sural, Yuelong Gu, Sakti Pramanik. SAC, pp.1232-1237, ACM 2004. Similarity between Euclidean and Cosine angle distance for nearest neighbor queries.

[8] T.W.Fox. IEEE 2005. Document Vector Compression and Its Application in Document Clustering.

[9] John Zakos, Brijesh Verma. ICDAR, pp.909-913, IEEE 2005.A Novel Context Matching Based Technique for Web Document Retrieval

[10] Yun-lei Cai, Duo Ji, Dong-feng Cai. NTCIR-8, 2010. A KNN Research Paper Classification Method Based on Shared Nearest Neighbor.

[11] Isa, D., Lee, L. H., Kallimani, V. P., and Rajkumar, R. IEEE Transactions on Knowledge and Data Engineering, Vol. 20, pp. 23-31. Text document preprocessing with the Bayes formula for classification using the support vector machine.

[12] Songbo, T., Cheng, X., Ghanem, M. M., Wnag, B., and Xu, H. Proceedings of Fourteenth ACM International Conference on Information and Knowledge Management, pp 469 – 476, 2005. A novel refinement approach for text categorization.

[13] Lan, M., Tan, C. L., Su. J., and Lu, Y. IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume: 31 (4), pp. 721 – 735, 2009. Supervised and Traditional Term weighting methods for Automatic Text Categorization.

[14] Juan Zhang, Yi Nui, Huabei Nie. International Conference on Computational Intelligence and Security 2009. Web Document Classification Based on Fuzzy k-NN Algorithm.Alok Ranjan, Eatesh Kandpal, Harish Verma, Joydip Dhar. IJCSIS Vol.7 ,No. 2, pp. 257-261, 2010. An Analytical Approach to Document Clustering Based on Internal Criterion Function.