Semantic Web Improved with IDF Feature of the TFIDF Algorithm

Jyoti Gautam, Ela Kumar, and Mehjabin Khatoon

***Abstract*— Search engines are growing and the development is taking at a very fast rate. Different algorithms have been tried and tested. Still, there is less relevance between user queries and web information retrieved. A lot of improvement can be done to enhance the results. Many interesting problems have generated as the success and popularity of the social network giants like delicious, Facebook and CiteUlike are also growing at a tremendous rate. This provides us with a new perspective on how to improve the quality of information retrieval. In addition to this, many techniques of text classification are based on the term frequency (tf) and inverse document frequency (idf) for representing importance of terms and computing weights in classifying a text document. Term weighting has a significant role in text classification. In this paper, we are extending the queries by "keyword+tags" instead of keywords only. In addition to this, we have developed a new ranking algorithm which utilizes semantic tags to enhance the already existing semantic web by using the IDF feature of the TFIDF algorithm**.

Index Terms— Expanded query, Ranking, Semantic NewSearch, Textclassification, Tf-idf.

I. Introduction

With lots of information available on networks, search engines have become the primary infrastructure for retrieving information and are the second largest Internet applications in addition to e-mail. Results-sets have less relevance in response to the user queries as required. The survey done estimates that 85% to 90% of the Internet users generally click on the first two pages of search results. It means that a good ranking algorithm is required to put the best results on the front.

Many popular Web services like Delicious, Citeulike and flickr.com rely on folksonomies(Gautam and kumar 2012).There are some websites such as CiteUlike (Research Paper Recommender), Delicious (online bookmarking), Flickr (online photo management and sharing application), Furl (File Uniform Resource Locators), Blinklist (links saver), Diigo (collect and organize anything e.g. bookmarks, highlights, notes, sceenshots etc.), Otavo (collaborative web search), Stumbleupon (discovery engine), Blummy (tool for quick access to favorite web services), and Folkd (saves bookmarks and links online) etc. which contain these tag information.

Research on folksonomies is growing at a very fast rate in spite of the various difficulties encountered. The focused areas have been on the study of the data properties, the analysis of usage patterns of tagging systems, the discovery of hidden semantics in tags, the using of annotations in enterprise search, and the user's interest in discovery for personalized search.

Searches based on social-bookmarking have become increasingly popular, which lets users specify their keywords of interest, or tags on web resources. Social tagging, also known as social annotation or collaborative tagging is one of the major characteristics of Web 2.0. Social-tagging systems allow users to annotate resources with free-form tags. The resources can be of any type, such as Web pages (e.g., delicious), videos (e.g., YouTube), photographs (e.g., Flickr), academic papers (e.g., CiteULIke), and so on [8].

In this paper, the following approach has been adopted. We have tried to use the metadata available in the form of user feedback from CiteUlike.

a. A novel approach has been developed. A ranking algorithm based on semantic tags which utilizes the IDF feature of TFIDF algorithm, is proposed and the data is obtained through CiteUlike.

b. The query was expanded. The idea was to use "keyword + tags" instead of keywords only.

c. The data for the tags was obtained through CiteUlike and the comparison of the approach was done with Google by several evaluation methods.

II. The Existing Ranking Methods

(Berger 1999)Text classification is the key technique in the data mining (DM) and information retrieval (IR) field and it has got a lot of attention from the research community in the current decades. Research has been done to improve the quality of text representation and develop high quality classifiers. Text classification (TC) is a task to sort automatically text documents into categories from a predefined set. Most of the machines learning methods treat text documents as bag of words.

(J Ramos) As one of the key techniques for Text Classification, TFIDF algorithm calculates values for each word in a document through an inverse proportion of the

Manuscript received December 27, 2013; reviewed on January 16, 2014.

Jyoti Gautam is with the the Department of Computer Science and Engineering, JSS Academy of Technical Education, C-20/1, Sec-62, NOIDA, Uttar Pradesh, India (Phone: +919958306194; fax: 01202400097; e-mail: jyotig@jssaten.ac.in).

Ela Kumar is with the School of Information and Communications Technology, Gautam Buddha University, Greater Noida, Uttar Pradesh, India (phone: +919873426162; e-mail: ela_kumar@gbu.ac.in).

Mehjabin Khatoon passed her M.Tech. from the Department of Computer Science and Engineering, JSS Academy of Technical Education, NOIDA, Uttar Pradesh, India in the year 2012.(Phone: +918802430496; e-mail:mehjabinkhatoon@gmail.com).

frequency of the word in a particular document to the percentage of documents the word appears in. Words with high TFIDF numbers imply a strong relationship with the document they appear in, suggesting that if that word were to appear in a query, the document could be of interest to the user.

(Jiang 2009)There have been numerous changes in the basic TFIDF algorithm. In yet another method of basic TFIDF model which uses supervised term weighting approach, the model uses class information to compute weighting of the terms. The approach is based on the assumption that low frequency terms are important, high frequency terms are unimportant, so it designs higher weights to the rare terms frequently.

(Zhanguo 2011)Unlike the case of unsupervised-based term weighting approach, supervised term weighting uses category information in the training set. It uses the inner and intra class information. In literature(Lertnattee 2004), interclass standard deviation(icsd), class standard deviation(csd) and standard deviation, were introduced to tf-idf model, the performance of classification is enhanced.

(Agichtein 2006) proposed a generalized approach to model user behavior beyond click through, which results in higher preference prediction accuracy than models based on click through information alone.

(Wu 2006) explored the technique of Social Annotations for the Semantic Web. These annotations are manually made by normal web users without a predefined formal ontology. Compared with the formal annotations, despite social annotations are coarse-grained, informal and vague, they are also more accessible to more people and better reflect the web resources' meaning from the user's point of views during their actual usage of the web resources. As an example of social bookmark service, it can be shown how emergent semantics can be statistically derived from the social annotations. Furthermore, the emergent semantics can be applied to discover and search shared web bookmarks. The evaluation of the approach shows that the method can effectively discover semantically related web bookmarks that current social bookmark service cannot discover easily.

(Farooq 2007) The author proposes six tag metrics to understand the characteristics of a social bookmarking system. Using the metrics, possible design heuristics was suggested to implement a social bookmarking system for Cite Seer.

(Xu 2008) proposed a personalized search framework to utilize folksonomy for personalized search. Specifically, three properties of folksonomy, namely the categorization, keyword, and structure property, were explored.

(Jin 2008) proposed the architecture of a semantic search engine and an improved algorithm based on TFIDF algorithm. The algorithm considers crawling of static web pages. The algorithm can be considered for crawling of dynamic web pages and for parallel crawling also.

(Shaikh 2012) proposed the Semantic Web based Intelligent Search Engine. SWISE required including domain knowledge in the web pages to answer intelligent queries. The layered model of Semantic Web provides solution to this problem by providing tools and technologies to enable machine readable semantics in current web contents.

(Jomsri 2010) proposed a framework for Tag-Based Research Paper Recommender system. User self-defined tags were used for creating a profile for each individual user and cosine similarity was used to compare a user profile and research paper index. The recommender system demonstrated an encouraging preliminary result with the overall accuracy percentage up to 91.66%. The number of subjects is considered to be small in the experiment.

(Parra-Santander 2010) developed and evaluated two enhancements of user-based collaborative filtering algorithms to provide recommendations of articles on CiteUlike. The results obtained after two phases of evaluation suggested that both enhancements were beneficial.

(Zhao 2010) proposed a new viewpoint on how to improve the quality of information retrieval. The queries are extended by "keywords+tags" instead of keywords only. A new tag based ranking algorithm (OSEARCH) was proposed and the results obtained were also compared with Google by several evaluation methods.

III. User Query Intent and Storage of Tags

3.1. Metadata Information in the Web Pages and Expansion of the Query

The search engine shows only a query input interface and the result pages after handling the query. The search engines which deal with the query are quite complex, which are based on traditional and contemporary methods of information retrieval. The ranking methods discussed above are all doing ranking according to the relevance between results page and query. The method should be such that which tries to capture the user's real query intent. The primary purpose howsoever remains the same .i.e. to return the optimal results. But before returning the results, it should be able to analyze the query clearly. The simple keywords can't express user's real query intent. In order to analyze the query, some metadata information is added along with the query. The metadata information is added by expanding the query.i.e., keyword+tags instead of the keywords only.

So, the idea is to consider utilizing metadata which is available in the form of semantic tags .One area that arises is to consider utilizing the semantic tag information with web page. When users submit their query, they can also submit some simple semantic description to narrow down the query. Then by matching the semantic information between query and web page metadata, we can understand user's query intent better and return better result.

So, the idea is to improve the already existing semantic web by using some good features of TFIDF. The development of a new algorithm based on semantic web and TFIDF.

3.2. Storage of Semantic Tags on Web Page

The semantic tags of a web page are some object properties that can reflect the content of the page, such as marked with "pdf", which signifies that the page contains information about the object of "pdf". Of course, there may

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

be multiple tags on a page, because the pages always contain multi information.

In our case, we are storing the tags from CiteUlike. A popular website in academia is CiteULike (www.CiteULike.org). CiteUlike is a free service for managing and discovering scholarly references.

- Easily store references you find online
- Discover new articles and resources
- Automated article recommendations
- Share references with your peers
- Find out who's reading what you are reading
- Store and search your PDF's

CiteULike has a filing system based on tags. Tags provide an open, quick and user-defined classification model that can produce interesting new categorizations.

Additionally, it is also capable to:

- 'tag' papers into categories.
- Add your own comments on papers.
- Allow others to see your library.

The tags are retrieved from CiteUlike. The URLs along with their tags are stored in a local database. For the semantic tags, each URL is opened in CiteUlike and those tags with their highest numeric values are stored in the database. We add tags' values in the five columns. The data was retrieved from April, 2012 to July, 2012 from CiteUlike for the 31 queries. A total of 3100 URLs were opened in CiteUlike and the database was created.

IV. A NEW OPTIMIZED RANKING ALGORITHM

In this paper, we are proposing a new algorithm based on semantic tags in the web pages. We are proposing an enhanced semantic web algorithm. The algorithm is based on utilizing the metadata information available with the web pages by integrating in the algorithm some good features of TFIDF.

Initially, when users want to submit a query, instead of just giving the query in the form of keywords, they will also expand the query by adding some metadata information along with the query. Afterwards, the algorithm compares the inputted tags in query with the semantic information on the web pages in order to provide the user with better results.

Accordingly, the user query can be expressed as:

Query = {keyword1, keyword2,…, tag1, tag2,…}

In the above formulation, keyword1, keyword2 is the main query keyword.Tag1; tag2 is the semantic information which we are adding to expand the query. For example, Query = {research papers, web mining) represents that the user wants to find information relating to research papers on web mining.

Similarly, Query = {resources, information retrieval}

represents that the user wants to find information relating to resources in the field of information retrieval.

Once, the query is submitted, the system creates a vector of all the user tags.

V_usrt = {user_tag1, user_tag2,…}

Once the query is submitted to the search engine, the engine returns an initial result page list. The vector of all the tags on the result pages is recorded.

V_rest = {r_tag1, r_tag2,…}

Where, r_tag1, r_tag2 represent semantic tags on result pages.

The similarity is calculated between the two tag vectors, and recorded as a Tg_score.

Then, the final score of the web page is:

TotalScore=google_score+Tg_score*IDFscore (1)

Score=Tg_score*IDFscore (2)

Re – rank the google results according to this score. Google_score represents the original Google results score when the query is applied in eq. (1).

Google_score=(p-q+1)/p (3),

Here, p represents the total no. of documents, which is 100 in the experiment; q represents the location of the document on search engine's result list. So, google_score for the 4th result is (100- 4 + 1) / 100 = 0.97.

In the Eq. (1), Tg_score is calculated by matching the tags of the user with the tags of the result page. The match between the two vectors is based on the following factors.

1. The similarity between the user tag vector and web page tag vector. The high value is obtained by high similarity between the two vectors.

2. The other factor being the weight of the tags on the result pages. Weight refers to the frequency of the tags in the result pages which match with the tags of the user.

Tg_score is defined as given below based on the factors considered:

$$\text{Tg\_score} = \frac{\sum_{i=1}^{|V\_usrt|}\sum_{k=1}^{|V\_rest|}(freq(V\_rest[k])*sim(V\_usrt[i],V\_rest[k]))}{\sum_{k=1}^{|V\_rest|}freq(V\_rest[k])} \quad (4)$$

In the above equation, freq (tag) represents the frequency or weight of the particular tag on the result page. $sim(V\_usrt[i]),V\_rest[k])$ represents the similarity between the user tag vector $V\_usrt[i]$ and the result page tag vector $V\_rest[k]$ and similarity is defined as given below:

$sim(V\_usrt[i]),V\_rest[k])$

= 1, V_usrt[i] and V_rest[k] have the same root,

= 1, V_usrt[i] and V_rest[k] have the same meaning,

= 0, V_usrt[i] and V_rest[k] does not have a semantic relation,

= 0.5, even if half of the V_usrt[i] tag resembles with the V_rest[k] tag. (5)

,e.g. let us say in the Query = {resources, information retrieval} , resources is the keyword and information retrieval is the tag, then in the tags of the result pages even if information or retrieval appears , we have taken the similarity score as 0.5.

Next, ,e.g. consider the query , Query = {artificial intelligence, pdf} to Google, The tenth result has the tags as "pdf", "pdfs", "research" and the frequency of the tags is 2, 3, 4 respectively. Then, the value of the Tg_score = (2*1+3*1+4*0)/(2+3+4) = 5/9 and google_score = (100-10+1)/100=0. 91.

Next in the equation (1) is the IDF score, we know from the TFIDF algorithm.

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Given a document collection D, a word w, and an individual document d ϵ D, we calculate

$$w_{d=}f_{w,d*}\log(|D|/f_{w,D}), \quad (6)$$

Where $f_{w,d}$ equals the number of times w appears in d, |D| is the size of the corpus, and $f_{w,D}$ equals the number of documents in which w appears in D. Words with high w_d imply that w is an important word in d but not common in D.

Here, if the above equation is analyzed properly, we see that if we replace words with tags, this equation (6) can be used in the context of semantic web. So, $f_{w,d}$ has already been considered as the Tg_score. Now remains the log ($|D|/f_{w,D}$), (which is IDF score). Here, for each query, we have taken the 100 Google results. So, for a particular query, D is 100 and $f_{w,D}$ equals the number of documents in which the particular tag of the query appears.

Now, why we have included this IDF score?

Suppose that Tg_score is large and $f_{w,D}$ score is small. Then $\log(|D|/f_{w,D})$ will be rather large, and so in Eq. (1), the score will be large. This is the case we are most interested in, since tags with high score imply that tag is important in d but not common in D. This tag is having a large discriminatory power. Therefore, when a query contains this tag, returning a document d where score is large will very likely satisfy the user.

Now, calculating the IDF score for the Query = {books, artificial intelligence}, let us say that the documents in which the tag artificial intelligence appears is 30 and the value of D is 100. So, the IDF score is log (100/30).

In the above equation (Eq. (4)), we are using java functions to calculate the similarity between user tags and result tags. The database is created using MYSQL. We cannot store all the tag values (because the size of the database will become unlimited and unmanageable) so that we are recording those tag's values which are the highest. We are storing the tag's values in the five columns only i.e., tag1, tag2, tag3, tag4, tag5.

For example, user submits the query "research papers, mobile computing", to Google, the 4th result of Google is having the tag's values, mobile computing = 37, mobile devices = 35, mobile interaction = 27, pedestrian navigation = 23, navigation = 12. And, the tag mobile computing appears in 37 documents. So, according to the above algorithm, the total score = (0.97) + (0.507) * log (100/37).

V. Experiments and Analysis

The experiments are performed as follows:

1. Initially, submit the query to Google, and obtain the original Google search results.
2. Re-rank the search results according to our algorithm.
3. Compare the two result-sets.

A. Data Set

Query Set: Initially, we determine the queries which we input to the search engine. We determine a total of thirty one queries.

Result Set: Now, submit each query to Google and record the first 100 results. This way, the result set of 31 queries is 3100 results.

Results Tag Set: Now, we submit the 3100 results to CiteUlike and the resulting tag vector is recorded. We obtain lots of tag values for a result, we cannot store all the tag values so that we are recording those tag's values which are the highest. We are storing the tag's values in the five columns i.e., tag1, tag2, tag3, tag4, tag5.

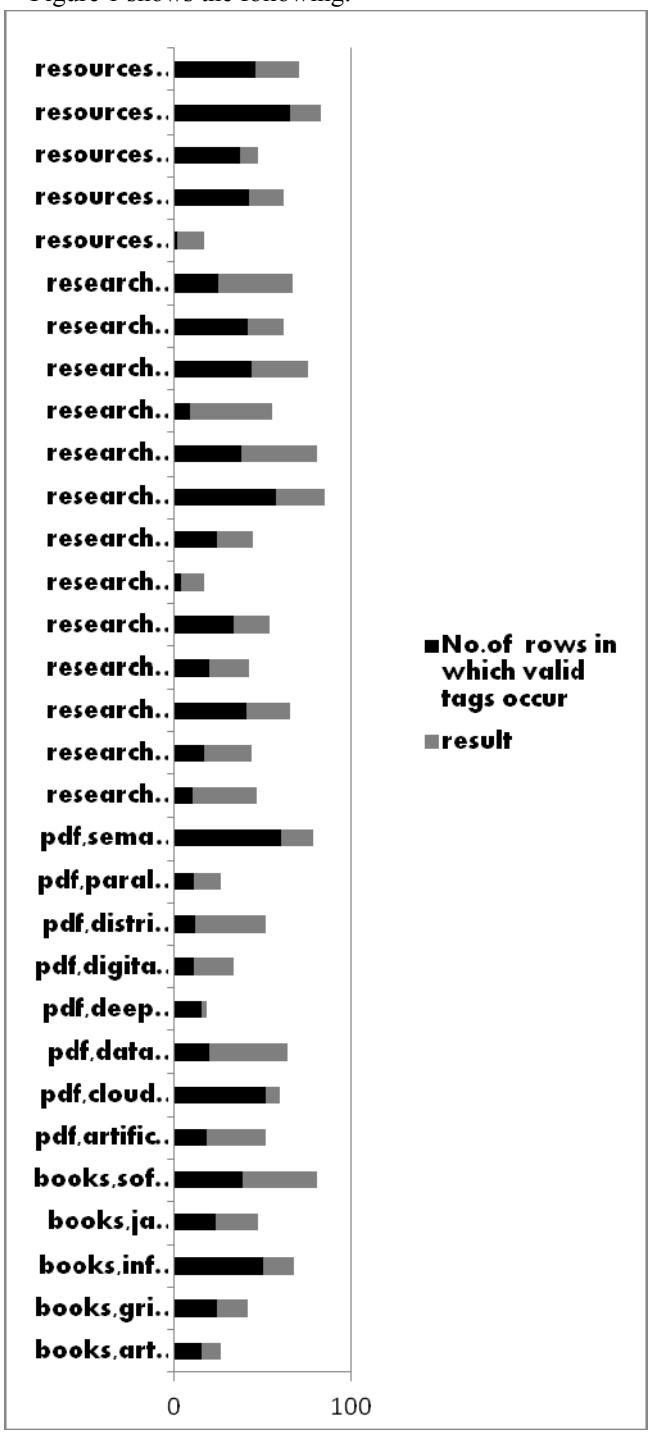
For example, user submits the query "research papers, mobile computing", to Google, the 4th result of Google is having the tag's values, mobile computing = 37, mobile devices = 35, mobile interaction = 27, pedestrian navigation = 23, navigation = 12. And, the tag mobile computing appears in 40 urls. So, according to the above algorithm, the total score = (0.97) + (0.507) * log (100/40).

Let us say, the following queries have been chosen.

Q1 = {books, artificial intelligence}
Q2 = {books, grid computing}
Q3 = {books, information retrieval}
Q4 = {books, java programming}
Q5 = {books, software engineering}
Q6 = {pdf, artificial intelligence}
Q7 = {pdf, cloud computing}
Q8 = {pdf, data structure}
Q9 = {pdf, deep web}
Q10 = {pdf, digital image processing}
Q11 = {pdf, distributed computing}
Q12 = {pdf, parallel algorithm}
Q13 = {pdf, semantic web}
Q14 = {research papers, communication}
Q15 = {research papers, compiler}
Q16 = {research papers, data mining}
Q17 = {research papers, genetic algorithm}
Q18 = {research papers, mobile computing}
Q19 = {research papers, pharmacology}
Q20 = {research papers, quantum cryptography}
Q21 = {research papers, semantic web}
Q22 = {research papers, software engineering}
Q23 = {research papers, statistics}
Q24 = {research papers, ubiquitous computing}
Q25 = {research papers, web mining}
Q26 = {research papers, wireless communication}
Q27 = {resources, electronics engineering}
Q28 = {resources, grid computing}
Q29 = {resources, information retrieval}
Q30 = {resources, semantic web}
Q31 = {resources, ubiquitous computing}

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Figure 1 shows the following:



A = Rows in which the specific tags occur
B = Rows of total tags
C = Difference Tags = B-A
e.g., for the query = {books, artificial intelligence}
A = 15, B = 26, C = 11.
Figure1. The number distribution of specific tags versus difference tags in a result set is shown .

B. Experimental Results

First, we determine the relevance between each query intent and each result page. Each result is assigned a relevance score according to its relevance, which ranges between 0 to 3 (totally irrelevant, basically irrelevant, basically relevant, and totally relevant).

We obtain normalized DCG values for our algorithm and Google as given in the Table 1.

TABLE 1.
COMPARISON OF NORMALIZED DCG (*nDCG*) VALUES FOR OUR ALGORITHM AND GOOGLE

Queries no.	nDCG(A)	nDCG(G)
q1	0.957424	0.970031
q2	0.888747	0.913824
q3	0.877744	0.862172
q4	0.938299	0.934294
q5	0.854472	0.881374
q6	0.887192	0.885906
q7	0.975138	0.97113
q8	0.86662	0.8918
q9	0.834386	0.796038
q10	0.920252	0.942012
q11	0.959862	0.953069
q12	0.995585	0.995332
q13	0.982126	0.981987
q14	0.897661	0.84126
q15	0.881929	0.848669
q16	0.933084	0.894468
q17	0.975616	0.983474
q18	0.908892	0.85308
q19	0.805438	0.801738
q20	0.929742	0.91508
q21	0.945845	0.938982
q22	0.92802	0.913109
q23	0.879856	0.770643
q24	0.956999	0.945143
q25	0.83687	0.760944
q26	0.934957	0.92141
q27	0.928905	0.928905
q28	0.994868	0.994253
q29	0.957072	0.964861
q30	0.993879	0.992997
q31	0.986664	0.984467

We obtain normalized DCG values for the 31 queries for our algorithm as well as for Google results. We observe that Figure 2. shows the normalized DCG values of 31 queries. The graph compares our algorithm with Google. It can be seen that our algorithm acquire higher values of DCG for 24 queries when compared to Google.

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

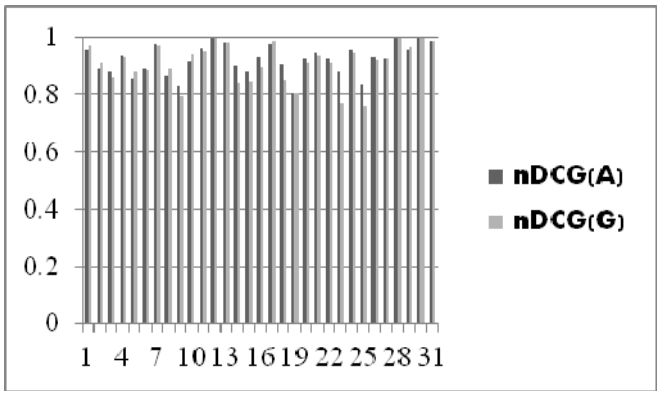


Figure2. The average DCG value of 31 queries (our algorithm Versus Google)

Next, we use Precision@k curve for various Relevance levels.

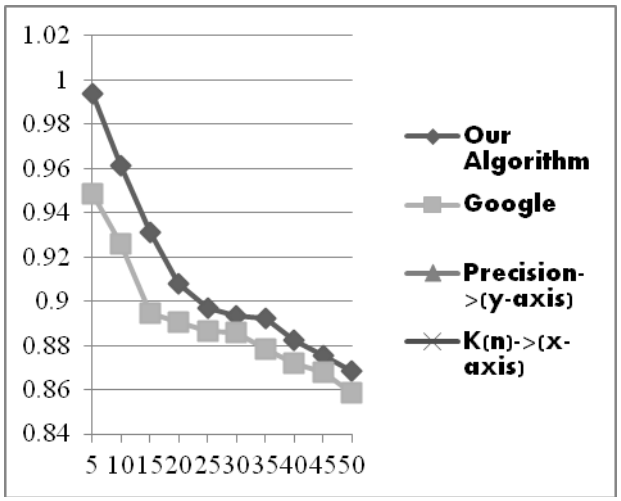


Figure3. The Precision@k curve of 31 queries when Rel>=1

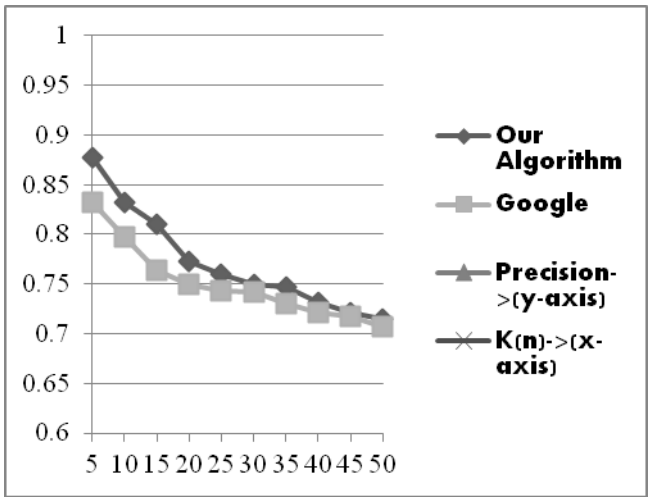


Figure 4. The Precision@k curve of 31 queries when Rel>=2

So, we can make the following conclusion from the Figure3. to Figure5. that our algorithm acquires higher precision in comparison to Google throughout the varying levels of K for all the 31 queries. The results obtained for Rel>=1 are the best as expected. The precision for Rel>=1 are better than Rel>=2, which is better than Rel>=3.

Now, we compute precision, recall and F1-score for our algorithm and Google (Table 2). We are calculating these values for all the queries. For all the queries, these values are calculated for their corresponding top 50 results for Rel>=2. We observe that the value of recall for our algorithm and Google remain at 1 as we have re ranked the top 100 results of Google for each query. The value of precision and F1-score are calculated and it has been observed that we are getting better results.

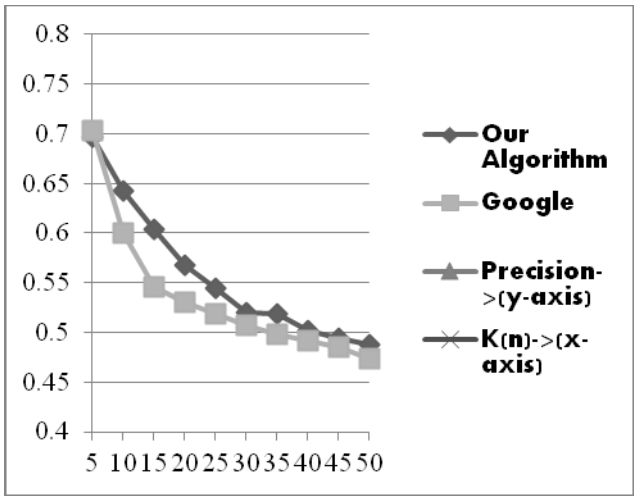


Figure 5. The Precision@k curve of 31 queries when Rel>=3

VI. Conclusions

In this paper, we have analyzed some existing ranking methods, and proposed a new algorithm based on the previous methods. We proposed the new algorithm using the already existing semantic web algorithm which basically calculates the weighted score of the tags. In addition to this, we have tried to integrate the features of TFIDF algorithm in the newly developed algorithm. We have utilized the IDF feature to improve the semantic web which uses tags. The Semantic tag of a web page is the metadata information associated with it and depicts a lot about the information associated with it. The match degree between user's real query intent and web page content is determined by calculating the similarity between query and web page tag.

In experiments, we have collected the data from Citeulike and implemented the above algorithm. Comparing with Google search results, we find that our method can acquire better ranking results, and can put more relevant results in front. In the future work, we will further improve the algorithm. We will consider combining with the search engines user logs, and mining out information repeated to user's query, such as the click information, the browse information and so on. The algorithm can be further enhanced by adding these effects.

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE 2
PRECISION AND F1-SCORE FOR OUR ALGORITHM AND GOOGLE

	Our Algorithm		**Google**	
Query	PRECISION	F1-score	PRECISION	F1-score
q1	0.94	0.969	0.96	0.98
q2	0.5	0.667	0.5	0.667
q3	0.34	0.507	0.38	0.551
q4	0.72	0.837	0.72	0.837
q5	0.72	0.837	0.7	0.823
q6	0.8	0.889	0.8	0.889
q7	0.96	0.979	0.94	0.969
q8	0.5	0.667	0.5	0.667
q9	0.32	0.485	0.3	0.461
q10	0.9	0.947	0.9	0.947
q11	0.88	0.936	0.86	0.925
q12	0.96	0.979	0.96	0.979
q13	0.98	0.99	0.98	0.99
q14	0.56	0.718	0.56	0.718
q15	0.5	0.667	0.48	0.649
q16	0.64	0.78	0.62	0.765
q17	0.92	0.958	0.88	0.936
q18	0.72	0.837	0.72	0.837
q19	0.26	0.413	0.24	0.387
q20	0.7	0.823	0.68	0.809
q21	0.86	0.925	0.86	0.925
q22	0.66	0.795	0.62	0.765
q23	0.42	0.591	0.42	0.591
q24	0.8	0.889	0.78	0.876
q25	0.48	0.649	0.48	0.649
q26	0.74	0.851	0.68	0.809
q27	0.6	0.75	0.6	0.75
q28	1	1	1	1
q29	0.84	0.913	0.84	0.913
q30	1	1	1	1
q31	0.94	0.969	0.94	0.969

REFERENCES

[1] Agichtein,E. and Brill E.and Dumais, S. and Rango,R. (2006) Learning user interaction models for predicting web search result preferences, in proc. of the 29th annual international ACM SIGIR conference on Research and Development in information retrieval, pp. 3-10

[2] Berger, A. and Lafferty, J. (1999) Information Retrieval as Statistical Translation, in Proc. of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR) ,pp. 222-229H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985, ch. 4.

[3] Farooq,U. and Kannampallil,T.G. and Song,Y.(2007) Evaluating Tagging Behaviour in Social Bookmarking Systems: Metrics and design heuristics, in the international ACM Conference on Supporting Group Work.E. H. Miller, "A note on reflector arrays (Periodical style—Accepted for publication)," *Engineering Letters*, to be published.

[4] Gautam, J. and Kumar, E. (2012) An Improved Framework for Tag-Based Academic Information Sharing and Recommender System, in Proc. of the World Congress on Engineering, Vol. 2, 845-850.

[5] Jiang, H. and Hu, X. and Li Ping and Wang, S.(2009) An improved method of term weighting for text classification, in International Conference on Intelligent Computing and Intelligent Systems, IEEE, Vol.1, pages 294-298.

[6] Jomsri,P. and Sanguansintukul, S. and Choochaiwattana, W. (2010) A Framework for Tag-Based Research Paper Recommender System: An IR Approach, in proc. of the 24th International Conference on Advanced Networking and Applications Workshops, IEEE, pages 103-108.

[7] Jin,Y. and Lin, Z. and Lin, H.(2008) The Research of Search Engine Based on Semantic Web, in proc. of International Symposium on Intelligent Information Technology Application Workshops(IITAW), IEEE, pages 360-363.

[8] Lu, C. and Hu, X. and Park, J. (Sep. 2011) Exploiting the Social Tagging Network for Web Clustering, (*Systems, Man, and Cybernetics – Part A: Systems and Humans*), vol. 41, pp. 840-852.

[9] Lertnattee,V and Theeramunkong,T. (2004) Effect of term distributions on centroid-based text categorization, *Information Sciences*.[Online].vol 158, pp. 89-115.Available: http://www.sciencedirect.com/science/article.

[10] Parra-Santander,D. and Brusilovsky,P.(2010) Improving Collaborative Filtering in Social Tagging Systems for the Recommendation of Scientific Articles., in proc. International Conference on Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM,vol.01, pages 136-142.

[11] Ramos, J. Using TFIDF to determine word relevance in document queries, in Proc. of the first Instructional Conference, Rutgers University, Piscataway.

[12] Shaikh, F. and Siddiqui, U.A. and Shahzadi, I.(2012) Semantic Web based Intelligent Search Engine, in proc. of International Conference on Information and Emerging Technologies, pp. 1-5.

[13] Salton, G. and McGill, M.J.(1983) *An Introduction to Modern Information Retrieval*, Mcgraw Hill.

[14] Shenliang, X. and Shenghua, B. and Fei, B.(2008) Exploring Folksonomy for Personalized Search, in proc. of the 31st annual international ACM SIGIR conference on Research and Development in information retrieval, pp. 155-162.

[15] Zhao, C. and Zhang, Z.(1-3rd Sept 2010) A New Keywords Method to Improve Web Search, in 12th International Conference on High Performance Computing and Communications, IEEE, pages 477-484..

[16] Zhanguo, M. and Jing, F. and Liang, C. and Xiangyi, H. And Yanqin, S.(2011) An improved approach to terms weighting in text classification , in proc. of the International Conference on Computer and Management, IEEE, pages1-4.

[17] Wu, X. and Zhang, L. and Yu, Y.(2006) Exploring Social Annotations for the Semantic Web, in proc. of the 15th International Conference on World Wide Web (WWW 06), ACM, pages 417-426.

[18] Keep, share, and discover the best of the Web using Delicious, the world's leading social bookmarking service. http:// delicious.com.

[19] Search, organize, and share scholarly papers. Indexes over 2 million articles. http://www.citeulike.org/ (accessed april, 2012 to july 2012)

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)