SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM

TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

VOLUME 2, ISSUE 5
ISSN: 2054 - 7390

Semantic Web Improved with Fuzziness added in Weighted Score

Jyoti Gautam and Ela Kumar
School of Information and Communication Technology, Gautam Buddha University, Greater NOIDA, INDIA;
jyotig@jssaten.ac.in; ela_kumar@gbu.ac.in

ABSTRACT

A lot of improvement has gone in the area of information retrieval. But, still improvements can be done. Social networking giants like Facebook, LinkedIn, CiteULike have taken a new role. There is a huge data collection from these sites. A lot of work is going on to convert this data into information. As we are aware that term weighting has a significant role in text classification. 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. 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 weighted score. The data for the tags has been obtained through CiteUlike. Here, we have manually added fuzziness in the weighted score for the purpose of improving the algorithm.

Keywords: Text classification; Semantic Web with weighted idf feature; Expanded query; Fuzzy Semantic Web; Fuzzy Ranking Algorithm.

1 Introduction

Result sets have less relevance in response to the queries given. To improve the results, a lot of research is happening in the direction of semantic web. There are lots of social networking sites and a lot of data is produced by them. But, there is urgent requirement of converting this data into meaningful information.

This paper utilizes the semantic tag information with web page. We have obtained this information from CiteULike [19] (Research Paper Recommender and online Tagging System). Semantic tag's information is added along with the query for the purpose of disambiguation. Then, by matching the semantic description between the query and web page, user's query intent can be well understood.

Research on folksonomies is growing at a very fast rate in spite of the various difficulties encountered. We have sites like Delicious [18], Facebook, LinkedIn and CiteUlike etc. which are producing this data. Social tagging, also known as social annotation or collaborative tagging is one of the major characteristics of Web 2.0.Users annotate resources with free-form tags in social-tagging systems 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 .

DOI: 10.14738/tmlai.25.333
Publication Date: 10th September, 2014
URL: http://dx.doi.org/10.14738/tmlai.25.333

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.

1. A new approach has been developed. A ranking algorithm based on semantic tags which adds the fuzziness in the weighted score, is proposed and the data is obtained through CiteULike.
2. The query was expanded. The idea was to use “keyword + tags” instead of keywords only.

   The data for the tags was obtained through CiteUlike.
3. The performance analysis of the approach was done with Google by using normalized DCG.

2 The Existing Ranking Methods

Tf-idf, term frequency-inverse document frequency, often used as a weighting factor in information retrieval, is a numerical statistic which reflects how important a word is to a document in a corpus. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to control for the fact that some words are generally more common than others (Wikipedia) [17]. Various variations of the tf–idf weighting scheme are often used by search engines. Search engines use these weighted measures as a central tool in scoring and ranking a document's relevance given a user query. The tf-idf is improved by many literatures.The literature [9] provides an improved approach named tf.idf.IG to remedy this defect by Information Gain from Information Theory.

The authors [12] explored the technique of Social Annotations for the Semantic Web. These annotations are manually made by normal web users without a predefined formal ontology. The evaluation of the approach shows that the method can effectively discover semantically related web bookmarks that current social bookmark service cannot discover easily.

[2] The authors use six tag metrics to understand the characteristics of a social bookmarking system. Possible design heuristics was suggested to implement a social bookmarking system for Cite Seer using the metrics. The authors Cilibrasi and Vitanyi in 2007 [1] described a technique for calculating the Google similarity distance. Jin, Lin and Lin [6] proposed the architecture of a semantic search engine and an improved algorithm based on TFIDF algorithm. The algorithm considers crawling of static web pages.
A personalized search framework was proposed by Shenliang, Shenghua and Fei [10] .It utilizes folksonomy for personalized search.

[5] The other method of basic TFIDF model uses supervised term weighting approach. The model uses class information to compute weighting of the terms.

The authors [14] proposed a trust-network based fuzzy knowledge sharing in semantic web in the year 2009. In this paper, we propose a trust network based fuzzy knowledge sharing model in semantic web. According to this model, any agent can broadcast its knowledge queries through the trust network, and aggregate different opinions according to the trustworthiness of each agent echoed.

Zhao and Zhang [15] 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.

The authors [8] focussed on search engine personalization and developed several concept-based user profiling methods that are based on both positive and negative preferences. The proposed methods were evaluated against the previously proposed personalized query clustering method.

Another supervised term weighting method, proposed by the authors, [13] provides an improved tf-idf-ci model to compute weighting of the terms. The method uses intra and inner class information.

The paper proposed by Yoo [11] suggests a hybrid query processing method for the effective retrieval of personalized information on the semantic web. When individual requirements change, the current method of query processing requires additional reasoning for knowledge to support personalization.

The authors [4] proposed the method of relevance feedback between hypertext and semantic web search. The paper proposed investigates the possibility of using semantic web data to improve hypertext web search.

The authors [16] proposed an effective pattern discovery method for text mining. The paper presents an innovative and effective pattern discovery technique which includes the processes of pattern deploying and pattern evolving, to improve the effectiveness of using and updating discovered patterns for finding relevant and interesting information.
The paper [7] proposes searching and ranking method of relevant resources by user intention on the semantic web.

This paper proposed by the authors [3] proposes a framework for a tag-based Academic Information Sharing and Recommender System which shares information such as question papers, assignments, tutorials and quizzes on a specific area.

3 User Query Intent and Storage of Tags

3.1 Metadata information in the Web Pages and Expansion of the Query

Semantic Web implies the content, meaning or the metadata which is related to the web pages. This metadata information is hidden in the web pages. Different websites are working upon it since a long time. Sites like Delicious, CiteUlike, Flickr, LinkedIn etc. allow different users to create their accounts. After creating the accounts, metadata can be added to the different web pages. This metadata conveys the content of the website as interpreted by different users.

The purpose of the search engine is to return optimal results. So, here we have tried to return the optimal results by modifying the ranking algorithm and expanding the query. We have added semantic information with the query. We have expanded the query by “ keyword+tags” instead of the keywords only.

So, the idea is to utilize this semantic tag information. Here, we are proposing the development of a new algorithm based on semantic tags and fuzziness is added in the tags to add a new dimension to the algorithm.

3.2 Storage of Semantic Tags on Web Pages

Multiple tags are associated with a web page, because the pages always contain multi information. These tags carry the semantic information or metadata along with them.

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 semantic tags are obtained from CiteUlike. The URLs along with their tags are stored in a local database. Each URL is opened in CiteUlike and the tags with their numeric values are stored in the database. We add tags' values in the MYSQL database. The data was retrieved from April, 2012 to June, 2013 from CiteUlike for the 10 queries. A total of 500 URLs were opened in CiteUlike and the database was created.

4 A new optimized ranking algorithm

4.1 A New Optimized Ranking Algorithm

Initially, when users want to submit a query, they will not only give the query in the form of keywords, they will also expand the query by adding some semantic 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 [15].

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 = {books, artificial intelligence) represents that the user wants to find information relating to books on artificial intelligence.

Similarly, Query = {research papers, statistics}

Represents that the user wants research papers in the field of statistics.

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,…}

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:

$$\text{TotalScore} = \text{google\_score} + \text{Tg\_score} * \text{IDF score} \quad (1)$$

$$\text{Score} = \text{Tg\_score} * \text{IDFscore} \quad (2)$$

Re – rank the google results according to this score.
Here, google score represents the original google results score when the query is applied.

$$\text{Google score} = (p-q+1)/p. \quad (3)$$

Here, p represents the total no. of documents, which is 50 in the experiment; q represents the location of the document on search engine's result list. So, google score for the 6th result is (50- 6 + 1) / 50 = 0.9.
In the Equation (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:

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

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

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

$$\begin{aligned} &\text{sim}(\text{V\_usrt[i]}), \text{V\_rest[k]}) \\ &= 1,\ \text{V\_usrt[i] and V\_rest[k] have the same root,} \\ &= 1,\ \text{V\_usrt[i] and V\_rest[k] have the same meaning,} \\ &= 0,\ \text{V\_usrt[i] and V\_rest[k] does not have a semantic relation,} \\ &= 0.5,\ \text{even if half of the V\_usrt[i] tag resembles with the V\_rest[k]tag.} \end{aligned} \quad (5)$$

The similarity between the user tag vector and the result page tag vector can have fuzzy values also. The fuzzy values range from 0 to 1. For example,

Let us take the example of the query {books, artificial intelligence}. For a single query, we are storing the tags for the first 50 Google results. We are storing tag weights for the tags with maximum values. For the tag agent, we have the weight as 3 and the fuzzy value that we are assigning is 0.1. Similarly for tags artificial, intelligence and systems, we have weights as 2, 2, 1 and the fuzzy values assigned by us are 0.5, 0.5 and 0.1. So, we get the weighted score (Tg_score) as (3*0.1+2*0.5+2*0.5+1*0.1)/ (3+2+2+1), which is equal to 0.3.

Google score for the first link is 1= ((50-1+1)/50).

The fuzzy values are assigned to tags, keeping in mind the relevance of that tag with the query. The assignment of fuzzy values is done by a group of users. The assignment is done manually by collecting users of that particular domain. Next in the equation (1) is the IDF score multiplied by Tg_score. We know from the TFIDF algorithm.

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

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

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 50 Google results. So, for a particular query, D is 50 and $f_{w,D}$ equals the number of documents which contain the tags.

Now, why we have included this IDF score?

Suppose that Tg_score is large and fw,D score is small. Then log ($|D|/f_{w,D)}$ will be rather large, and so in Equation. (1), the score will be large. This is the case we are most interested in, since this makes the score large in which we are interested. Here, we are calculating $f_{w,D}$ taking into consideration all the urls , which contain tags.

In the above equation Equation. (5), we have manually added fuzziness to the tags. The database is created using MYSQL.

For example, consider the query {books, web mining}.The query contains the tags - data mining, personalisation, personalization, web-personalisation, web- personalization. The weights of the tags are 3, 5, 6, 3, 3. The fuzzy values assigned are 0.9, 0.2, 0.2, 0.2, and 0.2. So, the Tg_score is 0.305 and Google score for the first link is 1. Since out of 50, 31 urls contain tags, so that the IDF score is .207608. Finally, the total score of the first link is 1.063321.

5 Experiments and Analysis

The experiments are performed as follows:

- Initially, submit the query to Google, and obtain the original Google search results.
- Now, submit the Google search results to CiteUlike to obtain the relevant tags.
- Re-rank the search results according to our algorithm.
- Compare the Google results with our algorithm.

5.1 Data Set

5.1.1 Query Set

Initially, we determine the queries which we input to the search engine. We determine a total of ten queries. The queries are a combination of keywords and tags. These queries are submitted to Google. We have chosen academic domain as CiteUlike provides tags for the academic database only

5.1.2 Result Set

Now, submit each query to Google and record the first 50 results. This way, the result set of 10 queries become 500 results.

5.1.3 Results Tag Set

Now, we submit the 500 results to CiteUlike and the resulting tag vector is recorded. We obtain lots of tag values for a result.

We have chosen the following queries.

Q1. {Books, artificial intelligence}	Q6. {Research papers, software engineering}
Q2. {Books, data mining}	Q7. {Pdf, genetic algorithm}
Q3. {Research papers, data mining}	Q8. {Research papers, statistics}
Q4. {Pdf, information retrieval}	Q9. {Books, web mining}
Q5. {Research papers, semantic web}	Q10. {Books, java programming}

For these queries, we compute the values of normalized DCG gains for Google as well as for our algorithm (Fuzzy JEKS algorithm) in Table1.

Table 1. Normalized DCG gains of Google and our fuzzy JEKS algorithm.

Query	Google Ranking	Fuzzy JEKS Algorithm Ranking
	nDCG(G)	nDCG(A)
Q1	0.980211	0.959274
Q2	0.896716	0.92342
Q3	0.937156	0.926431
Q4	0.979388	0.978542
Q5	0.987652	0.987472
Q6	0.94898	0.948706
Q7	0.98502	0.98638
Q8	0.91282	0.877635
Q9	0.900639	0.929049
Q10	0.943141	0.936474

We obtained normalized DCG values for the 10 queries for our algorithm as well as for Google results. It can be seen that our algorithm acquires higher values of normalized DCG for 3 queries out of 10 queries when compared to Google.

6 Conclusion

We have proposed a new ranking algorithm based on the previous methods only. We have added fuzziness in this new proposed algorithm. Semantic tag of a web page is the metadata information associated with it and depicts a lot about the information associated with it. Here, we have added fuzzy values with the weights of the tag. The fuzzy values have been associated with the tags by looking at the relevance of the tag with the query.

We have proposed the new algorithm using the already existing semantic web algorithm which basically calculates the weighted score of the tags. We have added fuzziness with the weighted score to improve the semantic web. In experiments, we have collected the data from Citeulike and implemented the above algorithm. The relevant fuzzy scores to the different web links have been given by a group of users. Comparing with Google search results, we find that Fuzzy JEKS algorithm acquires better ranking results for 3 queries out of a total of 10 queries. Our algorithm acquires higher values of normalized DCG for 3 queries out of a total of ten queries when compared to Google.

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. We can enhance the algorithm by adding these effects.

REFERENCES

[1]. Cilibrasi, R.L., Vitanyi, P.M.B., *The Google similarity distance*. Knowledge and Data Engineering, IEEE Transactions on, 2007. 19: p. 370-383.

[2]. *Farooq, U., Kannampallil, T.G., Song, Y., Evaluating Tagging Behaviour in Social Bookmarking Systems: Metrics and design heuristics, Supporting Group Work, 2007. Proceedings. The international ACM Conference on, 2007: p. 351-360.*

[3]. *Gautam, J., Kumar, E., An Improved Framework for Tag-Based Academic Information Sharing and Recommender System, Proceedings. World Congress on Engineering, 2012. 2: p. 845-850. (IAENG, London).*

[4]. Halpin, H., Lavrenko, V., *Relevance feedback between hypertext and Semantic Web search*, Journal of Web Semantics, 2011. 9: p. 474-489.

[5]. *Jiang, H., Hu, X., Li, P., Wang S., An improved method of term weighting for text classification, Intelligent Computing and Intelligent Systems, International Conference on, 2009.1:p. 294-298, IEEE Press.*

[6]. *Jin Y., Lin Z., Lin H., The Research of Search Engine Based on Semantic Web, Intelligent Information Technology Application Workshops (IITAW), 2008. Proceedings. International Symposium on,2008: p. 360-363, IEEE Press.*

[7]. Lee, M., Kim, W., Park, S., Searching *and ranking method of relevant resources by user intention on the Semantic Web,* Expert Systems with Applications,2012. 39: p. 4111- 4121.

[8]. Leung, K.W.T., Lee, D. L., Deriving *concept-based user profiles from search engine logs*. Knowledge and Data Engineering, IEEE Transactions on, 2010. 22: p. 969-982.

[9]. S. Lu, X. Li, S. Bai , S. Wang., *An improved approach to weighting terms in text*. Journal of Chinese Information Processing, 2000. 14: p. 8-13.

[10]. *Shenliang, X., Shenghua, B., Fei, B., Exploring Folksonomy for Personalized Search, Research and Development in information retrieval, the 31st annual international ACM SIGIR conference on,2008, p. 155-162. ACM, USA*

[11]. Yoo, D., *Hybrid Query Processing for Personalized Information Retrieval on the Semantic Web*, Knowledge-Based Systems,2012. 27: p.211-218.

[12]. *Wu, X., Zhang, L., Yu Y., Exploring Social Annotations for the Semantic Web, World Wide Web (WWW 06), the 15th International Conference on, 2006, p. 417-426, ACM Press, USA.*

[13]. *Zhanguo, M., Jing, F., Liang, C., Xiangyi H., Yanqin, S., An improved approach to terms weighting in text classification, Computer and Management, the International Conference on, 2011, p. 1-4, IEEE Press.*

[14]. *Zhang, C., Yan, M., Trust Network Based Fuzzy Knowledge Sharing, Computational Intelligence and Software Engineering, 2009, Proceedings, 2009 International Conference on, p. 1-5, IEEE Press.*

[15]. *Zhao, C., Zhang, Z., A New Keywords Method to Improve Web Search, High Performance Computing and Communications, 2010, Proceedings, 2010 International Conference on, p. 477-484, IEEE Press*

[16]. Zhong, N., Li, Y., Wu. S.T, *Effective Pattern Discovery for Text Mining*. Knowledge and Data Engineering, IEEE Transactions on, 2012. 24: p. 30-44.

[17]. "tf-idf," Wikipedia, http://en.wikipedia.org/wiki/(accessed June 2013).

[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).