Similarity Measures of web pages using Cosine Similarity

Prof.Uma Goradiya
Shree L.R.Tiwari college of Enginerring,
MumbaiUniverisity
umanashte@gmail.com

Prof.Bhavin Goradiya
ALLEN Academy ,
Mumbai Univerisity,
Bgoradiya@gmail.com

ABSTRACT: Scavenging information associated to particular person through search engines is one of the most common activities on the Internet. Result contains numbers of Web pages, which may be relevant to different person which includes queried name. Human languages are not correct. Text referring to the city "Roanoke" can mean "Roanoke, Virginia" or "Roanoke, Texas", depending on the surrounding context. Organizations and companies often have multiple nicknames, name variations, or common misspellings. Famous persons ("Amitabh Bachchan") often share a name with many non-famous individuals. In this paper, we propose a similarity measures system to solve the problem by using cosine similarity which is based on TF and IDF. Web pages having high cosine similarity are club together into one cluster.

Keywords-*Clustering, Web People Search, WePS, Social Network Analysis, Web Querying*

*****

I. INTRODUCTION

Web is a popular place for collecting information. The web size is increasing continuously. The more the Internet is growing; the more tendencies the people have to use the search engines. Moreover, since most of the commercial search engines are based on keyword indexing, there are many records in their result lists that are irrelevant to the user's information needs. Internet users access billions of web pages online using search engines.

It is shown that for retrieving more relevant and precise results, the following two points should be concerned: First of all, the query (either it is generated by a human or an intelligent agent) should be expressed in an accurate and exact manner. Second, we should empower search engines with the ability to capture the semantic relation between the words and the query context.

In the area of professional search where users usually spend comparatively more time on investigating a larger portion of the search result.

This problem motivated researchers to help people by following two different strategies, Changing the infrastructure of the current web to the semantic web or Placing the keyword based search engines as the base and doing some modifications to considering the query and web page context in order to improve their efficiency.

There was a big problem over the realization of the first idea. The problem was that there were already millions of millions documents in current web that should apply considerable modifications in their structure to express their content in RDF and RDFS. That's why our proposed architecture follows the second strategy. The goal of Similarity Measures of web pages using Cosine Similarity is to find similarity between web pages based on extracted entities. For finding Cosine similarity between web pages we extract entities for each URL by using alchemy API and then find TF-IDF for each entity and every URL

II. LITERATURE REVIEW

Many different approaches have been applied to the basic problem of disambiguation of people and document ranking they are as follows.

1. Using dependency structure for prioritization of functional test suites [5] in this paper, they proposed a new test case prioritization technique that uses the dependency information from the test suites to prioritize. Dependency structure prioritization technique includes four algorithms for prioritizing. The open dependency proves to have lower execution cost and closed dependency achieved better fault rate detection than the traditional methods. Average rate of fault detection is used to calculate the percentage of fault rate but clustering approach is not considered to improve the fault rate further.

2.Enhanced distributed document clustering algorithm using different similarity measures[6] in this paper, a distributed environment is considered in which all peers form a ring structure and the information are stored in DHT. A local model is formed using EDKmeans using similarity algorithm. All local models aggregated to form a global model using EPCP2P this improves clustering quality and accuracy. Even though Jaccard and Pearson coefficients show better results than cosine similarity in data mining but this is not possible in case of software testing.

3. Information retrieval (IR) aims at retrieving documents that are relevant to a user's information needs. To be able to effectively present the retrieved documents to the user, the probability ranking principle (PRP) states that [9]:"If an IR system's response to each query is a ranking of documents in orderof decreasing probability of relevance, the overall effectiveness of the system toits user will be maximized."

Information retrieval (IR) aims at retrieving documents that are relevant to a user's information needs. To be able to effectively present the retrieved documents to the user, the probability ranking principle (PRP) states that [9]:"If an IR system's response to each query is a ranking of documents in orderof decreasing probability of relevance, the overall effectiveness of the system toits user will be maximized."

III. APPROACH OVERVIEW

A.AlchemyAPI:
It utilizes natural language processing technology and machine learning algorithms to analyze content, extracting semantic meta-data: information about people, places, companies, topics, facts & relationships, authors, languages.API endpoints are provided for performing content analysis on Internet-accessible web pages, posted HTML or text content.To use AlchemyAPI, you need an access key. Output is in XML format.

B.TF: It's a kind of Document Vector. This scheme assigns a weight to each term (vocabulary word) in a given document. The weight increases proportional to the number of times the term occurs in the document.

C.TF-IDF: In contrast to plain Boolean retrieval where -in principle - only the presence of terms in documents needs to be recorded in the index, a term can also be assigned a weight that expresses its importance for a particular document. A commonly used term weighting method is tf-idf, which assigns a high weight to a term, if it occurs frequently in the document but rarely in the whole document collection.

D.Steps of the approach

INPUT=person name
OUTPUT=Cosine Similarity between web pages

Algorithm
1. User Input. The user issues a query via the input interface.
2. Top-K Retrieval. The system (middleware) sends a query consisting of a person name to a search engine, such as Google and retrieves the top-K returned web pages. This is a standard step performed by most of the current Web search engine. Get top k web page by using Google API GwebSearchClient().
3. Extract content of each web page by using Alchemy API.
4. For each web page extract location and organization using alchemy API. Store location and organization name for all pages in one auxiliary database.
5. For each entity in database calculate IDF=log10 (D/d)
D=total no of documents
d=no of documents containing word.
6. For each web page for all entity in database calculate TF=(no of times entity occurring in page / total words in page)
7. For each web page for all entity in database calculate TF-IDF=TF*IDF.
8. For K web pages
For i=1 to k-1
For j=i+1 to k;
S=compute cosine similarity (di,dj);
di,dj are web pages.

The resulting similarity ranges from −1 meaning exactly opposite, to 1 meaning exactly the same, with 0 usually indicating independence, and in-between values indicating intermediate similarity or dissimilarity.
A1=TF-IDF for first entity in database for 1st page
B1=TF-IDF for first entity in database for 2nd page

IV.RESULTS

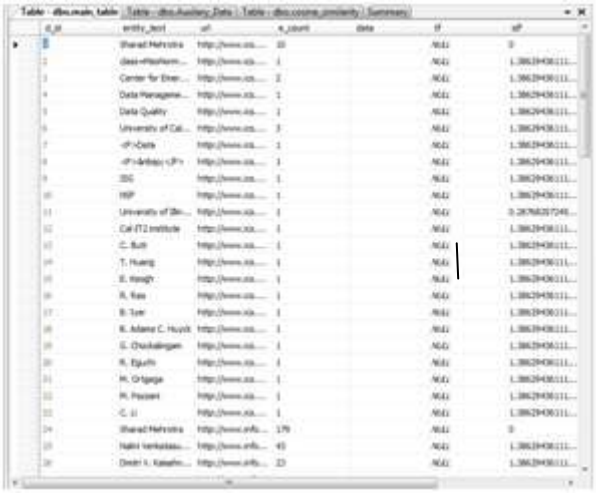

Fig 1:Term Frequency

This table stores extracted entities and its TF for each URL.

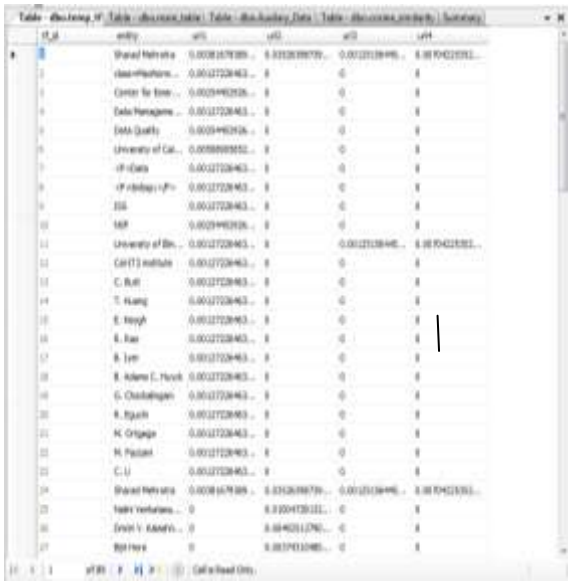

Fig.2: Inverse Document Frequency

Extracted entities Person name and organisation using alchemy API for all URL content and its IDF is stored in this table.

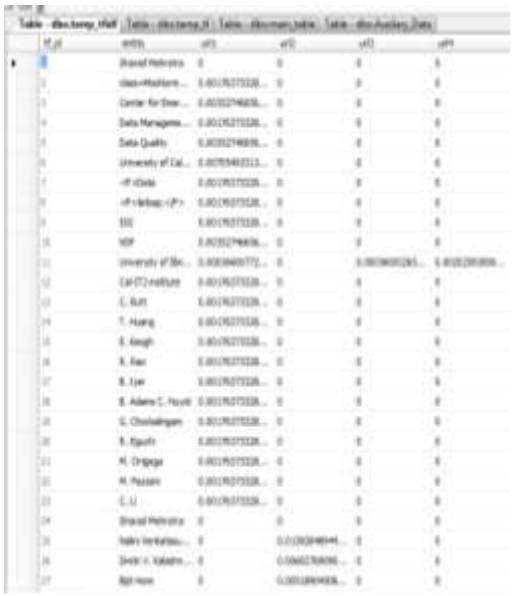

Fig 3. TfF-IDF

Table stores TFIDF for each entity and each URL.
Once we have a weight(TF and IDF) calculated for all the terms, we actually have a kind of *matrix*, or table, for our URL content, then we multiply temp_tf table with IDF column of main table to get TF-IDF for each entity and every URL. Here we get most entries are zero, because any given document will only contain a small percentage of all the words we've encountered in a large collection.

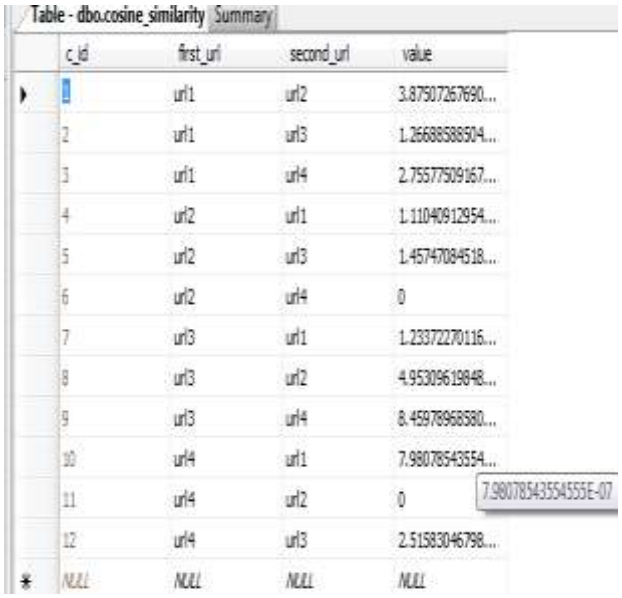

Table - dbo.cosine_similarity | Summary

c_id	first_url	second_url	value
1	url1	url2	3.87507267690...
2	url1	url3	1.26688588504...
3	url1	url4	2.75577509167...
4	url2	url1	1.11040912954...
5	url2	url3	1.45747084518...
6	url2	url4	0
7	url3	url1	1.23372270116...
8	url3	url2	4.95309619848...
9	url3	url4	8.45978968380...
10	url4	url1	7.98078543554...
11	url4	url2	0
12	url4	url3	2.51583046798...
NULL	NULL	NULL	NULL

7.98078543554555E-07

Fig 4: Cosine similarity between pages

Table shows Cosine similarity between URL. Cosine similarity is calculated by applying cosine similarity formula on temp_tfidf table. Here we are getting very low cosine similarity which is nearly equal to zero.

V. CONCLUSION AND FUTURE WORK

The proposed system initially extracts the entity that are people and organization using Alchemy API. Then find out TFIDF for each entity for each URL which is used to calculate similarity between two documents. We are getting very low cosine similarity between web pages.
To this end the web people search system that was able to accurately find the similarity between web pages and rank the given number of document according to information content.

As future work we plan to develop a new solution that would utilize the plethora of other features available in the data, including hyperlinks, emails, phone number, and so on.

REFERENCES

[1] NajimDehak, Patrick Kenny, RdaDehak, Pi.erreOuellet, and Pierre Dumouchel, “Front end Factor Analysis for Speaker Verification,” submitted to IEEE Transactions on Audio, Speech and Language Processing, 2010.

[2] Stephen Hin-Chung Shum, NajimDehak, RedaDehak, and Jim Glass, “Unsupervised Speaker Adaptation based on the Cosine Similarity for Text-Independent Speaker Verification,” Submitted to ODYSSEY, 2010.

[3] Ryan Carlson, Hyunsook Do, Anne Denton “a Clustering Approach to Improving Test Case prioritization”: An Industrial Case Study,‘ 27th IEEE International Conference on Software Maintenance 2011.

[4] K.P.N.V.Satyasree ,Dr.JVMurthy "clustering based on cosine similarity measure,' international journal of engineering science and advanced technology 2012.

[5] ShifaZehraHaidry and Tim Miller "Using Dependency Structures for Prioritization of Functional Test suites", IEEE transaction on software engineering.2013.

[6] Neethi Narayanan, J.E.Judith,Dr.J.Jayakumari "Enhanced Distributed Document Clustering Algorithm Using Different
Similarity Measures",IEEE Conference on Information and Communication Technologies.2013.

[7] VikasThada, Dr VivekJaglan, "Comparison of Jaccard, Dice, Cosine Similarity Coefficient To Find Best Fitness Value for Web Retrieved Documents Using Genetic Algorithm",International Journal of Innovations in Engineering and Technology.2013.

[8] Dmitri V. Kalashnikov SharadMehrotra,Exploiting, "Web querying for Web People Search in WePS2".

[9] Jun Wang, "Mean-Variance Analysis: A New Document Ranking Theory in Information Retrieval",ECIR 2009, LNCS 5478, pp. 4–16, 2009.