Scavenge Web Querying for Web People Search

Prof.Uma Goradiya
Shree L.R.Tiwari college of Enginerring,Mumbai Univerisity
Thane,Maharashta,India.

Prof.Garima Mishra
Shree L.R.Tiwari college of Enginerring, Mumbai Univerisity,
Thane,Maharashtra,India

Prof.Bhavin Goradiya
Rao IIT Academy ,Mumbai Univerisity,
Mumbai,Maharashta,India

Abstract - Primary interest of this paper is automatic identification of web presence of particular people. The problem appears to have a trivial solution when the name of the person is unique we can just goggle this person's name. However, the problem complexity grows significantly with the level of commonness of the personal name. Indeed, given a common name such as Tom Mitchell we find hundreds of different people called Tom Mitchell. The task of disambiguating and finding the Web Pages related to the specific person of interest is left to the user.
In this paper, we propose a Web Appearance Disambiguation (WAD) system to solve the problem by using TF/IDF similarity and Web querying. The approach is based on extracting named entities from the web pages by using Alchemy API and then uses TF/IDF similarity between web pages and querying the web to collecting co-occurrence statistics, which are used as additional similarity measures.

General Terms-Algorithms, Experimentation, Measurement

Keywords-Clustering, Web People Search, WePS, Named Entity Web Co-Occurrences, Social Network Analysis, Web Querying

I. INTRODUCTION

The rapid growth of the Internet has made the Web a popular place for collecting information. The web size is increasing continuously. Chinese researchers have shown that web growth obeys the Moor's law and the Internet will double in size every 5.32 years. 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.

Finding information about people using search engines is one of the most common activities on the Web. However, search engines usually return a long list of Web pages, which may be relevant to distinct namesakes who have the queried name, especially given the explosive growth of Web data. Human language is not exact. 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 ("Michael Jackson") often share a name with many non-famous individuals.

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 typical Web search setting it is important to users that the most relevant documents are top-ranked from given large number of potentially relevant documents. In the area of professional search where users usually spend comparatively more time on investigating a larger portion of the search result .

II. LITERATURE REVIEW

To be able to effectively present the retrieved documents to the user, the probability ranking principle (PRP) states that "If an IR system's response to each query is a ranking of documents in order of decreasing probability of relevance, the overall effectiveness of the system to its user will be maximized."[6]. Experimental results on the collaborative filtering problem confirms the theoretical insights with improved recommendation performance, e.g., achieved over 300% performance gain over the PRP-based ranking on the user-based recommendation.

The BIR is based on Boolean Logic and classical Set theory in that both the documents to be searched and the user's query are conceived as sets of terms. Retrieval is based on whether or not the documents contain the query terms. One of the issues with the Boolean search model is that results are unranked - every matching document for a query contains all of the terms in that query, and there's no real way of saying which are 'better'. However, if we could weight the terms in a document based on how representative they were of the document as a whole, we could order our results by the ones that were the best match for the query.

The vector space model has many advantages over the Standard Boolean model: and documents. Simple model based on linear algebra. Term weights not binary. It allows computing a continuous degree of similarity between queries. [7]

In Query History for ranking they Combining query results means that we merge the results from q1, ..., qk by taking an average of the rank of each document [8]. Such a method would reward a document that has been ranked high by all the queries. It is very difficult to maintain query history .It can be happen that there is no query history present for current query.

This problem motivated researchers to help people by following two different strategies,

1. Changing the infrastructure of the current web to the semantic web. 2. 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 [9].
That's why our proposed architecture follows the second strategy.

The goal of Scavenge web querying for Web People search system is to find similarity between web pages based on extracted entities and rank given number of documents based on term frequency of 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.

III. APPROACH OVERVIEW

A. Steps of the approach

1)COSINE SIMILARITY BETWEEN WEB PAGES:

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

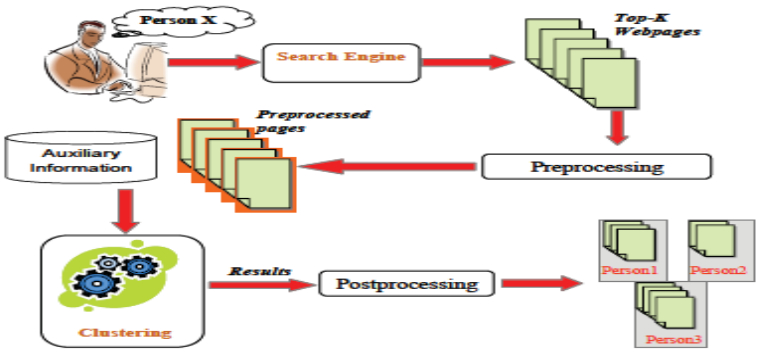

Fig1:Web processing steps

2)SORTING DOCUMENTS ACCORDING TO INFORMATION CONTENT:

This is second module which rank the given number of document related to particular person based on information content.

Input=number of documents related to particular person.
Output=number of documents in ranked order according to information content.
Algorithm:
1.Provide number of document related to person and in text box provide person name and click on search button.
2.Read content of each file using ReadFile() function and store in separate string.
3.Call wordcount function to count number of word in each string.
4. Set alchemy API key by using SetAPIKey( ) function.
5. Extract name and organization in query by using TextGetRankedNamedEntities( ) function.
6.Count number of occurance for name and organiztion in each file using InstanceCount() Function.
7.Claculate term frequencey of person and organization for each file. Using Instancecount/wordcount.
8. For each file add Term frequencey of name and organization and store it in databae.
9. Display the content of database according ti decnding value of total TF and we get document in ranked order.

B:RESULTS

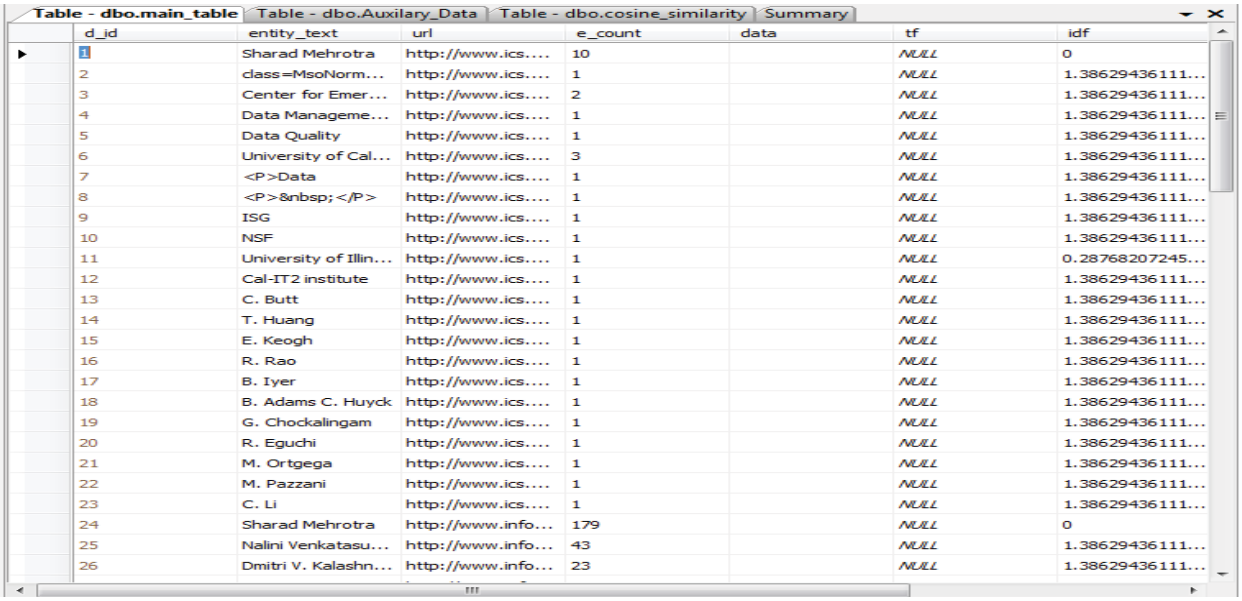

Table - dbo.main_table | Table - dbo.Auxilary_Data | Table - dbo.cosine_similarity | Summary

d_id	entity_text	url	e_count	data	tf	idf
1	Sharad Mehrotra	http://www.ics....	10		NULL	0
2	class=MsoNorm...	http://www.ics....	1		NULL	1.38629436111...
3	Center for Emer...	http://www.ics....	2		NULL	1.38629436111...
4	Data Manageme...	http://www.ics....	1		NULL	1.38629436111...
5	Data Quality	http://www.ics....	1		NULL	1.38629436111...
6	University of Cal...	http://www.ics....	3		NULL	1.38629436111...
7	<P>Data	http://www.ics....	1		NULL	1.38629436111...
8	<P> </P>	http://www.ics....	1		NULL	1.38629436111...
9	ISG	http://www.ics....	1		NULL	1.38629436111...
10	NSF	http://www.ics....	1		NULL	1.38629436111...
11	University of Illin...	http://www.ics....	1		NULL	0.28768207245...
12	Cal-IT2 institute	http://www.ics....	1		NULL	1.38629436111...
13	C. Butt	http://www.ics....	1		NULL	1.38629436111...
14	T. Huang	http://www.ics....	1		NULL	1.38629436111...
15	E. Keogh	http://www.ics....	1		NULL	1.38629436111...
16	R. Rao	http://www.ics....	1		NULL	1.38629436111...
17	B. Iyer	http://www.ics....	1		NULL	1.38629436111...
18	B. Adams C. Huyck	http://www.ics....	1		NULL	1.38629436111...
19	G. Chockalingam	http://www.ics....	1		NULL	1.38629436111...
20	R. Eguchi	http://www.ics....	1		NULL	1.38629436111...
21	M. Ortgega	http://www.ics....	1		NULL	1.38629436111...
22	M. Pazzani	http://www.ics....	1		NULL	1.38629436111...
23	C. Li	http://www.ics....	1		NULL	1.38629436111...
24	Sharad Mehrotra	http://www.info...	179		NULL	0
25	Nalini Venkatasu...	http://www.info...	43		NULL	1.38629436111...
26	Dmitri V. Kalashn...	http://www.info...	23		NULL	1.38629436111...

Figure 2:Term Frequency

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

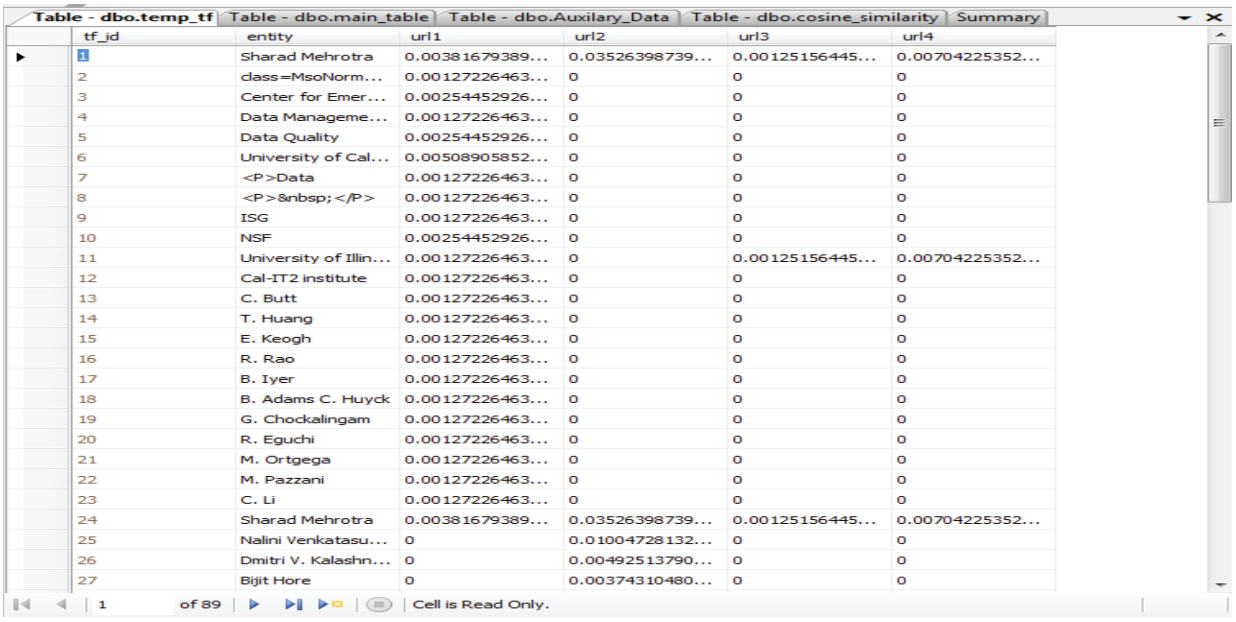

Table - dbo.temp_tf | Table - dbo.main_table | Table - dbo.Auxilary_Data | Table - dbo.cosine_similarity | Summary

tf_id	entity	url1	url2	url3	url4
1	Sharad Mehrotra	0.00381679389...	0.03526398739...	0.00125156445...	0.00704225352...
2	class=MsoNorm...	0.00127226463...	0	0	0
3	Center for Emer...	0.00254452926...	0	0	0
4	Data Manageme...	0.00127226463...	0	0	0
5	Data Quality	0.00254452926...	0	0	0
6	University of Cal...	0.00508905852...	0	0	0
7	<P>Data	0.00127226463...	0	0	0
8	<P> </P>	0.00127226463...	0	0	0
9	ISG	0.00127226463...	0	0	0
10	NSF	0.00254452926...	0	0	0
11	University of Illin...	0.00127226463...	0	0.00125156445...	0.00704225352...
12	Cal-IT2 institute	0.00127226463...	0	0	0
13	C. Butt	0.00127226463...	0	0	0
14	T. Huang	0.00127226463...	0	0	0
15	E. Keogh	0.00127226463...	0	0	0
16	R. Rao	0.00127226463...	0	0	0
17	B. Iyer	0.00127226463...	0	0	0
18	B. Adams C. Huyck	0.00127226463...	0	0	0
19	G. Chockalingam	0.00127226463...	0	0	0
20	R. Eguchi	0.00127226463...	0	0	0
21	M. Ortgega	0.00127226463...	0	0	0
22	M. Pazzani	0.00127226463...	0	0	0
23	C. Li	0.00127226463...	0	0	0
24	Sharad Mehrotra	0.00381679389...	0.03526398739...	0.00125156445...	0.00704225352...
25	Nalini Venkatasu...	0	0.01004728132...	0	0
26	Dmitri V. Kalashn...	0	0.00492513790...	0	0
27	Bijit Hore	0	0.00374310480...	0	0

1 of 89 | Cell is Read Only.

Fig.3: Inverse Document Frequency

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

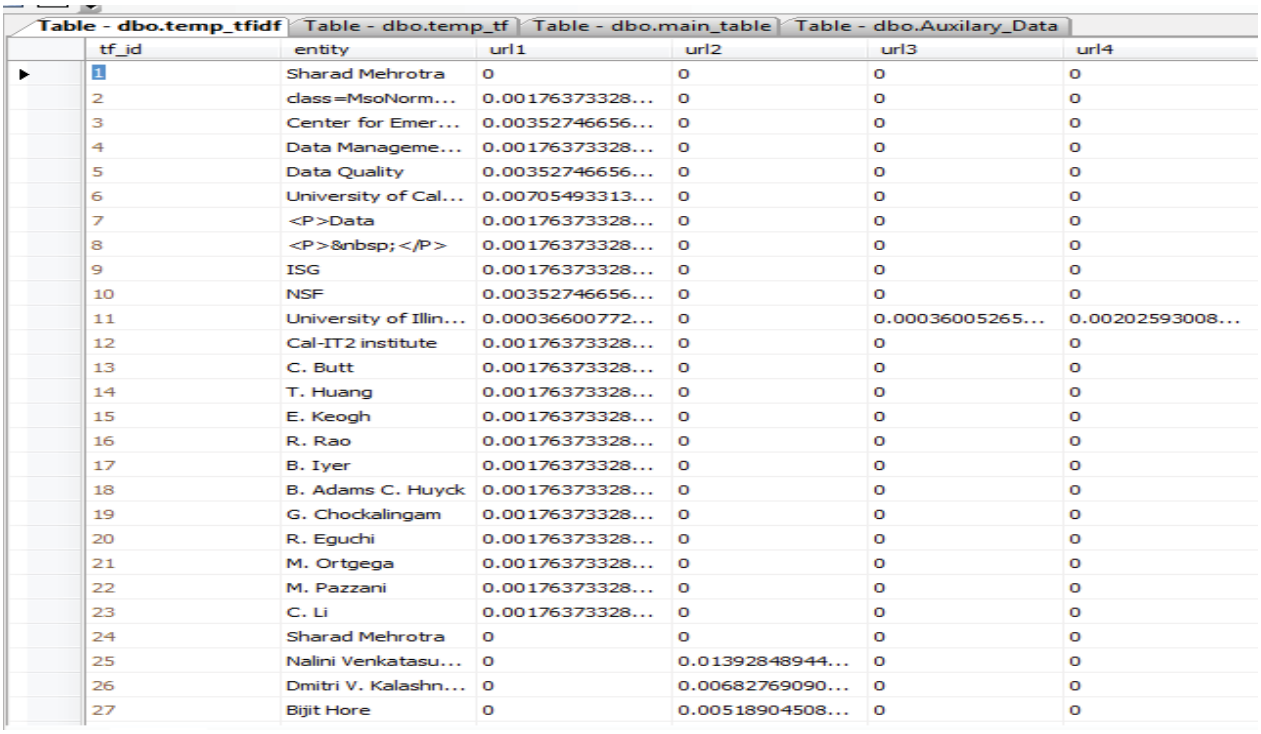

Table - dbo.temp_tfidf | Table - dbo.temp_tf | Table - dbo.main_table | Table - dbo.Auxilary_Data

tf_id	entity	url1	url2	url3	url4
1	Sharad Mehrotra	0	0	0	0
2	class=MsoNorm...	0.00176373328...	0	0	0
3	Center for Emer...	0.00352746656...	0	0	0
4	Data Manageme...	0.00176373328...	0	0	0
5	Data Quality	0.00352746656...	0	0	0
6	University of Cal...	0.00705493313...	0	0	0
7	<P>Data	0.00176373328...	0	0	0
8	<P> </P>	0.00176373328...	0	0	0
9	ISG	0.00176373328...	0	0	0
10	NSF	0.00352746656...	0	0	0
11	University of Illin...	0.00036600772...	0	0.00036005265...	0.00202593008...
12	Cal-IT2 institute	0.00176373328...	0	0	0
13	C. Butt	0.00176373328...	0	0	0
14	T. Huang	0.00176373328...	0	0	0
15	E. Keogh	0.00176373328...	0	0	0
16	R. Rao	0.00176373328...	0	0	0
17	B. Iyer	0.00176373328...	0	0	0
18	B. Adams C. Huyck	0.00176373328...	0	0	0
19	G. Chockalingam	0.00176373328...	0	0	0
20	R. Eguchi	0.00176373328...	0	0	0
21	M. Ortgega	0.00176373328...	0	0	0
22	M. Pazzani	0.00176373328...	0	0	0
23	C. Li	0.00176373328...	0	0	0
24	Sharad Mehrotra	0	0	0	0
25	Nalini Venkatasu...	0	0.01392848944...	0	0
26	Dmitri V. Kalashn...	0	0.00682769090...	0	0
27	Bijit Hore	0	0.00518904508...	0	0

Fig 4:TF-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.

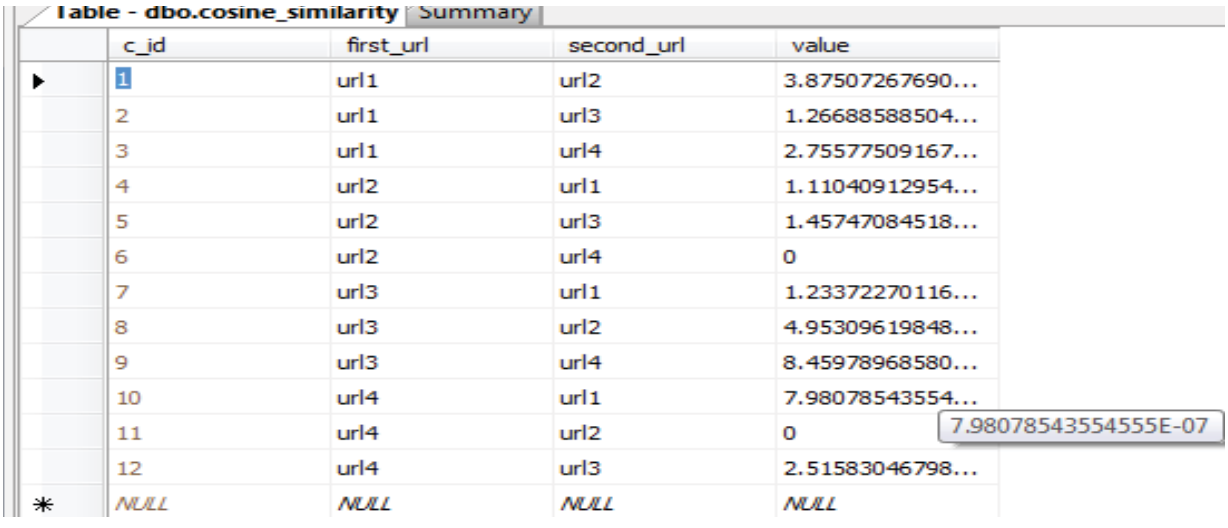

Table - dbo.cosine_similarity | Summary

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	8.45978968580...
10	url4	url1	7.98078543554...
11	url4	url2	0
12	url4	url3	2.51583046798...
NULL	NULL	NULL	NULL

7.98078543554555E-07

Fig 5:Cosine similarity between pages

Figure 5 shows Cosine similarity between URL. Cosine similarity is calculated by applying cosine similarity formula on TF_IDF table. Here we are getting very low cosine similarity which is nearly equal to zero thats why we cannot decide threshold for clustering process.

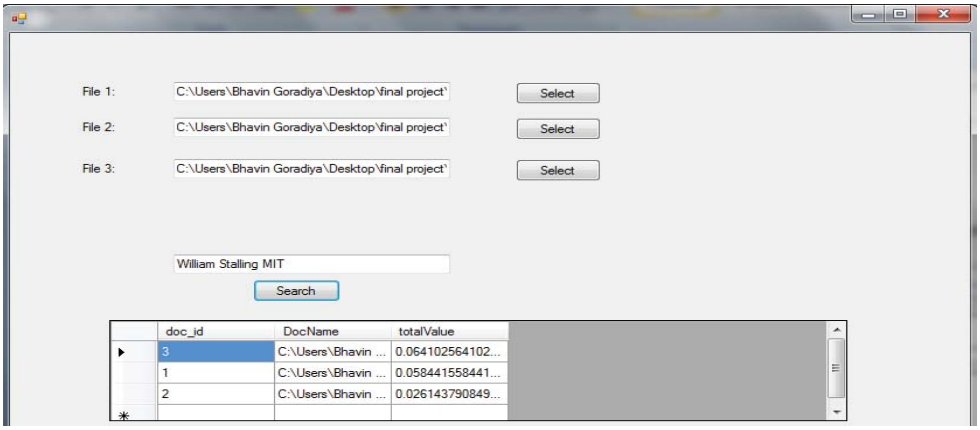

Fig 6:Ranked Document:

Here we get documents in ranked order according to decreasing value of total TF.

IV. CONCLUSION AND FUTURE WORK

In this we focused on the problem of disambiguate a personal name in web search results. To solve this problem, we have proposed a new method to measure the similarities between documents that is cosine similarity and we rank given number of document related to particular person based on term frequency of entities.

The proposed system find out cosine similarity which is based on TFIDF. The proposed system initially extracts the entity that are people and organization using Alchemy API. Then find out TF-IDF 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. The other module rank the given number of document based information content. To measure information it make use of Term Frequency of word provided in query for each document.
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] J. G. Javier Artiles and S. Sekine. Weps 2 evaluation campaign: overview of the web people search clustering task. In 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference, April 2009.
[2] D. V. Kalashnikov, Z. Chen, S. Mehrotra, and R. Nuray. Web people search via connection analysis. IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE), 20(11), Nov. 2008.
[3] D. V. Kalashnikov, Z. Chen, R. Nuray-Turan, S. Mehrotra, and Z. Zhang. WEST: Modern technologies for Web People Search. In Proc. of the 25th IEEE Int˘Sl Conference on Data Engineering (IEEE ICDE 2009), Shanghai, China, March 29 - April 4 2009. demo publication.
[4] D. V. Kalashnikov, S. Mehrotra, S. Chen, R. Nuray, and N. Ashish. Disambiguation algorithm for people search on the web. In Proc. of the IEEE
[5] D. V. Kalashnikov, R. Nuray-Turan, and S. Mehrotra. Towards breaking the quality curse. A web-querying approach to Web People Search. In Proc. of Annual International ACM SIGIR Conference, Singapore, July 20–24 2008. April 9–12 2007.
[6] Jun Wang, “Mean-Variance Analysis: A New Document Ranking Theory in Information Retrieval”,ECIR 2009, LNCS 5478, pp. 4–16, 2009.
[7] DIK L. LEE, “Document Ranking and the Vector-Space Model”, MARCH/APRIL 1997 0740-7459/97/$10.00 © 1997 IEEE 6 7
[8] Xuehua Shen, ChengXiang Zhai, “Exploiting Query History for Document Ranking in Interactive Information Retrieval”, SIGIR’03, July 28–August 1, 2003.
[9] http://www.rdfabout.com/intro/