Web Search Engines A Basis For Term Evaluation

Sukanya Manna and Tom Gedeon

Abstract

We present an approach for the evaluation of a term significance model, Gain Of Words (GOW) using two different web search engines (WSEs) : Google[1] and Yahoo[2]. Our term association model extracts significant terms from a single document without using a corpus. We extract the significant terms from the Gain of Words (GOW) parameters' values based on a sentence level analysis of the document. The experiment is modeled by three sets of queries for the search engines, as single queries, paired queries and triplet queries. The evaluation is based on the documents retrieved from the WSEs using these queries. The results show that the terms or words of higher rank according to this model extract more relevant documents from the WSE as shown through the similarity results with the source documents from where the terms were extracted. We have shown that our model works better than the term frequency inverse sentence frequency (TFISF) model using WSE rankings.

Keywords: term significance, gain of words, single documents, web search engines, evaluation

1 Introduction

Extraction of significant terms is an important technique for retrieval methods such as document retrieval, web page retrieval, text mining, information extraction, summarization and so on. These terms are also called content identifiers as they are used to choose which document to read to learn the appropriate relationship among documents.

The vector based information retrieval model identifies relevant documents by comparing query terms with terms from a document corpus. This is done by assigning the highest weights to the ones with most discriminative power Salton & Buckley (1987). Inverse Document Frequency is the most common retrieval model which considers the distribution of terms between documents. There are also modifications of the above concept into inverse sentence frequency and inverse term frequency. Inverse sentence frequency similarly reflects the distribution of terms between sentences and inverse term frequency likewise in sentences or phrases Blake (2006).

Almost every document has some hierarchical structure concerning the importance of the words or concepts occurring in it Gedeon & Koczy (1998). The basic idea of linking the terms in a document is based on their frequency of occurring together in different paragraphs or sentences, presuming them to have some relationship. This approach also does not require any previous knowledge about the domain. It is based on the degree of linkages found between different terms and brings out the relevant ones.

Domain independent keyword extraction, which does not require a large corpus, has many applications. For example, if we want to know the basic idea of a paper, the key terms will help us to identify them. For this we do not really need to have a detailed knowledge about the corpus of similar data. Here simply the frequency of the term or the word count is significant enough. It is also known Manna & Gedeon (2008) that sometimes the traditional models like Salton & Buckley (1987) are not that effective in cases when we need to analyze a single document or very few documents instead of a huge corpus as it cannot discriminate certain terms. So, Manna & Gedeon (2008) domain independent term extraction models are thus more suitable to analyse the semantic relations of a document, which might be of interest in some specific applications, for example, in legal cases, or official investigations, or counter terrorism, text summarization Zhang et al. (2002), question answering systems Katz & et al (1993) and so on which specifically deal with single or a few documents.

In this paper we present two studies: first by the term significance model and second by the detailed evaluation of this model using two web search engines (WSE): Google and Yahoo respectively. We extract the terms importance using the value of the factor called Gain of Words (GOW). We select the terms

[1]Google : trademark of Google Inc; http://www.google.com
[2]Yahoo: Yahoo! Pty Limited; http://www.yahoo.com

using a threshold on GOW. We queried the WSE using the high ranked and low ranked terms extracted by our method. The documents retrieved using these queries are compared with the source documents to find out the similarity between them. Then we compared the rank of words using GOW and TFISF using WSE rankings.

This paper is organized as follows. The next section explains the term significance models (GOW and TFISF) in detail, followed by the evaluation done using the WSE.

2 Term Significance Models

In this section we present two approaches, Gain of Words (GOW) and Term frequency Inverse Sentence Frequency (TFISF) to rank the importance of words in a document. TFISF is a modification of TFIDF at the sentence level instead of the document level.

A document consists of sentences. In this paper, a sentence is considered to be a set of words separated by a stop mark (“.”, “!”, “?”). We extracted all the terms from the document including the stop words. We calculated the frequency (i.e. number of occurrences of the words) of the words across each sentence and used it with a binary weighting factor to determine GOW.

2.1 Gain Of Words (GOW):

In this work, we propose a term significance model which extracts significant terms from a single document. This model is more effective than Salton’s Salton (1975) approach when considered for a single documents instead of a large corpus Manna & Gedeon (2008). The term significance is analyzed based on the concept Gain of Words (GOW). The main purpose of this method is to discriminate between the significant and non significant terms (or words). This is a simple model, which employs both the occurrences of words as well as the binary weighting of those words based on their presence or absence in the document. In order to proceed with this model, we have initially found all the unique words from the document. Then we stemmed these words using Porter’s stemmer Porter (1997).
Now, let n be the no. of words or terms considered. Let S be the vector of m sentences present in the document. So, we calculate the gain of words for each term by

$$GOW_j = \frac{\sum_{i=1}^{m} f_{ij} \times \sum_{i=1}^{m} w_{ij}}{m} \tag{1}$$

where, f_{ij} is the frequency of the term (no. of occurrences) j in the sentence i and w is the weight. We obtain weights by

$$w = \begin{cases} 1 & \text{if the term is present in the sentence} \\ 0 & \text{otherwise} \end{cases}$$

Words having high GOW values are discarded. We used a threshold of $0 < GOW < 10$ for the elimination process. It is our experience that the words beyond this range are generally not useful. The stopwords or some unwanted words or characters generally fall in this category.

2.2 Analysis of GOW in general:

We have used one of the CST Radev et al. (2004) datasets about a Milan plane crash. It consists of some single coherent documents. Each of these documents contain some facts about the plane crash collected from different news media like abcnews, cnn, etc.

We used our method on this dataset to compute GOW for each of these documents individually. We ranked the words based on their GOW in the document, and applied a threshold to extract the significant ones. After this, the word list was further processed by removing those infrequent stopwords which remained in the list. The Fig. 1. below gives an example of the nature of GOW of words. It is clearly seen that high values show unimportant words (here the stopwords like “in”, “the”). Using the threshold we can get rid of these easily. The terms with lower GOW like “milan”, “plane”, “crash” etc show that they are significant terms if we look into the context of the document considered.

2.3 Term Frequency Inverse Sentence Frequency (TFISF):

Term frequency inverse sentence frequency is the sentence level modification of the commonly used corpus weighting scheme term frequency inverse document frequency (TFIDF) proposed by Salton Salton & Buckley (1988) which is term frequency (TF) x inverse document frequency (IDF), where TF is the

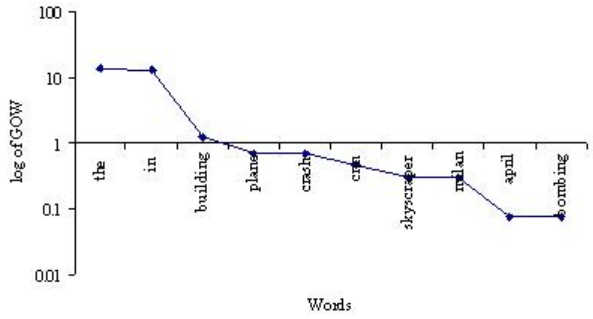

Figure 1: Illustration of GOW on different words

number of times a term appears in a document, and IDF reflects the distribution of terms within the corpus. Ideally, the system should assign the highest weights to terms with the most discriminative power. One component of the corpus weight is the language model used. The most common language model is the Inverse Document Frequency (IDF), which considers the distribution of terms between documents. So, here ISF similarly considers the distribution of terms between sentences of a document. Thus it is represented by

$$ISF(t_i) = log(N) - log(n_i) + 1 \tag{2}$$

where N is the total number of sentences in a document; n_i is the number of sentences that contain at least one occurrence of the term t_i; and t_i is a term, which is typically stemmed. So, the weight of each term is determined by term frequency (TF) x inverse sentence frequency (ISF). As we intend to find the term weight at the document level, we need some kind of aggregation on the tfisf values. But instead, we found the total frequency of each term across each sentence and multiplied by their corresponding isf score.

3 Analysis of GOW using Web Search Engines:

3.1 Motivation for using Google and Yahoo:

The sudden explosion of digital data on web has created an online repository of data for experimental purposes. Google and Yahoo are the most used web search engines. Starting from a trifling matter to an important one, people use either of these search engines to get their answers. Because of their popularity we also used these WSE search results as a basis for validating the term significance model described in Sect. 2.

This experiment is to demonstrate the suitability of the algorithm for keyword extraction from single documents. Our claim as to which words are significant will be justified if these WSE search results provide us with more similar documents related to the original one if we use higher ranked words we extracted. So we used a simple model to find the similarity between these documents with the original one. We queried both Google and Yahoo with single keywords from our ranked list as well as paired keywords. It is clear that for a single term query, different kinds of documents will be returned from different subject areas. When two words are paired from the same source document (and hence more likely to be from the same subject domain) are used as a search string, the results are a bit more specific than the single search query.

3.2 Method overview:

We extracted the first ten and last ten words from each of the documents from the dataset based on their *GOW* when arranged in descending order. Then we peformed the following tasks for each single document as shown in the pseudocode:

```
for each KEYWORD
    1. Query WSE with the KEYWORD
    2. Get weblinks from the webpages
    returned
```

```
    for each of these weblinks
        3. Download the corresponding
        webpages
        for each of these webpages
            4. Parse them and generate
            their wordlist
            5. Compute similarity with
            original wordlist
        end for
        6. Calculate average of the values
        of similarity
    end for
end for
```

This is done for both the first set of words in the ranked list, as well as the last set of words. The KEYWORD mentioned in the pseudocode is either a single term or the terms in pairs or the triplet of those terms taken from the sets of first and last ten words of the document concerned respectively. For the single term query, we directly used the selected words as WSE search queries. For the paired query, we found all 45 combinations of the ten words from each set separately, and used two words at a time as the WSE search string. But for triplets, we used 15 out of all the possible combinations of the triplet to perform a small scale experiment to see how it varied from the single and paired queries. For each query, we obtain a search page with the desired number of results. We then extracted web site addresses from these results. The corresponding webpages are downloaded using the weblinks we extracted from the WSE search results. These webpages are considered as the retrieved or found documents which are then compared with the source documents (the document from which we extracted the single and paired terms to carry out this experiment) to get the similarity.

The documents we obtained after the query were all processed by eliminating stop words, followed by stemming, and we generated a unique word list for each of these documents.

3.3 Similarity measure for the documents:

In Sect. 2, we explained how we chose the wordlist from the source documents based on their GOW. Now, in order to find the similarity between the original document and the corresponding found documents, we compute a simple similarity model. Our basic consideration was to find how many words of the found document matched with the original one. We used the binary match of the source word vector with the (found document) target word vector to calculate the matches between the two documents.

Let WL_{ori} be the processed wordlist (a set of words) of a single original document having n number of words in it. Let WL_{ret} be another processed wordlist (a set of words) obtained from the found pages having m number of words. So, we calculate the similarity (SIM) by,

$$SIM(ori, ret) = \frac{|(WL_{ori} \cap WL_{ret})|}{|(WL_{ori})|} \tag{3}$$

, where $|(WL_{ori} \cap WL_{ret})|$ is the number of words of the found document matching with the number of words in the original document and $|(WL_{ori})|$ is the number of words in the original document. We ignore the size of WL_{ret}, and consider only the n words of WL_{ori}.

The pseudocode below shows how we have performed this similarity calculation for a single original document with all the found documents:

```
R = Total no. of found docs. for that original doc
    for each found doc
        1. calculte SIM(ori,ret)
        2. SUM+=SIM(ori,ret)
    end for
3. AVERAGE = SUM / R
```

Here, SIM is our similarity measure, SUM is the summation of the similarity value for all the found documents with its corresponding original document and AVG is the mean of all these SIM values respectively. This is computed for all the original documents individually.

Table 1: Number of documents collected from WSE by single query string for each document (G = Google Y = Yahoo)

No. of weblinks	WSE	No. of docs
15	Y and G	$2 \times 300 = 600$
30	Y and G	$2 \times 600 = 1200$
50	only Y	1000

Table 2: Number of documents collected from WSE by paired query string for each document (G = Google Y = Yahoo)

No. of weblinks	WSE	No. of docs
15	Y and G	$2 \times 1350 = 2700$
30	Y and G	$2 \times 2700 = 5400$
50	only Y	4500

4 Experimental results:

4.1 Data set:

This work is based on the analysis of single documents. We have collected the original documents from a CST Radev et al. (2004) data set which comprises of 9 documents (D1, D2,...,D9) respectivly. These documents are related to a Milan plane crash. In this part of the experiment, we have used Google and Yahoo as a web repository of documents to establish the significance of this model. We retrieved large numbers of documents from each of these search engines. As mentioned in Sect. 3, we used single keywords, paired keywords and triplets in WSE as search string.

Keyword selection and data collection: We created three sets of data. Out of this, two of them are studied in detail; one for single queries and another for the paired queries. The third one is a small data set for the triplet queries. Again, we further subdivided each of these into two more sets, one related to the first ten terms of each document and the other related to the last ten.

For the **single** queries for each document, we directly input the terms into the WSE one at a time, and collected the top NUM weblinks from each of them. Then we downloaded those web pages as our experimental data. The total number of documents collected per original document for both sets = $2 \times 10 \times NUM$, where NUM is the number of weblinks considered each time for the experiment. For Google and Yahoo, we have chosen NUM to be 15 and 30 respectively. Since, Yahoo supports more documents to be retrieved, we used NUM to be 50 for a separate set of experiment with the Yahoo found documents. The table 1 shows the number of documents collected for single search string query.

Hence, the *total documents for whole data set for single query string* $= 9 \times (600+1200+1000) = 25200$.

For, the **paired** queries, for each set of ten words (first and last) we found all possible combination of the words. So, ${}^{10}C_2 = 45$ query strings for each set respectively. Like in the previous case, for each of these 45 queries, we again collected NUM weblinks each, where NUM is the number of weblinks considered each time for the experiment. The total number of documents collected per original document for both sets $= 2 \times 45 \times NUM$. The table 2 shows the number of documents collected for paired string queries.

Hence, *total number of documents collected for the whole data set using paired query string* $= 9 \times (2700 + 5400 + 4500) = 113400$.

Data processing: In the previous sections, we mentioned how we processed the data from the documents to generate unique wordlists for each document. We removed the stop words, removed unnecessary charaters of length one and two and then stemmed the remaining words using Porter's stemmer. For each source document, we computed the similarity with the documents found using the WSE individually for each group of NUM weblinks respectively. The pseudocode for the similarity was shown in the previous section. Here all the experimental results are based on the average similarity of the documents considered for each of the case studies.

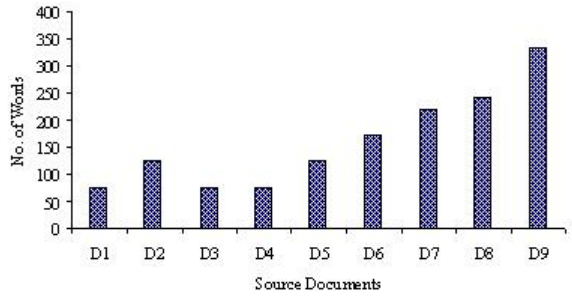

Figure 2: Number of words per document

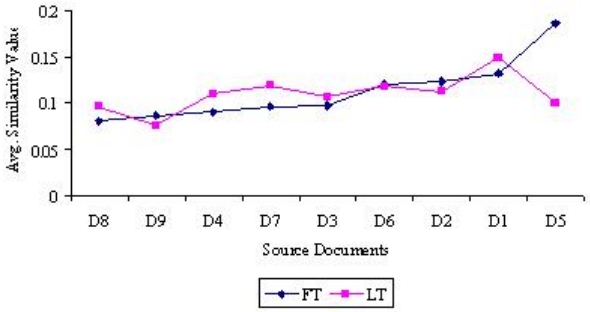

Figure 3: Average Similarity Comparison for Google Search Results for Single Query

The Fig. 2 represents the distribution of the number of significant words in each of the documents considered for this experiment.

4.2 Results:

In this subsection, we present the mean similarity value comparison of all the found documents with respect to their original source. In all the figures, FT is the abbreviation for "First Ten" and LT is the abbreviation for the "Last Ten" which represents the average similarity between the documents using the FT KEYWORDS and LT KEYWORDS respectively.

4.2.1 Query Results with 15 found documents for each query:

In this section, we present the query results with 15 found documents for each query. We present both nature of similarity obtained using the Google and Yahoo each time.

Single Query: Fig. 3 shows the average similarity comparison of the Google found documents with the source documents for single query words and fig. 4 for Yahoo. In fig. 3, we can see that for documents D8, D4, D7, D3, D2 and D1, the LT points are above the FT points, which deviate from our assumption that the FT related documents will be more similar to the original documents. But in fig. 4 we found only documents D4 and D1 did not fit our assumption. For this case the similarity value for the Yahoo found data seem to be consistent than that of Google.

Paired Query: Fig. 5 and fig. 6 show the comparisons between the average similarity values for the first ten and last ten groups of documents obtained by paired string query. The former is the illustration of the Google found documents and the later is Yahoo. In both the cases, we find that the curve for LT lies below FT except for document D4 in fig. 6, the graph for Yahoo, which violates our assumption. Here, Google shows more consistent result than Yahoo. But the similarity values of Yahoo seem to be higher than that of Google.

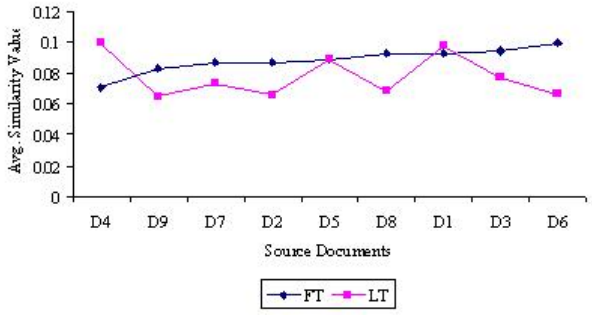

Figure 4: Average Similarity Comparison for Yahoo Search Results for Single Query

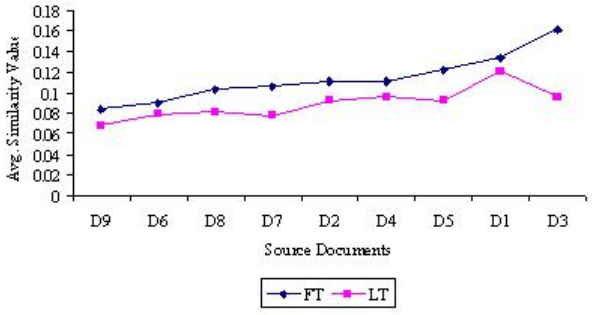

Figure 5: Average Similarity Comparison for Google Search Results for Paired Query

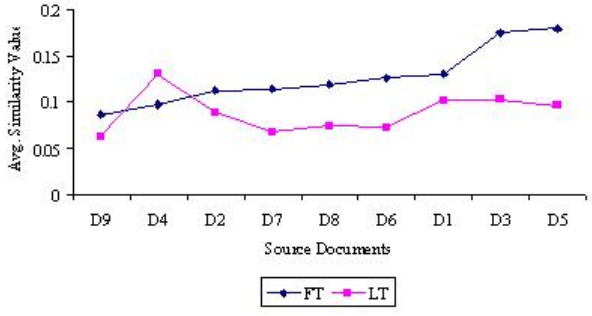

Figure 6: Average Similarity Comparison for Yahoo Search Results for Paired Query

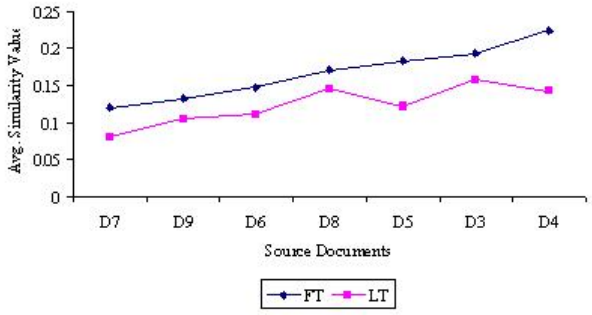

Figure 7: Average Similarity Comparison for Google Search Results for Triplet Query

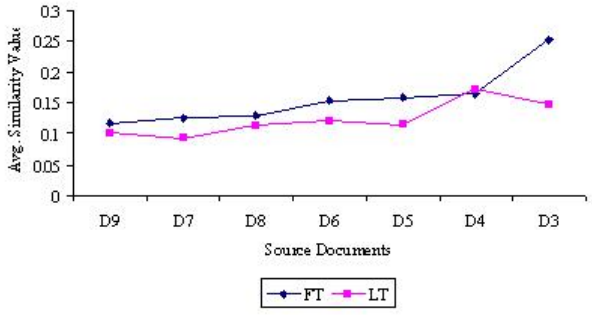

Figure 8: Average Similarity Comparison for Yahoo Search Results for Triplet Query

Triplet Query: Here we present the average similarity value of the of the sets of documents using triplet queries in the search string. The amount of data used here is comparatively less. We used only 15 combinations out of all the possible combinations generated by the top ten words and another similar set with the last ten and also used fewer documents from the data set. The main idea of this part of this experiment is to observe the variation in nature of curves obtained with the increase in queries along with the decrease in the retrieved documents.

The fig. 7 is the average similarity value for the document sets when retrieved from Google using triplet query words. After decreasing the number of queries, as well as the documents, we found the basic nature of the similarity curve remained unaltered. The curve for the first ten is above last ten consistently throughout, showing relevancy in our ranking.

In fig. 8, we find similar results for the Yahoo retrieved data. Here, like the other previous Yahoo results, the D4 on LT curve is different. In this figure also, the pattern of the curve is almost similar, just variation in the average similarity values due to the decrease in the amount of experimental data.

4.2.2 Query Results with 30 found documents for each query:

Like the previous section, we present here the results when 30 documents were found for each queries. Here also, the experiment is done for both Google and Yahoo respectively.

Single Query: Fig. 9 and fig. 10 depict the corresponding average similarity value comparison for single queries for Google and Yahoo found documents respectively. In fig. 9, the points for documents D8, D7, and D4 of the LT curve lies above FT. D2, D3, D6 of LT curve are almost at the same position of FT, which shows that there is not much difference in the results of the two. The distinction between the FT and LT is not that obvious here for Google found documents. Now, for fig. 10, except for D4 and D1 of LT curve, all are below the FT curve. This result is more like our assumption. Here, in this case also, like the 15 found document experiment, the results for yahoo seems to be more consistent for the single query than that of Google.

Paired Query: Fig. 11 and fig. 12 show the comparisons for the paired search string results. The former one is for Google and the later is the Yahoo one. For, fig. 11, the FT curve is above the LT

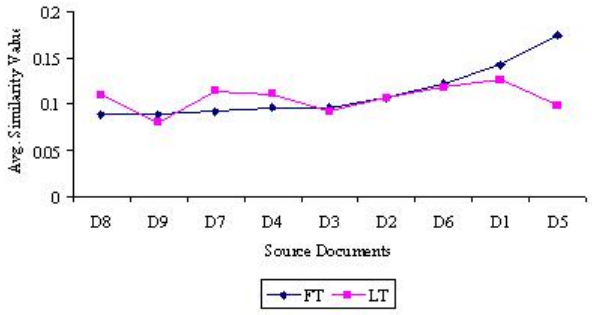

Figure 9: Average Similarity Comparison for Google Search Results for Single Query

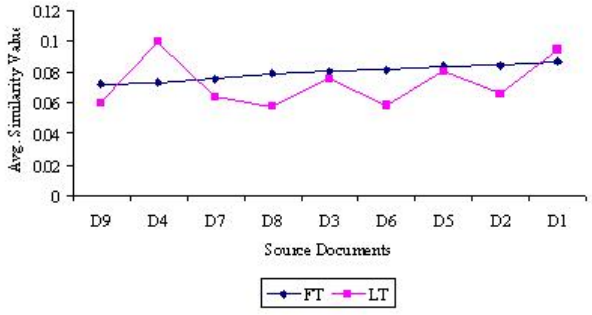

Figure 10: Average Similarity Comparison for Yahoo Search Results for Single Query

curve, which reflects that the documents retrieved using the top ranked words closely resemble the source documents than the last ranked terms. The Google results are consistent and validates our assumption. For fig. 12, expect for the document D4 of the LT curve, all the points lie below the FT curve. The result here is consistent as usual but not as accurate as Google in this case as shown in fig. 11.

Difference in result of two WSE: We have carried out similar experiments with both Google and Yahoo with the same sets of search query strings. The results obtained varies in their consistency and accuracy. The figures 3 to 12 illustrates the results at different condition. It can be clearly seen from the above graphs that the average similarity results for Yahoo is more accurate and consistent than Google for the single search strings. For paired results, however, Google is better as per as accuracy is concerned, though the average similarity values are more for Yahoo then Google. Except for document D4 of LT curve, Yahoo has maintained a consistent result in this case also. In case of triplet query words, Yahoo as well as Google shows a similar pattern with that of the paired results. Here, the triplet has just been introduced with a small amount of data just to provide a test case of variation of query strings and its impact on similarity value. We can not infer the cause of variation provided by these WSE results. But

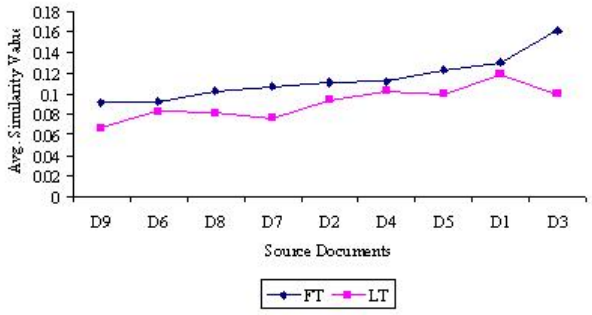

Figure 11: Average Similarity Comparison for Google Search Results for Paired Query

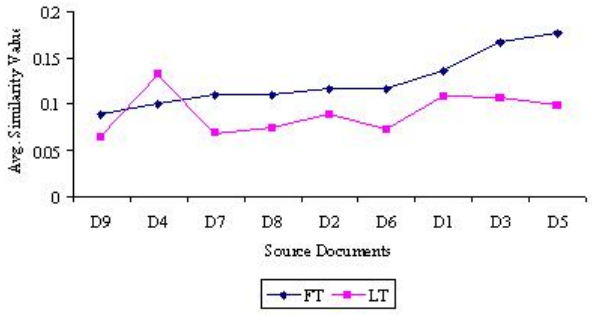

Figure 12: Average Similarity Comparison for Yahoo Search Results for Paired Query

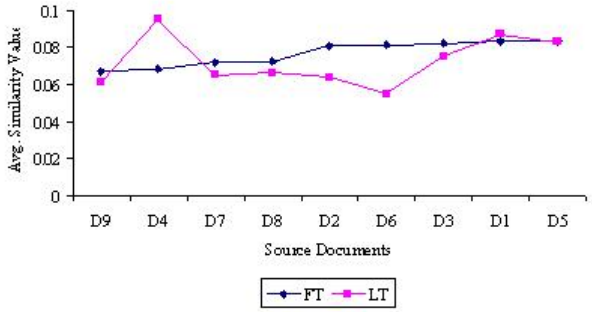

Figure 13: Average Similarity Comparison for Yahoo Search Results for Single Query

we can assume that due to the variation of indexing and ranking techniques the nature of results are different even for the same set of data.

4.2.3 Query Results with 50 found documents for each query:

In this section, we present the results obtained by Yahoo WSE only. Due to restrictions of Google APIs, we were unable to collect the weblinks for more than 30 per query string. This section is mainly included to show the impact of variation of number of documents on the similarity value with the source documents.

Single Query: Fig. 13 represents the results for single search query. Here, except for the documents D4 and D1 of LT curve, all lie below FT curve. It is seen that the result improves with the increase in number of documents. Though in fig. 4 and fig. 10, the documents D4 and D1 of LT curve deviate from our assumption in the same way as in fig. 13, but the overall average similarity value improved. Thus the basic pattern of results did not vary with the increase in number of documents.

Paired Query: Now, fig. 14 shows the result for the paired query string. Here, except for the point D4 on LT curve, all the others are below FT. Like the single query results for this case, here also the patterns of the graphs plotted are not altered with the change in number of documents. Like the previous cases like 15 documents, as well as 30 documents, D4 of LT curve deviated from our usual assumption. Here only the average similarity value is changed with the increase of the documents.

5 Comparison Between Rank of Terms Obtained by GOW and TFISF:

In this section, we compare the ranks obtained using GOW and TFISF respectively. In sect. 4, we have shown our data collection and evaluation of our method using the WSEs. Now, for this part of the analysis, we have also used the web collected data for the evaluation of the ranks found using these two models.

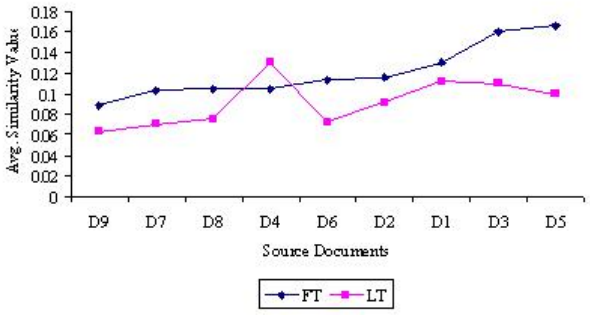

Figure 14: Average Similarity Comparison for Yahoo Search Results for Paired Query

Table 3: Top word ranks by different methods of single query (Columns 2 to 5 represent the ranks):

Words	GOW	TFISF	Y	G
building	1	2	3	7
com	2	3	10	6
plane	3	4	1	9
italy	4	6	4	10
italian	5	7	7	4
pirelli	6	8	9	1
scene	7	9	5	8
skyscrapper	8	10	2	5

5.1 Method in a nutshell:

We have used the web collected data from sect. 4. We prepared six sets of data (three from Google found data and the other three from Yahoo found data), two sets each for single query and paired query for the first ten word ranked list, and the remaining two from the last ten ranked words. Then, we calculated the average similarity of the original document with the found documents for each of the KEYWORDS and used these values to rank the word list. After this, we computed the mean squared error between the WSE obtained rank and term significance models individually for each of these data sets considered.

Results for single query words: We found the mean squared error for both the ranks obtained with Google and Yahoo ranked words with our model GOW higher than TFISF.

Table 3 shows the ranking of words using the different methods, and table 4 shows the mean squared error between pairs of these methods for the top ranked words. It can be seen that that MSE for Yahoo obtained rank (Y) and GOW and Y and TFISF are same indicating that the rank is pretty similar. Again, MSE for Google obtained rank (G) and GOW shows that it is better than G and TSISF. Table 5 on the other hand shows the same MSE for Y, GOW and Y, TFISF. A similar pattern is noticed for the MSE values of G,GOW and G, TFISF. This shows that rank list is same for the bottom ranked words using these two models, GOW and TFISF.

Results for paired query words: In sect. 4, we have shown that the results for paired query terms are better than the single query terms in terms of retrieving the documents from WSEs. So, we have done some simple calculations for getting the ranks of the words from the paired query average similarity results. Initially we had the average similarity value for each of the paired query words (e.g. plane+pirelli, pirelli+skyscraper etc 45 unique word pairs). We ranked them according to their average similarity value

Table 4: Comparison between Mean Squared Error (MSE) for the top ranked words of single query(Y: Yahoo, G: Google):

MSE	Y,GOW	Y,TFISF	G,GOW	G,TFISF
	1.58	1.58	1.91	1.97

Table 5: Comparison between Mean Squared Error (MSE) for the last ranked words of single query(Y: Yahoo, G: Google):

MSE	Y,GOW	Y,TFISF	G,GOW	G,TFISF
	1.32	1.32	1.40	1.40

Table 6: Top word ranks by different methods of paired query (Columns 2 to 5 represent the ranks):

Words	GOW	TFISF	Y	G
building	1	2	6	7
com	2	3	10	9
plane	3	4	2	2
italy	4	6	4	3
italian	5	7	5	8
pirelli	6	8	3	6
scene	7	9	8	4
skyscrapper	8	10	1	1

from 1 to 45. We obtained their individual significance by averaging the ranks of the pairs in which they occur. Thus we obtained a separate rank for the single terms of paired queries.

The table 6 shows the ranking of the words obtained by different methods when considered the data for paired query. The table 7 illustrates the MSE obtained when taken each of the methods at a time. The result for Y and G are consistent. Our method is even better when the retrieval ranks are compared using the WSE, as shown by the MSE.

6 Conclusion:

In this work, we have introduced an approach to evaluate term significance algorithms using web search engines. A model, Gain Of Words (GOW), a term significance model, is presented which works on single documents. It is based on the syntactic appearances of the terms or words in a single document and can be used for extraction of significantly relevant words from the document. Be it for search, or for simply associations between terms, the role of significant terms play a very important role. Here we have also presented another term weighting model called Term Frequency Inverse Sentence Frequency (TFISF), using which terms are extracted and ranked based on their significance and then compared with GOW's results. Ranks of the words produced by GOW is similar but better than TFISF. In order to further validate our term significance model, GOW, we have used two web search engines: Google and Yahoo to retrieve documents based on the first and the last ten words ranked by GOW. The experimental results have shown that the similarity of the search results to the original documents for the first ten words of the documents are better than the last ten. For the single query, Yahoo supports this fact better than Google. But this is particularly clear for paired queries as well as the triplet query results where Google provides more accurate result than Yahoo. Through both of these WSEs, it can be seen that the result supports our assumption that our ranking is meaningful, hence these results indicate that our technique works, as the first ten words are better than the last ten in finding similar documents on the web.

Table 7: Comparison between Mean Squared Error (MSE) for the top ranked words of paired query(Y: Yahoo, G: Google):

MSE	Y,GOW	Y,TFISF	G,GOW	G,TFISF	G,Y
	1.43	1.89	1.65	1.96	1.08

References

Blake, C. (2006), A comparison of document, sentence, and term event spaces, *in* 'ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics', Association for Computational Linguistics, Morristown, NJ, USA, pp. 601–608.

Gedeon, T. D. & Koczy, L. T. (1998), 'Hierarchical co-occurence relations', *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on* **3**, 2750–2755 vol.3.

Katz, B. & et al (1993), 'Start, natural language question answering system'.

Manna, S. & Gedeon, T. (2008), A term association inference model for single documents: A stepping stone for investigation through information extraction., *in* C. C. Yang, H. Chen, M. Chau, K. Chang, S.-D. Lang, P. S. Chen, R. Hsieh, D. Zeng, F.-Y. Wang, K. M. Carley, W. Mao & J. Zhan, eds, 'ISI Workshops', Vol. 5075 of *Lecture Notes in Computer Science*, Springer, pp. 14–20.

Porter, M. F. (1997), 'An algorithm for suffix stripping', *Program* **14**(3), 313–316.
URL: *http://portal.acm.org/citation.cfm?id=275705*

Radev, D., Otterbacher, J. & Zhang, Z. (2004), Cst bank: A corpus for the study of cross-document structural relationships, *in* 'Proceedings of LREC 2004'.

Salton, G. (1975), *Theory of Indexing*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Salton, G. & Buckley, C. (1987), Term weighting approaches in automatic text retrieval, Technical report, Ithaca, NY, USA.

Salton, G. & Buckley, C. (1988), 'Term-weighting approaches in automatic text retrieval', *Inf. Process. Manage.* **24**(5), 513–523.

Zhang, Z., Blair-Goldensohn, S. & Radev, D. R. (2002), Towards cst-enhanced summarization, *in* 'Eighteenth national conference on Artificial intelligence', American Association for Artificial Intelligence, Menlo Park, CA, USA, pp. 439–445.