J. Basic. Appl. Sci. Res., 4(6)215-223, 2014
© 2014, TextRoad Publication
ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

An Improved Similarity Measure for Text Documents

Muhammad Shoaib[1], Ali Daud[2] and Malik Sikandar Hayat Khiyal[3]

[1]PhD Scholar, Faculty of Basic and Applied Sciences, International Islamic University, Islamabad, Pakistan
[2]Assistant Prof., Faculty of Basic and Applied Sciences, International Islamic University, Islamabad, Pakistan
[3]Professor, Faculty of Computer Sciences, Preston University Islamabad, Pakistan

Received: April 14, 2014
Accepted: May 21, 2014

ABSTRACT

In text mining applications such as clustering documents, citation matching and author name disambiguation (AND) similar documents are grouped together by estimating similarity among them in pair wise fashion. Most of similarity functions are relative measures and their output may not be the real picture of the similarity between the documents. In this work we propose an improved similarity measure specially designed for matching terms of two textual documents in pair wise fashion. Our proposed similarity measure tries to depict the picture of the proportion of similarity between the documents. It needs not any information about collection of documents as it is required in vector space based similarity functions. It uses simple count of term frequency as term weights. We compare our proposed measure with state of the art similarity functions. Experiments on synthetic data show that our proposed measure is more logical and realistic than baseline methods.

KEYWORDS: Dice Coefficient, Cosine similarity, Jaccard Coefficient, Information Theoretic, Absolute Similarity, Relative Similarity.

1. INTRODUCTION

In many real life text mining applications such as clustering documents and author name disambiguation (AND) similar documents are grouped together by estimating similarity among them in pair wise fashion. In literature many similarity measures such as Jaccard coefficient, Cosine and Dice coefficient are proposed for the comparison of two publications. Out of similarity measures discussed in literature Cosine similarity is the most popular metric for measuring documents similarity [1, 2]. In document clustering works like [3, 4] and in author name disambiguation works like [5, 6] exploit Cosine measure. Cosine and many other measures use term[1] frequency and inverse document frequency (TFIDF) [7] to weight a term or word, while documents are represented in vector space model (VSM) [8]. Many other methods like repetition based filtering [9], locality sensitive hashing (LSH) method [10], short segment similarity [11] and Earth mover's based similarity [12] propose different methods to estimate documents similarity. A large range of similarity measures exist in literature. Pandit & Gupta [13] give comparative study on distance measuring approaches. Lee et al [14] provide empirical evaluation of models of text document similarity. Teghva and Veni [1] and Strehl et al [15] evaluate effects of similarity metrics on document clustering.

Most of similarity functions are relative measures and their output may not be the real picture of the similarity between the documents. For example, the similarity output "0.5" through Cosine function never means that the documents to be compared have 50% common data. The similarity output is usually a relative value rather than an absolute[2] one. Cosine's output is usually far away from absolute value.

We, in this paper, propose a novel similarity measure and name it as SDK[3] index. SDK index is specially designed for matching key words of two textual documents in pair wise fashion. SDK index tries to depict the picture of proportion of similarity between two documents. In other words, it tries to output similarity between two documents near to the absolute value. Defining similarity function in this way may help us adjusting threshold value for clustering or for any other purpose. For example if we wish to group all those documents which share almost 60 percent data with a particular document then threshold value can be fixed as 0.6. Devising similarity measures with the same concept for all types of attributes in a dataset may help select a single threshold value for all types of attributes. Further, SDK index does not need any information of the collection of documents (number of documents in collection, document frequency, etc.) as it is needed in VSM based similarity measures. It uses simple count of term frequency as term weights. It provides proportional weights to number of common, non common, frequent and

[1] Term and word are used interchangeably in this work.
[2] Here it means: If two documents *a* and *b* share 80% data the output should be 0.8.
[3] SDK (Shoaib, Daud, Khiyal).

***Corresponding Author:** Muhammad Shoaib, PhD Scholar, Faculty of Basic and Applied Sciences, International Islamic University, Islamabad, Pakistan. Ph# +923334569311, email: shoaibishaaqiiu@yahoo.com

rare words assigning logarithmic weights to their frequencies. The reason to give logarithmic weight to the frequency is that if a word occurs 10 times in a document, it is improper to say that it is 10 times important w. r. t. that word.

We compare SDK index with many well known and frequently used similarity measures. We specifically focus Cosine as it is the state of the art document similarity function. We use Cosine in two ways: words weighted by TFIDF (Cos-TFIDF) and by TF[1] (Cos-TF). Cos-TF is used because all other measures are estimated weighting words with TF. Experiments on synthetic data show that our proposed measure is more logical and realistic than the baseline methods as its output is more nearer to the absolute value. We use synthetic data because we want to show trends of different similarity measures in different scenarios. The trends shown in results and discussion section are very difficult, almost impossible, to get from actual datasets.

Rest of the paper is organized as: section 2 is related work. Section 3 is problem statement. Section 4 explains proposed similarity measure. Section 5 briefly describes existing similarity measures used as baseline. Section 6 explains results. Section 7 gives summary and future work. Section 8 is reserved for references used in this work.

2. RELATED WORK

In literature, many similarity measures such as Jaccard coefficient, Manhattan, Euclidean, Pearson correlation, Kullback-Leibler divergence, Chi-square, Dice and Overlap are proposed for comparison of two documents. Out of similarity measures discussed in literature Cosine is the most popular for measuring documents similarity [1, 2, 3, 4, 16]. In document clustering works like [3, 4, 15] and in author name disambiguation works like [5, 6, 17, 18, 19, 20, 21] exploit Cosine measure representing the documents in VSM [8]. Pandit & Gupta [13] give comparative study on distance measuring approaches. Lee et al [14] provide empirical evaluation of models of text document similarity. Teghva and Veni [1] and Strehl et al [15] evaluate effects of similarity metrics on document clustering, and Shoaib and Daud in their unpublished work[2] give brief overview of similarity measures used in AND.

Strehl et al [15] use YAHOO datasets already categorized by human experts in different categories. In order to evaluate different similarity measures they perform several different clustering algorithms exploiting four different similarity measures (Euclidean, Cosine, Pearson correlation and extended Jaccard). The experiments show that extended Jaccard and Cosine measures are very close to human performed results [2].

Many works in document clustering like [2, 15] and in author name disambiguation such as [22, 23, 24, 25] use topical information [26] to group similar documents. Donald et al [11] evaluate similarity measures that exploit topical like information present in documents. Rafi and Sheikh [2] propose a similarity measure based on topic maps representation of documents. Wan [4] proposes document similarity measure based on the earth mover's distance. These both works try to find subtopics similarity in documents. Our work is different from them in a sense that we focus keyword matching.

The work of Shoaib et al (unpublished)[3] matches to our work. They propose four different similarity measures for academic publications keeping in view that the output should be near to the absolute value. Their first two measures deal with the name entities; third and fourth measures are designed for short and long documents. Their third measure gives absolute similarity output provided that the documents do not repeat any term. Their fourth measure named as Shoaib index tries to output near to absolute value.

In this work we propose SDK index that provides proportional weights to number of common, non common, frequent and rare words assigning logarithmic weights to their frequencies. We are concerned to estimate syntactic similarity only, and not the semantic similarity. Syntactic similarity methods (such as Cosine, Jaccard) are those which compare two documents blindly and are unaware of the context and semantics of the word used. On the other hand, semantic similarity approaches such as topic modeling methods [26, 27] and WordNet based [28] approaches are aware of the meanings and context of the word used.

3. PROBLEM STATEMENT

Many similarity measures specially Cosine, the most used, do not output similarity value of two documents in the percentage of common data to the total data. Many available measures such as Dice coefficient give proportional

[1] Term Frequency
[2] Shoaib, M. and A. Daud, Author Name Disambiguation in Bibleographic Databases, a Survey. Frontiers of Computer Science, Submitted.
[3] Shoaib M., A. Daud and M. S. H. Khiyal, Improving Similarity Metrics for Publications with Special Focus on Athor Name Disambiguation. Arabian Journal of Science and Engineering, submitted

results if there are no repeating words in two documents. There is a problem how to handle the frequency of words. To weigh a word equal to its frequency is illogical and to ignore the frequency at all is also unrealistic. Logarithm is commonly used to weigh the frequent words. It gives relatively more weights to smaller numbers and vice versa. Measures such as Dice, Jaccard and Information Theoretic all use logarithms to give so called proper term weight. For these measures this weight can be considered proper when frequency difference of a word in two documents is a small number, say less than 10, but if this number is more than 100 then the logarithmic weight difference may not be proper.

Cosine similarity produces unrealistic outputs in some situations. For example it gives similarity value "1" between two documents if they have some common words and no non common words irrespective of their frequencies in both documents. Cosine also shows reverse trend of similarity on changing frequency of common words while remaining the non common words and their frequencies unchanged. It increases the similarity values when the difference in the frequencies of common words between two documents goes on increasing while remaining the non common words and their frequencies unchanged. Here we attempt to devise a measure to streamline the documents similarity giving appropriate weights to non-repeating, repeating, common and non common words.

4. EXISTING SIMILARITY MEASURES

Similarity functions such as Cosine, Dice, Jaccard base on VSM [8]. In VSM documents are represented as vectors of documents. A vector similarity function is used to compute the similarity between vectors. The weight $w_{d,t}$ associated with term t in any document d is calculated by $tf_{d,t}$ X idf_t, where $tf_{d,t}$ and idf_t are defined as follows:

- $tf_{d,t}$: the frequency of term t in document d.
- idf_t: $\log(N/d_t)$, where N is the total number of documents in collection and d_t is the number of documents containing term t.

Now we briefly describe the similarity functions as that we have used as base line to compare our SDK index.

4.1. Cosine Measure

Cosine similarity is the most popular measure [12] for estimating document similarity based on VSM. The similarity between two documents a and b can be defined as the normalized inner product of the two corresponding vectors **a** and **b**[1].

$$Sim_{cos}(a,b) = \frac{\mathbf{a}.\mathbf{b}}{|\mathbf{a}|\mathrm{X}|\mathbf{b}|} = \frac{\sum_{t\epsilon(\mathrm{a}\cap\mathrm{b})}(w_{a,t} \times w_{b,t})}{\sqrt{\sum_{t\epsilon a} w_{a,t}^2 \times \sum_{t\epsilon b} w_{b,t}^2}} \ldots \quad (1)$$

Where $(\mathrm{a} \cap \mathrm{b})$ represents common terms of documents a and b; $w_{a,t}$ and $w_{b,t}$ are the weights of term t in documents a and b respectively.

4.2. Dice Measure

Dice similarity measure can be defined as[2]:

$$Sim_{dice}(a,b) = \frac{2 \times \sum_{t\epsilon(\mathrm{a}\cap\mathrm{b})}(w_{a,t} \times w_{b,t})}{\sum_{t\epsilon a} w_{a,t}^2 + \sum_{t\epsilon b} w_{b,t}^2} \ldots \quad (2)$$

All symbols mean the same as they are in Cosine measure.

4.3. Jaccard Measure

The Jaccard similarity measure can be defined as follows:

$$\boldsymbol{Sim_{jacc}(a,b)} = \frac{\sum_{\boldsymbol{t\epsilon(\mathrm{a}\cap\mathrm{b})}}(\boldsymbol{w_{a,t} \times w_{b,t}})}{\sum_{\boldsymbol{t\epsilon a}} \boldsymbol{w_{a,t}^2} + \sum_{\boldsymbol{t\epsilon b}} \boldsymbol{w_{b,t}^2} - \sum_{\boldsymbol{t\epsilon(\mathrm{a}\cap\mathrm{b})}}(\boldsymbol{w_{a,t} \times w_{b,t}})} \ldots \quad (3)$$

All symbols mean the same as they are in Cosine measure.

4.4. Information Theoretic

Aslam and Frost [29] develop an information-theoretic measure for pair-wise document similarity and is given as follows:

$$Sim_{IT}(a,b) = \frac{2 \times \sum_t \min\{P_{a,t}, P_{b,t}\} \log \pi(t)}{\sum_t P_{a,t} \log \pi(t) + \sum_t P_{b,t} \log \pi(t)} \ldots \quad (4)$$

[1] Bold face letters represent vector form of a document.

2 There exist different formats of Dice and Jaccard measures. We took the definitions of these measures from the work of Xiaojun Wan [12].

In above equation, the probability $\pi(t)$ is the fraction of documents containing term *t* in the whole collection. $P_{a,t}$ and $P_{b,t}$ are the fractional occurrences of term *t* in documents *a* and *b* respectively.

5. PROPOSED SIMILARITY MEASURE (SDK INDEX)

In this section before going to SDK index we explain the concept of absolute similarity value.

5.1. Defining Absolute Similarity Value

How do we estimate that a document is near to or far from absolute value? This question can be answered in two different ways depending upon the nature of data. (1) If the documents do not have any repeating word then the answer is very simple. *It is the ratio between the number of common and number of total words or number of total unique words*. (2) If documents have multiple repeating words then the answer is not as simple as in above case. It is agreed upon that weighting a word equal to its frequency or ignoring the frequency at all is illogical and unrealistic [7]. Most of the similarity measures use logarithms to assign so called proper weight to term frequency. Logarithms assign comparatively more weight to less frequent words and vice versa. It is realistic to some extent because occurring of a word may be more important than its frequency greater than 1. Similarly little difference in frequency should have minute effect. These are ok with logarithmic weighting schemes. What is not ok with logarithmic weights? Suppose a word occurs in two documents *a* and *b* 100 and 200 times respectively. Logarithms of these numbers are 2 and 2.30. There is a great difference in frequency but little difference in logarithmic weight. Such scenarios are not ok with simple logarithmic weights. We can say that the output of similarity measures using simple logarithmic weighting methods may be near to absolute value when frequency difference is small; but on the other hand, when frequency difference is large then simple logarithmic weighting schemes may lead to far away from absolute values. From above discussion it is clear that defining absolute similarity value for documents having multiple repeating words with varying frequencies is very difficult. In rough words we can say that a similarity measure that justifies more and more the above scenarios is more nearer to absolute value. In spite of flaws in logarithmic weighting schemes we use them but in different styles to optimize their effects. In SDK index we have added sum of logarithmic squares of differences in frequencies of common words to the denominator of Shoaib index.

5.2. SDK Index

Here we propose and explain our novel document similarity measure. We derive SDK index by extending Shoaib index (unpublished)[1] shown in equation 5.

$$\mathrm{Sim}_{shoaib}(a,b) = \frac{\sum_{t_x \epsilon u}\left(\frac{1}{1+\log(\max(f_{t_x}(a,b))/\min(f_{t_x}(a,b))}\right)}{u+0.5*u'+(\sum_{t_y \epsilon u'} log(f_{t_y}(a,b))} \ldots \qquad (5)$$

Where $u = \{a \cap b)\}$ and $u' = \{a \cup b) - (p_i \cap b)$; $maxf_{t_x}(a,b)$ is the maximum frequency of term t_x in document a or b; $minf_{t_x}(a,b)$ is the minimum frequency of term t_x in document a or b; and $f_{t_y}(a,b)$ is the frequency of term t_y in document a or b.

Shoaib index, like other measures, is less effective for higher difference in term frequencies. To show comparatively more effect we add the sum of logarithmic squares of differences in frequencies of common words to the denominator of Shoaib index. We name this measure as SDK index and it is defined in equation 6.

$$\mathrm{Sim}_{sdk}(a,b) = \frac{\sum_{t_x \epsilon u}\left(\frac{1}{1+log(max(f_{t_x}(a,b))/min(f_{t_x}(a,b))}\right)}{u+0.5*u'+\sum_{t_x \epsilon u}(log(maxf_{t_x}(a,b) - minf_{t_x}(a,b)))^2+(\sum_{t_y \epsilon u'} log(f_{t_y}(a,b))} \ldots \qquad (6)$$

All symbols in SDK index are same as in Shoaib index. SDK index provides proportional weights closer to the absolute value than other measures. Like Shoaib index SDK index also needs not any information about collection of documents. In other words it is independent of the number of documents in collection. It needs just the information of the two documents to be compared.

When two documents *a* and *b* do not have any term frequency greater than 1 then all measures (discussed in this work), except Cosine and Jaccard are absolute measures. On the other hand, when frequency of some words is greater than 1 then they behave differently.

[1] Shoaib M., A. Daud and M. S. H. Khiyal, Improving Similarity Metrics for Publications with Special Focus on Athor Name Disambiguation. Arabian Journal of Science and Engineering, submitted

SDK index is basically designed for textual documents. It can also be applied for entity names where variations in names are minimal. It is not a better solution for entity names (e.g., co-authors of a publication) where a name has variant forms especially when a name has multiple tokens (parts). For example "Muhammad Shoaib Kamboh" can be written in many ways like: "M. S. Kamboh", M. Shoaib Kamboh, etc. SDK considers each variant form of a token as different term.

6. RESULTS AND DISCUSSION

Here we consider synthetic examples and data to prove that our proposed measure is more logical and realistic than base lines methods. We show different trends of similarity outputs by varying inputs in a sequential style. To have such analysis on original data is very difficult (perhaps impossible). We compare SDK index with Shoaib index, Cos-TFIDF, Cos-TF, Dice coefficient, Jaccard coefficient and Information Theoretic. For all measures except Cos-TFIDF we weigh the term frequency by log (TF). To avoid the possibility of log (0) case we weigh as log (1+TF). We implemented these measures in MS Excel 2007. We perform stemming and stop words removal as preprocessing steps to all the documents in synthetic datasets. We compare and explain the behavior of different similarity functions especially focusing Cosine measure in following four scenarios.

6.1. Scenario I

Effect of frequency difference of common words when there is no non common word: We take two synthetic documents *a* and *b* having a single common word between them and no non common word. We go on increasing the frequency of (common) word in document *b* from 1 to 20 remaining the document *a* unchanged. Figure 1 illustrates trend of different similarity functions for this scenario.

Figure 1 shows that the output of Cos-TFIDF and Cos-TF is unrealistic for this scenario because they are not affected by the frequency difference of common words. All other measures show logical trend as they decrease the similarity values when the frequency ratio of common words in both documents goes beyond 1. SDK index curve is more affected than other measures for all values of frequency difference of common words. Thus it can be considered more nearer to the absolute similarity value.

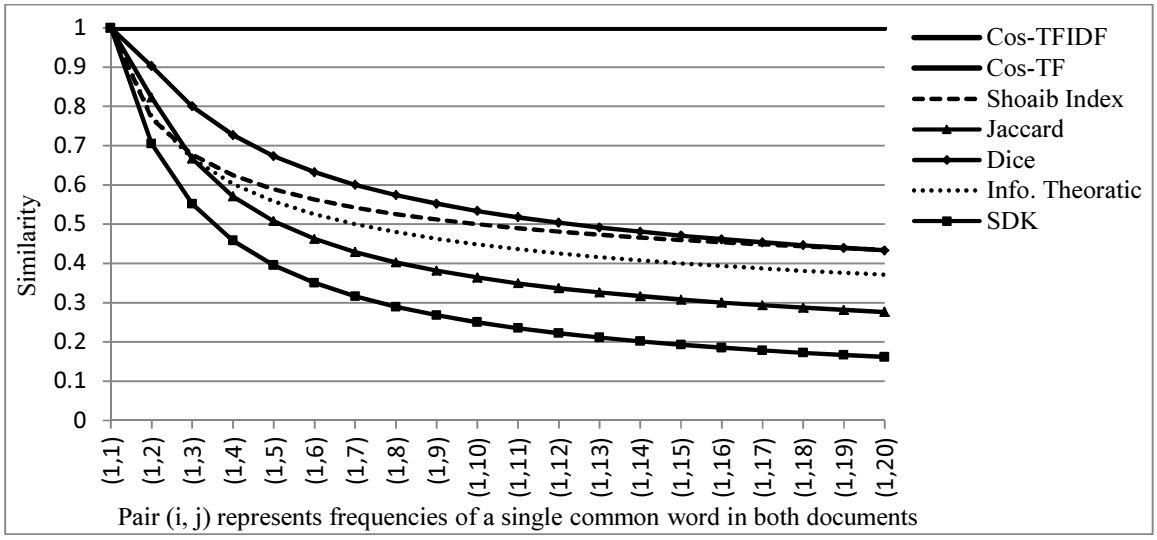

Figure 1: Effect of frequency difference of common words when there is no non common word

6.2. Scenario II

Effect of number of non common words: We take two documents having two common words and, initially, no non common word. We go on increasing the number of non common words each having frequency 1 in both documents alternatively. Figure 2 demonstrates the effect of number of non common words.

Figure 2 shows that Cos-TFIDF is much affected for smaller values of number of non common words and little (negligible small) for such larger values. SDK index, Shoaib index, Dice and Information Theocratic all are same in

above scenario. Cos-TF differs from these measures but the difference is negligible small[1]. SDK index, Shoaib index, Dice and Information Theocratic assign proportional weight to common and non common words. It is clear that when number of non common words (data) changes from 0 to 16 (from 0% to 80%) the similarity value changes with the same ratio (percentage). For example similarity value is 0.5 when percentage between common and non common data is 50. Cosine curve does not have this beauty. Jaccard's output is in between SDK index and Cos-TFIDF.

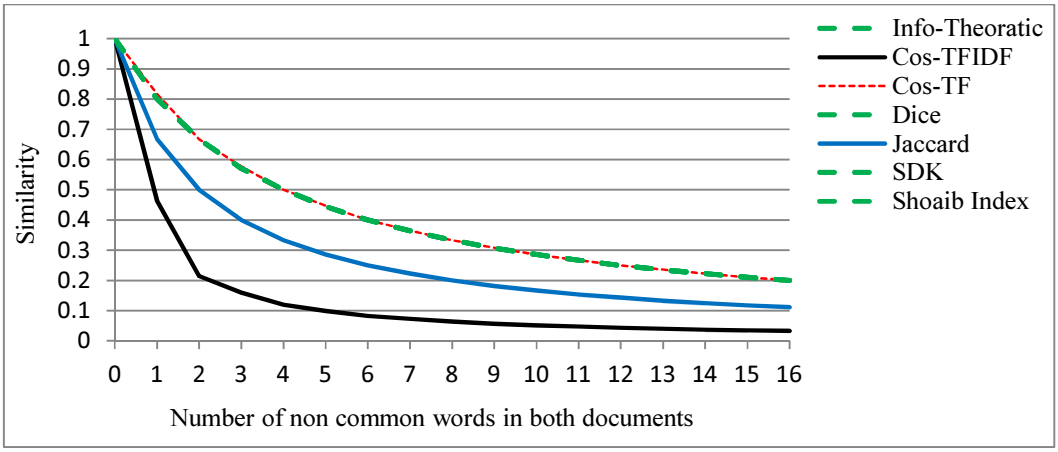

Figure 2: Effect of number non common words

6.3. Scenario III

Effect of frequency difference of common words when non common words also exist: We take two documents *a* and *b* having ten common and six non common words, initially all words having frequency 1. Each time we increase the frequency of each common word in document *b* by 1 without changing document *a* and non common words. Out of six non common words, three are in document *a* and three in *b*. Figure 3 depicts this scenario. This scenario is different from scenario I. In scenario I documents *a* and *b* have no non common words. In this scenario documents also have non common words.

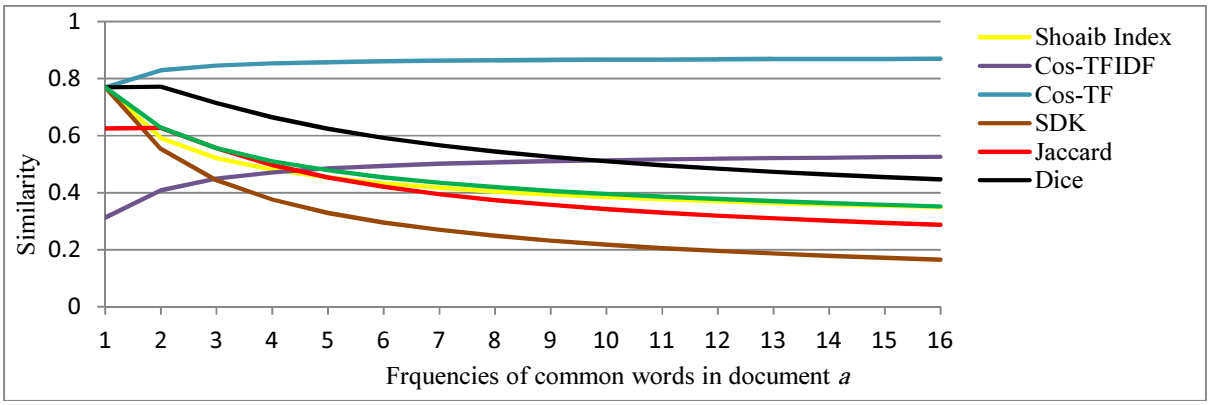

Figure 3: Effect of frequency difference of common words when non common words also exist.

Figure 3 shows that Cos-TFIDF and Cos-TF curves show positive (unrealistic) trend while, realistically, it should be negative. In above scenario, for higher frequency difference of common words without any change in non common words similarity values should be lower. All other measures in figure 3 follow negative (realistic) trend. Jaccard and Dice coefficients initially (from 1-2 on y-axis) show unrealistic trend[2] but for all other values they show realistic trend. SDK index is the lowest but never reaches to zero. Being the lowest is not guarantee to be more

[1] For example, when *a* and *b* have two common words, and *a* has one non common word, this difference is 0.016496581.
[2] For frequency 1-2: 0.625 to 0.627175473 for Jaccard, and 0.769231 to 0.770876 for Dice.

realistic. Actually SDK index assigns reasonable weights to initial values and it is also comparatively more fare for higher values than other measures. We can say that SDK index is more realistic and nearer to absolute value.

6.4. Scenario IV

Effect of existence of non common words either in both documents or only in one document: this scenario is elaborated in following two cases.

- All non common words are evenly distributed in both documents (e.g., if *a* and *b* have 10 non common words: 5 are in document *a* and five are in *b*. Here we ignore their frequency for simplicity)
- All non common words exist only in document *a* or *b*.

Figure 4 depicts this scenario. Figure 4 is drawn considering the data of figure 2. Here we investigate the behavior of different measures. In figure 4, Cos TFIDF and Cos TF illustrate the first case, and Cos TFIDF* and Cos TF* represent second case. In figure 4, it is clear that Cos-TFIDF and Cos-TFIDF* have different curves; similarly Cos-TF and Cos-TF* also behaves differently. In other words, Cosine's behavior is different for above two cases, where as realistically it should be same for both cases.

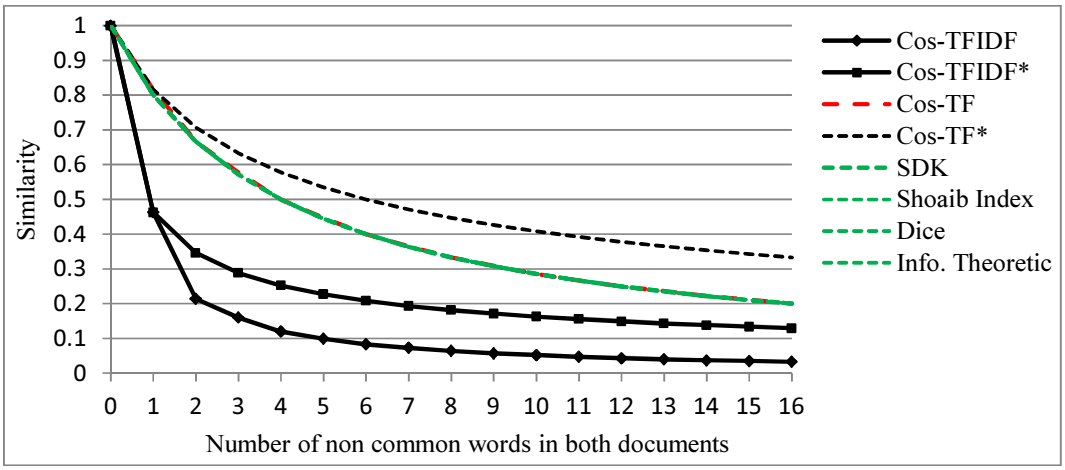

Figure 4: Effect of existence of non common words either in both documents or only in one document

SDK index, Shoaib index, Dice and Information Theoretic show same behavior for both cases that is why they are shown once. Cos-TF is equal to SDK index when number of non common words is same in both documents, and it is slightly higher than SDK index when one document has more number of non common words than the other.

In above discussion we have shown that SDK index is more logical and realistic in all scenarios. We can conclude that SDK index is more suitable to find document similarity in pair wise fashion than baseline measures.

7. Summary and Future Work

We propose a similarity measure (SDK index) for estimating textual documents similarity in pair wise fashion. SDK index shows realistic behavior in all scenarios discussed in results and discussion section. Its output is nearer to absolute similarity value than base line functions. It is, in some cases, equal to Shoaib index, Dice and Information Theoretic; and, in some cases it is better (more realistic) than these measures. We have compared the behavior of six well known similarity measures. Out of these, Cosine measure is the farthest from absolute similarity value and SDK index is the nearest. Cosine shows illogical behaviors in certain conditions while SDK index doesn't. SDK index also needs not any information about the number of documents in collection as it is needed in VSM based similarity functions. Trying to output near to absolute value may help us in deciding threshold value in clustering documents, author name disambiguation and in many other processes. Devising similarity measures with the same concept for all types of attributes in a dataset may help select a single threshold value for all types of attributes. SDK index is basically designed for textual documents. It can also be applied for entity names where variations in names are minimal. It is not a better solution for entity names (e.g., co-authors of a publication) where a name has variant forms especially when a name has multiple tokens (parts). The reason is that each variant form of a token is considered as a different token (term) in SDK. In future we want to enhance this work and will apply SDK index on standard dataset for AND in bibliographic databases, and for clustering academic documents. As future

directions SDK index can be employed and compared with other similarity measures in fields where document similarity is the main focus.

Acknowledgement

We are grateful to the Higher Education Commission (HEC) of Pakistan for its financial assistance to promote the research trend in the country under Indigenous 5000 Fellowship Program. We are also thankful to the anonymous reviewers whose suggestions would make this work worthwhile.

8. REFERENCES

[1] Kazem, T. and V. Rushikesh, 2010. Effects of Similarity Metrics on Document Clustering. In the Proceedings of the Seventh International Conference on Information Technology, pp: 222-226.

[2] Rafi, M. and M. S. Shaikh, 2013. An Improved Semantic similarity Measure for Document Clustering based on Topic Maps. Sindh University Research Journal (Science Series), 45(A-1), 59-64.

[3] Huang, A., 2008. Similarity Measures for Text Document Clustering. In the Proceedings of the 2008 New Zealand Computer Science Research Student Conference, pp: 49-56.

[4] Wan, X., 2007. Novel Document Dimilarity Measure based on EEarth Mover's Distance. Information Sciences, 177(18), 3718–3730.

[5] Han, H., L. Giles, H. Zha, C. Li and K. Tsioutsiouliklis, 2004. Two Supervised Learning Approaches for Name Disambiguation in Author Citations. In the Proceedings of the 2004 ACM/IEEE Joint Conference on Digital Libraries, pp: 296–305.

[6] Ferreira, A. A., A.Veloso, M. A. Gonçalves and A. H. F. Laender, 2010. Effective Self-training Author Name Disambiguation in Scholarly Digital Libraries. In the Proceedings of the 10th ACM/IEEE Joint Conference on Digital Libraries, pp: 39-48.

[7] Ramos, J., 2003. Using TF-IDF to Determine Word Relevance in Document Queries. In the Proceedings of the First Instructional Conference on Machine Learning.

[8] Salton, G., A. Wong and C. S. Yang, 1975. A Vector Space Model for Automatic Indexing. Communications of the ACM, 18(11), 613-620.

[9] Ghasemi, M. M. and A. Mahjur, 2013. A Method for Finding Similar Documents on the Basis of Repetition-Based Filtering. Journal of Basic and Applied Scientific Research 3(1), 603-607.

[10] Azgomi, H. and A. Mahjur, 2013. A Solution for Calculating the False Positive and False Negative in LSH Method to Find Similar Documents. Journal of Basic and Applied Scientific Research, 3(1), 466-472.

[11] Donald, M., D. Susan and M. Christopher, 2007. Similarity Measures for Short Segments of Text. Proceedings of the 29th European Confrence on information Retrieval, pp: 16-27.

[12] Wan, X., 2007. A Novel Document Similarity Measure based on Earth Mover's Distance. Information Sciences, 177(18), 3718–3730.

[13] Pandit, S., and S. Gupta,2011. A Comparative Study on Distance Measuring Approaches for Clustering. International Journal of Research in Computer Science, 2(1), 29-31.

[14] Lee, M. D., B. Pincombe and M. Welsh, 2005. An Empirical Evaluation of Models of Text Document Similarity. In the Proceedings of the XXVII Annual Conference of the Cognitive Science Society, pp: 1254-1259.

[15] Strehl, A., J. Ghosh and R. Mooney, 2000. Impact of Similarity Measures on Web-page Clustering. In the Proceedings of the 2000 AAA Workshop on Artificial Intelligence for Web Search, pp: 58-64.

[16] Larsen, B. and C. Aone, 1999. Fast and Effective Text Mining Using Linear-time Document Clustering. In the Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp: 16-22.

[17] On, B., D. Lee, J. Kang and P. Mitra, 2005. Comparative Study of Name Disambiguation Problem Using Scalable Blocking-based Framework. In the Proceedings of the 2005 ACM/IEEE Joint Conference on Digital Libraries, pp: 344-353.

[18] Lee, D., B. On, J. Kang and S. Park, 2005. Effective and Scalable Solutions for Mixed and Split Citation Problems in Digital Libraries. In the Proceedings of the 2nd Workshop on Information Quality in Informational Systems, pp: 69-76.

[19] Tan, Y. F., M.-Y. Kan and D. Lee, 2006. Search Engine Driven Author Disambiguation. In the Proceedings of the 6th ACM/IEEE Joint Conference on Digital libraries, pp: 314-315.

[20] Zhang, D., J. Tang, J. Li and K. Wang, 2007. A Constraint-based Probabilistic Framework for Name Disambiguation. In the Proceedings of the 2007 ACM Conference on Information and Knowledge Management, pp: 1019–1022.

[21] Cota, R. G., M. A. Gon˜A˘galves and A. H. F. Laender, 2007. A Heuristic-based Hierarchical Clustering Method for Author Name Disambiguation. Brazilian Symposium on Data Base, pp: 20–34.

[22] Shu, L., B. Long and W. Meng, 2009. A Latent Topic Model for Complete Entity Resolution.In the Proceedings of the 25th IEEE International Conference on Data Engineering, pp: 880-891.

[23] Bhattacharya, I., and L. Getoor, 2006. A Latent Dirichlet Model for Unsupervised Entity Resolution. In the Proceedings of the 2006 SIAM Conference on Data Mining, pp: 33-42.

[24] Song, Y., J. Huang and I. G. Councill, 2007. "Efficient Topic-based Unsupervised Name Disambiguation. In the Proceedings of the 2007 ACM/IEEE Joint Conference on Digital libraries, pp: 18–23.

[25] Wang, F., J. Tang, J. Li and K. Wang, 2010. A Constraint-based Topic Modeling Approach for Name Disambiguation. Frontiers of Computer Science, China, 4(1), 100-111, 2010.

[26] Daud, A., L. Z. J. Li and F. Muhammad, 2010. Knowledge Discovery through Directed Probabilistic Topic Models, a Survey. Frontiers of Computer Science in China, 4(2), 280-301.

[27] Blei, D. M., A. Y. Ng and M. I. Jordan, 2003. Latent Dirichlet Allocation. Journal of Machine Learning Research, vol. 3, 993–1022.

[28] Shoaib, M., M. N. Yasin, H. Niazi, M. I. Saeed and S. H. Khiyal, 2009. Relational WordNet model for Semantic Search in Holy Quran. In the Proceedings of the 2009 IEEE International Conference on Emerging Technologies, pp: 29-35.

[29] Aslam J. A. and M. Frost, 2003. "An Information-Theoretic Measure for Document Similarity. In the Proceeedings of the 26th International ACM/SIGIR Conference on Research and Development in Information Retrieval, pp: 449–450.