International Journal of Database Theory and Application
Vol.10, No.2 (2017), pp.23-30
http://dx.doi.org/10.14257/ijdta.2017.10.2.02

Comparative Analysis of Various Similarity Measures for Finding Similarity of Two Documents

Maedeh Afzali and Suresh Kumar

Manav Rachna International University, Faridabad 121004, India
maedeh.af@gmail.com and suresh.fet@mriu.edu.in

Abstract

Similarity measurements are elemental concepts in text mining and information retrieval that helps us to quantify the similarity between documents, which is effective in the improvement of the performance of search engines and browsing techniques. Nowadays, varieties of similarity measures are available, but it is not clear that which similarity measure is more effective in finding the similarity of text documents. The aim of this paper is to provide a comparative analysis of various term based similarity measures such as Cosine similarity, Jaccard and Dice coefficient in order to evaluate the performance of this similarity measures in finding the similarity of two text documents.

Keywords*: Cosine Similarity Measure, Jaccard Coefficient, Dice Coefficient*

1. Introduction

Similarity measurement between documents is an elemental concept in text mining and information retrieval. It is widely used in applications such as duplicate detection, automatic scoring, topic detection, document clustering, and text classification [1]. The main goal of similarity measures is to help us for quantifying similarity between two documents or a document and query. In other words, similarity measure is a function that calculates the degree of similarity between the pair of documents. All the similarity measures map to the range of [-1, 1] or [0, 1]. The 0 or -1 represents minimal similarity, and 1 represents absolute similarity. Presently, there is a variety of similarity measures available in literature. As stated in the paper written by W. H. Gomaa and A. A. Fahmy [1] the existing similarity measures are grouped under three categories namely; String-based similarity; Corpus-based similarity and Knowledge-based similarity.

String-based similarity measures are based on evaluating the similarity between two text strings through considering the text sequences and character decompositions. They are categorized into two groups, character-based such as Longest Common Substring (LCS) [2], Jaro Winkler [3], N-grams [4] and term-based such as Cosine similarity, Dice coefficient [5], Jaccard coefficient [6] and Euclidean distance. Moreover, in corpus-based similarity measures, they quantify the semantic similarity and linguistic meaning of words based on available corpora. Some well-known measures in this category are Hyperspace Analogue to Language (HAL) [7-8], Latent Semantic Analysis (LSA) [9], Generalized Latent Semantic Analysis (GLSA) [10], and Explicit Semantic Analysis (ESA) [11]. Furthermore, knowledge-based similarity is an approach to represent the degree of similarity between words by the help of semantic networks such as WordNet, which is a large lexical database containing English Nouns, verbs, adjectives and adverbs.

The remainder of this paper is organized as follows. In Section 2, some related work in literature on similarity measures is discussed. Further in Section 3 and 4 the Vector Space Model (VSM) and some well-known term-based similarity measures are explained in detail. In Section 5, the two phases of preprocessing and similarity calculations are discussed. Finally, in Section 6, the analyses of achieved results are provided.

ISSN: 2005-4270 IJDTA

2. Related Work

In literature similarity measures have been used for different purposes. In this section, some of the proposals are reviewed.

Anna Huang [12] compared and analyzed the effectiveness of similarity measures such as Euclidean distance, Cosine similarity, Jaccard coefficient, Pearson correlation coefficient and Averaged Kullback-Leibler Divergence for text documents clustering. They have selected the standard k-means as clustering algorithm in order to group similar documents to form coherent clusters. For an experiment, they have used seven data sets with different characteristics. The results obtained from the experiment showed that the Euclidean distance performed worst, while the performances of the other four measures were quite similar.

Singh P. [13], experimented on five well-known similarity and distance measures as such, Euclidean, Cosine, Mahalanobis, Jaccard and Pearson. They have compared the performance of these similarity measures using standard k-means algorithm. They believed that representation of objects, similarity measures and the clustering algorithm itself are the components that are influential in the final results of clustering.

H. Gupta and R. Srivastava [14] implemented a document clustering technique by using SVD (Singular Vector Decomposition) to find out the value of K required for the number of clusters. They have used K-means algorithm to create clusters and finally to make the algorithm faster than k-means algorithm, they have refined the clusters by feature voting.

V. Thada and D. V. Jaglan [15] have used genetic algorithm to provide a comparative analysis to find out the most relevent document for the given data set of keyword by using three similarity coefficients Jaccard, Dice and Cosine coefficients. Their results showed that best values were obtained using the Cosine similarity coefficients and then Dice and Jaccard, respectively.

T. Elsayed *et al.* [16] proposed a MapReduce algorithm to compute the pairwise document similarity in large collection of documents. The reason of using MapReduce framework was because they could break down the inner products calculations in computing document similarity into separate multiplication and summation stages in such a way that make the computation efficient through distributing it across several machines.

P. Niyigena *et al.* [17], have presented a new method to compute the pairwise document similarity in a corpus in order to reduce the time execution and save space execution resources. Their algorithm provided an efficient solution for pairwise documents similarity in a corpus.

3. Vector Space Model (VSM)

Vector space model (VSM), helps us to convert the original string text within a document into a vector of numbers. In VSM, each document is considered as a vector in a vector space. Assume $D = \{d_1, d_2, .., d_n\}$ is a data set that has n number of documents and $T = \{t_1, t_2, ..., t_n\}$ is a set of distinct terms, which occurs in D. Then the vector representation of document D is defined as,

$$v_d = \{tf(t_1, d), tf(t_2, d), ..., tf(t_n, d)\} \quad (1)$$

where $tf(t,d)$ denote the frequency of term $t \in T$ in document $d \in D$.

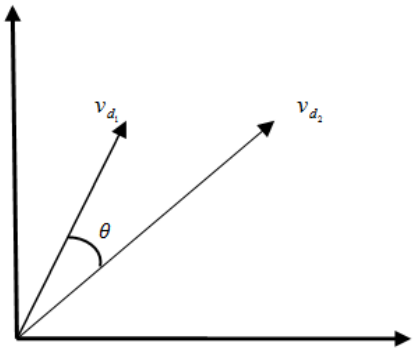

Figure 1. Documents Representation on VSM

As shown in Figure 1, the vector representation of two documents d_1 and d_2 is as follow:

$$v_{d_1} = \{tf(t_1, d_1), tf(t_2, d_1), ..., tf(t_n, d_1)\} \quad (2)$$

$$v_{d_2} = \{tf(t_1, d_2), tf(t_2, d_2), ..., tf(t_n, d_2)\} \quad (3)$$

where $tf(t_n,d_1)$ denote the frequency of the term $t_n \in T$ in document d_1 and $tf(t_n,d_2)$ denote the frequency of the term $t_n \in T$ in document d_2 [12]. Furthermore, in the vector space model after representing the documents as a vector, we can find out the similarity of documents with each other by measuring the angle between two vectors.

4. Similarity Measures

4.1. Cosine Similarity

Cosine similarity is an angle based measurement. It calculates the cosine of the angle between two vectors and helps us to find out how related two documents are. The cosine similarity of A and B is defined as,

$$\cos\theta = \frac{A.B}{\|A\|\|B\|} \quad (4)$$

or

$$\cos_{sim}(d_1, d_2) = \frac{d_1.d_2}{\|d_1\|\|d_2\|} \quad (5)$$

$$d_1.d_2 = [tf(t_1, d_1) * tf(t_1, d_2)] + [tf(t_2, d_1) * tf(t_2, d_2)] + ... + [tf(t_n, d_1) * tf(t_n, d_2)] \quad (6)$$

$$\|d_1\| = \sqrt{tf(t_1, d_1)^2 + tf(t_2, d_1)^2 + ... + tf(t_n, d_1)^2} \quad (7)$$

$$\|d_2\| = \sqrt{tf(t_1, d_2)^2 + tf(t_2, d_2)^2 + ... + tf(t_n, d_2)^2} \quad (8)$$

The cosine value varies between [-1, 1]. If documents are similar, their vectors will be in the same direction from origin, thus, they form a relatively small angle, which its cosine value will be near 1. On the other hand, when two vectors are different direction from origin, they form a wide angle and the value of the cosine is near to -1, consequently, the documents are dissimilar, and they map no similarity [12]. The most interesting property of cosine similarity is that it is easy to implement, efficient to evaluate and its independence from the length of documents.

4.2. Jaccard Coefficient

Jaccard coefficient [18] is used to compare the similarity and non-similarity of two documents based on the presence or absence of terms in documents. Ideally, it is calculated by dividing the total number of common terms between two documents by the entire number of terms that exists in at least one of the two documents. It is defined as,

$$Jaccard(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{9}$$

The degree of similarity is a value between 0 and 1. If the value is 1, the two documents are same and identical, conversely when it is 0 means that two documents are completely dissimilar. By considering the inverse of Jaccard coefficient it can be modified to a coefficient of dissimilarity, which is defined as,

$$Jaccard\ dissimilarity\ coefficient = 1 - J(A,B) \tag{10}$$

Moreover, while calculating the Jaccard coefficient, there is no need of equality in the size of two documents. The considerable drawback of this similarity is that it does not consider term frequency.

4.3. Dice Coefficient

Dice Coefficient is calculated through multiplying two into the number of common terms in the compared documents divided by the total number of terms in both documents. The formula is defined as,

$$Dice\ coefficient = 2\frac{|A \cap B|}{|A| + |B|} \tag{11}$$

The Dice coefficient value map to ranges from [0, 1], where 0 represents non-overlapping and 1 represents a perfect agreement. In addition, similar to Jaccard coefficient, Dice coefficient is also based upon the absence and presence of terms in documents.

5. Experiment

The experiment has been divided into two phases. In Section 5.1 the preprocessing steps that has been performed to make the data suitable for analysis, is discussed. Further in Section 5.2, the text data is transformed to a structured format through using the concept of Term frequency matrix. Further the similarity of documents has been measured and compared.

5.1. Preprocessing

As the data that has been used for analyzing is in the text format, there is a need to perform some preprocessing tasks to prepare the data for analysis. To perform the preprocessing tasks the R language is used, due to reason that it's free, open source and provides numerous packages that make the work easier. The tm package provided by R, presents applications for text mining. Table 1 demonstrates some predefined transformations provided by tm package.

Table 1. Predefined Transformations in Tm Package

getTransformations()	Specification of Functions
removeNumbers	Removes the numbers
removeWords	Removes the particular words provided by user
removePunctuations	Removes the punctuations
stemDocument	Applies stemming on text
stripWhitespace	Removes the extra white spaces

To apply these transformations to the documents presented in corpus, the tm_map() function is used. For other types of modifications, the content_transformer() function is helpful.

The steps that have been performed are as follows:

- Unwanted symbols such as "/", "@", "\\|" and any other symbols are converted to space.
- All words are transformed from uppercase to lowercase.
- All numbers and punctuation's are removed.
- English stop words, such as "*a*", "at", "*the*", that appears many times in the document are eliminated.
- White spaces have been stripped.

In addition to above steps, with the help of porter stemming algorithm available in SnowballC package the words in documents have reduced to its root form. For example, all three words of connection, connected and connecting has been transformed to connect. The reason of stemming is to reduce the total number of distinct terms in documents, which is useful in reducing the processing time of the final output [19].

5.2. Calculation and Evaluations

Text data are considered to be an unstructured data, which does not have a specific format and cannot be analyzed on its own. The text data has to be converted into a structured format for further analysis. To perform this conversion the concept of Term Document Matrix (TDM) is deployed. Term Document Matrix is a two-dimensional matrix which its columns correspond to documents, and its rows correspond to terms. The values of each cell is the term frequency of a word in a particular document. A small instance of a TDM is shown in Table 2.

In addition, in the corpus, there may be documents that are longer than other ones, to eliminate this bias, normalization is done, through dividing term frequency of every term by sum of term frequency of all the terms appearing in document d. The formula is defined as,

$$Normal\ \ Term\ Frequency = \frac{tf(d_i, t_j)}{\sum_k tf(d_i, t_j)} \tag{12}$$

Table 2. Term Document Matrix

Terms	Doc-1.txt	Doc-2.txt
Data	17	5
Process	8	5
Use	7	3
Store	7	0
File	4	4
System	4	3
Distribute	0	2

Further, to find the similarity of documents through using the cosine similarity measure, in the term frequency matrix the term which it's normal term frequency has value zero in any of two documents is discarded; consequently, the terms which are present in both documents are remained. For example, among the words presented in Table 2, store and distribute are eliminated. Finally, the cosine similarity measure is calculated by considering each document as a vector.

Moreover, as discussed earlier, the Jaccard and Dice coefficient are based on the absence and presence of the terms in two documents. The total number of rows in TDM which represents the terms available in both documents are considered as the union of two documents. Likewise, the total number of words that are remained after eliminating the terms that are having value zero in any of two documents returns the intersection of two documents.

6. Results

As shown in Table 3, four corpora with different properties has been considered for evaluation.

Table 3. Corpora

Terms	Specification of Documents in Corpora
Corpus 1	Contains two completely same documents
Corpus 2	Contains two documents, which one documents contains exactly a same paragraph of another document
Corpus 3	Contains two different documents about the same topic
Corpus 4	Contains two completely different documents

The preprocessing steps are applied to documents in all four different corpora. Then as discussed in Section 5 the term document matrix is created and normalized. Thereafter, the term which its normal term frequency has value zero in any two of the documents is eliminated. Table 4 depicts the dimension of the matrix, before and after elimination of terms.

Table 4. Phase I and II Results

Terms	Original Terms	Terms after Elimination
Corpus 1	1178 × 2	1178 × 2
Corpus 2	183 × 2	82 × 2
Corpus 3	1349 × 2	253 × 2
Corpus 4	1478 × 2	126 × 2

The cosine similarity, Jaccard and Dice coefficient are applied. The result of all three measures is shown in Table 5.

Table 5. Similarity Measures

Terms	Cosine Similarity	Dice Coefficient	Jaccard Coefficient
Corpus 1	1	1	1
Corpus 2	0.9165	0.6188	0.4481
Corpus 3	0.6401	0.3158	0.1875
Corpus 4	0.3447	0.1571	0.0852

To provide a better understanding of the three compared measures, the results are shown on a bar graph as depicted in Figure 2.

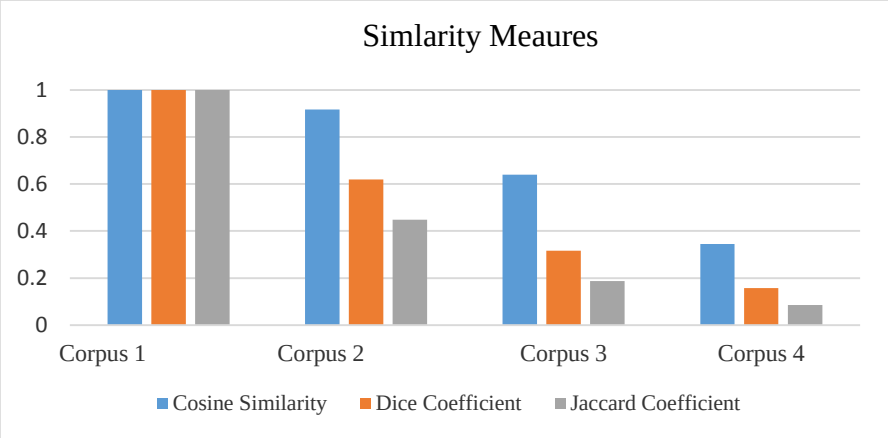

Figure 2. Similarity Measures Bar Graph

As shown in Figure 2, in corpus 1 which contain exactly two identical documents, all three measures provides the similarity of 1. In corpus 2, 3 and 4, the best result is provided by cosine similarity followed by Dice and Jaccard coefficient. However, selection of appropriate approach and similarity measure depends on properties of the experimental data and the work that users intend to perform.

7. Conclusion

This paper deals with the most useful concept in information retrieval called similarity measures. Document similarity is the process where two documents are compared to find out the similarity between them. The similarity between documents in four different corpora with different properties is calculated by considering various term based similarity measures such as, Cosine similarity, Jaccard and Dice coefficient. In future, will experiment on providing techniques in order to reduce the sparsity of the term frequency matrix in long documents. Moreover, it is significant to use methods to extract the most useful features, which will be highly helpful in finding the most similar documents.

References

[1] W. H. Gomaa and A. A. Fahmy, "A survey of Text Similarity Approaches", International Journal of Computer Applications, vol. 68, no. 13, **(2013)**, pp.13-18.
[2] D. Hirschberg, "Algorithms for the Longest Common Subsequence Problem", Journal of the ACM (JACM), vol. 24, no. 4, **(1977)**, pp. 664-675.
[3] Winkler, William E., "String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage", Proceedings of the Section on Survey Research Methods (American Statistical Association), **(1990)**.

[4] A. Barrón-Cedeno, P. Rosso, E. Agirre, and G. Labaka, "Plagiarism Detection Across Distant Language Pairs", Proceedings of the 23rd International Conference on Computational Linguistics, Beijing, China, **(2010)**.
[5] L. R. Dice, "Measures of the Amount of Ecologic Association between Species", Ecology, vol. 26, no.3, **(1945)**, pp. 297-302.
[6] P. Jaccard, "Etude comparative de la distribution florale dans une portion des Alpes et du Jura", Bulletin de la Société Vaudoise des Sciences Naturelles, **(1901)**, pp. 574-579.
[7] K. Lund and C. Burgess, "Producing High-dimensional Semantic Spaces from Lexical Co-occurrence", Behavior Research Methods, Instruments, & Computers, vol. 28, no. 2, **(1996)**, pp. 203-208.
[8] K. Lund, C. Burgess, and R. A. Atchley, "Semantic and Associative Priming in High-dimensional Semantic Space", Proceedings of the 17th annual conference of the Cognitive Science Society, vol. 17, **(1995)**.
[9] T. Landauer, P. W. Foltz and D. Laham, "An Introduction to Latent Semantic Analysis," Discourse processes, vol. 25, no. 2-3, **(1998)**, pp. 259-284.
[10] I. Matveeva, G.-A. Levow, A. Farahat, and C. Royer, "Generalized Latent Semantic Analysis for Term Representation", Proceeding of RANLP, **(2005)**.
[11] E. Gabrilovich and S. Markovitch, "Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis", Proceeding of IJCAI, vol. 7, **(2007)**, pp. 1606-1611.
[12] A. Huang, "Similarity Measures for Text Document Clustering", Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand, **(2008)**, pp. 49-56.
[13] Singh P., Sharma M., "Text Document Clustering and Similarity Measures", **(2013)**, pp. 1-8.
[14] H. Gupta, R. Srivastava, "k-means Based Document Clustering with Automatic "k" Selection and Cluster Refinement", International Journal of Computer Science and Mobile Applications, vol. 2, no. 5, **(2014)**, pp. 7-13.
[15] V. Thada and V. Jaglan, "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)**.
[16] T. Elsayed, J. Lin, D. W. Oard, "Pairwise Document Similarity in Large Collections with MapReduce", Proceedings of the 46th Annaual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers, pp. 265-268, Ohio, USA, **(2008)**.
[17] P. Niyigena, Z. Zuping, W. Li and J. Long, "Efficient Pairwise Document Similarity Computation in Big Datasets", International Journal of Database Theory and Application, vol. 8, no. 4, **(2015)**, pp. 59-70.
[18] T. Mardiana, T. B. Adji, and I. Hidayah, "The Comparation of Distance-Based Similarity Measure to Detection of Plagiarism in Indonesian Text", Proceedings of Intelligence in the Era of Big Data, Springer, Berlin, Heidelberg, **(2015)**.
[19] A. G. Jivani, "A comparative study of stemming algorithms", International Journal of Computer Technology and Applications (IJCTA), vol. 2, no. 6, **(2011)**, pp. 1930–1938.
[20] J. Ramos, "Using tf-idf to Determine Word Relevance in Document Queries", Proceedings of the first instructional conference on machine learning, **(2003)**.
[21] G. A. Al-Talib and H. S. Hassan, "A Study on Analysis of SMS Classification Using TF-IDF Weighting", International Journal of Computer Networks and Communications Security, vol. 1, no. 5, **(2013)**, pp. 189-194.
[22] J. Bank and B. Cole, "Calculating the Jaccard Similarity Coefficient with MapReduce for Entity Pairs in Wikipedia", Wikipedia Similarity Team, **(2008)**, pp. 1-17.
[23] G. Williams, "Hands-On Data Science with R Text Mining", **(2014)**, pp. 1-40.
[24] R. R. Tated and M. M. Ghonge, "A Survey on Text Mining-techniques and application", International Journal of Research in Advent Technology (E-ISSN: 2321-9637) Special Issue 1st International Conference on Advent Trends in Engineering, Science and Technology (ICATEST), **(2015)**.
[25] P. E. Coughlin, "Plagiarism in Five Universities in Mozambique: Magnitude, Detection Techniques, and Control Measures", International Journal for Educational Integrity, Springer Open Journal, vol. 11, no. 1, **(2015)**, pp. 1-19.
[26] R. Mihalcea, C. Corley, and Strapparava, "Corpus-based and Knowledge-based Measures of Text Semantic Similarity", Proceedings of the American Association for Artificial Intelligence (AAAI), vol. 6, **(2006)**, pp. 775-780.

