Available online at: https://ijact.in

Date of Submission	03/07/2019
Date of Acceptance	08/08/2019
Date of Publication	31/08/2019
Page numbers	3302-3306(5 Pages)

Cite This Paper: Mausumi Goswami, BS Purkayastha. (2019) An empirical analysis of similarity measures for unstructured data, 8(8), COMPUSOFT, An International Journal of Advanced Computer Technology. PP. 3302-3306.

ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

AN EMPIRICAL ANALYSIS OF SIMILARITY MEASURES FOR UNSTRUCTURED DATA

Mausumi Goswami[1] and B.S Purkayastha[2]

[1]CHRIST (Deemed to be University), Bangalore 560074, India
[2]Assam University, Silchar, Assam, India
mausumi.goswami@christuniversity.in

Abstract: With fast growth in size of digital text documents over internet and digital repositories, the pools of digital document is piling up day by day. Due to this digital revolution and growth, an efficient and effective technique is required to handle such an enormous amount of data. It is extremely important to understand the documents properly to mine them. To find coherence among documents text similarity measurement pays a humongous role. The goal of similarity computation is to identify cohesion among text documents and to make the text ready for the required applications such as document organization, plagiarism detection, query matching etc. This task is one of the most fundamental task in the area of information retrieval, information extraction, document organization, plagiarism detection and text mining problems. But effectiveness of document clustering is highly dependent on this task. In this paper four similarity measures are implemented and their descriptive statistics is compared. The results are found to be satisfactory. Graphs are drawn for visualization of results.

Keywords: similarity, cosine similarity, jaccard similarity, commonality, pearson, spearman's correlation.

I. INTRODUCTION

Technology has made us more productive and transforming our world. It has changed how we communicate, how learn. Computing systems are equipped with Artificial Intelligence. Nowadays computing systems are able to learn reason, hear and see. Enormous amount of new opportunities are created by Artificial Intelligence. Artificial Intelligence has given two promising technologies such as Natural Language Processing and Text Mining. These technologies enable and empower users to transform/map the key content in texts lying in documents into quantitative insight or to draw conclusion.
Text Analytics is also known as text mining which is the process of generating new knowledge or information. It examines the collection of existing written resources to map or transform the unstructured data written as text into structured data for use in further analysis. A text mining based search will identify related facts, relationships and similarity, assertions etc that would otherwise be difficult to identify and remains buried in a mass of free text or unstructured data. Most of this information available in the form text is uncertain / ambiguous / vague. Identifying plagiarism , Organizing documents, Categorizing a product customers into different categories, Identifying customers who love the product from market survey and formulation of strategies to convert rational thinking customers into a product lover , Automatic text Summarization etc work based on a similarity measure. This work is an extension

of our paper [6]. Document clustering and similarity measures are very closely connected [4, 17, 13, 2]. In [19] ,Siddiqui N., Islam S. (2019) reported Lk metric which is claimed as a novel measure. It is reported in literature that few similarity measures give better results only for low dimensional data. Such similarity measures are constructed on Euclidean distance (L2 norm). Also, it is reported that Hellinger distance-based proximity measure is restricted to only for specific data mining applications. Sohangir, S., & Wang, D. (2017), in [20] reported a novel similarity measure on the theme of sqrt-cosine similarity. This metric was claimed as an improved measure called as sqrt-cosine similarity. This was useful for document-understanding tasks. Such tasks include text categorization, text document unsupervised learning or clustering, and text query based search.

There are variety of file formats available for document files such as ASCII, hardcopies etc. OCR methods may be used for text extraction and recognition. In this research, only text documents are considered. If the input documents are images then those images may be converted to text and may be used further.

This paper is organized as mentioned below: section II describes the prominent techniques for representing documents. Section III discusses few important similarity measures used for text documents. Section IV discusses a methodology for implementation Section V discusses algorithms to compute similarity. In Section VI Experiments and Results Analysis related tasks are discussed.

II. DOCUMENT REPRESENTATION

Words are considered as the characteristic features of a document. Each document may be characterized by a collection tokens or words. Each row is a document. Each column corresponds to a feature. There are many ways to model a text document. A text document is signified as a pool or bag of words where the document is the collection of words and its frequencies. Bag of Words model is more suitable to a smaller data set. Doc2Vec algorithm [22, 23, 24, 25, 26, 27, 28, 29,] is used as a feature extraction technique in different applications such as sentiment analysis & to determine author demographics of texts. This algorithm is suitable for very rich or big training data set. In some cases it is reported that when using the same classifier, neural network based features under certain settings may outperform traditional features. It is reported that logistic regression classifier using TF-IDF outperforms doc2vec based Logistic regression classifier with a 'not so rich' training data. It is also reported that ML based model doc2vec need rich training data to learn actual contextual relation to generate sensible embedding. Also, doc2vec may generate some negative values but frequency based approaches generate only positive values. Due to this reason TF-IDF multinomial naïve bayes and doc2vec Gaussian naïve bayes are used.

III. PROXIMITY MEASURES IN DOCUMENT CLUSTERING

A proximity measure plays a very important role in clustering [8, 14, 9, 11, 5, 16, 7]. A similarity measure usage for applications using classification technique and clustering technique for text data[1, 3, 15, 18, 12, 10] is considered as one of the most vital task to understand proximity among data. The Pearson correlation estimates the direct relationship whereas Spearman correlation is based on ranked values of each data. Spearman is popularly used for ordinal values since ranking is easy. Cosine similarity is considered effective for high dimensional space. In text mining and information retrieval, each term is assigned a dimension in n-dimensional vector space. Due to high dimension, it is found to be effective for text data. In data mining, Cosine Similarity is used to measure cohesion. Computational Complexity associated with cosine similarity is low. Non-zero dimensions are considered in case of sparse vectors. Two non-zero vectors A & B may be considered to define cosine similarity as mentioned below.

$$A.B = ||A||\ \big||B\big| \cos\theta \qquad \text{(Error! Bookmark not defined.)}$$

Here, $A.B$ represents the dot product between A and B. $||A||\ is\ given\ by \sqrt{\sum_{i=1}^{n} A_i^2}$ and $\big||B|\big|\ is\ given\ by \sqrt{\sum_{i=1}^{n} B_i^2}$. Each of A and B indicate two document vectors with n terms in n dimensions. Cosine Similarity may be observed as a method to normalize the length of the documents while comparing them in the field of text mining. Let us take an example which is implemented using Python 2.7. Let us take three sentences to estimate the similarity: St1: mausent_m = "Mausumi really loves fish" , St 2: mausent_h = "Moksha loves fish too" , St3: mausent_w = "The Rohu is fish". If we try to compute counts then: mausent_1: Mausumi=1, really=1, loves=1, fish=1, too=0, Moksha=0, The=0, Rohu=0, is=0. Now for mausent_2: Moksha=0, really=0, loves=1, fish =1, too=1, Moksha=1, The=0, Rohu=0, is=0. Also for mausent_3: Moksha=0, really=0, loves=0, fish =1, too=0, Moksha=0, The=1, Rohu=1, is=1. To apply the similarity measure, it is important to compute the dot product between each pair of sentences. Also, it is required to compute the length or magnitude of each sentence .Algorithm used to compute the Cosine Similarity (A, B) may be used as described here. First, import numpy library and use np to refer to it. Next, compute dot product of A and B. This result is stored in dot_prod. Next, compute norm of A and norm of B. These results are stored in norm_A and norm_B. Finally, divide the dot product by norm of A and B. The counts we computed above are ([1, 1, 1, 1, 0, 0, 0, 0, 0]), ([0, 0, 1, 1, 1, 1, 0, 0, 0]) and ([0, 0, 0, 1, 0, 0, 1, 1, 1]). We should expect sentence_m and sentence_h to be more similar. It is found that display (cos_sim(([1, 1, 1, 1, 0, 0, 0, 0, 0]) , ([0,

0, 1, 1, 1, 1, 0, 0, 0]))) gives value 0.5 and diplay(cos_sim(, ([0, 0, 1, 1, 1, 1, 0, 0, 0]) and ([0, 0, 0, 1, 0, 0, 1, 1, 1]))) gives value as 0.25.

IV. PROPOSED METHODOLOGY

In this section, methodology applied on real life data set is described.

1. Input the Data set.
2. Apply lemmatization, stop words removal etc to preprocessing the data.
3. Select a suitable Feature Selection Technique using vector space model.
4. Construction of Term Document Matrix and reducing the space complexity.
5. Selection of suitable proximity Measure and Computation of Similarity Matrix using chosen proximity measure.
6. Computation of Descriptive Statistics for the chosen proximity measure.
7. Plotting the graph to exhibit the similarity of each document with others.

V. ALGORITHM TO COMPUTE COMMONALITY BASED SIMILARITY

In this section one of the algorithms to compute similarity is given. Remaining similarity measures are included in the experiment and results section.

Algorithm Commonality based Similarity of documents.

Input: n , D. Here, n represents number of documents from the collection of documents D.

Procedure :

1. Read number of documents, data set and initialize ES with nxn zero values.
2. For each document apply the below mentioned model to calculate the similarity with other documents.

$$(ES)nxn = (A)nxn + (I)nxn \quad (1)$$

3. Compute (A)n x n using commonality and inclusion between each pair of documents.

In this section implementation results are discussed. Python 2.7 is used to implement the preprocessing steps to transform the text data into a format suitable for computation of similarity. Implementation of similarity measures is done with MATLAB 2018 R using an Intel i3 processor and 4 GB RAM. Primary data sets of 6 documents are used for initial implementation.

Table 1 : Comparison of patterns created using ES1, ES2, ES4, ES5

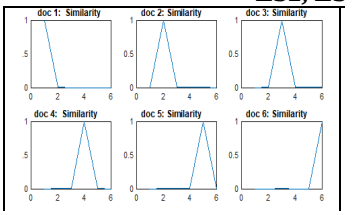

Figure 1 Proximity exhibited by ES1

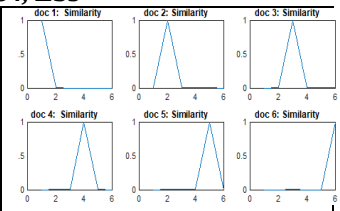

Figure 2 Proximity exhibited by ES2

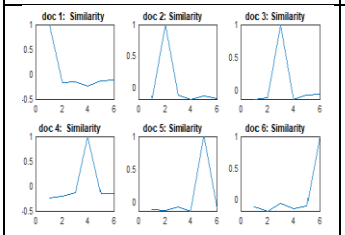

Figure 3 Proximity exhibited by ES4

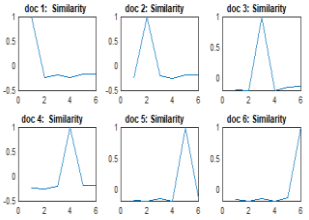

Figure 4 Proximity exhibited by ES5

Table 1 exhibits the similarity values computed using ES1, ES2, ES4, ES5. ES1 corresponds to cosine similarity, ES2 corresponds to Jaccard Similarity, ES4 corresponds to Pearson Correlation and ES5 corresponds to Spearman's rank order correlation coefficient. The similarity trend of each document doc 1, doc2, doc3, doc 4, doc5 and doc 6 are plotted and compared. Above table shows that doc1, doc2, doc 3, doc4, doc5 and doc6 have highest similarity with themselves in figure 1, 2, 3, 4 respectively when proximity is calculated by ES1, ES2, ES4 and ES5. Each figure demonstrates the similarity among documents using four different similarity measures. Models used for these measures are discussed in our work [6].

Table 2 Descriptive Statistics of Proximity measures

Statistics	ES1	ES2	ES4	ES5
min	0.0028	0.01765	-0.2292	-0.2347
max	1	1	1	1
mean	0.1715	0.2002	0.04219	0.00604
median	0.005	0.04366	-0.1257	-0.1765
mode	0.0028	0.01765	-0.2292	-0.2347
Std	0.4059	0.3922	0.4712	0.488
range	0.9972	0.9824	1.229	1.235

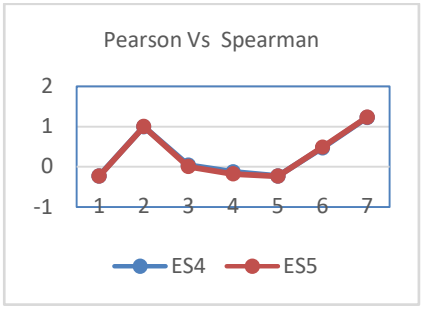

Figure 5 (a) Comparative study of descriptive statistics exhibited by ES4 and ES5

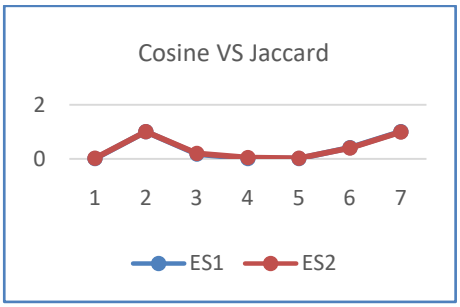

Figure 5(b) Comparative study of descriptive statistics exhibited by ES1 and ES2

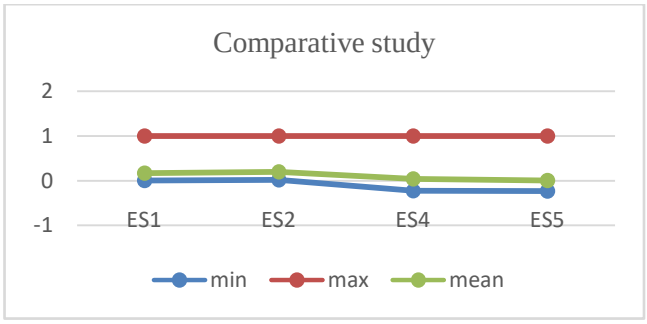

Figure 5(c) Comparative study of descriptive statistics

Fig.5. This figure describes the Comparative Descriptive Statistics of the four proximity measures used for the experimental analysis.

Figure 5.(c) demonstrates that Cosine similarity and Jaccard similarity has higher values of min and maxima compared to Pearson and spearman's rank order correlation coefficient. In figure the mean values are found to be same.

VI. Conclusion

Computing effective similarity is an important task in text mining and information retrieval. In this work a primary data set of 6 documents and 354 terms are used to calculate cosine, jaccard, pearson and spearman's correlation. Cosine similarity is one of the best when popularity is considered. Euclidean distance suffers from a drawback of not being able to perform well with the high increase in dimensionality of data. Cosine and Jaccard correlation showed similar trend for six documents. Also, the descriptive statistics was found to be highly similar. Pearson and Spearmen's correlation also showed similar trend when they are used to find the similarity among six documents of 354 terms. The experiments are conducted with secondary datasets like Reuters also. The results are found to be satisfactory. The results computed using similarity measures are used to perform clustering at a later stage. All the clusters computed using the results found using similarity matrix computed based on the four different similarity measures are found to exhibit values between 0 and 1 when validated using silhouette coefficient. These results also confirm the correctness of similarity computation. A theoretical comparison of all the similarity measures and results of clustering will be included in future work.

References

[1]. Wajeed, M. A., & Adilakshmi, T. (2011, September). Different similarity measures for text classification using KNN. In *2011 2nd International Conference on Computer and Communication Technology (ICCCT-2011)* (pp. 41-45). IEEE.

[2]. Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, *20*, 53-65.

[3]. Patidar, A. K., Agrawal, J., & Mishra, N. (2012). Analysis of different similarity measure functions and their impacts on shared nearest neighbor clustering approach. *International Journal of Computer Applications*, *40*(16), 1-5.

[4]. K.Chandan, B. R. B. (2017). A Novelistic Querying procedure for clustering the Legal Precedents. International Journal of Engineering and Computer Science, 6(1). Retrieved from http://www.ijecs.in/index.php/ijecs/article/view/2073.

[5]. Sindhiya, B., & Tajunisha, N. (2013). Concept and Term Based Similarity Measure for Text Classification and Clustering, 9(3), PP 28-33, International Journal of Engineering research & development.

[6]. Goswami, M., Babu, A., Purkayastha B.S., (2018) . A Comparative Analysis of Similarity Measures to find Coherent Documents". *Applied Sciences and Managment* , ,8(11),786-797

[7]. A, V.S.P. (2013), Space and Cosine Similarity measures for Text Document Clustering. *International Journal of Engineering Research & Technology* 2(2), 2278–0181

[8]. Amorim, de, R. C., & Hennig, C. (2015). Recovering the number of clusters in data sets with noise features

using feature rescaling factors. *Information Sciences*, *324*, 126-145.

[9]. Chim, H., & Deng, X. (2008). Efficient phrase-based document similarity for clustering. *IEEE Transactions on Knowledge and Data Engineering*, *20*(9), 1217-1229.

[10]. Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In Kdd (Vol. 96, No. 34, pp. 226-231).

[11]. Kalaivendhan, K., & Sumathi, P. (2014). An efficient clustering method to find similarity between the documents. Int. J. Innov. Res. Comput. Commun. Eng, 1.

[12]. Han, J., Pei, J., & Kamber, M. (2011). Data mining: concepts and techniques. Morgan Kaufmann Publishers, Elsevier, Waltham, MA, USA. ISBN: 978-0-12-381479-1.

[13]. Kaufman, L., & Rousseeuw, P. J. (2005). Finding groups in data: an introduction to cluster analysis. John Wiley & Sons, Hoboken, New Jersey.

[14]. Lin, Y. S., Jiang, J. Y., & Lee, S. J. (2014). A similarity measure for text classification and clustering. IEEE transactions on knowledge and data engineering, 26(7), 1575-1590.

[15]. Sebastiani, F. (2002). Machine learning in automated text categorization. ACM computing surveys (CSUR), 34(1), 1-47.

[16]. Sruthi, K., & Reddy, M. B. V. (2013). Document clustering on various similarity measures. International Journal of Advanced Research in Computer Science and Software Engineering, *3*(8), 1269-1273.

[17]. Karypis, G., Kumar, V., & Steinbach, M. (2000, August). A comparison of document clustering techniques. In KDD Workshop on Text Mining.

[18]. Tan, P. N. (2005). Introduction to data mining. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

[19]. S Siddiqui N., Islam S. (2019) k-Factor-Based Cosine Similarity Measurement. In: Satapathy S., Joshi A. (eds) Information and Communication Technology for Intelligent Systems. Smart Innovation, Systems and Technologies, vol 107. Springer, Singapore.

[20]. Sohangir, S., & Wang, D. (2017). Improved sqrt-cosine similarity measurement. Journal of Big Data, 4(1), 25.

[21]. Murty, M. N., & Devi, V. S. (2011). Pattern recognition: An algorithmic approach. Universities Press (India) Pvt. Ltd, Springer-Verlag London.

[22]. Lee, S., Jin, X., & Kim, W. (2016). Sentiment classification for unlabeled dataset using doc2vec with jst. In Proceedings of the 18th Annual International Conference on Electronic Commerce: e-Commerce in Smart connected World (p. 28). ACM. [feature selection , vector space model, tf-idf]

[23]. Bilgin, M., & Şentürk, İ. F. (2017). Sentiment analysis on Twitter data with semi-supervised Doc2Vec. In *2017 International Conference on Computer Science and Engineering (UBMK)* (pp. 661-666). IEEE.

[24]. Lau, J. H., & Baldwin, T. (2016). An empirical evaluation of doc2vec with practical insights into document embedding generation. arXiv preprint arXiv:1607.05368.

[25]. Markov, I., Gómez-Adorno, H., Posadas-Durán, J. P., Sidorov, G., & Gelbukh, A. (2016, October). Author profiling with doc2vec neural network-based document embeddings. In Mexican International Conference on Artificial Intelligence (pp. 117-131). Springer, Cham.

[26]. Kim, D., & Koo, M. W. (2017). Categorization of Korean news articles based on convolutional neural network using Doc2Vec and Word2Vec. Journal of KIISE, 44(7), 742-747.

[27]. Trieu, L. Q., Tran, H. Q., & Tran, M. T. (2017, December). News classification from social media using twitter-based doc2vec model and automatic query expansion. In Proceedings of the Eighth International Symposium on Information and Communication Technology (pp. 460-467). ACM.

[28]. Maslova, N., & Potapov, V. (2017, September). Neural network doc2vec in automated sentiment analysis for short informal texts. In International Conference on Speech and Computer (pp. 546-554). Springer, Cham.

[29]. Lee, H., & Yoon, Y. (2017). Engineering doc2vec for automatic classification of product descriptions on O2O applications. Electronic Commerce Research, 1-24.