Efficacy of Cosine Similarity Measure for Prevention of Terror Related Activities on Web

Syed Azzam Zafar[1], Prerona Das[2], Neyaz Wakil[3], Madhav Solanke[4]

[1-4]B.E, Dept. of Computer Engineering, Sinhgad Institute of Technology, Lonavala, Maharashtra, India

---***---

Abstract – *The terrorism activities across the globe has seen an unprecedented spike in recent years. There are certain organizations which have been collecting and classifying global terrorism activities information such as International Institute of Economics and Peace, National Consortium for the Study of Terrorism and Responses to Terrorism. Consequentially there is a sufficiently large classified and clustered database gathered at all the major intelligence agencies through web which is predominantly text. This paper attempts to show the efficacy of applying document cosine similarity measure among other similarity measures to the new set of text data collected from the web using any available robust database on terror related activities.*

***Key Words*: Cosine Similarity, Text Mining, Intelligence, terrorism, Propaganda Articles, Database.**

1.INTRODUCTION

In the past two decade the single most important factor in the downfall of global economy and world peace is terrorism. There are some countries which are completely ruined by terror attacks over the last two decades. [1] The Institute for Economics and Peace has released the Global Terrorism Index declaring no less than 10 countries to be the active hotbed of terror attacks. [2] The U.N recognizes internet as the major source for terror related activities. Hence, we can easily say that all the major intelligence agencies and their governing bodies have a sufficiently large database of terror related content acquired from internet. The past decade has also seen a rise in application of machine learning and data mining algorithms to web document analysis.

1.1 Similarity Measures

Similarity is the measure of how much alike two data objects are. Similarity in a data mining context is usually described as a distance with dimensions representing features of the objects. A small distance indicating a high degree of similarity and a large distance indicating a low degree of similarity. There are various domains of information which respond differently to different similarity measures. Before the data is clustered in any category, a similarity measure must be applied which will map the symbolic descriptions of two objects into a single numeric value. The similarity measure depends upon two factors, the properties of two objects and the measure itself. There is no similarity measure that is universally best for all kinds of clustering problems. There are various similarity measures used for different domains of data such as: Metric, Euclidean, Cosine, Dice, Jaccard. For text data mining various researchers have shown that Cosine similarity measure perform far better than other similarity measures.

1.2 Cosine Similarity and Text Mining

[3] A text document can be represented by thousands of attributes, each recording the frequency of a particular word or a phrase in the document. Thus, each document is an object represented by what is called a term-frequency-vector.

Cosine similarity is a measure of similarity that can be used to compare documents or, say, give a ranking of documents with respect to a given vector of query words. Let **x** and **y** be two vectors representing term frequencies in two different documents for comparison. Using the cosine measure as a similarity function, we have

$$sim(\mathbf{x},\mathbf{y}) = \frac{x.y}{\|x\|\,\|y\|},$$

$where \ \|x\|$

is the Euclidean norm of vector

$x = (x1, x2, ..., xp),$

defined as

$$\sqrt{x1^2 + x2^2 + ... + xp^2}.$$

The cosine similarity measure is used often in conjunction with the term frequency and inverse document frequency approach known as TFIDF for generating vector space model for text documents.

TF: Term Frequency, measures how frequently a word appears in a document.

IDF: Inverse Document, which measures how important a term is.

2. LITERATURE SURVEY

The Term Frequency-Inverse Document Frequency based Cosine similarity has seen a preference over other similarity measures in text data mining. Especially in terrorism related text data analysis, the cosine measure has been a favored preliminary approach for analysis in prediction. Here we present two kinds of papers, one the terrorism related analysis papers which have preferred cosine measure, and the other, similarity analysis papers which have showed the suitability of cosine measure over other measures for text data mining.

Cannon et. al. [4] have applied cosine similarity measure on the Syrian conflict. They have tried to show the change in Turkish military policies regarding Syria can be predicted by analyzing the articles published in government backed media agencies. The research which expanded a period of four years from 2012-2016 states "…. changed in 2014 when AA articles on safe zones and no-fly zones demonstrated a much greater degree of cosine similarity [0.95], followed by buffer zones and no-fly zones [0.75]. … we therefore posit that Turkish policy makers showed equal interest in no-fly zone/safe zones…."

Elovici et. al. [5] have presented a knowledge-based system in their research paper which performs real-time detection of users suspected of being engaged in terrorist activities. In this paper they have also proposed cosine similarity measure to be applied on the vector space model of the web document. The paper states, "In this study each Web page is considered as a document and is represented as a vector… The cosine similarity measure is commonly used to estimate between an accessed Web page and a given set of terrorists' topics of interests.

Hariharan et. al. [6] have analyzed and compared various similarity measures for text documents. They have concluded that concluded that "..Cosine yields higher similarity values..". Hence, we can say this measure takes a larger pool of data for further analysis.

Zahrotun, L., [7] in her paper has compared similarity algorithms on Shared Nearest Neighbor (SNN). She concludes that "Results of cosine similarity has the highest value in comparison with Jaccard similarity and the joint between Cosine and Jaccard similarity."

Goyal, M. M., et. al., [8] have done a running time analysis on the cosine and fuzzy similarity measure on text documents.

The work by Hung Chim. et. al. [9] has presented a successful approach to extend the usage of TFIDF weighting scheme: the term TFIDF weighting scheme is suitable for evaluating the importance of not only the keywords but also the phrases in document string.

Boya H., [10] in his masters paper has demonstrated various approaches of applying cosine measure on JSON files containing clinical data.

3. PROPOSED APPROACH

3.1 Data Preprocessing

```
Input: New set of suspicious text data collected from web.

Output: Processed Documents with junk words removed and term
vector size reduced through stop word removal and stemming.

1. Web page extraction and preprocessing:
        Scrape data from web pages into desired file format say
        JSON files.
        Data processing using python is tedious in JSON file
format. So the data from a subset can be stored in csv files.

2. Data Cleaning: Junk characters are currently present in the data
        which need to be replaced before calculating cosine
similarity.
        new_key = key.replace(u”\xa0’,””);

3. Stop Word Removal: Many of the most frequently used words in
        English are likely to be useless in text mining.
        These words are called StopWords.
        examples: the, of, and, to, an,...
        It reduces the dataset size by about 20-30 %.

4. Stemming: Techniques to find stem of a word.
        words: (User, users, used, using)
        stem: use
        Stemming reduces term vector size.
```

3.2 Ranking Documents

```
Input: Preprocessed Documents

Output: List of Documents with similarity measure greater than
        certain critical value from a certain set of documents in
        the database.
1. Representing Document through Term frequency Vector:
        1.1 Keyword Extraction
        1.2 Using TF-IDF to calculate Term frequency vector.

TF-IDF Calculation:
                tf = 1/(Number of distinct keywords in a
document)
        idf = log to the base 10 of(total number of
documents/number of documents the keyword appears in.)

        Calculate tf*idf value for each distinct keyword in each
        Di and represent all the values in the tf*idf table.

2. Similarity Measurement with documents in the database:
        Compute the cosine similarity between every pair of
        document Di as defined in section 1.2.
3. Selecting Suspicious Documents:

        3.1 Assign Critical value for similarity
                According to sensitivity of the context of
document
          search, we may assign a critical value above which
```

3.1
the documents from the scanned dataset will
be considered similar.

3.2 **Some sample similarity value classification:**
Sim(document, database) = 1:
- Old hate speech or terrrorism related propaganda articles circulating on web.

Sim(document, database) = 0.7-0.9
- Modified terror classified documents being spread on web platforms.

Sim(document,database)>= Critical Value:
- Potential terror related document which may go on for further classification and analysis.

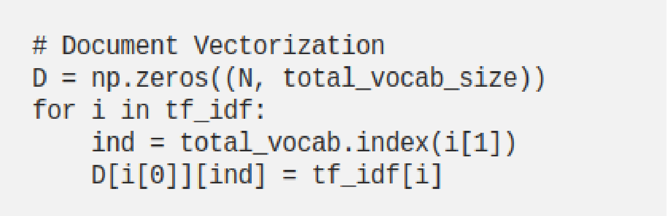
Sim(document,database)<= Critical Value:
- Document is not terrorism related.

4. EXPERIMENTAL RESULTS

Here we consider a certain set of text documents to be the data-sets obtained from the actual terrorists, present in the database.

We want to pairwise compare the new set of classified documents with the documents present in database on the standard bag of words query for the terrorism and hate speech data-sets.

4.1 Vectorization



```
# Document Vectorization
D = np.zeros((N, total_vocab_size))
for i in tf_idf:
    ind = total_vocab.index(i[1])
    D[i[0]][ind] = tf_idf[i]
```

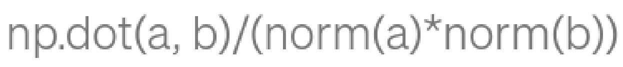
Fig 1: Document Vectorization

4.2 Term-Frequency Table

For vector, we need to calculate the TF-IDF values, TF we can calculate from the query itself, and we can make use of DF that we created for the document frequency, and finally we will store in a (1,vocab_size) numpy array to store the tf-idf values, index of the token will be decided from the total_vocab list.

4.3 Cosine Similarity Calculation

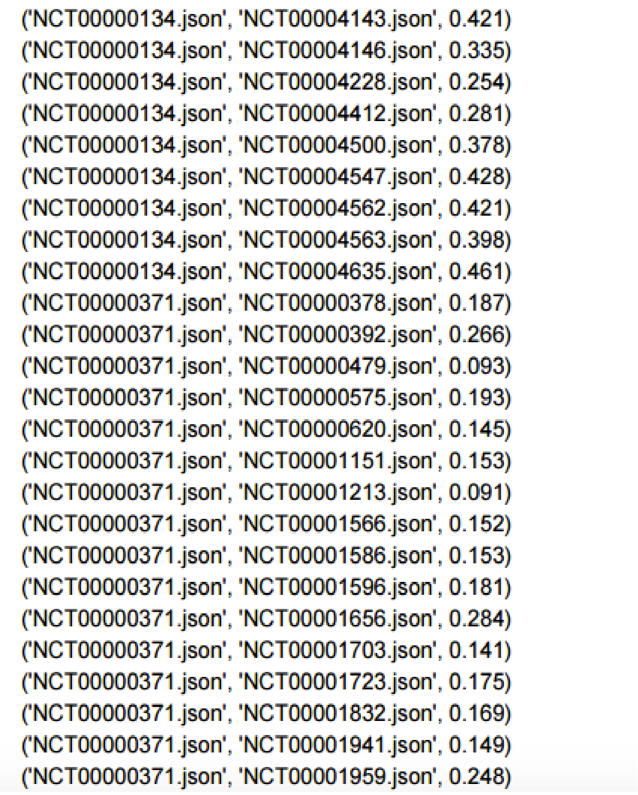
Now, all we have to do is calculate the cosine similarity for all the documents and return the maximum k documents. Cosine similarity is defined as follows.



np.dot(a, b)/(norm(a)*norm(b))

4.4 Sample Results

The resulting pairwise comparison of files would be similar to the below figure.



```
('NCT00000134.json', 'NCT00004143.json', 0.421)
('NCT00000134.json', 'NCT00004146.json', 0.335)
('NCT00000134.json', 'NCT00004228.json', 0.254)
('NCT00000134.json', 'NCT00004412.json', 0.281)
('NCT00000134.json', 'NCT00004500.json', 0.378)
('NCT00000134.json', 'NCT00004547.json', 0.428)
('NCT00000134.json', 'NCT00004562.json', 0.421)
('NCT00000134.json', 'NCT00004563.json', 0.398)
('NCT00000134.json', 'NCT00004635.json', 0.461)
('NCT00000371.json', 'NCT00000378.json', 0.187)
('NCT00000371.json', 'NCT00000392.json', 0.266)
('NCT00000371.json', 'NCT00000479.json', 0.093)
('NCT00000371.json', 'NCT00000575.json', 0.193)
('NCT00000371.json', 'NCT00000620.json', 0.145)
('NCT00000371.json', 'NCT00001151.json', 0.153)
('NCT00000371.json', 'NCT00001213.json', 0.091)
('NCT00000371.json', 'NCT00001566.json', 0.152)
('NCT00000371.json', 'NCT00001586.json', 0.153)
('NCT00000371.json', 'NCT00001596.json', 0.181)
('NCT00000371.json', 'NCT00001656.json', 0.284)
('NCT00000371.json', 'NCT00001703.json', 0.141)
('NCT00000371.json', 'NCT00001723.json', 0.175)
('NCT00000371.json', 'NCT00001832.json', 0.169)
('NCT00000371.json', 'NCT00001941.json', 0.149)
('NCT00000371.json', 'NCT00001959.json', 0.248)
```

[10]Fig 2: Sample similarity matrix

From the above similarity values we may able to trace the connection of the suspicious web documents to the existing database also.

5. CONCLUSION

The investigation is done for finding the efficacy of cosine similarity measure on prevention of terrorism related text content on web using the available database. Even though the cosine measure is slower in comparison to some other similarity measures, the simplicity of its algorithm and its large prevalence makes it the most suitable for preliminary evaluation of classified data with respect to the available database.

6. ACKNOWLEDGMENT

We express our hearty gratitude towards:

Prof. Prashant Raut for guiding us to understand the work conceptually and also for his constant encouragement to work on the 'Efficacy of Cosine Similarity for Prevention of Terrorism related Activities on Web'.

Dr. Sachin Babur Head of Department (Computer Engineering) for providing encouragement and necessary resources.

REFERENCES

[1] Institute for Economics and Peace(IEP), in "Global Terrorism Index", 2020

[2] United Nations Office on Drugs and Crime, Vienna in "Use of The Internet for Terrorist Purposes",2012

[3] Han J., Kamber M., Pei J. (2012) "Data Mining: Concepts and Techniques", Morgan Kauffmann.

[4] Cannon, Brendon J.; Nakayama, Miikiyasu; Sasaki, Daisuke; and Rossiter, Ash, "Shifting Policies in Conflict Arenas; A Cosine Similarity and Text Mining Analysis of Turkey's Syria Policy, 2012-2016." Journal of Strategic Security 11, no. 4 (2018):: 1-19

[5] Elovici, Y., Kandel, A., Last, M., Shapira, B., Zaafrany, O., "Using Data Mining Techniques for Detecting Terror-Related Activities on the Web" University of South Florida, 4202 E. Fowler Ave. ENB 118 Tampa, FL, 33620, USA.

[6] Hariharan, S., Srinivasan, R., "A Comparison of Similarity Measures for Text Documents, Journal of Information & Knowledge Management, Vol 7, No. 1. (2008) 1-8

[7] Zahrotun, L., "Comparison Jaccard similarity, Cosine Similarity and Combined Both of the Data Clustering With Shared Nearest Neighbor Method", Computer Engineering and Applications Vol. 5, No. 1, February 2016

[8] Goyal, M. M., Agrawal, N., Sarma, M. K., Kalita, J. N., "Comparison Clustering using Cosine and Fuzzy set based Similarity Measures of Text Documents", Internation Conference on Computing and Communication Systems 2015, arXiv:1505.00168

[9] Chim, H., Ding, X., "A New Suffix Tree Similarity measure for Document Clustering" in ACM 978-1-59593-654-7/07/0005

[10] Boya, H., "Finding Similarity using Metadata of Clinical Trials using Natural Language Processing in DATABRIDGE", Faculty of School of Information and Library Science, University of North Carolina at Chapel Hill,2016.