Scientific Journal of Impact Factor(SJIF): 3.134

e-ISSN(O): 2348-4470
p-ISSN(P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 2,Issue 4, April -2015

AN EFFECTUAL APPROACH FOR CALCULATING COSINE SIMILARITY

Prof. Sarika N Zaware[1], Mr. Asmit Gautam[2], Ms. Sumedha Nashte[3], Ms. Puneet Khanuja[4]

[1](HOD)Dept of Computer Engineering,AISSMS's Institute Of Information Technology,University of Pune, India
[2,3,4]Dept of Computer Engineering,AISSMS's Institute Of Information Technology,University of Pune, India

Abstract *- Cosine similarity is a technique to find the similarity measures. A cosine similarity formula would measure similarity between two vectors. This is done by finding the angle space between the two vectors. The cosine values range from 0 to 1, accordingly the angle varies from 90 to 0 degrees respectively. If the cosine value is 1, it means the angle between the vectors is 0. This implies that the two vectors are the same. Whereas if the cosine value is 0, it means that the angle between them is 90 degrees; which subsequently implies that the two vectors are distinctively different. Cosine similarity has various applications, especially in data mining, like text summarization. This paper recommends a novel technique of calculating cosine similarity. The technique is validated by observing experimental results which prove to be more cost effective and efficient.*

Keywords- *Cosine coefficient, tf-idf, Text similarity.*

I. INTRODUCTION

Large volumes of data are available on the net today. Referring to each and every source of data is a tedious job. A text summarizer could be used for this purpose. A text summarizer would condense the large amount of data. This might be helpful to skim out the important topics and relevant data.

Also, in acts of plagiarism, there could be unauthorized use of copyright data, which could hamper the motivation to produce new and original content. It also disregards the potential and efforts of the author.

The above reasons and many more applications require calculating text similarity. Text similarity takes two sentences at a time and checks for similarity between them. A text summarizer could check similar sentences and pick up unique sentences and excerpts from a bunch of documents and append them to give a summary. An application for plagiarism checker could compare the sentences of one document with any possible relative documents for similarity. Thus text similarity stands an important aspect with references to various applications.

Cosine similarity is one of the techniques to calculate similarity between sentences. The traditional cosine similarity formula uses term frequency (tf) of words to measure similarity between sentences. In this paper, we propose a new method for calculating cosine similarity which reduces the cost and also makes its implementation simpler.

The organization of this paper is as follows. In section II, we take into consideration the various other related works of text similarity. Section III, presents the motivation of this work. In section IV, we describe the proposed methodology along with the advantages of it over the traditional approach. Section V shows the experimental results followed by conclusion and references.

II. RELATED WORK

We need similarity measures for finding the similarities between two or more marked documents. In current scenario there are many methods to calculate the similarity between documents. As some methods which are renowned and used in different applications like plagiarism detection, summarizers etc. are Cosine Coefficient, Dice Coefficient, Jaccard Coefficient. These methods are basically used to find the similarity coefficient between two or more text contents. As we have done a deep study on the use of cosine coefficient we will see more efficient use of it in the further part. But for now let's go through a comparative view of all the methods present.

@IJAERD-2015, All rights Reserved

1)The Sorensen-Dice Coefficient : Dice is mainly used by ecological community data. Dice provides a justification that is more empirical than the theoretical , though it can be represented theoretically as the intersection of two fuzzy sets. As compared to Euclidean distance, Sorensen distance retains more sensitivity. The formula proposed for Dice coefficient is :

$$QS = \frac{2C}{A+B} = \frac{2|A \cap B|}{|A| + |B|}$$

where *A* and *B* are the number of species in samples A and B, respectively, and *C* is the number of species shared by the two samples. QS is the quotient of similarity and ranges between 0 and 1(5)

$$s = \frac{2|X \cap Y|}{|X| + |Y|}$$

For sets *X* and *Y* of keywords used in information retrieval, the coefficient may be defined as twice the shared information (intersection) over the sum of cardinalities :[6]

When taken as a string similarity measure, the coefficient may be calculated for two strings, *x* and *y* using bigrams as follows:[7]

$$s = \frac{2n_t}{n_x + n_y}$$

where n_t is the number of character bigrams found in both strings, n_x is the number of bigrams in string *x* and n_y is the number of bigrams in string *y*.

2)Jaccard Index: The other name for Jaccard index is Jaccard Similarity Coefficient (coined on the name of Paul Jaccard), is a measure used for comparing the similarity and difference of the sample se which can be a string or a whole text document. The Jaccard coefficient measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}.$$

(If A and B are both empty, we define J(A,B) = 1.) Clearly, $0 \leq J(A,B) \leq 1$. [8]

The Jaccard distance which is complementary to Jaccard coefficient and is used to calculate dissimilarities between the sample set, is calculated by subtracting the Jaccard coefficient from 1. As stated in the formula below.

$$d_J(A,B) = 1 - J(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}.$$

3) *Cosine Coefficient:* The cosine similarity is used to measure the similarity or dissimilarity angle . The cosine of 0° is 1, and it is less than 1 for any other angle. It depends on the orientation and not the magnitude: two vectors with the same orientation have a Cosine similarity of 1, two vectors at 90° have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude. Cosine similarity is particularly used in positive space, where the outcome is neatly bounded in [0,1].[]The cosine of two vectors can be derived by using the Euclidean dot product formula:

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \cos\theta$$

Given two vectors of attributes, *A* and *B*, the cosine similarity, *cos(θ)*, is represented using a dot product and magnitude as

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

III. MOTIVATION

As in the above sections we have discussed the similarity measures and the ways of calculating it through different channels. Cosine similarity or rather Cosine coefficient is mostly used for similarity in most of the systems. The cosine is calculated using the TF value or the IDF value calculated between the contents. The calculation of the Cosine using the TF has quite a few drawbacks which basically lead to the invention of this technique for cosine calculation. Traditional approach for cosine calculation is as scripted in the below approach.

Step1) Create a TF vector or matrix which is of type double or float.
Step2) Consider two rows of the matrix for calculation of cosine.
Step3) Multiply the values of TF for each element of the first row to corresponding element of the second row and these products.
Step4) Find the modulus of each of these rows.

@IJAERD-2015, All rights Reserved

Step5) Divide the sum calculated by the product of the modulus as per the cosine formula.

According to the above approach, a lot of memory is wasted as a matrix has to be stored which is of float or double.
Multiplication of float and double is very costly.
Inaccuracy may occur if many words have same TF value or some similarity may be calculated between two totally different sentences on the basis of TF.
Keeping the above factors in mind, we have proposed a new approach to overcome these difficulties in the calculation of cosine coefficient.

IV. PROPOSED MODEL

The proposed system is designed to overcome the drawbacks of the pre-existing system.

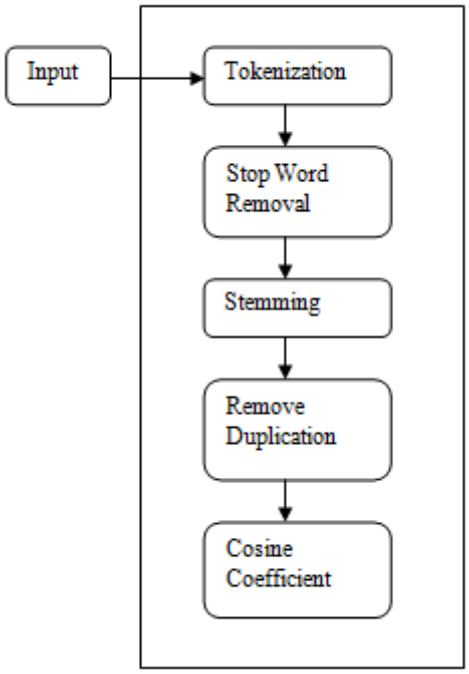

Fig 1: System Diagram

The system can be divided into the following phases:

a) Tokenisation:
Tokenisation is the separation of words in a sentence in the form of tokens. Each word accounts for one token. Tokenization can be done first on the basis of sentences, i.e., each sentence forms a token and then theses sentences can be further tokenised on the basis of white spaces, commas, etc.

b) Stop word removal:
Stop words are words that are not of much importance or do not form an integral part of a sentence. Stop words could be words like: is, an, the, if, he, of, etc. A list of stop words can be used to compare and then filter out such words.

c) Stemming:
Stemming can be defined as the identification of the route word. Using stemming allows us to ensure that two words used in two different forms are not treated as two separate words altogether. Here the prefix and suffix of the words are removed to get the root word. For example: 'polymorphism' becomes 'morph' by removing 'poly' i.e. prefix and 'ism' i.e. suffix.

d) Remove duplication :

@IJAERD-2015, All rights Reserved

If there is repetition of words in the same sentence, keep only one copy of the word and eliminate the replicas.

e) Cosine:

Cosine similarity is used to determine the similarity between two sentences. The traditional approach uses tf –idf i.e. term frequency and inverse document frequency to calculate cosine of two sentences. A more accurate solution can be obtained using the following simple steps:

i) Compare the length of the two sentences and consider the greater length.

ii) For the words in the first sentence, increment count of fi by 1.
fi+=1;
where √fi gives |x|.

iii) Compare each word of the first sentence with all the words of the second sentence. If the word is present in the second sentence, increment qu and se by 1,
qu+=1;
se+=1;
where, qu gives $\sum X_i * Y_i$ and √se gives |y|.

iv) Calculate cosine coefficient as:
Cos= qu/(√fi*√se) ;

The proposed system enhances the efficiency as:

a) Calculation of tf-idf is eliminated.
b) Use of double or float arrays is not required, which reduces memory requirements.
c) Complex multiplication is replaced by simple addition of 1.

When we consider the Tf-idf vector or matrix used for the calculation of cosine, as in the earlier approaches, there are chances that multiple words have the same value of tf-idf, which could lead to incorrect calculation of similarity. The proposed system eliminates this possibility as we are actually considering only those words in the sentences that are exactly the same.

Algorithm:
Input:
Text Document for which similarity is to be calculated.
Output:
Cosine Similarity coefficient for the input sentences.
Steps:
1) Tokenise the input document.
2) Remove the Stop words.
3) Apply stemming
4) Remove duplicate words from single sentences.
5) Calculate the cosine as follows:
a) Find the number of words in the longer sentence.
b) for i=0 to max-1
i) compare word with other sentence and fi+=1;
ii) if comparison result is true,
qu+=1;
se+=1;
c)Calculate cosine coefficient as:
Cos= qu/(√fi*√se) ;
6)Repeat step 5 for n a sentences.

Fig 2: Algorithm of System.

@IJAERD-2015, All rights Reserved

V. EXPERIMENTAL RESULTS

Consider a sample input as a text document with a few sentences as shown in Fig 3.

If we calculate the similarity between the sentences using the traditional method of calculation, i.e. using tf, the result may not be as accurate. The result of this is illustrated in Fig 4. If you notice, there is no similarity between sentence 1 and 5 but, using the tf matrix, the cosine coefficient calculated is not 0.0, indicating some amount of similarity.

Similarity between sentences calculated by the proposed system which gives a more accurate measure is illustrated in Fig 5. If you notice, there is no similarity between sentence 1 and 5 and using the proposed method, the cosine coefficient calculated is 0.0, indicating some amount of similarity.

VI. RESULTS

Since text similarity plays a vital role in text mining, this paper suggests a new approach which allows us to calculate the cosine coefficient of similarity in a more productive and cost efficient way. It provides numerous advantages over the traditional approach of calculating similarity and therefore could be applied in various applications.

VII. REFERENCES

1. http://www.sekj.org/PDF/anbf40/anbf40-415.pdf
2. van Rijsbergen, Cornelis Joost (1979). *Information Retrieval*. London: Butterworths. ISBN 3-642-12274-4.
3. Kondrak, Grzegorz; Marcu, Daniel; Knight, Kevin (2003). "Proceedings of HLT-NAACL 2003: Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics". pp. 46–48
4. .http://en.wikipedia.org/wiki/Jaccard_index
5. http://en.wikipedia.org/wiki/Cosine_Similarity

Input:

Football refers to a number of sports that involve, to varying degrees, kicking a ball with the foot to score a goal. Various forms of football can be identified in history, often as popular peasant games. These different variations of football are known as football codes. Contemporary codes of football can be traced back to the codification of these games at English public schools in the eighteenth and nineteenth centuries. It is a lot of fun.

Fig 3: Input to System.

@IJAERD-2015, All rights Reserved

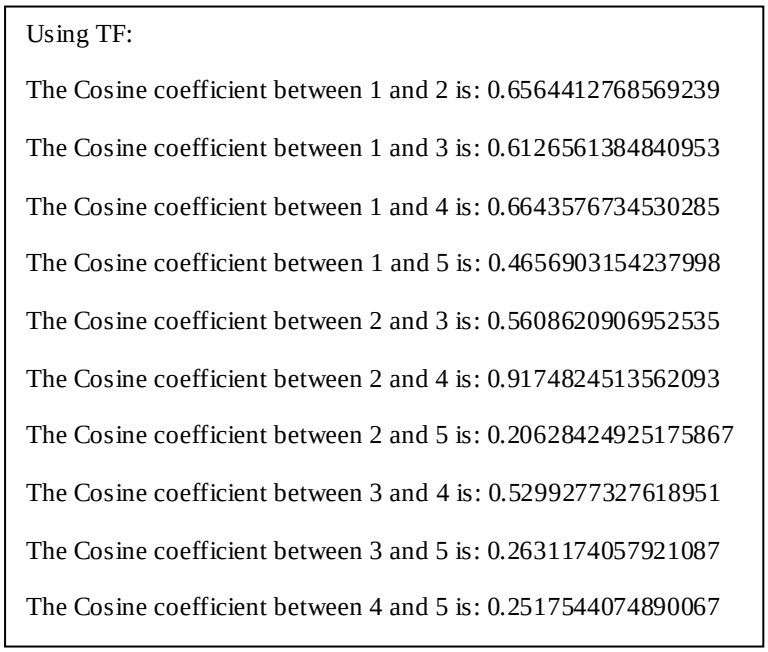

Fig 4: Result using Tf

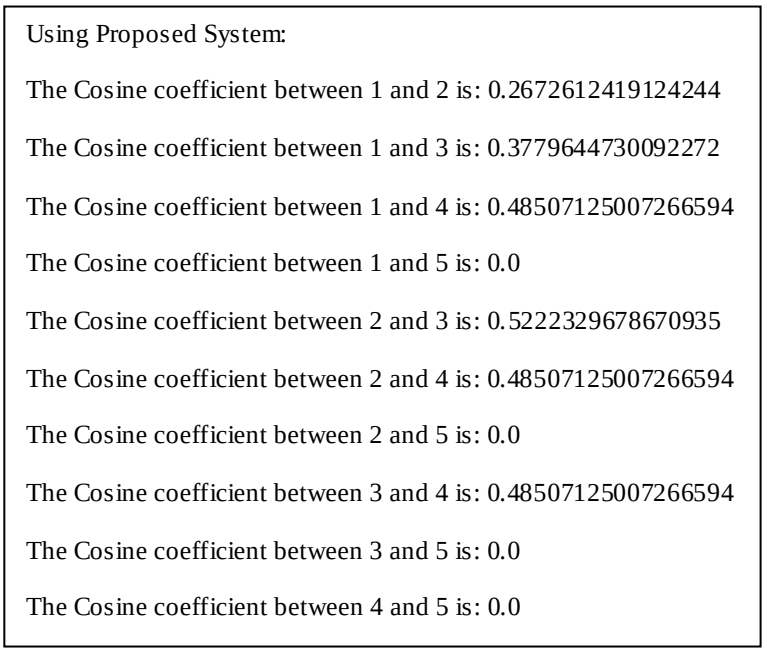

Fig 5: Result Using Proposed System.

@IJAERD-2015, All rights Reserved