AMSE JOURNALS –2014-Series: Advances B; Vol. 57; N° 2; pp 86-97
Submitted Oct. 2014; Revised Nov 11, 2014; Accepted Nov. 30, 2014

An Integrated and Efficient Approach to Measure Semantic Similarity between Short Sentences and Paragraphs

*A.Kavitha, **N.Rajkumar, ***S.P.Victor

*Manonmaniam Sundaranor University,
Tirunelveli, India, (aka.mca@rmkec.ac.in).
** Dept. of M.E Software Engg., Sri Ramakrishna Engineering College,
Coimbatore, India, (nrk29@rediffmail.com).
*** Dept. of MCA, St. Xavier College
Palayamkottai,Tirunelveli, India, (victorsp@rediffmail.com).

Abstract: Sentence similarity measures play an important role in text-related research and applications in areas such as Text Mining, Natural Language processing and Information Retrieval system. Similarity calculation for the short texts and paragraphs improve the retrieval effectiveness of the system. It is a complex concept which has been widely discussed in various domains. The target approach is to find that how the given text is semantically similar to another text. The proposed technique provides an efficient method to measure similarity for sentences and two short paragraphs based on the Similarity score. The scores indicate the Similarity at semantic level between two input text segments.

Keywords

Similarity Measure of short texts, Sentence similarity, String Similarity, phrases Similarity, natural language processing, word similarity, corpus based measures.

1. Introduction

The application of natural language processing presents a need for an effective method to compute the similarity between very short texts or sentences. In the field of Information retrieval systems, similarity calculation for the short texts and paragraphs improves the retrieval effectiveness of the system. Sentence similarity is used in the field of Text Mining [16]. It discovers the unknown information by repeatedly extracting information from a typically large amount of unstructured textual resources. Text mining also known as text data mining is widely used to discover knowledge from Text databases. It consist of large collections of documents from various sources, such as news articles, research articles, books, digital collection, e-mail messages and Web pages. A method to measure the similarity between short texts or sentences has many applications in natural language processing and related areas such as information retrieval and image retrieval from the Web [10]. The similarity measure of short text surrounding the images can achieve a higher retrieval precision than the use of whole document in which the image is embedded.

The Similarity mechanism has two terms, one is detection of similarities and another one is difference. The similarity is referred as synonyms or the meaningful information where as difference is antonym contradiction or the inconsistency information [1][3]. For our proposed task, given two input text area, we have to automatically find out a score that indicates their similarity at semantic level. The proposed technique provides an efficient method to find a

Similarity for the sentences and two short paragraphs based on the Similarity score. The scores indicate the Similarity at semantic level.

2. Related Work

Literature on measuring the similarity between documents has large number of publication. Similarity for the short texts has few publications Aminul Islam and Diana Inkpen [2]. The proposed system introduces a method for measuring the semantic similarity of texts using a corpus-based measure of semantic word similarity using a normalized and modified version of the LCS string matching algorithm. The method determines the similarity of two texts from semantic and syntactic information in terms of common-word order that they contain. Another word similarity measure SOC-PMI [1][11] is a new corpus-based method for calculating the semantic similarity of two target words. The method uses Point wise Mutual Information to sort list important neighbour words of the two target words. The words which are frequent in both lists are considered and their PMI values (from the opposite list) are aggregated to calculate the relative semantic similarity [8]. Corpus-based and Knowledge-based method [4] suggested a hybrid method for measuring the semantic similarity of texts by exploiting the information that can be drawn from the similarity of the component words. Specifically, they used two corpus based and six knowledge-based measures of word semantic-similarity, and combined the results to show how these measures can be used to derive a text-to-text similarity metric [5][9].

The main disadvantage of the previous similarity calculation method is domain dependency. Once the Similarity measure method designed for an application domain, it cannot be easily adapted to other domains. The vector-based document model methods are commonly used in Information Retrieval (IR) systems. The document most relevant to an input query is determined by representing a document as a word vector and queries are mapped to analogous documents in the document database via a similarity metric[6][13]. This technique relies on the assumption that more similar documents have more words in common but it is not always the case those texts with similar meaning necessarily share many words.

The corpus based approach can be divided into two methods: (LSA) latent semantic analysis and Hyperspace Analogues to Language (HAL) model. LSA (uses Singular Value Decomposition (SVD) to find the semantic representations of words by analysing the statistical relationships among words in a large corpus of text. High-dimensional linear association model analyses a large corpus of natural text and generate a representation that captures the similarity of words and text passages [12][14].

Point wise Mutual Information (PMI) and Information Retrieval PMI-IR is well known LSA approach. It measures the similarity based on context of a word which is composed of words co-occurring within a certain large corpus. Distributional measures use statistics from a large text corpus to determine how similar the contexts of two words. Another corpus-based method is HAL which applies lexical co-occurrence to produce a high-dimensional semantic space. A semantic space is a hole in which words are represented as points, and the location of each word along the axes is related to the word's meaning. Once the hole is constructed, a distance measure can be used to determine relationships between words. HAL was not as promising as LSA in the computation of similarity for short texts [7][15].

3. Proposed Work

The proposed system overcomes the limitations of the existing system through data pre-processing to information retrieval. String similarity is given more significance to provide an efficient similarity score. Hybrid approach combines the two different similarity functions using Root Mean Square method. Final calculation aggregates the selected maximum scores to find similarity of the given text. The proposed system architecture is shown in Fig. 1.

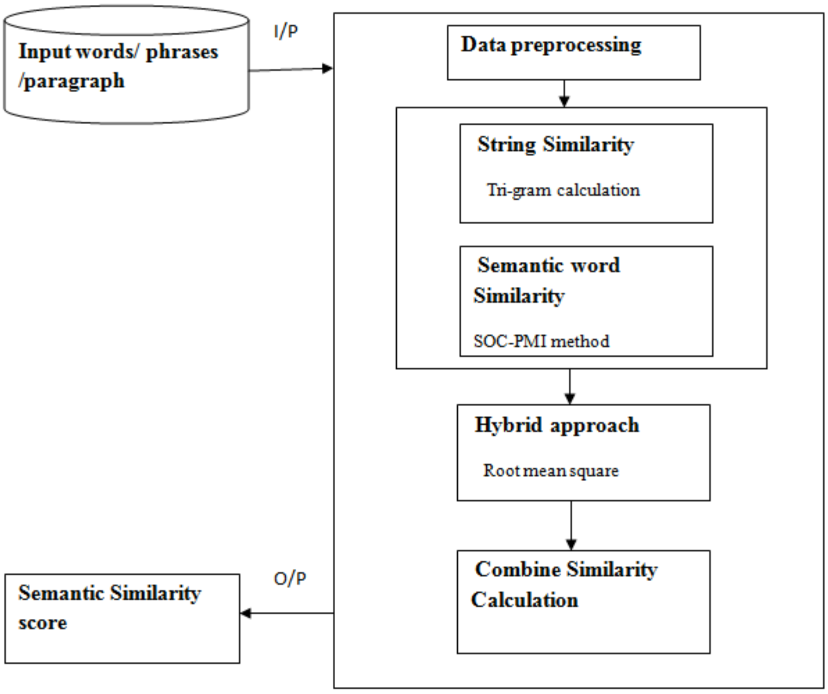

Fig. 1. Proposed System Architecture

3.1 Data Pre-Processing

The data pre-processing is an important task which consists of three phases namely, special character removal, stop words removal and case conversion. The data pre-processing is a process of Data cleaning. The data pre-processing helps to reduce the document size and reduce the comparison time. The first phase, removes all the special characters from all documents. The special characters list is shown in Fig. 2. The second phase removes all the stop words from the whole documents. Stop words are not a root words and few stop words are shown in Fig. 3 and the third phase converts the entire document to upper case to lower case.

!, @, # , . , " , \ , $, % , ^ , & , * , (,) , - , + , = , _ , { , } , [,], ; , : , | , <, >, ? , /, ~ , `, , , \

Fig. 2. Special Characters List

a, an, the , is , are , there, who, what, when, how, much, this, that,.. etc.

Fig. 3. Stop Words List

The user query terms are not necessary to compare with the stop words like a, as, but. .etc. To overcome this problem, the proposed method helps to reduce the searching time and identifying the root word and also help to reduce the inverted index size. This method is common for all text mining problems and operations.

3.2 String Similarity

The String similarity computes the score using Tri-gram approach. The existing methods are not giving much significance to String Similarity. Good similarity score can be obtained using string similarity. After preprocessing, the given data is taken to find the String Similarity score using Tri-gram function. Trigrams are a special case of the N-gram where N value is 3. It may also refer to a three-letter acronym. The word pairs are broken down into trigrams (Three letters sequences) to find string similarity. Each word is broken down into three letters contiguous sequence. The word sequence compared with other word to calculate string similarity score.

Example

The word is "eloquent" let as x and Tri-gram of the given word will be

tri(x)={elo,loq,oqu,que,uen,ent}

Fig. 4. Tri-gram of a word

String Similarity score can be calculated by comparing the three letter acronym of the one word with another word and using the Tri-gram mathematical formula to calculate the score. String Similarity score is required to compute to each and every word in the document. The common three letter acronym and the number of total acronym are used in the calculation. The string similarity using Tri-gram can be obtained using Eqn. (1).

$$\text{Simtri}(x,y) = \frac{1}{1 + |\text{tri}(x)| + |\text{tri}(y)| - 2 \text{ x } |\text{tri}(x) \cap \text{tri}(y)|} \quad (1)$$

Where x and y are two word pairs, tri(x) and tri(y) are trigram (three letter acronym count) for the word pairs and tri(x) ∩ tri(y) is common trigrams.

Walkthrough Example

1. x= **grandiloquent** and y= **eloquent**

tri(x) → {gra, ran, and, ndi, dil, ilo, log, ogu, gue, uen, ent}
tri(y) → {elo, loq, oqu, que, uen, ent}

tri(x) = 11 and **tri(y) = 6**
tri(x) ∩ tri(y) =5 (The common tri-gram are 5)

A string similarity score can be obtained using Equation (1).

$$\text{Simtri}(x,y) = \frac{1}{1 + |\ 11 + 6\ | - 2 \text{ x } |\ 5\ |}$$

Simtri(x,y) = 1/8
Simtri(x,y) = 0.125

The String similarity score for the word pairs “grandiloquent” and “eloquent” is **0.125.** The String function calculates the similarity score which can be used in the following hybrid approach to find the Semantic sentence similarity. The time taken to calculate string similarity is shown Fig. 5.

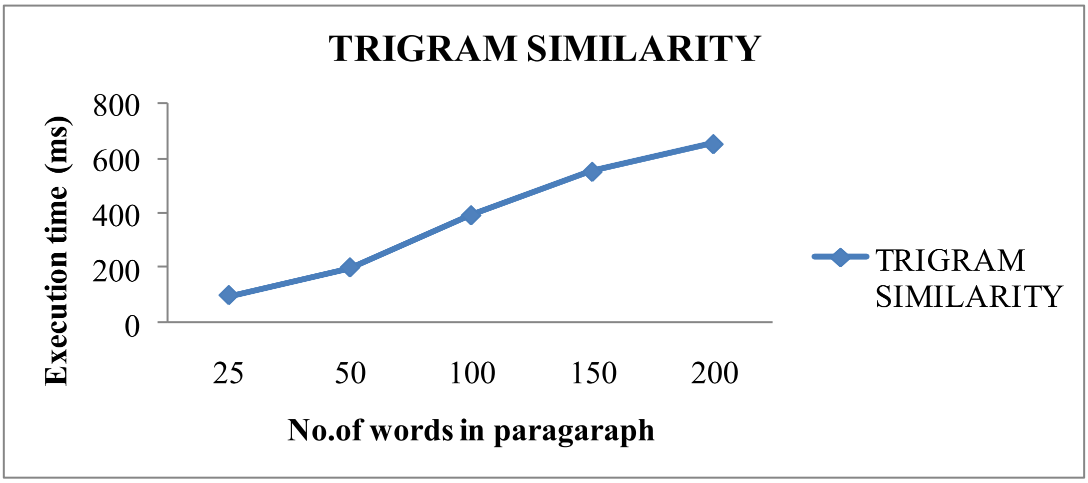

Fig. 5. Number of Words versus Time taken to generate String similarity score

3.3 Semantic Word Similarity

The preprocessed word pairs are taken to calculate semantic word similarity using Second order Co-occurrence Point wise Mutual Information. It is a corpus-based method for determining the semantic similarity of two target words. It uses Second Order Co-occurrence Point wise Mutual Information (SOC-PMI) to sort list essential neighbor words of the two target words and distinguish the words which are common in both lists and aggregate their PMI values from the opposite list to calculate the relative similarity score. Evaluation result shows that the method outperforms several competing corpus-based methods. This method focuses on measuring the similarity between two target words. After finding the similarity between all words in the document, the retrieval of similar information can be performed to user query also.

Walkthrough Example -2

Consider the word pairs *“apple”, “computer” and “car”, “automobile”* .

Apple and *computer* are different words but they are semantically related to each other. *Apple* company manufactures computers and there is so much relationship between the two words which is measured by co-occurrence of the word with another word. The following steps are involved to find the Semantic word similarity:

Step 1: $f^t(t_i) = |\{k:c_k = t_i\}|$, where i =1,2,…,n
Where $f^t(t_i)$ is how many times the type ti appeared in the entire corpus.

Step 2: $f^b(t_i,W) = |\{k: t_k = W \text{ and } (t_{k+j} = t_i \text{ or } t_{k-j} = t_i)\}|$, where i = 1,2,…,n

Where $f^b(t_i,W)$ is how many times word ti appeared with word W in the window.
The Point wise Mutual Information function works for only those words having $f^b(t_i,W)>0$.

Step 3 : $f^{pmi}(t_i,W) = \log_2 ((f^b(t_i,W) * m) / (f^t(t_i) * f^t(W))$
Where $f^t(t_i) * f^t(W) > 0$.

Step 4 : β_i =(log (ft(Wi)))2(log2(n)/δ), where i = 1,2 ,...,n.
Where δ is a constant and the value depends on the size of the corpus.

Step 5: Now for word W_1, a set of words X is sorted in descending order by their PMI Values with W_1 and the top-most β_1 Words having $f^{pmi}(t_i,W_1)>0$ are taken for further processing..

$$X = \{X_i\}, \text{ where } i =1, 2,\ldots, \beta_1$$

Similarly, for word W_2, Y is the collection of words and sorted in descending order by their PMI values with W_2 and the top-most β_2 Words having $f^{pmi}(t_i,W_2)>0$ are considered for next step.

$$Y= \{Y_i\}, \text{ where } i =1,2,\ldots, \beta_2$$

Step 6 : β-PMI summation function, for word W_1 is

$$f^\beta(W_1) = \sum(f^{pmi}(X_i,W_2))\Upsilon \quad \text{where } i = 1 \text{ to } \beta_1,$$
$$f^{pmi}(X_i,W_2)>0 \text{ and } f^{pmi}(X_i,W_1)>0$$

Similarly for word W_2

$$f^\beta(W_2) = \sum(f^{pmi}(X_i,W_1))\Upsilon \quad \text{where } i = 1 \text{ to } \beta_2,$$
$$f^{pmi}(X_i,W_1)>0 \text{ and } f^{pmi}(X_i,W_2)>0$$

Step 7 : Semantic similarity function between words W1 and W2 is

$$Sim(W_1,W_2) = (f^\beta(W_1)/ \beta_1)+(f^\beta(W_2)/ \beta_2)$$

The above steps of SOC-PMI function calculates the Semantic word similarity value which can be used in the following process to find the Semantic sentence similarity. The time taken to measure similarity between words using SCO-PMI is shown in Fig. 6.

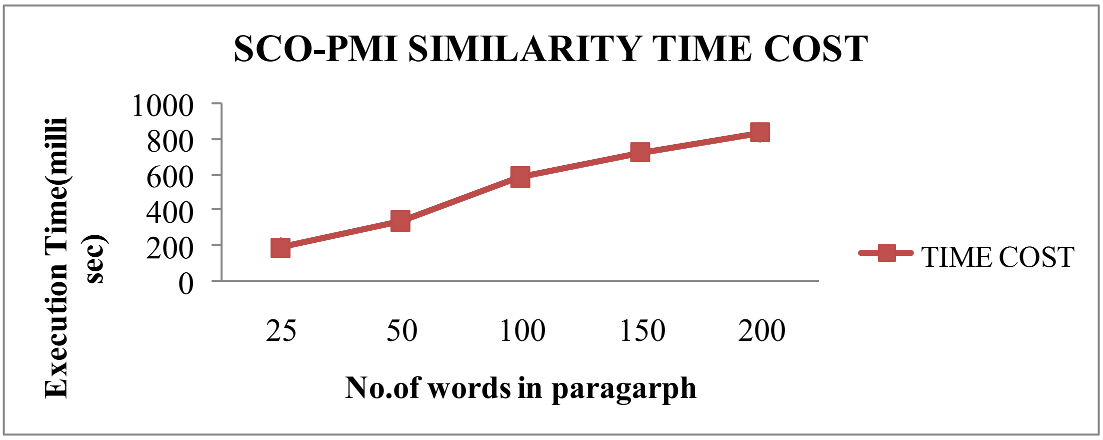

Fig.6. Number of Words versus Time taken to generate Word similarity score

3.4 Hybrid Approach

Hybrid method uses both corpus-based measure and knowledge-based measure of word semantic similarity to determine the text similarity. The combined method measures the semantic similarity of texts by exploiting the information that can be drawn from the similarity of the component word. The two similarity function such as String similarity and Semantic word similarity can be combined to calculate the better Similarity scores. The proposed hybrid approach combines the String Similarity scores and Semantic Word Similarity scores using Root Mean Square method and finds the final similarity score. Semantic Similarity of the given text can be obtained by selecting maximum values from the Root Mean Square scores list and aggregate them to find the final result.

The proposed approach applies the following procedure to uncover the finalized similarity scores:

Step 1: After pre-processing, Consider that the text P = $\{p_1, p_2, \ldots, p_m\}$ has m tokens of text. R = $\{r_1, r_2, \ldots, r_m\}$ has n tokens and $n \geq m$. Otherwise, switch P and R. It removes exactly matched words from the list.

Step 2: Create a list of String similarity and Semantic word similarity scores which obtained by Tri-gram and SOC-PMI method.

Step 3: Combine the scores using Root Mean Square method (RMS) and each word pairs of the two text can be added together. Root Mean Square method adds the two values based on the formula.

$$RMS = \frac{\sqrt{M1^2 + M2^2}}{2} \quad (2)$$

Where S is String Similarity score by trigram approach, W is Semantic word similarity score by SOC-PMI method. Both the scores S and W are the two different score for the same word.

Step 4: The RMS score matrix can be generated and the highest score will be selected and then removed from the list. Both row and column will be removed. Repeatedly doing this step until it has a single final scores value.

Where S is String Similarity score by trigram approach W is is Semantic word similarity score by SOC-PMI method. Both the scores S and W are the two different score for the same word

Step 5: The RMS score matrix can be generated and the highest score will be selected and removed from the list. Both row and column will be removed. Continue this process until it reaches a single final scores value.

Step 6: Selected scores can be used to calculate the final similarity of the documents. Aggregation functions are applied to find the final similarity score.

$$\text{Similarity} = \frac{(\delta + \sum_{i=1}^{|\rho|} \rho i\ )}{|\delta + \rho|} \qquad (3)$$

Where *ρ is* number of Maximum values in which is selected in RMS scores matrix, ρ_i is the selected Maximum values list i= 1,2,3…. and *δ* is the number of exact matches words.

Example: Two Sentences from different document

"Father's Day" is a celebration honouring fathers and celebrating fatherhood! Paternal bonds and the influence of fathers in society.”

Fig. 7. Sentences from document 1

“"Father’s Day" is celebration is to honour and respect fathers. On this day people buy gifts for their fathers for treat them to special things.”

Fig. 8. Sentences from document 2

Step1: After pre-processing, the sentences from document 1 and 2 are :

P → {fathers, day, celebration, honouring, celebrating, fatherhood, paternal, bonds, influence, society.}

R → {fathers, day, celebration, honour, respect, people, buy ,gifts, treat, special, things}

There are 3 words in the sentence P exactly matches with R. Remove the words {fathers, day, celebration} and set *δ to 3. A*fter removing the common words,

P → { honouring, celebrating, fatherhood, paternal, bonds, influence, society }

R → {honour, respect, people, buy gifts, treat, special, things}.

Step 2: Both the String similarity calculation scores and Semantic word similarity score can be calculated separately and listed .The Sting (S) and word(W) similarity score for the word pairs:

{Fatherhood, gifts} S =0.083 W = 0.656

Step 3: The listed scores are combined using RMS approach. The word pair *Fatherhood* and *gift* RMS score values calculated using Eqn. (2).

RMS String Similarity score = 0.467

Table 1: RMS score values for all the words

	Buy	Honour	things	gift	special	people	respect	Treat
fatherhood	0.469	0.709	0.467	0.467	**0.466**	**0.467**	0.708	0.467
honouring	0.472	**0.728(1)**	0.470	0.469	0.467	0.468	0.709	0.469
celebrating	0.468	0.709	**0.467**	0.467	0.466	0.466	0.708	0.467

paternal	0.472	0.710	0.468	**0.469(4)**	0.467	0.468	0.709	0.469
bonds	0.485	0.714	0.472	0.474	0.470	0.472	**0.711(2)**	0.474
influence	0.470	0.710	0.467	0.468	0.467	0.467	0.709	**0.468**
society	**0.474(3)**	0.711	0.469	0.470	0.468	0.469	0.710	0.470

Step 4: Select the maximum value and leave the row and column. Then, select next maximum values again and leave the rows and column. Continue this procedure until we reach single row and column. The selected maximum RMS values are ρi = {0.728, 0.711, 0.474, 0.469, 0. 468, 0 .467, 0.466}.

Step 5: The aggregation method computes the final similarity score by aggregating the RMS ρ values with the δ exactly similar values.

$$\text{Sim} = \frac{(\delta + \sum_{i=1}^{|\rho|} \rho i\ )}{|\delta + \rho|} \qquad \text{Using Eqn. (3)}$$

Where number of maximum values ρ is = 7 and

ρ_i = {0.728, 0.711, 0. 474, 0. 469, 0. 468, 0 .467, 0.466 }

Where the number of accurate matches are δ is = .3

Sim = (3+(0.728+ 0.711+ 0.474+ 0.469+ 0.468+ 0.467+ 0.466)) / |3+7|
= (3+(0.540)/10
= 6.78/10
Sim = **0.678**

The Final Similarity score for the two inputted text is **0.678**

The performance of SCO-PMI and Trigram is shown in Fig. 9.

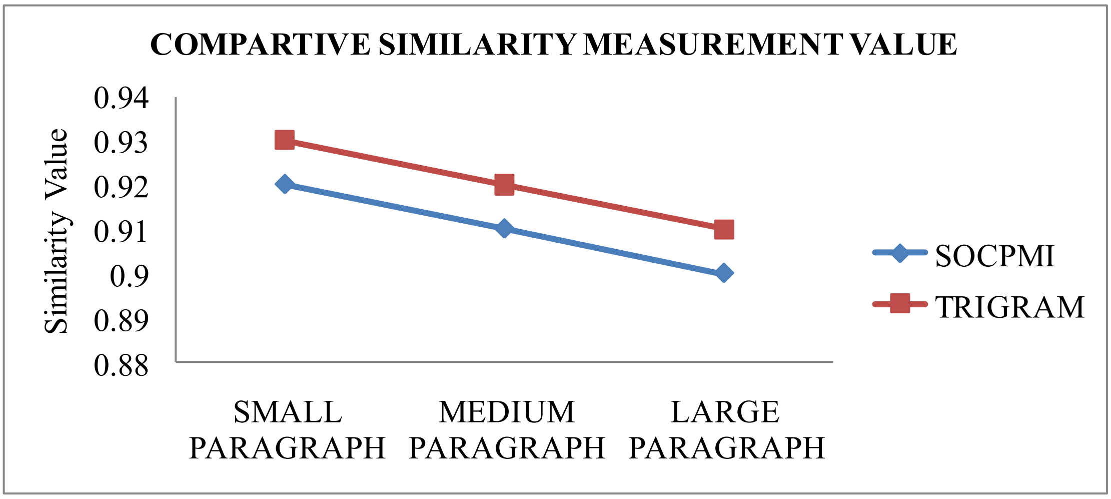

Fig. 9. Comparison graph between different input values

The time taken to measure similarity between words using the hybrid approach is represented in Fig. 10.

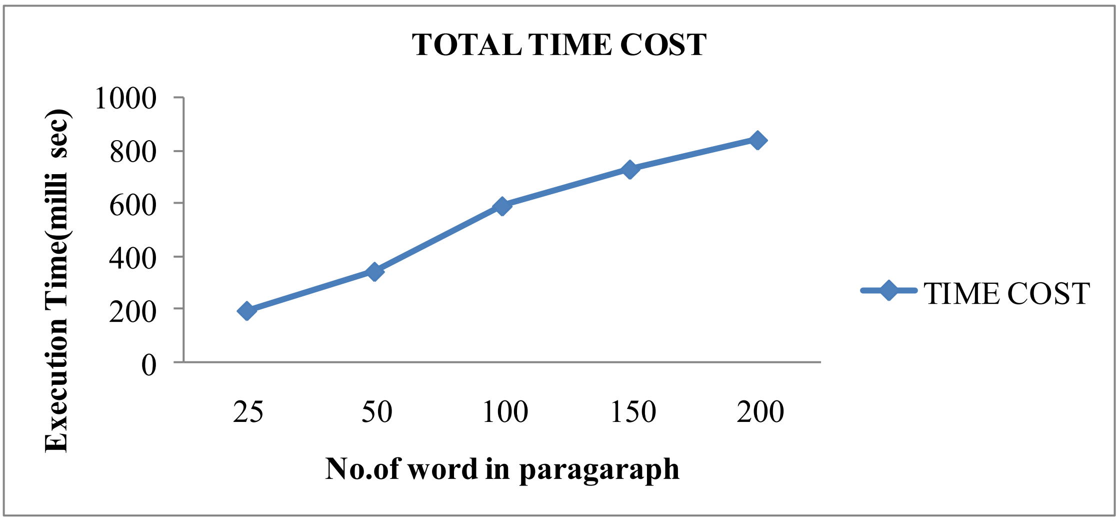

Fig. 10. Number of sentences versus total time cast

4. Performance Evaluation

This system is very informative because the similarity values for all input documents are saved for further processing. The Similarity calculation provides better scores and it can be achieved by hybrid approach. More accurate documents scores can be calculated with the help of String Similarity. It provides better scoring methods. Earlier approaches can only find the similarity of the documents or larger corpus. This method mainly focuses on short texts. Time taken to compute the similarity for the short sentences and phrases can be effectively performed in our approach.

The data preprocessing reduces the document size and short list the root words for the faster calculation. Hence, the similarity calculation time can be reduced. Combining the two different similarity calculations with the help of Root Mean Square method provides a better similarity scores. Word Similarity can be directly calculated with help of two approaches and it skips the preprocessing for reducing the time.

5. Conclusion

The Proposed approach measures the Semantic Similarity of the given input successfully with the help of various similarity calculation methods. Similarity calculation of word pairs and short sentences can be automatically generated. Hence, it improves the retrieval effectiveness of the system**.** The simplified approaches are used in the similarity calculation. String is given much importance in our system. String Similarity utilises Tri-gram approach which is easier to calculate the String similarity score and consumes lesser time to calculate the score. Semantic word similarity of the words successfully calculated with the help of SOC-PMI method which measures the similarity score using co-occurrence neighbouring words. The proposed Hybrid approach successfully combines the two similarity function with the help of Root Mean Square calculation. The calculation provides better scores that should be listed. Effective approaches take the maximum values from the list and calculate the score with the help of aggregation function. It comparatively provides better similarity scores than existing methods.

The future methods compare the base or querying sentences with multiple documents to find the similar documents with the help of sentence similarity approach. For example to search the entire documents to find similar documents, it can be easily performed with abstracts. Finding similarity scores of abstract of multiple documents, list out the similarity scores which can be helpful in the field of Information retrieval system.

References

[1] Islam A and Ink pen (2006), "Second order co-occurrence PMI for determining the semantic similarity of words", Proc. International Conference on Language Resources and Evaluation, pp.1033-1038, Genoa, Italy.

[2] Islam A and Inkpen D, Kiringa I (2008), "Applications of corpus-based semantic similarity and word segmentation to database schema matching", VLDB Journal, Vol.17, No.5, pp.1293-1320.

[3] Dekang Lin (1998), "An Information-Theoretic Definition of Similarity", Proc. 5th International Conference on Machine Learning, pp.296-304, San Francisco, USA, 1998.

[4] Mihalcea, R. and Corley C., And Strapparava C (2006), "Corpus-based and knowledge based measures of text semantic similarity", Proc. 21st International Conference on Artificial Intelligence, Vol.1, pp.775-780, AAAI Press.

[5] Bollegala, D, Matsuo Y and Ishizuka, M. (2011), "A Web Search Engine-Based Approach to Measure Semantic Similarity between Words", Knowledge and Data Engineering, Vol. 23, No.7, pp.977-990, 2011.

[6] Achananuparp, P, Hu, X. and Xiajiong, S. (2008), "The Evaluation of Sentence Similarity Measures", Proc. 10th international conference on Data Warehousing and Knowledge Discovery, pp.305-316, Springer-Verlag Berlin, Heidelberg.

[7] Choi, D., Ko B., Lee, E. and Hwang, M. (2012), "Automatic Evaluation of Document Classification using N-gram Statistics", International Conference on Network-Based Information Systems, pp.739-742, Melbourne.

[8] Bollegala, D.,Matsuo, Y. And Ishizuka,M. (2007), "Measuring semantic similarity between words using web search engines", Proc. 16th International Conference on World Wide Web, pp. 757–766, New York.

[9] Aziz, M and Rafi, M. (2010), "Sentence based Semantic Similarity Measure for Blog-Posts", 6th International Conference on Digital Content, Multimedia Technoloy and its Applications, pp.69-74, Seoul.

[10] Naresh Kumar Nagwani (2011), "A Frequent Term and Semantic Similarity based Single Document Text Summarization Algorithm", International Journal of Computer Applications, Vol. 17, No.2, pp.36-40.

[11] Gerlof Bouma, (2009),"Normalized (Pointwise) Mutual Information in Collocation Extraction", Proc. of the Biennial GSCL Conference, Potsdam.

[12] Guangchuang Yu,(2012), ”GO-terms Semantic Similarity Measures”, Bioinformatics, Vol.26, No.7, pp.976-978.

[13] Yuhua Li et al.(2006), “Sentence Similarity Based on Semantic Nets and Corpus Statistics”, Knowledge And Data Engineering, Vol. 18, No. 8.

[14] Prodromos Malakasiotis and Ion Androutsopoulos (2007), “Learning Textual Entailmentusing SVMs and String Similarity Measures”, Proc. of the Workshop on Textual Entailment and Paraphrasing, pages 42–47, Prague.

[15] Aminul Islam, Evangelos Milios, and Vlado Keselj (2010), ”Text Similarity using Google Tri-grams” , Proc. of 25th Canadian conference on Advances in Artificial Intelligence, pp. 312-317, Toronto, Canada.

[16] Jiawei Han, Micheline Kamber (2006) ”Data Mining: Concepts and Techniques”, Second edition. University of Illinois at Urbana-Champaign, vol. pp.614-640.

[17] Benslimane, S. M, Malki M.and Rahmouni M. K. (2007), “From data-intensive web sites into ontology-based semantic web: a reverse engineering approach”, AMSE journals; Advances D: Computer Science and Statistics, Vol. 12, No.1-2, pages 18-40.