International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 13
October 2017

An Evaluation of Short Text Similarity Matching from Text Pairs

Rangam sreelatha & Smt V.Priya Darshini
[1]M.Tech Student, Department of Computer Science and Technology, SRKR Engineering College, Bhimavaram, West Godavari, Andhra Pradesh, India.
[2]Assistant Professor, Department of Computer Science and Technology, SRKR Engineering College, Bhimavaram, West Godavari, Andhra Pradesh, India.

Abstract

Retrieving semantic similar short texts is a crucial issue to many applications. Cosine similarity coefficient, a pace that's generally found in clustering, measures the similarity between groups. Jaro-Winkler approach to use Cosine's similarity co-efficient increases time complexity greatly. Hence Cosine's similarity coefficient is replaced with Jaro Winkler similarity measure to obtain the cluster similarity matching. Jaro-Winkler does a better job at working the similarity of strings because it takes order of characters into account using positional indexes to estimate relevancy. It is presumed that Jaro-Winkler performance regarding one-to-many data linkages offers an enhanced performance in contrast to Cosine driven CACT's workings. So we propose to replace Cosine's similarity coefficient with Jaro Winkler similarity measure to obtain the similarity matching of text pairs. For Similarity-matching, we evaluated the performance of Jaro-Winkler, CACT's, WordNet-based and Wikipedia-based. After using different model to test these similarity metrics, we found that Jaro-Winkler performed better than CACT's, WordNet-based and Wikipedia-based. First, we explored record linkage similarity metrics to determine which are suitable for predicting Short text. In current scenario, we are .

Index Terms—Short text, Text similarity, Jaro–Winkler distance;

I. Introduction

Short Text matching is used heavily in areas such as de-duplication detection and text mining. Most existing work that computes the similarity of two strings only considers Jaro Winkler similarity measure e.g., number of common letter in the string. Although there are many difference cases, like small string or long string can identify. For ex:- 'sum' is short form of 'summary', 'summer'. Such equivalence information can help us identify semantically similar short string by using Jaro-Winkler approaches.

In this paper, we study related to approximate string matching. Similarity measure between two strings based on the inputs. The Traditional similarity functions cannot be easily extended to handle the short text in similarity matching.

In this paper, we investigate the Jaro-Winkler's approach of short texts retrieval, which is important to many applications. Cosine similarity coefficient, a pace that's generally found in clustering, measures the similarity between groups. Jaro-Winkler approach to use Cosine's similarity co-efficient increases time complexity greatly. Hence Cosine's similarity coefficient is replaced with Jaro Winkler similarity measure to obtain the cluster similarity matching. Jaro-Winkler does a better job at working the similarity of strings because it takes order of characters into account using positional indexes to estimate relevancy. It is presumed that Jaro-Winkler driven performance regarding one-to-many data linkages offers an enhanced performance in contrast to Cosine driven CACT's workings. So we propose to replace Cosine's similarity coefficient with Jaro Winkler similarity measure to obtain the similarity matching of text pairs.

The Jaro-Winkler distance is a measure of similarity between two short strings. It is a variant of the Jaro distance metric and mainly used in the area of record linkage (Similar data detection). The higher the Jaro-Winkler distance for two strings is, the more similar are. The Jaro-Winkler distance metric is mainly implemented and best suited for short strings such as person names. The matching score is normalized such that 0 defined for **no similarity** and 1 is defined **exact match**. Jaro-Winkler string distance algorithm estimates a difference or, roughly, a number of character replacements if takes to convert one string to another.

In many domains, the orders in which properties occur are just as important as their existence; this is certainly true for strings. Because of this, Jaro-Winkler is a lot more efficient than Cosine Similarity.

The Jaro-Winkler is a combination of Similarity Coefficient and Jaro Distance; I don't know if Winkler worked in tandem with Jaro. Winkler added an important component to the Jaro distance algorithm that weighted or penalized strings based on their similarity at the beginning of the string (the prefix). The algorithm for the Jaro-Winkler distance is a slight complicated, in

part because it requires iteration over the short string to specify if a non-matching character fits within a predetermined "window" of the other string. For instance, the strings "Martha" and "marhta" are considered a complete match because the transposed "th" and "ht" are within 2 characters of each other.

Jaro–Winkler:-

The Jaro–Winkler distance is a string metric for measuring the **Edit Distance** between two text pairs (source text and destination text). The Jaro–Winkler distance uses a prefix scale Pr, which gives more favorable ratings to strings that match from the beginning for a set prefix length **Len**.

Here, Jaro–Winkler distance for two strings is, the more similar the strings are not. The matching score is normalized such that 1 equates to no similarity and 0 is an exact match. The Jaro–Winkler similarity is given by 1 − Jaro–Winkler distance.

Jaro-Winkler distance will determine the distance between two strings, **S1** and **S2**. In the formula below, **m** is the number of matching characters; **t** is half the number of transpositions.

Jaro–Winkler distance uses a prefix scale **Pr**, which gives more favorable ratings to strings that match from the beginning for a set prefix length **Len**. Given two strings $\mathbf{S_1}$ and $\mathbf{S_2}$, their Jaro–Winkler distance $\mathbf{d_w}$ is:

$$\mathbf{d_w = d_j + (Len*Pr(1-d_j))},$$

Where:

- $\mathbf{d_j}$ is the Jaro distance for strings S_1 and S_2
- **Len** is the length of common prefix at the start of the string up to a maximum of four characters.
- **Pr** is a constant scaling factor for how much the score is adjusted upwards for having common prefixes. Pr should not exceed 0.25, otherwise the distance can become larger than 1. The standard value for this constant in Jaro-Winkler's work is Pr=0.1;

Although often referred to as a distance metric, the Jaro-Winkler distance is the mathematical sense of term because it does not obey the triangle inequality.

II. Function for Jaro-Winkler Similarity Distance Measure

Step:1 Similarity of first few letters is most important.

1) Let 'pr' be the length of the common prefix of S_1 and S_2.
2) $Sim_{winkler}(S_1, S_2) = Sim_{jaro}(S_1, S_2) + (1- Sim_{jaro}(S_1, S_2))$ pr/10 = 1 if common prefix is >=10

Step 2:- Longer Strings with even more common letters

1) $Sim_{winkler, long}(S_1, S_2) = Sim_{winkler}(S_1, S_2) +(1- Sim_{winkler}(S_1, S_2))$ [c-(pr+1)/ (|s1|+|s2|-2(pr-1)]

Where **C** is overall number of common letters, Apply only if: Long Strings: min(|S1|,|S2|)>=5

Two additional common letters: **C-Pr** >= 2

At least half remaining letters of shorter string are common **C-Pr**>= min(|S1|,|S2|)-pr/2

Edit distance:-

Edit Distance is a best way of quantifying how dissimilar two strings (e.g., words) are to one another by counting the minimum number of operations required to transform one string into the other. Edit distances find applications in natural language processing, where automatic spelling correction can determine candidate corrections for a misspelled word by selecting words from a dictionary that have a low distance to the word in sentence.

Example:-

Take String1 is '**DWAYNE**' and String 2 is '**DUANE**' for example. According to the proposed technique we should end up with the following results:

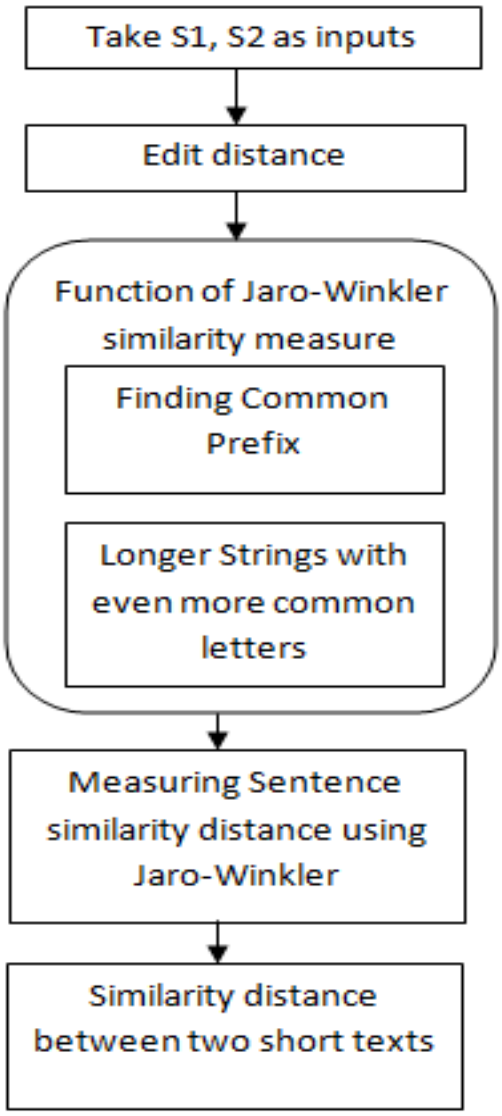
d_j = 1/3 ((4/6)+ (4/5) + ((4-0)/4)) = 0.822

The lesser of the two strings, DUANE, is 5 characters long. When we set minLen to 5 in (minLen/2 + minLen%2 is 3) , we get a radius of 3.

Next we find the similar characters in each direction. Comparing at DWAYNE to DUANE with a radius of 3. We can see that D, A, N and E will match, giving us a S_1 = "DANE". Identically, Comparing at DUANE to DWAYNE, we get exactly the same thing. That is S_2 = "DANE" as well. As you might have estimated by the fact both sets of similar strings are the same, we have zero transpositions in this example. So now, we just plug in the numbers and get 1/3* (4/6 + 4/5 + (4-0)/4) = 0.822.

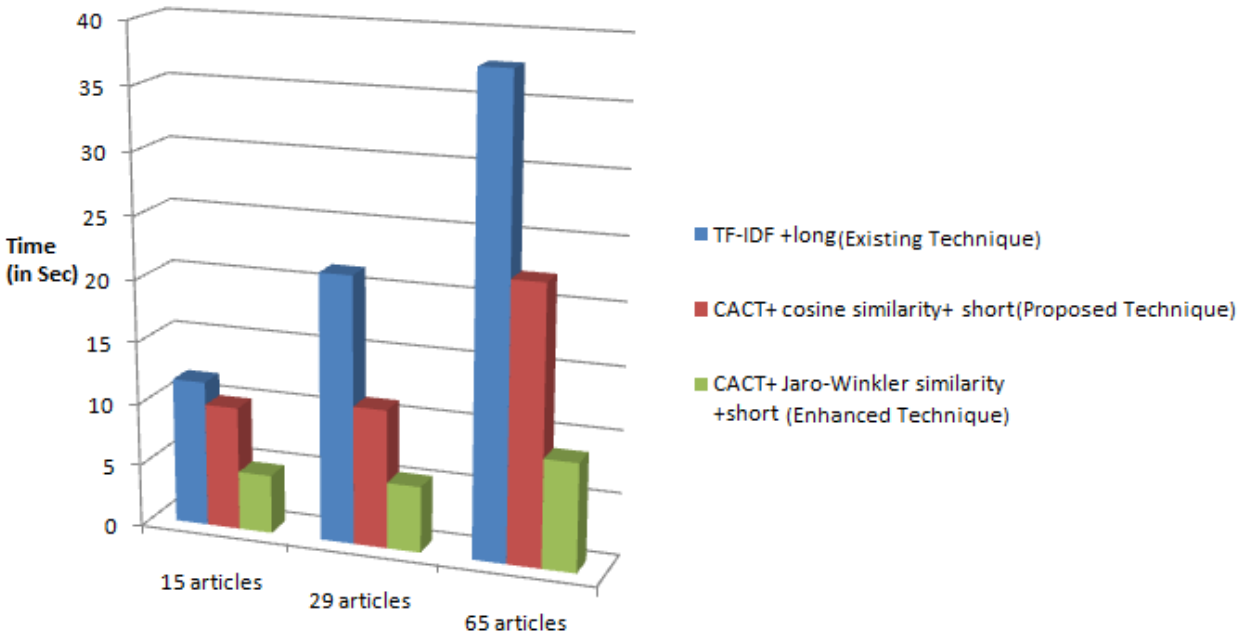
III. Current Article Work Flow

Fig:- Jaro-Winkler process flow



IV. Result Analysis

	15 articles	29 articles	65 articles
TF-IDF +long	11.7sec	21.4sec	37.7sec
CACT+ cosine similarity measure+ short	9.9sec	11.1sec	22.2sec
CACT+ Jaro-Winkler similarity measure +short	4.7sec	5.3sec	8.7sec



In existing system, text retrieval methods, such as TF-IDF and CACT with cosine similarity measure, have made significant achievements in most text-related applications. Recently, propose a new information retrieval mechanism called Jaro-Winkler Similarity measure. Compared with traditional methods, such as TF-IDF and CACT with cosine similarity measure, their Jaro-Winkler measure model achieves comparable retrieval performance.

V. Conclusion

We propose to replace Cosine's similarity coefficient with Jaro Winkler similarity measure to obtain the similarity matching of text pairs. For Similarity-matching, we evaluated the performance of Jaro-Winkler, CACT's, WordNet-based and Wikipedia-based. After using different model to test these similarity metrics, we found that Jaro-Winkler performed better than CACT's, WordNet-based and Wikipedia-based.

VI. References

[1] M. Salami and T. D. Hellman, "A web-based kernel function for measuring the similarity of short text snippets," in Proc. 15th Int. Conf. World Wide Web, 2006, pp. 377–386.

[2] W. tau Yam and C. Meek, "Improving similarity measures for short segments of text," in Proc. 22nd Nat. Conf. Art if. Intel., 2007, pp. 1489–1494.

[3] D. Sheen, R. Pan, J.-T. Sun, J. J. Pan, K. Wu, J. Yin, and Q. Yang, "Query enrichment for web-query classification," ACM Trans. Inf. Syst., vol. 24, no. 3, pp. 320–352, 2006.

[4] C. Erlbaum, WordNet: An Electronic Lexical Database. Cambridge, MA, USA: MIT Press, 1998.

[5] X. Hub, N. Sun, C. Zhang, and T.-S. Chua, "Exploiting internal and external semantics for the clustering of short texts using world knowledge," in Proc. 18th ACM Conf. Inf. Know. Manage., 2009, pp. 919–928.

[6] S. Bannered, K. Raman than, and A. Gupta, "Clustering short texts using Wikipedia," in Proc. 30th Annul. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2007, pp. 787–788.

[7] E. Gabrilovich and S. Marko itch, "Computing semantic relatedness using Wikipedia-based explicit semantic analysis," in Proc. 20th Int. Joint Conf. Art if. Intel., 2007, pp. 1606–1611.

[8] E. Gabrilovich and S. Marko itch, "Feature generation for text categorization using world knowledge," in Proc. 19th Int. Joint Conf. Art if. Intel., 2005, pp. 1048–1053.

[9] W. Wu, H. Li, H. Wang, and K. Q. Zhu, "Probate: A probabilistic taxonomy for text understanding," in Proc. Int. Conf. Manage. Data, 2012, pp. 481–492.

[10] Y. Song, H. Wang, Z. Wang, H. Li, and W. Chen, "Short text conceptualization using a probabilistic knowledge base," in Proc. 22nd Int. Joint Conf. Art if. Intel., 2011, pp. 2330–2336. [11] D. Kim, H. Wang, and A. H. Oh, "Context-dependent conceptualization," in Proc. 23rd Int. Joint Conf. Art if. Intel., 2013, pp. 2654–2661.

[12] B. Stein, "Principles of hash-based text retrieval," in Proc. ACM 30th Annul. Int. Conf. Res. Develop. Inf. Retrieval, 2007, pp. 527–534.

[13] R. Salakhutdinov and G. E. Hinton, "Semantic hashing," Int. J. Approx. Reasoning, vol. 50, no. 7, pp. 969–978, 2009.

[14] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," Neural Comput., vol. 14, no. 8, pp. 1771–180

0, 2002.