opción

Revista de Antropología, Ciencias de la Comunicación y de la Información, Filosofía, Lingüística y Semiótica, Problemas del Desarrollo, la Ciencia y la Tecnología

Año 35, 2019, Especial N° 19

Revista de Ciencias Humanas y Sociales
ISSN 1012-1587/ ISSNe: 2477-9385
Depósito Legal pp 198402ZU45

Universidad del Zulia
Facultad Experimental de Ciencias
Departamento de Ciencias Humanas
Maracaibo - Venezuela

Opcion, Año 35, Nº Especial 19 (2019):*2899-2921*
ISSN 1012-1587/ISSNe: 2477-9385

Modifying Jaccard Coefficient for Texts Similarity

Sura Mahmood Abdullah[1], Sura Mazin Ali[2], Mohammed Abduljaleel Makttof[3]

[1]Department of Computer Sciences, University of Technology - Iraq, Baghdad; 110050@uotechnology.edu.iq, [2]College of Political Sciences, AlMustansiriyah University, Iraq, Baghdad; suraaz2007@yahoo.com, [3]AlTurath University Collage, Iraq, Baghdad; cabdeljaleelmohammed@gmail.com

Abstract

Calculating similarities between texts written in any language remains one of the extremely important challenges encounter natural language processing. This paper presents the modified Jaccard similarity coefficient for the texts; the main aim from this modification is to count the number of similar sentences between texts instead of counting the number of similar words between them as in previous works. This modification is applied by produced an equation which combining the Jaccard coefficient and the similarity coefficient, furthermore, two criteria are employed in the proposed equation; where the first one is multiplied by the Jaccard coefficient and the second criterion is multiplied by the similarity coefficient. The objective of these criteria is to keep the similarity degree between 0 and 1. The experimental results are logical, in which the similarity degree of the proposed equation increased approximately 3% on Jaccard coefficient degree when chosen texts from the same class, while it became less than the Jaccard coefficient degree when chosen texts from the various classes.

Key words: Text Mining, Text Similarity, Lexical Similarity, String-Based Similarity, Jaccard Coefficient.

Modificación del coeficiente de Jaccard para similitud de textos

Calcular similitudes entre textos escritos en cualquier idioma sigue siendo uno de los desafíos extremadamente importantes que enfrenta el procesamiento del lenguaje natural. Este artículo presenta el coeficiente de similitud de Jaccard modificado para los textos; El objetivo principal de esta modificación es contar el número de oraciones similares entre textos en lugar de contar el número de palabras similares entre ellos como en trabajos anteriores. Esta modificación se aplica produciendo una ecuación que combina el coeficiente Jaccard y el coeficiente de similitud, además, se emplean dos criterios en la ecuación propuesta; donde el primero se multiplica por el coeficiente de Jaccard y el segundo criterio se multiplica por el coeficiente de similitud. El objetivo de estos criterios es mantener el grado de similitud entre 0 y 1. Los resultados experimentales son lógicos, en los que el grado de similitud de la ecuación propuesta aumentó aproximadamente un 3% en el grado de coeficiente de Jaccard cuando se eligieron textos de la misma clase, mientras menor que el grado de coeficiente de Jaccard cuando se eligen textos de varias clases.

Palabras clave: minería de texto, similitud de texto, similitud léxica, similitud basada en cadenas, coeficiente Jaccard.

Introduction

Text mining alludes to the process of extracting and discovering beneficial information from unorganized texts. Computing the similarity between texts is an important element in different tasks like text summarization, machine translation, text clustering, text categorization and others (Su and Seoung, 2017).
Text similarity consists of two types; which are the lexical and semantic similarity. In lexical similarity, the similarity based on matching the characters between words or statements. While in semantic similarity, the similarity based on the meaning, for e.g. “Support Vector Machine” and “SVM” are both similar to each other semantically.

This paper focuses on lexical similarity, which is a measure uses to count the similarity degree of a set of words from two particular texts by the

characters matching process. A lexical similarity of 1 (means 100%) means full overlap between words, while Lexical similarity of 0 means that there is no common word in a particular text. Lexical similarity is measured by using String-Based algorithms; which are classified to character-based similarity and term-based similarity which was used in this paper (Nitesh et al., 2015).

Jaccard Similarity is one of the techniques that measures the lexical similarity depending on the term-based similarity. Uusually, the Jaccard coefficient counts the similarity degree between the two texts by computing the frequency of the shared words between them. (Suphakit et al., 2013).

The Jaccard coefficient counts only the similarity between words without taking into consideration the similarity between the sentences which increases the ratio of similarity. So, the Jaccard coefficient was modified in this paper to increase the real similarity and reduce the similarity error ratio which may occur as a result of using the Jaccard coefficient alone.

The remnant of this paper is systematized as follows: section 2 explains some previous works, section 3 presents the measures of the text similarity, section 4 shows the string-based similarity in detail, section 5 demonstrates the proposed system, results and experiments are displayed in section 6, and finally, the conclusions are clarified in section 7.

Previous Work

The text similarity concept is becoming most common in natural language processing. Therefore, several studies are carried out to show the various methods which are used to measure the similarity degree between the words, sentences, paragraphs, and texts.

A study was conducted in (Suphakit et al., 2013) by Suphakit N. and et al. where they proposed a method to measure the similarity between words using a Jaccard coefficient with Prolog programming language. the performance of this proposed method was calculated using F-measure, recall, and precision. The performance measures proved the ability of the proposed method to handle high consistently when failure and mistake spelling occurred.

In addition, in (Neha et al., 2014) Neha A. and et al presented a comparative study between Jaccard coefficient and cosine similarity with regard to the complexity of time and result pertinent to the query to categorize web documents depending on their field. The two techniques are best-known techniques to detect the similarity between the two documents. The time needed to generate a cluster using the cosine similarity measure is less than the jaccard coefficient because of the mathematical equation used to count the similarity between the documents. Furthermore, the Jaccard coefficient takes much time when matching all the words of one document to another document words. Through applying the paradigm of the Jaccard coefficient and cosine similarity, the cluster created by cosine similarity gives the most precise and pertinent result as compared to the Jaccard Coefficient.

Moreover, in (Sheetal and Sushma, 2010) Sheetal A. T. and Sushma S. N. presented a new approach which calculated the semantic similarity between terms using keywords got from Wikipedia extracts with the five various similarity measures of association which are (simple matching, Dice, Jaccard, Overlap, Cosine coefficient). The Porter algorithm is used by the new approach to remove the suffix from extracts; after that, the Lohan's idea is applied to detect the important words from the extracts which processed in advance. Finally, the five measures performance is evaluated using a standard data set from Miller and Charle. The result shows that the extracts in Wikipedia have an important impact on the precision of semantic similarity measure between words.

Furthermore, in (Praveenkumar and Harinarayana, 2018) Praveenkumar V. and N.S. Harinarayana submitted two approaches (Jaccard similarity test and statistical technique that is called term frequency-inverse document frequency (TF-IDF)) which are significant of the information retrieval process. These approaches are used in 'relevancy ranking' of words used in the whole text of the digital resource. The popular words are found in the whole text of the essay and social tags. The results display that it's possible to set the "weight" for keywords to improve results and also to identify important tags that the user assigns. The Jaccard coefficient test was depended to comprehend the word similarity between whole text words of the essay and the social tags.

A different study was presented in (Vikas and Vivek, 2013) by Vikas T.

and Vivek J. where they perform a comparative analysis to discover the most appropriate document for a particular group of keywords using three measures for similarity (DICE, Cosine, and JACCARD). This is done utilizing the genetic algorithm, where the best value for the fitness function is the rate of 10 operations of the same code for a constant number of iterations. The similarity measure is found for a set of documents returned for a particular search query from Google, and then fitness values are computed using similarity measures. in this study, 10 diverse generations averaged for each search query by operating the program 10 times for the constant value of prospect of crossover Pc = 0.7 and the prospect of Mutation Pm = 0.01 mutation.

Text Similarity Measures

The similarities between words can be measured in two ways lexical and semantic. when the words have a similar sequence of characters then these words are similar in lexicon, but when the words have the same subject then these words are similar in semantic. Usually, the lexical similarity can be measured by String-Based algorithms; but the semantic similarity is measured by the Knowledge-Based and Corpus-Based algorithms (Vijaymeena and Kavitha, 2016).

String-Based Similarity

String-Based algorithms are used to measure lexical similarity; String-Based measures work on a series of strings and characters composition. String metric is used to measure the similarity and difference between the text strings. This metric is used to match the string or comparison but it is approximate. String-Based algorithms are splitted into two types Character-based and Term-based similarity (Wael and Aly, 2012).

A. Character-based similarity

The counting process is used to count the distance between any two strings. The minimum number of processes desired converting one string into another and the processes like insertion, deletion, or replacement of one character and alteration of two characters that are adjoining are determined by Damerau-Levenshtein (Vijaymeena and Kavitha, 2016). The character-based similarity has many algorithms to measure the similarity

such as Smith-Waterman, N-gram, Damerau–Levenshtein, Jaro–Winkler, Needleman–Wunsch, Jaro and Longest Common Substring (LCS) (Khuat et al., 2015).

B. Term-based similarity

The absolute distance is calculated as the distance that will be visited to move from one point of data to another when follows the networked path. The absolute distance between elements is the total variations of their corresponding components Wael and Aly, 2012).

The term-based similarity measurement contains many algorithms like Block Distance, Cosine similarity, Dice's coefficient, Euclidean distance, Matching Coefficient, Overlap coefficient and Jaccard coefficient (Khuat et al., 2015).
Jaccard similarity coefficient uses to calculate similar words between the two texts by dividing the number of intersecting words between two texts on the union of all words of the two texts as shown below in equation 1 (Lisna, 2016).

$$J(A,B) = \frac{A \cap B}{A \cup B} \quad \ldots\ldots\ldots\ldots\ldots (1)$$

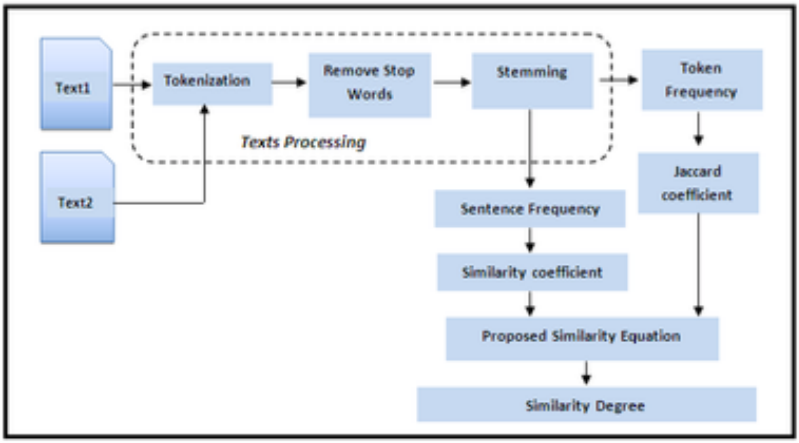
Where J(A,B) illustrates the similarity degree between text A and text B.

The modifying on the Jaccard coefficient includes inserting both of the number of similar sentences and two criteria which are α and β respectively; each of them has a weight. The proposed equation ensures a logical examination of the texts depending on the similar words according to the Jaccard coefficient and on the number of similar sentences according to the proposed equation.

The Proposed System

The system consists of three main stages representing the foundation of the system work. These stages are (text processing, frequency computation, and statistical methods) integrated together to give logical results in calculating the similarity between the two texts. Fig.1 shows the stages of the proposed system which goes through them to compute the similarity degree.

Figure 1. Proposed System Framework



A. Text Processing

The system deals with short and long English texts. After loading the text to the proposed system, the processing stage is beginning which considered a primary step in the system. This stage composed of three processes which are tokenization, stop words removal and stemming.

Tokenization is a task of chopping up any text into a set of tokens (words) and throwing away punctuation and other unwanted characters.

Stop Words Removal is a set of recurring tokens that show in each text like pronouns (they, we, you) and conjunctions like (for, and, while) and etc. These tokens must be deleted from the text because they don't have any influence on the similarity process. The stop words also include special characters and numbers which also must be deleted from the text.
Stemming is a task of deleting any additions from token and restoring it to its root. The additions mean affixes (prefixes and suffixes) that should be deleted from tokens at this step which represents an important step in the system. The stemming process is used to improve the performance of the system by decreasing the diverse forms of a token in the token space under the root of that token.

Without removing the stop words or using the stemming process or both will reduce system performance because the stop words constitute most

words in any text and when not removed from the text, it increases the complexity of text processing, especially when using long texts. Usually, the stemming process reduces the text size when placing the various forms of token under its root. When this process is ignored, the results are logically affected. So system performance will increase when the stop words removal and the stemming process are executed.

B. Frequency Computation

Token Frequency (TF) is a measure of how many times a certain token (stem of each token) occurs in a text.

TF= number of occurrences of a token (t) in the text (txt).

Sentence Frequency (SF) is a measure of counting the shared sentences between the two texts.

SF= number of shared sentences (S) between text1 (txt1) and text2 (txt2).

C. Statistical Methods

The below coefficients used statistical methods to count the similarity between texts:

1) Jaccard Coefficient: This coefficient counts the similarity degree between the two texts by dividing the number of similar tokens which appear in them on all tokens of the two texts; this can be done by using equation 2:

$$JC(A,B) = \frac{\sum_{j=1}^{n}(Aj,Bj)}{\sum_{j=1}^{n}(Aj,Bj) + \sum_{j=1}^{n}|Aj - Bj|} \quad \ldots\ldots\ldots\ldots\ldots (2)$$

Where $JC(A,B)$ illustrates the similarity degree between text A and text B.

2) Similarity Coefficient: The value of this coefficient can be counted by dividing the number of shared sentences between the two texts on the whole number of sentences in both texts without the shared sentences between them; this can be applied by using equation 3:

$$SC = \frac{X}{Y - X} \qquad \ldots\ldots\ldots\ldots (3)$$

Where SC represents the Similarity Coefficient value, X represents the number of shared sentences between the two texts and Y represents the whole number of sentences in both texts.

3) An equation of Proposed System: The proposed equation combining the Jaccard coefficient and similarity coefficient as shown in equations 1 and 2 respectively with two criteria (α, β) to calculate the similarity degree between the two texts. The weight of each criterion (α, β) is multiplied with the Jaccard coefficient and the similarity coefficient respectively. The aim of these criteria is to retain the value of the similarity degree does not override the value between (0 and 1), when the value is equal to 1 means that the two texts are similar but when there is no similarity between them the value becomes equal to 0. So the two criteria have an effect on the proposed equation. The proposed system equation is illustrated in equation 4

$$Similarity(A,B) = \underbrace{\alpha}_{0.8 \le \alpha < 1} \bullet \overbrace{JC}^{\text{Jaccard Coefficient}} + \underbrace{\beta}_{\beta = \alpha - 1} \bullet \overbrace{SC}^{\text{Similarity Coefficient}} \qquad \ldots\ldots\ldots\ldots (4)$$

Where ***similarity (A, B)*** illustrates the similarity degree between text A and text B. ***JC*** and ***SC*** represent the Jaccard coefficient and similarity coefficient respectively. α and β represent the two criteria.

1- Results and Experiments

Texts that have been collected are saved in a dedicated place for using them later by the proposed system for the test. Test data is a set of texts that distributed in the 3 main classes (computer sciences, applied mathematics and applies physics) include (4, 3 and 3) sub-classes respectively. Table 1 shows the set of test texts.

Table1: The Set of Test Texts.

Main Class	Sub-Class	No. of Text Documents
Computer Science	Artificial Intelligence (AI)	21
	Network (NW)	26
	Data Security (DS)	26
	Information Technology (IT)	24
Applied Mathematics	Numeric Analysis (NA)	28
	Number Theory (NT)	24
	Cryptography (C)	24
Applied Physics	Physics Laser (PL)	28
	Material (M)	26
	Optics (O)	20

Table 2: Experimental Results

Text1		Text2	Similarity Degree for Jaccard Coefficient	Similarity Degree for Proposed Equation
AI	3	AI.3	1	1
	5	AI.3	0.71	0.80
	7	NW.9	0.23	0.17
	11	IT.8	0.12	0.08
	13	DS.13	0.11	0.07
	16	NT.12	0	0
	21	M.7	0	0
NW	3	NW.3	1	1
	6	NW.3	0.88	0.93
	11	IT.12	0.45	0.33
	15	DS.16	0.1	0.08
	21	NA.14	0	0
	26	PL.10	0	0
DS	4	DS.4	1	1
	8	DS.4	0.79	0.91
	13	IT.23	0.13	0.06
	19	C.16	0	0
	26	O.18	0	0
IT	1	IT.1	1	1
	7	IT.1	0.82	0.94
	14	NA.20	0	0
	24	M.19	0	0

NA	2	NA.2	1	1
	9	NA.2	0.81	0.89
	5	NT.8	0.71	0.59
	21	PL.14	0	0
	28	M.20	0	0
NT	3	NT.3	1	1
	6	NT.3	0.82	0.89
	15	C.10	0.73	0.68
	24	O.15	0	0
C	1	C.1	1	1
	5	C.1	0.83	0.87
	13	PL.11	0	0
	22	O.19	0	0
PL	2	PL.2	1	1
	7	PL.2	0.81	0.89
	17	M.11	0	0
M	3	M.3	1	1
	8	M.3	0.83	0.87
	22	O.18	0	0
O	1	O.1	1	1
	1	O.4	0.79	0.87
	3	IT.2	0	0
	5	C.5	0	0

Table 2 displays the obtained results from the system after testing a set of selected texts from multi-classes as shown in Table 1. Results represent the similarity degrees which calculated using Jaccard coefficient initially then by using the proposed equation on the same texts; the differences in texts similarity degrees were observed between the proposed equation and the Jaccard coefficient. whenever the texts from the same class or nearby to that class, the similarity degrees increased in the proposed equation on the Jaccard coefficient; but whenever the texts from different classes, the similarity degrees decreased in the proposed equation on the Jaccard coefficient. In other words, the similarity degrees in the proposed equation have improved relatively in both cases, so the results from the proposed equation are logical.

Conclusions

The proposed equation is given better results than using the Jaccard co-

efficient alone, where improved the nearby texts similarity degrees and diverged the similarity degrees of various texts. This improvement was the result of the influence of the degree of the similarity coefficient on the system equation, where its degree depends on the number of shared sentences between texts.

Two criteria that have been adopted are: α its value between (0.80 and 1) and β= α-1. They are noted in the proposed equation of the current system which showed Jaccard coefficient multiplied by α, to give greater importance to shared words were not to ensure to existence shared sentences between the two texts. After the experience of many values for these criteria, it shows that these values give better results than the other values.

Typically, the stop words considered 20-30% of the total number of words. Removing the stop words is useful to reduce the statistical operations, especially if the similarity between long texts is required to calculate.

Stemming process has an important effect on the size of texts; it decreases the text size when the different forms of the token are put under the root of that token. Therefore, the stemming process improves system performance.

In the proposed equation, the similarity degree increased by 0.0285 (about by 3%) on the degree of the Jaccard coefficient when selected texts from the same sub-class, while it became less than the Jaccard coefficient degree when selected texts from different sub-classes. As a result, the similarity results became more logical.

REFERENCES

Khuat, T. T., Nguyen, D. H. and Le, T. M. H. (2015). A Comparison of Algorithms used to measure the Similarity between two documents. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Vol. 4, No. 4, pp. 1117-1121, Australia.

Lisna, Z. (2016). 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, pp. 11-18, Indonesia.

Neha, A., Mukesh, R. and Vijay, M. (2014). Comparative Analysis of Jaccard Coefficient and Cosine Similarity for Web Document Similarity Measure. International Journal for Advance Research in Engineering and Technology, Volume 2, Issue X, India.

Nitesh, P., Manasi, G. and Rajesh, W. (2015). A Review on Text Similarity Technique used in IR and its Application. International Journal of Computer Applications, vol. 120, No. 9, pp. 29-34, India.

Praveenkumar, V. and Harinarayana, N.S. (2018). Social Semantics and Similarities from User-generated Keywords to Information Retrieval: A Case Study of Social Tags in Marine Science. Journal of Library & Information Technology, Vol. 38, No. 1, pp. 11-15,India.

Sheetal, A. T. and Sushma, S. N. (2010). Measuring Semantic Similarity between Words Using Web Documents. International Journal of Advanced Computer Science and Applications, (IJACSA), Vol. 1, No.4,pp. 78-85, India.

Su, G. C. and Seoung, B. K. (2017). A Data-Driven Text Similarity Measure Based on Classification Algorithms. Journal of Industrial Engineering International, vol. 24, No. 3, pp. 328-339, Korea.

Suphakit, N., Jatsada, S., Ekkachai, N. and Supachanun, W. (2013). Using of Jaccard Coefficient for Keyword Similarity. Proceedings of the International MultiConference of Engineers and Computer Scientists, Vol. I, Hong Kong.

Vijaymeena, M.K. and Kavitha, K. (2016). A Survey on Similarity Measures in Text Mining. Machine Learning and Applications: An International Journal (MLAIJ), Vol. 3, No. 1, pp. 19-28,India.

Vikas, T. and Vivek, J. (2013). Comparison of Jaccard, Dice, Cosine Similarity Coefficient To Find Best Fitness Value for Web Retrieved Documents Using Genetic Algorithm. International Journal of Innovations in Engineering and Technology (IJIET), Vol. 2 Issue 4, pp. 202-205,India.

Wael, H. G. and Aly, A. F. (2012). Short Answer Grading Using String Similarity and Corpus-Based Similarity. International Journal of Ad-

vanced Computer Science and Applications (IJACSA), Vol. 3, No. 11, pp. 115-121,Eygpt.

UNIVERSIDAD DEL ZULIA

opción
Revista de Ciencias Humanas y Sociales

Año 35, Especial N° 19, 2019

Esta revista fue editada en formato digital por el personal de la Oficina de Publicaciones Científicas de la Facultad Experimental de Ciencias, Universidad del Zulia.
Maracaibo - Venezuela

www.luz.edu.ve

www.serbi.luz.edu.ve

produccioncientifica.luz.edu.ve