Comparison of document similarity measurements in scientific writing using Jaro-Winkler Distance method and Paragraph Vector method

To cite this article: S C Cahyono 2019 IOP Conf. Ser.: Mater. Sci. Eng. 662 052016

View the article online for updates and enhancements.
Comparison of document similarity measurements in scientific writing using Jaro-Winkler Distance method and Paragraph Vector method

S C Cahyono

Magister of Information Systems, Faculty of Postgraduate Studies, Universitas Komputer Indonesia, Bandung, Jawa Barat, Indonesia.

Email : stefanus@email.unikom.ac.id

Abstract. The purpose of this research is to study the methods of measuring the similarity of documents and tell us which is the most suitable for Indonesian Scientific Writing. This research method used was Jaro-Winkler Distance as method. Jaro-Winkler is a method that calculates the distance between strings and then measures the similarity. Doc2Vec (Paragraph Vector) is a method that aims to represent documents in vector form for comparison with the machine learning process. The results of this study compare the results of plagiarism detection between the Jaro-Winkler Distance method and the Doc2Vec method. The best measurement comparison method used is the accuracy of the comparison of documents and their speed. Using the dataset created, Doc2Vec outperformed the Jaro-Winkler Distance algorithm in comparing document similarities. Therefore, the development of a document similarity method will be easier in the future by using Doc2Vec (Paragraph Vector) in Indonesian scientific works.

1. Introduction

Nowadays, getting information and doing research is easier than ever. The big thing that makes this happen is information technology with the world's best network tool, "the Internet". For example, Google, now known as the largest search engine in the world, leads its ranking in response time and precision [1]. This country is equipped with several pluses and minuses, especially for the academic world. In the academic world, research will continue to grow over time. This clearly influences the increasing number of Scientific Writing documents. As for the advantages and disadvantages of them, the advantage is that researchers can easily search for information and have less difficulty. The minus is that many types of research and scientific writing will have less quality and have more opportunities for plagiarism. Centre study in 2011, around 55% of students claimed to be involved in plagiarism of scientific work, to avoid this, the application of various methods in the document comparison process has been applied to determine the similarity of a document with other documents. By using two popular methods, we can compare which method is best applied in measuring the similarity of documents from Scientific Writing [2].

One of the most used methods to compare some documents and detect plagiarism in Indonesian scientific writing is Jaro-Winkler Distance method. Jaro-Winkler distance is a measurement to measure the similarity between two strings [3]. W W Cohen et al, tells that the Winkler variant substantially improves all the Jaro variants [4]. T Tinaliah et al. found that Jaro-Winkler Distance algorithm gives better comparison rather than Latent Semantic Analysis algorithm [5]. S Rinusantororo and P Novantara et al, also prove that the implementation of Jaro-Winkler Distance can be used to detect plagiarism on scientific writings [6-7].
Even though the Jaro-Winkler Distance is one of the best methods, Jaro-Winkler has been introduced since 1990 by W E Winkler [8] and it was quite a long time ago. According to a thesis by J E Alvarez, there is another state-of-the-art document similarity algorithm that we can use [9]. One of the best is an algorithm proposed by Google Inc researcher Q Le and T Mikolov called Doc2Vec. Doc2Vec or Paragraph Vector itself, called as an unsupervised learning algorithm which learns vector representations for variable length pieces of texts such as sentences and documents [10]. Not only for detecting plagiarism, Doc2Vec had also been used to do sentiment analysis classification [11]. While Jaro-Winkler Distance had been applied in many cases for scientific writings written in Indonesian Bahasa, Doc2Vec which had acquired state-of-the-art methods for document similarity algorithm and can do pretty many things is unpopular yet.

The purpose of this research is to do some testing on computational time and precisions of the Jaro-Winkler Distance method and Doc2Vec method to find document similarities then compare it.

2. Method

2.1. Jaro-Winkler Distance

Jaro-Winkler Distance is a variant of Jaro distance metric that measure an edit distance between two sequences or strings. Jaro-Winkler distance is widely used in the areas of information extraction, record linkage, entity linking since it performs well in matching personal and entity names [3]. The higher score of Jaro-Winkler distance between two strings, the more likely that those strings were similar. The score threshold is 0 for unsimilar and 1 for similar.

The string comparator first introduced by Jaro [12]. This contributes to insertion, deletion and transposition. The main components of Jaro algorithm is:

- String lengths computing.
- Common characters counting in the two strings.
- Transpositions counting.

According to W E Winkler (2006), the definition of common is that the agreeing character must be within half the length of the shorter string [13]. The definition of transposition is that the character from one string is out of order with the corresponding common character from the other string. The string comparator value (rescaled for consistency with the practice in computer science) is:

$$\phi_{J}(s_1, s_2) = \frac{1}{3} \left( \frac{N_C}{\text{len}_{s_1}} + \frac{N_C}{\text{len}_{s_2}} + \frac{0.5N_T}{N_C} \right)$$

(1)

$s_1$ and $s_2$ are the strings with lengths $\text{len}_{s_1}$ and $\text{len}_{s_2}$, $N_C$ is the common characters counts between those two strings where the distance for common is half of the minimum length of $s_1$ and $s_2$, and $N_T$ is the transpositions count. The number of transpositions $N_T$ is computed somewhat differently from the obvious manner.

Using truth data sets, Winkler (1990a) introduced methods for modelling how the different values of the string comparator affect the likelihood (1) in the Fellegi-Sunter decision rule. Winkler also showed how a variant of the Jaro string comparator $\phi$ dramatically improves matching efficacy in comparison to situations when string comparators are not used (Table 1).
Table 1. Comparison of String Comparators Using Last Names and First Name [13].

| Two strings     | String Comparator Values |        |        |        |
|-----------------|--------------------------|--------|--------|--------|
| SHACKLEFORD     | SHACKELFORD              | 0.970  | 0.982  | 0.925  | 0.818  |
| DUNNINGHAM      | CUNNINGHAM               | 0.896  | 0.896  | 0.917  | 0.889  |
| NICHLESON       | NICHULSON                | 0.926  | 0.956  | 0.906  | 0.889  |
| JONES           | JOHNSON                  | 0.790  | 0.832  | 0.000  | 0.667  |
| MASSEY          | MASSIE                   | 0.889  | 0.933  | 0.845  | 0.667  |
| ABROMS          | ABRAMS                   | 0.889  | 0.922  | 0.906  | 0.833  |
| HARDIN          | MARTINEZ                 | 0.000  | 0.000  | 0.000  | 0.143  |
| ITMAN           | SMITH                    | 0.000  | 0.000  | 0.000  | 0.000  |
| JERALDINE       | GERALDINE                | 0.926  | 0.926  | 0.972  | 0.889  |
| MARHTA          | MARTHA                   | 0.944  | 0.961  | 0.845  | 0.667  |
| MICHELLE        | MICHAEL                  | 0.869  | 0.921  | 0.845  | 0.625  |
| JULIES          | JULIUS                   | 0.889  | 0.933  | 0.906  | 0.833  |
| TANYA           | TONYA                    | 0.867  | 0.880  | 0.883  | 0.800  |
| DWAYNE          | DUANE                    | 0.822  | 0.840  | 0.000  | 0.500  |
| SEAN            | SUSAN                    | 0.783  | 0.805  | 0.800  | 0.400  |
| JON             | JOHN                     | 0.917  | 0.933  | 0.847  | 0.750  |
| JON             | JAN                      | 0.000  | 0.000  | 0.000  | 0.667  |

2.2. Doc2Vec (Paragraph Vector)

Paragraph Vector or also known as Doc2Vec was introduced by Q Le and T Mikolov (2014) which called as unsupervised framework that learns continuous distributed vector representations for piece of texts [10]. Doc2Vec was first proposed to cover the weaknesses of bag-of-words algorithm which are lose the ordering of the words and also ignore words’ semantics. For Example, “powerful”, “strong” and “Paris” are equally distant despite the fact that semantically, “powerful” should be closer to “strong” than “Paris”. Doc2Vec also an advancement of distributed vector representation of words algorithm called as Word2Vec by Mikolov et al [10].

2.2.1. Vector Representation of Words.

The word2vec, capture ordering and semantic of the words by giving a numeric representation for each word. This framework is part of a wider concept in machine learning called the feature vectors. To represent and encapsulate different relation between words, like antonyms, synonyms, or analogies, the method will build some model using words, for example, in Figure 1 you can see that king to queen is like man to woman.
Word2Vec representation combined 2 algorithms which is *Continuous Bag-of-Words model (CBOW)* and the *Skip-Gram* model. To predict the “context” of a word from the surrounding words, CBOW model creates a sliding window around current word. Each word is represented as a feature vector. After training, these vectors become the word vectors. In Figure 2, we can see that the context of three words (“the”, “cat” and “sat”) is used to predict the fourth word (“on”). The input words are mapped to columns of the matrix W to predict the output word.

The second algorithm *Skip-Gram* is actually the opposite of *CBOW*. It uses a word to predict all surrounding words (“context”) instead of predicting one word. Skip gram is considered more accurate with infrequent words than *CBOW* but much slower. Given a sequence of training words, word vector model maximizes its average log probability.
\[
\frac{1}{T} \sum_{t=k}^{T-k} \log p(w_t | w_{t-k}, \ldots, w_{t+k})
\]  

Then the prediction thing is done via a multiclass classifier, such as softmax like so:

\[
p(w_t | w_{t-k}, \ldots, w_{t+k}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}}
\]

Each of \(y_i\) is un-normalized log-probability for each output word \(i\), computed as:

\[
y = b + Uh(w_t | w_{t-k}, \ldots, w_{t+k}; W)
\]

Where \(U, b\) are the softmax parameters and \(h\) is a concatenation or average of word vectors extracted from \(W\).

2.2.2. Paragraph Vector

The main goal of doc2vec is to create a numeric representation of a document, regardless of its length. But unlike words, documents do not come in logical structures such as words, so Mikolov and Le used a simple, yet clever concept that add a vector to store the Paragraph ID in Word2Vec scheme. In Figure 3, we can see the framework which is similar to the framework presented in Figure 2; the only different is the additional paragraph token that is mapped to a vector via matrix \(D\). In this model, the concatenation or average of this vector with a context of three words is used to predict the fourth word. The paragraph vector represents the missing information from the current context and can act as a memory of the topic of the paragraph.

Figure 3. A framework for learning paragraph vectors. [10]

In summary, Mikolov et al. says that the algorithm itself has two key stages: 1) training to get word vectors \(W\), softmax weights \(U, b\) and paragraph vectors \(D\) on already seen paragraphs; and 2) “the inference stage” to get paragraph vectors \(D\) for new paragraphs (never seen before) by adding more columns in \(D\) and gradient descending on \(D\) while holding \(W, U, b\) fixed. They make a prediction about some particular labels using \(D\).

This model called Distributed Memory version of Paragraph Vector (PV-DM). It remembers what is missing from the current context or as the topic of the paragraph. This simple concept makes big different from word vector because the document vector now intends to represent the concept of a document.

Another algorithm called Distributed Bag of Words version of Paragraph Vector (PV-DBOW) that similar to Skip-Gram may be used. This algorithm forces the model to predict words randomly sampled
from the paragraph in the output by ignores the context words in the input. In reality, at each iteration of stochastic gradient descent, it samples a text window, then sample a random word from the text window and form a classification task given the Paragraph Vector. In Figure 4 show the PV-DBOW representation. In this version, the paragraph vector is trained to predict the words in a small window.

![Diagram of Paragraph Vector](image)

**Figure 4.** Distributed Bag of Words version of paragraph vectors. [10]

This algorithm is actually faster and store less data, because there is no need to save the word vectors. It only needs to store the softmax weights instead of both softmax weights and word vectors in the previous model. Mikolov and Lee also state that it is recommended using a combination of both algorithms, though the PV-DM model is superior and usually will achieve state of the art results by itself. The combination is usually more consistent across many tasks.

3. Results and Discussions

To test and compared the computational time and precisions between Jaro-Winkler Distance method and Doc2Vec method, we need to do some steps. First, we collect datasets which is some academic text and scientific writings with various characteristics. Then we compare the documents using each method or algorithm. While we compare the documents, we also have to count the computational time of each method. After the document’s comparisons have done, we then have to compare the precision between these 2 algorithms.

3.1. Datasets

The dataset is collected from the electronic library of Universitas Komputer Indonesia (ELIB UNIKOM) that contain thesis files of Information System Magister’s students. All documents are written in Indonesian’s Bahasa and the total documents collected were 120 documents. To make the document similarity measurements more comparative, we should tweak some data to make the datasets have some more characteristics. We then add some similar text on certain paragraph in 100 documents so they have exact similarity to measure. Now, we have datasets ready to be compared with composition as seen on the Table 2.

| Num. | Tweaks                        | Quantity |
|------|-------------------------------|----------|
| 1.   | Original Documents             | 60       |
| 2.   | Add 20% of similar text       | 20       |

**Table 2.** Datasets to be compared using 2 document similarity algorithms.
3.2. Build Model & Corpus

As we know before we get to compare the document, we have to make a model and or corpus using document pre-processing and train the data. According to M Allahyari et al., the pre-processing step usually consists of the tasks such as tokenization, filtering, lemmatization and stemming [14]. We use tokenization and stemming to build the datasets into compared ready model. Tokenization is the task of breaking a character sequence up into pieces called tokens and at the same time throw away punctuation marks while stemming is a method that aim at obtaining stem of derived words. After the model is build and trained, we can begin the similarity measurements.

3.3. Similarity Measurements

First, we will start the similarity measurements using Jaro-Winkler Distance algorithm then Doc2Vec algorithm. Measurements begin sequentially one by one documents starting with the Original Documents to 100% similar text contained Documents (see Table 3). Computation will count by execution times with milliseconds as unit.

3.4. Algorithm Comparison

After the similarity measurements, we then know how big the computational process of each algorithm is. The good one algorithm determined by how low the execution time is. In other way, precision will be determined by percent unit. For Jaro-Winkler Distance, we can use the threshold and make it percent values. And for Doc2Vec or Paragraph Vector, we can also convert the accuracy to percent values.

In this section, we have performed a comparison of state-of-the-art document similarity measurements. According to our main purpose, we need to know which algorithm and method best suit for measuring document similarities of scientific writings in Indonesian Language [15]. We have looked on two methods, the first one was Jaro-Winkler Distance. Jaro-Winkler distance is still more popular than Doc2Vec in terms of document similarity algorithms. By giving a query string, Jaro-Winkler distance similarity search is able to finds all strings in a data set as long as it is no more than a given threshold. The Jaro–Winkler distance gives more favourable ratings to strings that match from the beginning for a set prefix length using a prefix scale.

The more similar the strings are, the lower Jaro–Winkler distance for two strings. The score is normalized, 1 equates to no similarity and 0 is an exact match. The Jaro–Winkler similarity is given by 1 − Jaro–Winkler distance [12].

Another one is Doc2Vec. Doc2Vec is almost a direct translation of Word2Vec into the context of documents. The only difference between Word2Vec and Doc2Vec is that Doc2Vec introduces an extra global vector to the context that represents the document the window is in. This way, Doc2Vec forces these global vectors to learn the information missing from the local context and that is necessary to perform predictions in the global document context. Doc2Vec is well known in the community, but mostly because it follows Word2Vecs fame. In truth, its results are somewhat underwhelming, both in third party benchmarks and in J E Alvarez tests. Doc2Vec barely outperforms the naïve benchmark in the best of cases, yet, it is the best performing algorithm we have analyse, which says a lot about the state of the field. It was also, by far, the fastest implementation; although it is not clear if it is the fastest algorithm, as it does not have a clear advantage on theoretical complexity [9].

We have performed analysing on these two different algorithms to measures document similarities. Overall, the algorithms do not perform too differently to each other. But according to the measurement results, we can see in the Table 3 that Doc2Vec is a little bit outperform the Jaro-Winkler Distance according to its computational and precisions by adding the similar text on each document.
Table 3. The comparison of document similarities measurements.

| Num. | Documents                  | Computational        | Precisions |
|------|----------------------------|----------------------|------------|
|      |                            | Jaro-Winkler | Doc2Vec | Jaro-Winkler | Doc2Vec |
| 1.   | Original Documents         | 2320 ms      | 2234 ms | 0.74         | 0.88    |
| 2.   | Add 20% similar text       | 2250 ms      | 1802 ms | 0.85         | 0.90    |
| 3.   | Add 60% similar text       | 2241 ms      | 1752 ms | 0.92         | 0.89    |
| 4.   | Add 100% similar text      | 2130 ms      | 1600 ms | 0.90         | 0.97    |

4. Conclusions
This research has been performed as a review of document similarities measurements algorithms for scientific writings of Indonesian Bahasa. The main purpose of this work is to determine whether a state-of-the-art document similarity algorithm, Doc2Vec is suitable for this area or not. Fortunately, yes, it is. We found that Doc2Vec can outperform Jaro-Winkler Distance as a popular algorithm for this area.

References
[1] Setiawan, E. B., & Setiyadi, A. 2018. Web vulnerability analysis and implementation. In IOP Conference Series: Materials Science and Engineering. 407(1), pp. 012081.
[2] Pupovac V and Fanelli D 2015 Scientists Admitting to Plagiarism: A Meta-analysis of Surveys Sci. Eng. Ethics 21(5), pp 1331–52
[3] Y Wang, J Qin and W Wang 2017 Efficient Approximate Entity Matching Using Jaro-Winkler Distance WISE 2017, Part I, LNCS 10569, 23(4)pp. 231–239
[4] W W Cohen, P Ravikumar and S E Feinberg 2003 A Comparison of String Distance Metrics for Name-Matching Tasks Kdd workshop on data cleaning and object consolidation. 3, pp. 73–78).
[5] T Tinaliah and T Elizabeth.2018.Perbandingan Hasil Deteksi Plagiarisme Dokumen dengan MetodeJaro-Winkler Distance dan Metode Latent Semantic Analysis Jurnal Teknologi dan Sistem Komputer. 6(1), pp 2 – 8.
[6] S Rinusantoro 2014 Aplikasi Deteksi Kemiripan Dokumen Teks Thesis Pasca Sarjana Universitas Gajah Mada Yogyakarta. 54(2),pp 221–306
[7] P Novantara and O Pasruli.2017.Implmentasi Algoritma Jaro-Winkler Distance Untuk Sistem Pendeteksi Plagiarisme Pada Dokumen Skripsi Jurnal Buffer Informatika, 3(2) ISSN 2527-4856
[8] W E Winkler 1990 String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage Educational Resources Information Center (ERIC) ED325505, 11(2), pp 54–87
[9] J E Alvarez and H Bast 2017 A Review of word embedding and document similarity algorithms applied to academic text Bachelor’s Thesis University of Freiburg
[10] Q Le and T Mikolov 2014 Distributed Representations of Sentences and Documents Proceedings of the31st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP 3(27), (pp 116-123)
[11] W C Widyaningtyas, Adiwijaya and S A Faraby 2018 Sentiment Analysis Classification of Movie Review in English Language using Doc2Vec and Support Vector Machine (SVM) e-Proceeding of Engineering : 5(1), pp 1570
[12] M A Jar 1989 Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida Journal of the American Statistical Association 84 pp 414-420.
[13] W E Winkler 2006 Overview of Record Linkage and Current Research Directions Technical Report Statistical Research Report Series RRS2006/02 Bureau of the Census. 06(2) pp 34-33
[14] Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. 2017. A brief survey of text mining: Classification, clustering and extraction techniques. arXiv preprint arXiv:1707.02919.

[15] Soegoto, E. S., & Jayaswara, M. R. 2018. Web and Android Programming Course Information System. In IOP Conference Series: Materials Science and Engineering . 407(1), p. 012063.