DETECTING CONTEXT SIMILARITY OVER MULTIPLE DOCUMENTS USING LINGUISTIC FEATURES

ARIE NOVANDI, HANDRI SANTOSO

Ph. D, Surya University
E-mail: [1]arie.novandi14@student.surya.ac.id, bondry@gmail.com

Abstract—Presenting a method of comparing multiple documents and determining if both documents have the same context. Context are defined by the setting of an event, statement, and ideas in which each document can be understood and assessed. The language handled by this study, will be in the Indonesian language. 3 different approaches will be used to determine context similarity. 2 of the approaches would be adopted from TF*IDF method, while the last one would extract information via evaluating Indonesian language's forms. By using these methods, keywords would be automatically generated in the algorithm, requiring minimal human participation to get the desired result. Evaluation of the algorithm shows discerning results between matching and nonmatching documents.

Index Terms- Context Similarity, Linguistic, Similarity of Documents

I. INTRODUCTION

In the digital age, everyone having access to creating and publishing their own information would create a staggering amount of content. An unrestrained media might allow unreliable and/or irresponsible information to reach the eyes of the masses. Hoax could easily be shared across multiple social media. Consider a particular hoax that has been circulated all over the social media and blogs. In order to recognize and remove these threats, every document have to be read and manually identified if it contains the same information as the originally discovered hoax. This task would be time consuming and inefficient considering the massive amount of available information sharing media in existence.
Using the method presented in this paper, one practical application would be using a document that has been identified as the hoax and compare it with all the suspected documents.

II. DEFINITION FOR SIMILARITY

Similarity by definition is the state of more than 1 objects resembling each other without being identical. In linguistics, [1] Divides semantic typing in terms of 2 mechanism: the detection of similarities and differences. Our task is to detect this similarity automatically between multiple documents.
This paper defines 2 documents similar if: (a) The participants and objects of both documents are the same and (b) the actions taken to these objects are the same. Defining point (a) would be done by extracting all names of both people and objects. Point (b) would be defined by extracting all the word essences in the Indonesian language while taking into consideration their word forms such as verb, nouns, and adjectives.

III. SIMILAR WORKS

Reference [3] and [4] has given us extensive literature on measuring similarities between long texts using vector-based and corpus-based methods. While [5] discusses the similarity between short text passages using word co-occurrence and corpus-based methods. One of the most common information retrieval method using a word query would describe documents as a work vector, and then the query is matched into similar document via similarity metric [6].

Most existing literature describing text similarities works in the English language. The algorithm created in this paper would be adjusted for the Indonesian language using hybrid methods. The said methods are Term Frequency * Inverse Document Frequency and word co-occurrence method. This paper would not need any word queries and instead would generate the keywords automatically when comparing documents. To help achieve automatically generated keywords, the algorithm would utilize the document vector method described in [6] while making it a working model for the term frequency * inverse document frequency and word co-occurrence method.

IV. PROPOSED METHOD

A. Building Indonesian Language Database

The first step is to build a comprehensive Indonesian language database. This database is required to log every word in existence, including its word forms. Nouns, verbs, adjectives will be categorized as the core words. Less meaningful words such as prepositions and conjunctions must be logged into their own separate categories (will be categorized as *filler* words). The purpose of this dictionary is to filter more important words such as the core words and names of people or objects. These filtered words will serve the purpose of being keywords for the next steps in the methodology. This database will also be used as a database to automatically generate keywords by the rest of the similarity detection method.

*B. Term Frequency * Inverse Document Frequency*

Term Frequency * Inverse Document Frequency

(TF*IDF)is a method devised as a text indexing system [2]. This method is superior to other indexing system because of its term weighting system. Term Frequency measures how frequently a term occurs in a document. Since each document is different in length, some terms would appear more in a longer document than shorter ones. Each term's frequency is calculated by dividing the total number of terms in the document as a way of normalization as follow:

$$TF(t) = \frac{Number\ of\ times\ term\ t\ appears}{Total\ number\ of\ terms}$$

Term frequency will be used twice in our methodology. Its purpose is to (a) Determine the frequency and weight of unique words that doesn't exist in the language database and (b) Determine the frequency and weight of core words that exists in the database.

By finding non-existent words in point (a), we figure out the term frequency of unique words such as names of people, organization, places, etc. Hence the importance of logging every word including filler words in the database. Filler words are logged to avoid weighing in words of lesser importance in the algorithm.
The purpose of point (b) is to measure the term frequency of core words, excluding the words that have already been measured at point (a).

C. Word Co-occurrence

The third method used is to separate all the sentences in the document. Each of these sentence's essence would then be retrieved by extracting core words that has been established in the dictionary (verbs, nouns, and adjectives). Filler words would not be extracted as they would not contribute as keywords and would instead blur the result in the next step. Each word extracted would be treated as keywords in the sentence they belong in.

D. Comparing Between Documents

Both term frequency methods and word co-occurrence method are run within all the documents that are about to be compared. Each method's result is compared to the corresponding method in the other document. The shorter document is used as the source document, and the longer document acts as the comparator. This is done because shorter document would contain more condensed information than the longer one.
From both term frequencies of unique words and core words, we would get a list of keywords with their respective weight in the document. The similarity of both document using keyword term frequencies will be calculated as:

$$S(a) = \frac{\sum(TF(matched\ unique\ word))}{Sum\ of\ keywords}$$

$$S(b) = \frac{\sum(TF(matched\ core\ word))}{Sum\ of\ keywords}$$

Where *TF(matched unique word)*and *TF(matched core word)* is the term frequency of words in source document that also exists in the comparator document and sum of words is the total number of keywords that exists in the source document. This would get us a percentage of matching words in both document where the weight of each word is put into consideration.
From the word co-occurrence method, we would get a list of sentences with their own respective keywords. A sentence from source document would be considered similar if there exists another sentence in the comparator document that also has the exact same keywords. For example:

source: Jack sedang **mengendaraimobil**
(translated: Jack **is driving** a **car**)

comparator: **Mobil** itu sedang **dikendarai** oleh Jack
(translated: The **car** is being **driven** by Jack)

Those sentences would be considered similar since the comparator sentences contains all the keyword that also exists in the source sentence, which in this case is mobil (translated: **car**) and kendara (translated: **drive**). Prefixes and suffixes in the language will be omitted. The similarity of both documents using the word co-occurrence method will be calculated by counting the percentage of matched sentences from the source document:

$$S(c) = \frac{Amount\ of\ matched\ sentences}{Total\ amount\ of\ sentences\ in\ source}$$

I. RESULTS

A. Experimentation Details

The algorithm was tested between 120 different combination of documents. All documents were extracted from various online Indonesian news sources. Context relations between documents are manually marked to compare actual context similarity and the similarity defined as a result from the algorithm. Manual context similarity detection is done to ensure the validity of the algorithm's result.

B. Experimental Results

From all 3 used similarity detection method, a final result is generated including the confidence level presented as a percentage. There is immense difference between the results of nonmatching documents and matching documents. Using the first method of unique words term frequency, matching documents has an average similarity result of 41,85% while nonmatching documents has an average of 5,07%. Core words term frequency between matching documents has an average similarity result of 54,75% while nonmatching document has 13,65%. The last method of word co-occurrence has 21,80% average between matching documents, and 4,65% on nonmatching documents. To eliminate data variances

and make the result more robust, a 10% trimmed mean is applied to the result [7]

Table I. Result between matching documents

	Min	Average	Max
Unique TF *S(a)*	12%	41,85%	88%
Core TF *S(b)*	33%	54,75%	98%
Word Co-occurrence *S(c)*	3%	21,80%	72%

Table II. Result between nonmatching documents

	Min	Average	Max
Unique TF *S(a)*	0%	5,07%	16%
Core TF *S(b)*	2%	14,01%	29%
Word Co-occurrence *S(c)*	2%	4,65%	9%

To merge all 3 methods into one result, first the weight of each method is calculated. Meaning how accurate would each method be when used on its own. This is done by comparing the result of each matched document to the average result of each method. As an example, the term frequency of unique words *S(a)* method would need the document pair to have at least 41,85% similarity to be considered correct. Running these to all 3 methods generates the weight of 37,2% of method *S(a)*, 46,51% from method *S(b)*, and 16,27% from method *S(c)*. Each method would give their own points, adding them into the final confidence result.

All previous results are done to generate the final confidence result*S(r)*. *S(r)* is the combined result of all 3 methods considering their respective weights. Each method considers a document pair similar when it reaches the method's threshold of their maximum result on nonmatching documents seen on Table II. Therefore, if a method considers a document pair similar, that method gives full point that is the method's weight (points given cannot exceed weight value). *S(r)* would be the confidence percentage of the sum of all 3 methods.

$$S(r) = \min\left\{\frac{S(a) * 0.372}{0.16}, 0.372\right\} + \min\left\{\frac{S(b) * 0.465}{0.29}, 0.465\right\} + \min\left\{\frac{S(c) * 0.162}{0.09}, 0.162\right\}$$

Table III. Confidence Result *S(r)*

	Min	Average	Max
Matching Documents	89.1%	98.41%	99.9%
Nonmatching Documents	19.87%	43.21%	82.26%

Considering the result *S(r)*, there are clearly discernable differences between matching and nonmatching documents where the trimmed average of matching documents is 98.41% and nonmatching documents is 43.21%. It is proposed that the algorithm would consider 85% as the similarity threshold.

REFERENCES

[1] W. Frawley, "Linguistic Semantics,"Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1992.

[2] G. Salton, and C. Buckley, "Term-Weighting Approaches in Automatic Text Retrieval. Information Processing & Management," Information Processing and Management, 25(5):513-523, January 1988.

[3] T. K. Landauer, S. T. Dumais, "A Solution to Plato's Problem: The Latent Semantic Analysis Theory of the Acquisition, Induction, and Representation of Knowledge," Psychological Review, November 1996.

[4] C. T. Meadow, B. R. Boyce, and D. H. Kraft, "Text Information Retrieval Systems," Information Retrieval Systems, 3rd edition, March 2007.

[5] V. Hatzivassiloglou, J. L. Klavans, and E. Eskin, "Detecting Text Similarity over Short Passages: Exploring Linguistic Feature Combination via Machine Learning," Joint SIGDAT Conference, 1999.

[6] G. Salton, and M. Lest, "Computer Evaluation of Indexing and Text Processing," Prentice Hall, Ing. Englewood Cliffs, New Jersey, PP. 143-180, 1971.

[7] M. B. Brown, and A. B. Forsythe, "Robust Tests for the Equality of Variances," Journal of the American Statistical Association," vol. 69, 1974.

★★★