Journal of Theoretical and Applied Information Technology
10th May 2014. Vol. 63 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

PLAGIARISM DETECTION ALGORITHM USING NATURAL LANGUAGE PROCESSING BASED ON GRAMMAR ANALYZING

[1]ANGRY RONALD ADAM, [2]SUHARJITO

[1]Lecturer, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia

[2] Assoc Prof., Magister of Information Technology Program, Bina Nusantara University, Jakarta, Indonesia

E-mail: [1]aadam@binus.edu, [2]suharjito@binus.edu

ABSTRACT

Plagiarism has become one of the most concerned problems since there are several kinds of plagiarism that are hard to detect. Extrinsic plagiarism is now being handled well, but intrinsic plagiarism is not. Intrinsic plagiarism detection is being distracted by the mixed up structure and the using of another word which have the same meaning. Several methods have been research to handle this problem, not only using the pattern reading but also parsing the sentences, but still couldn't give one more accurate way to detect it. In this research, we propose to use *Natural Language Processing* (NLP) to create the new way to detect plagiarism. Begin with using syntactic parsing method to parse the suspicious document and find the list of words which have the same meaning with it while considering the *Part-Of-Speech* (POS) element of that word (*semantic parsing*). This algorithm also includes creating the new structure of the object before comparing them. The result of this research presents the accuracy comparison between Ferret, WCopyFind, and this algorithm. This algorithm gives the significant way to detect the plagiarism which is proven by T-Test.

Keywords: *Plagiarism Detection, Natural Language Processing, Intrinsic Plagiarism, Syntactic Parsing, Semantic Parsing, Ferret, WCopyFind, Part-Of-Speech.*

1. INTRODUCTION

Increased use of the internet has brought a lot of influence in social lifestyle not only in rapid improvement of science but also in increasing crimes. Plagiarism is one of its which is citing a part or whole document that have been copyrighted without mentioning the author(s) in the correct way. It is also described as a form of stealing other people's ideas by copying intrinsically or extrinsically, but still closely resembles the idea of the source document without mentioning its author correctly. [1]

Based on the research of 6,096 undergraduate students at 31 universities, 67.4% was found committed in plagiarism. The results of similar study on several different campuses with more than 6,000 participants from the high school and undergraduate students, showed 76% were found committed in plagiarism. [2]

Several kinds of plagiarism detection methods have been developed, but not all kinds of plagiarism can be detected. Plagiarism is divided into two types which are *Intrinsic* and *Extrinsic* Plagiarism. Four kinds of intrinsic plagiarism that should be considered in the detection of plagiarism are *Near Copies*, *Disguised*, *Translated*, and *Idea*. *Near Copies* is a form of plagiarism that copying (exactly the same with) the source without citing it in the right way, *Disguised* is a form of plagiarism that cites a part or whole document, restructures the sentences and changes the words with the similar words, *Translated* is a form of plagiarism that translates source document into foreign language, and *Idea* is a form of plagiarism that changes everything using its own structure and words but discuss about exactly the same topic with the source ones. Based on the research, *Near Copies* can be detected very well, but *Disguised*, *Translation*, and *Idea* plagiarism is still hard to detect. [3] [4]

This research will design a pattern of plagiarism detection algorithm using NLP to analyze the structure of the sentences grammatically and collect the similar words with the same element (*Part-Of-Speech*) and use it to determine the strength of plagiarism between the documents and classify them as an intrinsic or extrinsic plagiarism. In the end of this research, there will be comparisons

Journal of Theoretical and Applied Information Technology
10th May 2014. Vol. 63 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

between this algorithm and Ferret and also WCopyFind.

2. NATURAL LANGUAGE PROCESSING (NLP)

Natural Language Processing (NLP) is a method to translate the sentences into another form that can be logically processed and its meaning can be understood by computer. *Chomsky Normal Form* (CNF) is one of NLP method which is used to parse sentences into words and each word will be analyzed and given a tag that define the *Part-Of-Speech* of its. The results of CNF can be illustrated through the Grammar Tree where each element of the sentence will be split and have their own *POS-tag*. Lexical, Semantic, and Syntactic Analysis are the applied knowledge of Natural Language Processing. [5]

Context Free Grammar (CFG) or usually called Right Linear Grammar is an improvement of *Chomsky Normal Form* (CNF) where each token is a non – terminal that can be derived. In this case, the non - terminal tokens can be described as variables and terminal tokens as constants in algebra equations. [6]

2.1 Plagiarism Detection Algorithm Based on Semantic Analysis

Plagiarism detection algorithm based on semantic analysis is the applied of natural language processing (NLP) in plagiarism detection that parses sentences into terminal tokens (words) which might have another words with the same meaning in order to be able to detect plagiarism. The terminal token that have another branch means that the token (word) have similar token (word). One of the most well-known methods using semantic analysis in detecting plagiarism is *Latent Semantic Analysis* (LSA). LSA is measuring similarity between two words by measuring the cosine value of two vectors which are reflected by the compared words. The smaller the value, the words are more similar. [7] [8]

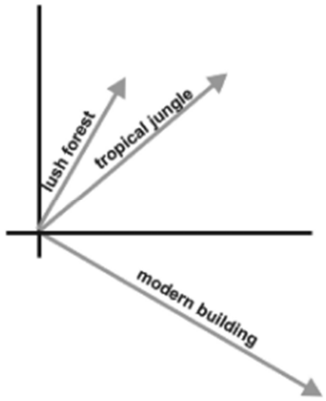


Figure 2.1: Sample of the vectors in LSA. [9]

In Figure 2.1, the words that have similar meanings tend to be put on the same quadrant and the words that have different meanings will be placed in a different quadrant. The closer the vectors located, the more similar the words are. The steps of LSA algorithm are described below:

1) Analyzing the contents of the document and build dimensional matrix where each row represents a unique word and each column represents a document, a paragraph, a sentence, and so on.
2) Vector will be built based on the measurement linguistic of the similar word related to the compared word.
3) The initial matrix will be decomposed into Singular Vector Decomposition (SVD), a mathematical technique to decompose the matrix X into three other matrices (decompose into k number, according to the k value given). Vectors of the similarity matrix described as U, singular vector described as S, and the vector of a document described as V. So the equation is $X = USV^T$ where U-V and S-V. This makes it possible to perform comparisons between a word and another word (either from a collection of words, sentences, paragraphs, essays, and summaries). The similarity measurement is done by measuring vectors distance. If the vectors are located side by side (*Adjoining Vector*), then they have similar meaning.

The equation of cosine ratio between two vectors is given in equation (1) below, where *Vw1* is the vector of a sentence that will be compared, *Vw2* is the vector of sentences or document source that will be compared, and *k* is the dimension of the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

number of documents will be compared to the existing document. The equation of distance measurement between two vectors is described in equation (2).

$$Cos(Vw1, Vw2) = \sum_{i=1}^{k} (Vw1_i \cdot Vw2_i)/(|Vw1| . |Vw2|) \quad (1)$$

$$Dis(Vw1, Vw2) = \sqrt{\sum_{i=1}^{k} (Vw1_i - Vw2_i)^2} \quad (2)$$

2.2. Plagiarism Detection Algorithm Based on Lexical Analysis

Plagiarism detection algorithm based on lexical analysis is the applied of natural language processing (NLP) in plagiarism detection that parses sentences into tokens (a token represented a word) that will be compared to another tokens lexically. [10]. One method that is commonly used is *character n-grams*. This algorithm is resistant to disturbances such as *noise*. This algorithm usually used to detect plagiarism by the *style of writing*, but this method has a disadvantage in determining plagiarism in short sentences. These techniques make comparisons based on cutting the sentences into pieces of words with length adjusted by *n*. Cutting position of the next *n-gram* will start from last shift's position *n-gram* until last character and the number of cutting is according to *offset value*. The parameter *n* depends on the division that will be used by the n-gram method. For example, if n-grams created from the merging of the words then *offset* is the value of the passed word when it has created the next n-gram. If the n - gram is made by combining several letters without counting the last index of character, then *offset value* will represent the value of the passed letters when the next *n-gram* is made. *N-gram* has various division values compared to other plagiarism detection methods which have the same approach. N-gram's cutoff is divided into 2 kinds which are described below:

- *Overlapping n-grams*, each n-gram start at the next position where the pieces have the same word with n-gram before. For example, cutting the word "ABCDEBHAAC" into n-grams with a value of n = 3 and *offset* = 1 (which determines the value of the letter that will be passed). The result of the *n-grams* are "ABC", "BCD", "CDE", "DEB", "EBH", "BHA", "HAA", and "AAC".
- *Non-overlapping n-grams*, there is no *n-grams* that are built from the same starting point with the last *n-grams*.

Equation (3) describes the similarity measurement based on *n-grams* algorithm.

$$sim(A, B) = \frac{Gram(A) \cap Gram(B)}{Gram(B)} \times 100 \quad (3)$$

Where, "A" represents the number of n-gram in document that will be compared against document "B". However, n-gram method doesn't work well on short sentences. This is because the two sentences will do a comparison between the segmentation of two sentences and identify paragraphs whether the paragraphs have the same writing style. [7] [11] [12]

2.3. Plagiarism Detection Algorithm Based on Syntactic Analysis

Plagiarism detection algorithm based on syntactic analysis is the applied of natural language processing (NLP) in plagiarism detection that parses sentences into tokens then analyzes structural pattern of each word according to its position in the sentence. Context-Free Grammar (CFG) is the applied knowledge of this algorithm and the result will be shown as a parse tree. Parse tree is the tree that represent the structure of the words in the sentences and also describe each word element and define it whether it is a phrase or not. The similarity measurement is done by analyzing the similarity of each word and considering its element using *syntactic dependency trees*. [7][13][14]. Some common methods that are used on this algorithm are listed below:

- Top - Down Parsing, the parsing process begin from node S (sentence) and then parse into NP (Noun Phrase) and VP (Verb Phrase) until the last node.
- Bottom - Up Parsing, the parsing process begins from the first word in a sentence and builds the parse tree.
- Depth - First Parsing, the parsing process begins by finding the deepest node and the nodes will expand while reaching any node on the tree.
- Repeated Parse Sub-trees, the parsing process is repeated in every sub-tree. This method is designed to solve the problems of ambiguity and to improve the efficiency of other parsing methods.
- Dynamic Programming Parsing Algorithms, using partial parsing method to solve the problem of ambiguity.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Plagiarism detection based on syntactic analysis is applied in *Part-Of-Speech* (POS) which has the same method with Context Free Grammar. This algorithm begins by parsing a sentence into tokens (words) with given *POS-tag*. Similarity measurement of this algorithm is shown by equation (4):

$$Sim = \frac{num\ (matched\ words\ with\ identical\ tag)}{num\ (matched\ words)} \quad (4)$$

Below is a sample table of POS parsing process using simple sentence.

Table 2.1: Sample Comparison Between Two Sentences That Has Been Paraphrased.

Sentence 1 (S1) : The man likes the woman				
Sentence 2 (S2) : The woman is like by the man				
Word	S1 : Tag	S2 : Tag	S1 : Phrase	S2 : Phrase
man	NN	NN	NP	PP
like	VBZ	VBZ	VP	PP
woman	NN	NN	VP	NP

Table 2.1 shows a simple plagiarism detection using simple sentence based on POS parsing algorithm. The first sentence (S1) is the original sentence in the source document, while second sentence (S2) is a plagiarism sentence that has been changed. However, this method still has disadvantages such as not having capability to handle the paraphrase plagiarism very well due to the change of the structure and the use of different words. To obtain more accurate results, this algorithm will use with semantic analysis to handle the changing structure and the use of different words. This algorithm only focused on matching the words with the same tag. However, the paraphrased sentences also have different tags caused by the changing structure of the original sentence. [15][16]

2.4. Plagiarism Detection Algorithm Based on Grammar Analyzing

Plagiarism detection algorithm based on grammar analyzing is an improvement method of syntactic analysis. This algorithm uses context free grammar (CFG) concept to analyze the plagiarism. This method aims to analyze the type of plagiarism that had been paraphrased. Some algorithms which are applying this algorithm concept are *Plag-Inn* and *APL2*. *Plag-Inn* is a plagiarism detection algorithm with the approach to detect plagiarism by checking the grammar of the author. *Plag-Inn* algorithm process is done without doing a comparison with other documents to detect plagiarism. This algorithm uses pattern analysis to analyze the grammar whether the grammar pattern always the same or changes significantly. When the grammar changes, it will calculate how big the changes is and define the plagiarism value on that. However, this algorithm has disadvantages since it is not doing any comparison with any document. Another algorithm that applied this concept is APL2. APL2 is also parsing the document with the concept of *context free grammar*, but uses *Minimum Spanning Trees* (MSTs) to define the similarity. This algorithm similarity measurement is done by calculating the *means* of its sentences according to its grammar. This algorithm differs from Plag-Inn that does not do comparison with other documents. This algorithm also has disadvantages which is the limitation of using this algorithm for *source code* only. [17][18]

3. PROPOSED ALGORITHM

In this study the proposed plagiarism detection method is focusing on matching each paragraph on suspicious journal with the source journal. Considering that plagiarism is citing a part or whole document from another documents, not only *copy-paste* plagiarism, but also paraphrasing plagiarism might be happen. These are the things considered why this algorithm is focusing on paragraph matching.

- A sentence has a main idea. The main idea is an element of a main idea in a paragraph.
- A paragraph consists of several sentences to express the main idea of the main topic in that document.
- Different paragraphs expressing the difference or contrast main idea which also needs to be combined to express the topic of the document. As long as the main idea is still the expressing the same thing, it would put on the same paragraph.
- Plagiarism is done by taking the same point or the same idea from another document.
- A main idea of the sentence is determined by elements builder of the sentence, in this case, the element define by POS (*Part-of-Speech*).
- The element of sentence that has not changed in the document and the document source of plagiarism is the used of the same word or a synonym that has

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the same meaning with the same POS. Because, some words with a certain element will always be the same and it also be the main element of a sentence that build the main idea of the sentences.
- A sentence that has a word or a synonym with the same POS will build the same main idea. The more they are, the more similar the document is.

Detection process is carried out by the two main process, patterns analyze (*Part-Of-Speech*) and similarity adjustment. In patterns analysis, POS is focusing on the main elements, namely Noun, Verb, Adjective and Adverb. In similarity adjustment, the list of similar words will be collected using an appropriate synonym libraries and databases that have been provided by *WordNet*. Detection process begins by measuring similarity and looking for the words and its similar words while considering its POS tag. After the third process is done it will be found that the document is plagiarism or not. The following diagram (Figure 3.1. on Appendix) shows the proposed algorithm in this research.

The proposed plagiarism detection algorithm is divided into three phases. Below is the detail explanation of each step in this proposed algorithm.
Phase I: *Suspicious Journal Parsing Side*
- Suspicious journal will be parsed into paragraphs.
- Each paragraph will be parsed into sentences.
- Each sentence will be parsed again using POS-tagger algorithm into words with POS tags.
- The words with POS tag will be transformed into *metadata,* where each metadata is represents each paragraph so a document will be represented by several metadata objects.
- Each redundancy of the same word which is represented in metadata will be reduced according to its POS tag.

Phase II: *Database Journals Parsing Side*
- The journals in the database will be parsed into paragraphs.
- Each paragraph will be parsed into sentences.
- Each sentence will be parsed again using POS-tagger algorithm into words with POS tags.
- The words with POS tag will be transformed into *metadata,* where each metadata represents each paragraph so a document will be represented by several metadata objects.
- Just like on the suspicious journal, these journals redundancy also will be reduced.
- Metadata that has been reduced will be paired with its similar words with the same POS tag and will be stored in *List of Object*.

Phase III: *Detection Processing Side*
- Similarity measurement value is collected by matching the words on each paragraph in suspicious journal with each paragraph in database journals. Equation that will be used is *Jaccard Similarity Coefficient.* Matching process is comparing the same word or similar word with the same POS.
- The value of each metadata (in this case, metadata represent a paragraph in journal) will be compared to each metadata in that journal compared. The biggest value will define the similarity of the suspicious journal and the specific journal in database. The rest of data will be store as the result data which are the original text of the biggest value's paragraph, journal title, author(s), volume, issue no, and website of journal.

These phases above illustrate the algorithm which will be applied in this study grouped by the phases of the processes. The general algorithm is described below:
- Journal that will be compared is parsed per paragraph.
- Each paragraph will be parsed into sentences.
- Sentence that has been parsed will be parsed using *POS-tagger* algorithm (Using Library of Stanford NLP). In this step, the words which have no meaning (e.g. like indefinite and definite article, adverb preposition, conjunction, and so on) will be eliminated. Only the main words that can build the main idea of the paragraph will be stored, such as nouns, verbs, adjectives, and adverbs.
- Every same word with the same POS will be grouped into an object that contains the word, the show up counter of that word in that paragraph, and the POS.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

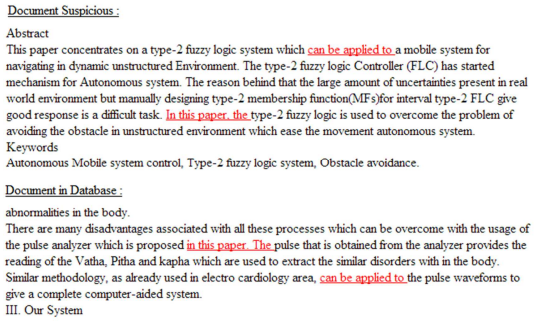
- For Journal on database, every words that has been grouped and eliminated in each metadata will collect the similar words with the same POS (*WordNet* is the Library and Database used in this research).
- At this step, the suspicious journal and each journal in database will be doing the similarity measurement where the comparing process focuses on the paragraphs. Each word in each paragraph in the suspicious journal will be matched with each word in each paragraph in each journal in database. The matching process of the words focuses on the same word with the same POS (Matching Process I). *Jaccard Similarity Coefficient* equation is used as the similarity measurement. If the result is less than 0.9, then both of each word on those metadata will be having similarity matching which match each word on suspicious journal with each word on database journal with the same POS (Matching Process II). The results of the matching process I and II will be compared and the biggest value will be set as the similarity value of those paragraphs. [11]
- The results that will be shown are the value of plagiarism between each journal on database, journal title, author(s), year published, volume, issue number and description of each paragraph with the largest value also the original text of its.

4. RESULT AND DISCUSSION

4.1. Result

4.1.1. Result Using WCopyFind

Figure 4.1.1.1 shows the comparison result of two journals using WCopyFind. Since WCopyFind based on *word-grams* algorithm, this figure below shows that the matching is done by fragmented word with specific count. The comparison focuses on whole document. So, this method matching the specific words fragment even every fragment position is located far apart. [19]


Document Suspicious :

Abstract
This paper concentrates on a type-2 fuzzy logic system which can be applied to a mobile system for navigating in dynamic unstructured Environment. The type-2 fuzzy logic Controller (FLC) has started mechanism for Autonomous system. The reason behind that the large amount of uncertainties present in real world environment but manually designing type-2 membership function(MFs)for interval type-2 FLC give good response is a difficult task. In this paper, the type-2 fuzzy logic is used to overcome the problem of avoiding the obstacle in unstructured environment which ease the movement autonomous system.
Keywords
Autonomous Mobile system control, Type-2 fuzzy logic system, Obstacle avoidance.

Document in Database :

abnormalities in the body.
There are many disadvantages associated with all these processes which can be overcome with the usage of the pulse analyzer which is proposed in this paper. The pulse that is obtained from the analyzer provides the reading of the Vatha, Pitha and kapha which are used to extract the similar disorders with in the body. Similar methodology, as already used in electro cardiology area, can be applied to the pulse waveforms to give a complete computer-aided system.
III. Our System

Figure 4.1.1.1: Sample of one comparison results using WCopyFind.

Figure 4.1.1.2 shows WCopyFind's program after detection done. It shows the result in sides, suspicious journal's side and compared journal's side. So, the similarity measurement is calculated according to the suspicious journal compare to compared journal and also the opposite. [19]

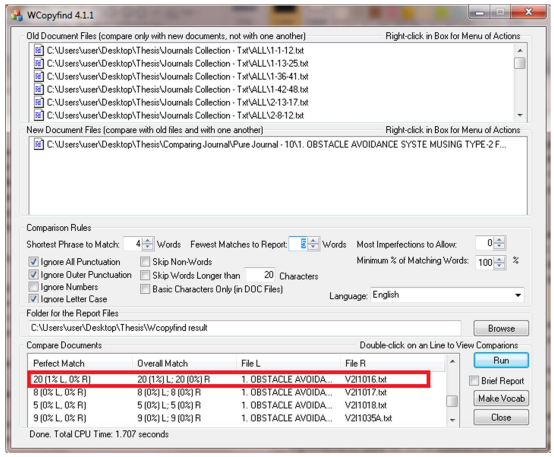


Figure 4.1.1.2: Result of the comparison above using WCopyFind

4.1.2. Result Using Ferret

Figure 4.1.2.1 shows the comparison result of two journals using Ferret. Ferret with *n-gram* algorithm. This figure below shows that the matching is done by fragmenting word with specific count. In this case, Ferret using *tri-grams* algorithm. The comparison focuses on whole document. So, this method matches the specific words fragment even every similar fragment located in the different paragraphs. [20] [21]

Journal of Theoretical and Applied Information Technology
10th May 2014. Vol. 63 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

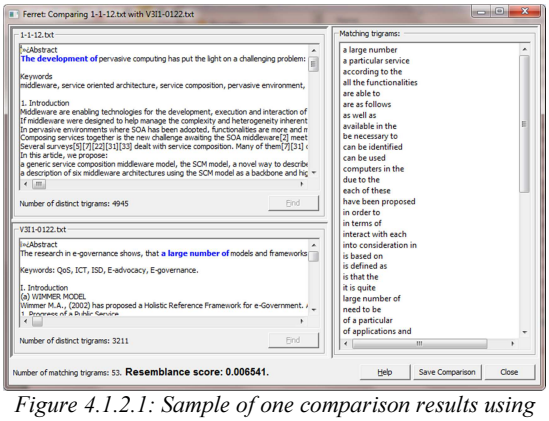


Figure 4.1.2.1: Sample of one comparison results using Ferret.

Figure 4.1.2.2 shows Ferret's program after detection done. It shows the result only in one way. So, the similarity measurement is computed according to the suspicious journal compare to compared journal. [20] [21]

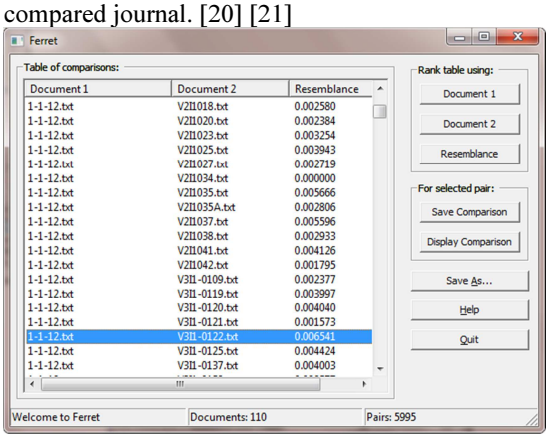
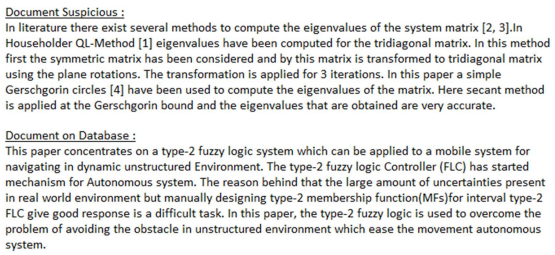


Figure 4.1.2.2: Result Of The Comparison Above Using Ferret

4.1.3. Result using Grammar Analyzing

Figure 4.1.3.1 shows the comparison of paragraphs from two journals using Grammar Analyzing, the proposed algorithm in this research, as the sample. Since this algorithm focuses on comparing each paragraph from suspicious journal with each paragraph from compared journal, this figure below shows the paragraph that would be compared. Because of that, this method is able to find the plagiarism with the journal which is built from joining several parts of other journals.



Document Suspicious :

In literature there exist several methods to compute the eigenvalues of the system matrix [2, 3].In Householder QL-Method [1] eigenvalues have been computed for the tridiagonal matrix. In this method first the symmetric matrix has been considered and by this matrix is transformed to tridiagonal matrix using the plane rotations. The transformation is applied for 3 iterations. In this paper a simple Gerschgorin circles [4] have been used to compute the eigenvalues of the matrix. Here secant method is applied at the Gerschgorin bound and the eigenvalues that are obtained are very accurate.

Document on Database :

This paper concentrates on a type-2 fuzzy logic system which can be applied to a mobile system for navigating in dynamic unstructured Environment. The type-2 fuzzy logic Controller (FLC) has started mechanism for Autonomous system. The reason behind that the large amount of uncertainties present in real world environment but manually designing type-2 membership function(MFs)for interval type-2 FLC give good response is a difficult task. In this paper, the type-2 fuzzy logic is used to overcome the problem of avoiding the obstacle in unstructured environment which ease the movement autonomous system.

Figure 4.1.3.1: Sample of the paragraph on suspicious journal compared to paragraph on compared journal.

Figure 4.1.3.2 shows the object that is built from the sentences. Both of objects below represent the paragraph elements that will be compared. Each of these objects below contains the elements which are word, POS tag, POS, word counter show up, and so on. But, in this figure below only shows these four elements. Each word that will be compared will also consider the POS. So, the comparing progress not only based on the word, but also based on the POS. Besides that, this algorithm provides the feature to find the similar word of each word on these objects.

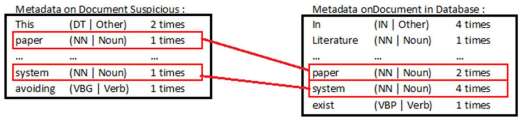


Figure 4.1.3.2: Metadata's Objects Which Are Built From The Sentences On Figure 4.1.3.1.

Figure 4.1.3.3 shows Grammar Analyzing's program after detection done. It shows the result only in one way. So, the similarity measurement is count according to the suspicious journal compare to several compared journals according to the journals saved in database.

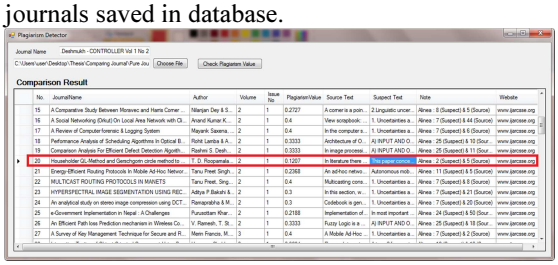


Figure 4.1.3.3: Result Of The Comparison Above Using Grammar Analyzing.

4.2. Discussion

The result of comparison between the three methods which are Ferret, WCopyFind, and this proposed algorithm (Grammar Analyzing) will be explained below. The test will be divided into 4 phases which are

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

- Test Phase I, Test plagiarism detection using different journal that has not bound with the journal collection on database. The sample is 10 journals and tested to 100 journals on database. Each result that shown below is representing the biggest value on one of the journal on database.

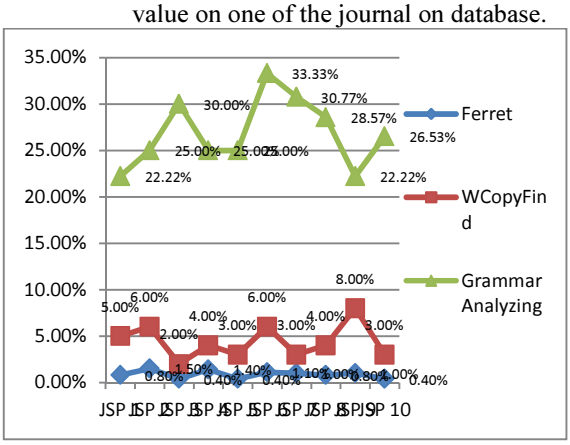


Figure 4.2.1: Precision Value Comparison Chart Between Ferret, Wcopyfind, And Grammar Analyzing For Phase I.

In Figure 4.2.1 shown that the result of precision value proposed on this research is higher than the Ferret and WCopyFind. This is because this plagiarism detection done by matching the text while considering its grammar structure (in this case, represent by POS) of the sentence and also collecting and matching the similar word with the same POS. [19] [20] [21]

- Test Phase II, Test plagiarism detection using same journals with the journal collection on database. The sample is 5 journals and tested to 100 journals on database.

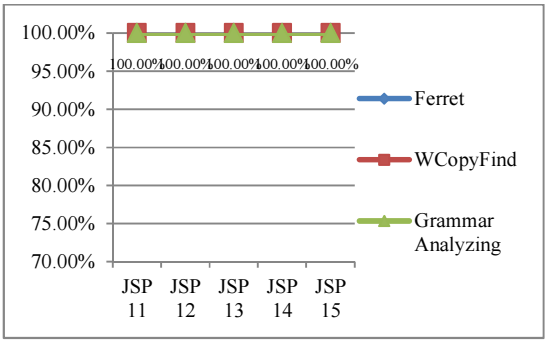


Figure 4.2.2: Precision Value Comparison Chart Between Ferret, Wcopyfind, And Grammar Analyzing For Phase II.

In Figure 4.2.2 shown that all of the method gives the same result for the journal which is a *copy-paste* plagiarism. [19] [20] [21]

- Test Phase III, Test plagiarism detection using mixed up journals with the journal collection on database. These journals are built by combining several journals into a journal. The sample is 5 journals and tested to 100 journals on database.

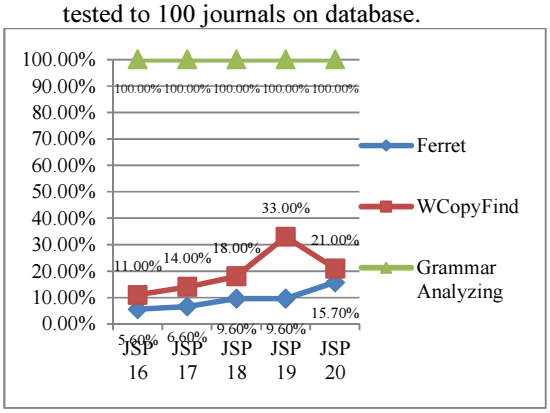


Figure 4.2.3: Precision Value Comparison Chart Between Ferret, Wcopyfind, And Grammar Analyzing For Phase III.

In this phase, the proposed method gives the significant result rather than Ferret and WCopyFind. This is because this algorithm focusing on each paragraph of the journal. Ferret and WCopyFind focus on the whole journal to detect the plagiarism. This made this algorithm could find whether this journal is mixed journal or not while matching and comparing each paragraph of suspicious journal with another journals. [19] [20] [21]

- Test Phase IV, Test plagiarism detection using paraphrased journals with the journal collection on database. These journals are built by *paraphrasing* a paragraph that is *copy-paste* from a journal in database. The sample is 5 journals and tested to 100 journals on database.

Journal of Theoretical and Applied Information Technology
10th May 2014. Vol. 63 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

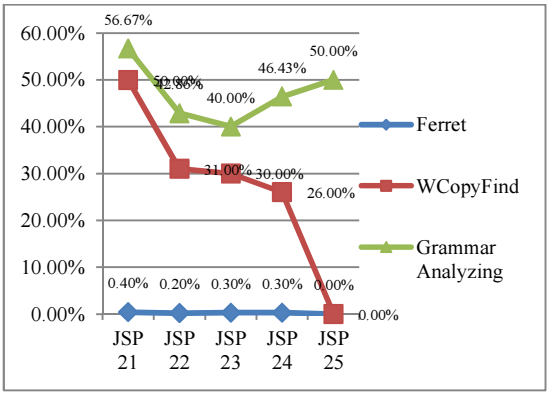


Figure 4.2.4: Precision Value Comparison Chart between Ferret, WCopyFind, and Grammar Analyzing for phase IV.

In Figure 4.2.4, it is shown that Ferret could not detect the paraphrase plagiarism well. However, WCopyFind still can give the good result when the paraphrased journal is more similar the compared journal. But, when the plagiarism became more disguised, WCopyFind is having problem to detect it. Grammar Analyzing as the proposed algorithm in this research give more significant and more stable result than the others. This is because this algorithm provides the similarity matching by taking consideration to analyze it's POS. [19] [20] [21]

Table 4.2.1 (on Appendix) is the comparison table of the three methods above which are Ferret, WCopyFind, and Grammar Analyzing. Where JSP are the suspicious journals and JSR are the journals on database. Precision value shown above is the biggest precision value of precision value with other journals in database.

Table 4.2.2: T-Test Table of Ferret and Grammar Analyzing Comparison.

N = 25	Mean	SD	T	P
Ferret	22.284	39.8439		
Grammar Analyzing	60.184	34.21232	-6.089	2.74E-06

Table 4.2.3: T-Test Table of WCopyFind and Grammar Analyzing Comparison.

N = 25	Mean	SD	t	P
WCopyFind	31.12	37.20475		
Grammar Analyzing	60.184	34.21232	-5.022	3.94 E-05

From table 4.2.1, it shows that Ferret is able to handle the plagiarism detection on *copy-paste plagiarism*, but it is not able to handle several kinds of plagiarism which are *paraphrase* and *mixed up plagiarism*, journal that is built from more than one journal. However, WCopyFind able to handle *copy-paste plagiarism* also still able to give a reliable result for *paraphrase plagiarism* in condition the paraphrased paragraph still closely resembles the original paragraph. But, WCopyFind is not able to handle *hard paraphrase* (completely different with the original) and *mixed up plagiarism*. This proposed method is able to handle all of them also present more accurate similarity value. While the T-Test result on Table 4.2.2 and Table 4.2.3 presents significant value from the comparison of this proposed method to Ferret and WCopyFind which is described by the value of *p* (significant if $p < 0.05$).

5. CONCLUSION

This research gives results of a new algorithm to detect plagiarism more accurately in detecting plagiarism which is capable in dealing with extrinsic or intrinsic plagiarism. This shows that with several changes in detecting plagiarism algorithm below could give more accurate result:

1) The use of *Semantic Parsing* increase the accuracy of detecting plagiarism especially in paraphrased plagiarism,
2) The use of *Syntactic Parsing* in this case POS-tagger in detecting plagiarism. Considering the using of this method while doing the semantic analysis to find the similar words will give the more suitable words to get, and
3) The structuring the words in a sentence before doing the matching will give the efficient time.

6. ACKNOWLEDGE

We would like to thank Sergey Tihon for the *Stanford POS-Tagger Parsing Library* and Matthew Gerber for the *WordNet Semantic Parsing Library* and *WordNet Thesaurus Database*.

REFRENCES:

[1] M. Kashkur, and S. Parshutin, "Research into Plagiarism Cases and Plagiarism Detection Methods", *Scientific Journal of Riga Technical University*, Vol. 44, No. 1, 2010, pp. 138 - 143.

[2] P. M. Scanlon, & D. R. Neumann, "Internet Plagiarism among College Students", *Journal of*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

College Student Development, Vol. 43, No. 3, 2002, 374 - 385.

[3] N. Meuschke, & B. Gipp, "State of the art in the detecting academic plagiarism", *International Journal For Educational Integrity*, Vol. 9, No. 1, 2013, 50-71.

[4] S. Alzahrani, N. Salim, & A. Abraham, "Understanding Plagiarism Linguistic Patterns, Textual Features, and Detection Methods", *IEEE*, Vol. 42, No. 2, 2012, 133-149.

[5] R. Mihalcea, H. Liu, & H. Lieberman, "(NLP) Natural Language Processing for (NLP) Natural Language Programming", *Proceedings of International Conference on Computational Linguistic and Intelligent Text Processing (CICLing)*, Mexico City (Mexico), February 19-25, 2006, pp. 319-330.

[6] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, & P. Kuksa, "Natural Language Processing (Almost) from Scratch", *Journal of Machine Learning Research*, Vol. 12, 2011, pp. 2493-2537.

[7] E. Stamatatos, "Intrinsic Plagiarism Detection Using Character n-gram Profiles", *Proceedings of International Conference on Spanish Society for Natural Language Processing (SEPLN)*, San Sebastian (Spain), September 8-10, 2009, pp. 38 – 46

[8] M. Mozgovoy, T. Kakkonen, and G. Cosma, "Automatic Student Plagiarism Detection : Future perspectives", *Journal of Educational Computing Research*, Vol. 43, No. 4, 2010, pp. 507-527.

[9] G. Botana, J. Leon, R. Olmos, & I. Escudero, "Latent Semantic Analysis Parameters for Essay Evaluation using Small-Scale Corpora", *Journal of Quantitative Linguistics*, Vol. 17, No.1, 2010, pp. 1-29.

[10] D. Micol, R. Munoz, & O. Ferrandez, "Investigating Advanced Techniques for Document Content Similarity Applied to External Plagiarism Analysis", *Proceedings of International Conference on Recent Advances in Natural Language Processing (RANLP)*, Hissar (Bulgaria), September 12-14, 2011, pp. 240-246.

[11] M. Chong, L. Specia, & R. Mitkov, "Using Natural Language Processing for Automatic Detection of Plagiarism", *Proceedings of International Conference on International Plagiarism Conference (IPC)*, Northumbria University (Newcastle), June 21-23, 2010.

[12] T. Kucecka, "Plagiarism Detection in Obfuscated Documents Using an N-gram Technique", *ACM*, Vol. 3, No. 2, 2011, pp. 67-71.

[13] R. Bose, "Natural Language Processing : Current state and future directions", *International Journal of the Computer*, Vol. 12, No. 1, 2004, pp. 1-11.

[14] D. Micol, R. Munoz, & O. Ferrandez, "Investigating Advanced Techniques for Document Content Similarity Applied to External Plagiarism Analysis", *Proceedings of International Conference on Recent Advances in Natural Language Processing (RANLP)*, Hissar (Bulgaria), September 12-14, September 12-14, 2011, pp. 240-246.

[15] L. Ramya, & R. Venkatalakshmi, "Intelligent Plagiarism Detection", *International Journal of Research in Engineering & Advanced Technology (IJREAT)*, Vol. 1, No. 1, 2013,pp. 171-174.

[16] W. Y. Lin, N. Peng, C. C. Yen, & S. D. Lin, "Online Plagiarism Detection Through Exploiting Lexical, Syntactic, and Semantic Information", *Proceedings of International Conference on Association for Computational Linguistics (ACL)*, Jeju (Korea), July 8-14, 2012, pp. 145-150.

[17] M. Cebrián, M. Alfonseca, & A. Orte, "Towards the Validation of Plagiarism Detection Tools by Means of Grammar Evolution", *IEEE*, Vol. 13, No. 3, 2009, pp. 477-485.

[18] M. Tschuggnall, & G. Specht, "Detecting Plagiarism in Text Documents through Grammar-Analysis of Authors", *Proceedings of International Conference on Business, Technology and Web (BTW)*, Otto-Von-Guericke University (Magdeburg), March 11-15, 2013, pp. 241-259.

[19] E. V. Balaguer, "Putting Ourselves in SME's Shoes: Automatic Detection of Plagiarism by the WCopyFind tool", *Proceedings of International Conference on Spanish Society for Natural Language Processing (SEPLN)*, San Sebastian (Spain), September 8-10, 2009, pp. 34 - 35.

[20] J. Malcolm, & P. Lane, "Tackling the PAN'09 External Plagiarism Detection Corpus with a Desktop Plagiarism Detector" , *Proceedings of International Conference on Spanish Society for Natural Language Processing (SEPLN)*, San Sebastian (Spain), September 8-10, 2009, pp. 29 - 33.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[21] A. Bin-Habtoor, and M. Zaher, "A Survey on Plagiarism Detection Systems", *International Journal of Computer Theory and Engineering*, Vol. 4, No. 2, 2012, pp. 185-188.

Journal of Theoretical and Applied Information Technology
10th May 2014. Vol. 63 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

APPENDIX

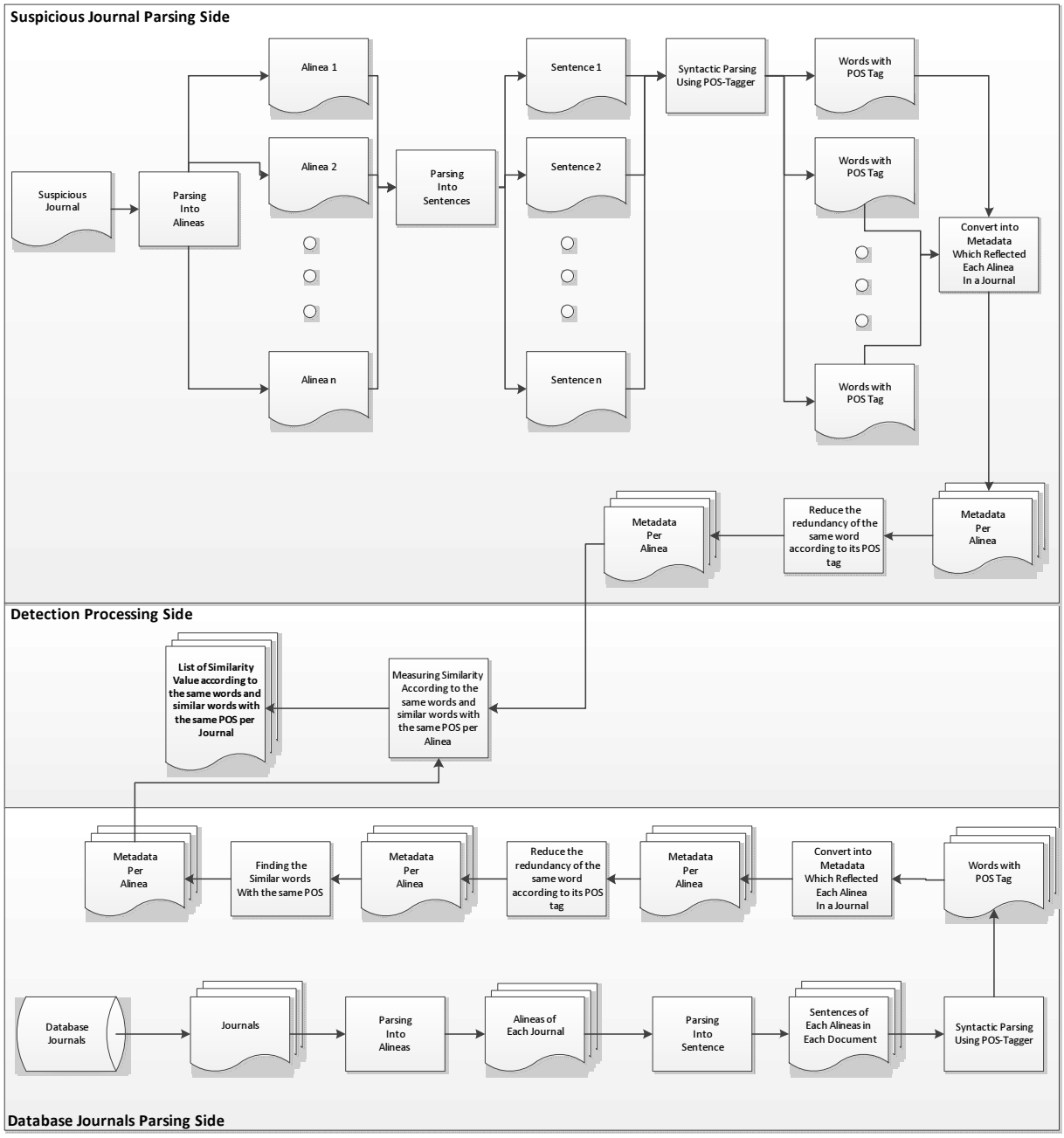


Figure 3.1: Proposed Algorithm in Flowchart Diagram.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 4.2.1: Table of Comparison Result Between Ferret, WCopyFind, and Grammar Analyzing with 25 sample suspicious journals to 100 journals on database.

Suspicious Journal	Precision Value (%)		
	Ferret	WCopyFind	Grammar Analyzing
JSP 1	0.80% [JSR 18]	5.00% [JSR 92]	22.22% [JSR 48]
JSP 2	1.50% [JSR 70]	6.00% [JSR 27]	25.00% [JSR 1]
JSP 3	0.40% [JSR 20]	2.00% [JSR 79]	30.00% [JSR 62]
JSP 4	1.40% [JSR 76]	4.00% [JSR 30]	25.00% [JSR 19]
JSP 5	0.40% [JSR 8]	3.00% [JSR 98]	25.00% [JSR 75]
JSP 6	1.10% [JSR 15]	6.00% [JSR 14]	33.33% [JSR 19]
JSP 7	1.00% [JSR 23]	3.00% [JSR 55]	30.77% [JSR 35]
JSP 8	0.80% [JSR 38]	4.00% [JSR 90]	28.57% [JSR 69]
JSP 9	1.00% [JSR 98]	8.00% [JSR 50]	22.22% [JSR 4]
JSP 10	0.40% [JSR 97]	3.00% [JSR 83]	26.53% [JSR 51]
JSP 11	100.00% [JSR 60]	100.00% [JSR 60]	100.00% [JSR 60]
JSP 12	100.00% [JSR 92]	100.00% [JSR 92]	100.00% [JSR 92]
JSP 13	100.00% [JSR 44]	100.00% [JSR 44]	100.00% [JSR 44]
JSP 14	100.00% [JSR 66]	100.00% [JSR 66]	100.00% [JSR 66]
JSP 15	100.00% [JSR 21]	100.00% [JSR 21]	100.00% [JSR 21]
JSP 16	5.60% [JSR 21]	11.00% [JSR 1]	100.00% [JSR 7, 9, 46, 49, 51]
JSP 17	6.60% [JSR 33]	14.00% [JSR 89]	100.00% [JSR 60, 70, 91]
JSP 18	9.60% [JSR 93]	18.00% [JSR 24]	100.00% [JSR 4, 49]
JSP 19	9.60% [JSR 6]	33.00% [JSR 96]	100.00% [JSR 65, 79, 88]
JSP 20	15.70% [JSR 32]	21.00% [JSR 60]	100.00% [JSR 69, 93]
JSP 21	0.40% [JSR 14]	50.00% [JSR 21]	56.67% [JSR 14]
JSP 22	0.20% [JSR 52]	31.00% [JSR 52]	42.86% [JSR 52]
JSP 23	0.30% [JSR 66]	30.00% [JSR 66]	40.00% [JSR 66]
JSP 24	0.30% [JSR 98]	26.00% [JSR 98]	46.43% [JSR 98]
JSP 25	0.00% [Not Found]	0.00% [Not Found]	50.00% [JSR 92]