1. INTRODUCTION

Plagiarism is stealing and passing off the ideas or words of another as one’s own without crediting the source [1]. Nowadays plagiarism is a big problem in academic and seems to be getting more serious each day. Due to the easy access to the Internet and World Wide Web, many well written articles are available freely and publicly on the websites, thus it is very easy for plagiarist person to obtain the information related to their assignments from the internet.

Following acts are common types of the plagiarisms. The first type is when the whole phrases, sentences, or paragraphs are used directly without any modifications. The second type is when the source is modified so that they do not appear the same. The modifications that often found are; removing some part from the source; inserting more information to the source; and replacing the existing words, phrases, or sentences with their synonyms. The last type is when the structure of the paragraph or the whole article is completely changed. The levels of difficulty in detecting sameness of these three types are easy, moderate, and hard, respectively.

According to IEEE website that defines plagiarism in scientific papers [2], there are important keys to help determining plagiarism. First is the quantity. That is how much in two documents that are identical, for example, a section, a page, a paragraph, or a sentence. Second is the use of quotation marks for all copied text. Third is the appropriate placement
of credit notices, and last is improper paraphrasing. The author of the website [2] suggested that the similarity between two documents should not be greater than 20%.

Several studies have been done to detect the plagiarism. For example, Brin et al. [3] invented document comparison mechanisms called COPS based on sentence unit. Their mechanism works very well when sentences from the source and the target are identical. A major drawback of this work is that it cannot identify partial sentence overlaps. To overcome such a problem, many schemes have been proposed, for example, the works of Shivakumar et al. [4], Si et al. [5], Lodhi et al. [6], Jun-Peng et al. [7]. The first work [4] invented a tool called SCAM that simply employed the document word frequency to check similarity. Their scheme solved the partial matching in sentences, but still it sometimes mistakenly identifies non-plagiaristic documents that also share many words with the source. The second work [5] developed a tool called CHECK uses the document keywords and information depended on document structure to check partial sentence overlap. However, this scheme depends mainly on the structure of each type of document. Lodhi et al. [6] used a string Kernel method which depends on each document's structural information to find common sequences between documents. This method is proved experimentally to be better than the bag-of-words approach in terms of accuracy. Yet, this method suffers drastically from using huge amount of storage space. Similar to Ledhi's work, Jun-Peng et al. [7] use the semantic sequences of documents based on local word's frequency and apply a Kernel function to calculate the similarity between documents. In this paper, we proposed a novel approach that extends the possible lists of phases and sentences, generated by \( s \)-grams.

The \( s \)-gram is a variant of \( n \)-gram [8-10] that allows some terms in \( n \) contiguous sequence to be skipped. The term “\( gram \)” of \( s \)-gram means unit which can be character, word, sentence, or paragraph.

By specifying the number of term skipped \( k \) and the number of remaining grams \( n \) in a sliding window of width \( n+k \), \( s \)-gram can be specifically called as \( s_{n,k} \)-gram according to the number of skips and the number of grams. When \( s \)-gram is applied on word, sentence, and paragraph, it is called word based \( s \)-grams, sentence based \( s \)-gram, and paragraph based \( s \)-gram, respectively. For example, the set of \( s_{0,2} \)-grams and \( s_{1,2} \)-grams of “I know that you know” is \{I-know, I-that, I-you, I-know, know-that, know-you, know-know, that-you, that-know, you-know\}. \( s \)-gram is more flexible than \( n \)-gram since it allows words to be skipped in side of the original sequence where the order of the units are still preserved.

2. Our Approach

Throughout this section, the source document is the original document and the target document is to be examined. Figure 1 illustrates processes used in our approach. Our approach can be described in steps as follows.

For both an original document and a testing document, we apply the following steps to construct a semantic graph. There are two levels of the graph: the sentence level and the paragraph level. For the graph in the sentence level, the node of the graph is in fact just a bag of \( s \)-grams in the word level. The edge of the graph represents the connection between sentences. Likewise, the graph in the paragraph level comprises nodes that are sentence-based \( s \)-gram and edges that show the connection between paragraphs.

First, we break each document into a sequence of sentences. Second, we generate word based \( s \)-grams from each sentence. Last,
we generate sentence based \( s \)-grams from each paragraph. We then compute the similarity between each document using the similarity between nodes and between paragraphs obtained from the source document and the target document.

The following steps are done to compute the similarity between two semantic graphs.

1) Use cosine similarity of node which is represented with word based \( s \)-grams to find nodes that are similar in semantics graphs of a source document and a target document. The node pairs of which their cosine similarities are above the threshold are assumed to be similar. The similarity between two nodes that is represented with word based \( s \)-gram can be formulated as below.

\[
\text{Sim}(\text{Sen}_{S,i}, \text{Sen}_{T,j}) = \frac{\text{Sen}_{S,i} \cap \text{Sen}_{T,j}}{\| \text{Sen}_{S,i} \| \| \text{Sen}_{T,j} \|}
\]

where \( \text{Sen}_{S,i} \) is the \( i \)th sentence node of the source document and \( \text{Sen}_{T,j} \) is the \( j \)th sentence node of the target.

2) Assume that the sentence nodes of the source and of the target of which the similarity between them is larger than some threshold are identical, we find the cosine similarity between paragraph by comparing the sentence based \( s \)-gram constructed of a source document and a target document to compute the similarity between paragraph.

\[
\text{Sim}(\text{Par}_{S,i}, \text{Par}_{T,j}) = \frac{\text{Par}_{S,i} \cap \text{Par}_{T,j}}{\| \text{Par}_{S,i} \| \| \text{Par}_{T,j} \|}
\]

where is the \( i \)th paragraph node of the source document and is the \( j \)th paragraph node of the target document.

3) Compute the similarity between the source document \( (S) \) and the target document \( (T) \) by using the following formula

\[
\text{Sim}(S, T) = \max(\text{Sim}(\text{Par}_{S,i}, \text{Par}_{T,j}))
\]

\[
\text{Dis}_{\text{sen}} = \sum_{i} \sum_{j} \text{Sim}(\text{Sen}_{S,i}, \text{Sen}_{T,j})
\]

Where \( p_{x,y} \) and \( p_{y,z} \) are the \( i \)th paragraph in \( S \) and the \( j \)th paragraph in \( T \).

The following example illustrates how our proposed method works. Consider two versions of sentences in a paragraph of the source document \( (S) \) and a test document \( (T) \) where letters these documents are representing words and numbers are representing extra terms.

\[
S: A B C D. E F G.
T: A 1 B 2 3 C. 4 5 6 7. E 8 G 9 0.
\]

Demonstrating with the maximum number of word two in the word based \( s \)-grams, two nodes in \( S \) are represented with the following word based \( s \)-gram:

\[
\text{Sen}_{S,1} = \{ \text{ABCDA-BA-CADB-C-BD-C-D} \},
\text{Sen}_{S,2} = \{ \text{EF-G-E-F-G-E-G} \},
\]

where \( \text{Sen}_{S,1} \) and \( \text{Sen}_{S,2} \) represent the first node obtained from the first sentence and the second node obtained from the second sentence of the source documents, respectively. In comparison, three nodes in \( T \) are

\[
\text{Sen}_{T,1} = \{ \text{A 1 B 2 3 C A-1 A-B A-2 A-3 A-C 1-B 1-2 1-3 1-C B-2 B-3 B-C 2-3 2-C 3-C} \},
\text{Sen}_{T,2} = \{ \text{4 5 6 7 4-5 4-6 4-7 5-6 5-7 6-7} \},
\text{Sen}_{T,3} = \{ \text{E 8 G 9 0 E-8 E-G E-9 E-0 8-G 8-9 8-0 G-9 G-0 9-0} \}.
\]

The graph representation of these documents using sentence node and paragraph node is below.

With a threshold of 0.3, we find that \( \text{Sen}_{S,1} \) is similar to \( \text{Sen}_{T,1} \) and \( \text{Sen}_{S,2} \) is similar to \( \text{Sen}_{T,3} \) with \( \text{Sim}_{\text{sen}} \) 0.41 and 0.32, respectively. Assume that \( \text{Sen}_{S,1} \) and \( \text{Sen}_{T,1} \) and \( \text{Sen}_{S,2} \) and \( \text{Sen}_{T,3} \) are equivalent we replace \( T_{\text{sen}1} \) and \( T_{\text{sen}3} \) with \( S_{\text{sen}1} \) and \( S \). The sequences of sentences in \( S \)
and $T$ are then reduced to:

$S$: $Sen_{S,1}, Sen_{S,2}$

$T$: $Sen_{T,1}, Sen_{T,2}, Sen_{T,3}$.

Apply the $s$-gram concept to these sentences in each paragraph to create a node, we obtain:

$Par_{S,1} = \{ Sen_{S,1}, Sen_{S,2}, Sen_{S,1} - Sen_{S,2} \}$

$Par_{T,1} = \{ Sen_{T,1}, Sen_{T,2}, Sen_{T,1} - Sen_{T,2} \}$.

As our example has only one paragraph for both the source document and the target.

Figure 1. A flow chart illustrating how our approach is processed.
document, each document comprises only one node in a graph representation.

We find that the similarity between Par$_{S,1}$ and Par$_{T,1}$ is 0.71. Globally, we find that the plagiarist level of these two documents is 0.52.

3. EXPERIMENT

For document preparation, we collected online essays about global warming from many websites that are available freely on the internet. Each essay has a length of 0.5-2 pages (1-4 KB in size). These documents are used as the source documents. Table 1 contains the list of documents and types that are used in our experiment.

For word based s-grams and sentence based s-gram, we applied s-gram with both the maximum number of grams and the maximum of skips three. In our proposed method, we used a threshold of 0.3 to be a cutoff of for a node similarity.

We synthesized plagiaristic documents by modifying documents such as adding meaningless units to or removing existing units from the original message where the units are sentences, paragraphs, and also restructuring.

<table>
<thead>
<tr>
<th>Document Types</th>
<th>Description</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>Source documents</td>
<td>50</td>
</tr>
<tr>
<td>Plagiaristic documents</td>
<td>Adding extra terms to and removing some terms from the source document.</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Adding extra sentences to and removing some sentences from the source document.</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Replacing some terms with their synonyms</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Re-structure documents</td>
<td>50</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>250</td>
</tr>
</tbody>
</table>

The recall is used to measure how complete we get all solution whereas the precision tells how accurate is the solution returned from the method.

5. RESULTS AND DISCUSSION

The results of the run from our proposed method against two baselines are given in Table 2. The results show that our s-gram based semantic graph scheme outperforms the vector space model and the String Kernel method. The improvement on recall are precision are up to 4.46% and 9.4%, respectively compared to the string kernel methods.

Since the vector space model evaluates the similarity between two documents
globally; two documents that share many common terms with high frequency get high similarity, the results are not very accurate. As our document collection is about global warming, many non-plagiarist documents share many common terms with the source resulting high similarity. This explains why the recall and precision is considerably low for the vector space model.

In comparison to the string kernel method that detects the plagiarism from the close locations of the common words between two documents, this approach performs better than the vector space model but still when the plagiaristic documents are inserted with many extra words to and removed many words from the documents, they usually get a low score due to the sensitivity of the distance between it. In addition, when the documents are sharing many common terms which they sometimes happen to be close, the string kernel usually mistakenly assumes the non-plagiarist documents to be plagiarist one.

We also discovered that the performance of our proposed method depends mainly on the level of modification. We found that it works well when there are many differences between the plagiaristic document and the original document. If they are slightly different, our approach and the String Kernel method perform close to each other. This can be explained that when there are a few modifications to the original method, the number of word based s-grams of the original document is close to those of the plagiaristic document resulting no different than using String Kernel.

6. CONCLUSION

This paper proposed a new way to detect the plagiarism. Majorly used in this work is the s-gram concept, the concept that a series of objects can form many sequences whose object orders may or may not be contiguous. We applied s-gram concept to the sentence level and the to the paragraph level to find the similarity between sentences and paragraphs of two documents, one of which is expected to be the plagiarist versions of the others. We found that our proposed method outperforms an existing work of a standard vector space model, and the string kernel model.

REFERENCES

[1] Merriam Webster, webster.com

Table 2. The average recall ($\bar{R}$) and the mean average precision ($\bar{P}$) obtained from two baseline methods: the vector space model and the String Kernel and from our approach, the s-gram based semantic graph evaluated at top 10.

<table>
<thead>
<tr>
<th>Methods</th>
<th>$\bar{R}$</th>
<th>$\bar{P}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vector space model</td>
<td>30.2</td>
<td>38.7</td>
</tr>
<tr>
<td>String Kernel</td>
<td>47.1</td>
<td>54.2</td>
</tr>
<tr>
<td>S-gram based semantic graph</td>
<td>49.2</td>
<td>59.3</td>
</tr>
</tbody>
</table>


