ISBN : 978-602-6806-05-5

THE USE OF PHRASEWORD AND LOCAL-WEIGTED TERMS AS FEATURES FOR TEXT REUSE AND PLAGIARISM DETECTION

Lucia D. Krisnawati
Teknik Informatika, Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana
Email : krisna@staff.ukdw.ac.id

ABSTRACT

This study presents a framework for detecting text reuse which is based on two novel features for its two different stages. On the source retrieval subtask, it introduces the use of phrasewords, while on the text alignment subtask, significant words weighted locally are introduced as seeds. The experiment results shows that the proposed methods are capable of recognizing not only the (near-) duplicate cases, but partially reused cases, and the paraphrased texts as well.

Keywords: *text reuse, plagiarism detection, phrasewords, local weighting technique*

1. INTRODUCTION

The massive availability of research articles on the Web and their easy access provide two-fold impacts on the academic life. One of them is the possibility of committing plagiarism or reusing the available texts. This condition has triggered researches on automatic plagiarism detection which have flourished well for the last 20 years.

However, most plagiarism detection software and prototypes have not considered using citations as a filter in their detection as in (Khan, et al., 2015; Mardiana, et al., 2015). Even, *TurnItIn* provides an option to its users whether they like to include references and citations to be compared or not. This is against the nature of plagiarism which is defined as "the **reuse** of someone else's prior ideas, processes, results, or words **without acknowledging** explicitly the **original author and source**".[1] Hence, the term *text reuse detection* (TRD) is more appropriate to address this task rather than plagiarism detection.

The rationale lies on the task of text reuse detection which is to locate text similarity regardless of the citation presence. This implies that the scope of text reuse detection task is broader since it deals with both the acceptable and unacceptable reuse of texts (Krisnawati & Schulz, 2017), whereas the unacceptable text reuse is commonly addressed as plagiarism.

[1] A definition provided by IEEE available at: http:// www.ieee.org/publications_standardspublications/rights/plagiarism.html

ISBN : 978-602-6806-05-5

The challenge of TRD's task is set on the reuse cases in which the text similarity degree varies greatly. In regard to this challenge, Bendersky and Croft (Bandersky & Croft, 2009) argue that the task of text reuse detection lies right in the middle of Information Retrieval, which retrieves topically related documents in response to a given query, and (near-) duplicate detection. However, the writer perceived that the similarity degree detected by text reuse checkers spans from semantically similar texts to (nearly) identical texts as shown in Figure 1. This is a really challenging task.

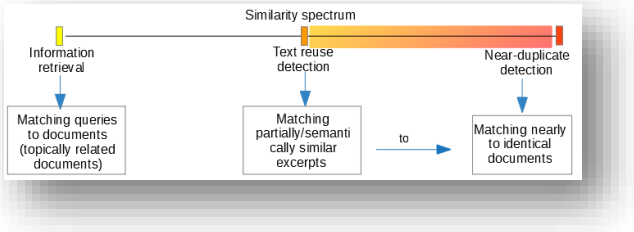

Figure 1. The span of text reuse detection task cf. to (Krisnawati & Schulz, 2017).

Other challenges of text reuse and plagiarism detection (TRPD) task are manifested into 2 subtasks, i.e. source document retrieval and the detailed analysis (Stein, Eissen, & Potthast, 2007). Both subtasks should address the aforementioned challenge, which means that the detector should be able to retrieve a small set of documents which are likely to be the sources of the reused texts, and then extract the source-reused passage pairs having a wide range of similarity degree and length. To cope with these challenges, this study proposes a text reuse detection framework for alternate execution of various detection methods based on distinct document representations in a system workflow. The workflow is schematized as a three-stage approach which consists of a source retrieval, detailed analysis, and post-processing stages. Such workflow is an adaptation of the system architecture proposed in (Stein, Lipka, & Pettenhofer, 2011).

2. PREVIOUS WORK

The current TRPD prototypes perform one subtask only, either source document retrieval (Haggag & El-Beltagy, 2014; Prakash & Saha, 2014) or the detailed analysis subtask (Alvi, Stevenson, & Clough, 2014; Gross & Modaresi, 2014). Only a handful of TRPD prototypes perform both subtasks (Kong, et al., 2015). The previous state-of-the-art algorithms in TRPD did not distinguish their tasks into source retrieval and detailed analysis as in (Charikar, 2008).

ISBN : 978-602-6806-05-5

The approaches of the state-of-the-art algorithms for source document retrieval subtask follow the building blocks proposed in (Potthast, et al., 2012) which comprise of document chunking, keyphrase extraction, query formulation, search control and download filtering. The chunking strategies vary from word-based chunking (Prakash & Saha, 2014) to no chunking, which sees the whole document as one chunk (Elizalde, 2014). The keyphrases are extracted by weighting schema such as tf-idf, BM25, enhanced weirdness or simply using an available tools such as NLTK lemmatization or KP-miner. The query is formulated by selecting top-10 weighted keyphrases (Elizalde, 2014), or the topmost keyterms in a chunk consisting only 1-3 terms (Haggag & El-Beltagy, 2014). The queries will be tailored and fed into an Application Programming Interface (API) of a search engine.

Given a suspicious document, d_{plg}, the retrieved source document candidates are then analysed further to match the similar passages in the detailed analysis subtask, a.k.a. text alignment. The matching features could take forms of a term (Kong, et al., 2015), n-grams, word k-skip n-grams (Gross & Modaresi, 2014), or fingerprints (Alvi, Stevenson, & Clough, 2014) in content-based matching. The structural-based matching will employ stopword n-grams (Stamatatos, 2011) or citation patterns (Gipp, 2014). As for the detailed analysis, there has been 4 (four) approaches applied, i.e. rule-based approach (Stamatatos, 2011), clustering (Gross & Modaresi, 2014), classification (Kong, et al., 2015), and dynamic programming (Glinos, 2014).

The researches on TRPD for Indonesian texts flourished as well. However, our survey on 16 research articles shows that most TRPD for Indonesian (69%) deal with (near-) duplicate detection. Our survey shows also fingerprints generated through Rabin-Karp algorithm and selected through winnowing technique become the favourite features (Suryata, Wibowo, & Romadhany, 2014). The analysis is performed either by rule-based comparison or classification (Alfikri & Purwarianti, 2012).

3. THE PROPOSED FRAMEWORK OF TRPD

As mentioned previously, our TRPD framework is manifested in a system workflow proposed by (Stein, Eissen & Potthast, 2007). However, we simplified it into a two-stage process. The third subtask, the post-processing, has been integrated into the text alignment process. The architecture of our overall system is depicted in Figure 2.

ISBN : 978-602-6806-05-5

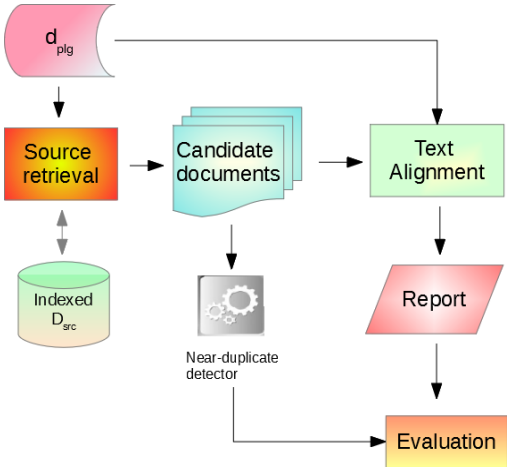

Figure 2 The architecture of our proposed TRPD systems.

Figure 2 presents two separate processes. The first displays the main processes of our prototype, knows as *PlagiarIna*, which ends on the stage where PlagiarIna outputs the Report. The second process is the automatic evaluation which assesses the outputs of both subtasks, i.e. the source retrieval and the text alignment. The detail of evaluation scenario will be discussed in the experiment and result subsection.

3.1 The Source Retrieval Subtask

The building block of our source retrieval subtask consists of text pre-processing, document feature weighting and extraction, query formulation, similarity measurement, and a filtering process. The overall process of source retrieval subtask is illustrated in figure 3. The Pre-processing and feature extraction were applied to both the document collection, in which the source documents, D_{src}, were hidden, and to an inputted suspicious document, d_{plg}. The queries were formulated from d_{plg} only, while the D_{src} were indexed. The similarity between d_{plg} and each $d_{src} \in D_{src}$ was computed and the results were filtered to output the source candidate documents.

ISBN : 978-602-6806-05-5

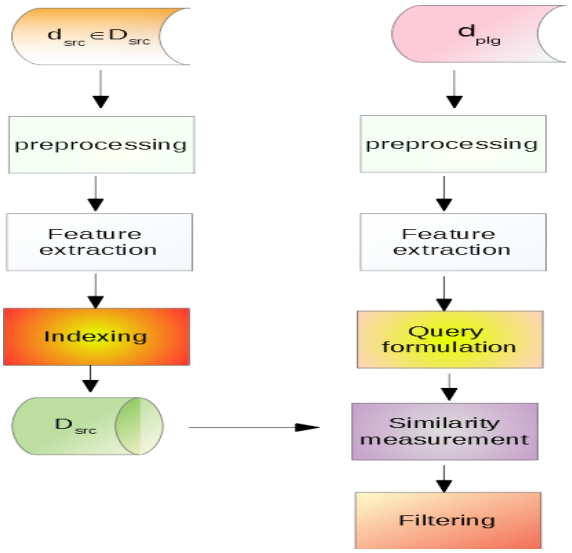

Figure 3. The block diagram of the source retrieval subtask.

A. Pre-processing

The pre-processing is aimed to normalized the text using the common text normalization steps such as case folding, stopword elimination, the removal of non-readable characters, and stemming. However, the treatment to the pre-processing steps were varied during the experiment process.

B. Document Feature Extraction

In this study, we experimented 3 different document features: *Phraseword* which was introduced in (Krisnawati & Schulz, 2013), character n-grams, and word unigram. Phraseword is a meta-term for n-tokens that is designed to capture phrases and consecutive words which have been modified lexically or morphologically. It represents each token in two characters only. The Phraseword building process depends on two parameters which practically defines its types, i.e the token length and the first or the first two characters of a token. The 1st type of Phraseword is formed by the token length and the first token character. The token length is represented by the digit 1-9. Any token length ≥ 10 will be represented by a star sign (*). The 2nd type of Phraseword uses simply the 1st two character of a token. For example, we have two sentences:

(a) Dia menyerahkan diri ke polisi

(b) Mereka menanyai saya tentang uang yang dirampok Amir kemarin.

The phraseword building process after text pre-processing of two sentences above is illustrated in Table 1 below.

ISBN : 978-602-6806-05-5

Table 1. Two types of Phraseword building; the 1st row shows tokens after stopwords removal, while the 2nd shows tokens pre-processed by stemming and stopword list.

Preprocessed tokens	Preprocessed metaterm	Phraseword 3-grams
Type I		
menyerahkan polisi menanyai uang dirampok amir	*m 6p 8m 4u 4d 4a	*m6p8m 6p8m4u 8m4u4d 4u4d4a
serah polisi tanya uang rampok amir	5s 6p 5t 4u 6r 4a	5s6p5t 6p5t4u 5t4u6r 4u6r4a
Type II		
serah polisi tanya uang rampok amir	se po ta ua ra am	sepota potaua tauara uaraam

Along with phrasewords, the character 4-7 grams were used as document features. The rationale of using the short chunk is to make possible the capturing of the morphological modification within a word level. We applied stop n-gram lists which were tailored from our corpus. Another document feature takes the form of a word which undergoes different kinds of token normalization.

The term weighting applied for each document feature is tf-idf which is considered to be a global term weighting. It considers term frequency not only on the whole document but also its occurrences across all documents in the corpus. The computation of tf-idf weight uses the formula [1], where *N* refers to the total number of documents in our corpus, *df* is the document frequency.

$$tf - idf_t = tf_t \cdot \log_{10} \frac{N}{df_t} \qquad [1]$$

C. Query formulation

Our strategy of query formulation considers the fact that the 'similar' passages are unknown and hidden inside the suspicious document, d_{plg}. Thus we consider the possibility of a query distribution on the whole text. The steps of query formulation are:

1. Segmenting d_{plg} into a non-overlapping chunks of 75, 100, 200, 300 weighted phrasewords, n-grams, or tokens.
2. Sorting these features within each chunk according to their weights in descending order.
3. Selecting the top-10 features for each chunk.
4. Merging these selected feature from each chunk into an array.

ISBN : 978-602-6806-05-5

5. Eliminating their redundancy by deleting the similar features.
6. The remaining unique features become the queries of d_{plg}.

D. Similarity Measurement and Filtering

A similarity measure quantifies a similarity between the symbolic representations of two objects and map them into a single numeric value. A high similarity value signifies that two objects share most of their properties. In Information Retrieval field, there lies alternatives of similarity measures between 2 documents. However, this study applied Cosine Similarity (CS) with 3 reasons: CS measures global similarity between 2 documents; it favors rare; and it compensates the effect of document length. The equation 2 presents how cosine similarity should be computed.

The outputted documents are not practically considered as source document candidates, for they are documents which match queries no matter if they match 1 query only. For this reason, we applied a two-step filtering method. The 1st filtering step is to discard documents having less than 3 number of matches. The second step of filtering was based on a threshold. Any document with cosine similarity value which was less than 0.1, 0.05, 0.007, depending on the feature types, was discarded. The remaining documents become the source document candidates, D_{can}.

$$S_{Cos} = \frac{\sum_{i=1}^{|d|} \vec{d}_i \vec{q}_i}{\sqrt{\sum_{i=1}^{|d|} \vec{d}_i^2} \sqrt{\sum_{i=1}^{|q|} \vec{q}_i^2}} \qquad [2]$$

3.2. The Text Alignment Subtask

The outputs of the source retrieval subtask, become the input of the text alignment whose main tasks are to find real-world instances of text reuse and annotate them. Hence, the strategies for locating these similar passage pairs include seeding, seed merging, and seed extension.

A. Seeding

The model used to generate seeds is based on 2 assumptions i.e. a paragraph is the smallest chunk having a single theme which is expressed through several keywords, and these keywords are rarely altered in many cases of reusing academic texts. The context or surrounding words of these keywords are likely to be the objects of alteration. Therefore, this

ISBN : 978-602-6806-05-5

study borrowed the local word scoring proposed by (Kiabod, Dehkordi, & Sharafi, 2012) to generate seeds of a chunk.

Proceeding seed generation, the text normalization and segmentation in the candidate documents, D_{can}, and d_{plg} were performed. The texts were segmented into chunks of paragraphs by the use of a single new line as the delimiter. If the paragraph is shorter than 150 characters which could be subtitles, headings and table or figure captions, then it was merged to its successive chunk.

Applied for generating seeds, this study modified the locality of Kiabod's word local score to a paragraph scope and used 2 statistical criteria: the relative term frequency (TF_{rel}) and a paragraph count (ParCount). TF_{rel} refers to the relative frequency of a term in a paragraph chunk, while the ParCount is the number of paragraph in which a seed occurs normalized by the total number of paragraphs in a text. The modified word local score (Wlscore) is presented in Equation 3, where α refers to a parameter weight in the range of (0-1) and fixed to 0.5.

$$Wlscore = \alpha * TF_{rel} + (1 - \alpha) * ParCount \quad [3]$$

The seeds or the keywords were obtained by discarding the common terms through a threshold defined in Equation 4, where i refers to the word index, and PF to Pruning factor. PF was defined to decide how many percentage of words in a paragraph were selected to be its seeds. We set up PF to be 0.6 in our experiment.

$$\Theta_{Wlscore} = \frac{\sum_i Wlscore_i}{total\ number\ of\ text\ words} * PF \quad [4]$$

The seeds had dual functions: as paragraph queries and a heuristic match for a local comparison. Using paragraph seeds, each paragraph of d_{plg}, $pass_{plg}$, was compared to each paragraph of d_{can}, $pass_{can}$. The similarity between $pass_{pgl} \subseteq d_{plg}$ and $pass_{can} \subseteq d_{can} \in D_{can}$ were measured through Dice coefficient and Jaccard similarity.

In our setting, Dice coefficient was implemented to capture the source-reused passage pairs modified through paraphrase and summary which needs only a handful of queries, while Jaccard similarity was applied to capture as many similar terms as possible. This is to anticipate paragraph reuses with obfuscation cases of *copy and paste*, *shake and paste* or a slight modification by changing the word synonyms etc. In this study, the Dice coefficient

ISBN : 978-602-6806-05-5

computation was based on the seed vectors weighted through the word local score (Eqs 3-4). Its equation was borrowed from (Cha, 2012) and could be seen in Equation 5.

$$S_{Dice} = \frac{2 \sum_{i=1}^{d} P_i Q_i}{\sum_{i=1}^{d} P_i^2 + \sum_{i=1}^{d} Q_i^2} \qquad [5]$$

Where P_i refers to vector value of $pass_{can}$, and Q_i represents the weighted vectors of $pass_{plg}$. Only pairs of paragraphs whose scores are above 0.35 for Jaccard AND 0.4 for Dice coefficient would be processed further.

B. Seed merging

The seed merging function processes only pairs of $pass_{plg}$ and $pass_{cand}$ whose similarity scores are above the defined thresholds as previously explained. The merging of $pass_{plg}$ seed heuristics was performed if there were minimally 2 seed matches found on $pass_{cand}$, and the distance between those seeds was within the gap threshold. We defined two gap thresholds: α and β. α was set to 35 characters in d_{plg} and 50 in d_{src}. In this step, the defined α value produced short sequences of seeds. This is intentionally done as a longer gap will result in a greedy seed merging. These short similar sequences were remerged by considering 2 parameters which are the sequence length (len) and the gap, β. This time, the β was defined to be 75 characters and len is greater than or equals to 35.

C. Extending the Seed Scope

The task of seed merging process is to join the similar seeds found in both pas_{ssrc} and $pass_{plg}$ within a paragraph scope. However, there are very often that the length of text reuse could be more than a paragraph. To cope with this problem, the seed extension function will extend the start and end offsets of similar chunks by rules. The formation of the rules were based on the following 3 types of relations as defined in (Alvi, Stevenson, & Clough, 2014; Krisnawati & Schulz, 2017).:

1) **Containment** which identifies a match within another match. For example, there are 2 pairs of merged sequences with {(x1, y1, l1) → (a1, b1, ln1), (x2, y2, l2) → (a2, b2, ln2)} where *x, y, l* refers to start, end offsets, and length of $pass_{src}$, while *a, b, ln* refer to the same things in a $pass_{plg}$. The 2nd match is said to be within the first if x2 ≥ x1, y2 ≤ y1, and l1 ≥ l2 .
2) **Overlap** which identifies that there is a small portion of a match is within another match or if y2 ≥ y1 ≥ a2 ≥ x1.

ISBN : 978-602-6806-05-5

3) **Near-disjoint** which identifies two matches which share no similar offset and the distance between y1 and a2 are within the gap threshold.

Practically, these relation-based rules will merge two or more matched sequences into a longer one. The matches which fall into far-disjoint relation, i.e whose gap is beyond the threshold underwent no extension.

D. Filtering Process

In (Stein, Eissen, & Potthast, Strategies for Retrieving plagiarized documents, 2007), the filtering becomes a post-processing subtask. However, this study integrated it into the text alignment subtask due to its simple task. The filtering function will remove all matched sequences which are less than 125 characters in $pass_{src}$ which are aligned to the merged sequences which are less than 150 characters in $pass_{plg}$.

4. Evaluation and Experiments

4.1 Evaluation Scenario

To evaluate the proposed methods on the whole framework of our TRPD system, we set up an evaluation scenario comprising of the following stages:

a. evaluating the performance of the source retrieval subtask independently.
b. evaluating the text alignment (TA) subtask by conducting an oracle experiment, and
c. evaluating the whole system performance.

This evaluation strategies let us assess each subtask to its maximum performance. For example, the performance of TA would be hardly measured in cases where not all source documents are retrieved.

4.2. The Evaluation Corpus Building

Our evaluation corpus is a collection of source and test documents comprising of 2247 documents. The source document corpus take a form of bachelor theses, articles, papers in proceedings and journals. Table 2 presents the statistics on our evaluation corpus.

ISBN : 978-602-6806-05-5

Table 2. The statistic data on the evaluation corpus

Corpus	size	Obfuscation types
Source candidates	2014	
Test documents :		
simulated cases	105	literal copy, shake & copy, paraphrase, summary
artificial cases	125	shuffle, deletion, insertion, synonym replacement, deletion & insertion

The generation of test documents were performed through 2 different methods:

1. **Algorithmic generation** which generates a new text out of given source documents by random text operation and semantic word variation. The random text operation was performed by deleting, inserting word from a lexicon, and shuffling the word order. The semantic word variation was done by using *Wordnet Bahasa* to replace some words by their synonyms, antonyms or other related concepts. This method results in artificail cases of text reuse.
2. **Simulative model** in which **test** documents were composed by human writers. This method involved 37 persons who produced 102 reused texts with 4 types of text obfuscation (cf. Table2). The texts produced by this method is addressed as simulated text reuse.

Each test document was completed with its meta-document which saves the information about the obfuscation types, the start and end offsets of the reused passages, and its source document IDs.

4.3. The Evaluation Measures

In measuring the performance of our source retrieval subtask, we adopted the idea of using the near-duplicate detector. Since (near-) duplicates for a d_{src} are hidden and unknown., the only way to find them is by checking the retrieved source candidates, $D_{ret}(d_{plg})$. We built our near-duplicate detector using binary-weigted word unigram and decided a source candidate is defined to be a (near-) duplicate document, $d_{dup}(d_{src})$, if their Jaccard coeeficient is greater than 0.7. The number of $d_{dup}(d_{src})$ found were added to $D_{src}(d_{plg})$ to form a new set $\check{D}_{src}(d_{plg})$. Based on these sets, the precision and recall were defined as follows:

$$Prec(d_{plg}) = \frac{|\check{D}_{src}(d_{plg}) \cap D_{ret}(D_{plg})|}{|D_{ret}(D_{plg})|} \quad [6]$$

ISBN : 978-602-6806-05-5

$$\mathrm{Rec}(d_{plg}) = \frac{|\check{D}_{src}(d_{plg}) \cap D_{ret}(D_{plg})|}{|\check{D}_{src}(D_{plg})|} \quad [7]$$

To evaluate the performances of Text Alignment and the whole system, we applied precision and recall on the character, passage and document levels. In character-level measure, s □ S refers to characters of d_{plg} and d_{src} which specify passages s_{plg} and s_{src}, so does r □ R. r is said to detect s iff s □ r ≠ 0 , r_{plg} □ s_{plg} ≥ 150 characters, and r_{src} □ s_{src} ≥ 125 characters. Based on these definitions, the macro average precision and recall in character level were defined as in (Potthast, stein, Baron-Cedono, & Rosso, 2010):

$$prec_{char}(S,R) = \frac{1}{|R|}\sum_{r \in R}\frac{|\bigcup_{s \in S}(s \sqcap r)|}{|r|} \quad [8]$$

$$rec_{char}(S,R) = \frac{1}{|S|}\sum_{s \in S}\frac{|\bigcup_{r \in R}(s \sqcap r)|}{|s|} \quad [9]$$

The precision and recall on the passage and document levels are similar to the equations in [8,9], only S and R refer to the passages and documents. The drawback of passage-level measure is that the obfuscation type of a reused passage remains unknown. To address this problem, we introduced a measure for recognizing the obfuscation type ($reco_{obtype}$). Let $\boldsymbol{S_C}$ denotes a set of passage pairs having a specific obfuscation type in *S*, and $\mathbf{R_C}$ denotes the same thing in *R*, where S and R refer to the same sets used in the former measures. The $reco_{obtype}$(S,R) of a single obfuscation case in the whole document is defined as in Equation [10]:

$$reco_{obtype}(S,R) = \frac{\sum_{i=1}^{D_C}|S_C \cap R_C|}{\sum_{i=1}^{D_C}|D_C|} \quad [10]$$

To evaluate the robustness of the proposed method, we experimented also test documents with no-reuse or no-plagiarism cases. We perceived that precision and recall become inappropriate measures to recognize the no-reuse cases. Therefore, we introduced a no-reuse recognition measure which is based on the Boolean function. We abbreviated this measure into *noReU*. The Boolean value of bol(S, R) of a given d_{plg} is defined as in Equation [11]:

ISBN : 978-602-6806-05-5

$$bol(S,R) = \begin{cases} 1, \ if \exists bol(\vec{s},\vec{r}) \in bol(S,R) whose \ value \ is \ 1 \\ 0, \ otherwise \end{cases} \quad [11]$$

Based on Eq. [11], the noReU score is defined in equation [12], where N refers to the total number of tested d_{plg} with no-reuse cases:

$$noReU(S,R) = 1 - \frac{\sum_{i=1}^{N} bol(S_i,R_i)}{N} \quad [12]$$

In the evaluation process, we applied F-1 measure which is a harmony mean between precision and recall.

4.4 Result and Discussion

A. Source Retrieval Subtask

In our experiments, we observed the use of 3 document features: phrasewords (PW), n-grams (NG), and tokens (TK). We experimented also the influence of stopping and stemming on each feature by using a frequency stopword list and Tala's stopword list. The variation of using the stopword lists and stemming applied to 3 features produced 14 sets of feature variations.

Our test set consists of 70 documents with 30 documents from artificial cases, 30 from simulated cases, and 10 documents with no-reuse cases. Due to space limitation, we presented only the experiment results in their higher scores. The computation of Macro-Average Precision (MAP), and Recall (MAR) of all tested documents were based on Equations [6,7]. The experiment results on the simulated reuse cases are displayed on Table 3, while table 4 displays the results on artificial reuse cases.

Table 3. Experiment results of source retrieval in simulated reused texts

Features	F-1	MAP	MAR	Information
PW1	0.65	0.64	0.66	2-gram, Tala stopping, no stemming
PW2	0.55	0.66	0.46	4-gram, Tala stopping & stemming
TK1	0.5	0.4	0.67	Freq. stopping only
NG	0.23	0.16	0.63	character 7-gram

Table 4. Experiment results of source retrieval given documents with artificial reused texts

ISBN : 978-602-6806-05-5

Obfuscation	Features	F-1	MAP	MAR	Information
deletion	PW	**0.58**	**0.55**	**0.66**	PW1, 4-grams
	TK	0.45	0.35	0.66	Tala + stemming
	NG	0.18	0.11	0.5	4-grams
insertion	PW	**0.88**	**0.83**	**1**	PW1, 4-grams
	TK	0.54	0.4	0.83	Tala + stemming
del + insert	PW	**0.94**	**0.91**	**1**	PW1, 4-grams
	TK	0.37	0.24	0.83	freq. Stopping only
shuffle	PW	0.44	0.39	0.66	PW1, 2-grams
	TK	**0.6**	**0.47**	**1**	freq. Stopping only
synonym	PW	**0.83**	**0.73**	**1**	PW2, 4-grams
	TK	0.43	0.29	0.83	Tala + stemming

Tables 3 and 4 show that averagely phrasewords perform better than other features. In artificially obfuscated cases, phraseword's recall rate achieves the optimal value – 1. However, token outperforms its performance in retrieving $D_{src}(d_{plg})$ in shuffled cases. The recall rates of token are competitive to phraseword in almost cases, however its precision rates are much lower so that its F-1 rates are quite low. The n-gram scores of all measures on all obfuscation types are the worst. For this reason, Table 4 presents its score only for deletion cases.

B. The Oracle Experiment on Text Alignment subtask

To evaluate the performance of the Text Alignment (TA) subtask, we conducted a comparison between our prototype, *PlagiarIna*, to Alvi's algorithm (Alvi, Stevenson, & Clough, 2014) which makes use of fingerprints as document features. Table 5 presents TA performance on 3 levels of measures for a simulated reused cases.

Table 5. The experiment results of TA performance tested on simulated text reuse corpus.

Systems	Methods	Char-based			Pass-based			Doc-based		
		Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
PlagiarIna	TK1	**.76**	**.60**	**.67**	**.76**	**.66**	**.67**	.89	.72	.80
	TK2	.75	.59	.66	.74	.58	.61	**.92**	**.74**	**.82**
Alvi's		**.76**	.45	.57	.75	.52	.60	.87	.68	.76

Table I shows that averagely the PlagiarIna's F1 scores are higher than Alvi's algorithm. Unlike Alvi's algorithm, PlagiarIna seems to be more stable as its recall rates are almost in balance to its recall. This stability is also reflected to its TA performance results tested on artificial reused texts as displayed in Table 6 below.

ISBN : 978-602-6806-05-5

Table 6. The experiment results of TA performance tested on artificial text reuse corpus.

Obfuscate	Systems	Char-based			Pass-based			Doc-based		
		Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Delete & Insert	PlagiarIna	**.95**	**.84**	**.89**	**.91**	**1**	**.95**	**.91**	**1**	**.95**
	Alvi's	.83	.40	.53	.45	.83	.52	.83	.83	.83
shuffle	PlagiarIna	.57	.08	.14	.58	.66	.66	.58	.66	.52
	Alvi's	0	0	0	0	0	0	0	0	0
synonym	PlagiarIna	.61	.83	.70	.83	.83	.83	.83	.83	.83
	Alvi's	.33	.06	.10	.25	.33	.28	.33	.33	.33

Table 6 shows that PlagiarIna outperforms Alvi's algorithm in all obfuscation types for all levels of obfuscation. Tables 4-6 show that *PlagiarIna's* rates tested on artificial cases are much higher than its rates on simulated ones. This indicates that algorithmically obfuscated texts present few problems to PlagiarIna. In contrast, texts obfuscated by human writers become challenges for our prototype. Some possible explanations are that firstly human writers tend to obfuscate texts on the **different levels of linguistic structure** such as on morphological, or syntactic structures, while the algorithmic obfuscation occurs on the lexical level only. Secondly, the artificially generated documents contain 1 type of obfuscation per document, while those in simulated cases tend to comprise **different obfuscation types per document**.

The recognition rates of the obfuscation types for simulated documents are presented in Table 7. Again, PlagiarIna outperforms Alvi's algorithm in most obfuscation types, except on *Summary*. This table show that PlagiarIna is capable of recognizing paraphrased reuse text till the medium level. The heavy paraphrased and summarized text present problems to its since their recognition rates decline drastically.

Table 7 Alvi's and PlagiarIna's recognition rates on the obfuscation types. In this table para refers to paraphrased; L, M, H refer to light, medium, and heavy levels of paraphrase techniques

Systems	Meth	Copy	para-L	para-M	para-H	shake	smry
PlagiarIna	TK3	.90	.91	.81	.44	.74	.37
	TK2	.81	.88	.48	.48	.70	.37
Alvi's		.68	.55	.42	.42	.64	.45

In detecting no-reuse cases, 3 methods of PlagiarIna reach its maximum rates, 1 for seed units TK2, TK4, and character 7-grams. In general, some PlagiarIna's methods produce rates

ISBN : 978-602-6806-05-5

that higher than Alvi's score except for TK1. Table 8 presents the noReU scores of Avi's and PlagiarIna algorithms.

Table 8. The NoReU scores of Alvi's and PlagiarIna.

	PlagiarIna					Alvi's
	TK1	TK2	TK3	TK4	7GR	
Rates	.80	1	.90	1	1	.90

The experiment results on the whole system performance cannot be presented here due to space limit. The scores of Alvi's and PlagiarIna algorithms in the last scenario show the same tendency as in the scenario of oracle experiment, only they are averagely lower since not all members of $D_{src}(d_{plg})$ are retrieved in source retrieval subtask.

5. CONCLUSION

This study has proved that the proposed framework of TRPD, which experimented several document features and used global weighting techniques on the source retrieval subtask combined with local term weighting on the text alignment subtask, works very well. The phrasewords proved to be a robust feature in retrieving source documents, while the use of significant words proved to be a competitive technique in detecting heavily paraphrased texts. Another strength of our proposed methods is that it produces almost no-overlapping detection. One drawback of this method lies on its passage boundary which may produce an unproper start and end of a sentence.

This study has proved also that the complexity of a test document corpus correlates highly with the text reuse detection system's performance. This is validated by the higher rates on all obfuscation types in all levels of measures for artificial test document corpus than the simulated one. Lastly, this study has successfuly provided a standard evaluation corpus for assessing text reuse detecion systems for Indonesian.

REFERENCES

Alfikri, Z. F., & Purwarianti, A. 2012. The Construction of Indonesian English Cross Language Plagiarism Detection. *Computer Science and Information Journal*, *5*(1), 16-23.

Alvi, F., Stevenson, M., & Clough, P. 2014. Hashing and Merging Heuristics for Text Reuse Detection. *Notebook Papers of PAN CLEF 2014 Labs and Workshops.* Retrieved from http://www.uni-weimar.de/medien/webis/events/pan-14/pan14web/about.html#proceedings.

ISBN : 978-602-6806-05-5

Bandersky, M., & Croft, W. B. 2009. Finding Text Reuse on the Web. *Proc. 2nd ACM International Conf*, (pp. 262-271). ACM.

Cha, S. H. 2012. Comprehensive survey on distance/ similarity measures between probability functions. *Journal of Math. Models and Methods in Applied Sciences, 1*(4), 300-307.

Charikar, M. 2008. Similarity Estimation Techniques from Rounding Algorithm. *Proceeding of 34th Annual Symposium on Theory of Computing (STOC)*, (pp. 380-388).

Elizalde, V. 2014. Using Noun Phrases and tf-idf for Plagiarized Document Retrieval. *Notebook Papers of PAN at CLEF 2014.* Retrieved from http://www.uni-weimar.de/medien/webis/events/pan-14/pan14-web/about.html

Gipp, B. 2014. *Citation-based Plagiarism Detection: Detecting Disguise and Cross-language Plagiarism Using Citation Pattern Analysis.* Wiesbaden: Springer Verlag.

Glinos, D. 2014. A Hybrid architecture of plagiarism detection. *PAN CLEF´14 Labs and Workshop.*

Gross, P., & Modaresi, P. 2014. Plagiarism Alignment Detection by Merging Context Seeds. *Notebook Papers of PAN CLEF 2014 Labs and Workshops.*

Haggag, O., & El-Beltagy, S. 2014. Plagiarism Candidate Retrieval Using Selective Query Formulation and Discriminative Query Scoring. In Forner (Ed.), *Notebook Papers of PAN at CLEF 2013 (2013).* Retrieved from http://www.uni-weimar.de/medien/ webis/events/pan-13/pan13-web/about.html#proceedings

Khan, I. H., et al. 2015. A Freamework for Plagiarism Detection in Arabic Documents. (N. e. al., Ed.) *CCSEA, 5*, 01-09.

Kiabod, M., Dehkordi, M. N., & Sharafi, S. M. 2012. A novel method of significat words identification in text summarization . *Journal of Emerging Technologies in Web Intelligence, 4*(3), 252-258.

Kong, L. et al. 2015. Source Retrieval and Text Alignment Corpus Construction for Plagiarism Detection. In L. Cappellato, N. Ferro, G. Jones, & E. S. Juan (Ed.), *CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers.* Toulose, France.

Krisnawati, L. D., & Schulz, K. U. 2013.. Plagiarism detection for Indonesian texts. (E. Weipple, Ed.) *Proceedings of the 15th Int. Conference on Information Integration and Web-based Applications and Services (iiWAS2013)*, pp. 595-599.

Krisnawati, L. D., & Schulz, K. U. 2017. Significant word-based Text Alignment for Text Reuse Detection. *Int. Conf. on Research and Innovation in Computer, Electronics and Manufacturing Engineering (RICEME-17)* (pp. 7-12). Bali: EIRAI.

ISBN : 978-602-6806-05-5

Mardiana, T., Adji, T., & Hidajah, I. 2015. The Comparation of Distance-based Similarity Measure to Detection of Plagiarism in Indonesian Text. In R. e. Intan (Ed.), *ICSIIT 2015* (pp. 155-164). Springer Verlag.

Potthast, et. Al. 2012. Overview of the4th International Competitionon PlagiarismDetection. In P. Forner (Ed.), *Notebook Papers of CLEF 2012 Labs and Workshops.* Rome, Italy. Retrieved from http://www.uni-weimar.de/medien/webis/events/pan-12/pan12-web/about.html

Potthast, M., et.al. 2010. An evaluation framework for plagiarism detection. *2nd International Conference on Computational Linguistics (COLING'10)*, pp. 997-1005.

Prakash, A., & Saha, K. S. 2014. *Experiments on Document Chunking and Query Formulation for Plagiarism Source Retrieval. .* Retrieved from Notebook for PAN at CLEF 2014.

Stamatatos, M. 2011. Plagiarism detection using stopword n-grams. *American Society for Information Science and Technical Journal, 62*(15), 2512-2527.

Stein, B., Eissen, S. M., & Potthast, M. 2007. Strategies for Retrieving plagiarized documents. *SIGIR 07.* Amsterdam: ACM.

Stein, B., Lipka, N., & Pettenhofer, P. 2011. Intrinsic Plagiarism Analysis. *Journal of Languange Resources and Evaluation, 45*(1), 63-82.

Suryata, A. F., Wibowo, A. T., & Romadhany, A. 2014. Performance efficicency in plagiarism indication detection system using indexing method with data tree 2-3. *Int. Conf. Information and Communication Tech. (IcolICT)*, pp. 403-408.