EASR

Engineering and Applied Science Research

https://www.tci-thaijo.org/index.php/easr/index

Published by the Faculty of Engineering, Khon Kaen University, Thailand

# A novel technique for Thai document plagiarism detection using syntactic parse trees

Wansuree Massagram, Sorawat Prapanitisatian and Kraisak Kesorn*

Department of Computer Science and Information Technology, Faculty of Science, Naresuan University, Phitsanulok 65000, Thailand

Received 11 September 2017
Accepted 16 January 2018

**Abstract**

The act of plagiarism is a serious offense and all involved parties will be penalized according to most Thai university rules. The lack of effective tools for plagiarism detection in the Thai language is a problem for academic and research institutes in Thailand. A practical framework and detection tool would facilitate the development of academic integrity and honesty. This paper presents an effective alternative method to detect plagiarism in Thai academic articles utilizing a syntactic parse tree technique (SPT). The main concept of this method is the dynamic weighing of each sentence according to the roles of its words. The experimental results, empirically compared with three existing tools: tri-grams, semantic role labeling (SRL), Turnitin and Akarawisut, yield comparable or higher precision and recall in all four plagiarism study cases of word-by-word, word-reordering, modifier-insertion, and synonym-replacement plagiarism. SPT shows promise and should be incorporated in similarity comparison tools to improve the accuracy of plagiarism detection in the Thai language.

**Keywords:** Text analysis, Plagiarism detection, Natural language processing, Thai

## 1. Introduction

Research and educational institutes in Thailand have faced increasing problems with violations of academic integrity [1]. According to a survey by the National Institute of Development Administration (NIDA) [2], 16.3% of undergraduate students in Thai universities have cheated on their exams, and 29.7% have helped other students cheat. Students and academic personnel are required to acknowledge the intellectual contributions of others and exhibit fairness and transparency in all aspects of scholarly endeavors [3]. Plagiarism is the act of taking other people’s work or ideas without referring to the source of information. Most cases of plagiarism can be avoided simply by citing sources and acknowledging the contributions in the work [4]. However, a small portion of the Thai academic community thinks that plagiarism is inconsequential. This practice threatens and reduces the research quality and innovation of Thai universities and institutes. This alarming fact leads to the need for effective tools to detect plagiarism in the Thai language.

To deal with plagiarism, the act itself must be detected before it is penalized. Some Thai scholars use “Akarawisut” [5] and “Turnitin” [6] to check for plagiarism in academic articles and dissertations. Turnitin, an effective commercially-available service, is widely used for this task, especially for English documents. Nonetheless, its use for plagiarism detection in the Thai language is still far from practical due to the unstructured aspects of Thai sentences. For example, there is lack of parsing cues, in particular, spaces between words and there are no punctuation marks at the end of a sentences. This leads to ambiguity and difficulty in automated detection algorithms. Chulalongkorn, a leading university in Thailand, has recently developed “Akarawisut” for its students and the general public to serve as a plagiarism checker for the Thai community. However, these tools do not employ semantic-based techniques, which makes certain types of plagiarisms, such as synonym replacement and modifier insertion, difficult to detect.

According to Osman et al. [7], there are four common ways to plagiarize documents: 1) copy and paste, 2) redrafting or paraphrasing of the text, 3) plagiarism through translation from one language to another, and 4) plagiarism of ideas. On the contrary, Ali et al. [8] considered the categories to be textual plagiarism or source code plagiarism. Source code or programming plagiarism can be considered as a structure-based plagiarism, which will be described in more detail in Section 2.2.

The current study focuses on Osman’s first and second methods. This is because the third and fourth methods, cross-language translation and plagiarism of ideas, present other kinds of complexity that requires human analysts to perform such a task. In this study, plagiarism is grouped into four categories: 1) word-by-word plagiarism, 2) word-reordering plagiarism, 3) synonym-replacement plagiarism, and 4) modifier-insertion plagiarism. Word-by-word copying is a technique where text is copied from an original document without any modifications and pasted into a new document. This cut-and-paste method is the easiest to commit and detect with current techniques. In word-reordering, a plagiarizer

*Corresponding author. Tel.: +66 5596 3263
Email address: kraisakk@nu.ac.th

doi: 10.14456/easr.2018.39

swaps the positions of terms in a sentence. This method is more difficult to detect, since a simple string-matching method cannot effectively perform this task. Synonym-replacement techniques modify the original sentences by replacing some words with their synonyms. This method creates different appearances between the original and the plagiarizing sentences, while maintaining similar semantic meaning, and is also difficult to detect with simple tools. Finally, the fourth method of plagiarism in this study is modifier-insertion or adjective-insertion technique, in which the copied sentences are altered with inserted adjectives or adverbs. This approach extends the length of an original sentence and creates difficulty for the plagiarism detection software.

In addition to simply comparing strings for plagiarism, similarity between sentences can be measured semantically based on word roles. This method is called a semantic-based technique. This study proposes a plagiarism detection algorithm based on a semantic role labelling technique with dynamic weighing of a sentence depending the roles of particular words. These algorithms are called syntactic parse trees (SPT). The remainder of this paper is organized as follows. Section 2 presents the background and related work in the field, particularly the effectiveness and limitations of string-based, structure-based, and semantic-based detection techniques as tools for plagiarism detection in Thai documents. The proposed SPT algorithm and framework are shown in Section 3. Section 4 presents comparison results between SPT vs. tri-grams and SRL, when the documents are plagiarized in the four experimental cases of plagiarism. The effectiveness of SPT against Turnitin and Akarawisut, which are commercially available tools, is also illustrated in this section. The discussion and conclusion of this study comprise Section 5.

## 2. Literature review

Humans are the most effective and sophisticated tool for plagiarism detection [9]. However, the vast amount of data available makes this approach impractical. To reduce plagiarism in academia, Lukashenko et al. [10] proposed a two-step approach, plagiarism prevention and detection. Plagiarism prevention includes precautionary measures such as academic policies and/or disciplinary actions. Plagiarism detection uses computerized tools to detect the offense. However, before detection methods can be applied to a suspicious document, it must be quantified and characterized using one or more of the following: lexical, syntactic, semantic, and structural textual features [11]. The lexical features include character n-grams (of fixed or variable length) and word n-grams. The syntactic features include chunks, phrases, sentences, word position/order, and part-of-speech structures. Semantic features include synonyms, hyponyms, and other semantic dependencies. Structural features include block or content specific features considering contextual information. Moreover, some authors plagiarize original work by adding modifiers, i.e., adverbs and adjectives, to sentences. Once the comparison is performed, the similarity score for such a document is biased due to the increased number of modifiers. Most conventional detection tools do not recognize sophisticated copying methods. Although this paper focuses on the Thai language, this section will survey state-of-the art plagiarism detection frameworks for various languages in the past decade. Based on our literature survey, plagiarism detection tools can be classified into three categories, string-based, structure-based, and grammar semantics hybrid techniques.

### *2.1 String-based plagiarism detection techniques*

The similarity between documents can be measured using fingerprints. Document fingerprints can be obtained through application of a Winnowing algorithm. A Winnowing algorithm processes data at the document level by transforming a document into hash numbers using hash functions and encode those numbers as document fingerprints. Winnowing has been applied to plagiarism detection for English and Bahasa by Schleimer et al. [12] and Wibowo et al. [13] respectively. This algorithm can effectively detect the word-by-word and word-reordering approaches to document copying. However, two different documents might accidentally have similar document fingerprints, thus, this method may be imprecise for comparison at the document level. Schleimer et al. [12] proposed deployment of a similar algorithm to process data at the sentence level. By measuring the similarity of sentence fingerprints, the accuracy of document-level detection significantly improves. Moreover, this enables checking of sentences against several other documents. However, limitations still exists when different documents or sentences contain equivalent fingerprints. Gipp and Meuschke [14] proposed a method to detect scientific research integrity based on references and citations. Their approach relies not only on analyzing the body of the text, but also the reference citations. They hypothesized that citations were valuable language-independent markers and have similar fingerprints to the documents in which they are cited. It was revealed that citation sequence in a plagiarized document often remained similar even if the text has been paraphrased or translated to other languages. Nonetheless, the major limitation of this approach is that plagiarism cannot be detected if the authors do not cite the original papers.

### *2.2. Structure-based plagiarism detection techniques*

With the limitations of string-based approaches, some researchers choose to analyze suspicious documents using their structure to improve the detection performance. This approach is called structure-level or grammar-based plagiarism detection. Akewonganone and Aroonmanakul [15] proposed using document structure in the form of trees and compare keywords (only nouns, verbs, adjectives, and adverbs) in tree branches of the suspected document with other documents in repository using a support vector machine (SVM). Their study effectively detected cross-language plagiarism for Thai and English documents.

Copying of a computer source code, which has plagued computer science and engineering departments in most universities, is a main example for this type of plagiarism. Manually detecting similarity is laborious. However, this task can be automated through various techniques, e.g., string matching, tokens, parse trees, program dependency graphs (PDGs), metrics, and hybrid approaches. White and Joy [16] presented an approach to check computer program plagiarism based on the structure of the source code and its comments. The study demonstrated that their proposed method was able to detect sophisticated obfuscation as well as direct copying. However, this method cannot process unstructured documents because of its explicit structure. Jedalla and Elnagar [17] developed a Plagiarism Detection Engine for Java (PDE4Java) which uses three steps: 1) tokenizing of the Java code, 2) measurement of similarity, and 3) clustering. This particular tool is applicable to all programming languages due to its flexibility. Kustanto and Liem [18] developed a tool called Deimos to check source code plagiarism in LISP and Pascal programming languages. Lesner et al. [19] introduced a novel framework to detect source code plagiarisms, which was applied to a corpora of languages and showed good results. Marianiand Micucci [20] also proposed the AuDeNTES technique to identify computer program plagiarism. The experiments showed that AuDeNTES can detect more plagiarism cases than other techniques with small inspection efforts.

### *2.3. Grammar semantics hybrid plagiarism detection techniques*

Ontology is a popular technique for semantic reasoning, comparison, and searching that broadly applies to several research areas. Therefore, some researchers have deployed this approach for plagiarism detection. Liu and Wang [21] used a support vector machine (SVM) to compare the similarity between sentences or between documents using ontology. They transformed text in a document and then matched it to the concepts in the ontology. With the aid of ontology structure, the experiments showed that the proposed technique was efficient and flexible enough to allow the user to make comparisons without any additional dictionary or corpus information. Although this method can show a similarity score between the suspected and other documents in repository, it cannot identify plagiarized sentences in the suspected document to an investigator.

Semantic role labeling (SRL) has traditionally been used to restructure sentences based on word function. Osman et al. [7] proposed a plagiarism detection method based on SRL to compare sentences with semantically similar verbs. The results of their study show improved accuracy in plagiarism detection. However, SRL has two major limitations: 1) it cannot be applied to a sentence containing words with unknown functions, and 2) its detection accuracy is reduced with sentences containing adverbs or adjectives (causing an increase in number of words to be compared).

Based on the analysis of SRL, its limitation occurs when the suspected sentence contains additional modification words. When SRL is used with Thai articles [22] we have found the five following problems:

(1) Thai language does not employ spaces for lexical unit separation, unlike English, which makes SRL sentence restructuring inaccurate.

(2) Thai vocabulary is often ambiguous in both lexical unit separation and word type, which also makes SRL sentence restructuring inaccurate.

(3) If word segmentation is done incorrectly, the semantic comparison will not be correct.

(4) If word roles are assigned incorrectly, those words will not be compared.

(5) If every word is weighted equally, additional modifiers in the suspected article may bias the similarity measurement.

These limitations drive the research objectives described in the following sections. The solutions to these problems are vital to achieving accuracy in the plagiarism detection and are the main focus of this paper.

## 3. Proposed syntactic parse trees framework

This research presents a SPT technique, a modification of SRL, to enhance plagiarism detection, particularly for the Thai language. The SPT framework is presented in Figure 1. It is consisted of five processes: 1) document filtering, 2) word role prediction, 3) grammar mutation, 4) sentence re-structuring, and 5) word role weighting using SPT. First, the title and keywords of the suspected article are mapped to the keywords of the original articles in the existing database. Since Thai language does not employ word boundary markers, the segmentation of lexical units must be done using the SWATH API [23], developed by Thailand's National Electronics and Computer Technology Center (NECTEC). The sentences are then reordered into SRL structure after the segmentation. Finally, in the similarity comparison, each word is compared according to its classification.

### *3.1. Document filtering*

Most of the plagiarism detection process is spent in comparing pairs of articles. Larger datasets of articles take more time to process. In this study, the size of such a set is restricted to 1,000 research articles containing 56,380 sentences in the computer science and information technology research areas. This database contains SRL-structured articles (see example in Figure 1) in XML format. Once the articles from the journal database were selected, the detection process can begin with phase two.

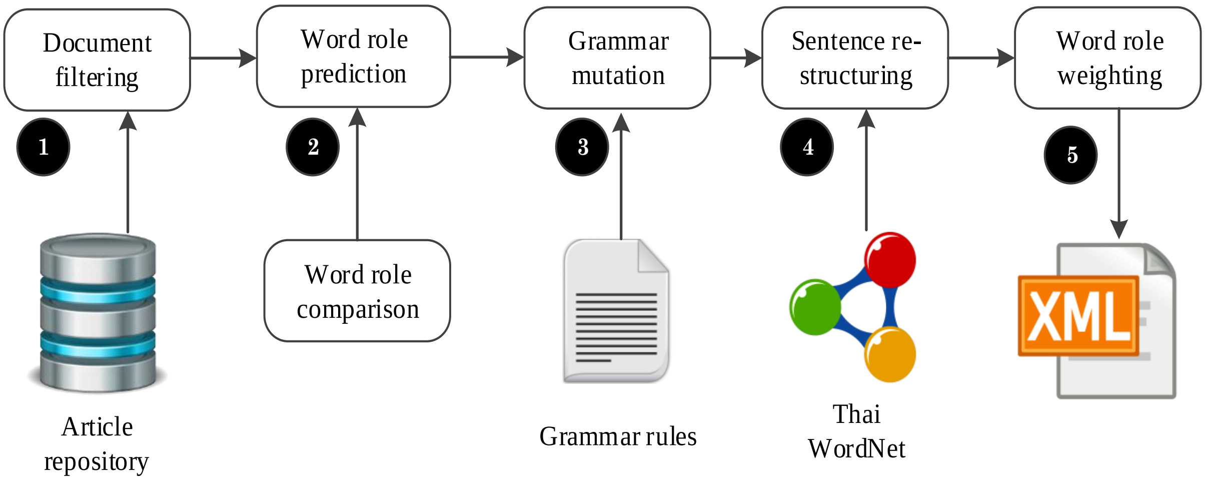


**Figure 1** SPT framework for Thai language plagiarism detection

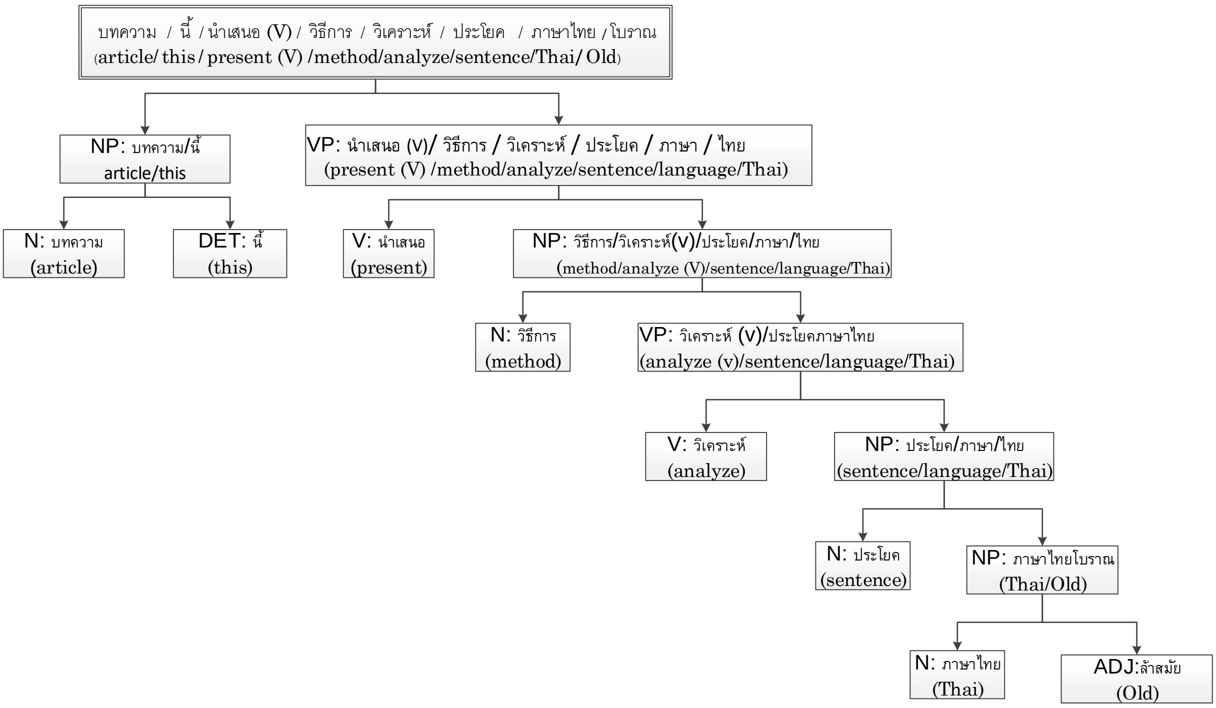

บทความ / นี้ /นำเสนอ (V)/ วิธีการ / วิเคราะห์ / ประโยค / ภาษาไทย /โบราณ
(article/ this/ present (V) /method/analyze/sentence/Thai/ Old)
NP: บทความ/นี้
article/this
VP: นำเสนอ (V)/ วิธีการ / วิเคราะห์ / ประโยค / ภาษา / ไทย
(present (V) /method/analyze/sentence/language/Thai)
N: บทความ
(article)
DET: นี้
(this)
V: นำเสนอ
(present)
NP: วิธีการ/วิเคราะห์(V)/ประโยค/ภาษา/ไทย
(method/analyze (V)/sentence/language/Thai)
N: วิธีการ
(method)
VP: วิเคราะห์ (V)/ประโยคภาษาไทย
(analyze (v)/sentence/language/Thai)
V: วิเคราะห์
(analyze)
NP: ประโยค/ภาษา/ไทย
(sentence/language/Thai)
N: ประโยค
(sentence)
NP: ภาษาไทยโบราณ
(Thai/Old)
N: ภาษาไทย
(Thai)
ADJ:ล้าสมัย
(Old)

**Figure 2** The SRL structure of a sample Thai sentence incorporating the Thai WordNet and LEXiTRON (translation: "This article presents a method to analyze an old Thai language sentence.")

### *3.2. Word role prediction*

Word segmentation is problematic and could lead to inaccuracy in similarity comparison. The SWATH API [23] was selected to perform the task of separating Thai lexical units. Its capability to perform word segmentation on ambiguous terms helps improve the accuracy of sentence restructuring. Thai language does not use punctuation marks to end sentences. The Thai writing style is written without spaces between words. When spaces are used, they generally serve as punctuation markers. Our detection system uses a decision tree with tri-grams to identify whether the space is the actual end of a sentence.

In restructuring sentences, there are two significant problems: 1) a particular word cannot be found in the database, and 2) the word role is ambiguous. For example, the word 'research' can be both noun and verb. A forecasting model was created to predict words roles for non-existent or ambiguous word based on the two previous words. The decision tree model was created based on an Orchid database [24] that contained the pre-determined word roles from research articles. The word was then transformed into a tri-gram structure. The prediction model output a table of word role possibilities in tri-gram structure. Akewonganone and Aroonmanakul [15] found that tri-gram prediction is most accurate for the Thai language due to its grammatical structure. However, tri-grams can incorrectly identify a word role in some cases. Thus, the efficiency of word role identification can be further improved using a grammar mutation technique, which is detailed as follows.

### *3.3. Grammar mutation*

To enhance the word role identification method of tri-grams, the detection system also needs to correctly identify nouns or verbs that are used in a sentence as modifiers. A Thai grammatical rule based on immediate-constituent analysis [25] was used in our system. The rule is modeled into the four following categories:

1) noun/noun = noun/adjective
2) adjective/adjective = noun/adjective
3) verb/verb = verb/adverb
4) adverb/adverb = verb/adverb

For example, the words 'ช่วยสร้าง (help build)' are both verbs (the 3rd rule). But 'สร้าง (build)' is a modifier of 'ช่วย (help)' thus its role will be identified as an 'adverb modifier' instead.

### *3.4. Sentence Re-structuring*

This step is the rearrangement of a sentence into the SRL structure. For each word, our system has to perform a search for its semantic counterparts. Once the data have been searched thoroughly, the system uses immediate-constituent analysis to categorize the sentence into two parts, a noun phrase and verb phrase. The sentence is segmented when the first verb is encountered. This step is repeated within each sub-sentence until the sub-sentence contains only one word. If the sub-sentence does not contain a verb, it will be segmented after a noun is encountered. This step is repeated until the sub-sentence contains only one atomic unit with an identified role. An example of the sentence re-structuring is illustrated in Figure 2.

A plagiarism detection technique based on SRL analyzes and compares articles using the location of each term in the sentence. Before any similarity comparison can be performed, the sentences must first be pre-processed (text segmentation, stop word removal, stemming, and lemmatization) and then re-structured. SRL can then effectively generate arguments for each sentence semantically. Quantification of similarity can be done by comparing the SRL structured sentence with those in the database. When the suspected sentence is found to have a semantically similar word to the possibly plagiarized

**Table 1** SRL categorized words with synonyms (original sentence)

| **Original sentence: บทความนี้เสนอวิธีการวิเคราะห์ประโยคภาษาไทยล้าสมัย (This article shows the method to analyze an obsolete Thai sentence)** | | |
|---|---|---|
| **Word Roles** | **Words** | **Synonyms** |
| Noun (4 words) | บทความ (article) | รายงาน (report) งานวิจัย (research) |
| | วิธีการ (method) | กระบวนการ (process) กลยุทธ์ (strategy) ขบวนการ (procedure) |
| | ประโยค (sentence) | - |
| | ภาษาไทย (Thai) | - |
| Verb (2 words) | เสนอ (show) | นำเสนอ (present) ยกตัวอย่าง (demonstrate) อธิบาย (describe) |
| | วิเคราะห์ (analyze) | พิจารณา (consider) ตรวจสอบ (check) สืบสวน(investigate) |
| Adverb (0 word) | - | - |
| Adjective (1 word) | ล้าสมัย (obsolete) | เก่า (old) เก่าแก่ (ancient) เชย (chuck) ล้าสมัย (obsolete) สมัยก่อน (yore) ดึกดำบรรพ์ (primitive) หมดสมัย (outmoded) หัวเก่า (fossil) |
| Total words | 7 | |

**Table 2** Words and their synonyms categorized according to their roles in the suspected sentence using SRL technique

| **Suspected sentence : รายงานนี้นำเสนอวิธีการวิเคราะห์สายอักขระของคำภาษาไทยโบราณ (This report presents the method to analyze ancient Thai string)** | | |
|---|---|---|
| **Word Role** | **Words** | **Synonyms** |
| Noun (5 words) | รายงาน (report) | บทความ (article) หนังสือ (book) |
| | วิธีการ (method) | กระบวนการ (process) กลยุทธ์ (strategy) ขบวนการ (procedure) |
| | อักขระ (character) | ตัวอักษร (letter) สัญลักษณ์ (symbol) |
| | คำ (word) | คำ (term) คำศัพท์ (vocabulary) |
| | ภาษาไทย (Thai) | - |
| Verb (2 words) | นำเสนอ (present) | แสดง (show) ยกตัวอย่าง (demonstrate) อธิบาย (describe) |
| | วิเคราะห์ (analyze) | พิจารณา (consider) ตรวจสอบ (check) สืบสวน(investigate) |
| Adverb (1 word) | สาย (late) | ช้า (slow) |
| Adjective (2 words) | ของ (belonging) | สรรพสิ่ง (thing) สิ่งของ (object) |
| | โบราณ (ancient) | เก่า (old) เก่าแก่ (ancient) เชย (chuck) ล้าสมัย (obsolete) สมัยก่อน (yore) ดึกดำบรรพ์ (primitive) |
| Total words | 10 | |

sentence, the number of matched arguments is counted and compared to the number of all arguments for each word type of in those two sentences.

To demonstrate how to calculate similarity, we compared the original sentence "บทความนี้เสนอวิธีการวิเคราะห์ประโยคภาษาไทยล้าสมัย (This article shows the method to analyze an obsolete Thai sentence)", against the suspected sentence "รายงานนี้นำเสนอวิธีการวิเคราะห์สายอักขระของคำภาษาไทยโบราณ (This report presents the method to analyze ancient Thai string)". LEXiTRON [26] and Thai WordNet [27] are deployed to identify function of words in a sentence. LEXiTRON is a Thai Corpus-Based Dictionary database used for Natural Language Processing that contains 103,534 words and their function information, e.g., verb, adjective, noun, or adverb.

Thai WordNet was developed by the National Electronics and Computer Technology Center (NECTEC) Thailand to create and share WordNet among Asia languages based on WordNet® Version 3.0 established in October 2007. Thai WordNet maintains a lexical database of synonyms in Thai, similar to those of WordNet. Once the sentences have been rearranged into SRL, they are restructured into a tree model using LEXiTRON [26] and Thai WordNet [27]. The result is shown in Figure 2, where NP is noun phase, VP is verb phase, V is verb, N is noun, and DET is determiners. Since Thai grammar does not employ articles or tenses, this particular Thai sentence actually reads as follows.

บทความ/ นี้/ เสนอ/ วิธีการ/ วิเคราะห์/ ประโยค/ ภาษาไทย/ ล้าสมัย
article/ this/ show/ method/ analyze/ sentence/ Thai/ obsolete

Next, the synonyms of each word are retrieved from the corpus. Data about the words and their synonyms are presented in Tables 1 and 2.

After the synonyms are extracted, each word's local weight is determined according to its role, as shown in Equation (1). The weight of each role, $Weight\ (Role_i)$, is defined by the number of words of the same type in both original and suspected sentences divided by the total number of words in both sentences. The weight role determines the significance of synonym modification.

$$Weight(Role_i) = \frac{C(Role_{i,j}) + C(Role_{i,k})}{C(Args_j) + C(Args_k)} \quad (1)$$

where $Role_i$ is the word type (noun, verb, adjective, etc.). $C(Args_x)$ represents the concepts of the argument sentence in document $x$, i.e., the number of words with the same role that appear in the particular sentence of document $x$, $j$ denotes the suspected document, and $k$ denotes the original document. The results of calculated local weights for particular word roles are shown in Table 3. Then the sentence weight is determined by creatinga set of unordered unique words and their synonyms from both sentences. The results of these sets are shown in Table 4.

**Table 3** Results of $Weight\ (Role_i)$ computation using SRL

| Word role | $\boldsymbol{Weight\ (Role_i)}$ |
|---|---|
| Noun | (5+4)/17 = 0.53 |
| Verb | (2+2)/17 = 0.24 |
| Adverb | 1/17 = 0.06 |
| Adjective | (2+1)/17 = 0.18 |

**Table 4** Results from the word union of original and suspected sentences

| Word role | Words |
|---|---|
| Noun (7 words) | {รายงาน (report) บทความ (article) วิธีการ (method) กระบวนการ (process) กลยุทธ์ (strategy) ขบวนการ (procedure) ภาษาไทย (Thai)} |
| Verb (8 words) | {นำเสนอ (present) แสดง (show) ยกตัวอย่าง (demonstrate) อธิบาย (describe) วิเคราะห์ (analyze) พิจารณา (consider) ตรวจสอบ (check) สืบสวน (investigate)} |
| Adverb (0 words) | { } |
| Adjective (4 words) | {โบราณ ล้าสมัย เก่า เชย } |
| Total words | 7+8+4 = 19 words |

After that the synonym weight for each word type, $SynWeight\ (Role_i)$, is calculated which is modified from the Jaccard coefficient [7] to compute the similarity between arguments as illustrated in Equation (2).

$$SynWeight(Role_i) = \frac{C(Args_j) \cap C(Args_k)}{C(Args_j \cup Syn_j) \cup C(Args_k \cup Syn_k)} \quad (2)$$

where $Syn_x$ represents the synonym of each word. These two example sentences can now be processed using the SRL technique in Equation (2). Table 5 shows the $SynWeight\ (Role_i)$ calculated according to word roles from this process.

**Table 5** Results of $SynWeight\ (Role_i)$ computation using SRL

| Word Role | $\boldsymbol{SynWeight\ (Role_i)}$ |
|---|---|
| Noun | 7/19 = 0.37 |
| Verb | 9/19 = 0.47 |
| Adverb | 0/19 = 0.00 |
| Adjective | 4/19 = 0.21 |

Once the $SynWeight\ (Role_i)$ is calculated, the sentence similarity can then be calculated as shown in Equation (3):

$$similarity(S_j, S_k) = \sum_{i=1}^{n} \left[ Eq.(1) \times Eq.(2) \right] \quad (3)$$

where $similarity\ (S_i, S_k)$ represents the similarity between the suspected $j$ and original $k$ sentences. $N$ is the number of all types of words contained in these two sentences. The results are depicted in Table 6.

**Table 6** Similarity computation based on SRL

| Word Role | Similarity |
|---|---|
| Noun | 0.53 x 0.37 = 0.20 |
| Verb | 0.24 x 0.47 = 0.11 |
| Adverb | 0.06 x 0.00 = 0.00 |
| Adjective | 0.18 x 0.21= 0.04 |
| Total similarity | 0.35 |

As shown in Table 6, the calculated similarity of two sentences is equal 0.35 or 35%. However, the expected similarity score from a person comparing these two sentences would be more than 50%. This is because both sentences have the same meaning but contain different words. The suspicious sentence was modified by replacing some synonyms (บทความ (article, report)) and inserting some adjectives (ล้าสมัย (obsolete, ancient)). Thus, using SRL alone cannot effectively detect such similarity. Additionally, there are some errors in the word role assignment using SRL technique. For example, the role of the word “ภาษาไทย (Thai)” in both sentences is an adjective of the words “ประโยค (sentence)” and “สายอักขระ (string)”, is not a noun as previously assumed in Table 4. As such we modified SRL technique using word role weighting as describe in the following section.

### 3.5. Word role weighting

To improve upon the SRL technique, we propose word role weighting in this study to increase the accuracy of the similarity computations. The sentences must be restructured as shown in Figure 2 to calculate each word role according to Equation (1). The comparison of each word role can be done by tracing the tree structure to find the number of similar words using the six following rules [22]:

1) If the words are identical and perform the same role, the weight is 1.
2) If the words are synonyms of each other and perform the same role, the weight is 0.9.
3) If the words are identical but perform different roles, the weight is 0.8.
4) If the words are synonyms of each other but perform different roles, the weight is 0.7.
5) If the sentence contains modifiers, the weight of modifier is 0.8 whereas the weight of the modified element (the head) is 1.2. This particular rule emphasizes the importance of the main element rather than the modifier.
6) Determiners and stop words are excluded due to their lack of importance in the sentences.

From our two previous sample sentences, the detection system could process and separate words by their roles accordingly using SPT as shown in Table 7. The results

**Table 7** Words with their roles in each sentence using SPT

**Original sentence: บทความนี้เสนอวิธีการวิเคราะห์ประโยคภาษาไทยล้าสมัย (This article shows the method to analyze an obsolete Thai sentence)**

| Word Role | Words |
|---|---|
| Noun (3 words) | บทความ (article)<br>วิธีการ (method)<br>ประโยค (sentence) |
| Verb (2 words) | เสนอ (show)<br>การวิเคราะห์ (analyze) |
| Adverb (0 word) | - |
| Adjective (2 words) | ภาษาไทย (Thai)<br>ล้าสมัย (obsolete) |
| Total | 7 words |

**Suspected sentence: รายงานนี้นำเสนอวิธีการวิเคราะห์สายอักขระของภาษาไทยโบราณ (This report presents the method to analyze ancient Thai string)**

| Word Role | Words |
|---|---|
| Noun (3 words) | รายงาน (report)<br>วิธีการ (method)<br>สายอักขระ(string) |
| Verb (2 words) | นำเสนอ (present)<br>การวิเคราะห์ (analyze) |
| Adverb (0 words) | - |
| Adjective (2 words) | ภาษาไทย (Thai)<br>โบราณ (ancient) |
| Total | 7 words |

**Table 8** Results of $Weight\ (Role_i)$ computation using SPT

| Word Role | $\boldsymbol{Weight\ (Role_i)}$ |
|---|---|
| Noun | (3+3)/14 = 0.43 |
| Verb | (2+2)/14 = 0.29 |
| Adjective | (2+2)/14 = 0.29 |

**Table 9** Sets of words remaining after SPT intersection

| Word Role | Words |
|---|---|
| Noun (3 words) | { บทความ (article), วิธีการ (method), ประโยค (sentence)} |
| Verb (2 words) | {เสนอ (show), การวิเคราะห์ (analyze)} |
| Adverb (0 words) | { } |
| Adjective (2 words) | {ภาษาไทย (Thai), ล้าสมัย (obsolete)} |

**Table 10** Words with assigned weights according word function

| Word role | Assigned weights |
|---|---|
| Noun (3 words) | { (บทความ (article), 0.9), (วิธีการ (method), 1.0), (ประโยค (sentence), 0) } |
| Verb (2 words) | {(เสนอ (show), 0.9) , (การวิเคราะห์ (analyze), 1.0)} |
| Adverb (0 words) | { } = 0 words |
| Adjective (2 words) | {(ภาษาไทย (Thai), 1.0), (ล้าสมัย (obsolete), 0.9)} |

of word separation using SPT differ slightly from those of SRL. For example, the word ภาษาไทย (Thai) is considered to be an adjective when processed with SPT. However, this word is considered as noun in SRL, as shown in Table 1 and Table 2, because SRL does not take into account that the word "ภาษาไทย" (Thai) acts as a modifier for the word "ประโยค (sentence)". Therefore, the word role of "ภาษาไทย" (Thai) assigned by SRL is incorrect. Having correctly assigned word type by SPT, $Weight\ (Role_i)$ can be computed using Eq. 1. The results from this process are shown in Table 8.

Typically, SRL exploits all synonyms extracted from the corpus. This is different from the SRL technique. SPT will intersect all synonyms from each category. As such, the number of words is reduced because duplicate words are eliminated. Table 9 shows the results of this process. It is notable that the results are different from the results shown in Table 6 which was computed using SRL.

According to six rules previously defined in Section 3.5, "ของ (of)" is excluded because it is a stop word and each word role will be assigned the defined weight in the form (*word, weight*). For example, "บทความ (article, 0.9)" indicates that the word "article" will be weighted more, 0.9, because it is a synonym of "รายงาน" (report) and they are both nouns. Likewise, "วิธีการ (method, 1.0)" in both sentences are identical in both spelling and function. Therefore, its weight is 1, whereas there is no synonym for the word "ประโยค" (sentence, 0) and, thus, its weight is 0. Table 10 shows the weights of the remaining words.

From Equation (2), we found a limitation of SRL when the suspected sentence contains additional modification words. Therefore, we designed a new dynamic weight function to adjust for the importance of each role. Without role weighing, plagiarism by adding modifiers will not be efficiently detected. As a result, we modified the $SynWeight\ (Role_i)$ computation, which is defined as:

$$SynWeight_{new}(Role_i) = \frac{\sum_{i=1}^{n} w_i}{N_r} \tag{4}$$

where *w* is the weight of word in the same category and *N* is number of words with the same function. The results from computations using Equation (4) are shown in Table 11.

After the similarity of each role is calculated, the sentence similarity can then be calculated using Equation (3). From the above example, the total sentence similarity is equal to 0.83 or 83% as shown in Table 12.

**Table 11** Results of $SynWeight_{new}\ (Role_i)$ computation using SPT

| Word Role | $\boldsymbol{SynWeight_{new}\ (Role_i)}$ |
|---|---|
| Noun | (0.9+1.0+0.0)/3 = 0.63 |
| Verb | (0.9+1.0)/2 = 0.95 |
| Adverb | 0/0 = 0.00 |
| Adjective | (1.0+0.9)/2 = 0.95 |

**Table 12** Similarity computation based on modified SPT

| Word Role | Similarity |
|---|---|
| Noun | 0.43 x 0.63 = 0.27 |
| Verb | 0.29 x 0.95 = 0.28 |
| Adverb | 0.00 x 0.00 = 0.00 |
| Adjective | 0.29 x 0.95 = 0.28 |
| Total similarity | 0.83 |

Based on the above example, the similarity value increased using the developed method (SPT). This is because, typically, higher similarity values represent more semantic similarity detected between two sentences. If these two sentences were not weighted according to word roles, it would have similarity of 0.35, which is less than using SPT weight role function. Therefore, our proposed method can alleviate the similarity comparison problems when 1) the two sentences contain different numbers of words, or 2) the suspected sentence contains modifiers or synonyms.

The next section illustrates how the proposed SPT technique solves the problems of SRL and is more efficient compared to other techniques.

## 4. Performance evaluation

To verify the effectiveness of plagiarism tools, we compared the SPT plagiarism detection technique against other state-of-the art techniques. The precision and recall values were computed and compared using the following tools:

1) Tri-grams - a typical statistical analysis for natural language processing,
2) SRL - a semantic comparison with traditional semantic role labeling,
3) Turnitin - a commercially-available service used in plagiarism detection, and
4) Akarawisut - a free web-based service especially designed for Thai language plagiarism detection.

The corpus in this study contained 56,380 sentences from scientific research articles in ten areas of computer science and engineering (computer network, database, information technology, information retrieval, system analysis and design, data warehouse, decision support systems, multimedia, management information systems, and cloud computing). Four types of plagiarism data sets, each with twenty five different cases, were created specifically for these experiments. These four types of plagiarism are word-by-word, word-reordering, modifier-insertion, and synonym-replacement plagiarism.

### *4.1. State-of-the art algorithms comparison*

For each experiment, we set the similarity score threshold to less than 20%. This either means (1) 20% of text is in similar within one document, or (2) 1% of text is similar among 20 different documents. Therefore, the system will flag any document that plagiarizes those documents in the repository at a level over the threshold. Precision and recall, as defined in Equation (5) and (6), are the indicators of detection accuracy.

$$\text{Precision} = \frac{\eta}{\text{N}} \tag{5}$$

$$\text{Recall} = \frac{\eta}{\text{P}} \tag{6}$$

where $\eta$ is the number of correctly-detected plagiarized sentences and $N$ refers to the total number of detected sentences that each technique flagged with a similarity score greater than 20%. $P$ is the number of *actual* plagiarized sentences in the dataset. For example, $N$= 25 indicates that 25 of 56,380 sentences were determined to be plagiarized. $P$=25 means there are actual 25 plagiarized sentences in the dataset. Finally, $\eta$ =25 refers to the 25 sentences that the system correctly detected as plagiarized sentences.

**Table 13** Results of word-by-word plagiarism detection

| Algorithms | *N* | *P* | $\eta$ | Average Precision | Average Recall |
|---|---|---|---|---|---|
| Tri-grams | 25 | 25 | 25 | 1.0 | 1.0 |
| SRL | 75 | 25 | 25 | 0.33 | 1.0 |
| SPT | 35 | 25 | 25 | 0.79 | 1.0 |

### *4.1.1. Word-by-Word plagiarism detection*

Pattern-matching techniques, such tri-grams, are considered superior for detecting word-by-word plagiarism. Unlike SRL and SPT, the pattern-matching algorithm is not affected by two of SRL detection problems: 1) absence of particular words in the database, and 2) re-structuring of the sentences. The experimental results in Table 13 show the average precision and recall obtained from three algorithms. The average recall values of SRL and SPT are the same as those of tri-grams. All detection methods yielded average recall values of 1.0. These results are attributed to the accurate performance of the SWATH API [23] in lexical unit segmentation which facilitates correct sentence re-structuring. The tri-gram method has the highest average precision, 1.0, whereas SPT obtained an average precision of 0.79 and SRL had the lowest, just 0.33. This is because the two SRL techniques identified some sentences as false positives due to their semantic similarity. This results in lower average precision of those SRL techniques. Moreover, the lexicon-based comparison also suffers when a word in the text does not exist in the corpus.

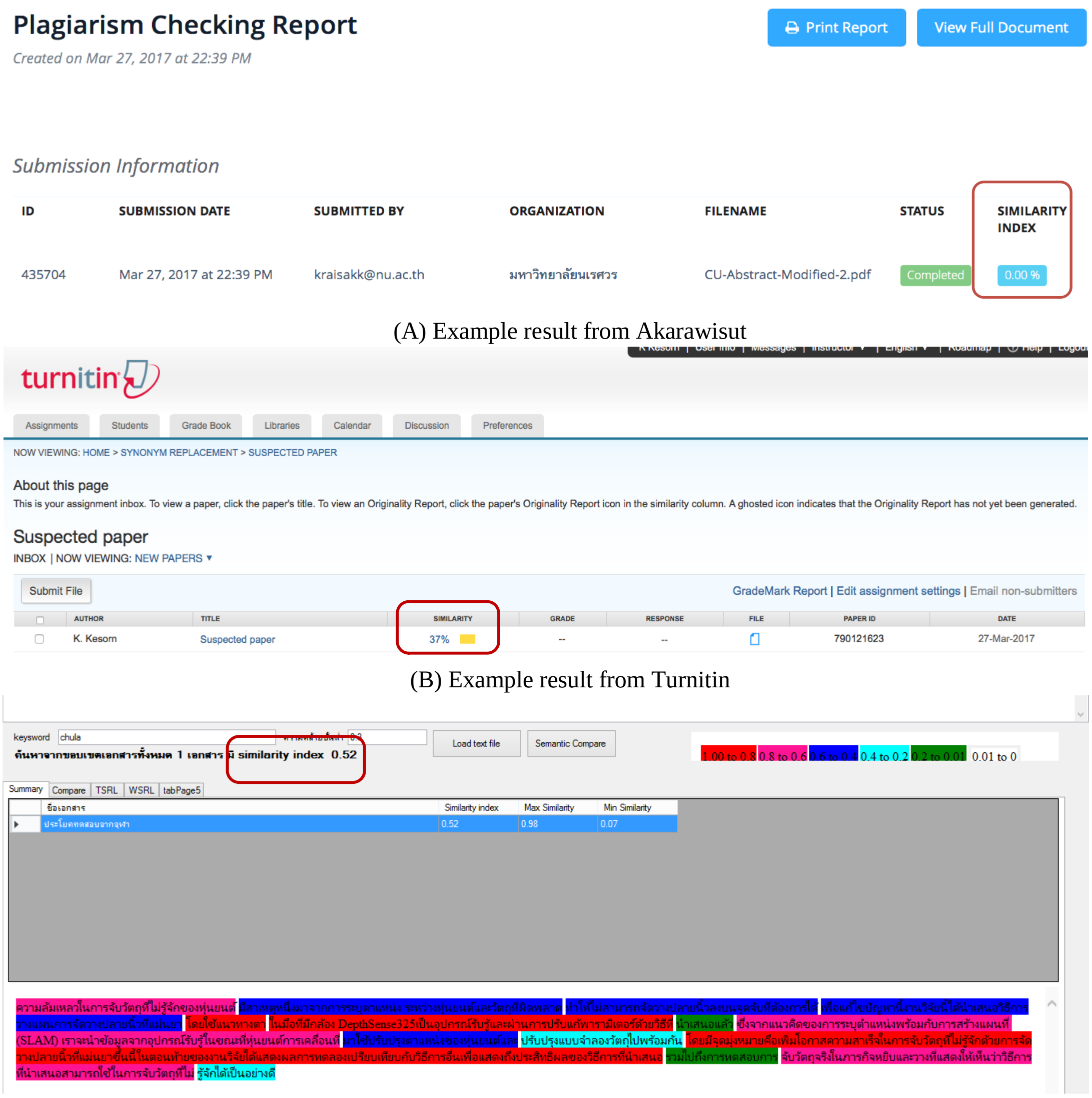
### *4.1.2. Word-Reordering plagiarism*

Our hypothesis before evaluating the performance of each tool against reordered word sequences was that the tri-gram technique would exhibit the poorest performance. This is due to changes in fingerprints of the text. Tri-grams cannot detect the similarity of sentences when the fingerprints are different. Thus Tri-gram accuracy was expected to be poorer.

The experimental results in Table 14 show that the tri-gram technique could not detect any plagiarized sentences. SRL performed better than tri-grams but worse than SPT techniques, obtaining only a precision of 0.28 because word reordering may grammatically change word roles. SRL is unable to detect plagiarism using this method. This is in contrast to the case where the SPT algorithm could compare words with different roles. SPT effectively assigned words role and therefore obtained higher precision (0.96) and recall (0.96).

**Table 14** Results of word-reordering plagiarism detection

| Algorithms | *N* | *P* | $\eta$ | Average Precision | Average Recall |
|---|---|---|---|---|---|
| Tri-grams | 0 | 25 | 0 | 0 | 0 |
| SRL | 42 | 25 | 12 | 0.29 | 0.48 |
| SPT | 25 | 25 | 24 | 0.96 | 0.96 |



Plagiarism Checking Report

Created on Mar 27, 2017 at 22:39 PM

Submission Information

| ID | SUBMISSION DATE | SUBMITTED BY | ORGANIZATION | FILENAME | STATUS | SIMILARITY INDEX |
|---|---|---|---|---|---|---|
| 435704 | Mar 27, 2017 at 22:39 PM | kraisakk@nu.ac.th | มหาวิทยาลัยนเรศวร | CU-Abstract-Modified-2.pdf | Completed | 0.00 % |

(A) Example result from Akarawisut

turnitin
K Kesorn | User Info | Messages | Instructor ▾ | English ▾ | Roadmap | Help | Logout
Assignments
Students
Grade Book
Libraries
Calendar
Discussion
Preferences

NOW VIEWING: HOME > SYNONYM REPLACEMENT > SUSPECTED PAPER

About this page

This is your assignment inbox. To view a paper, click the paper's title. To view an Originality Report, click the paper's Originality Report icon in the similarity column. A ghosted icon indicates that the Originality Report has not yet been generated.

Suspected paper

INBOX | NOW VIEWING: NEW PAPERS ▾

Submit File
GradeMark Report | Edit assignment settings | Email non-submitters

| □ | AUTHOR | TITLE | SIMILARITY | GRADE | RESPONSE | FILE | PAPER ID | DATE |
|---|---|---|---|---|---|---|---|---|
| □ | K. Kesorn | Suspected paper | 37% | -- | -- | 🗋 | 790121623 | 27-Mar-2017 |

(B) Example result from Turnitin

keysword: chula

ค้นหาจากขอบเขตเอกสารทั้งหมด 1 เอกสาร มี similarity index 0.52

| | ชื่อเอกสาร | Similarity index | Max Similarity | Min Similarity |
|---|---|---|---|---|
| ▸ | ประโยคทดสอบจากจุฬา | 0.52 | 0.98 | 0.07 |

ความล้มเหลวในการจับวัตถุที่ไม่รู้จักของหุ่นยนต์ มีสาเหตุหนึ่งมาจากการระบุตาแหน่ง ระหว่างหุ่นยนต์และวัตถุที่ผิดพลาด ทำให้ไม่สามารถจัดวางปลายนิ้วลงบนจุดจับที่ต้องการได้ เพื่อแก้ไขปัญหานี้งานวิจัยนี้ได้นำเสนอวิธีการวางแผนการจัดวางปลายนิ้วที่แม่นยา โดยใช้แนวทางตา ในมือที่มีกล้อง DepthSense325เป็นอุปกรณ์รับรู้และผ่านการปรับแก้พารามิเตอร์ด้วยวิธีที่ นำเสนอแล้ว ซึ่งจากแนวคิดของการระบุตำแหน่งพร้อมกับการสร้างแผนที่ (SLAM) เราจะนำข้อมูลจากอุปกรณ์รับรู้ในขณะที่หุ่นยนต์การเคลื่อนที่ มาใช้ปรับปรุงตาแหน่งของหุ่นยนต์และ ปรับปรุงแบบจำลองวัตถุไปพร้อมกัน โดยมีจุดมุ่งหมายคือเพิ่มโอกาสความสาเร็จในการจับวัตถุที่ไม่รู้จักด้วยการจัดวางปลายนิ้วที่แม่นยาขึ้นนี้ในตอนท้ายของงานวิจัยได้แสดงผลการทดลองเปรียบเทียบกับวิธีการอื่นเพื่อแสดงถึงประสิทธิผลของวิธีการที่นำเสนอ รวมไปถึงการทดสอบการ จับวัตถุจริงในภารกิจหยิบและวางที่แสดงให้เห็นว่าวิธีการที่นำเสนอสามารถใช้ในการจับวัตถุที่ไม่ รู้จักได้เป็นอย่างดี

(C) Sample result from SPT

**Figure 3** Plagiarism detection result comparison between SPT and commercial tools (Akarawisut and Turnitin)

**Table 15** Results of modifier-insertion plagiarism detection

| Algorithms | *N* | *P* | *η* | Average Precision | Average Recall |
|---|---|---|---|---|---|
| Tri-grams | 0 | 25 | 0 | 0 | 0 |
| SRL | 24 | 25 | 16 | 0.67 | 0.64 |
| SPT | 30 | 25 | 25 | 0.83 | 1.0 |

**Table 16** Results of synonym-replacement plagiarism detection

| Algorithms | *N* | *P* | *η* | Average Precision | Average Recall |
|---|---|---|---|---|---|
| Tri-grams | 0 | 25 | 0 | 0 | 0 |
| SRL | 30 | 25 | 21 | 0.70 | 0.84 |
| SPT | 32 | 25 | 25 | 0.78 | 1.0 |

### *4.1.3. Modifier-Insertion plagiarism*

In the particular case of modifier-insertion plagiarism, our hypothesis was that the structure-based method would yield a lower similarity due to the increased magnitude of the denominator (a higher number of words to taken into account). The experimental results in Table 15 show that the tri-gram technique could not detect any plagiarized sentences because the document fingerprints were not similar. The tri-gram technique performs poorer than the SRL technique which obtained 0.67 and 0.64 average precision and recall respectively. However, SRL had a poorer performance than SPT because it incorrectly identified word role and did poor sentence segmentation. Thus, the total similarity was less than expected. SPT had the best performance as it weighted the word according to its role (noun, verb, adjective, and adverb). This allows the developed system to effectively detect similarities between original and suspicious documents, which made SPT superior to other methods (0.83 and 1.0 for average precision and recall, respectively).

### *4.1.4. Synonym-Replacement plagiarism*

In this experiment, some words in the sentences were replaced by their synonyms. The performance of the tri-gram technique suffered the most. The experimental results in Table 16 show that tri-grams, a non-semantic-based technique, could not detect any plagiarized sentences. This, again, was because this technique relies on document fingerprints. When the words are replaced by their synonyms, causing the fingerprints to change, the tool was unable to detect sentences with the same meaning and, consequently, the similarity between the existing sources and suspected articles was very low.

### *4.2. Commercial tool comparison*

Additional empirical validation is studied using synonym replacement. The plagiarized document was altered from the original using modifier-insertion and word reordering plagiarism with various the degrees of synonym replacements between 10% and 40%. These test settings were used to analyze ten categories of documents. Table 17 reports the similarity index computed using three detection tools, SPT, Turnitin, and Akarawisut. As shown in Table 17, SPT out-performed other techniques in all cases examined, obtained similarity indices of up to 52% (in the 40% alteration case) whereas Akarawisut and Turnitin obtain 0% and 37% respectively. This is because the SPT algorithm used a semantic-based technique for plagiarism detection, assigning weights to different word roles, allowing the system to effectively compute the similarity of those sentences. As a result, SPT obtained higher detection efficiency even when some words were replaced by their synonyms, modifiers were inserted, and word order were changed. Turnitin and Akarawisut obtained poorer performance than SPT in these plagiarisms test cases due to reliance on document fingerprinting. When words were changed, the fingerprints of documents were changed. Hence, these tools could not effectively detect plagiarized sentences. Figures 3 (A), (B), and (C) illustrate the detection results of these three tools when test documents were plagiarized by approximately 40%.

**Table 17** Similarity Index for four different datasets using various plagiarism detection techniques

| **Algorithms** | **Percentage of altered plagiarism** | | | |
|---|---|---|---|---|
| | **10%** | **20%** | **30%** | **40%** |
| Akarawisut | 0.42 | 0.36 | 0.14 | 0 |
| Turnitin | 0.85 | 0.64 | 0.51 | 0.37 |
| SPT | ***0.92*** | ***0.84*** | ***0.63*** | ***0.52*** |

## 5. Conclusions

A modified SPT framework was developed in this study to solve the role-labeling and role-weighing problems in SRL. SRL and other techniques do not effectively detect plagiarized sentences for the Thai language due to the complexity of Thai sentences, i.e., the lack of spaces between words, no full stops, and word ambiguity (synonyms and polysemy). The developed method differs from existing ones in that it analyzes a sentence based on several methods such as word role prediction, grammar mutation, sentence re-structuring, and word role weighting. These processes can significantly improve the similarity computation over that of traditional methods, e.g., tri-grams and SRL. Additionally, this method can effectively detect plagiarized documents without relying on character or document fingerprints leading to more effective detection of four types of plagiarism, word-by-word copying, word-reordering, modifier-insertion, and synonym-replacement plagiarism, compared to commercial tools such as Turnitin and Akarawisut.

However, some limitations still exist in SPT. The performance of the presented method heavily depends on the accuracy of the techniques employed and the lexical corpus. These methods lack of ability to learn *unknown* words that are not contained in the corpus. If unknown words are found, SPT cannot correctly compute a similarity between words and consequently, its power of plagiarism detection is decreased.

One possible direction for our future work to deal with the unknown data using an ontology model. Ontology has been shown to effectively handle unknown data. To lessen the dependency upon the information in the corpus, SRL, and SPT techniques, we could replace SRL and SPT techniques with an ontology model to effectively reason and detect plagiarized information.

## 6. Acknowledgements

This research was supported in part by Naresuan University, Thailand. Grant no. R2559C229. The authors would like to thank Noah Hafner (Ph.D.) for English proof reading.

## 7. Author contributions

S. Prapanitisatian and K. Kesorn collected data for the experiments. W. Massagram and K. Kesorn conceived, designed the experiments, and wrote the paper. K. Kesorn was involved in the discussions and analysis plans for the paper from its inception, including the idea for the data analysis. The authors declare that no competing interests exist. The funder had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

## 8. References

[1] Young D. Perspectives on cheating at a thai university. Lang Test Asia. 2013;3(6):1-15.

[2] Nidapoll. Survey of cheating of students in universities of Thailand [Internet]. Thailand: Dailynews 2013 [cited 2015 Oct 8]. Available from: http://www.dailynews.co.th/education/196761.

[3] Bretag T. Challenges in addressing plagiarism in education. PLoS Med 2013;10:e1001574. doi:10.1371/journal.pmed.1001574.

[4] Roig M. Avoiding plagiarism, self-plagiarism, and other questionable writing practices: a guide to ethical writing. New York: St. Johns University Press; 2006.

[5] Chulalongkorn University [Internet]. Thailand: Akarawisut 2015 [cited 2015 Oct 8]. Available from: http://akarawisut.com/.

[6] Turnitin [Internet]. Defining Plagiarism: The Plagiarism Spectrum 2015 [cited 2015 Oct 8]. Available from: http://go.turnitin.com/paper/plagiarism-spectrum.

[7] Osman AH, Salim N, Binwahlan MS, Alteeb R, Abuobieda A. An improved plagiarism detection scheme based on semantic role labeling. J. Appl. Soft Comput. 2012;12(5):1493-502.

[8] Ali AMET, Abdulla HMD, Snasel V. Overview and comparison of plagiarism detection tools. In: V. Snášel, J. Pokorný, K. Richta, editors. Proceedings of the Dateso 2011: Annual International Workshop on DAtabases, TExts, Specifications and Objects; 2011 Arpr 20; Pisek, Czech Republic; 2011. p. 161-72.

[9] Donaldson J, Lancaster A-M, Sposato P. A plagiarism detection system. Proceedings of the twelfth SIGCSE technical symposium on Computer science education; 1981 Feb 26-27; St. Louis, Missouri, New York: ACM; 1981. p. 21-5.

[10] Lukashenko R, Graudina V, Grundspenkis J. Computer-based plagiarism detection methods and tools: an overview. Proceedings of the 2007 international conference on Computer systems and technologies; 2007 Jun 14-15; Bulgaria. New York: ACM; 2007. p. 40:1-6.

[11] Alzahrani SM, Salim N, Abraham A. Understanding plagiarism linguistic patterns, textual features, and detection methods. IEEE Trans. Syst. Man. Cybern. Part C Appl. Rev. 2012;42(2):133-49.

[12] Schleimer S, Wilkerson DS, Aiken A. Winnowing: local algorithms for document fingerprinting. Proceedings of the 2003 ACM SIGMOD international conference on Management of data; 2003 Jun 9-12; San Diego, California: ACM; 2003. p. 76-85.

[13] Wibowo AT, Sudarmadi KW, Barmawi AM. Comparison between fingerprint and winnowing algorithm to detect plagiarism fraud on Bahasa Indonesia documents. 2013 International Conference of Information and Communication Technology; 2013 Mar 20-22; Bandung, Indonesia: IEEE; 2013. p. 128-33.

[14] Gipp B, Meuschke N. Citation pattern matching algorithms for citation-based plagiarism detection: greedy citation tiling, citation chunking and longest common citation sequence. Proceedings of the 11th ACM symposium on Document engineering; 2011 Sep 19-22; California: ACM; 2011. p. 249-58.

[15] Akewonganone A, Aroonmanakul W. Identification of Thai and transliterated words by N-Gram models. Thailand: Chulalongkorn University; 2005.

[16] White DR, Joy MS. Sentence-based natural language plagiarism detection. J. Educ. Resour. Comput. 2004;4(4):1-20.

[17] Jadalla A, Elnagar A. PDE4Java: plagiarism detection engine for java source code: a clustering approach. Int. J. Bus. Intell. Data Min. 2008;3(2):121-35.

[18] Kustanto C, Liem I. Automatic source code plagiarism detection. 2009 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing; 2009 May 27-29; Daegu, South Korea: IEEE; 2009. p. 481-6.

[19] Lesner B, Brixtel R, Bazin C, Bagan G. A novel framework to detect source code plagiarism: now, students have to work for real!. Proceedings of the 2010 ACM Symposium on Applied Computing (SAC) ; 2010 Mar 22-26; Sierre, Switzerland. New York: ACM; 2010. p. 57-8.

[20] Mariani L, Micucci D. AuDeNTES: automatic detection of teNtative plagiarism according to a rEference Solution. ACM Trans. Comput. Educ. 2012;12(1):2:1-26.

[21] Liu H, Wang P. Assessing text semantic similarity using ontology. J. Softw. 2014;9(2):490-7.

[22] Prapanitisatian S, Kesorn K. Semantic-based technique for Thai documents plagiarism detection. KKU Eng. J. 2014;41(1):109-17.

[23] Charoenpornsawat P. Feature-based Thai Word Segmentation [Thesis]. Thailand: Chulalongkorn University; 1999.

[24] Isahara H, Sornlertlamvanich V, Takahashi N. ORCHID: building linguistic resources in Thai. Lit. Linguist Comput. 2000;15(4):465-78.

[25] Pankhuenkhat R. Thai sentence analysis. J. Thai Lang. Cult. 2007;1:42-57.

[26] Mekpiroon O, Tammarattananont P, Apitiwongmanit N, Buasroung N, Pravalpruk B, Supnithi T. Dictionary-based translation feature in open source LMS a case study of Thai LMS: LearnSquare. In: Díaz P, Kinshuk, Aedo I, Mora E, editors. 2008 Eighth IEEE International Conference on Advanced Learning Technologies; 2008 Jul 1-5; Santander, Spain. USA: IEEE; 2008. p. 369-70.

[27] Thoongsup S, Robkop K, Mokarat C, Sinthurahat T, Charoenporn T, Sornlertlamvanich V, et al. Thai WordNet construction. Proceedings of the 7th Workshop on Asian Language Resources; 2009 Aug 6-7; Suntec, Singapore. USA: ACM; 2009. p. 139-44.