Expert Systems With Applications 42 (2015) 8936–8946

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

PDLK: Plagiarism detection using linguistic knowledge

Asad Abdi [a,*], Norisma Idris [a], Rasim M. Alguliyev [b], Ramiz M. Aliguliyev [b]

[a] Department of Artificial Intelligence Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia
[b] Institute of Information Technology, Azerbaijan National Academy of Sciences, 9, B. Vahabzade Street, AZ1141 Baku, Azerbaijan

ARTICLE INFO

Keywords:
Automatic plagiarism detection
Text reuse
String matching
Semantic analysis

ABSTRACT

Plagiarism is described as the reuse of someone else's previous ideas, work or even words without sufficient attribution to the source. This paper presents a method to detect external plagiarism using the integration of semantic relations between words and their syntactic composition. The problem with the available methods is that they fail to capture the meaning in comparison between a source document sentence and a suspicious document sentence, when two sentences have same surface text (the words are the same) or they are a paraphrase of each other. Therefore it causes inaccurate or unnecessary matching results. However, this method can improve the performance of plagiarism detection because it is able to avoid selecting the source text sentence whose similarity with suspicious text sentence is high but its meaning is different. It is executed by computing the semantic and syntactic similarity of the sentence-to-sentence. Besides, the proposed method expands the words in sentences to tackle the problem of information limit. It bridges the lexical gaps for semantically similar contexts that are expressed in a different wording. This method is also capable to identify various kinds of plagiarism such as the exact copied text, paraphrasing, transformation of sentences and changing of word structure in the sentences. As a result, the experimental results have displayed that the proposed method is able to improve the performance compared with the participating systems in PAN-PC-11. The experimental results also displayed that the proposed method demonstrates better performance as compared to other existing techniques on PAN-PC-10 and PAN-PC-11 datasets.

1. Introduction

With the increasing information in World Wide Web, it makes easy to represent someone else's thought as own work without providing the appropriate credit for the first owner or original source. Avoiding plagiarism is essential with regard to ethics. Recently, it made a significant issue in both academic and non-academic worlds. In plagiarism, the plagiarists attempt to change the contribution, the idea or the words of others as their own work (Geravand & Ahmadi, 2014; Osman, Salim, Binwahlan, Alteeb, & Abuobieda, 2012). The plagiarism can be performed by exact copy, cutting sentences, combining sentences, paraphrasing, replacing the original words with the similar words or synonym words (El-Alfy, Abdel-Aal, Al-Khatib, & Alvi, 2015; Sánchez-Vega, Villatoro-Tello, Montes-y-Gomez, Villaseñor-Pineda, & Rosso, 2013). The challenge involving plagiarism can be obtained from several areas and therefore has effects on us in several ways. Most of these areas contain: academia, scientific research, journalism, patents and literature (Oberreuter & VeláSquez, 2013).

Anti-plagiarism tool can have key role to prevent people performing plagiarism inadvertently or intentionally, so that the people provide much attempt to contribute new thoughts or even methods based on their investigation to the academic world. Plagiarism identification can be done in two main ways: manually and automatically. While the automatic plagiarism recognition is performed by the computer system, the manual plagiarism recognition is carried out by human. The plagiarism identification is also divided into the external plagiarism identification and internal plagiarism identification. In internal plagiarism identification method an unknown document is compared with a set of known documents by the same author, in order to determine whether the unknown document has been written by the same author published the known documents (Mahdavi, Siadati, & Yaghmaee, 2014; Oberreuter & VeláSquez, 2013). In external plagiarism identification a suspicious document is compared with a set of source documents to find plagiarized text between them (Rao, Gupta, Singhal, & Majumder, 2011; Wang, Qi, Kong, & Nu, 2013).

Nowadays there are several methods to identify the plagiarized content. Usually, these methods compare two documents on word, phrase or sentence level (Sarkar, Marjit, & Biswas, 2014) . This paper also aims to propose a method that considers both the semantic and

* Corresponding author. Tel.: +60 104232342.
E-mail addresses: asadabdi55@gmail.com (A. Abdi), norisma@um.edu.my (N. Idris), r.alguliev@gmail.com (R.M. Alguliyev), r.aliguliyev@gmail.com (R.M. Aliguliyev).

http://dx.doi.org/10.1016/j.eswa.2015.07.048
0957-4174/

syntactic information in comparison between a source document and a suspicious document.

In text relevance context, linguistic information for instance semantic relations between words and their syntactic structure, have key role in sentence comprehension. Syntactic information, like word-order, can prepare beneficial information to distinguish the meaning of two sentences, when two sentences share the similar bag-of-words. For example, "*Alex calls John*" and "*John calls Alex*" will be judged as similar sentences because they have the same surface text. However, their meaning is different. Therefore, to compare two documents, the source document and a suspicious document, the proposed method should contribute syntactic information to determine suspicious similarity between two documents; otherwise, it fails to capture the meaning in comparison and often there is a conflict to identify suspicious similarity between documents. However, it leads to incorrect or even unnecessary matching results.

On other hand, in comparison between two sentences, two sentences are considered to be similar if most of the words are the same or if they are a paraphrase of each other. However, it is not always the case that sentences with similar meaning necessarily share many similar words. Hence, semantic information such as semantic similarity between words and synonym words can provide useful information when two sentences have similar meaning, but they used different words in the sentences. This is because people can express the same meaning using various sentences in terms of word content. However, the more similar sentence may be represented with similar words, rather than the original words expressed in the source document sentences; hence the semantic information will help to identify the similar ideas, when an author presents someone else's idea as his or her own words by text manipulation approach, paraphrase or synonym words.

The proposed method is used to detect the plagiarized text. The method includes three important points. First, it is a comprehensive plagiarism method, which can detect different types of plagiarism such as the exact copied text, paraphrasing (similar or synonym words replacing), transformation of sentences and changing of word structure in the sentences. The second point is related to the comparison mechanism, where our method considers both the semantic and syntactic information to compute the similarity measure between two sentences. The third point indicates that the method can capture the meaning of sentences using the combination of semantic and syntactic information.

The method is called Plagiarism Detection using Linguistic Knowledge (PDLK), since the suspicious documents are identified using semantic information obtained from a lexical database and syntactic information is given by analysing the structure of the sentence. The structure of this paper is as follows. Section 2 provides a short overview of the previous methods that are used to produce summaries. Section 3 introduces the proposed method. Section 4 discusses the performance analysis and presents the results of the analysis. Finally, in Section 5, we summarize the works discussed and the progress of the project.

2. Brief review of literature

The plagiarism detection process contains three main steps (Barrón-Cedeño, Gupta, & Rosso, 2013; Potthast et al., 2012): candidate retrieval, detailed comparison, and heuristic post-processing. Given a suspicious document (denoted by $doc_{suspicious}$) and a large corpus of documents (denoted by $cor_{document}$). First, the candidate retrieval selects a set of document from the corpus (denoted by doc_{source}; $doc_{source} \subseteq cor_{document}$) that are more similar to the $doc_{suspicious}$ and can be a source of plagiarized content. Second, each source document, doc_{source}, is compared with the suspicious document, $doc_{suspicious}$, then each pair of sections of both document are extracted and are considered as a plagiarism, if they have high similarity. Third, heuristic post-processing presents all extracted sections pairs. Below a set of methods that have been proposed to detect plagiarism are introduced.

Paul and Jamal (2015) proposed a method to for plagiarism detection. The method comprises five main steps: pre-processing; candidate retrieval; sentence ranking; semantic role labeling and similarity detection. Pre-processing includes two sub-steps that are text segmentation and stop word removal. The text segmentation splits the text into several sentences. The stop word removal, eliminate some of the English words that are most frequently used. Candidate retrieval determines a subset of source documents for a suspicious document. It uses *n*-gram and Jaccard coefficient similarity for text comparison. Sentence ranking is employed to rank sentences in the suspicious and original document to retrieve original and suspicious sentence pairs. The similarity between sentences is calculated using the cosine similarity. Semantic Role Labeling (SRL) aims to determine the semantic roles of each term of a sentence based on the semantic relationship between their terms. It determines the object and subject of a sentence for identifying the semantic roles of each term. Plagiarism detection using SRL aims to detect the semantic similarity between the ranked sentences. Similarity detection, in this stage, sentence similarity between ranked suspected and original sentences is performed. Sentences in suspected documents are compared with each sentence in the candidate. The experimental displayed that the application of sentence ranking in current method decreases the time of checking.

Oktoveri, Wibowo, and Barmawi (2014) proposed a method to reduce non-relevant documents, which have no similar topic with query document. The proposed method used several algorithm and approach: winnowing algorithm; AVL Tree for indexing documents; Longest Common Subsequence and term frequency. AVL Tree algorithm is a data structure algorithm based on Binary Tree (AdelsonVelskii & Landis, 1963; Foster, 1965; Irving & Love, 2003). It is used a data structure including of terms from each document. Each indexed documents is omitted based on the similarity measure calculated using asymmetric similarity equation (Raphael, 2002) based on term frequency. Term frequency is the number of term occurrences in a document. It is used to compute the similarity measure between two documents (Raphael, 2002; Stein & Zu Eissen, 2006). Winnowing algorithm (Schleimer, Wilkerson, & Aiken, 2003) is employed reduce the number of terms in order to accelerate the detection process. LCS is used to compare two strings and to find out the longest overlapping path (Campos & Martinez, 2012; Iliopoulos & Rahman, 2009). The result shows that reducing non-relevant document shortens the processing time compared to non-reduced process. It also provides good result in terms of speed and accuracy.

Mahdavi et al. (2014)) proposed an external plagiarism detection method based on the vector space model (VSM) (Raghavan & Wong, 1986). The proposed method contains three main phases: data preparation, relevant documents retrieval and detailed string matching. The main task of data preparation is to convert both source and suspicious documents into vectors of the corpus terms. The terms are high frequent ones of the corpus. The data preparation phase includes six sub-steps: Text normalisation, stop words removal, stemming, synonyms replacement, tokenisation and feature selection. In relevant document retrieval phase a suspicious document is compared to all source documents using vector cosine similarity in order to retrieve the most relevant source documents. In detailed string matching phase the suspicious document and all retrieved documents from previous phase are converted into tri-grams. Then, using the overlap coefficient, the most similar documents to the suspicious document are determined as plagiarized source documents. The experimental result demonstrated that the accuracy of the method was encouraging.

Soleman and Purwarianti (2014) proposed a method for plagiarism detection based on the Latent Semantic Analysis (LSA) (Franzke & Streeter, 2006). The method includes three main components:

pre-processing (PRE) component, heuristic retrieval (HR) component and detailed analysis (DA) component. LSA is used in both heuristic retrieval and detailed analysis components. The pre-processing is performed for both source and suspicious documents. This component contains three processes such as stop word removal, stemming and tokenisation. The aim of heuristic retrieval component to reduce the number of documents that must be analysed detailed analysis component. It used LSA method to find the most relevant source documents. Detailed analysis component identifies most suspected plagiarism section in the source document. To determine this, the source document and suspected document are split into section such as paragraph or sentences then each section in both documents is compared. In this component, LSA is also used as the document comparison method.

Hussein (2015) proposed a method for document Similarity Estimation. It includes pre-processing, Phrase Extraction, Building Document Model and Similarity Estimation. In pre-processing step the main linguistic functions such as tokenisation and stop word removal is performed. In the next step, Phrase Extraction, all documents are split into n-gram, e.g. unigram, bigram and trigram. Building Document Model represents documents using a matrix, where columns represent documents and rows represent phrases. Each element of matrix represents the weighted occurrence frequency of phrase extracted from previous step. The last stage, Similarity Estimation, computes the mutual pairwise document similarity. For this purpose, the Singular Value Decomposition (SVD) is applied to decompose the matrix A into three independent matrices (Ceska, Toman, & Jezek, 2008). Finally, if a query vector q, represents a suspicious document the similarity measure between the query vector q and the m document vectors is computed using the cosine similarity. The experimental results displayed that the method could generate better results than other methods.

Wang et al. (2013) proposed a method based on the VSM (Bao, Shen, Liu, & Song, 2003) and Jaccard coefficient (Kong et al.) to detect the plagiarism. VSM is based on TF-IDF scheme. In the VSM the source document and the suspicious document are divided into sentences, and then the sentences are represented in term of vector. The weight of each term in vector is calculated by the TF-IDF method. Finally the similarity between two sentences is computed using the cosine distance. The VSM focuses on computing the global similarity measure. It is good to detect plagiarism when the passages or sentences include plagiarized phrases. However, the similarity measure based on the VSM is not able to identify all kinds of plagiarized content; hence the Jaccard coefficient based on the term-matching also used to detect the plagiarism passages. The proposed method applies both VSM and Jaccard coefficient to compute the similarity measure between two sentences. The experimental results displayed that the method could generate better results than the methods which employs only the VSM or Jaccard coefficient.

Ekbal, Saha, and Choudhary (2012) proposed a method for plagiarism detection. It includes three major steps. First, in pre-processing step the basic tasks of natural language processing are done. This step contains the following functions. The generate tokens, the POS-classes and stop-word removal. In the second step, a number of source documents that are more similar to the suspicious document are selected. To identify the source documents for each suspicious document, it used Vector Space Model (VSM). In this model the both source document and suspicious document are represented in term of vector. Each cell of vector is weighted using term-frequency and inverse document frequency (TF-IDF) scheme. Finally the similarity measure between two documents is calculated using the cosine similarity. If the obtained similarity measure exceeds the predefined threshold, the document is considered as a source document for the suspicious document. In third step, the similar text in both source document and suspicious document are found using the n-gram method. Both documents are split into n-gram (i.e. $n = 3$), then using the similarity coefficient method (Abdi, Idris, Alguliev, & Aliguliyev, 2015; Nawab, Stevenson, & Clough, 2010) the similarity is calculated. If the similarity value exceeds the threshold value, the plagiarized texts are selected using the graph-based approach and the depth first search algorithm. Finally, in fourth step the plagiarized text are presented.

Ceska (2008) proposed a method based on Singular Value Decomposition (SVD) for plagiarism detection. It called SVDPlag. The proposed method includes the following steps. First, pre-processing performs several tasks such as, stop-word removed, Lemmatisation to identify the root of each word (Toman, Tesar, & Jezek, 2006) and part-of-speech of each word. Second, in Phrase Extraction, the documents are split into phrases in term of n-gram. Third, the phrase reduction aims to reduce the number of phrases. It used document frequency to identify whether a phrase is important or not. If a phrase appears only in one document, then the phrase will be removed, otherwise it is considered as an important phrase. Fourth, creating a matrix, in this step the documents are represented in form of matrix, where rows indicates documents and the column indicates the phrases. The weight of each cell, a_{ij}, in matrix is equal to the number of times that a phrase i was appeared in document j. Finally in the last step, the SVD is applied to the matrix, obtained from previous step, and the similarity measure between each pair of documents is computed. If the similarity score between two documents exceeds the pre-defined threshold, both documents are considered as plagiarized.

3. Proposed method

In this section we describe our plagiarism detection method. The general architecture of our proposed method is presented in Fig. 1. Our method includes three main steps. In the first step, pre-processing the basic natural language processing tasks is done. At detailed comparison step, the $doc_{suspicious}$ and the doc_{source} are decomposed into several sentences in order to identify pairs of sentences (S_q, S_x), where $S_q \in doc_{suspicious}$ and $S_x \in doc_{source}$, which are more similar. Finally, in the last step, the results of the previous step are considered as input for the post-processing step in order to present the plagiarized sentences.

We describe each of the aforementioned steps in the subsequent sections.

3.1. Pre-processing

The main task of this step is to prepare the source document doc_{source} and the suspicious documents $doc_{suspicious}$ for further processing. This step consists of three main functions, such as sentence segmentation, stop word removal and stemming.

Sentence segmentation — in this process, the source document and the suspicious documents are split into individual sentences, which are the textual units considered for comparison between the source document and the suspicious documents. A sentence ends with full stop (.) whereas a paragraph is ended by new line. Therefore, a paragraph consists of a group of sentences.

Stop word removal— stop words, are words which occurred frequently in a document and are meaningless words, such as articles, propositions and conjunctions (van Rijsbergen, 1986). According to the results of research (Tomasic & Garcia-Molina, 1993), the stop words include 50% of documents text words. Removing such words speeds the system processing and improves the performance of the method (Baeza-Yates, 1992). Using stop word removal, the words that are very common within a text and are also considered as noisy terms are removed. Obviously, their removal can be effective before the accomplishment of a natural language processing task. Removal of such words can improve accuracy and time requirements for comparisons by saving memory space and thus by increasing the speed of processing (Paul & Jamal, 2015). Such removal

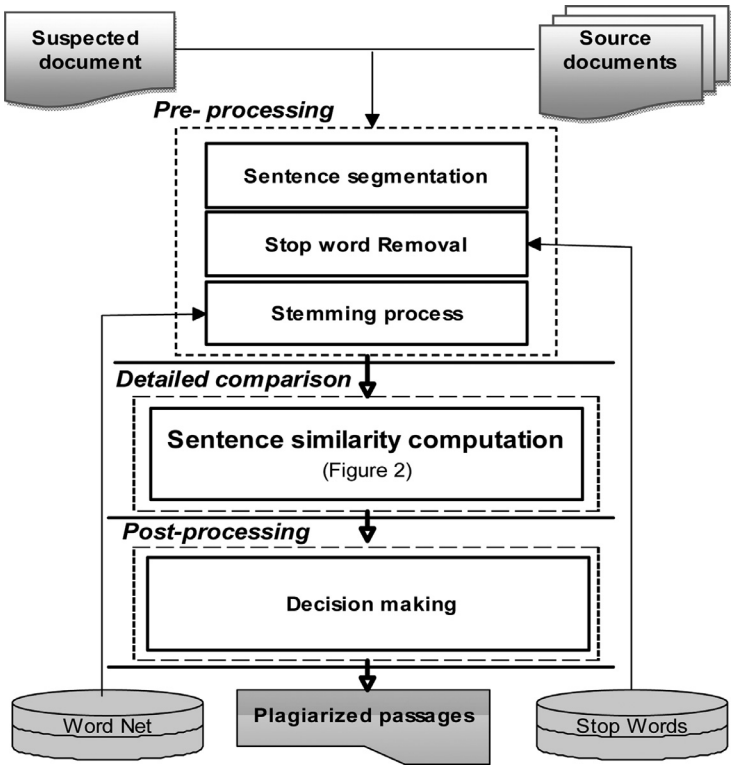

Fig. 1. The Architecture of the PDLK.

Table 1
Examples of stop words.

Stop words
'About, above, across, after, afterwards, again, against, all, almost, alone,'
'already, also, although, always, am, among, amongst, amongst, amount, an,'
'another, any, anyhow, anyone, anything, anyway, anywhere, are, around, as,'
'along, and, in, the, of' …

is usually performed by word filtering with the aid of a list of stop words. In our work, the stop words extracted from the English stop word list (http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop). Table 1 shows some of these stop words that may appear in a sentence.

Stemming—it is used to reduce word to its stem form. It is useful to identify words that belong to the same stem (e.g. *went* and *gone*, both come from the verb *go*). This process obtains the root of each word using the lexical database, Word Net. Our method used stemming in comparison between two documents.

Word Net is a lexical database for English which was developed at Princeton University (Miller & Charles, 1991). It includes 121,962 unique words, 99,642 synsets (each synset is a lexical concept represented by a set of synonymous words) and 173,941 senses of words.

3.2. Detailed comparison

The detailed comparison step includes the identification of all plagiarized sentences from the suspicious document and their corresponding source sentences from the source document. For this, given a suspicious document, $doc_{suspicious,}$ and a source document, doc_{source}. Both documents the doc_{source} and the $doc_{suspicious}$ are split into several sentences. After that a pair of sentences (S_q, S_x), where $S_q \in doc_{suspicious}$ and $S_x \in doc_{source}$, are considered as pairs of plagiarism sentences if their similarity measures exceeded the threshold value. We apply our method to detect all plagiarism sentences. The proposed method combined the semantic similarity and syntactic similarity to calculate the similarity measure between two sentences. The overall process of applying the semantic and syntactic information to calculate the similarity measure is shown in Fig. 2. These processes are as follows:

1. Two sentences S_q, S_x are considered as input.
2. The word-set is created using the two sentences.
3. The semantic-vector is created for each of two sentences.
4. The word-order vector is created for each of the two sentences.
5. The semantic word similarity approach is used to find the similar words. The steps 3 and 4 employ this method to create the semantic-vector and word-order vector.
6. It measures the semantic similarity measure between two sentences. The semantic similarity measure is determined by the cosine between the two corresponding semantic vectors.
7. It computes the word-order similarity measure between two sentences. The similarity score is determined by the syntactic-vector approach (Li, McLean, Bandar, O'shea, & Crockett, 2006). This approach will be explained in the next section.
8. Finally, it calculates the similarity measure between two sentences (S_q and S_x) using a linear equation that combines the obtained similarity measures from steps 6 and 7.
9. The final score obtained from the previous step is checked. If the similarity score exceed the threshold, a pair of sentences (S_q, S_x) is considered as plagiarized sentences.

Fig. 2 includes several components such as word-set, context word expansion, semantic similarity and syntactic similarity between sentences. The tasks of each component are explained in detail in the following sections.

(a) The Word Set

Given two sentences S_q and S_x, a "word-set" is produced using distinct words from the pair of sentences. Let $WS = \{W_1, W_2 \ldots W_N\}$ denote word set, where *N* is the number of distinct words in the word set. The word set between two sentences is obtained through certain steps as follows:

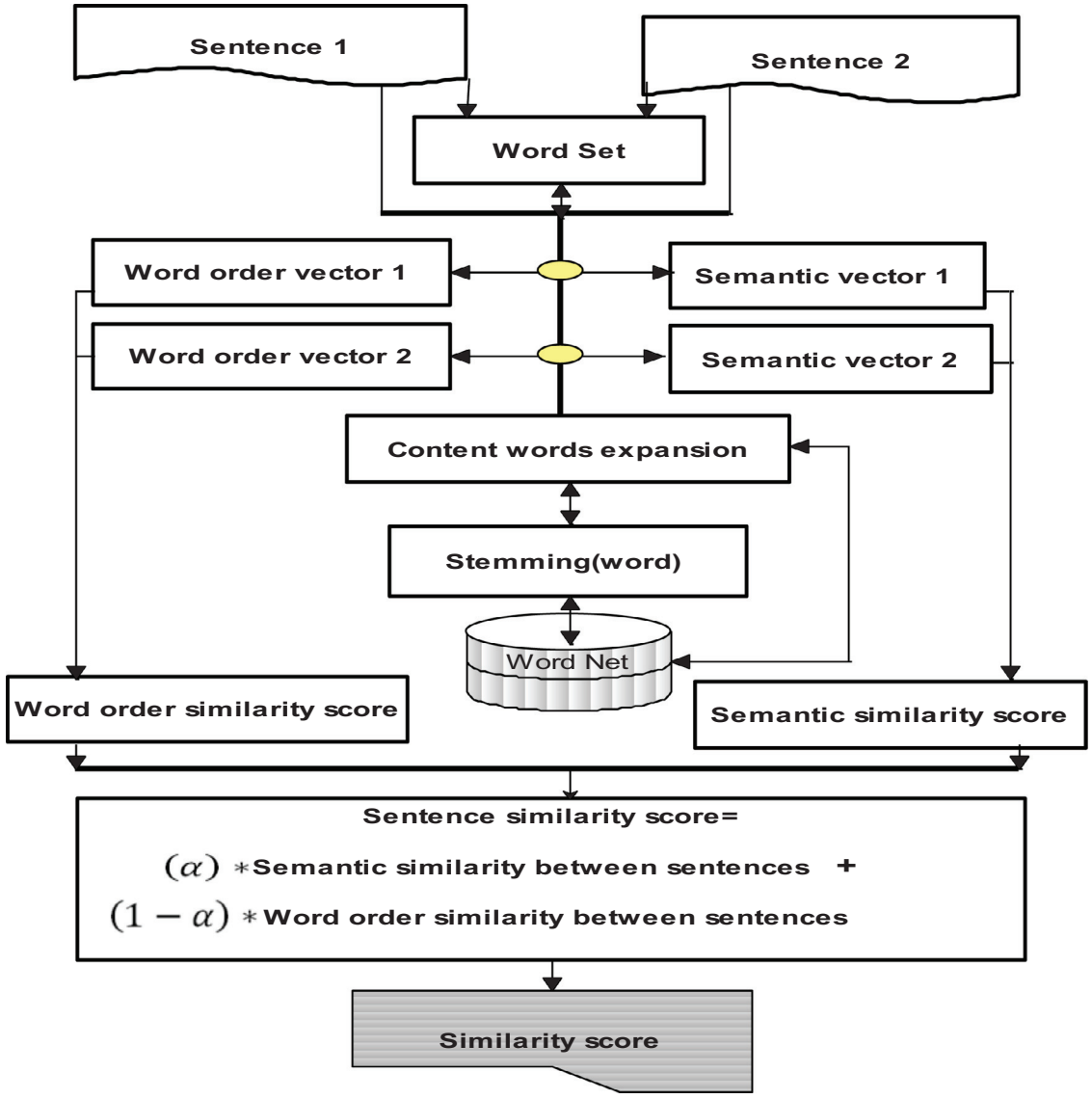

Fig. 2. Sentence similarity computation.

1. It takes two sentences as input.
2. By a loop for each word, *w*, from S_q, it undertakes certain tasks, which include:
   (i) It determines the root of the *w* (denoted by the *RW*) using the Word Net.
   (ii) It checks if the *RW* appears in the *WS*, it jumps to the step 2 and continues the loop by the next word from S_q, otherwise, it jumps to step iii.
   (iii) If the *RW* does not appear in the *WS*, then the *RW* is assigned to the *WS* and then it jumps to the step 2 to continue the loop by the next word from S_q.
   (iv) It conducts the same process for S_x.

(b) Semantic similarity between words (SSW)

Semantic similarity between words (Lin, 1998; Tian, Li, Cai, & Zhao, 2010) plays an important role in our method. It provides useful information to detect the plagiarism when an author presents someone else's idea as his or her own words using paraphrasing or rewording. The semantic similarity between two words is determined through these steps:

1. It takes two words, W_1 and W_2, as input.
2. It gets the root of each word using the lexical database, Word Net.
3. It gets the synonym of each word using the Word Net.
4. It determines the number of synonyms of each word.
5. It determines Least Common Subsume (LCS) of two words and their length.
6. It computes the similarity score between words using Eqs. (1) and (2).

We use the following equations to calculate the semantic similarity between two words (Aytar, Shah, & Luo, 2008; Mihalcea, Corley, & Strapparava, 2006; Warin, 2004):

$$IC\ (w)\ = 1 - \frac{Log(synset(w) + 1)}{\log(\max\_w)} \tag{1}$$

$$Sim(w_1, w_2) = \begin{cases} \frac{2*IC(LCS(w_1, w_2))}{IC(w_1) + IC(w_2)} & \text{if } w_1 \neq w_2 \\ 1 & \text{if } w_1 = w_2 \end{cases} \tag{2}$$

Where *LCS* stands for the least common subsume, *max_w* is the number of words in Word Net, Synset (*w*) is the number of synonyms of word w, and *IC*(*w*) is the information content of word *w* based on the lexical database Word Net.

(c) Semantic similarity between sentences

We use the semantic-vector approach (Alguliev, Aliguliyev, & Mehdiyev, 2011; Li et al., 2006) to measure the semantic similarity between sentences. The following tasks are performed to measure the semantic similarity between two sentences.

1. To create the semantic-vector.
   The semantic-vector is created using the word set and corresponding sentence. Each cell of the semantic-vector corresponds to a word in the word set, so the dimension equals the number of words in the word set.
2. To weight each cell of the semantic-vector.
   Each cell of the semantic-vector is weighted using the calculated semantic similarity between words from the word set and corresponding sentence. As an example:
   (i) If the word, *w*, from the word set appears in the sentence S_q, the weight of the *w* in the semantic vector is set to 1. Otherwise, go to the next step.
   (ii) If the sentence S_q does not contain the *w*, then compute the similarity score between the *w* and the words from sentence S_q using the *SSW* approach.
   (iii) If exist similarity values, then the weight of the *w* in the semantic-vector is set to the highest similarity value. Otherwise, go to the next step.
   (iv) If there is no similarity value, then the weight of the *w* in the semantic-vector is set to 0.
3. The semantic-vector is created for each of the two sentences. The semantic similarity measure is computed based on the two semantic-vectors. The following equation is used to calculate the semantic similarity between sentences:

$$Sim_{semantic}(S_q, S_x) = \frac{\sum_{j=1}^{m}\left(w_{1j} \times w_{2j}\right)}{\sqrt{\sum_{j=1}^{m} w_{1j}^2} \times \sqrt{\sum_{j=1}^{m} w_{2j}^2}} \tag{3}$$

where $S_q = (w_{11}, w_{12}, \ldots, w_{1m})$ and $S_x = (w_{21}, w_{22}, \ldots, w_{2m})$ are the semantic vectors of sentences S_q and S_x, respectively; w_{pj} is the weight of the j^{th} word in vector S_p, *m* is the number of words.

(d) Word-order similarity between sentences

We use the syntactic-vector approach (Li et al., 2006) to measure the word-order similarity between sentences. The following tasks are performed to measure the word-order similarity between two sentences.

1. To create the syntactic-vector.
   The syntactic-vector is created using the word set and corresponding sentence. The dimension of current vector is equal to the number of words in the word set.
2. To weight each cell of the syntactic-vector.
   Unlike the semantic-vector, each cell of the syntactic-vector is weighted using a unique index. The unique index can be the index position of the words that appear in the corresponding sentence. However, the weight of each cell in syntactic-vector is determined by the following steps:
   (i) For each word, *w*, from the word set. If the w appears in the sentence S_q the cell in the syntactic-vector is set to the index position of the corresponding word in the sentence S_q. Otherwise, go to the next step.
   (ii) If the word w does not appear in the sentence S_q, then compute the similarity score between the *w* and the words from sentence S_q using the *SSW* approach.
   (iii) If exist similarity values, then the value of the cell is set to the index position of the word from the sentence S_q with the highest similarity measure.
   (iv) If there is not a similar value between the w and the words in the sentence S_q, the weight of the cell in the syntactic-vector is set to 0.
3. For both sentences the syntactic-vector is created. Then, the syntactic similarity measure is computed based on the two syntactic-vectors. The following equation is used to calculate word-order similarity between sentences:

$$Sim_{\text{word order}}(S_q, S_x) = 1 - \frac{||\boldsymbol{O}_1 - \boldsymbol{O}_2||}{||\boldsymbol{O}_1 + \boldsymbol{O}_2||} \tag{4}$$

where $O_1 = (d_{11}, d_{12}, \cdots, d_{1m})$ and $O_2 = (d_{21}, d_{22}, \cdots, d_{2m})$ are the syntactic-vectors of sentences S_q and S_x, respectively; d_{pj} is the weight of the j^{th} cell in vector O_p.

(e) Sentence similarity measurement

The similarity measure between two sentences is calculated using a linear equation that combines the semantic and word-order similarity. The similarity measure is computed as follows:

$$\begin{aligned}\text{Sim}_{\text{sentences}}(S_q, S_x) &= \alpha \cdot \text{sim}_{\text{semantic}}(S_q, S_x) \\ &\quad + (1-\alpha) \cdot \text{sim}_{\text{wordorder}}(S_q, S_x)\end{aligned} \tag{5}$$

where $0 < <1$ is the weighting parameter, specifying the relative contributions to the overall similarity measure from the semantic and syntactic similarity measures. The larger the α, the heavier the weight for the semantic similarity. If $= 0.5$ the semantic and syntactic similarity measures are assumed to be equally important.

3.3. Post-processing

Given a set of pairs sentences (S_q, S_x), extracted from previous Section 3.2. The post-processing step selects all pair sentences as plagiarized sentences using the following steps.

Step 1. It removes all pair sentences that do not meet certain criteria. This step removes a pair of sentences whose the similarity measure under Eq. (5) is below a threshold value. The following formula is used to judge whether one sentence exists plagiarism or not:

$$P(S_q, S_x) = \begin{cases} 1, & Sim(S_q, S_x) \geq t_1 \\ 0, & others \end{cases} \tag{6}$$

where $Sim(S_q, S_x)$ is Eq. (5) described above, and 1 indicates plagiarisms, 0 indicates none-plagiarism. The t_1 is the threshold value.

Step 2. Let $d_{src} = \{S_{q1}, S_{q2} \ldots S_{qN}\}$ represent all sentences from a source document, $\text{doc}_{\text{source}}$, where *N* is the number of sentences. S_x indicates a sentence of a suspicious document, $\text{doc}_{\text{suspicious}}$. Let $Arr_{Pla} = \{(S_{q1}, S_x, Value_{sim(S_{q1}, S_x)}), (S_{q2}, S_x, Value_{sim(S_{q2}, S_x)}) \ldots (S_{qM}, S_x, Value_{sim(S_{qM}, S_x)})\}$ represent all the sentences from the $\text{doc}_{\text{source}}$ that their similarity measure with S_x exceeded the threshold t_1, where $M \leq N$ and $Value_{sim(S_{qM}, S_x)}$ indicates the similarity measure between two sentences S_{qM} and S_x . Based on the previous step, a sentence from a suspicious document can have several matching sentences from a source document, and thus generate several sentence pairs. In such case, we select a pair $(S_{qM}, S_x, Value_{sim(S_{qM}, S_x)})$ of sentences from the Arr_{Pla} which have a greatest similarity measure, $Value_{sim(S_{qM}, S_x)}$. Finally, the selecteted pair of sentence which judged as plagiarized sentences is screened.

4. Experiments

Our proposed method, PDLK, has been applied for plagiarism detection. We conducted the experiments on the data sets provided by PAN-PC-10 and PAN-PC-11 (http://www.pan.webis.de/).

4.1. Data set

In this section, we describe the data used throughout our experiments. For assessment of the performance of the proposed method we used the datasets provided in PAN-PC-10 and PAN-PC-11. Each

dataset includes suspicious and corresponding source document sets. The PAN-PC-10 corpus comprises 27,073 documents split into a set of 15,925 suspicious documents and a set of 11,148 source documents. The PAN-PC-11corpus also includes 22,186 documents split into a set of 11,093 suspicious documents and same number of source documents. Both corpuses are based on the 22,730 books from the Project Gutenberg (www.gutenberg.org). There are several plagiarism cases in the PAN-10 and PAN-11 corpus. The plagiarism cases have been generated by human (simulated) or by a computer program (artificial) able to obfuscate a text by removing, inserting, or replacing words or short phrases by one of its synonyms and antonyms.

In both corpuses, the main task is to find all plagiarized passages in the suspicious documents and, if available, the corresponding source passages. In order to evaluate the performance of our method, we conducted two experiments. In the first experiment, we used plagiarized documents plus the original documents from PAN-PC-10 corpus for parameter tuning (the threshold and the α). In the second experiment, we used the data provided in PAN-PC-11 to compare our method with the other method and with the systems that participated in PAN-PC-11.

4.2. Evaluation metrics

In order to evaluate and compare the performance of our proposed method, we used four various standard measures, macro-average Precision (Prec), Recall (Rec), F-measure and granularity (gran) (Potthast, Barrón-Cedeño, Eiselt, Stein, & Rosso, 2010; Stamatatos, 2011). In more detail, we define S is the set of plagiarism cases and R is the set of detections that found by method. Let r and s are the elements in R and S respectively, so the macro-average precision and recall are computed using the Eqs. (7) and (8). Precision denotes what portion of the detection cases identified by system are plagiarism cases. Recall denotes what portion of the plagiarism cases are identified by the system.

$$prec\ (S,R) = \frac{1}{|R|}\sum_{r\in R}\frac{|\mathrm{U}_{s\in S}(s\cap r)|}{|r|} \tag{7}$$

$$\mathrm{Rec}\ (S,R) = \frac{1}{|S|}\sum\nolimits_{s\in S}\frac{|\mathrm{U}_{r\in R}(s\cap r)|}{|s|} \tag{8}$$

where,

$$s\cap r = \begin{cases} s\cap r, & \text{if r detects s,}\\ \emptyset, & \text{otherwise.}\end{cases}$$

There is an anti–correlation between precision and recall (Manning et al., 2008). It means the recall drops when the precision rises and vice versa. In other words, a system attempts for recall will get lower precision and a system attempts for precision will get lower recall. To take into consideration the two metrics together, a single measure, called F-score, is used. F-score is a statistical measure that merges both precision and recall. It is calculated as follows:

$$\mathrm{F-measure} = \frac{1}{\alpha\times\frac{1}{P}+(1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)P\times R}{\beta^2\times P+R} \tag{9}$$

where $\beta^2 = \frac{1-\alpha}{\alpha}$, $\alpha\in[0,1]$, and $\beta^2\in[0,\infty]$. If a large value ($\beta>1$) assigns to the β, it indicates that precision has more priority. If a small value ($\beta<1$) assigns to the β, it indicates that recall has more priority. If $\beta=1$ the precision and recall are assumed to have equally priority in computing F-score. F-score for $\beta=1$ is computed as follows:

$$\mathrm{F-measure} = \frac{2\times P\times R}{P+R} \tag{10}$$

where P is precision and R is recall.

Besides precision and recall we used another evaluation metric, granularity, to determine the efficiency of our detection method. Granularity is described as the ratio of number of identified plagiarized source section to given plagiarized source section. The granularity measure is determined as follows:

$$Gran\ (S,R) = \frac{1}{S_R}\sum_{s\in S_R}|R_S| \tag{11}$$

where $S_R\sqsubseteq S$ cases are identified by detections in R and $R_S\sqsubseteq R$ are the dictions of a given s:

$$S_R = \{s|s\in S\wedge\ \ni r\in R: r\ detects\ s\} \tag{12}$$

$$R_S = \{r|r\in R\ \wedge\ r\ detects\ s\} \tag{13}$$

The domain of $gran(S,R)$ is $[1,|R|]$. The minimum and ideal granularity value is 1 and $|R|$ indicates the worst case.

Moreover, in order to makes a unique ranking among detection methods, the above measures are combined into a single overall score as follows:

$$plagdet\ (S,R) = \frac{F-measure}{\log_2(1+gran(S,R))} \tag{14}$$

4.3. Parameter setting

The proposed method requires two parameters to be determined before use: a weighting parameter (α) for weighting the significance between semantic information and syntactic information and a threshold (t_1). Both parameters in the current experiment were found using 300 suspicious documents and source documents. All documents are decomposed into some sentences. The stop words from sentences are removed. We ran our proposed method on the current data set. We used Eqs. (5), (7), (8), (10), (11) and (14). Eq. (5) is used to calculate the similarity measure. Eqs. (7), (8) and (10) are used to calculate the value of precision, recall and F-measure respectively. Eq. (11) is used to calculate the granularity. We evaluate our method for each peer (α) between 0.1 and 0.9 with a step of 0.1 and (t_1) between 0.1 and 1 with a step of 0.1, (e.g. = 0.4, $\beta=0.7$). Table 1 presents our experimental results achieved by using the various α and the t_1 values. We evaluate the results in terms of recall, precision, F-measure and granularity. By analysing the results, we find that the best performance is achieved by a = 0.8 and $t_1=0.6$. This α and the t_1 produced the recall, precision, F-measure, plagdet and granularity values as follows: 0.685 (recall), 0.802 (precision), 0.739 (F-measure), 0.733 (plagdet) and 1.010 (granularity). The best values of Table 2 have been marked in boldface. As a result, using the current data set, we obtained the best result when we use 0.8 as the α value and 0.6 as the t_1 value. Therefore, we can recommend this the α and the t_1 values for use on the rest of the data set.

4.4. Comparison with PAN-PC-11 systems

To confirm the aforementioned results, we validate our proposed method, PDLK, using a comparison of the overall recall, precision, F-measure and plagdet value obtained by PDLK and the participating systems in PAN-PC-11(Overview of the 3rd International Competition on Plagiarism Detection): (a) the top four systems with the highest plagdet value: Sys-1 (Cooke, Gillam, Wrobel, Cooke, & Al-Obaidli, 2011), Sys-2 (R. M. A. Nawab, Stevenson, & Clough, 2011), Sys-3 (Rao, et al., 2011) and Sys-4 (Grman & Ravas, 2011).

We apply our method to the 200 previously unused suspicious documents and source documents only with the α value 0.8 and the threshold value 0.6. Table 3 and Fig. 3 present the obtained results of recall, precision, F-measure and plagdet with the α of 0.8 and the threshold of 0.6. The obtained results prove that PDLK outperforms the other examined methods and that our method produces very competitive results. PDLK is also able to obtain the Plagdet of (0.789) in comparison with the best existing method, Sys-4, which has the Plagdet of (0.615).

Table 2
Performance of the PDLK against various threshold and the α values on PAN-PC-10 data set. (Due to space limitations of this paper, a sample results are shown.)

Weighting (α)	Threshold	Precision	Recall	F-measure	Plagdet	Granularity
$\alpha = (0.1–0.7)$	*0.1*	–	–	–	–	–
	–	–	–	–	–	–
	1	–	–	–	–	–
$\alpha = 0.8$	0.1	0.625	0.500	0.556	0.509	1.131
	0.2	0.659	0.528	0.586	0.543	1.114
	0.3	0.693	0.552	0.615	0.594	1.050
	0.4	0.722	0.639	0.678	0.666	1.023
	0.5	0.775	0.655	0.710	0.695	1.031
	0.6	**0.802**	**0.685**	**0.739**	**0.733**	**1.010**
	0.7	0.701	0.614	0.655	0.641	1.029
	0.8	0.612	0.555	0.582	0.567	1.038
	0.9	0.522	0.468	0.494	0.309	2.025
	1	0.478	0.350	0.404	0.245	2.133
$\alpha =(0.9)$	*0.1*	–	–	–	–	–
	–	–	–	–	–	–
	1	–	–	–		–

Table 3
Performance comparison between PDLK and PAN-PC-11 systems.

Comparative performance results				
System	Precision	Recall	F-measure	Plagdet
PDLK	0.902	0.702	0.790	0.789
Sys-1	0.711	0.150	0.248	0.247
Sys-2	0.278	0.089	0.134	0.267
Sys-3	0.454	0.162	0.239	0.199
Sys-4	0.893	0.473	0.618	0.615

Table 4
Performance comparison between PDLK and other methods.

Comparative performance results				
System	Precision	Recall	F-measure	Plagdet
PDLK	0.902	0.702	0.790	0.789
Crm-1	0.659	0.190	0.295	0.289
Crm-2	0.858	0.685	0.762	0.757
Crm-3	0.742	0.659	0.698	0.696
Crm-4	0.557	0.697	0.619	0.609
Crm-5	0.867	0.555	0.677	0.674
Crm-6	0.834	0.500	0.626	0.625
Crm-7	0.893	0.552	0.683	0.683

4.5. Comparison with related methods (Crm)

In this section, the performance of our method is compared with other well-known or recently proposed methods. In particular, to evaluate our methods on PAN-PC-11 data set, we select the following methods: Crm-1 (Ekbal et al., 2012), Crm-2 (Wang et al., 2013), Crm-3 (Grozea, Gehl, & Popescu, 2009), Crm-4 (Kasprzak, Brandejs, & Kripac, 2009), Crm-5 (Oberreuter, Ríos, & Velásquez, 2010), Crm-6 (Rodríguez-Torrejón & Martín-Ramos, 2010) and Crm-7 (Suchomel, Kasprzak, & Brandejs). These methods have been chosen for comparison because they have achieved the best results on the PAN-PC data set. The evaluation metrics values are reported in Table 4 and Fig. 4.

4.6. Detailed comparison

From the comparison of the evaluation metrics values for PAN-PC-11 systems and other methods, PDLK obtains a considerable improvement. Tables 5 and 6 show the improvement of PDLK for all four evaluation metrics. It is clear that PDLK obtains the high Plagdet and outperforms all the other methods. We use the relative improvement ($\frac{\text{Our method}-\text{Other method}}{\text{Other method}}$) $\times$ 100, for comparison. In Tables 5 and 6 "+" means the proposed method improves the PAN-PC-11 systems and existing methods. Table 5 shows among the PAN-PC-11 systems the Sys-4 displays the best results compared to Sys-1, Sys-2 and Sys-3. In comparison with the method Sys-4, PDLK improves the performance of the Sys-4 method as follows: 1.070% (precision), 48.419% (recall), 27.697% (F-measure) and 28.153% (plagdet).

Table 6 displays among the existing methods the Crm-2 shows the best results compared to Crm-1, Crm-3, Crm-4, Crm-5, Crm-6 and Crm-7. In comparison with the method Crm-2, PDLK improves the performance of the Crm-2 method as follows: 5.168% (precision), 2.541% (recall), 3.690% (F-measure) and 4.209% (plagdet).

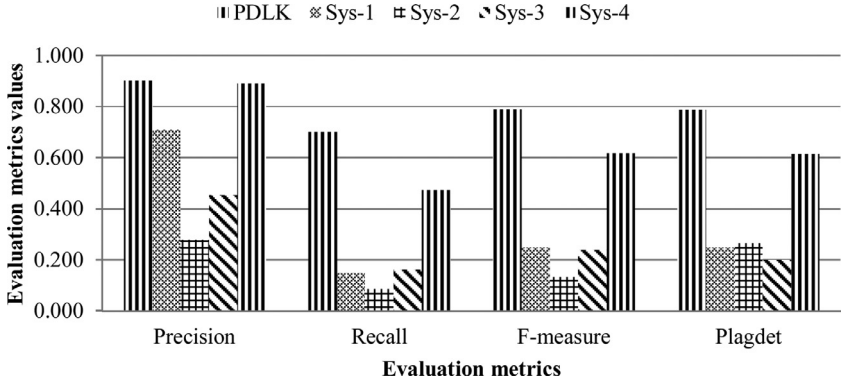

Fig. 3. Performance comparison between PDLK and PAN-PC-11 systems.

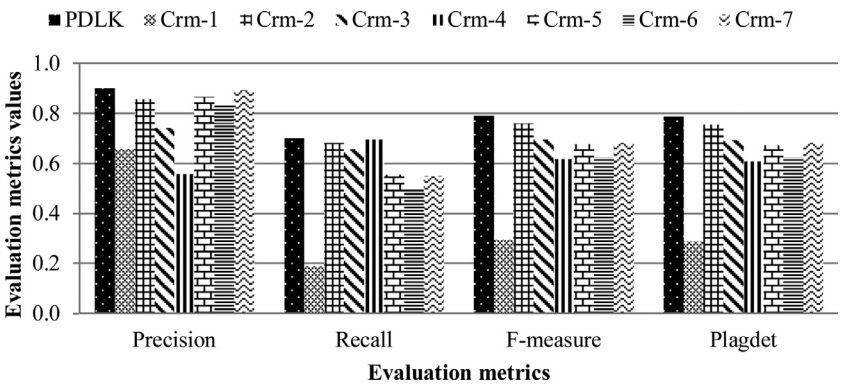

Fig. 4. Performance comparison between PDLK and other methods.

Table 5
Performance evaluation compared between the PDLK and PAN-PC-11 systems.

PDLK improvement (%)				
System	Precision	Recall	F-measure	Plagdet
Sys-1	+26.968	+367.990	+218.743	+219.634
Sys-2	+224.568	+693.458	+488.249	+195.127
Sys-3	+98.694	+333.981	+231.008	+296.125
Sys-4	+1.070	+48.419	+27.697	+28.153

Table 6
Performance evaluation compared between the PDLK and other methods.

PDLK improvement (%)				
System	Precision	Recall	F-measure	Plagdet
Crm-1	+36.857	+268.808	+167.295	+172.792
Crm-2	+5.168	+2.541	+3.690	+4.209
Crm-3	+21.637	+6.638	+13.202	+13.349
Crm-4	+61.906	+0.791	+27.538	+29.437
Crm-5	+4.034	+26.453	+16.642	+17.086
Crm-6	+8.134	+40.324	+26.236	+26.142
Crm-7	+1.023	+27.104	+15.689	+15.514

4.7. Discussion

The current section presents the following main findings that obtained from Tables 2– 5. The obtained results validate our method. This is due to the fact that, (a) it is able to identify the synonym or similar words among all sentences using a lexical database, WordNet. It is very important to consider this aspect (identifying the synonym or similar words) when measuring the similarity score of sentence-to-sentence. (b) Given two sentences (i.e., S_1: *Alex likes Allen;* S_2: *Allen likes Alex*), unlike other method, our method is able to distinguish the meaning of two sentences by using the combination of semantic and syntactic information. Moreover, the main feature of the proposed method is its ability to carry out the sentence matching semantically and syntactically. (c) Tables 4 and 6 show that our method obtained good result in precision, recall and F-measure scores. The results confirm that our method outperforms the other methods. In addition, the results show that the combination of semantic and syntactic information; and the semantic word similarity can improve the performance.

4.8. Influence of semantic similarity between sentences, word-order similarity between sentences and semantic similarity between words

To examine the efficiency of semantic similarity between sentences, word-order similarity between sentences and semantic similarity between words on our proposed method, PDLK, we applied our method to current dataset (PAN-PC-11) using three different tests:

Table 7
Performance of the PDLK against various tests (SOW, SSBW, SSW).

Various tests			
Method	Test 1 (*SOW*) Precision	Test 2 (*SSBW*) Recall	Test 3 (*SSW*) F-measure
PDLK	0.887	0.823	0.854
PDLK	0.652	0.571	0.609
PDLK	0.239	0.452	0.313

1. Test 1 – SOW, to calculate sentence similarity measurement using semantic similarity between sentences, word-order similarity between sentences and semantic similarity between words.
2. Test 3 – SSBW, to calculate sentence similarity measurement using semantic similarity between sentences and semantic similarity between words, without word-order similarity between sentences.
3. Test 2 – SSW, to calculate sentence similarity measurement using semantic similarity between sentences and word-order similarity between sentences, without semantic similarity between words.

We aim to determine what combination (SOW, SSBW and SSW) should be chosen to calculate similarity measure between two sentences. Table 7 and Fig. 5 show the results obtained with recall, precision and F-measure for different tests. This table shows that the best result is obtained with SOW; the mean result is obtained by SSBW; and the worst result is obtained by SSW. Based on the evaluation results using the test cases 1, 2 and 3, the SSBW and SSW have not much effect on improving system performance. This is because of the fact that (a) SSW is not able to identify the synonym words among all sentences; (b) SSBW does not take into account the word order or syntactic information to compute text similarity.

The experimental results indicated that the SOW gave higher performance than the SSBW and SSBW. SOW is capable to improve the system performance. It calculates similarity measure between two sentences using semantic similarity between sentences, word-order similarity between sentences and semantic similarity between words. Therefore, we employed the SOW to compute the similarity measure between two sentences in our proposed method to obtain high performance.

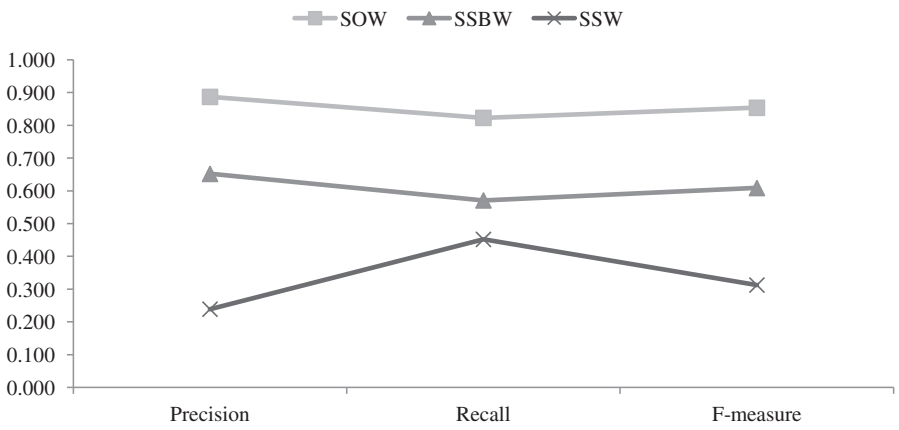

Fig. 5. Performance of the PDLK against various tests (SOW, SSBW, SSW).

4.9. Runtime complexity analysis

In this section, we analyse the time complexity of the proposed method. The time complexity of the proposed method usually depends on the number of source documents and their total sentences. In order to estimate the amount of computation during the comparison of the suspected document and source documents, we make the following assumptions:

1. Let m be the total number of sentence in suspected document. Let n be the total number of sentences in source documents. The n is calculated as follows:

$$n = \sum_{i=1}^{N} \text{Count}_{\text{sentence}}(\text{Doc}_i) \tag{15}$$

where
N is the total number of source documents. Count sentence (Doc_i) is the total number of sentences in a source document. Therefore, the time complexity for the comparison between source documents and suspected one is $O(n \times m)$.

2. According to proposed method, after the comparison operation, we need take one more step which is the post-processing. In this step the pairs sentences extracted from the previous step are stored in an array. This step stores a pair of sentences whose similarity measure is above a threshold value. We also need to sort the elements of the array based on the similarity values from large to small. The elements of array are sorted using the quick sort. The total amount of computation to do quick sort in array is calculated as follows.

$$O(k \log_2 k) \tag{16}$$

where k is the length of array.

Thus summing up the above complexities, total time complexity becomes,

$$O(n \times m \times k \log_2 k) \tag{17}$$

5. Conclusion and future work

Plagiarism detection in large document collections should be both efficient and effective. In this paper, we propose a method based on the linguistic knowledge to detect the plagiarized text. It employs three similarity metrics to calculate similarity measure between two sentences: (a) semantic similarity between sentences; (b) word-order similarity between sentences; and (c) semantic similarity between words. We analysed the influence of three similarity metrics on our proposed method. Due to the results as shown in Table 7 and Fig. 5, we selected the best combination of them. The main feature of the proposed method is its ability to capture the meaning the meaning in comparison between a source text passage and suspicious text passage, when two passages have same surface text or different words have been used in the passages. This method is also able to detect common actions performed by plagiarists such as the exact copied text, paraphrasing, transformation of sentences and changing of word structure in the sentences.

The plagiarism detection evaluation of PDLK is conducted over PAN-PC plagiarism dataset that comprises a wide variety of text lengths. The proposed method is very easy to follow and requires minimal text processing cost. Initially, parameters of PDLK are optimized over the PAN-PC-10 dataset. Later the actual plagiarism detection evaluation is done over PAN-PC-11 dataset. PDLK is compared with the participating system in PAN-PC-11 and the current methods which are well-known existing methods that are used in plagiarism detection. The experimental results display that the performance of the proposed method is very competitive when compared with other methods. The results also displayed that PDLK improved the performance of the participating system in PAN-PC-11 and the current methods. We observed that PDLK is able to obtain the plagdet of 0.789 in comparison with the best participating system in PAN-PC-11, (Sys-4), which had plagdet of 0.615 and the best existing system, (Crm-2), which had plagdet of 0.757.

As future work we plan to improve the proposed method by considering identifying passive and active sentence, and expanding the semantic knowledge base, which are limitations of the current method. (a) The method is not able to distinguish between an active sentence and a passive sentence. Given a suspicious sentence (*A*: '*Teacher likes his student.*') and two source sentences (*B*: '*student likes his teacher.*'; *C*: '*student is liked by his teacher.*'), although the similarity measure between sentences (*A* and *B*) and (*A* and *C*) is same, but as we can see the meaning of sentence *A* is more similar to the sentence *C*. hence, it is important to know what passive and active sentences are before comparisons can be drawn. (b) The method used WordNet as the main semantic knowledge base to calculate the semantic similarity between words. The comprehensiveness of Word Net is determined by the proportion of words in the text that are covered by its knowledge base. However, the main criticism of WordNet concerns its limited word coverage to calculate semantic similarity between words. Obviously, this disadvantage has a negative effect on the performance of our proposed algorithm. To tackle this problem, in addition to WordNet, other knowledge resources, such as Wikipedia and other large corpus should be used.

In addition to the aforementioned future works, the following works are also considered as future works. In future, we aim to extend our method to detect intrinsic plagiarism where there is no

reference collection. Further enhance proposed method by reduction the runtime complexity with a parallel programming and adding additional functionality to the method. Also, we aim to add a selecting candidate documents step to method, to select a set of document from the corpus that are more similar to the doc suspicious. This step can be useful for increasing the overall performance of the proposed method.

References

Abdi, A., Idris, N., Alguliev, R. M., & Aliguliyev, R. M. (2015). Automatic summarization assessment through a combination of semantic and syntactic information for intelligent educational systems. *Information Processing & Management, 51*, 340–358.

AdelsonVelskii, M., & Landis, E.M. (1963). An algorithm for the organization of information. DTIC document.

Alguliev, R. M., Aliguliyev, R. M., & Mehdiyev, C. A. (2011). Sentence selection for generic document summarization using an adaptive differential evolution algorithm. *Swarm and Evolutionary Computation, 1*, 213–222.

Aytar, Y., Shah, M., & Luo, J. (2008). Utilizing semantic word similarity measures for video retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition, 2008. CVPR 2008.* (pp. 1–8). IEEE.

Baeza-Yates, R.A. (1992). Introduction to data structures and algorithms related to information retrieval.

Bao, J.-P., Shen, J.-Y., Liu, X.-D., & Song, Q.-B. (2003). A survey on natural language text copy detection. *Journal of Software, 14*, 1753–1760.

Barrón-Cedeño, A., Gupta, P., & Rosso, P. (2013). Methods for cross-language plagiarism detection. *Knowledge-Based Systems, 50*, 211–217.

Campos, R. A. C., & Martinez, F. J. Z. (2012). Batch source-code plagiarism detection using an algorithm for the bounded longest common subsequence problem. In *Proceedings of the 9th international conference on electrical engineering, computing science and automatic control (CCE), 2012* (pp. 1–4). IEEE.

Ceska, Z. (2008). Plagiarism detection based on singular value decomposition. In *Advances in natural language processing* (pp. 108–119). Springer.

Ceska, Z., Toman, M., & Jezek, K. (2008). Multilingual plagiarism detection. In *Artificial intelligence: Methodology, systems, and applications* (pp. 83–92). Springer.

Cooke, N., Gillam, L., Wrobel, P., Cooke, H., & Al-Obaidli, F. (2011). A High-performance plagiarism detection system-notebook for PAN at CLEF 2011. In *Proceedings of the CLEF (notebook papers/labs/workshop)*.

Ekbal, A., Saha, S., & Choudhary, G. (2012). Plagiarism detection in text using vector space model. In *Proceedings of the conference on hybrid intelligent systems, HIS* (pp. 366–371).

El-Alfy, E.-S. M., Abdel-Aal, R. E., Al-Khatib, W. G., & Alvi, F. (2015). Boosting paraphrase detection through textual similarity metrics with abductive networks. *Applied Soft Computing, 26*, 444–453.

Foster, C. C. (1965). Information retrieval: Information storage and retrieval using AVL trees. In *Proceedings of the 20th national ACM conference.* (pp. 192–205). ACM.

Franzke, M., & Streeter, L.A. (2006). Building student summarization, writing and reading comprehension skills with guided practice and automated feedback. Highlights from research at the University of Colorado, a white paper from Pearson Knowledge Technologies.

Geravand, S., & Ahmadi, M. (2014). An efficient and scalable plagiarism checking system using Bloom filters. *Computers & Electrical Engineering, 40*, 1789–1800.

Grman, J., & Ravas, R. (2011). Improved implementation for finding text similarities in large collections of data: Notebook for PAN at CLEF 2011. In *Notebook Papers of CLEF*.

Grozea, C., Gehl, C., & Popescu, M. (2009). ENCOPLOT: Pairwise sequence matching in linear time applied to plagiarism detection. In *Proceedings of the 3rd PAN workshop on uncovering plagiarism, authorship and social software misuse* (p. 10).

Hussein, A. S. (2015). Arabic document similarity analysis using n-grams and singular value decomposition. In *Proceedings of the IEEE 9th International Conference on Research Challenges in Information Science (RCIS), 2015* (pp. 445–455). IEEE.

Iliopoulos, C. S., & Rahman, M. S. (2009). A new efficient algorithm for computing the longest common subsequence. *Theory of Computing Systems, 45*, 355–371.

Irving, R. W., & Love, L. (2003). The suffix binary search tree and suffix AVL tree. *Journal of Discrete Algorithms, 1*, 387–408.

Kasprzak, J., Brandejs, M., & Kripac, M. (2009). Finding plagiarism by evaluating document similarities. In *Proceedings of the conference on SEPLN: Vol. 9* (pp. 24–28).

Kong, L., Qi, H., Wang, S., Du, C., Wang, S., & Han, Y. (2012). Approaches for candidate document retrieval and detailed comparison of plagiarism detection—notebook for PAN at CLEF 2012. In Proceedings of the conference on CLEF 2012 evaluation labs and workshop–working notes papers, (pp. 17–20).

Li, Y., McLean, D., Bandar, Z. A., O'shea, J. D., & Crockett, K. (2006). Sentence similarity based on semantic nets and corpus statistics. *Knowledge and Data Engineering, IEEE Transactions on, 18*, 1138–1150.

Lin, D. (1998). An information-theoretic definition of similarity. In *Proceedings of the conference on ICML: Vol. 98* (pp. 296–304).

Mahdavi, P., Siadati, Z., & Yaghmaee, F. (2014). Automatic external Persian plagiarism detection using vector space model. In *Proceedings of the 4th international conference on computer and knowledge engineering (ICCKE), 2014* (pp. 697–702). IEEE.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*: Vol. 1. Cambridge: Cambridge university press.

Mihalcea, R., Corley, C., & Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the conference on AAAI: Vol. 6* (pp. 775–780).

Miller, G. A., & Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes, 6*, 1–28.

Nawab, R., Stevenson, M., & Clough, P. (2010). University of sheffield: Lab report for PAN at CLEF 2010. In *Proceedings of the conference on CLEF 2010 LABs and workshops, notebook papers*. CLEF.

Nawab, R. M. A., Stevenson, M., & Clough, P. (2011). External plagiarism for PAN at CLEF 2011. In *Proceedings of the 5th international workshop on uncovering plagiarism, authorship, and social software misuse*. Sheffield.

Oberreuter, G., Ríos, S.A., & Velásquez, J.D. (2010). FASTDOCODE: Finding approximated segments of n-grams for document copy detection lab report for PAN at CLEF 2010. Detection using information retrieval and sequence alignment-notebook.

Oberreuter, G., & VeláSquez, J. D. (2013). Text mining applied to plagiarism detection: The use of words for detecting deviations in the writing style. *Expert Systems with Applications, 40*, 3756–3763.

Oktoveri, A., Wibowo, A. T., & Barmawi, A. M. (2014). Non-relevant document reduction in anti-plagiarism using asymmetric similarity and AVL tree index. In *Proceedings of the 5th International Conference on Intelligent and Advanced Systems (ICIAS), 2014* (pp. 1–5). IEEE.

Osman, A. H., Salim, N., Binwahlan, M. S., Alteeb, R., & Abuobieda, A. (2012). An improved plagiarism detection scheme based on semantic role labeling. *Applied Soft Computing, 12*, 1493–1502.

Paul, M., & Jamal, S. (2015). An improved SRL based plagiarism detection technique using sentence ranking. *Procedia Computer Science, 46*, 223–230.

Potthast, M., Barrón-Cedeño, A., Eiselt, A., Stein, B., & Rosso, P. (2010). Overview of the 2nd international competition on plagiarism detection. In *Proceedings of the conference on CLEF (notebook papers/labs/workshops)*.

Potthast, M., Gollub, T., Hagen, M., Kiesel, J., Michel, M., Oberländer, A., Tippmann, M., Barrón-Cedeno, A., Gupta, P., & Rosso, P. (2012). Overview of the 4th international competition on plagiarism detection. In *Proceedings of the conference on CLEF (online working notes/labs/workshop)*.

Raghavan, V. V., & Wong, S. M. (1986). A critical analysis of vector space model for information retrieval. *Journal of the American Society for information Science, 37*, 279–287.

Rao, S., Gupta, P., Singhal, K., & Majumder, P. (2011). External & intrinsic plagiarism detection: VSM & discourse markers based approach-notebook for PAN at CLEF 2011. In *Proceedings of the conference on CLEF (notebook papers/labs/workshop)*.

Raphael, A. (2002). Signature extraction for overlap detection in documents. In *Proceedings of the twenty-fifth Australasian conference on computer science: Vol. 4*.

Rodríguez-Torrejón, D., & Martín-Ramos, J. (2010). CoReMo system (contextual reference monotony) a fast, low cost and high performance plagiarism analyzer system: Lab report for PAN at CLEF 2010. In *Notebook Papers of CLEF*.

Sánchez-Vega, F., Villatoro-Tello, E., Montes-y-Gomez, M., Villaseñor-Pineda, L., & Rosso, P. (2013). Determining and characterizing the reused text for plagiarism detection. *Expert Systems with Applications, 40*, 1804–1813.

Sarkar, A., Marjit, U., & Biswas, U. (2014). A conceptual model to develop an advanced plagiarism checking tool based on semantic matching. In *Proceedings of the 2nd International Conference on Business and Information Management (ICBIM), 2014* (pp. 104–108). IEEE.

Schleimer, S., Wilkerson, D. S., & Aiken, A. (2003). Winnowing: Local algorithms for document fingerprinting. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data* (pp. 76–85). ACM.

Soleman, S., & Purwarianti, A. (2014). Experiments on the Indonesian plagiarism detection using latent semantic analysis. In *Proceedings of the 2nd international conference on information and communication technology (ICoICT), 2014* (pp. 413–418). IEEE.

Stamatatos, E. (2011). Plagiarism detection using stopword n-grams. *Journal of the American Society for Information Science and Technology, 62*, 2512–2527.

Stein, B., & Zu Eissen, S. M. (2006). Near similarity search and plagiarism analysis. *From data and information analysis to knowledge engineering* (pp. 430–437). Springer.

Suchomel, Š., Kasprzak, J., & Brandejs, M. Three way search engine queries with multi-feature document comparison for plagiarism detection—notebook for PAN at CLEF 2012. Forner et al (Eds.) ISBN, 978–988.

Tian, Y., Li, H., Cai, Q., & Zhao, S. (2010). Measuring the similarity of short texts by word similarity and tree kernels. In *Proceedings of the IEEE youth conference on information computing and telecommunications (YC-ICT), 2010* (pp. 363–366). IEEE.

Toman, M., Tesar, R., & Jezek, K. (2006). Influence of word normalization on text classification. *Proceedings of InSciT, 4*, 354–358.

Tomasic, A., & Garcia-Molina, H. (1993). Query processing and inverted indices in shared: Nothing text document information retrieval systems. *The VLDB Journal—The International Journal on Very Large Data Bases, 2*, 243–276.

van Rijsbergen, C. J. (1986). (invited paper) A new theoretical framework for information retrieval. In *Proceedings of the 9th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 194–200). ACM.

Wang, S., Qi, H., Kong, L., & Nu, C. (2013). Combination of VSM and Jaccard coefficient for external plagiarism detection. In *Proceedings of the international conference on machine learning and cybernetics (ICMLC), 2013 : Vol. 4* (pp. 1880–1885). IEEE.

Warin, M. (2004). Using wordnet and semantic similarity to disambiguate an ontology. *Retrieved January, 25*, 2008.