Cross-Language Plagiarism Detection using Word Embedding and Inverse Document Frequency (IDF)

Hanan Aljuaid
Computer Sciences Department, College of Computer and Information Sciences
Princess Nourah bint Abdulrahman University (PNU), 84428 Saudi Arabia, Riyadh

***Abstract*—The purpose of cross-language textual similarity detection is to approximate the similarity of two textual units in different languages. This paper embeds the distributed representation of words in cross-language textual similarity detection using word embedding and IDF. The paper introduces a novel cross-language plagiarism detection approach constructed with the distributed representation of words in sentences. To improve the textual similarity of the approach, a novel method is used called CL-CTS-CBOW. Consequently, adding the syntax feature to the approach is improved by a novel method called CL-WES. Afterward, the approach is improved by the IDF weighting method. The corpora used in this study are four Arabic-English corpora, specifically books, Wikipedia, EAPCOUNT, and MultiUN, which have more than 10,017,106 sentences and uses with supported parallel and comparable assemblages. The proposed method in this paper combines different methods to confirm their complementarity. In the experiment, the proposed system obtains 88% English-Arabic similarity detection at the word level and 82.75% at the sentence level with various corpora.**

***Keywords*—NLP; cross-language plagiarism detection; word embedding; similarity detection; IDF**

I. INTRODUCTION

Plagiarism is a major problem today. Cross-lingual plagiarism (CLP) is a type of plagiarism that occurs when texts are translated from one language to another without citing the original sources. Monolingual plagiarism analysis, which detects plagiarism in documents written in the same language, has been executed by many researchers, but CLP remains a challenge. Earlier studies have used approaches such as cross-lingual explicit semantic analysis (CL-ESA), syntactic alignment using character N-grams (CL-CNG), dictionaries and thesauruses, statistical machine translation, online machine translators [1] [6], and more recently, semantic networks and word embedding [7]. However, these approaches are specific to bilingual plagiarism detection tasks and are normally not sufficient for limited resource languages.

Conversely, word embedding is a significant representation theory used to represent sentence units used in natural language processing (NLP) applications [15]. This process depends on the low-dimensional vector representation of words, and it can easily measure the syntax vs. semantic relationship. Currently, a variety of NLP applications are contingent on two-word embedding models: the word2vec model [12] and the GloVe model [17]. The word2vec model is a neural network that includes three layers: one input layer, one output layer and one hidden layer. However, the GloVe word embedding model uses a global vector for word representation [21].

In this paper, we explore the performance of the distributed representation of word embedding to propose novel cross-lingual similarity procedures for similarity detection. We use word embeddings with the IDF weighting method.

II. RELATED WORK

Word embedding is used in natural language processing as a representation of the vocabulary of a document. This method depends on identifying the context of a word (syntactic and semantic similarities) relative to other words using vector representation and involves two models: the word2vec and GloVe models. Recently, these two-word embeddings models have been used in various natural language processing applications [21].

However, this processing starts by converting words into vectors. Consequently, the cosine similarity is used to measure the semantic similarity between two words [13]. The previous method for representing a word vector was a “one-hot” representation, where the number of dimensions of each vector is matched to the number of dimensions of the vocabulary. Modern word embeddings are accessible for the study of semantic and syntax similarities.

Word2vec is one type of neural network with three layers: an input layer, hidden layer, and output layer. The number of dimensions of the vector that represents a word is the same as the number of neurons in the hidden layer. Typically, the word2vec model applies big datasets in the training phase to optimize the syntax and semantics correctly. Word2vec mathematically detects similarities to cluster the vectors of similar words together in vector space. The created vectors detect the word features by distributed arithmetic representations without human mediation. Additionally, using the given data, word2vec can determine highly accurate solutions about a word’s meaning based on past sentences. Those solutions can be used to launch a word’s connection with other words or cluster documents and classify them by topic (for example, “man” is to “boy”, and “woman” is to “girl”). In addition, those clusters can be used in a sentiment analysis, where each item in the vocabulary has a vector attached to it and can be fed into a deep-learning networked or analysed to discover the relations between words.

The main approaches of word2vec are the skip-gram model and the bag-of-words model (BOW), and both of these models have achieved developments in computational cost and accuracy. In these two approaches, the same hyperparameters are used, such as the window size denoted by C and the vocabulary size (represents the number of words in the corpus) denoted by |v|. In the next paragraph, these two approaches are explained briefly.

Conversely, the continuous bag-of-words technique (CBOW) inputs the context of each word using a linear classifier and predicts the middle word corresponding to the adjacent features in that context [10][21]. The deeper analysis of CBOW can show that the input words comprise a one-hot encoded CxV dimension matrix of the context words, and the output layer comprises a vector with the elements being the softmax values of V length; the hidden layer contains N neurons and takes an average over all the C context input words, as shown in Fig. 1.

The continuous skip-gram approach or skip-gram technique (the second approach of the word2vec model) is very similar to the CBOW model. However, the difference between the two approaches exists in the input and output layers. The input in CBOW is the context words, and the output is the middle word, whereas the opposite occurs in the skip-gram model, where the input is the present word, and the output is the context words.

Fig. 2 shows that the skip-gram model has three layers. The input layer includes the input vector with length V for only one word. The hidden layer has the same definition as it does in the CBOW model, where h in formula (1) denotes the relationship between the input and hidden layers, i.e., h is simply transposed onto a row with two layers with weight matrix, W, which is supplementary to the input word wI:

h=WT:=vT, (1) (k,·) wI (1)

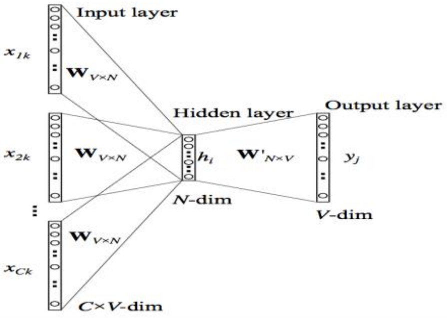

Fig. 1. CBOW Model Architecture [19]; [10].

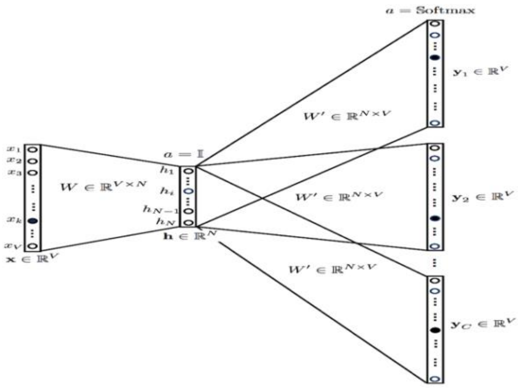
$a = \text{Softmax}$
$\mathbf{y}_1 \in \mathbb{R}^V$
$W' \in \mathbb{R}^{N\times V}$
x_1
x_2
x_3
$a = \mathbb{I}$
h_1
$W \in \mathbb{R}^{V\times N}$
h_i
$W' \in \mathbb{R}^{N\times V}$
$\mathbf{y}_2 \in \mathbb{R}^V$
h_{N-1}
x_k
h_N
x_V
$\mathbf{h} \in \mathbb{R}^N$
$\mathbf{x} \in \mathbb{R}^V$
$W' \in \mathbb{R}^{N\times V}$
$\mathbf{y}_C \in \mathbb{R}^V$

Fig. 2. Skip-Gram Model Architecture [19].

For the output layer of the model outputting C probability distributions, each context position has C probability distributions with V probabilities (one for each word) [19].

The skip-gram model is efficient when training small datasets with irregular words. However, the CBOW model is proficient when used with common words [15]. Moreover, the considerable challenge with both word2vec representations is learning the output vectors. To appropriately learn the output vectors, the proposed hierarchical softmax and negative sampling algorithms can be used [13]. The first algorithm (hierarchical softmax) is centred on the Huffman tree (a binary tree), which uses word frequencies to estimate the words in a tree. Then, the algorithm uses normalization in each step from the root to the target word [15]. The second algorithm, negative sampling, targets the noise distribution to update the samples of the output vectors. Correspondingly, negative sampling is used in the case of low-dimension vectors with more common words, whereas hierarchical softmax is used in the case of irregular words.

III. Preprocessing Management

A. Dataset

The dataset used throughout our study is the new dataset familiarized by Aljuaid [2]. The characteristics of this dataset are as follows:

- written in English and Arabic;
- united at different levels (the document, sentence, and word chunk levels);
- uses supported parallel and comparable assemblages;
- conceals several subjects;
- translates automatically or by humans, regardless of whether the translations are performed by professionals;
- collected from more than 3,000 random documents that were checked manually.

Table I shows the details of the dataset and presents the number of aligned units. Table II presents the different characteristics of the dataset within each corpus.

B. Outline of State-of-the-Art Methods

Cross-language plagiarism estimates the textual similarity between two languages in two textual units. In this section, the state-of-the-art methods that are used in this paper are discussed.

TABLE. I. Corpora Description of our Dataset

Corpus	Language	#document	# sentences	# word chunks
Books	English/ Arabic	≈ 6,000	≈ 120,000	≈ 720,000,0
Wikipedia	English/ Arabic	≈ 10,000	≈ 800,000	≈ 480,000,00
EAPCOUNT	English/ Arabic	≈341	≈ 53,000	≈ 5,392,491
MultiUN	English/ Arabic	≈1659	≈1,124,609	≈ 300,000,000

TABLE. II. CORPORA CHARACTERISTICS OF OUR DATASET

corpus	Alignment	Written by	Translated by
Books	Parallel	Computer scientists	Professional translators
Wikipedia	Comparable	Anyone	Student translators
EAPCOUNT	Parallel	Politicians	Machine translated
MultiUN	Parallel	Politicians	Machine translated

Cross-language character n-gram (CL-CnG) is dependent on the comparison of dual textual units according to their n-gram vectors based on the [11].

Cross-language conceptual thesaurus-based similarity (CL-CTS) is used to extract the roots of the textual units to measure the semantics of the words [16].

Cross-language alignment-based similarity analysis (CL-ASA) is used as a bilingual unigram dictionary to determine the ability of one textual unit to translate to another textual unit and their probabilities extracted from a parallel corpus [18].

Cross-language explicit semantic analysis (CL-ESA) denotes the meaning of a document by a vector based on concepts derived from Wikipedia according to the explicit semantic analysis [8].

Translation + monolingual analysis (T+MA) involves translating elements in two different languages into the same language to perform monolingual identification among the elements [3]. This state-of-the-art method is discussed in depth in our previous paper [2].

IV. PROPOSED METHODS

A. *Model used*

The word embedding representation is achieved and is compatible with the corpus context. Words with similar contexts should be projected onto a continuous multidimensional space. However, word embedding can be used to detect and calculate similarities between sentences in the same or different languages.

Consequently, we used the word2vec CBOW approach toolkit offered by MultiVec [4]. To build and train the vectors, we use the large collection corpus discussed in [2].

To train the CBOW embedding system, some parameters are selected to affect the resulting vectors. The selected parameter has a vector size of 100 with a window size of 5, and a number of negative examples in training 10 are shown in Table III.

TABLE. III. THE ARABIC CBOW MODEL PARAMETERS FOR TRAINING THE CONFIGURATION PARAMETERS

Parameter	Significance
Window	5
Vector size	100
Negative	10
Sample	1e − 5
Frequency threshold	0.02

B. *Textual Similarity*

We introduce a new method to identify the similarity among textual words. However, the lexical resource in the cross-language conceptual thesaurus-based similarity (CL-CTS) is replaced with the distributed representation of words. To construct the words with the BOW model, we used the CBOW model to detect pairs of two words, wi and wj. Each word is represented by vectors vi and vj, respectively. The similarity between wi and wj is obtained by comparing their vectors vi and vj that were evaluated using cosine similarity. We call this new implementation CL-CTS-CBOW, and this method is used to improve textual similarity.

Then, we implement a method that performs a comparison between two sentences S and S' in different languages. We call this method CL-WES, which uses the cosine similarity of the embedded vectors of all units among the sentences to represent the distribution of the sentences [6], where S' = w1,w2...,wi and S" = w1′ , w2′ ,...,wj′ , with two textual units U' and U" in two different languages. Then, CL-WES builds the bilingual corpus of the two different languages. The two representation vectors V' and V" utilize cosine similarity.

The calculation of the distributed representation V around a textual unit U is:

V =∑_(i=1)^n▒(ui) (2)

where V is the vector of the function that gives the word embedding, and ui is the textual unit. Fig. 3 shows our proposed system.

C. *Syntax Similarity*

In this section, the CL-WES model is improved by adding the syntax aspect, as discussed in Section 4.2, where U is a textual unit with n words, as shown in formula (1). However, we start by applying the part of speech tagger (POS) to syntactically tag U, which is used to weight every word in the sentence representation, classifying it into its morphosyntactic category. Then, we normalize the tags using the universal tagset [20]. Then, a weight is assigned to each tag according to this formula:

$$\mathrm{V} = \sum_{k=1}^{i} \mathrm{Pos\ weight(Poswk)} * \mathrm{vk} \qquad (3)$$

where Poswk is the function used to determine the weight of the POS tagging of wk [14].

Moreover, if U_1 and U_2 are two textual units with different languages, their representation vectors V_1 and V_2 are built using formula (4); then, cosine similarity is applied between them.

$$\mathrm{V} = \sum_{i=1}^{n} \mathrm{(weight(pos(ui)).\,vector(ui))} \qquad (4)$$

where the variable weight is a function that determines the weight of a POS, and the variable vector is a function that outputs the word embedding vector.

D. *Combining Multiple Methods*

To improve our method's performance in detecting cross-language similarity in English and Arabic languages, we combine our method with the IDF weighting method, where during weight processing, the similarity score of each method is assigned, and the composite score is calculated (weighted), as shown in Fig. 3. The distribution of the weights is optimized with the Bersini method[5]. However, one fold of every corpus is used to train the IDF weights, so the other evaluates the IDF method.

1) IDF weighting method: The IDF method constructs a compound weight of every word in a sentence. The IDF weight operates as a measurement term related to the absolute similarity between documents.

However, the Salton and [9] method is employed, where one fold of each corpus is used as an input to be semantically verified. To compute the *IDF weight* for every word, the other folds in the corpus are used as the background quantity. Moreover, the IDF is calculated with the following formula:

$$\text{idf}(w)=\log\left(\frac{s}{ws}\right) \tag{5}$$

where S is the number of sentences in the corpus written in the two languages of Arabic and English, and WS is the number of sentences containing *w*. Then, the cosine similarity between V1 and V2, cos(V1, V2), in L_1 and L_2 is calculated to obtain the similarity between S1 and S2:

$$\begin{cases} V_1 \sum_{k=1}^{i} idf(w_k)v_k \\ V_2 \sum_{k=1}^{i} idf(w'_k)v'_k \end{cases} \tag{6}$$

where *idf (wk)* is the weight of w*k* in the background.

Regarding the state-of-the-art methods for clustering capacity, the similar and different terms are correctly separated, and their ability to predict a (mis)match is determined. We combine these methods with IDF weighting to reduce uncertainties in the classification and exploit the complementarities of these methods. However, we find that these methods are processed differently according to their features. Some of them are lexical syntax-based, others are semantic-based and process the aligned words, and others capture the context with word vectors.

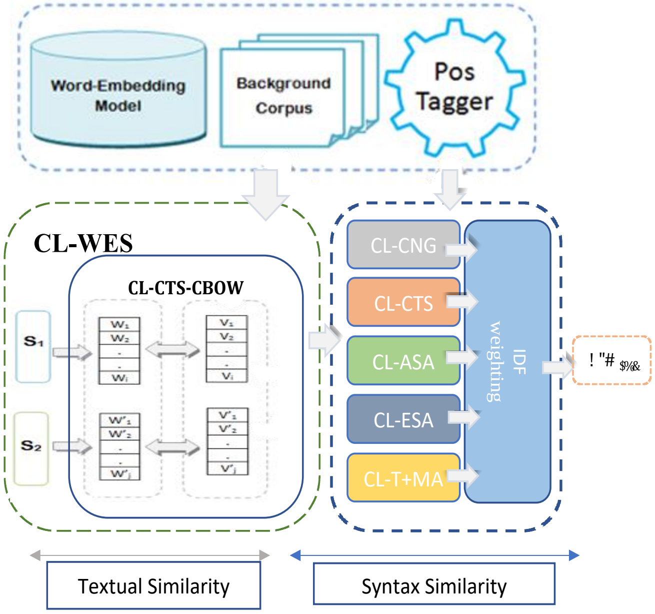

Fig. 3. The Proposed System Architecture.

V. Experiments and Results

A. Evaluation Indicators

To evaluate our method, a distance matrix of size NxM is built, where M=1,000 and N is the evaluated sub-corpus we previously denoted as (S). However, to operate S, every textual unit is matched with its consistent units in the intentioned language (i.e., to detect the similarity in the cross-lingual analysis); in addition, it is compared to M-1, which is a unit randomly selected from S. In the comparison, each obtained matching score leads to the distance matrix. To identify the threshold of the matrix, the best F-score is used and defined as the symmetrical mean of precision and recall, where precision is the number of matches in similar units that is retrieved using all of the matches. All of the methods are applied to the Arabic-English corpus at the word and sentence levels. In every construction, a particular method is applied to the sup-corpus for training and evaluation when considering a particular level. The evaluation folds are supported by varying the M selected units. The formulas for calculating the F-score, precision and recall are shown in formulas (7) -(9), respectively.

$$\text{precision} = \frac{TP}{TP+FP} \tag{7}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{8}$$

$$F = \frac{2\times\text{precision}\times\text{Recall}}{\text{precision}\times\text{Recall}} \tag{9}$$

where *TP* is the number of samples with positive similarity. *TN* is the number of samples with negative similarity. *FP* is the number of samples that have a negative similarity tagged as a positive similarity, and *FN* is the number of samples that have a positive similarity tagged as a negative similarity.

1) Use of word embedding evaluation: The F-score, which presents the distributed representation of words compared with lexical resources, improves the *CL-CTS-WE* performance to 78% at the word level, which is better than the performance of the C*L-CTS* method, which obtains a 59% performance at the word level and 54% performance at the sentence level, as shown in Table IV. However, the use of *CL-WES* improves the performance at the word level to 86%, which is higher than the state-of-the-art method performances, as shown in Fig. 4. Focusing on the state-of-the-art methods, we found that the best performance is from the CL-ASA method at the word and sentence levels, but the overall performance of the method is lower than the CL-WES performance, which is the best single method evaluated.

2) IDF evaluation: The results of the IDF method are recorded at both the word and sentence levels in Table IV and Fig. 5. In each case, we combine five state-of-the-art approaches and the proposed novel approach. The IDF weighting method is better than the state-of-the-art approaches and the embedding-based approaches at all levels. At the word level, the IDF method has an F-score of 88%. However, the best single method achieves an F-score of 86.5%. At the sentence level, the IDF method also obtains a trend of 82.75 against the CL-WES trend (81.5), which was recorded as the best single method. The results obtained in Table IV confirm that the altered approaches proposed experience enhanced performance. Additionally, the obtained results in Table IV indicate that the embeddings are practical for Arabic-English cross-language similarity detection.

Finally, the performances of the methods indicate their capabilities with the dataset. In Fig. 6, we find that the precision improved by 1.54% in the Wikipedia and MultiUN corpora; the recall increased to 1.23%, and the F-score also increased by 2.05 in the Wikipedia and MultiUN corpus. By combining the performances of each method for the dataset, we find that the effect of the IDF method is better than that of the state-of-the-art methods, as discussed previously.

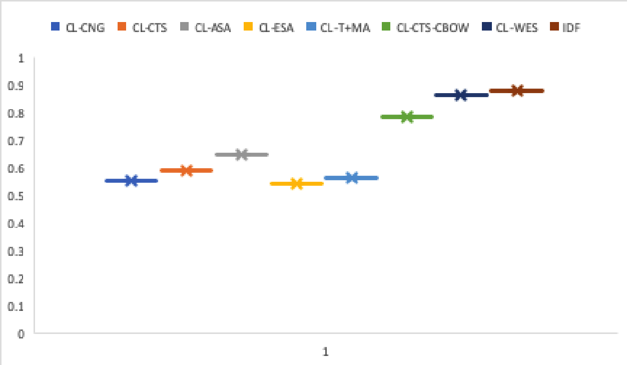

Fig. 4. Comparison of State-of-the-Art Method Performances and the Proposed Method Performance.

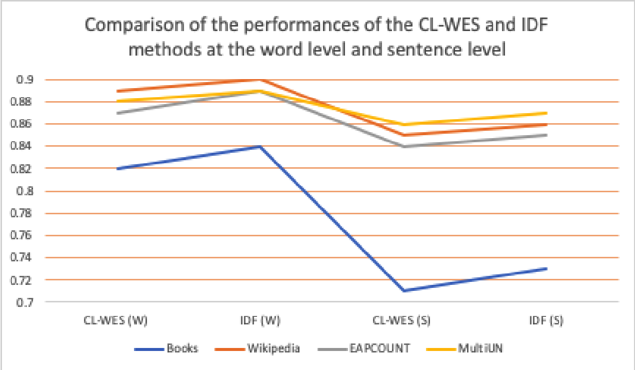

Fig. 5. Comparison of the Performances of the CL-WES and IDF Methods at the Word Level and Sentence Level.

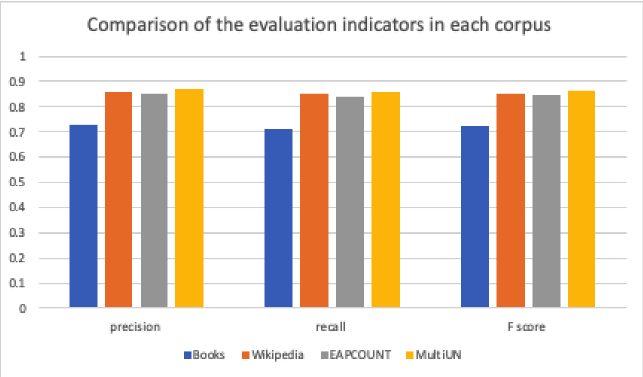

Fig. 6. Comparison of the Evaluation Indicators in each Corpus.

TABLE. IV. THE PERFORMANCES OF CROSS-LANGUAGE SIMILARITY DETECTION METHODS ON ARABIC-ENGLISH CORPORA

Word level					
Methods	***Books (%)***	***Wikipedia (%)***	***EAPCOUNT(%)***	***MultiUN (%)***	***Overall (%)***
CL-CNG	***0.44***	***0.61***	***0.58***	***0.57***	***0.55***
CL-CTS	*0.58*	*0.65*	*0.57*	*0.56*	*0.59*
CL-ASA	***0.56***	***0.74***	***0.66***	***0.63***	***0.6475***
CL-ESA	***0.47***	***0.57***	***0.53***	***0.60***	***0.5425***
CL-T+MA	***0.54***	***0.59***	***0.54***	***0.58***	***0.5625***
CL-CTS-CBOW	***0.75***	***0.80***	***0.79***	***0.80***	***0.785***
CL-WES	***0.82***	***0.89***	***0.87***	***0.88***	***0.865***
IDF	***0.84***	***0.90***	***0.89***	***0.89***	***0.88***
Sentence level					
Methods	***Books (%)***	***Wikipedia (%)***	***EAPCOUNT(%)***	***MultiUN (%)***	***Overall (%)***
CL-CNG	***0.44***	***0.61***	***0.58***	***0.57***	***0.55***
CL-CTS	***0.48***	***0.55***	***0.57***	***0.56***	***0.54***
CL-ASA	***0.54***	***0.67***	***0.64***	***0.65***	***0.625***
CL-ESA	***0.51***	***0.53***	***0.65***	***0.66***	***0.5875***
CL-T+MA	***0.56***	***0.59***	***0.54***	***0.58***	***0.5675***
CL-WES	***0.71***	***0.85***	***0.84***	***0.86***	***0.815***
IDF	***0.73***	***0.86***	***0.85***	***0.87***	***0.8275***

VI. CONCLUSION AND FUTURE WORK

A novel approach for a word embedding-based system is presented in this paper to measure similarities in two cross-linguistic plagiarism. This method could be used for different cross-language similarities and in the training and evaluation phases applied in the Arabic-English corpus as a special case. The proposed methodology improves upon a syntactically weighted distribution representation that operates using the cosine similarity of imbedded vectors (*CL-WES*). The CL-WES model dominates all of the top state-of-the-art methods. Conclusively, the outcomes achieved from the proposed system confirmed that all methods are complementary and that their IDF weights are beneficial to the performance of cross-language textual similarity detection. The IDF method indicates an overall F-score of 88% at the word level; however, the CL-WES method obtains an 86.5% F-score at the word level, whereas the best single method obtains an F-score of only 64.75%. Additionally, at the sentence level, the methods show the same trends.

Our future work will be to improve the *CL-WES* method by exploring the syntactic and semantic weights according to the plagiarist's stylometry. Additionally, a smart hybridization between both IDF weighting and POS tagging procedures will be applied to improve the results.

VII. Funding

This research was funded by the Deanship of Scientific Research at Princess Nourah bint Abdulrahman University through the Fast-track Research Funding Programme.

References

[1] Al-Suhaiqi M, Hazaa MAS, Albared M (2018) Arabic english cross-lingual plagiarism detection based on keyphrases extraction, monolingual and machine learning approach. Asian J Res Comput Sci 2:1-12. https://doi.org/10.9734/ajrcos/2018/v2i330075.

[2] Aljuaid H. (2020) Arabic-English corpus for cross-language textual similarity detection. In: 10th International Conference on Information Science and Applications, ICISA 2019; Seoul; South Korea; 16 December 2019 through 18 December 2019; Information Science and Applications, Lecture Notes in Electrical Engineering, Springer Nature, Volume 621, 2020, Pages 527-536.

[3] Barron-Cedeno A (2012) On the mono- and cross-language detection of text re-use and plagiarism. PhD thesis, Universitat Politenica de Velenica, Span.

[4] Berard A, Servan C, Pietquin O, and Besacier L. (2016.). MultiVec: a Multilin- gual and Multilevel Representation Learning Toolkit for NLP. . In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). Portoroz, Slovenia,: European Language Resources Association (ELRA).

[5] Berghen FV, Bersini H (2005) CONDOR, a new parallel, constrained extension of powell's UOBYQA algorithm: experimental results and comparison with the DFO algorithm. J Comput Appl Mathemat 181:157-175. https://doi.org/10.1016/j.cam.2004.11.029.

[6] Ferrero J, Agnès F, Besacier L, Schwab D (2017) CompiLIG at semeval-2017 Task 1: cross-language plagiarism detection methods for semantic textual similarity. arxiv preprint arxiv:1704.01346.

[7] Franco-Salvador M, Rosso P, Montes-Y-Gómez M (2016) A systematic study of knowledge graph analysis for cross-language plagiarism detection. Inf Process Manag 52:550-570. https://doi.org/10.1016 /j.ipm.2015.12.004.

[8] Gabrilovich E, Markovitch S (2007) Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: Proceedings of the 20th international joint conference on artifical intelligence (IJCAI'07), Hyderabad, India, pp 1606–1611.

[9] Gerard Salton and Christopher Buckley. 1988. Term- weighting approaches in automatic text retrieval. In- formation processing & management, 24(5):513– 523.

[10] Karani D (2018) Towards data science. https://towardsdatascience. com/introduction-to-word-embedding-and-word2vec-652d0c2060fa.

[11] McNamee P, Mayfield J (2004) Character N-gram tokenization for european language text retrieval. Inf Retri 7:73-97. https://doi.org/ 10.1023/B:INRT.0000009441.78971.be.

[12] Mikolov T, Chen K, Corrado G, Dean J (2013a) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[13] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013b) Distributed representations of words and phrases and their compositionality. Adv Neural Inform Process Syst 26:9.

[14] Nagoudi ES (2017) Semantic similarity of arabic sentences with word embeddings. In: Proceedings of the third Arabic natural language processing workshop. Association for Computational Linguistics, Valencia, Spain, pp 18–24.

[15] Naili M, Chaibi AH, Ben Ghezala HH (2017) Comparative study of word embedding methods in topic segmentation. Proced Comput Sci 112:340-349. https://doi.org/10.1016/j.procs.2017.08.009.

[16] Pataki M (2012) New approach for searching translated plagiarism. In: Proceedings of the 5th international plagiarism conference, Newcastle, UK.

[17] Pennington J, Socher R, Manning C (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pp 1532-1543.

[18] Pinto D, Civera J, Barrón-Cedeño A, Juan A, Rosso P (2009) A statistical approach to crosslingual natural language tasks. J Algorithms 64:51-60. https://doi.org/10.1016/j.jalgor.2009.02.005.

[19] Rong X (2016) word2vec parameter learning explained. arXiv preprint arXiv:1411.2738.

[20] Slav P, Dipanjan D, Ryan M (2012) A universal part-of-speech tagset. In: Proceedings of the eight international conference on language resources and evaluation (LREC'12). European Language Resources Association (ELRA), Istanbul, Turkey, pp 2089-2096.

[21] Suleiman D, Awajan A (2018) Comparative study of word embeddings models and their usage in Arabic language applications. In: The 19th internationnal Arab conference on information technology – ACIT 2018. IEEE, Werdanye, Lebanon, Lebanon, pp. 0857-1812