Word Embedding for High Performance Cross-Language Plagiarism Detection Techniques

https://doi.org/10.3991/ijim.v17i10.38891

Chaimaa Bouaine(✉), Faouzia Benabbou, Imane Sadgali
Laboratory of Modeling and Information Technology, University Hassan II, Casablanca, Morocco
chaimaa.bouaine-etu@etu.univh2c.ma

Abstract—Academic plagiarism has become a serious concern as it leads to the retardation of scientific progress and violation of intellectual property. In this context, we make a study aiming at the detection of cross-linguistic plagiarism based on Natural language Preprocessing (NLP), Embedding Techniques, and Deep Learning. Many systems have been developed to tackle this problem, and many rely on machine learning and deep learning methods. In this paper, we propose Cross-language Plagiarism Detection (CL-PD) method based on Doc2Vec embedding techniques and a Siamese Long Short-Term Memory (SLSTM) model. Embedding techniques help capture the text's contextual meaning and improve the CL-PD system's performance. To show the effectiveness of our method, we conducted a comparative study with other techniques such as GloVe, FastText, BERT, and Sen2Vec on a dataset combining PAN11, JRC-Acquis, Europarl, and Wikipedia. The experiments for the Spanish-English language pair show that Doc2Vec+SLSTM achieve the best results compared to other relevant models, with an accuracy of 99.81%, a precision of 99.75%, a recall of 99.88%, an f-score of 99.70%, and a very small loss in the test phase.

Keywords—plagiarism, cross-language, FastText, Word2Vec, Doc2Vec, GloVe, Sen2Vec, BERT, SLSTM

1 Introduction

The development and democratization of the Internet network have enabled several benefits, namely: huge free resources, exchange of ideas, access to innovative technologies, enrichment of knowledge, etc. Besides these advantages, some challenges emerged due to the misuse of resources, and the big one is plagiarism, especially in the academic sector. Plagiarism is defined as the partial or total reuse of the work of someone else without citing the original work. Plagiarism can be applied to various contents such as text, ideas, painting, music, code, etc. The most common forms of plagiarism [1] include copying without attribution, resubmitting an entire work under a different author's name, using translation, and copying more than 100 words of an original work without citing it. Using original works is essential for advancement in

any domain, as long as the authors cite the origin of the idea or work [2]. The campaign against plagiarism in the academic domain has become a crucial way to contribute to scientific development and to conduct an honest competition between researchers. Many types of research were devoted to surveying academic plagiarism factors [3], [4] detection systems [5], and punishment measures [6]. Indeed, current educational systems do not pay enough attention to teaching students to respect copyright and preserve original works. However, some universities have started to consider the plagiarism problem and require students to check their reports, dissertations, and thesis with online tools [7] such as iThenticate [8], Urkund [9], Plagscan [10], etc.

Academic plagiarism is conducted with different methods such as: copy and paste, translation from other languages, translation and back translation, manipulation of the text with paraphrasing, style modification, self-plagiarism, plagiarism of ideas, or a combination of them [11], [12].

The efforts of researchers have resulted in several proposals for plagiarism detection that can be classified into intrinsic and extrinsic plagiarism detection methods. Extrinsic methods [13] are based on the correlation between the suspicious text and a collection of candidate texts. The intrinsic plagiarism detection methods [14], [15] exploit the data inside the text in focus, such as the form, language, style, symbols, images, contrasts, and structure. The intrinsic approach is also known as "formalism" because it is primarily concerned with the form of the text [16]. Lexical and semantic techniques are generally recommended for detecting plagiarism. The lexical approach focuses on using the lexical features of the documents, which act at the word level of the text, to identify the plagiarism scenarios in the document.

This approach attempts to enhance standard string matching for plagiarism detection [17]. The semantic approach focuses on the meaning of words, sentences, or texts by finely analyzing the word combinations and their local and global context. The latter usually relies on the capacity of the Word Embedding (WE) techniques to represent the units (word, sentence, paragraph, etc.) of documents while saving their contexts. Word embedding techniques, in fact, seek to represent a text using real-number vectors in a predefined vector space. These new representations of textual data have enhanced the performance of Natural Language Processing (NLP) techniques like topic modeling and sentiment analysis [18]. The present paper focuses on cross-lingual plagiarism detection (CLPD) with extrinsic methods. Cross-lingual plagiarism might include different types of plagiarism, from copying and pasting and paraphrasing to plagiarism of ideas, where the text may be quite different but the ideas it depicts are copied from other works published in other languages.

In this paper, we propose a method for CLPD based on Doc2Vec and SLSTM. For this aim, we conducted a state-of-the-art analysis of inter-language plagiarism detection methods based on WE techniques. In order to evaluate the performance of our method, we combined the PAN11, JRC_AQCUIS, Europarl, and Wikipedia (Spanish-English) datasets in order to have enough training data. Using this dataset, we compared the performance results of Word2Vec, FastText, Doc2Vec, Word2Vec+Sen2Vec, and BERT methods, as they were the most used in literature reviews.

The rest of this document is organized as follows: The next section presents the state of the art in CLPD techniques. In Section 3, we describe our proposal and methodology.

Section 4 is devoted to the experiments and results. Finally, we conclude the main results and discuss avenues for future work.

2 Related work

Numerous methods have been developed in an effort to identify various forms of plagiarism, such as paraphrasing, citation, cross-language (CL), and monolingual plagiarism, as a result of research into plagiarism detection strategies. These methods frequently rely on several NLP preprocessing techniques, such as the elimination of stop words, tokenization, normalization, lemmatization, and stemming, to prepare the data. The data must be cleaned and prepared in order for machine learning techniques to use it for training. In this section, we will give a study of research publications that use WE approaches to address the issue of CL-PD and suggest a comparison to show the benefits and drawbacks of each strategy.

Aljuaid et al. [19] addressed the problem of CL-PD for English-Arabic languages using WE and Inverse Document Frequency (IDF) techniques. The approach used semantic and syntactic methods and treated word and sentence levels. A new bag of words was proposed, called CL Conceptual Thesaurus-based Similarity Continuous Bag-of-Words (CL-CTS-CBOW), for word level, and another method called CL Word Embeddings Similarity (WES) based on the cosine similarity for sentence level. The proposed system achieved an F-score of 88% for English-Arabic similarity detection at the word level and an F-score of 82.75% at the sentence level based on different corpora. Nguyen et al. [20] used Siamese recurrent architectures to define instances of Vietnamese-English CL paraphrases. English and Vietnamese sentences were preprocessed using the Part of Speech (POS) tag revision method to update the POS of English and Vietnamese sentences. A parallel Long Short-Term Memory Model (LSTM) was used to measure the similarity of two sentences and to identify the paraphrase instances encoded by the Word2Vec method. The experimental results on the English-Vietnamese paraphrase corpus achieved an accuracy of 89.61%. Glavaš et al. [21] proposed a Low-Resource CL Semantic Textual Similarity (STS) based on the measure of the semantic similarity between texts in various languages. In this approach, the Word2Vec model was applied for each language, and used a Linear Translation Matrix (LTM) model to project vectors from the source language into the embedding space of the target language. The performance of the proposed system was mainly measured with three different STS datasets including three language pairs: English-Spanish, English-Italian, and English-Croatian. The results of the evaluation indicate that CL-STS exhibits concurrent performance and stability for various language pairs, including Croatian as an underserved language pair. Mahmoud et al. [22] proposed a CL-PD system for Arabic-English languages using the Sent2Vec technique and CNN algorithm. For the preprocessing stage, they used the removal of irrelevant data, normalization to reduce ambiguities, annotation of words by their grammatical classes, and tokenization. Three layers were presented: 1) a feature extraction layer; 2) a max-pooling layer that allows the generation of a reduced semantic vector; and 3) a comparison layer to evaluate the similarity between sentences and convert the output

score into a probability distribution. The results showed that Sent2Vec outperformed Word2Vec and achieved a precision of 85% and a recall of 86.8%. Alotaibi et al. [23] presented a CL-PD system for Arabic-English languages based on syntactic and semantic features and using different Machine Learning (ML) classifiers. The preprocessing phase included tokenization, POS tagging, removing punctuation marks, normalization, and the use of the Word2Vec and Term Frequency-Inverse Document Frequency (TF-IDF) techniques. Multilingual Unsupervised and Supervised Embedding encoders (MUSE) were used to extract features, and various ML classifiers were tested, including logistic regression (LR), support vector classification (SVC), decision trees (DT), K-Nearest Neighbors (KNN), and extreme gradient boosting (XGBoost). Support Vector Classifier (SVC) performed with an F-score of 87.9%. Lachraf et al. [24] addressed the extrinsic and intrinsic approaches to plagiarism detection. For the extrinsic approach, they used the CL semantic similarity to produce the semantic and syntactic properties of words in two different languages, such as Arabic and English. While in the intrinsic method, they employed the Skip-Gram and CBOW embedding methods to evaluate the word translation task. Three methods of learning models were applied, namely: Parallel Mode, Word by Word Alignment Mode, and Random shuffling Mode. The random shuffle method with the skip-gram model provided the highest-performing approach with a correlation rate of 75.7%. Alzahrani et al. [25] addressed the case of Arabic-English CL plagiarism and used Deep Neural Networks (DNN), Logistic Regression (LR), and SVM models. Two tasks were performed: CL-STS with an LR model and the classification task with an SVM. Two types of classification are performed. The first one enabled the detection of the plagiarized pair of documents, and the second classification intent was to detect four types of plagiarism: IW (independently written), ST (translated and summarized), PT (translated and paraphrased), and LT (literally translated). The SVM model achieved 96.65% accuracy, the LR model 96.64% accuracy, and the DNN model 97.01% accuracy. Zubarev et al. [26] addressed the problem of CL-PD to handle the task of alignment of Russian-English CL text. Three methods for translation plagiarism detection were compared. The first one achieved a precision of 75% and was based on Sentence Embedding, neural machine translation, and various textual similarity methods. The second method is built using the fine-tuning of the pre-trained model BERT, which achieved a precision of 96%. The last one used the Language-Agnostic Sentence Representations (LASER) model and achieved a precision of 90%. Chi et al. [27] proposed English-Vietnamese CL-PD for the task of identifying paraphrases in a pair of documents. They used the Multi-Task Deep Neural Network (MTDNN) model, which is a combination of the pre-trained models BERT Multilingual (M-BERT) and CL Mode Roberta (XLM-R). Using the GLUE datasets, XLM-R provides high performance compared to M-BERT with a 9% and 6% increase in accuracy, which reached 82.8% and an F-score of 87.6%, respectively, before the fine-tuning step. After the fine-tuning step, this difference increased to 84.3% for precision and 88.5% for F-score. Nagoudi et al. [28] proposed a CL-PD system based on two WE approaches to compare the semantic text similarity of sentences in Arabic and English. The idea is to grasp the syntactic and semantic properties of the words by employing machine translation (MT) and word embedding. The weighted aligned words (WA) and Bag of

Words (BoW) WE methods were applied to assess semantic similarity. Additionally, IDF and POS weights were applied to sentences to identify the most meaningful words in each sentence. POS, mixed weights and IDF weights achieved a correlation rate of 77.39%. Al-Suhaiqi et al. [29] proposed an Arabic-English CL-PD system that combines the extraction of key phrases to compute the frequency of phrase and the list of candidate key phrase rankings. The approach used N-gram similarity, LCS, Dice Coefficient, fingerprint-based Jaccard similarity, and fingerprint-based Containment similarity. Linear logistic regression (LLR), Naive Bayes, and SVM machine-learning models were applied, and the result showed that the SVM technique achieved 92% for the F-score with the use of more than three methods of similarity computation. The study confirms that the choice of similarity calculation methods has a clear effect on the quality of the detection method. In [30], the authors proposed an English-Arabic CL-PD system based on sentence similarity. This model used two steps to represent sentence vectors. Firstly, they used the CL-WE-Tw machine translation-based method, which mixed Word2Vec, POS, and Word2Vec with TF-IDF methods. The second step is the combination of the MUSE model with the CL-WE-Tw method. The Word2Vec model combined with POS and TF-IDF weighting performed well, with a Pearson correlation of 0.69 and 0.77, respectively. Measuring the similarity of two sentence vectors with the MUSE model gives the best results, with a correlation of 0.78 for POS and 0.79 for TF-IDF. They concluded that the combination of the CL-WE-Tw and MUSE models gave more important results than using them independently. Yinhan et al. [31] proposed a Chinese-Thai CL sentence similarity calculation method based on sentence embedding. The sentence-embedding model is based on word vectors obtained using Word2Vec, and then they are summed and averaged to obtain the sentence vectors. The Chinese sentence embedding is mapped to the Thai sentence embedding space, and the Chinese-Thai CL sentence similarity is determined by using the cosine similarity. The Chinese-Tai interlingual model performed better than the machine translation and bilingual Latent Dirichlet allocation (LDA) algorithms. Stegmüller et al. [32] presented a method, CL-PD based on ontology and similarity analysis (CL-OSA). They used open knowledge graphs and covered the following language pairs: Spanish-English, Japanese-English, Japanese-Chinese, and English-French. The CL-OSA approach groups documents by topics, annotates each word with POS, and extracts entities from the Wikidata open knowledge network to represent the documents as entity vectors. The candidate documents are ranked based on their similarity score, and the relationships between entities are calculated using the cosine similarity. CL-OSA outperformed the CL-PD methods such as CL-ESA, CL-ASA, USE-ML, and ConceptNet for the five multilingual test corpora. Chang et al. [33] developed the CL Word Mover's Distance (CL-WMD) technique to deal with the English-Chinese CL-PD issue. For each language, the skip-gram method was employed to create WE spaces. The first step is to place the word space into the integration space by considering a small set of bilingual word translations and calculating the semantic distance between the texts using the word displacement distance. CL-WMD achieved a Hit score of 97.09% (Hit score is a quantitative measurement for evaluating the performance) for plagiarism detection at the paragraph level and 86.09% at the sentence level. Ferrero et al. [34] proposed a CL similarity detection method for English-French

language pairs based on WE-CBOW. The results show that CL Word Embedding-based Syntax Similarity (CL-WESS) was the most effective method. Also, all methods can be complementary, and their fusion significantly improved the performance of CL textual similarity detection. This fusion of chunk and sentence gives an F-score of 89.15% for similarity detection at the chunk level and 88.5% at the sentence level. Montes-y-Gómez et al. [35] proposed a CL-PD system that used knowledge graphs to represent text fragments as a language-independent model of their content. They made use of Word Sense Disambiguation and Distributed Concept Weighting (DCW) (WSD). By utilizing BabelNet synsets and the skip-gram model to give vector representations of concepts, DCW enables the expression of the strength of association between concepts (synset) to generate the distributed representation of contexts. WSD is designed to alleviate the problem of determining the meaning of a word since it can have several possible senses. The combination of WSD and DCW by using the skip-gram model provides important results, with a PlagDet of 66.3% for the language pair Spanish-English and 59.5% for Allemand-English. They used Dice's coefficient and cosine distance to measure the relationship between the synsets. The latter is the set of synonymous words that can be used to express the same meaning in a given language.

To have a better view of this state-of-the-art, Table 1 summarizes the results for the CLPD based on six characteristics:

- **Language Pairs (LPairs):** shows a couple of languages studied: English-Arabic (En-Ar), Vietnamese-English (Vi-En), English-Spanish (En-Sp), English-Italian (En-It), English-Croatian (En-Cr), Russian-English (Ru-En), Chinese-Thai (Ch-Th), English-French (En-Fr), Japanese-Chinese (Ja-Ch), Japanese-English (Ja-En), English-Chinese (En-Ch).
- **Preprocessing (Preproc.):** describes the preprocessing techniques applied in the approach, such as Named Entity Recognition (NER), Parts of Speech (POS), Semantic Role Labeling (SRL), Spatial Role Labeling (SpRL), and Bag of Meanings (BOM).
- **Feature Extraction:** presents the feature extraction techniques used: Word2Vec (Skip-Gram, CBOW), Glove, TF-IDF, Sent2Vec, BERT, MUSE, Transformer Encoder (TE), and Lexicon Encoder (LE).
- **Techniques (Tech.):** means Machine Learning (ML) or Deep Learning (DL) techniques such as LSTM, CNN, DNN, LR, SVM, XGBoost (XGB), etc.
- **Dataset:** describes the dataset used for the training step.
- **Performance (Perf.)** presents the performance metrics used, such as Accuracy (A), Precision (P), F-score (F), Recall (R), Correlation (C), and Hit score (H).

Table 1. Comparaison of cross language plagiarism detection approaches

Ref	LPairs	Preproc.	Feature Extraction	Tech.	Dataset	Perf. (%)
[19]	En-Ar	POS	**CLWES** IDF CBOW	–	Books Wikipedia EAPCOUNT MultiUN	**F:88** F:78 F:86
[20]	Vi-En	POS	Word2Vec	SLSTM	TED	A:89.61
[21]	En-Sp En-It En-Cr	–	Word2Vec + GloVe	–	SBW hrWaC Wikipedia	R:94.8
[22]	Ar-En	POS	**Sen2Vec** Word2Vec	CNN	OSAC	**P:85** P: 83.2
[23]	Ar-En	POS	Sen2Vec + Word2Vec + TF-IDF	**SVC** LR DT KNN RF XGB LSVC	SemEval-2017	**F:87.9** F:87.1 F:87.1 F:85.2 F:86.1 F:86.4 F:87.5
[24]	Ar-En	–	**Skip-Gram** CBOW	–	SemEval-2017	**C:75.7** C:52.8
[25]	Ar-En	POS NER SRL BOM SpRL	Word2Vec	**DNN** SVM LR	71,910 of En-Ar pairs	**A:97.01** A:96.64 A:96.65
[26]	Ru-En	POS	**BERT** LASER Sentence Embedding	LR	source multiple	**P: 96** P: 90 P: 75
[27]	En-Vi	–	**XLM-R** MBERT	–	GLUE SemEval	**A:84.3** A:73.7
[28]	Ar-En	POS	IDF	–	SemEval-2017	C:77.39
[29]	Ar-En	–	–	**SVM** NB LLR	318 files Arabic 54 files English	**F: 92** F: 88 F: 85
[30]	En-Ar	POS	Sen2Vec + TF-IDF + Word2Vec + MUSE	–	SemEval-2017	C:81.47
[31]	Ch-Th	–	Word2Vec	–	Chinese-Thai Parallel	A: 39.63
[32]	Sp-En Ja-En Ja-Ch En-Fr	POS NER	Open Knowledge Graph	–	source multiple	P: 50.6 R: 34.9
[33]	En-Ch	–	Word2Vec	–	NDLTD	H:97.09
[34]	En-Fr	–	Word2Vec	–	–	F:89.15
[35]	Sp-En	–	Skip-gram	–	PAN-11	P :76.1 R :58.8

From Table 1, we can see that the most commonly studied language pair is English-Arabic. For the preprocessing phase, some works used basic NLP techniques like stop word removal, lowercase conversion, tokenization, and lemmatization. Other works used techniques like NER, POS, SRL, BOM, and SpRL for deep processing. Different embedding techniques were investigated for feature extraction. The state-of-the-art shows that Word2Vec and TF-IDF methods are the most widely used techniques for vector representation.

Most studies, after document representation with embedding techniques, used the similarity computation to compare a couple of documents without using any classifier. In fact, models such as NB, LLR, SVC, LR, DT, KNN, RF, XGBoost, and LSVC were not always used for classification purposes. However, approaches based on a classifier have a more promising performance compared to models using just embedding techniques. For example, Word2vec+DNN outperformed with an accuracy of 97.01%, Word2Vec+SVM achieved an accuracy of 96.64%, and Word2Vec +LR performed with an accuracy of 96.65% for the textual semantic similarity task. Also, BERT contextual embedding achieved an accuracy of 96% with the LR model. The cosine is used in the majority of works to calculate the similarity of processed documents. Other functions were used, including Dice-Coefficient and Jaccard similarity. Different datasets are devoted to the detection of CL plagiarism, such as PAN11, OSAC, SemEval, Wikipedia, etc. All of them are made up of sentences and documents.

3 Research method

According to the state of the art, various performance results have been discussed to analyze the effect of the word embedding techniques on the CL-PD systems using different datasets and different language pairs. In this paper, we propose a Spanish-English CL-PD based on Doc2Vec embedding techniques and Siamese LSTM models, as depicted in Figure 1.

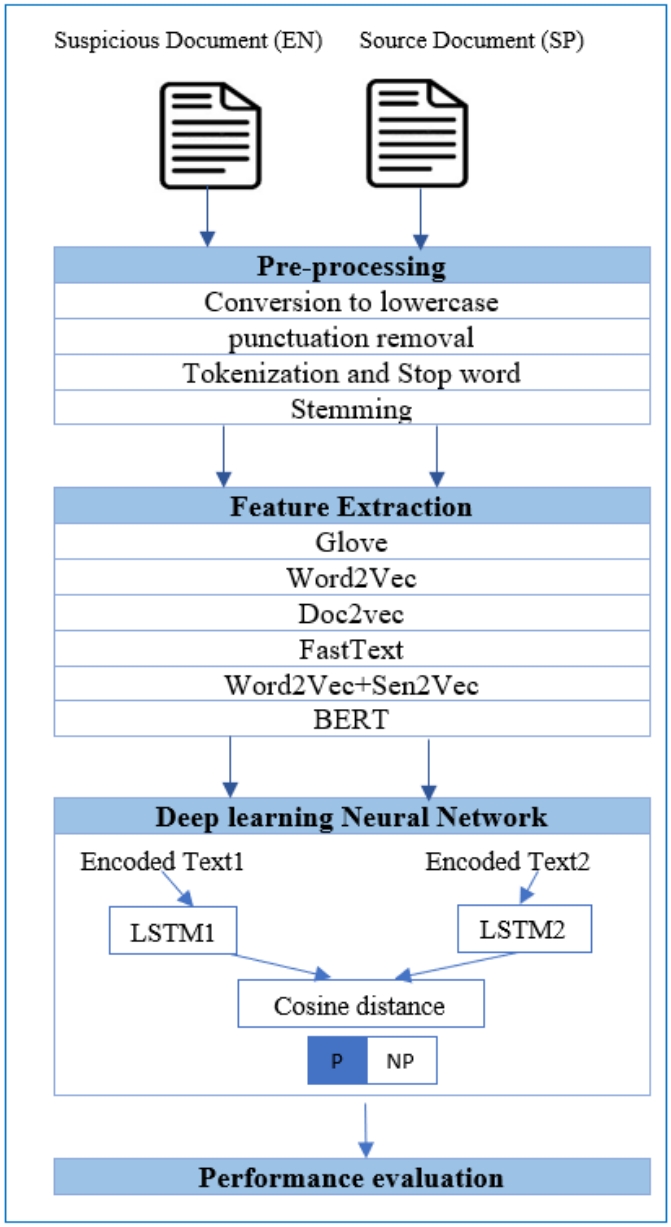

Fig. 1. The proposed methodology

3.1 Dataset collection

In order to have sufficient training data, we have gathered four datasets: PAN-PC-11, JRC-Acquis, Europarl, and Wikipedia (Spanish-English) [36, 37, 38, 39]. An

evaluation corpus for automatic plagiarism detection algorithms is the PAN 2011 (PAN-PC-11). The datasets are a collection of source documents in Spanish and their corresponding suspicious documents in English. It also contains another language pair (English-German), but for now, we focus on the SP-EN pair of languages. The JRC-Acquis parallel corpus is suitable for all types of CL research; it is excerpted from the Acquis Communautaire (AC), and 10.000 documents in SP and EN pair languages were used. The Europarl Parallel Corpus is excerpted from the proceedings of the European Parliament and includes 21 European languages. We used 9423 documents for the Spanish language. The Wikipedia parallel corpus contains multiple languages; to reach a balanced dataset, we used 9423 documents for the English language, which are very different from the 9423 Spanish documents. Hence, our dataset includes 19633 source documents and 19633 suspect documents. The PAN-PC-11 and JRC-Acquis datasets include only the plagiarized documents, and in order to add the non-plagiarized documents, we used the source documents from the Europarl dataset and suspect documents from the Wikipedia dataset and labeled them as not plagiarized. The final dataset used is described in Table 2.

Table 2. Characteristics of dataset used

DATASET	Documents-pairs	label	Size
PAN-11	210	Plagiarized	88Mo
JRC-Acquis	10000	Plagiarized	436Mo
Wikipedia+ Europarl	9423	Not Plagiarized	382Mo

3.2 Preprocessing techniques

After dataset collection, we have as input a list of pairs of source documents and suspicious documents with two different languages (ES-EN) annotated as plagiarized or not. To prepare and clean the data, we apply some NLP techniques to the documents, such as the removal of punctuation and stop words, the conversion to lowercase, and the tokenization of words. We used the Natural Language Toolkit (NLTK), which is a software library in Python that supports the lemmatization of the documents into Spanish and English.

3.3 Feature extraction

The feature extraction techniques are used to convert documents into vectors using different WE techniques. Word embedding is a technique that converts individual words into a vector numerical representation. The vector captures various properties of this word about the text where it is located. These features can contain semantic and syntactic information. This step is essential when working with text using machine-learning models. Below, we propose a brief description of the techniques studied.

Word2Vec: is a word embedding technique proposed by Thomas Mikolov [40], [41] that represents each word in a vector space. The semantic similarity between the words is calculated using the vectors. Word2Vec contains two methods:

- Continuous Bag of Words Model (CBOW): uses the surrounding words to predict the central word.
- Skip-Gram Model: seeks to predict the surroundings based on the center word.

Doc2Vec: can be considered an extension of Word2Vec, which aims to create representation vectors for long text such as documents, paragraphs, and sentences. The representation of paragraphs in vectors is inspired by the way words are represented in vectors. This neural network uses words and paragraphs to generate vectors corresponding to the paragraphs. Doc2vec implements two techniques known as Paragraph Vector Distributed Memory (PV-DM) and Paragraph Vector Distributed Bag of Words (PV-DBOW). In PV-DM, each paragraph is represented by a distinct vector, in the form of a column in the matrix M, and each word is likewise presented by a distinct vector, appearing as a column in the N matrix. The paragraph vector and the word vectors are then combined to anticipate the next word in the context. As for PV-DBOW, this involves disregarding the context of the incoming words and having the model guess words chosen at random from the output paragraph. This implies that, at each iteration of the stochastic gradient descent, a window of text is selected, then a randomly chosen word from the text window and a classification undertaking is carried out, taking into account the paragraph vector [42].

GloVe: (global vectors for word representation) is used to efficiently capture contextual relationships between words [43]. It constructs a word-word co-occurrence matrix Mij by approximating the probability that a word wi occurs in a word wj. It is given by a function J for generating fixed-dimensional vectors from the vocabulary size V, scalar distortions bi and bj, and weighted frequencies. It is calculated as follows in equation (1):

$$J=f(X_{ij})(w_i^T w_j + b_i + b_j - log\ log\ (X_{ij}))^2 \quad (1)$$

FastText: is a free public-source library created by the Facebook AI Research team for learning WE and classification [44, 45]. For each word, FastText generates a word vector that contains both the term's meaning and its context in the document. It provides two models for computing word representations: Skip-Gram and CBOW, and covers 157 languages. Rather than teaching the word vectors directly, FastText represents every word as an n-gram character. Once words are mapped using n-grams of characters, a skip-gram model is formed to learn the embedded word. The model considers a word model with a sliding window on the words, as the internal structure of the words is not considered. The position of the n-grams does not matter as long as the characters are within this window. Even if a word does not appear during training, it may be broken down into n-grams to maintain its integration.

BERT: (Bidirectional Encoder Representations from Transformers) is designed to take into account both the left and right context in all layers to pre-train deep bidirectional representations of unlabeled text [46]. BERT applies two methods:

Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). The objective of MLM is to hide a random selection of input tokens and then attempt to predict those obscured tokens. The NSP task aims to ascertain whether a certain sequence A is followed by a certain sequence B or not. This task is conducted by combining two sequences at each iteration, with sentence A being followed by sentence B in 50% of cases.

3.4 SLSTM deep learning

Long-Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that is designed to address the gradient problem, making it effective for tasks that require long-term memory, such as text sequence prediction. This is accomplished by adding an internal memory state to the processed input, decreasing the impact of vanishing gradients [47], [48]. As illustrated in Figure 2, the forget gate is responsible for controlling the effects of previous input over time. Additionally, the cell has two other gates, the input gate, and the output gate.

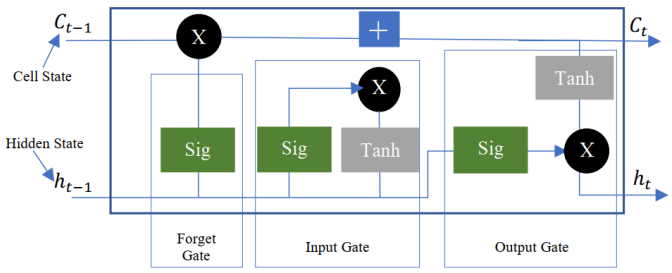

Fig. 2. The LSTM Architecture [49]

The formulas used to calculate Input Gate, Forget Gate, and Output Gate are depicted in equation (2), (3), and (4).

$$\text{Input Gate:} \quad i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \tag{2}$$

$$\text{Forget Gate:} \quad f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \tag{3}$$

$$\text{Output Gate:} \quad O_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \tag{4}$$

In our approach, the SLSTM model is used to learn plagiarism cases through a more accurate representation of the documents and also to detect similarity between pairs of objects, as it works well on similarity tasks. We used the Siamese LSTM, a version of the Manhattan LSTM model. The SLSTM has two networks, the left LSTM and the right LSTM, and each one will process a document and the other its corresponding suspect in a dependent manner. Furthermore, the vector representations of two texts return a hidden state encoding the semantic meaning of the texts. These hidden states are compared using a similarity metric to return a similarity score [50, 51].

4 Experiments and results

In this section, we present the results of our approach and make comparisons with FastText, Word2Vec, GloVe, BERT, and combinations of Word2Vec and Sent2Vec techniques using the accuracy, precision, recall, and f-score metrics.

4.1 Performance measure

The analysis of the suggested models is performed using performance metrics. They are computed using True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values.

- **Precision:** is a measure that counts the number of accurate predictions that turn out to be true:

$$Precision = TP/(TP + FP) \tag{5}$$

- **Recall:** the following formula is used to determine the number of accurate class predictions:

$$Recall = TP/(TP + FN) \tag{6}$$

- **Accuracy:** is the more intuitive performance metric. The formula used to calculate the accuracy is the following:

$$Accuracy = (TN + TP)/(TN + FP + TP + FN) \tag{7}$$

- **F-score:** is employed as a statistical metric to evaluate performance. An F-score could be supported by two factors, i.e., precision (P) and recall (R). The formula used to calculate the F-score is [52]:

$$F - Score = 2 * (P * R)/(P\ +\ R) \tag{8}$$

Table 3. Parameters of SLSTM Model

Model	Parameter	Value
SLSTM	Neuron	100
	Dropout	0.5
	Activation Function	Sigmoid
	Optimizer	SGD
	Loss Function	Binary cross Entropy
	Batch size	64
	Epochs	50

4.2 Results and analysis

Our objective is to propose a CLPD method based on Doc2Vec embedding and the Siamese Long Short-Term Memory (SLSTM) model, which takes as input the Doc2Vec embedding vectors of the first text at the LSTM layer and the embedding vectors of the second text at the LSTM layer separately and obtains a dense representation for the first and the second text. The fusion layer takes the dense representation of the first text and the second text and calculates the cosine distance between them. The experimental results of Doc2Vec were compared to other embedding techniques such as GloVe, FastText, Doc2Vec, Word2Vec+Sen2Vec, and BERT. For each method, we present the confusion matrix, the accuracy, recall, precision, F-score, and AUC values. The confusion matrix is shown in Figure 3.

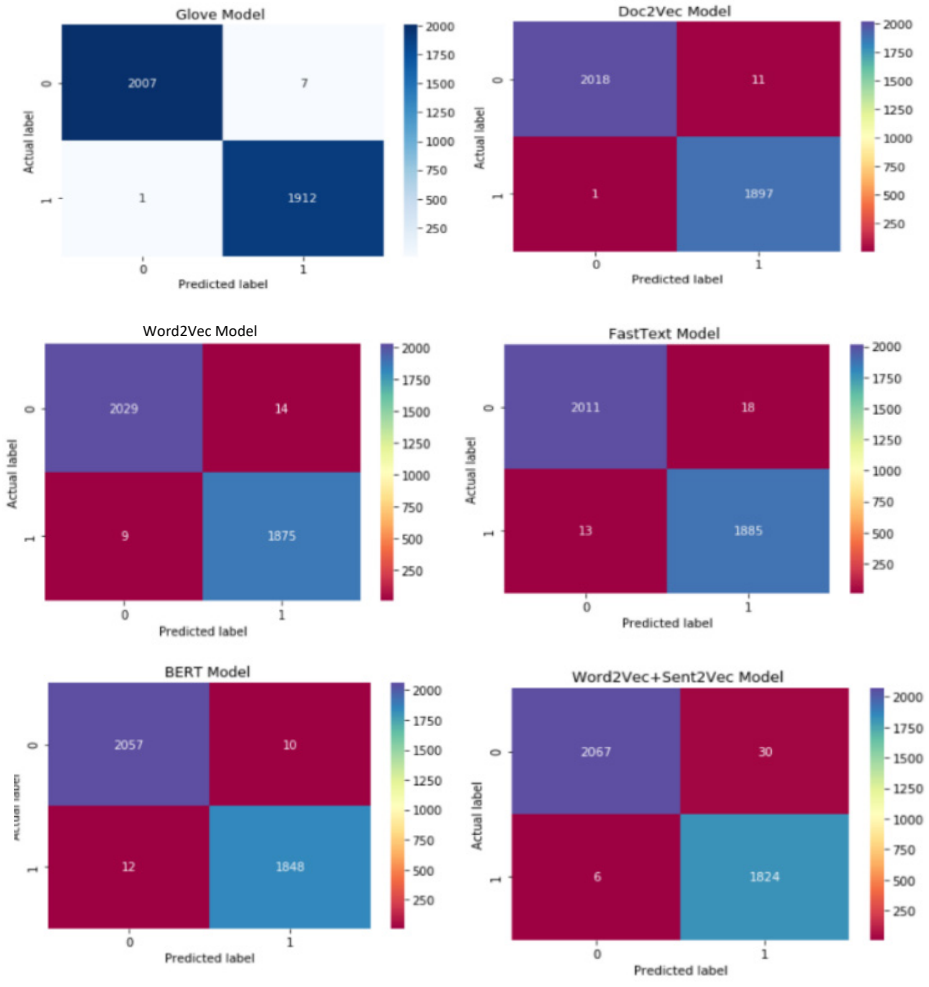

Fig. 3. Confusion matrix for each model

For binary classification, the confusion matrices of each model are as follows: 1 represents plagiarized documents, and 0 represents non-plagiarized documents. According to the confusion matrix, the glove method correctly classified 3927 documents; only 7 were incorrectly classified as false positives, and one false negative was incorrectly classified by the model. For the Doc2Vec model, we can see that among 3927 documents, 11 cases were incorrectly classified as false positives, and 1 was incorrectly classified as false negatives by this model. For the four models, Word2Vec, BERT, Word2Vec + Sent2Vec, and FastText, we can notice that the misclassified are more lifted compared to Glove and Doc2Vec.

Table 4 shows the performance measures of the Doc2Vec+SLSTM model in the test phase compared with the feature extraction techniques GloVe, FastText, BERT, Word2Vec, and Sent2Vec. All feature techniques investigated performed well on the five measures of performance. The Doc2Vec model achieved high performance with an accuracy of 99.81%, a precision of 99.75%, a recall of 99.88%, an f-score of 99.70%, and an AUC of 99.96%. The GloVe method provides a good result with an accuracy of 99.59%. The Glove+SLSTM model achieves good performance for all metrics, with 99.39% of precision, 99.88% of recall, 99.80% of f-score, and 99.89% of AUC. In the third range, BERT performed well, with 99.49% of accuracy and an AUC of 99.91%. Word2Vec achieved an accuracy of 99.14% and an AUC of 99.85%, which is also a good result. FastText and Sen2Vec performed lower than the previous models, with an accuracy of 98.82 and 98.41%, respectively.

Table 4. Models Performance in test set

Model	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)	AUC (%)
Doc2Vec+SLSTM	**99.81**	**99.75**	**99.88**	**99.70**	**99.96**
GloVe+SLSTM	99.59	99.39	99.82	99.80	99.89
FastText+SLSTM	98.82	98.37	99.39	99.23	99.84
BERT+SLSTM	99.49	99.45	99.57	99.46	99.91
Word2Vec +SLSTM	99.14	99.03	99.33	99.43	99.85
Sen2Vec+ SLSTM	98.41	98.03	98.98	99.13	99.74

To identify whether there is a learning problem, such as a model that is underfitting or overfitting, we examined the accuracy of the model during the training phase. After each cycle of optimization, the accuracy and loss values of a model indicate whether it is performing well or poorly. A decrease in loss and an increase in accuracy are expected after each iteration or several iterations. The accuracy of curve in Figure 4 shows that the Doc2Vec+SLSTM model is well-trained. The accuracy of the validation and training datasets increased for the last few epochs and reached 99.97% for training and 99.81% for validation. Thus, the loss of the model is 0.0049 for training and 0.0089 for validation, while the variance is smaller, which makes the model perform well.

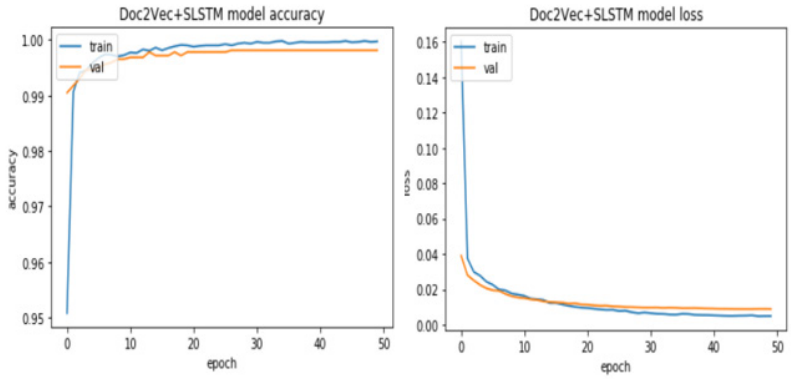

Fig. 4. Accuracy/Loss of Doc2Vec Model

The curves of the Glove+SLSTM model in Figure 5 show the accuracy and loss for the training and validation data. For accuracy, the training and validation data are approaching 1, with 99.90% for the training data and 99.59% for the validation data. For the loss curve presented in Figure 5, the model converges to 0 for both types of data, which indicates that the model has a good prediction because it reaches a loss of 0.0078 for training and 0.02 for validation.

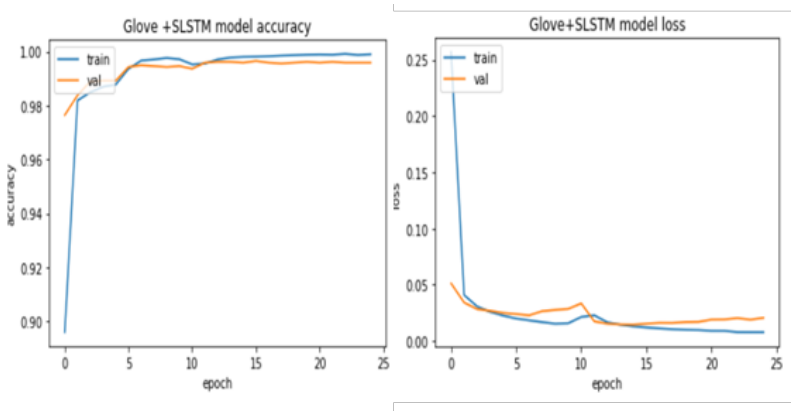

Fig. 5. Accuracy/Loss of GloVe Model

The graph of the accuracy/loss of FastText+SLSTM presented in Figure 6 achieves an accuracy of 99.98% for training and 98.82% for validation. The plot of loss shows that the model has a reasonable loss of 0.0047 for training and 0.03 for validation.

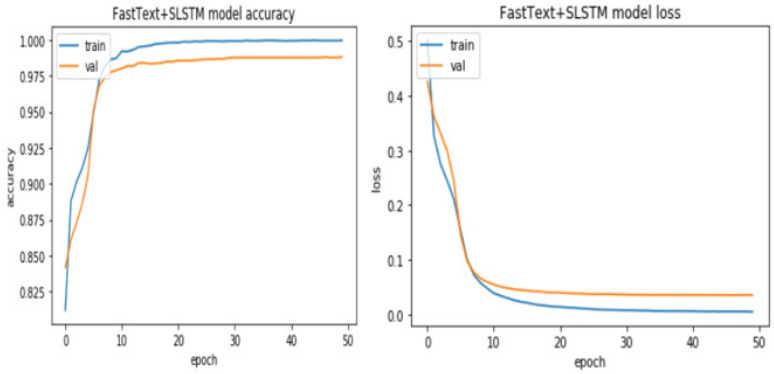

Fig. 6. Accuracy/Loss of FastText Model

The BERT+SLSTM model also achieved an interesting result, and Figure 7 shows that the loss on the training declines rapidly during the top five epochs. For the validation, the loss does not decline at the same rate as the training set but remains almost flat for several epochs.

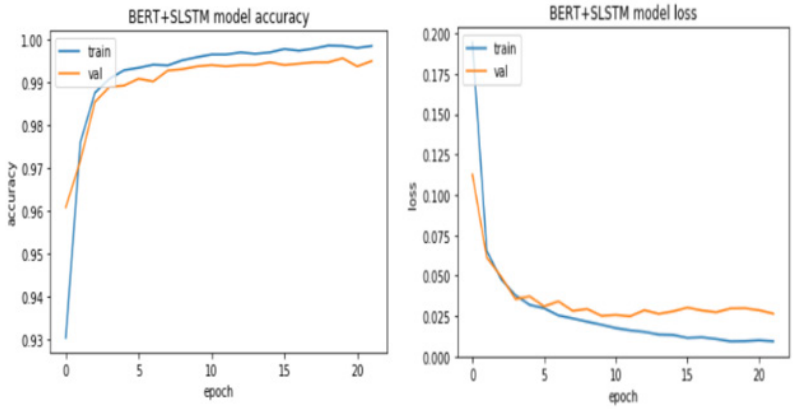

Fig. 7. Accuracy/Loss of BERT Model

The curves in Figure 8 show that the Word2Vec+SLSLTM model reaches 99.98% accuracy for training and 99.14% accuracy for validation. The loss has remained stable over the last few epochs and decreased for training to 0.0055 and 0.03 for validation.

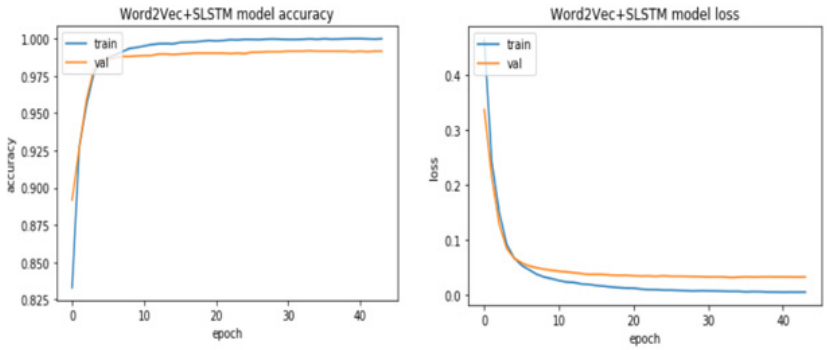

Fig. 8. Accuracy/Loss of Word2Vec Model

The graph in Figure 9 shows that Sen2Vec has good convergence for loss and accuracy for the training and validation. The performance remained closed with an accuracy of 98.71% for the training and 98.41% for the validation. The loss is 0.04 for training and 0.05 for validation.

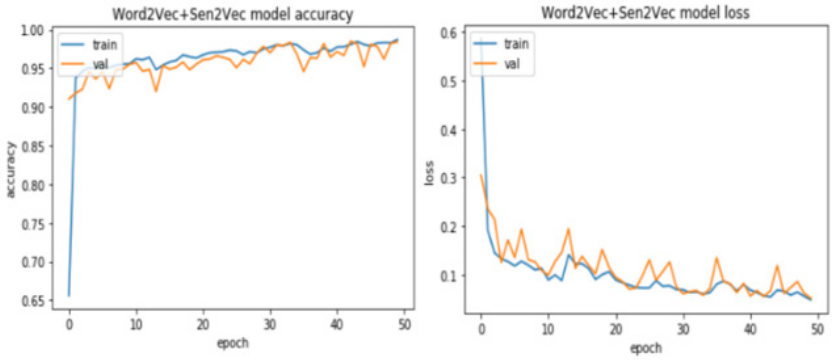

Fig. 9. Accuracy/Loss of Sen2Vec Model

The experiment results showed that all the models gave interesting results, but the Doc2Vec and GloVe models outperformed the embedding models Word2Vec, FastText, BERT, and Sen2Vec. The Doc2Vec model achieved better results than the glove model, with an accuracy of 99.81 and a loss of 0.0089 for the test phase. Our approach outperformed the baseline based on PAN 11, such as in [25], where Word2Vec + DNN achieved an accuracy of 97.01%, and in [26], where BERT + LR achieved a high accuracy of 96% on a different dataset. The performance results tested on the same language pair, Spanish- English, show that the combination Word2Vec+SLSTM model achieved a precision of 93%, which is higher than the Skip Gram model [34] based on Knowledge Graph that performed at 76.1% of precision, also based on the PAN-PC-11 dataset.

Collecting more training data, the Doc2Vec feature extraction method, and the SLSTM model contributed to improving the performance of Spanish-English CLPD, which achieved an accuracy of 99.81% and outperformed the baseline.

5 Conclusion

In this paper, we address the problem of Spanish-English CLPD. We proposed an approach based on Doc2Vec and SLSTM models as a novel approach for CL-PD. The models were trained on the aggregation of four publicly available corpora, namely Pan11, JRC-Acquis, Europarl, and Wikipedia. The use of the deep learning technique SLSTM improved the performance of five embedding models such as Word2Vec, GloVe, BERT, Word2Vec+Sen2Vec, and FastText. The experiments demonstrated that the Doc2Vec+SLSTM model achieved the highest results, with an accuracy of 99.81% and a very low loss in the test phase. The method was able to interpret sequential information and maintain long-term dependencies between words efficiently. Future work will aim to extend the current methodology to other contextual integration techniques, generalize it to other language pairs, and study the impact of document size in the training phase on the overall performance of integration techniques.

6 References

[1] E. Wager, "Defining and responding to plagiarism," Learn. Publ., vol. 27, no 1, p. 33-42, 2014. https://doi.org/10.1087/20140105

[2] N. Son, H. Le, et C. T. Nguyen, "A two-phase plagiarism detection system based on multi-layer LSTM networks," IAES Int. J. Artif. Intell. IJ-AI, vofol. 10, p. 636-648, sept. 2021. https://doi.org/10.11591/ijai.v10.i3.pp636-648

[3] R. Comas-Forgas et J. Sureda-Negre, "Academic Plagiarism: Explanatory Factors from Students' Perspective," J. Acad. Ethics, vol. 8, no 3, p. 217-232, sept. 2010. https://doi.org/10.1007/s10805-010-9121-0

[4] Husain, F. M., Al-Shaibani, G. K. S., & Mahfoodh, O. H. A. (2017). "Perceptions of and Attitudes toward Plagiarism and Factors Contributing to Plagiarism: a Review of Studies," Journal of Academic Ethics, 15(2), 167–195. https://doi.org/10.1007/s10805-017-9274-1

[5] Foltýnek, T., Meuschke, N., & Gipp, B. (2020). "Academic Plagiarism Detection: ACM Computing Surveys," 52(6), 1–42. https://doi.org/10.1145/3345317

[6] R. G. S. Berlinck, "The academic plagiarism and its punishments - a review," Rev. Bras. Farmacogn., vol. 21, no 3, p. 365-372, june 2011. https://doi.org/10.1590/S0102-695X–2011005000099

[7] R. R. Naik, M. B. Landge, et C. N. Mahender, "A review on plagiarism detection tools," Int. J. Comput. Appl., vol. 125, no 11, 2015.

[8] Sabeeh, M., & Khaled, F. (2021). "Plagiarism Detection Methods and Tools: An Overview," Iraqi Journal of Science, 2771–2783. https://doi.org/10.24996/ijs.2021.62.8.30

[9] Bechhoefer, J. (2007). "Plagiarism: text-matching program offers an answer," Nature, 449(7163), 658–658. https://doi.org/10.1038/449658b

[10] Weber-Wulff, D. (2016). "Plagiarism Detection Software: Promises, Pitfalls, and Practices," Handbook of Academic Integrity, 625–638. https://doi.org/10.1007/978-981-287-098-8_19

[11] Sulaiman, R. (2018). "Types and factors causing plagiarism in papers of english education students," Journal of English Education, 3(1), 17–22. https://doi.org/10.31327/jee.v3i1.471
[12] R. Safriyani, R. Rakhmawati, et L. U. Sadieda, « What to Accommodate to Develop Students' Academic Writing? Need Analysis for a Research-Based Textbook Development », IJET Indones. J. Engl. Teach., vol. 10, no 1, Art. no 1, juill. 2021. https://doi.org/10.15642/ijet2.2021.10.1.86-98
[13] S. Alzahrani et N. Salim, "Fuzzy Semantic-Based String Similarity for Extrinsic Plagiarism Detection," Lab Report for PAN at CLEF 2010".
[14] AlSallal, M., Iqbal, R., Palade, V., Amin, S., & Chang, V. (2019). "An integrated approach for intrinsic plagiarism detection," Future Generation Computer Systems, 96, 700–712. https://doi.org/10.1016/j.future.2017.11.023
[15] S. M. Z. Eissen, B. Stein, et M. Kulig, "Plagiarism detection without reference collections," in in Advances in Data Analysis, 2007, p. 359-366. https://doi.org/10.1007/978-3-540-70981-7_40
[16] P. Kawachi, "Initiating Intrinsic Motivation in Online Education: Review of the Current State of the Art," Interact. Learn. Environ., vol. 11, no 1, p. 59-81, janv. 2003. https://doi.org/10.1076/ilee.11.1.59.13685
[17] S. Yousf, M. Ahmad, et N. Sheikh, "A review of plagiarism detection based on Lexical and Semantic Approach," 2013, p. 5. https://doi.org/10.1109/C2SPCA.2013.6749430
[18] K. Palasundram, N. M. Sharef, N. Nasharuddin, K. Kasmiran, et A. Azman, « Sequence to Sequence Model Performance for Education Chatbot », Int. J. Emerg. Technol. Learn. IJET, vol. 14, no 24, p. 56-68, déc. 2019. https://doi.org/10.3991/ijet.v14i24.12187
[19] H. Aljuaid, "Cross-Language Plagiarism Detection using Word Embedding and Inverse Document Frequency (IDF)," Int. J. Adv. Comput. Sci. Appl., vol. 11, janv. 2020. https://doi.org/10.14569/IJACSA.2020.0110231
[20] L. T. Nguyen et D. Dien, "Vietnamese- English Cross-Lingual Paraphrase Identification Using Siamese Recurrent Architectures," in 2019 19th International Symposium on Communications and Information Technologies (ISCIT), sept. 2019, p. 70-75. https://doi.org/10.1109/ISCIT.2019.8905116
[21] G. Glavaš, M. Franco-Salvador, S. P. Ponzetto, et P. Rosso, "A Resource-Light Method for Cross-Lingual Semantic Textual Similarity," arXiv, 19 janvier 2018. https://doi.org/10.1016/j.knosys.2017.11.041
[22] A. Mahmoud et M. Zrigui, "Sentence Embedding and Convolutional Neural Network for Semantic Textual Similarity Detection in Arabic Language," Arab. J. Sci. Eng., vol. 44, août 2019. https://doi.org/10.1007/s13369-019-04039-7
[23] N. Alotaibi et M. Joy, "English-Arabic Cross-language Plagiarism Detection," in Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), Held Online, sept. 2021, p. 44-52. https://doi.org/10.26615/978-954-452-072-4_006
[24] R. Lachraf, E. M. Billah Nagoudi, Y. Ayachi, A. Abdelali, et D. Schwab, "ArbEngVec : Arabic-English Cross-Lingual Word Embedding Model," Florence, Italy, juill. 2019. https://doi.org/10.18653/v1/W19-4605
[25] S. Alzahrani et H. Aljuaid, "Identifying cross-lingual plagiarism using rich semantic features and deep neural networks: A study on Arabic-English plagiarism cases," J. King Saud Univ. - Comput. Inf. Sci., vol. 34, no 4, p. 1110-1123, avr. 2022. https://doi.org/10.1016/j.jksuci.2020.04.009
[26] D. V. Zubarev et I. V. Sochenkov, "Cross-language text alignment for plagiarism detection based on contextual and context-free models," 2019, p. 809-820.

[27] H. V. T. Chi, D. L. Anh, N. L. Thanh, et D. Dinh, "English-Vietnamese Cross-Lingual Paraphrase Identification Using MT-DNN," Eng. Technol. Appl. Sci. Res., vol. 11, no 5, Art. no 5, oct. 2021. https://doi.org/10.48084/etasr.4300

[28] E. M. B. Nagoudi, J. Ferrero, D. Schwab, et H. Cherroun, "Word Embedding-Based Approaches for Measuring Semantic Similarity of Arabic-English Sentences," in Arabic Language Processing: From Theory to Practice, Cham, 2018, p. 19-33. https://doi.org/10.1007/978-3-319-73500-9_2

[29] M. Al-Suhaiqi, M. A. S. Hazaa, et M. Albared, "Arabic English Cross-Lingual Plagiarism Detection Based on Keyphrases Extraction, Monolingual and Machine Learning Approach," Asian J. Res. Comput. Sci., p. 1-12, févr. 2019. https://doi.org/10.9734/ajrcos/2018/v2–i330075

[30] N. Alotaibi and M. Joy, "Using Sentence Embedding for Cross-Language Plagiarism Detection," Artificial Intelligence XXXVII, pp. 373–379, 2020. https://doi.org/10.1007/978-3-030-63799-6_28

[31] F. Yinhan, Z. Gang, M. Weixiu, L. Shunbao, Y. Shijie, et Z. Kui, "Calculation of Chinese-Thai Cross-Language Similarity Based on Sentence Embedding," in 2020 5th International Conference on Smart Grid and Electrical Automation (ICSGEA), Zhangjiajie, China, juin 2020, p. 268-271. https://doi.org/10.1109/ICSGEA51094.2020.00064

[32] J. Stegmüller, F. Bauer-Marquart, N. Meuschke, T. Ruas, M. Schubotz, et B. Gipp, "Detecting Cross-Language Plagiarism using Open Knowledge Graphs," p. 853881 Bytes, 2021. https://doi.org/10.6084/m9.figshare.17212340.v3

[33] C. Chang, C. Chang, et S. Hwang, "Employing word mover's distance for cross-lingual plagiarized text detection," Proc. Assoc. Inf. Sci. Technol., vol. 57, no 1, oct. 2020. https://doi.org/10.1002/pra2.229

[34] J. Ferrero, F. Agnes, L. Besacier, et D. Schwab, "UsingWord Embedding for Cross-Language Plagiarism Detection," arXiv, 10 février 2017. https://doi.org/10.18653/v1/E17-2066

[35] M. Franco-Salvador, P. Rosso, et M. Montes-y-Gómez, "A systematic study of knowledge graph analysis for cross-language plagiarism detection," Inf. Process. Manag., vol. 52, no 4, p. 550-570, juill. 2016. https://doi.org/10.1016/j.ipm.2015.12.004

[36] M. Potthast, B. Stein, A. Eiselt, A. Barrón-Cedeño, et P. Rosso, "PAN Plagiarism Corpus 2011 (PAN-PC-11)," Zenodo, 1 juin 2011. https://doi.org/10.5281/zenodo.3250095

[37] R. Steinberger, B. Pouliquen, A. Widiger, C. Ignat, T. Erjavec, et D. Tufi, "The JRC-Acquis: A Multilingual Aligned Parallel Corpus with 20+ Languages,".

[38] A. Barrón-Cedeño, C. España-Bonet, J. Boldoba, et L. Màrquez, "A Factory of Comparable Corpora from Wikipedia," in Proceedings of the Eighth Workshop on Building and Using Comparable Corpora, Beijing, China, juill. 2015, p. 3-13. https://doi.org/10.18653/v1/W15-3402

[39] P. Koehn, "Europarl: A Parallel Corpus for Statistical Machine Translation," in Proceedings of Machine Translation Summit X: Papers, Phuket, Thailand, sept. 2005, p. 79-86.

[40] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, et J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in Advances in Neural Information Processing Systems, 2013, vol. 26.

[41] P. Juric, M. Brkic Bakaric, et M. Matetic, «Implementing M-Learning System for Learning Mathematics Through Computer Games and Applying Neural Networks for Content Similarity Analysis of an Integrated Social Network», Int. J. Interact. Mob. Technol. IJIM, vol. 15, p. 145, juill. 2021. https://doi.org/10.3991/ijim.v15i13.22185

[42] Q. Le et T. Mikolov, "Distributed Representations of Sentences and Documents," in Proceedings of the 31st International Conference on Machine Learning, juin 2014, p. 1188-1196.
[43] J. Pennington, R. Socher, et C. Manning, "GloVe: Global Vectors for Word Representation," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, oct. 2014, p. 1532-1543. https://doi.org/10.3115/v1/D14-1162
[44] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, et T. Mikolov, "FastText.zip: Compressing text classification models," arXiv, 12 décembre 2016.
[45] P. Donner, "Identifying constitutive articles of cumulative dissertation theses by bilingual text similarity. Evaluation of similarity methods on a new short text task," Quant. Sci. Stud., vol. 2, no 3, p. 1071-1091, nov. 2021. https://doi.org/10.1162/qss_a_00152
[46] J. Devlin, M.-W. Chang, K. Lee, et K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv, 24 mai 2019.
[47] S. Hochreiter et J. Schmidhuber, "Long Short-Term Memory," Neural Comput., vol. 9, no 8, p. 1735-1780, nov. 1997. https://doi.org/10.1162/neco.1997.9.8.1735
[48] E. M. Hambi et F. Benabbou, "A New Online Plagiarism Detection System based on Deep Learning," Int. J. Adv. Comput. Sci. Appl., vol. 11, no 9, 2020. https://doi.org/10.14569/IJACSA.2020.0110956
[49] H. ElMoaqet, M. Eid, M. Glos, M. Ryalat, et T. Penzel, "Deep Recurrent Neural Networks for Automatic Detection of Sleep Apnea from Single Channel Respiration Signals," Sensors, vol. 20, no 18, Art. no 18, janv. 2020. https://doi.org/10.3390/s20185037
[50] J. Mueller et A. Thyagarajan, "Siamese Recurrent Architectures for Learning Sentence Similarity," Proc. AAAI Conf. Artif. Intell., vol. 30, no 1, Art. no 1, mars 2016. https://doi.org/10.1609/aaai.v30i1.10350
[51] W. Bao, W. Bao, J. Du, Y. Yang, et X. Zhao, "Attentive Siamese LSTM Network for Semantic Textual Similarity Measure," in 2018 International Conference on Asian Language Processing (IALP), Bandung, Indonesia, nov. 2018, p. 312-317. https://doi.org/10.1109/IALP.2018.8629212
[52] D. Chicco et G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," BMC Genomics, vol. 21, no 1, p. 6, déc. 2020. https://doi.org/10.1186/s12864-019-6413-7

7 Authors

Chaimaa Bouaine graduated with a Master's degree in Big Data and Data Science option Big Data from Hassan II University in Casablanca, Morocco in 2021. Currently, she is preparing her PhD in Computer Science at the Laboratory of Information Processing and Modeling (LTIM) at the Faculty of Science Ben M'SIK. Her research interest is cross-language plagiarism detection including natural language preprocessing, machine learning, and deep learning. (email: chaimaa.bouaine-etu@etu.univh2c.ma).

Faouzia Benabbou is a professor of Computer Science and member of Computer Science and Information Processing laboratory. She is Head of the team "Cloud Computing, Network and Systems Engineering (CCNSE)". She received his Ph.D. in Computer Science from the Faculty of Sciences, University Mohamed V, Morocco, 1997.

His research areas include cloud Computing, data mining, machine learning, and Natural Language Processing. She has published several scientific articles and book chapters in these areas. (email: faouzia.benabbou@univh2c.ma).

Imane Sadgali graduated as a state engineer in computer science and new technologies in 2009 from the INPT, and after 9 years in payment systems as project manager, returned to scientific research to obtain a doctorate in computer science in 2020. Currently, consultant in payment systems and still active in scientific research. (email: sadgali.imane@gmail.com).

Article submitted 2023-02-05. Resubmitted 2023-03-15. Final acceptance 2023-03-15. Final version published as submitted by the authors.