UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

DELTON DE ANDRADE VAZ

Cross-language plagiarism detection with contextualized word embeddings

Work presented in partial fulfillment
of the requirements for the degree of
Bachelor in Computer Engineering

Advisor: Prof. Dr. Viviane P. Moreira

Porto Alegre
June 2021

“Nothing in life is to be feared, it is only to be understood.
Now is the time to understand more, so that we may fear less.” — MARIE CURIE

AGRADECIMENTOS

Primeiramente, gostaria de agradecer e dedicar esse trabalho à minha família, por sempre acreditar, investir e me apoiar incondicionalmente durante meus estudos. Quero agradecer à Polytech Montpellier (Université Montpellier II) e à Universidade Federal do Grande do Sul (UFRGS) pela oportunidade de fazer o duplo diploma, fazendo parte do programa de intercâmbio BRAFITEC (BRAsil France Ingénieurs TEChnologie), pela parceria entre essas duas instituições, não esquecendo de mencionar a CAPES pelo financiamento da minha bolsa. Esse trabalho vem para finalizar um ciclo de 8 anos, 6 de graduação somados a 2 anos de duplo diploma.

Aos meus amigos e meu namorado que me dão suporte e forças para eu continuar sempre crescendo. Quero agradecer em especial ao Jonas, que foi meu irmão de graduação, agradecer ao Felipe pelas dicas e à Maria por ser minha dupla durante o período do duplo diploma na Polytech. Merci beaucoup!

À professora orientadora Viviane Moreira, que aceitou o convite para desenvolver esse trabalho e me ajudou durante o processo de desenvolvimento. Mesmo me orientando à distância e com fuso horário diferente, sempre fez isso com qualidade e maestria.

ABSTRACT

Plagiarism is the use of someone else's work without the proper acknowledgment and citation, being one of the most significant publishing issues in academia and science. A study conducted by CopyLeaks in 2020 showed that plagiarism increased by 10% after the transition to online classes during the COVID-19 pandemic. In some cases, authors may translate texts from another language and include them in their work. This more "sophisticated" behavior is known as cross-language plagiarism. In this work, we investigate methods that are used for cross-language plagiarism detection. Although some of the approaches developed until now use word embeddings as part of their pipelines, few explore contextualized word embeddings. Contextualized embeddings can help address fundamental characteristics of language such as polysemy and synonymy by taking into account the context in which a particular word occurs. Pre-trained multilingual models have shown outstanding performance in downstream natural language understanding tasks, such as sentence similarity and next sentence prediction. Motivated by these promising results in tasks related to plagiarism detection, we present a new proposal for cross-language plagiarism detection using pre-trained multilingual models with contextualized embeddings. Experiments performed on different datasets, such as PAN-PC-12, show that the proposed cross-language plagiarism detection using contextualized embeddings outperforms state-of-the-art models by 9% and 11% regarding plagdet results obtained for the English-Spanish and English-German language pairs.

Keywords: Cross language plagiarism detection. BERT. cross language information retrieval. word embeddings.

Detecção de plágio multilíngue usando word embeddings contextualizadas

RESUMO

Plágio é o uso do trabalho de outra pessoa sem o devido reconhecimento e citação, sendo um dos maiores problemas editoriais da academia e da ciência. Um estudo realizado em 2020 pela CopyLeaks mostrou que o plágio aumentou em 10% após a transição para aulas online durante a pandemia da COVID-19. Em alguns casos, os autores podem traduzir textos de outro idioma e incluir em seus próprios trabalhos. Este comportamento mais "sofisticado" é conhecido como plágio multilíngue. Neste trabalho, investigamos métodos que são usados para a detecção do plágio multilíngue. Embora algumas das abordagens desenvolvidas até agora utilizem *word embeddings* como parte de seu *pipeline*, poucas delas exploram *contexualized word embeddings*. *Contexualized word embeddings* consideram características fundamentais da linguagem, como a polissemia, levando em conta o contexto no qual uma palavra em particular ocorre. Modelos multilíngues pré-treinados têm demonstrado grande desempenho em tarefas multilíngues, tais como similaridade de sentenças e predição de próxima sentença. Assim, com resultados promissores para tarefas relacionadas à detecção de plágio, apresentamos uma nova proposta para a detecção de plágio multilíngue utilizando modelos multilíngues pré-treinados com embeddings contextuais. Experimentos realizados em diferentes conjuntos de dados, como o PAN-PC-12, mostram que a detecção de plágio multilíngue utilizando modelos multilíngues pré-treinados com embeddings contextuais supera supera em 9% e 11% os modelos de última geração em relação aos resultados de plagdet obtidos para os pares de idiomas inglês-espanhol e inglês-alemão.

Palavras-chave: plágio multilíngue, BERT, recuperação de informação multilíngue, word embeddings.

LIST OF ABBREVIATIONS AND ACRONYMS

CLIR	Cross-language Information
CLPD	Cross-language plagiarism
DNN	Deep Neural Network detection
IR	Information Retrieval Retrieval
MT	Machine Translation
NLP	Natural Language Processing
SOTA	State-of-the-art
WE	Word Embeddings

LIST OF FIGURES

Figure 1.1 Generic plagiarism detection process 12

Figure 2.1 Architecture of a single-layer perceptron 16
Figure 2.2 Encoder-decoder architecture 17
Figure 2.3 Encoder-decoder with the attention mechanism layer 18
Figure 2.4 SBERT architecture 21
Figure 2.5 Query translation 23

Figure 3.1 Taxonomy of plagiarism 27

Figure 4.1 Overall workflow for CLPD-CWE 30
Figure 4.2 Pre-processing sentence grouping 32
Figure 4.3 Tolerance threshold 35
Figure 4.4 Output format for the detected plagiarism cases 36

Figure 5.1 Plagiarized passage and detections 41
Figure 5.2 Information retrieval returned sorted query results G' 44
Figure 5.3 Results for ECLaPA dataset with different similarity thresholds. 47
Figure 5.4 Results for PAN-PC-12 Spanish subset with different similarity thresholds 47
Figure 5.5 Results for PAN-PC-12 German subset with different similarity thresholds 48
Figure 5.6 Results for CrossLang dataset with different similarity thresholds. 50

Figure 6.1 PCA plot for ECLaPA source document embeddings 53

LIST OF TABLES

Table 4.1 Data storage format with two sentences 32

Table 5.1 Datasets details 38
Table 5.2 Dataset statistics 39
Table 5.3 Google Colab Virtual Machine Specifications 40
Table 5.4 CLPD-CWE – parameters used in the experiments 41
Table 5.5 Results of CLIR - ECLaPA Dataset 46
Table 5.6 Results of CLPD-CWE and baseline on the ECLaPA Dataset 46
Table 5.7 Results of CLIR - PAN-PC-12 Dataset 48
Table 5.8 Comparison of the proposed approach on PAN-PC-12 Spanish partition 49
Table 5.9 Comparison of the proposed approach on PAN-PC-12 German partition 49
Table 5.10 Results of CLIR - CrossLang Dataset 49
Table 5.11 Comparison of the proposed approach on CrossLang dataset. 50
Table 5.12 Pre-processing time and cross-language similarity time analysis 51

CONTENTS

1 INTRODUCTION....11
1.1 Motivation and goals....11
1.2 Overview....13
1.3 Contributions....13
1.4 Organization of the text....14
2 BACKGROUND....15
2.1 Neural networks....15
2.2 Attention Mechanism....17
2.3 The Transformer....18
2.4 Word Embeddings....19
2.5 Plagiarism Detection....21
2.6 Semantic Textual Similarity....22
2.7 Cross-Language Information Retrieval....22
2.8 Background Summary....24
3 RELATED WORK....25
3.1 Cross-language Candidate Document Retrieval....25
3.2 Cross-language Plagiarism Detection....26
3.3 PAN Evaluation Campaigns....28
3.4 Related Work Summary....29
4 CROSS-LANGUAGE PLAGIARISM DETECTION WITH CONTEXTUALIZED WORD EMBEDDINGS....30
4.1 Pre-processing....31
4.2 Candidate Retrieval....32
4.3 Cross-language Similarity Analysis....34
4.4 Post-processing....35
5 EVALUATION....37
5.1 Materials and Methods....37
5.1.1 Trec eval....37
5.1.2 Datasets....38
5.1.3 Tools....39
5.1.4 Detection Parameters....40
5.2 Evaluation Metrics....41
5.2.1 Recall....42
5.2.2 Precision....42
5.2.3 Granularity....42
5.2.4 Overall Score....43
5.2.5 Mean Average Precision....43
5.3 Experimental Results....45
5.3.1 ECLaPA - Experimental Results....45
5.3.2 PAN-PC-12 - Experimental Results....46
5.3.3 CrossLang - Experimental Results....49
5.3.4 Processing Time Analysis....51
6 CONCLUSION....52
REFERENCES....54

1 INTRODUCTION

1.1 Motivation and goals

Plagiarism is the use of someone else's work without the proper acknowledgment and citation, being one of the biggest publishing issues in academia and science. This unethical behavior is growing rapidly, fueled by the easiness of sharing and retrieving information on the Internet (SÁNCHEZ-VEGA et al., 2019).

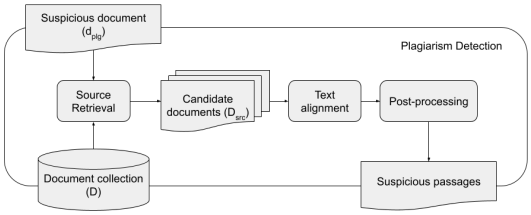
Nowadays, with the significant increase in the number of articles available on the Internet, it is easier for students and researchers to reuse texts from other authors deliberately without giving credit. A study by CopyLeaks[1], a company that sells plagiarism detection software, showed that plagiarism increased by 10% after the transition to online classes during the COVID-19 pandemic. In the literature, we can find several types of plagiarism (Soleman; Fujii, 2017), which can vary from copying someone else's work to paraphrasing a translated text in an attempt to mask the counterfeiting. Plagiarism also includes self-plagiarism since the text author can reuse its text in order to boost their production. There are few institutions that use a system based on the number of papers to offer grants. One example is Brazil which ranks researchers in publication per year.

Automated solutions for plagiarism detection (PD) can help humans identify possible copied contents and their sources since manual evaluation is not feasible on a large scale (POTTHAST et al., 2011a). Furthermore, authors may translate texts from another language and include them in their work. This more "sophisticated" behavior is known as cross-language plagiarism. While PD systems are of great interest as a countermeasure to maintain the highest possible level of integrity in the scientific community, they must also account for cross-language plagiarism.

PD systems may be classified as monolingual or cross-language systems. Monolingual systems are used when the suspicious and source documents are all written in the same language. The majority of plagiarism identification research falls into this group. In the second group, cross-language systems, the suspicious and source documents are written in different languages. The detection of cross-language plagiarism has become a research challenge since the textual similarity between the original text fragment and the plagiarized one is lost in translation (ROOSTAEE; SADREDDINI; FAKHRAHMAD, 2020).

[1] <https://copyleaks.com/media/COVID-19_STATE_OF_PLAGIARISM_REPORT.pdf>

Figure 1.1: Generic plagiarism detection process



Source: Stein, Eissen and Potthast (2007)

Figure 1.1 depicts a generic PD process for a suspicious document d_{plg} and a very large document collection D of potential source documents. The detection process is divided into three main phases used for most PD systems (POTTHAST et al., 2013). The first phase is candidate retrieval, which identifies a limited number of candidate source documents $D_{src} \in D$ that are possible sources for d_{plg}. Second, text alignment, in which each candidate document $d_{src} \in D_{src}$ is compared to d_{plg}, using a similarity model. Third, knowledge-based post-processing involves cleaning and filtering the collected passage pairs. It is important to point out that although PD systems aim at detecting plagiarized passages, they actually detect textual similarity. In order to determine if a case of textual similarity is indeed plagiarism, human intervention is needed.

Nevertheless, the approaches presented so far, such as Cross-language Word Embedding-based Straightforward (CL-WES) and Cross-Language Word Embedding-based Syntax Similarity (CL-WESS) (FERRERO et al., 2017) rely on word embeddings (WE). However, none of them takes *contextualized word embeddings* into account. For a number of downstream tasks, contextualized word embeddings have proved to be useful (PETERS et al., 2018; PETERS et al., 2017). Such embedding models encode both words and their contexts and create context-specific representations, unlike conventional WE that represent words as fixed vectors.

In light of that, this work aims to develop a new approach called CLPD-CWE (cross-language plagiarism detection with contextualized word embeddings). The difference between our approach and the existing ones is the use of contextualized word embeddings generated by a pre-trained Bidirectional Encoder Representations from Transformers (BERT) (DEVLIN et al., 2018) model.

1.2 Overview

We split the proposed approach into four main phases: pre-processing, candidate retrieval, cross-language similarity analysis, and post-processing. As our approach is intended to use context-generated embeddings from pre-trained multilingual models, we do not rely on machine translation systems in any of the phases of CLPD-CWE. Still, we use a language detector tool that is required to tokenize documents into sentences correctly.

After building a data frame with the source documents and their sentence embeddings, we move on to the candidate retrieval step. We generate keywords from the suspicious document during the candidate retrieval phase and then generate embeddings of the sentences containing at least one of the generated keywords. This sentence selection reduces the processing time by decreasing the number of sentences used in the further similarity analysis. In order to recover documents that are most likely plagiarized, we compare the suspicious document embeddings with all previously created embeddings. The source documents with the highest number of similar sentences concerning the suspicious document are selected as candidates. Then, a detailed plagiarism analysis is performed on the candidates. The final step consists of removing the false-positive cases and joining adjoining plagiarized cases.

We conducted experiments using three datasets in different languages. The results show a significant improvement outperforming SOTA models by 9% and 11% regarding plagdet results obtained for the Spanish and German.

1.3 Contributions

Briefly, the main contributions of this work are:

- A new multilingual PD method that uses contextualized embeddings generated by a pre-trained model.
- The implementation is publicly available so that it can be used by other researchers[2].

[2]<https://bit.ly/3xWFp4N>

1.4 Organization of the text

This work is organized as follows: Chapter 2 covers the background work describing techniques that are used in this monograph. Chapter 3 reports on related work in cross-language information retrieval (CLIR) and cross-language plagiarism detection (CLPD). Chapter 4 introduces the proposed approach. In Chapter 5, we present the experimental evaluation and compare our results against existing CLPD approaches. In Chapter 6, we summarize our main contributions and point out directions for future work.

2 BACKGROUND

In this section, we describe essential concepts necessary to the understanding of this work.

2.1 Neural networks

A neural network is a massively parallel distributed processor made up of simple processing units that have a natural propensity for storing experiential knowledge and making it available for use (HAYKIN, 2009).

The perceptron (ROSENBLATT, 1958), one of the first neural network models, corresponds to a two-class model in which the input vector x is first transformed using a fixed nonlinear transformation to give a feature vector $\phi(x)$, and this is then used to construct a generalized linear model of the form $y(x) = f(w^T\phi(x))$ (BISHOP, 2006), where the non linear function $f(\cdot)$ is given by a step function of the form:

$$f(a) = \begin{cases} +1 & a \geq 0 \\ -1 & a < 0 \end{cases}$$

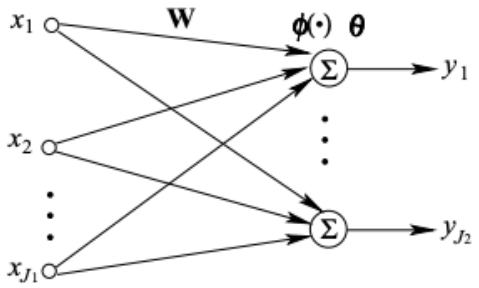
However, the function $f(a)$ can take several forms. This function is usually flattened into a predefined range to have better results related to the task goal (e.g., classification and regression). For instance, the classification model can use the sigmoid logistic function $f(a) = \frac{1}{1+e^{w^T\phi(x)}}$ in order to estimate the probability of a class membership.

When a perceptron is used alone, it does not have much power to handle or store information. However, other neural network models have been proposed. These models show that when perceptrons are used and trained in a combined way, they have the power to solve complex problems. An example is the Multilayer perceptron.

Multilayer perceptron (MLP): Also known as a feed-forward neural network, the multilayer perceptron is a layered arrangement of various perceptron neural units. It consists of an input layer, one or more hidden layers, and an output layer. Figure 2.1 shows the architecture of a single-layer perceptron.

Convolutional neural network (CNN): Convolutional neural networks (CNNs) are at the core of state-of-the-art approaches in a variety of computer vision tasks, including

Figure 2.1: Architecture of a single-layer perceptron



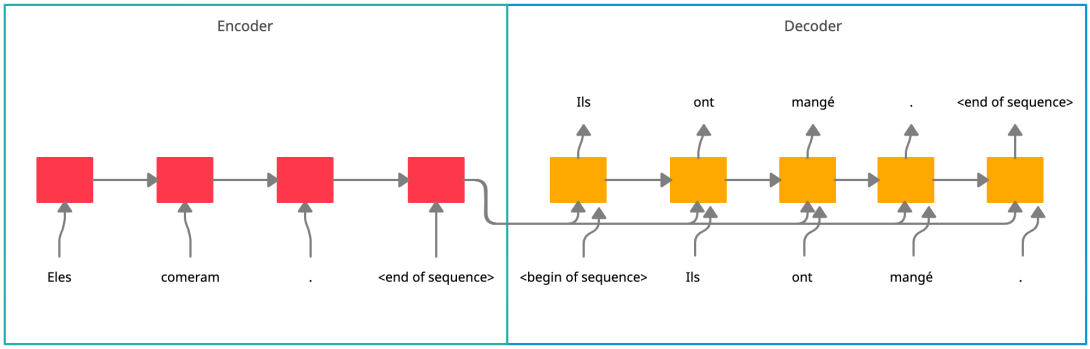
Source: Du and Swamy (2019)

image classification and object detection (HARLEY, 2015). However, in the context of natural language processing (NLP), tasks are usually sentences or documents represented as a matrix. In Section 2.4, we present embeddings, which is a one-hot vector word representation that indexes a word into a vocabulary.

Recurrent Neural Network (RNN): RNN is a type of neural network architecture in which there are connections between layers forming a directed cycle, creating an internal state and, achieving a dynamic behavior in time (UNANUE; BORZESHI; PICCARDI, 2017). In the NLP field, RNNs are extremely useful as they can model a sequence of tokens. For example, if one wants to develop a topic detection tool, the sentences "the blue sky" and "the sky blue" should produce different outputs. When modeling textual data, it is desirable that when a word or token is detected, its sequence in the text is also captured; otherwise, the relationship is missed, and the neural network will produce the same results. However, in practice, these architectures fail to learn long-term dependencies since they tend to be biased by the most recent results (BENGIO; SIMARD; FRASCONI, 1994). RNNs for NLP have traditionally struggled with vanishing, or exploding gradients, which has hampered their acceptance before new approaches to dealing with long sequences were created. As multiple neurons are used, the vanishing gradient effect will occur since the gradient decreases in the opposite direction of propagation. On the other hand, the exploded gradient effect happens as the gradient rises at each layer during training, allowing the anterior layers to change in response to small perturbations in the posterior layers, destabilizing the network significantly. Hochreiter and Schmidhuber (1997) proposed Long Short-Term Memory (LSTM) networks solve recurrent networks' gradient problem. LSTM neurons will "forget" information that is not important to the problem while storing and only forwarding information valuable to the next layer of neurons.

Sequence-to-sequence: Also called *seq2seq* encoder-decoder architecture (CHO et al., 2014) (SUTSKEVER; VINYALS; LE, 2014a), is the pillar of state-of-the-art generation models in sequence transduction tasks (i.e., a model that produces a time step for each input time step provided). To tackle transduction tasks, it suggests encoding the whole sequence at once and then uses this encoding as the foundation for producing the decoded sequence. A variable-length sentence is fed into the RNN encoder, which converts it into a fixed-shape hidden state. In other words, the RNN encoder's hidden state contains information from the input sequence. A separate RNN decoder will predict the next token depending on what tokens have been shown to produce the output sequence token by token (BENGIO et al., 2003). Since it is difficult to retain the encoder's meaning for longer sequences, the attention mechanism (see Section 2.2) is developed to "pay attention" to specific words in a sentence that contribute significantly to the generation of the target sequence.

Figure 2.2: Encoder-decoder architecture

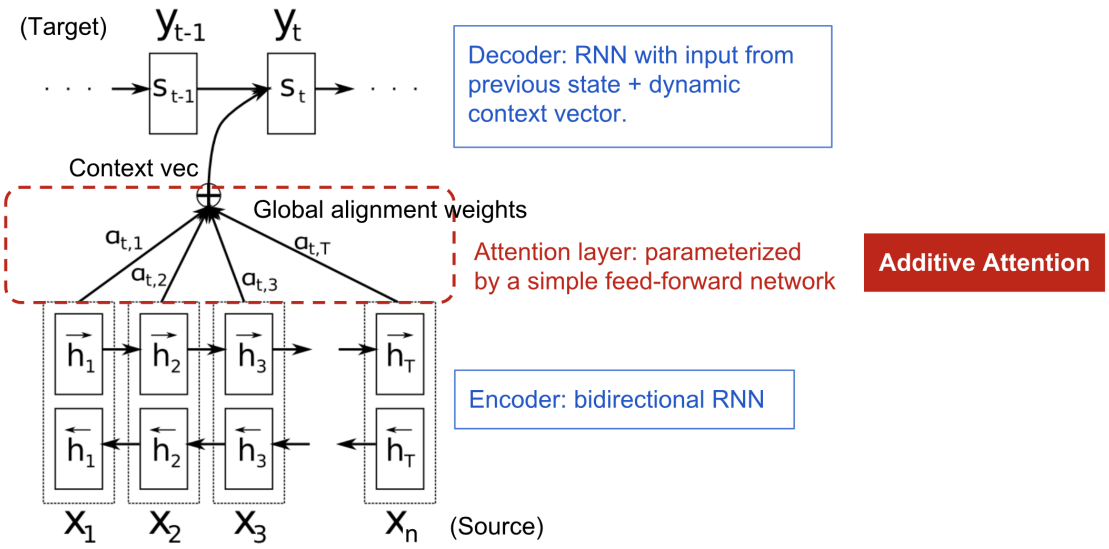


Source: The author

2.2 Attention Mechanism

In deep networks (SZEGEDY; TOSHEV; ERHAN, 2013), the attention mechanism imitates the mechanism of human vision. For example, when we observe an image, we can give more attention to specific points of that image, or in the case of NLP, some words present in the text. The mechanism of attention solves the problem of sequence-to-sequence networks (SUTSKEVER; VINYALS; LE, 2014b) that has a critical problem: the inability to memorize long sentences. With the help of the attention mechanism, the dependencies between source and target sequences are not restricted by in-between distance.

Figure 2.3: Encoder-decoder with the attention mechanism layer



Source: Lil'Log Blog <https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html>

2.3 The Transformer

Vaswani et al. (2017) proposed The Transformer, which is an architecture based on the attention mechanism in order to accelerate the speed at which language models can be trained. Language models analyze text data to assess next word likelihood. They use an algorithm to analyze the data, which sets rules for meaning in natural language. The model then uses these principles to predict or generate new sentences in language tasks correctly. The model learns the fundamental features and properties of language and then applies them to new phrases. Hence, The Transformer preserves the parallelization characteristics of CNNs and RNNs to treat the problems of input size variation and the dependency on sequential information. The *multi-head self-attention mechanism* is the transformer's most important component. The encoded representation of the input is viewed by the transformer as a series of *key-value* pairs (K,V), both of with dimension n (input sequence length). The previous output is compressed into a query Q of dimension m in the decoder, and the subsequent output is produced by mapping this query to the set of keys and values (WENG, 2018).

2.4 Word Embeddings

Neural word embeddings represent meaning via geometry. These representations are based on creating a model that projects a language's words into space where semantic relations between them can be identified and evaluated. Words are projected into a continuous multidimensional space, and those with a similar context will typically be the closest in this space. Similarities between terms can thus be measured using the cosine between their representations. Furthermore, the angles between the projections (vectors) of the words are influenced by the various relationships that connect the words.

Thanks to this, it is possible to exploit these relations with arithmetic operations on their vectors. For instance, the fact that the result of the operation *vector(Paris) - vector(France) + vector(Brazil)* will be more close to *vector(Brasilia)* than the others vectors. Noteworthy, word embeddings (WE) and deep neural network (DNN) methods in NLP have been a recurrent topic in the literature, with remarkable performance results in many different tasks, such as classification, clustering, and Semantic Textual Similarity (STS) (LI; YANG, 2018).

Contextual Embeddings. By assuming a single universal vector for each term, distributed word representations incur some limitations. This means disregarding important language characteristics such as polysemy and synonymy, not reflecting distinct meanings that the same word can infer. This trait is referred to as a *conflation deficiency* sense, in which the various definitions of the term are projected in a single vector space point. In assigning vectors for each term of reference, contextual representations of terms take on a different perspective, allowing for variance according to the context in which the term occurs. Depending on the context, the vector of a word will dynamically shift between the processing of a sentence and another.

Embedding from Language Model (ELMo) (PETERS et al., 2018) produces different word embeddings for each context in which the word is integrated, thus allowing different representations for the same word. Furthermore, ELMo uses internal character-based representations, thus allowing the consideration of the terms' morphological characteristics, making it possible to represent words outside the vocabulary used during the word embedding creation.

A limitation of this method is the disregard of the context to the right and left of the word. In this sense, a new proposal was developed, Bidirectional Encoder Representations

from Transformers (BERT) (DEVLIN et al., 2018), which established the new state-of-the-art parameters for different NLP benchmarks hence considering the context of words both from left and right.

Multilingual BERT (M-BERT) Multilingual BERT (M-BERT) is a language model with 104 languages pre-trained in the concatenation of monolingual corpora. M-BERT makes for a straightforward solution to cross-language model conversion with zero-shot (or few shots), e.g., without being exposed to any examples from a class in the training dataset, zero-shot seeks to predict the correct class. Typically, the model is fine-tuned using task-specific supervised training data from one language of the M-BERT languages, and inference is rendered using this multilingual language model. Pires, Schlinger and Garrette (2019) demonstrated that M-BERT creates multilingual representations, but these representations exhibit systematic deficiencies affecting specific language pairs. Languages with fewer data were over-sampled, whereas languages with large amounts of data (e.g., Russian) were under-sampled since the majority of data used to train M-BERT was in English.

DilstilBERT Multilingual Cased DistilBERT multilingual model (SANH et al., 2019) is a distilled (lightweight) version M-BERT (PIRES; SCHLINGER; GARRETTE, 2019). This model is cased: it does make a difference between *english* and *English*. Hence, distillation is a method of compressing a large model, known as the *teacher*, into a smaller model, known as the *student*. This compression has a significant impact on the processing time for word embeddings generation.

Sentence-BERT The lack of separate sentence embeddings in the BERT network structure is a significant disadvantage, making it challenging to extract sentence embeddings from BERT. To get around these limitations, researchers ran single sentences through BERT and then calculated a fixed-size vector by either averaging the outputs or combining them (similar to average word embeddings) (QIAO et al., 2019).

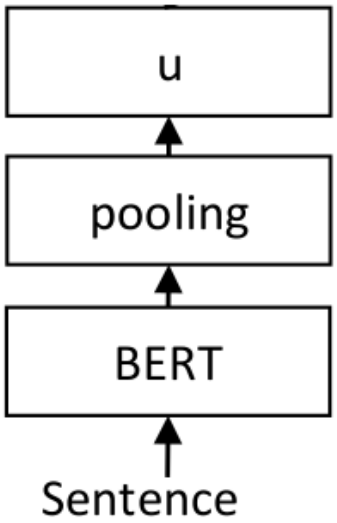
Sentence-BERT (SBERT) (REIMERS; GUREVYCH, 2019) feeds a transformer network, such as BERT, with an input sentence or text. For all input tokens, BERT generates contextualized word embeddings. Then, a pooling layer is used since SBERT generates a fixed-size output representation (vector u). There are various pooling options available, the most basic of which is mean-pooling: SBERT adds up all of the contextualized word embeddings that BERT provides. Equation 2.1 can be thought of as an average pooling operation. All word embeddings are averaged over each of the K dimensions,

resulting in a representation z of the same dimension as the embedding. Intuitively, z uses the addition operation to consider the information of each sequence element (SHEN et al., 2018).

$$z = \frac{1}{L}\sum_{i=1}^{L} v_i \tag{2.1}$$

Regardless of how long the input text or sentence is, the output is a fixed K-dimensional output vector, where K depends on the BERT model. Figure 2.4 depicts SBERT network architecture.

Figure 2.4: SBERT architecture



Source: (REIMERS; GUREVYCH, 2019)

2.5 Plagiarism Detection

Recently, there has been a spike in the number of research on plagiarism identification methods and techniques (ALZAHRANI; SALIM; ABRAHAM, 2011; GUPTA et al., 2016). PD systems are classified as either internal (intrinsic) or external (extrinsic)) in general. Internal detection systems are focused on examining linguistic features without comparing them to other documents (internal methods) to distinguish text fragments inconsistent with the rest of the document. However, external detection systems compare the suspicious document to a series of reference documents (external methods) to locate specific textual fragments in the source documents for the suspicious document's content. Furthermore, PD systems are often categorized into monolingual and multilingual

or cross-language detection systems to retrieve documents in language *L* that have been plagiarized from source documents in a language other than *L*.

2.6 Semantic Textual Similarity

Semantic Textual Similarity (STS) is one of the bases of PD since it measures the degree of equivalence between textual units (ALMEIDA et al., 2016). Computing cross-language semantic similarity poses additional challenges not faced in monolingual cases and usually requires additional linguistic resources such as parallel corpora and multilingual semantic networks (FRANCO-SALVADOR et al., 2016).

2.7 Cross-Language Information Retrieval

Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within extensive collections (usually stored on computers) (MANNING; RAGHAVAN; SCHÜTZE, 2008). CLIR is closely related to CLPD (PEREIRA; MOREIRA; GALANTE, 2010), as in CLIR, documents in one language are retrieved by a query Q in another language.

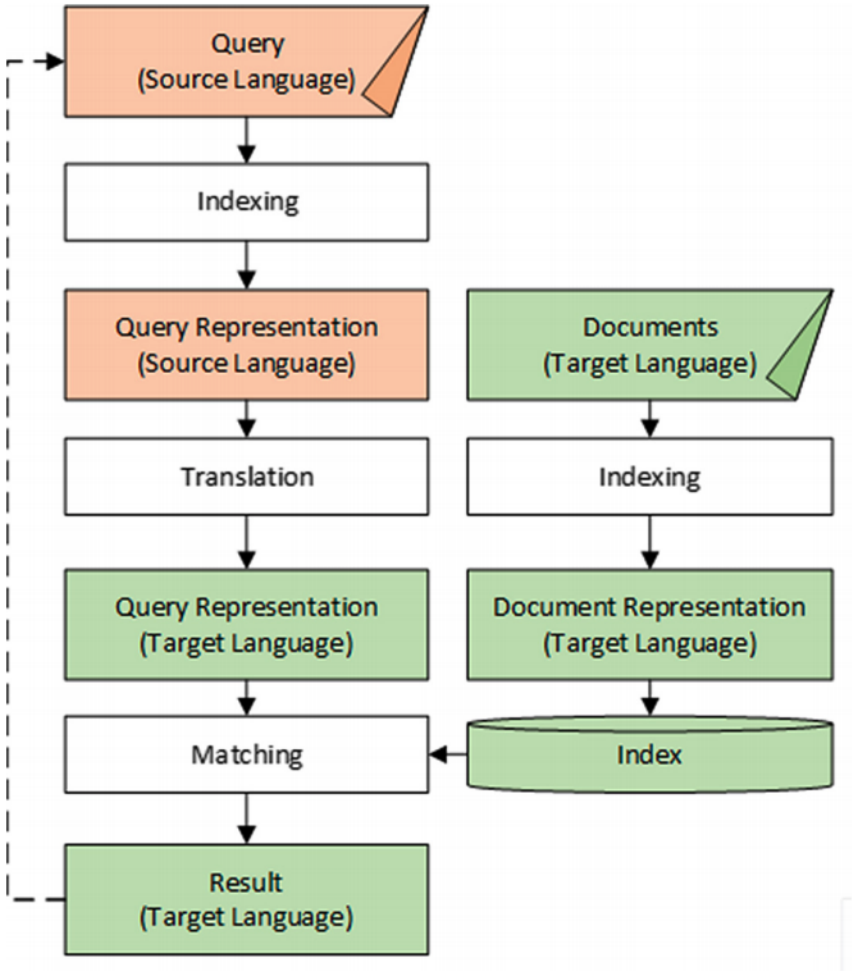
According to Zhang and Zhao (2020) CLIR approach can be included in three main categories: Translation (Query, Document, Pivot), Word Embedding, and Query Expansion.

Translation: Currently, query translation is the most common methodology used in CLIR (WU; HE, 2010). Query translation lays on the translation of the query Q in the source language to the suspicious language (OARD; HE; WANG, 2008). Hence, the CLIR is reduced to a monolingual retrieval problem. Compared with other approaches, an advantage is that query translation uses less computational resources. Figure 2.5 depicts the structure of query translation.

Another approach that uses translation is called Document Translation. Instead of translating only the query, this approach aims to translate all documents from the target language into the source language. It demands a considerable amount of computational resources and depends on the quality of translations.

Furthermore, there is a pivotal translation approach where the documents, both source, and target language, have scarce translation resources. This method aims to trans-

Figure 2.5: Query translation



Source: Zhang and Zhao (2020)

late one or both, source and target, into the pivot language. I.e., this approach is the combination of the document and query translation approaches.

Word Embedding: Vulić and Moens (2015) devised a method based on a text-level aligned bilingual corpus, employs the Skip-Gram model (GUTHRIE et al., 2006) to obtain cross-language word embedding composed of query and document embedding, measures the cosine similarity between them, and then sorts them by similarity. The method eliminates the need for tools like a bilingual dictionary or a machine translation (MT) system. Also, Bhattacharya, Goyal and Sarkar (2016) uses the continuous bag of Words (CBOW) model to capture words in the context of a particular word in the source language and then interprets these words as translations of the target language to acquire language pair word embedding. Then, using a dictionary of word translation pairs, map the relationship between the two languages,

Query Expansion: Query expansion (QE) reformulates the user's original query to enhance the information retrieval effectiveness. For instance: user query "car"; expanded query: "car cars automobile auto". Azad and Deepak (2019) describe the process of expanding query in four steps:

1. *Preprocessing*: This step aims to extract a set of terms from the data source that meaningfully augment the user's original query.
2. *Term weights and ranking*: In this step of QE, weights, and ranks are assigned to query expansion terms obtained after data pre-processing. The input to this step is the user's query and texts extracted from the data sources in the first step. Assigned weights denote the relevancy of the terms in the expanded query and are further used in ranking retrieved documents based on relevancy.
3. *Term selection*: Since the last step produces many expansion terms, this step selects the top-ranked ones.
4. *Query reformulation*: The expanded query is then reformulated to achieve better results when used for retrieving relevant documents. The reformulation is done based on the weights assigned to the individual terms of the expanded query.

2.8 Background Summary

In this chapter, we present a background review related to this work. We start by introducing neural networks and the types of networks related to NLP. We introduce the attention mechanism and the transformer model that BERT uses to generate contextualized word embeddings. We also present BERT's multilingual model, PD divided into intrinsic and extrinsic plagiarism, STS, finishing the chapter with CLIR. In the next chapter, we discuss work related to CLPD and present some existing approaches in the literature.

3 RELATED WORK

In this chapter, we present a literature review of existing approaches for CLPD. First, we present work related to CLIR, then CLPD itself. We present the PAN competition in which we use the same metrics to evaluate this work in Chaper 5. Finally, we end with a summary that summarizes the chapter.

3.1 Cross-language Candidate Document Retrieval

Since it is unfeasible to compare a suspicious document to all documents in the reference collection, the candidate document retrieval phase in an automated PD system is needed to reduce the search space. Irrelevant documents should be discarded in order to reduce processing time. Hence, the accuracy of this step is critical because missing a possible source document will result in it being lost in the process, resulting in non-detection.

Potthast (2012) proposes an approach that uses a collection of queries obtained from suspicious documents d_{plg} in language L to retrieve relevant documents from the D' set in language L'. Keywords are essential in this approach. The benefit of this approach is that it is highly independent of translation systems, and the keywords extracted from the d_{plg} can be translated using a dictionary. However, since any word may have multiple definitions, translating without considering the context ends in an incorrect result. Ehsan, Tompa and Shakery (2016) presents an approach for candidate document retrieval that considers a collection of terms and phrases as content representatives rather than the entire text. The method begins by segmenting the text, extracting keywords from each segment, and translating the words with a dictionary. The system then generates queries based on the translated words in order to retrieve similar documents.

Cedeño (2013) employs a MT system to translate the entire suspicious document into the source document language. To be more specific, the suspicious document d_{plg} is first translated from L to L' by MT to obtain d'_{plg}. The keywords are then extracted from d'_{plg} in order to obtain the D'_{src} candidate set by running the queries on the source set D'. Roostaee, Sadreddini and Fakhrahmad (2020) presented one of the most recent approaches proposing a fusion model that benefits a concept-based and keyword-based representation for retrieving candidate documents. A concept-based representation tackles the issue of vocabulary mismatch, which emerges in the keyword-based representation

of a suspicious document's words and those of the source document. Also, the keyword-based representation addresses issues such as out-of-vocabulary terms in conceptual representation. By covering each other, they increase candidate retrieval accuracy. The proposed fusion outperforms state-of-the-art models (EHSAN; TOMPA; SHAKERY, 2016) for cross-language plagiarism candidate retrieval.

3.2 Cross-language Plagiarism Detection

According to the taxonomy by Alzahrani, Salim and Abraham (2011), cross-language plagiarism is a type of intelligent plagiarism, as the authors plagiarizing a document try to deceive readers by changing original contributions from others to appear as their own work. Figure 3.1 depicts the proposed taxonomy, with translation highlighted. It is important to notice that other intelligent plagiarism can also occur within translation, with paraphrasing of translated texts, for instance.

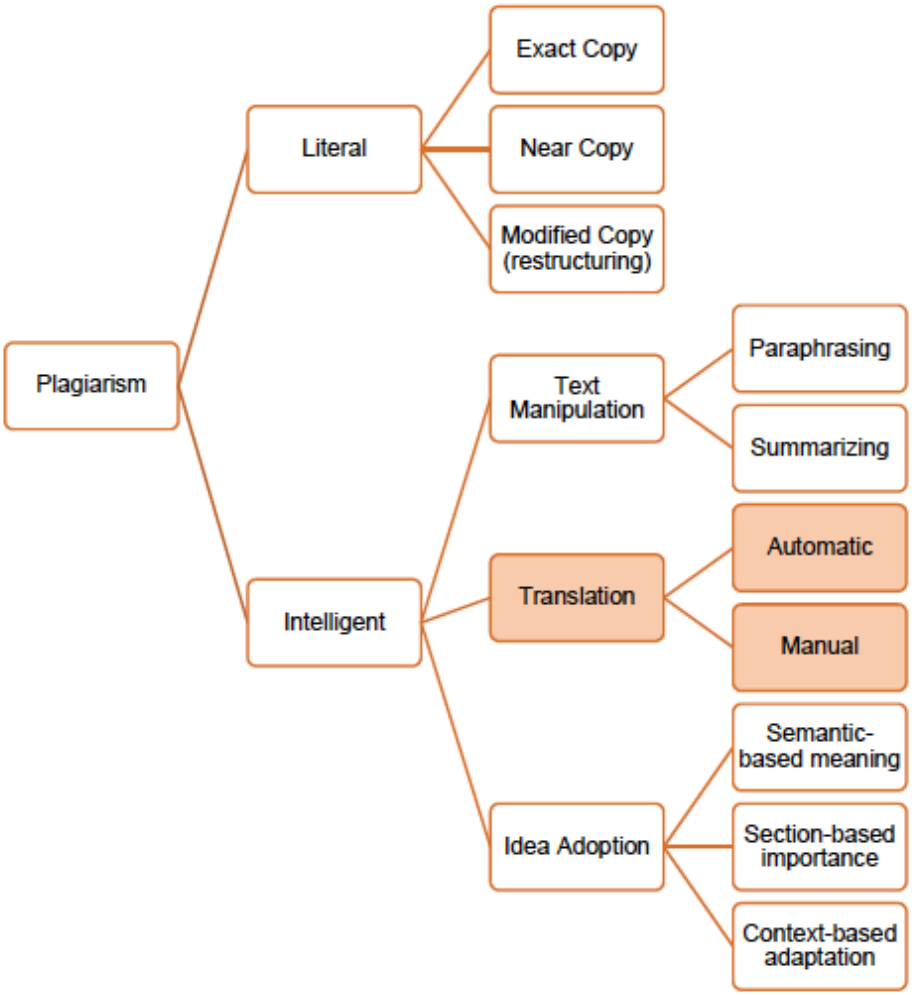
CLPD models can be included in five main categories: Syntax-based, Dictionary-based, Parallel corpora-based, Comparable corpora-based, and Machine Translation (MT) -based (POTTHAST et al., 2011a; DANILOVA, 2013).

Syntax-based: Syntax-based models use syntax information of the languages being analyzed to evaluate similarities among documents in different languages. The most representative model in this category is the Cross-Language Character N-Gram (CL-CnG) (BARRóN-CEDEñO; GUPTA; ROSSO, 2013). CL-CnG compares two textual units under their n-grams vectors representation, which achieves the best results for languages with similar syntactic structures.

Dictionary-based: As the name suggests, dictionary-based models are based on dictionaries and knowledge bases that map words from both languages to a common conceptual space. The Cross-Language Conceptual Thesaurus-based Similarity (CL-CTS) is an example of a dictionary-based approach (FRANCO-SALVADOR et al., 2016). For instance, CL-CTS aims to use abstract concepts from terms in textual units to measure semantic similarity.

Corpora-based: Comparable corpora can be defined as a corpus containing components that are collected using the same sampling frame and similar balance and representativeness (MCENERY, 2012), e.g., the *same proportions* of the texts of the *same genres* in the *same domains* in a range of different languages in the *same sampling period*. In contrast, parallel corpora can be defined as a corpus that contains source texts and their

Figure 3.1: Taxonomy of plagiarism



Source: Alzahrani, Salim and Abraham (2011)

respective translations.

Hence, corpora-based models make use of comparable, parallel, or both types of corpora. Documents are usually mapped to language-independent concept vectors, and their similarities evaluated (BARRÓN-CEDEÑO et al., 2014).

Cross-Language Explicit Semantic Analysis (CL-ESA) and Cross-Language Alignment-based Similarity Analysis (CL-ASA) are examples of comparable corpora-based models and parallel corpora-based models, respectively (POTTHAST et al., 2011a). For the sake of understanding, CLS-ASA attempts to establish if a textual unit uses a bilingual unigram dictionary containing language pairs to interpret another textual unit potentially. CL-ESA is based on the explicit semantic analysis model, which is the definition of a vector-based text based on concepts derived from Wikipedia.

Machine translation (MT)-based: Machine translation-based models first use

an MT system to translate a suspicious document to the target language, which may contain the possible sources. After translation, the analysis can be employed using existing monolingual PD models (BARRóN-CEDEñO; GUPTA; ROSSO, 2013; PEREIRA; MOREIRA; GALANTE, 2010; CER et al., 2017).

3.3 PAN Evaluation Campaigns

The first PAN, the PAN'09 evaluation campaigns, took place in 2009. Eiselt and Rosso (2009) created the first standardized plagiarism detection evaluation framework, the second and third tasks at PAN 2010 and 2011 (POTTHAST et al., 2010; POTTHAST et al., 2011b) consolidated this framework. The goal of this framework was to evaluate the PD process shown in Figure 1.1.

The best approaches at PAN 2012 (LEILEI et al., 2012) and PAN 2014 (SANCHEZ-PEREZ; GELBUKH; SIDOROV, 2015) relied on monolingual techniques. The most common approaches for detecting cross-language plagiarism were translating one of the two corpora (source or suspicious) into the other language and then applying monolingual techniques. Pereira, Moreira and Galante (2010) suggested a classification system that involves first translating, normalizing, and dividing the text into subdocuments. Pataki (2012) detects translated plagiarised texts by chunking sentences and looking for all possible translations by specifying a resemblance measure dependent on the number of common terms. Muhr et al. (2010) proposed an approach using the MT method as a pre-processing stage, replacing each word with up to five translation candidates. After the words have been replaced, the documents are treated similarly to the English source documents.

Ehsan, Shakery and Tompa (2019) proposes a two-step retrieval model. The first step selects candidate sentences with a strong recall goal. The second step refines the results with a text alignment model connecting related segments to filter out false-positive detections. Roostaee, Fakhrahmad and Sadreddini (2020) presented a two-level method that aims to understand both syntactic and semantic knowledge to identify cross-language plagiarism. A vector space model with a multilingual word embeddings-based dictionary and a local weighting technique is used to identify a minimal set of highly potential candidate fragment pairs between suspicious and source documents. Roostaee, Fakhrahmad and Sadreddini (2020) shows that candidate pairs can be identified by selecting the best translation of each term rather than selecting possible translations. In the pairwise

analysis, texts are modeled using graph-of-words representations to understand the words and their relationships. Bakhteev et al. (2019a) presents a system for PD in the English-Russian language pair with a monolingual approach by reducing the problem to a monolingual task using a MT system. The approach is then divided into two stages: candidate retrieval and document comparison stage. The first stage consists of finding the most relevant documents from the collection given a suspicious document. The second stage consists of a document comparison algorithm based on an aggregation of semantically close words into word classes followed by sentence embeddings similarity analysis.

3.4 Related Work Summary

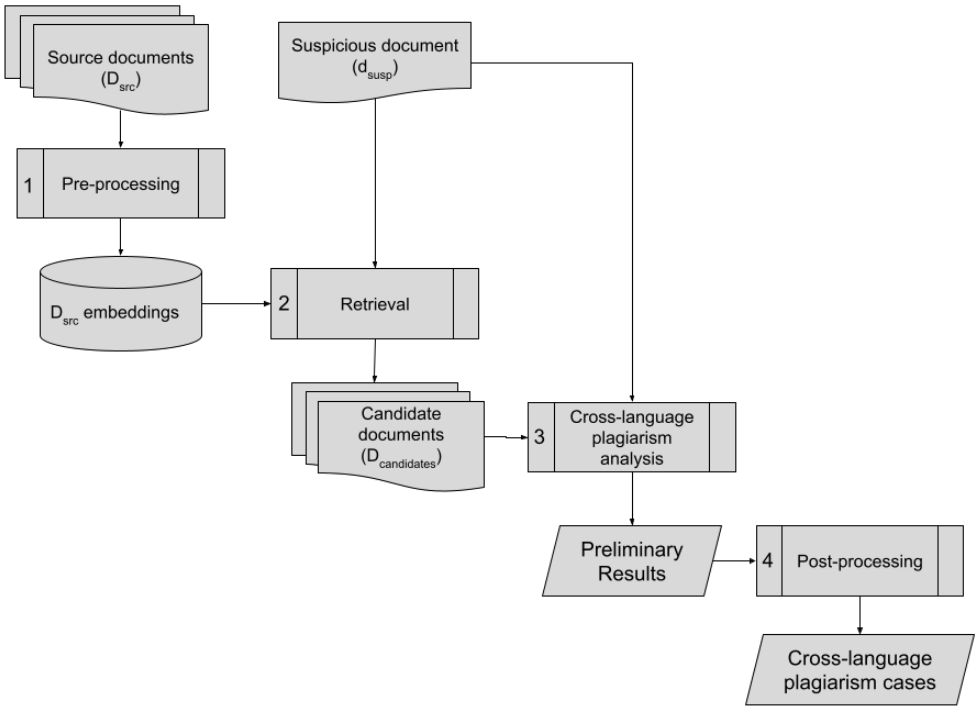
In this chapter, we presented recent work related to CLIR and CLPD. Then, we discussed the importance of a CLIR system in the CLPD task since it is not feasible to compare a suspicious document against the entire collection of source documents. We presented CLIR techniques such as a fusion model that benefits from concept and keyword-based representation. Then, we discussed the methods used in CLPD and that there are five main categories: syntax-based, dictionary-based, parallel and comparable corpora-based, and MT-based. These existing methods have some limitations, such as translating the entire corpus of suspicious documents using a translation system. However, to the best of our knowledge, none of the existing approaches exploit contextualized word embeddings for CLPD. This is the gap we address in this work with the approach proposed in the next chapter.

4 CROSS-LANGUAGE PLAGIARISM DETECTION WITH CONTEXTUALIZED WORD EMBEDDINGS

This chapter introduces our proposed approach – cross-language plagiarism detection with contextualized word embeddings (CLPD-CWE). Given a reference corpus C_{source} of source documents and a corpus $C_{suspicious}$ of suspicious documents, CLPD-CWE aims to detect all plagiarised passages $P_{suspicious} \in C_{suspicious}$ from $P_{source} \in C_{source}$. To accomplish this task, we employ contextualized multilingual word embeddings in order to detect cross-language plagiarism. This method tries to overcome the problems encountered in CLP described in Section 3.2 without the use of MT systems. Instead, the contextualized word embeddings generated by M-BERT are used. CLPD-CWE is divided into the four main phases depicted in Figure 4.1 together with their inputs and outputs. Each of these phases is detailed in the following sections.

1. *Pre-processing*: Initially, we store the C_{source} corpus of source documents with their respective sentence embeddings as a data frame with rows and columns. It is noteworthy that this step is done only once and not every time a new suspicious

Figure 4.1: Overall workflow for CLPD-CWE



Source: The Author

document is analyzed.

2. *Candidate retrieval*: During this phase, we extract a set of $K_{suspicious}$ keywords from $P_{suspicious} \in C_{suspicious}$. Then, we retrieve a set of $D_{candidate}$ documents with $Sim(P_{source}, P_{suspicious}) \geq \Delta_{similarity}$ where $P_{suspicious} \cap K_{suspicious} \neq \emptyset$.
3. *Cross-language similarity analysis*: During this phase, each passage $P_{suspicious}$ is compared against its respective $D_{candidate}$ document set in order to identify whether the suspicious passage is plagiarised or not.
4. *Post-processing - Plagiarism results*: This phase consists of generating the output with the identified plagiarism cases detected by CLPD-CWE and removing possible false-positive cases.

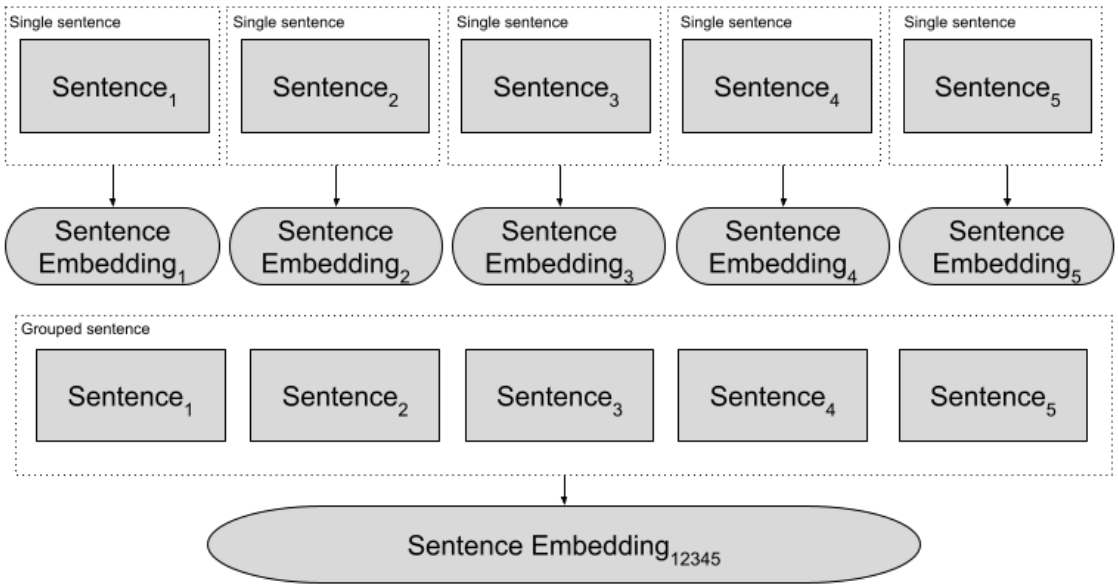
4.1 Pre-processing

The pre-processing phase is divided into three smaller tasks. Given a document d_{source} from C_{source}, we first identify its language. Secondly, in order to tokenize the document into sentences, we use an unsupervised multilingual model for detecting sentence boundaries (KISS; STRUNK, 2006). Thus, knowing the language of the document is required so that a language-specific model can be applied. Then, we join the tokenized sentences so that they are at least δ_w words long. For example, in Figure 4.2, we consider that each sentence has 100 words. If we use $\delta_w = 100$, we will generate sentence embeddings for each sentence. However, if we use $\delta_w = 500$, we will be generating only one sentence embedding. Sentence grouping decreases the number of sentences stored in the data frame. Finally, we generate the embeddings of the grouped sentences.

It is noteworthy that δ_w is an important threshold. For instance, if δ_w is equal to the number of words in a document d_{source}, only a single sentence embedding is generated for the entire document and this can lead to loss of information. On the other hand, if we generate a sentence embedding for each word, we lose the context in which the word is inserted (i.e., it would just be non contextualized word embeddings). Once the sentence embeddings are generated, they are stored according to the format shown in Table 4.1 which is composed of:

- **Docno**: Is the document identifier, in Table 4.1 the two sentences are part of the same document.
- **Sentence**: A sentence containing at least δ_w words.

Figure 4.2: Pre-processing sentence grouping



Source: The Author

Table 4.1: Data storage format with two sentences

Docno	Sentence	Language	SentenceEmbedding
1	Je suis le dernier de cette Haute Assemblée qui soit né avant la première guerre mondiale et qui puisse se souvenir de cet événement qui a marqué...	French	denseVector
1	J'ai personnellement vécu trois autres guerres mondiales, car la guerre froide ne fut en fait rien d'autre que la troisième guerre mondiale. J'ai...	French	denseVector

Source: The author

- **Language**: Sentence language.
- **SentenceEmbedding**: The sentence embedding as a dense vector. It is noteworthy that the size of the dense vector depends exclusively on the transformer network model used to generate them.

4.2 Candidate Retrieval

Albeit it is ideal, comparing the suspicious document with the entire corpus of source documents is too costly. Therefore, this phase aims to identify $D_{candidate}$ documents that are likely sources of eventual plagiarism cases. Initially, we select the keywords that are representative of the suspicious document. For this, we extract sentence embeddings with SBERT (REIMERS; GUREVYCH, 2019) to obtain a document-level representation. Then, for each keyword n-gram, we generate its word embedding. Finally, we use the cosine similarity to find the most similar keyword n-grams in the text

(GROOTENDORST, 2020). The cosine similarity for vectors A and B is given by Equation 4.1.

$$Sim(A, B) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \, \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{4.1}$$

To have a greater diversity of keywords, we use maximal marginal relevance (MMR) (CARBONELL; GOLDSTEIN, 1998) to diversify the chosen keywords (Equation 4.2). A keyword has high marginal relevance if it has a similarity with the document and contains minimal similarity to previously selected keywords.

$$MMR \overset{\text{def}}{=} \underset{D_i \in R \setminus S}{\arg\max} [\lambda(Sim_1(D_i, Q) - (1 - \lambda) \max_{D_j \in S} Sim_2(D_i, D_j))] \tag{4.2}$$

where Q is the query (sentence as a dense vector), D is a set of keywords related to query Q, S is a subset of queries in R already registered, $R \setminus S$ is a set of unselected keywords in R and $\lambda \in [0, 1]$ is a constant for diversification of results. The greater the λ, the more diversified the keywords are. Sim_1 and Sim_2 are the cosine scores calculated by Equation 4.1.

Let $K_{suspicious}$ be the set of keywords obtained in the previous keyword extraction task. We create sentence embeddings $SE_{suspicious}$ for every passage $P_{suspicious} \in C_{suspicious}$ if $k_{suspicious} \in P_{suspicious}$, where $k_{suspicious} \in K_{suspicious}$.

To illustrate, consider the following set of two keywords: {*irresponsibility, education*} and the following sentences s_1, s_2 respectively: {*Investments in* ***education*** *and equity will increase student learning and graduation rates and in turn secure our nation's economic future*, *The cat is black*}. We only generate sentence embedding for sentence s_1 since it has the keyword *education*.

With the list of sentence embeddings from the suspicious document, we generate a matrix containing the similarity values between the list of suspicious sentence embeddings and each d_{source} document embedding generated earlier in the pre-processing phase. Then, the number of passages in which the $Sim(P_{source}, P_{suspicious}) \geq \Delta_{similarity}$ are added up. As a result, we have the number of sentences within which the suspicious document has at least $\Delta_{similarity}$ —- yielding a document ranking concerning the other documents in the source corpus. For instance, a source document with a hundred similar sentences is more likely to be the source of a more serious plagiarism case than a document with five similar sentences. Therefore, this document is a candidate to be analyzed

in the next phase.

4.3 Cross-language Similarity Analysis

Once we have a collection of $D_{candidate}$ documents for the suspicious document, we move on to the cross-language similarity analysis phase. Given a suspicious document with $P_{suspicious}$ passages and a collection of $D_{candidate}$ documents, the goal of this phase is to identify if there are any plagiarized $P_{suspicious}$ passage from $D_{candidate}$ set.

Firstly, we identify the suspicious document language and tokenize it into sentences, applying the same unsupervised multilingual model used in Pre-processing 4.1. Unlike what was done in the pre-processing phase, we have δ_c to determine the minimum **character** length of a sentence.

In order to select the most similar sentences, we compare all suspicious sentence embeddings against all other candidate sentence embeddings and return a list with the pairs satisfying $Sim(A, B) \geq \delta_{similarity}$. To exemplify, consider a suspicious sentence $S_{suspicious}$ that has similarity scores of *0.6, 0.7, and 0.8* for the candidate sentences s_1, s_2, and s_3, respectively. If we use $\delta_{similarity} = 0.8$, only the candidate sentence s_3 is considered plagiarized since it has $\delta_{similarity} \geq 0.8$. The other sentences s_1 and s_2, are discarded. However, if we use $\delta_{similarity} = 0.9$, all three sentences would be discarded. For each candidate document, the output of the similarity analysis is a 4-tuple containing the following characteristics:

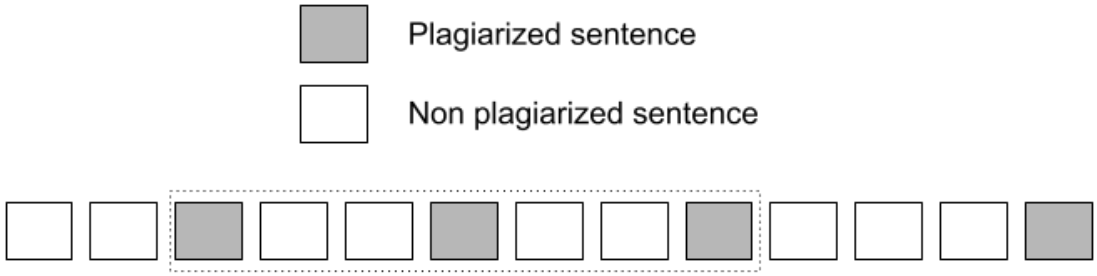
- **Plagiarized offset**: initial character offset where the plagiarism case occurred in the plagiarised document.
- **Plagiarized length**: length of the plagiarised passage in the number of characters.
- **Candidate offset**: initial character offset where the case of plagiarism occurred in the source document.
- **Source offset**: length in the number of characters of the plagiarised sentence in the candidate document.

For example, consider the 4-tuple $(200, 150, 1530, 200)$ as the output of cross-language similarity analysis. The 4-tuple has identified a plagiarism case that starts at the 200^{th} character and has a length of 150 characters in $C_{suspicious}$. This 150-character passage was plagiarised from d_{source} document starting at the 1530^{th} character, where the original passage has 200 characters in length.

4.4 Post-processing

This phase aims at removing false-positive cases and joining contiguous plagiarism cases to decrease the granularity score. Briefly, the granularity score is a measure that tells whether a case of plagiarism was identified as a whole or in small parts. It is detailed in Section 5.2.3.

Figure 4.3: Tolerance threshold



Source: The Author

Consider Figure 4.3, where cross-language similarity analysis identified three cases of plagiarism, and all of them are two sentences away from each other. For cases like this, we create $\Gamma_{tolerance}$ parameter, where the system tolerates that two plagiarism cases can be up to $\Gamma_{tolerance}$ away from each other. In the example, if we use $\Gamma_{tolerance} = 2$, the five plagiarism cases are identified as a single case. However, if $\Gamma_{tolerance} = 1$, the five cases are identified as distinct plagiarism cases.

Finally, we generate a document containing plagiarism the output with the plagiarism cases identified for each suspicious document in $C_{suspicious}$. This document has the following characteristics shown in Figure 4.4. We use this format since it is the same format used to evaluate multilingual plagiarism detection systems in the PAN competition.

Figure 4.4: Output format for the detected plagiarism cases

```
<document reference="susp.txt"> <!-- file name of the suspicious document        -->
<feature
  name="detected-plagiarism"    <!-- type of the plagiarism annotation            -->
  this_offset="5"               <!-- char offset within the suspicious document -->
  this_length="1000"            <!-- number of chars beginning at the offset    -->

  source_reference="src1.txt"   <!-- file name of the source document             -->
  source_offset="100"           <!-- char offset within the source document      -->
  source_length="1000"          <!-- number of chars beginning at the offset     -->

/>
...                             <!-- more detections in this suspicious document -->
</document>
```

Source: The Author

5 EVALUATION

In this chapter, we evaluate the proposed approach – CLPD-CWE. We carried out experiments using three different datasets presented in Section 5.1.2. We chose these datasets as they already have results obtained using different approaches. For each dataset, we compare our results with the results obtained by baseline models.

We start by describing the resources used in the experiments (Section 5.1). Our results are presented in Section 5.3.

5.1 Materials and Methods

This section describes all resources and frameworks employed to obtain the final results described in Section 5.3.

5.1.1 Trec eval

Trec eval is a method for evaluating rankings of documents (or other types of information) sorted by a similarity function. The assessment is based on the following two files: The first, known as *qrels* (query relevance), contains the ground truth annotations, i.e., in our context, it contains the source documents for the cases of plagiarism. The second file includes the ranked list of candidate source documents retrieved by the IR system.

A list of documents considered important for each query can be found in the *qrels* file. This relevance assessment is made by humans who manually pick documents to be collected when a specific query is run. This file can be conceived as the "right response", and the documents obtained by the IR system should come as close to it as possible. It is written in the following format:

query-id 0 document-id relevance

The fields **query-id** and **document-id** are alphanumeric sequences that define the query and the judged document, respectively. The field **relevance** is a number that indicates the degree of relevance between the document and query (0 for not relevant and 1 for relevant). In practice, only the relevant judgments are used to compute the evaluation metrics. A blank space or tabulation is used to distinguish the fields.

The results file contains a ranking of documents provided by the IR system for each query. This is the file that *trec eval* will test using the "right response" from the *qrels* file. The following is the format of the results file:

query-id Q0 document-id rank score RUN

The **query-id** field contains an alphanumeric sequence that identifies the query. The second field, which currently has a value of "Q0", is ignored by *trec eval* but must be included in the results file. The field **document-id** is an alphanumeric sequence used to identify the document that is retrieved. The field **rank** is an integer value that reflects the document's ranking position; however, *trec eval* also ignores this field. The field **score** is an integer or float value that indicates the degree of similarity between the document and the query, the higher, the more relevant the document. The last value, "RUN" is only used to identify the execution run (this value is also shown in the output). It is noteworthy that Mean average precision (mAP) is the metric we are the most interested in, which is better explained in Section 5.2.5. Furthermore, we use *Trec Eval* as baseline during the evaluation phase.

5.1.2 Datasets

During the evaluation of the method, we used four datasets described next. The details of the datasets are in Table 5.1 and the statistics of each source dataset are in Table 5.2.

Table 5.1: Datasets details

–		#Docs	Size in MB	Documents in					
				English	Portuguese	French	Spanish	Russian	German
ECLaPA	**Suspicious**	300	89	300	0	0	0	0	0
	Source	348	11	0	174	174	0	0	0
PAN-PC-12 Test-set	**Suspicious**	500	65.8	0	0	0	263	0	237
	Source	3000	402	3000	0	0	0	0	0
CrossLang Test-set	**Suspicious**	316	16.3	0	0	0	0	316	0
	Source	1343	45.8	1343	0	0	0	0	0

1. *The ECLaPA Test Collection multilingual subset*[1]: The ECLaPA test set is divided into two corpora: one for monolingual plagiarism cases and the other for multilingual plagiarism cases. The plagiarism cases of both corpora are the same. The monolingual records are all written in English. The suspicious documents are written in English in the multilingual corpus, but the source documents are written

[1]<http://www.inf.ufrgs.br/~viviane/ECLaPA.zip>

in Portuguese or French. ECLaPA's author, Rafael C. Pereira, developed a script that randomly extracts passages from a Portuguese or French text, locates the corresponding English passages, and incorporates them into an English document in order to simulate plagiarism between these languages.

2. *Spanish-English and German-English subset of PAN-PC-12 text-alignment corpus*[2]: This dataset contains German-English and Spanish-English language partitions used for the text-alignment task in PAN-PC-12 including documents that have been automatically plagiarized as well as documents that have been manually plagiarized. The former was created using a so-called random plagiarist, a computer program that generates plagiarism based on a set of criteria, while the latter was created using crowdsourcing by Amazon's Mechanical Turk.
3. *English-Russian CrossLang Dataset*[3]: Bakhteev et al. (2019b) used 100k Wikipedia articles in English to generate the source document collection and created the suspicious documents using random samples of Wikipedia articles in Russian.

Table 5.2: Dataset statistics

-	**#Documents**	**#Sentences**	**#Unique Words**
ECLaPA Source corpus (English)	348	81.176	16.686.663
PAN-PC-12 Source corpus (English)	3000	312.981	32.141.696
CrossLang Source corpus (English)	1343	101.506	3.597.293

The three corpora described above have all cases of plagiarism annotated (Figure 4.4), thus making it possible for us to verify the cases of plagiarism detected correctly.

5.1.3 Tools

To implement CLPD-CWE, we used *Jupyter Notebook*[4]. *Jupyter Notebook* is an open-source application that allows users to write and share Python code, equations, visualizations, and text-based documents. Our *Jupyter Notebooks* are stored in *Google*

[2]<https://doi.org/10.5281/zenodo.3715851>
[3]<http://tiny.cc/cl_ru_en>
[4]<https://jupyter.org/>

Colaboratory[5], allowing easy code editing with free GPU access (upon availability). The *Google Colaboratory* virtual environment can be used up to 12 hours; after this the computer session is restarted. The code *Google Colaboratory* virtual machine (VM) specifications are stated in Table 5.3.

Table 5.3: Google Colab Virtual Machine Specifications

Machine name	n1-highmem-2
Virtual Cores	2vCPU @ 2.2Ghz
Memory	13GB RAM
Disk space	64GB
Idle cut-off	90 minutes
Maximum	12 hours
GPU	Tesla K80 2496 CUDA cores 12GB GDDR5 VRAM

We use *Pandas*[6] for the pre-processing phase and *NLTK*[7] for sentence tokenization. *Pandas* is a free software library written for the Python programming language. It includes data structures and operations for manipulating numerical tables and time series. *NLTK* is a suite for text processing such as tokenization, stemming, tagging, and parsing. Together with *Pandas* and *NLTK*, use used *Numpy*. *Numpy* is a Python library that adds support for large, multidimensional arrays and matrices as well as a large number of high-level mathematical functions.

5.1.4 Detection Parameters

During the CLPD-CWE evaluation, some parameters are defined to achieve the results presented in this chapter. Table 5.4 shows the parameters values used for the four phases: **Pre-processing**, **Candidate Retrieval**, **Cross-language similarity analysis**, **Documents Post-processing**. It is important to note that all datasets are analyzed with the same parameters, and we use *DistilBERT multilingual cased* to generate word embeddings. For the sake of clarification, we use SBERT to generate sentence embeddings which in turn uses DistilBERT multilingual cased, a multilingual BERT model (M-BERT). Furthermore, the parameter $\delta_c = 150$ may not be ideal for languages such as Chinese and Hindi since this quantity would be insufficient to identify cases of plagiarism

[5]<https://colab.research.google.com/>
[6]https://pandas.pydata.org/
[7]https://www.nltk.org/

in these languages. Another critical point is that the DistilBERT multilingual cased is better on high resource language but worse in languages with scarce resources such as Urdu and Arabic.

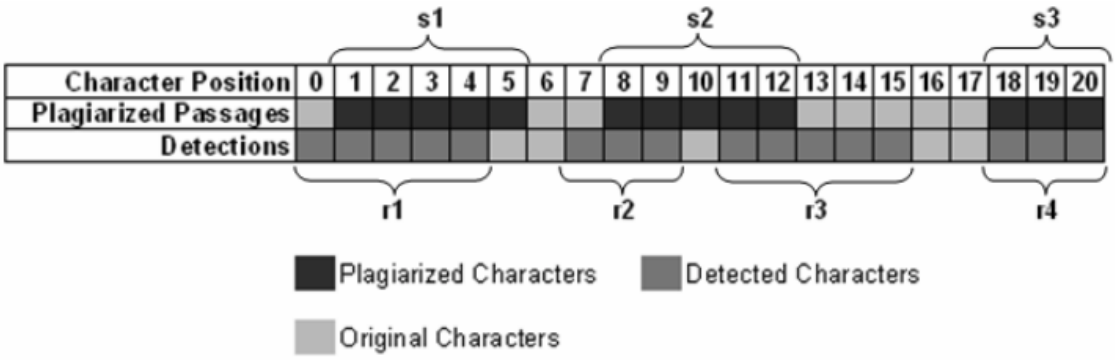
Table 5.4: CLPD-CWE – parameters used in the experiments

Pre-processing Parameters	
δ_w (min. sentence length in words)	400
Candidate Retrieval Parameters	
λ (keyword diversification)	0.7
$\Delta_{similarity}$ (CLIR sentence similarity)	0.7
Cross-language Similarity Analysis Parameters	
δ_c (min. sentence length in characters)	150
$\delta_{similarity}$ (sentence similarity)	[0.5,0.95]
Post-processing Parameters	
$\Gamma_{tolerance}$	5

5.2 Evaluation Metrics

We use PD metrics proposed by (EISELT; ROSSO, 2009): recall, precision, granularity, and plagdet. Figure 5.1 depicts an example to help explain the metrics. *s1, s2*, and *s3* are plagiarised passages to be detected. However, the plagiarism detector reported four of them, *r1, r2, r3*, and *r4* (PEREIRA; MOREIRA; GALANTE, 2010). In the following sub-sections we describe how to calculate each metric based on the given example.

Figure 5.1: Plagiarized passage and detections



Source: Pereira, Moreira and Galante (2010)

5.2.1 Recall

Let *S* be the set of all plagiarized passages, and let s be a plagiarized passage from document *d*. Let *R* be the set of all detections, and let *r* be a detection caught by the plagiarism detector. The number of plagiarized characters found by the plagiarism detector is the recall measure. The plagiarism detector recognizes all three plagiarized passages in the example given in Figure 5.1. (*s1, s2*, and *s3*) albeit it does not detect every plagiarized character. Equation 5.1 shows the formula for calculating recall, where s_i is a plagiarized passage from the set *S*.

$$Recall(S, R) = \frac{1}{|S|} \sum_{s \in S} \frac{\bigcup_{r \in R}(s \sqcap r)}{|s|} \tag{5.1}$$

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

5.2.2 Precision

The number of characters identified by the plagiarism detector that are actually plagiarized is measured by precision. The plagiarism detector marked all three plagiarized passages in the example shown in Figure 5.1. (*s1, s2*, and *s3*). Not all of the detected characters, however, were plagiarized. Equation 5.2 gives the formula for calculating the precision measure, where r_i denotes a detection from the set *R*.

$$Precision(S, R) = \frac{1}{|R|} \sum_{s \in S} \frac{\bigcup_{s \in S}(s \sqcap r)}{|r|} \tag{5.2}$$

5.2.3 Granularity

Granularity characterizes the algorithm's ability to detect a plagiarism case $s \in S$ identified in its entirety or in fragments. Figure 5.2 shows that the plagiarism detector reported four detections despite the fact that there were actually three plagiarized passages

to be identified. Equation 5.3 shows how to calculate the granularity measure.

$$Granularity(S, R) = \frac{1}{S_r} \sum_{s \in S_r} |Rs| \tag{5.3}$$

Where $S_r \subseteq S$ denotes cases discovered by R detections, and $R_s \subseteq R$ are the detections of a given s: $S_r = \{s | s \in S \wedge \exists r \in R : r \ detects \ s\}$ and $R_s = \{r | r \in R \wedge r \ detects \ s\}$. The domain of ***Granularity(S,R)*** is *[1, |R|]*, with *1* denoting the optimal one-to-one correspondence and |*R*| denoting the worst-case scenario, in which a single $s \in S$ is detected repeatedly.

5.2.4 Overall Score

A partial ordering of plagiarism identification algorithms is possible thanks to precision, recall, and granularity. To obtain an overall score, these metrics are combined into the *plagdet* score, as seen in Equation 5.4.

$$Plagdet(S, R) = \frac{F_1}{log_2(1 + Granularity(S, R)))} \tag{5.4}$$

Where F_1 is the weighted harmonic mean of precision and recall presented in Equation 5.5. The logarithm is used to reduce the effect of granularity on the total ranking.

$$F_1(S, R) = \frac{2 * Recall(S, R) * Precision(S, R)}{Recall(S, R) + Precision(S, R)} \tag{5.5}$$

5.2.5 Mean Average Precision

The mean average precision (mAP) is a widely used metric for assessing the performance of models performing information retrieval and object detection tasks. A common task in information retrieval is for a user to submit a query to an index and then retrieve results that are very close to the query. To explain mAP we define a typical retrieval task by defining firstly $precision$ as Equation 5.6. When it comes to information retrieval, the definition of precision is slightly different from the one described in Subsec-

tion 5.2.2.

$$precision = \frac{|relevantdocuments| \cap |retrieveddocuments|}{|retrieveddocuments|} \tag{5.6}$$

In other words, $precision$ is evaluated by considering all the retrieved documents. However, it can also be evaluated at a certain number of retrieved documents (commonly known as cut-off rank). The model is only evaluated by considering the top-most results. The measure is called precision at k or $P@K$.

For instance, let us consider a calculation for precision with three ground truth positives (GPT), also known as the relevant documents. First, we shall define the following variables:

- Q to be the user query.
- G to be a set of labeled data in the database.
- $IR_{score}(i, j)$ to be a score function to show how similar object i is to j.
- G' an ordered set of G according to score function $IR_{score}(i, j)$.
- k to be the index of G'.

After calculating the $IR_{score}(i, j)$ for each of the documents with Q, we can sort G and get G'. For instance, let say the model returns the G' as Figure 5.2 illustrates we get the following results:

- $P@1 = 1/1 = 1$
- $P@2 = 1/2 = 0.5$
- $P@3 = 1/3 = 0.33$
- $P@4 = 2/4 = 0.5$
- $P@5 = 3/5 = 0.6$
- $P@n = 3/n$

Figure 5.2: Information retrieval returned sorted query results G'

Source: Towards Data Science <https://bit.ly/2QKorWO>

We can now determine the Average Precision after being familiar with $P@k$ set in Equation 5.7, where the total number of ground truth positives is GTP, n is the total number of documents we are interested in, and the relevance function is $rel@k$. The

relevance function is an indicator function that returns 1 if a document at rank k is relevant and 0 if it is not.

$$AP@n = \frac{1}{GTP} \sum_{k}^{n} P@k \times rel@k \tag{5.7}$$

Hence, after determining a corresponding AP for each query Q. A user can run as many queries against this named database as he or she wants. The mAP (Equation 5.8) is simply the mean of all of the queries performed.

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{5.8}$$

5.3 Experimental Results

This section presents the results obtained after using CLPD-CWE on the datasets described in subsection 5.1.2. For each one of the datasets, we performed two different evaluations. More specifically, in the first moment, we evaluate our candidate retrieval phase using *trec eval*, described in Section 5.1.1. After that, we performed a plagiarism analysis using the metrics from the PAN evaluation campaign (described in Section 5.2). The code used in the experiments is publicly available in <https://bit.ly/3xWFp4N>.

5.3.1 ECLaPA - Experimental Results

The results of evaluating the ECLaPA test set are presented in this subsection. It is worthy of mention that this test set includes two corpora: one for monolingual plagiarism cases and the other for cross-language plagiarism cases. We are only analyzing the cross-language one in this study. We obtained the following results shown in Table 5.5 using *trec eval*, described in Section 5.1.1 during the information retrieval phase.

After the candidate document retrieval phase, we proceed to the cross-language similarity analysis phase. We analyzed results with different $\delta_{similarity}$ thresholds, presented in Figure 5.3. Moreover, we compare our best result ($\delta_{similarity} = 0.85$) obtained with the CLPD-CWE approach with the results obtained by Pereira, Moreira and Galante (2010) for the same multilingual partition of the ECLaPA dataset.

Table 5.6 shows that recall, accuracy, and plagdet are all significantly improved

Table 5.5: Results of CLIR - ECLaPA Dataset

Trec eval metric	**Result**
Total number of evaluated queries	232
Total number of retrieved documents	942
Total number of relevant documents (according to the qrels file)	490
Total number of relevant documents retrieved (in the results file)	410
Mean average precision	0.7403
Precision of the first R documents, where R are the number of relevants	0.7244
Precision of the 5 first documents	0.3457
Precision of the 10 first documents	0.1750

regarding the results obtained by Pereira, Moreira and Galante (2010). The granularity, on the other hand, is worse. This happened is due to the value of $\Gamma_{tolerance} = 5$, which combines multiple instances of plagiarism into a single instance, thus increasing the granularity.

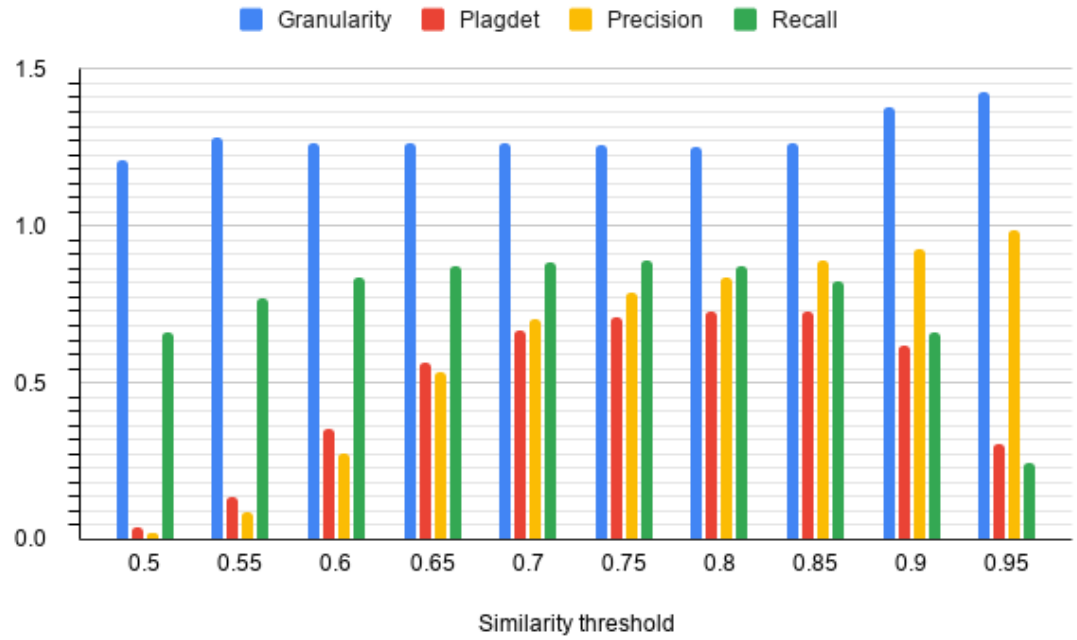
Table 5.6: Results of CLPD-CWE and baseline on the ECLaPA Dataset

Approach	**ECLaPA Dataset**			
	recall	precision	granularity	plagdet
Pereira, Moreira and Galante (2010)	0.3580	0.5684	**1.0**	0.4393
Proposed approach (CLPD-CWE)	**0.8230**	**0.8877**	1.2624	**0.7251**

5.3.2 PAN-PC-12 - Experimental Results

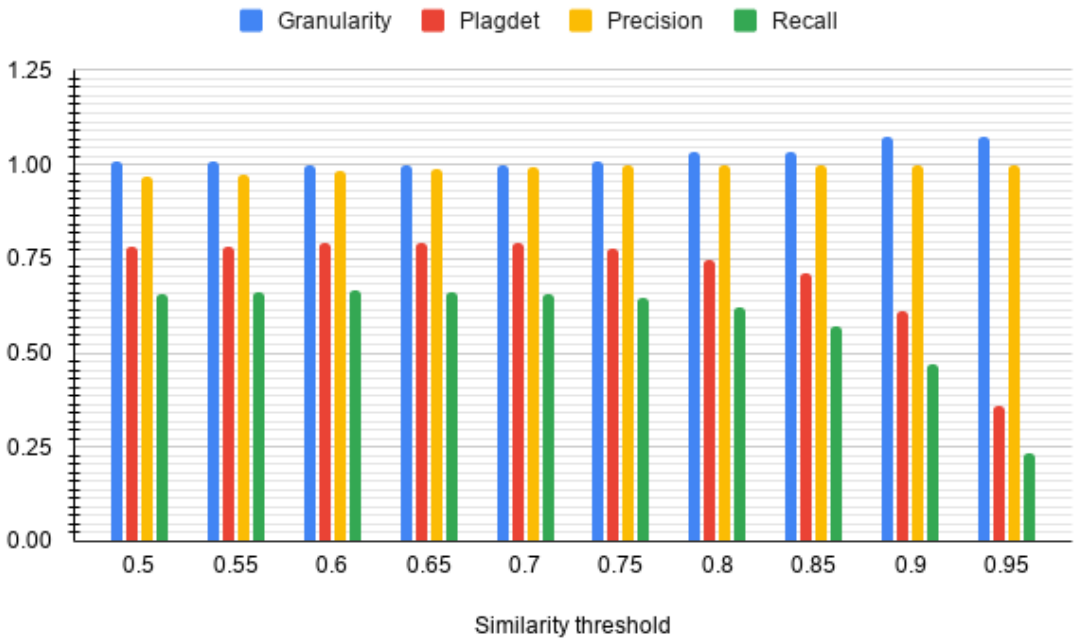
This subsection presents the results obtained by analyzing both the Spanish and German partition of PAN-PC-12. Initially, we present the results obtained in the CLIR phase in Table 5.7.

Figure 5.3: Results for ECLaPA dataset with different similarity thresholds.



Source: The Author

Figure 5.4: Results for PAN-PC-12 Spanish subset with different similarity thresholds



Source: The Author

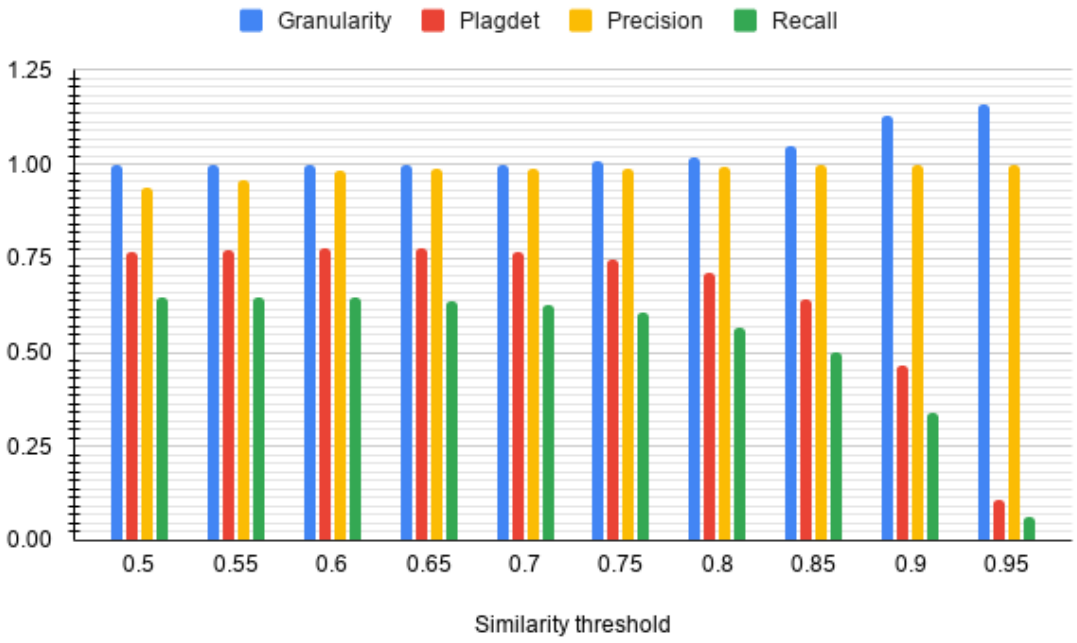
The results obtained for the Spanish and German dataset are shown in Figure 5.4 and 5.5. Furthermore, we compare our best results for the Spanish partition ($\delta_{similarity} = 0.65$) and the German partition ($\delta_{similarity} = 0.6$) with the results obtained by Roostaee,

Table 5.7: Results of CLIR - PAN-PC-12 Dataset

Trec eval metric	**Result**	
	Spanish partition	**German partition**
Total number of evaluated queries	237	263
Total number of retrieved documents	266	337
Total number of relevant documents (according to the qrels file)	237	263
Total number of relevant documents retrieved (in the results file)	159	172
Mean average precision	0.6646	0.6473
Precision of the first R documents, where R are the number of relevants	0.6582	0.6464
Precision of the 5 first documents	0.1342	0.1293
Precision of the 10 first documents	0.0671	0.650

Sadreddini and Fakhrahmad (2020), the CLPD state-of-the-art.

Figure 5.5: Results for PAN-PC-12 German subset with different similarity thresholds



Source: The Author

The results obtained in for cross-language similarity analysis are shown in Tables 5.8 and 5.9, respectively. Our proposed approach outperformed the baseline for both German and Spanish in terms of precision, granularity, and plagdet. On the other hand, the baseline has a better recall. This low recall obtained by CLPD-CWE is a direct consequence of the candidate retrieval phase. For example, a case of plagiarism may have been made from several source documents. By not selecting one of the source documents dur-

ing the candidate retrieval phase, the plagiarized characters associated with it will never be selected, decreasing the recall value. Although this work aims only to use contextualized word embeddings, to increase the recall value, we would use an MT system for the candidate retrieval phase.

Table 5.8: Comparison of the proposed approach on PAN-PC-12 Spanish partition

Approach	**PAN-PC-12 Spanish partition**			
	recall	precision	granularity	plagdet
Ehsan, Shakery and Tompa (2019)	0.9496	0.2620	**1.0**	0.4106
Roostaee, Fakhrahmad and Sadreddini (2020)	**0.9746**	0.5560	**1.0**	0.7080
Proposed approach (CLPD-CWE)	0.6624	**0.9904**	**1.0**	**0.7939**

Table 5.9: Comparison of the proposed approach on PAN-PC-12 German partition

Approach	**PAN-PC-12 German partition**			
	recall	precision	granularity	plagdet
Ehsan, Shakery and Tompa (2019)	0.8721	0.2960	**1.0**	0.4419
Roostaee, Fakhrahmad and Sadreddini (2020)	**0.9228**	0.5177	**1.0**	0.6633
Proposed approach (CLPD-CWE)	0.6442	**0.9820**	**1.0**	**0.7780**

5.3.3 CrossLang - Experimental Results

This subsection presents the results obtained by analyzing the CrossLang dataset. Initially, we present the results obtained in the CLIR phase in Table 5.10.

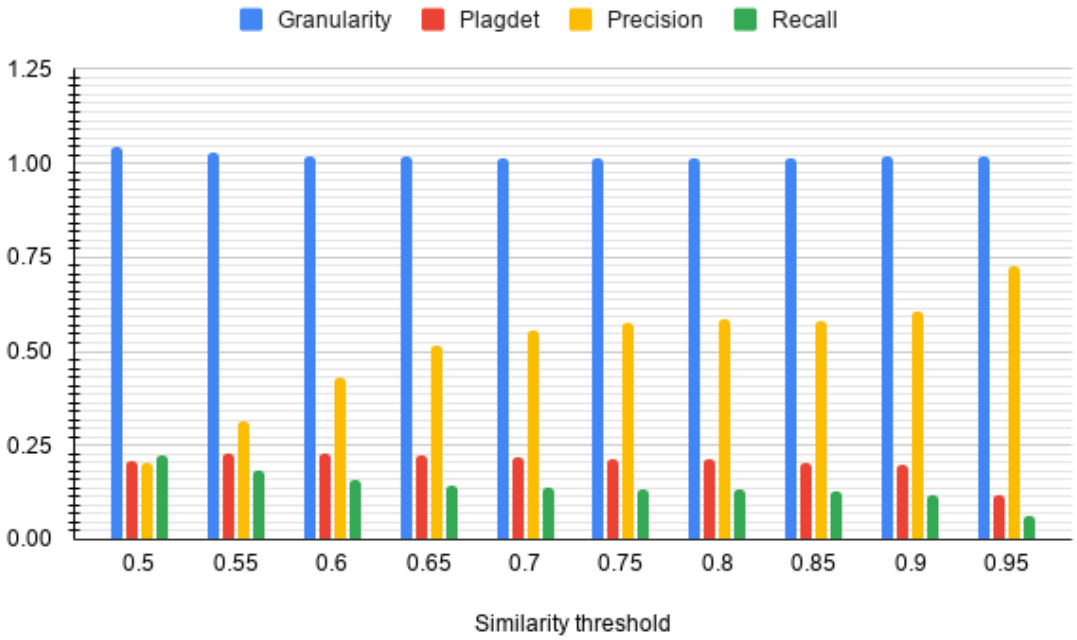
Table 5.10: Results of CLIR - CrossLang Dataset

Trec eval metric	**Results**
Total number of evaluated queries	316
Total number of retrieved documents	454
Total number of relevant documents (according to the qrels file)	1378
Total number of relevant documents retrieved (in the results file)	445
Mean average precision	0.4335
Precision of the first R documents, where R are the number of relevants	0.4334
Precision of the 5 first documents	0.2816
Precision of the 10 first documents	0.1408

We can see that 933 documents were not found in the candidate retrieval phase, which directly impacts the recall of our final result shown in Figure 5.6. We attribute these poor results to the annotations in the CrossLang dataset. For instance, CrossLang has plagiarism annotations as short as 46 characters. One of the parameters used for PD

is $\delta_c = 150$, the minimum number of characters to be considered plagiarism. We use this threshold to remove false-positive cases in short sentences. According to the results, this threshold has a negative impact on our results in this dataset.

Figure 5.6: Results for CrossLang dataset with different similarity thresholds.



Source: The Author

Figure 5.11 shows that as the similarity threshold is increased, the precision increases. We get a precision of 0.72 by using $\delta_{similarity} = 0.95$. We note a significant effect on the recall value and, hence, on the plagdet after losing 933 candidate documents during the retrieval process. Table 5.11 shows a comparison between the results obtained by our approach with ($\delta_{similarity} = 0.65$) the approach presented in the work that created the CrossLang dataset. The authors did not provide the *granularity* and *plagdet* metrics in their work and are therefore depicted as hyphens in table 5.11.

Table 5.11: Comparison of the proposed approach on CrossLang dataset.

Approach	**CrossLang Dataset**				
	recall	precision	granularity	plagdet	F1
Bakhteev et al. (2019b)	**0.79**	**0.83**	**–**	–	**0.80**
Proposed approach (CLPD-CWE)	0.15	0.43	1.0	0.22	0.22

5.3.4 Processing Time Analysis

In this subsection, we present the times spent for pre-processing and the analysis time for cross-language similarity. The elapsed time for processing each dataset is described in table 5.12.

Table 5.12: Pre-processing time and cross-language similarity time analysis

–	**ECLaPA**	**PAN-PC-12 Spanish Partition**	**PAN-PC-12 German Partition**	**CrossLang**
Total pre-processing time	9 minutes and 65 seconds	54 minutes and 48 seconds		6 minutes and 6 seconds
Total analysis time	2 hours and 47 minutes	1 hour and 27 minutes	1 hour and 43 minutes	32 minutes and 24 seconds

The total pre-processing time for the Spanish and German PAN-PC-12 datasets are clustered since both use the same source corpus, and therefore, only one pre-processing is needed for both datasets. The machine specification used to obtain these results is shown in chapter 5.1.3. It is noteworthy that all the processing is done using a GPU provided by Google Colaboratory.

6 CONCLUSION

This work proposed and evaluated CLPD-CWE, a new method for CLPD. The evaluation experiments show that CLPD-CWE is an appropriate approach and that the results outperform state-of-the-art models. The proposed method uses techniques and strategies from different areas, such as the cutting-edge technology of contextualized word embeddings generated by BERT.

The proposed method does not use any translation system. The tests performed have demonstrated that CLPD-CWE works for different language pairs such as English-French, English-Spanish, English-Portuguese, English-German, and English-Russian. The resources needed to obtain these results are a document tokenizer, a language identifier, and a pre-trained multilingual BERT model.

The method was divided into four phases. Each phase can be modified by changing its respective parameters to test different strategies, such as using *Word2Vec* to generate word embeddings. The main difference between the method elaborated in this work and the existing solutions is adopting contextualized word embeddings. To the best of our knowledge, no other proposal had explored these embeddings for CLPD.

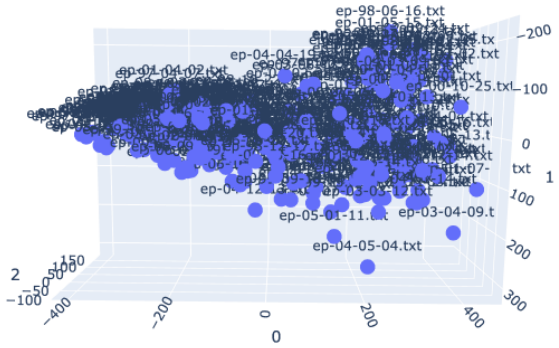
Contextualized word embeddings caught our attention because they have been getting impressive results in NLP downstream tasks and, therefore, an excellent potential for multilingual PD. We evaluated our method on three open-access datasets. All the datasets already had results obtained in other approaches, thus facilitating the comparison with our proposed method.

We evaluated the similarity of multilingual sentences using a different threshold to achieve the best result during the experiments. Using the parameters presented, we obtained excellent results in a dataset that outperforms the state-of-the-art models.

Although we have obtained good results in the experiments, there are still some points that we can improve. Considering that this work is focused on cross-language similarity analysis, we realized that it is possible to use sentence embeddings for document retrieval. A point for future work is the analysis of embeddings of plagiarized passages. One experiment that we have done during the development of CLPD-CWE was to try to find what characteristics the plagiarized cases had in common and how to correlate them—for example, making use of principal component analysis (PCA) or t-SNE (t-Distributed Stochastic Neighbouring Entities) to search which components are more representative in plagiarism cases. PCA is a technique for reducing the number of dimen-

sions while retaining most information. It is done using the correlation between some dimensions and tries to provide a minimum number of variables that keep the maximum variation or information about how the original data is distributed. For illustration, we apply PCA to the sum of all embeddings of each source document in the ECLaPA dataset and plot it in a three-dimensional space shown in Figure 6.1. Hence, reducing the number of dimensions would lead to a faster analysis phase by identifying the most plagiarized sentences, consequently speeding up the plagiarism detection process.

Figure 6.1: PCA plot for ECLaPA source document embeddings



Source: The Author

REFERENCES

ALMEIDA, R. M. V. R. et al. Plagiarism allegations account for most retractions in major latin american/caribbean databases. **Science and Engineering Ethics**, v. 22, n. 5, p. 1447–1456, Oct 2016. ISSN 1471-5546. Available from Internet: <https://doi.org/10.1007/s11948-015-9714-5>.

ALZAHRANI, S. M.; SALIM, N.; ABRAHAM, A. Understanding plagiarism linguistic patterns, textual features, and detection methods. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, IEEE, v. 42, n. 2, p. 133–149, 2011.

AZAD, H. K.; DEEPAK, A. Query expansion techniques for information retrieval: a survey. **Information Processing & Management**, Elsevier, v. 56, n. 5, p. 1698–1735, 2019.

BAKHTEEV, O. et al. Crosslang: the system of cross-lingual plagiarism detection. In: **Workshop on Document Intelligence at NeurIPS 2019**. [S.l.: s.n.], 2019.

BAKHTEEV, O. et al. Crosslang: the system of cross-lingual plagiarism detection. In: **Workshop on Document Intelligence at NeurIPS 2019**. [S.l.: s.n.], 2019.

BARRÓN-CEDEÑO, A. et al. A comparison of approaches for measuring cross-lingual similarity of wikipedia articles. In: RIJKE, M. de et al. (Ed.). **Advances in Information Retrieval**. Cham: Springer International Publishing, 2014. p. 424–429. ISBN 978-3-319-06028-6.

BARRóN-CEDEñO, A.; GUPTA, P.; ROSSO, P. Methods for cross-language plagiarism detection. **Knowledge-Based Systems**, v. 50, p. 211 – 217, 2013. ISSN 0950-7051. Available from Internet: <http://www.sciencedirect.com/science/article/pii/S0950705113002001>.

BENGIO, Y. et al. A neural probabilistic language model. **The journal of machine learning research**, JMLR. org, v. 3, p. 1137–1155, 2003.

BENGIO, Y.; SIMARD, P.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. **IEEE transactions on neural networks**, IEEE, v. 5, n. 2, p. 157–166, 1994.

BHATTACHARYA, P.; GOYAL, P.; SARKAR, S. Using word embeddings for query translation for hindi to english cross language information retrieval. **Computación y Sistemas**, Centro de Investigación en computación, IPN, v. 20, n. 3, p. 435–447, 2016.

BISHOP, C. M. **Pattern Recognition and Machine Learning (Information Science and Statistics)**. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN 0387310738.

CARBONELL, J.; GOLDSTEIN, J. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: **Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval**. [S.l.: s.n.], 1998. p. 335–336.

CEDEñO, A. On the mono- and cross-language detection of text re-use and plagiarism. **Procesamiento de Lenguaje Natural**, v. 50, p. 103–105, 03 2013.

CER, D. et al. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. **Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)**, Association for Computational Linguistics, 2017. Available from Internet: <http://dx.doi.org/10.18653/v1/S17-2001>.

CHO, K. et al. Learning phrase representations using rnn encoder-decoder for statistical machine translation. **arXiv preprint arXiv:1406.1078**, 2014.

DANILOVA, V. Cross-language plagiarism detection methods. In: **Proceedings of the Student Research Workshop associated with RANLP 2013**. Hissar, Bulgaria: INCOMA Ltd. Shoumen, BULGARIA, 2013. p. 51–57. Available from Internet: <https://www.aclweb.org/anthology/R13-2008>.

DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018.

DU, K.-L.; SWAMY, M. Perceptrons. In: **Neural Networks and Statistical Learning**. [S.l.]: Springer, 2019. p. 81–95.

EHSAN, N.; SHAKERY, A.; TOMPA, F. W. Cross-lingual text alignment for fine-grained plagiarism detection. **Journal of Information Science**, SAGE Publications Sage UK: London, England, v. 45, n. 4, p. 443–459, 2019.

EHSAN, N.; TOMPA, F. W.; SHAKERY, A. Using a dictionary and n-gram alignment to improve fine-grained cross-language plagiarism detection. In: **Proceedings of the 2016 ACM Symposium on Document Engineering**. [S.l.: s.n.], 2016. p. 59–68.

EISELT, M. P. B. S. A.; ROSSO, A. B.-C. P. Overview of the 1st international competition on plagiarism detection. In: **3rd PAN Workshop. Uncovering Plagiarism, Authorship and Social Software Misuse**. [S.l.: s.n.], 2009. p. 1.

FERRERO, J. et al. Using word embedding for cross-language plagiarism detection. In: **Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers**. Valencia, Spain: Association for Computational Linguistics, 2017. p. 415–421. Available from Internet: <https://www.aclweb.org/anthology/E17-2066>.

FRANCO-SALVADOR, M. et al. Cross-language plagiarism detection over continuous-space-and knowledge graph-based representations of language. **Knowledge-based systems**, Elsevier, v. 111, p. 87–99, 2016.

GROOTENDORST, M. **KeyBERT: Minimal keyword extraction with BERT.** Zenodo, 2020. Available from Internet: <https://doi.org/10.5281/zenodo.4461265>.

GUPTA, D. et al. Study on extrinsic text plagiarism detection techniques and tools. **Journal of Engineering Science & Technology Review**, v. 9, n. 5, 2016.

GUTHRIE, D. et al. A closer look at skip-gram modelling. In: CITESEER. **LREC**. [S.l.], 2006. v. 6, p. 1222–1225.

HARLEY, A. W. An interactive node-link visualization of convolutional neural networks. In: **ISVC**. [S.l.: s.n.], 2015. p. 867–877.

HAYKIN, S. S. **Neural networks and learning machines**. Third. Upper Saddle River, NJ: Pearson Education, 2009.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

KISS, T.; STRUNK, J. Unsupervised multilingual sentence boundary detection. **Computational linguistics**, MIT Press, v. 32, n. 4, p. 485–525, 2006.

LEILEI, K. et al. Approaches for candidate document retrieval and detailed comparison of plagiarism detection. **Forner et al.[33]**, 2012.

LI, Y.; YANG, T. Word embedding for understanding natural language: a survey. In: **Guide to Big Data Applications**. [S.l.]: Springer, 2018. p. 83–104.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. **Introduction to Information Retrieval**. [S.l.]: Cambridge University Press, 2008.

MCENERY, T. **Corpus linguistics**. [S.l.]: Oxford University Press Inc, 2012.

MUHR, M. et al. External and intrinsic plagiarism detection using a cross-lingual retrieval and segmentation system. In: **Notebook Papers of CLEF 2010 LABs and Workshops**. [S.l.: s.n.], 2010. p. 22.

OARD, D. W.; HE, D.; WANG, J. User-assisted query translation for interactive cross-language information retrieval. **Information Processing & Management**, Elsevier, v. 44, n. 1, p. 181–211, 2008.

PATAKI, M. A new approach for searching translated plagiarism. In: **5th International Plagiarism Conference**. Newcastle: [s.n.], 2012. v. 2012, p. 49. Available from Internet: <http://eprints.sztaki.hu/6539/>.

PEREIRA, R. C.; MOREIRA, V. P.; GALANTE, R. A new approach for cross-language plagiarism analysis. In: AGOSTI, M. et al. (Ed.). **Multilingual and Multimodal Information Access Evaluation**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 15–26. ISBN 978-3-642-15998-5.

PETERS, M. E. et al. Semi-supervised sequence tagging with bidirectional language models. **arXiv preprint arXiv:1705.00108**, 2017.

PETERS, M. E. et al. **Deep contextualized word representations**. 2018.

PIRES, T.; SCHLINGER, E.; GARRETTE, D. How multilingual is multilingual bert? **arXiv preprint arXiv:1906.01502**, 2019.

POTTHAST, M. **Technologies for reusing text from the web**. Thesis (PhD) — Citeseer, 2012.

POTTHAST, M. et al. Overview of the 2nd international competition on plagiarism detection. In: . [S.l.: s.n.], 2010. v. 1176.

POTTHAST, M. et al. Cross-language plagiarism detection. **Knowledge-Based Systems**, v. 45, p. 45–62, 03 2011.

POTTHAST, M. et al. Overview of the 3rd international competition on plagiarism detection. In: CEUR WORKSHOP PROCEEDINGS. **CEUR workshop proceedings**. [S.l.], 2011. v. 1177.

POTTHAST, M. et al. Overview of the 5th international competition on plagiarism detection. In: CELCT. **CLEF Conference on Multilingual and Multimodal Information Access Evaluation**. [S.l.], 2013. p. 301–331.

QIAO, Y. et al. Understanding the behaviors of bert in ranking. **arXiv preprint arXiv:1904.07531**, 2019.

REIMERS, N.; GUREVYCH, I. Sentence-bert: Sentence embeddings using siamese bert-networks. **arXiv preprint arXiv:1908.10084**, 2019.

ROOSTAEE, M.; FAKHRAHMAD, S. M.; SADREDDINI, M. H. Cross-language text alignment: A proposed two-level matching scheme for plagiarism detection. **Expert Systems with Applications**, Elsevier, v. 160, p. 113718, 2020.

ROOSTAEE, M.; SADREDDINI, M. H.; FAKHRAHMAD, S. M. An effective approach to candidate retrieval for cross-language plagiarism detection: A fusion of conceptual and keyword-based schemes. **Information Processing & Management**, Elsevier, v. 57, n. 2, p. 102150, 2020.

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, v. 65 6, p. 386–408, 1958.

SANCHEZ-PEREZ, M. A.; GELBUKH, A.; SIDOROV, G. Adaptive algorithm for plagiarism detection: The best-performing approach at pan 2014 text alignment competition. In: SPRINGER. **International Conference of the Cross-Language Evaluation Forum for European Languages**. [S.l.], 2015. p. 402–413.

SÁNCHEZ-VEGA, F. et al. Paraphrase plagiarism identification with character-level features. **Pattern Analysis and Applications**, v. 22, n. 2, p. 669–681, May 2019. ISSN 1433-755X. Available from Internet: <https://doi.org/10.1007/s10044-017-0674-z>.

SANH, V. et al. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. **arXiv preprint arXiv:1910.01108**, 2019.

SHEN, D. et al. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. **arXiv preprint arXiv:1805.09843**, 2018.

Soleman, S.; Fujii, A. Toward plagiarism detection using citation networks. In: **2017 Twelfth International Conference on Digital Information Management (ICDIM)**. [S.l.: s.n.], 2017. p. 202–208.

STEIN, B.; EISSEN, S. M. zu; POTTHAST, M. Strategies for retrieving plagiarized documents. In: **Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval**. [S.l.: s.n.], 2007. p. 825–826.

SUTSKEVER, I.; VINYALS, O.; LE, Q. V. Sequence to sequence learning with neural networks. **arXiv preprint arXiv:1409.3215**, 2014.

SUTSKEVER, I.; VINYALS, O.; LE, Q. V. **Sequence to Sequence Learning with Neural Networks**. 2014.

SZEGEDY, C.; TOSHEV, A.; ERHAN, D. Deep neural networks for object detection. In: **Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2**. Red Hook, NY, USA: Curran Associates Inc., 2013. (NIPS'13), p. 2553–2561.

UNANUE, I. J.; BORZESHI, E. Z.; PICCARDI, M. Recurrent neural networks with specialized word embeddings for health-domain named-entity recognition. **Journal of biomedical informatics**, Elsevier, v. 76, p. 102–109, 2017.

VASWANI, A. et al. **Attention Is All You Need**. 2017.

VULIĆ, I.; MOENS, M.-F. Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In: **Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval**. [S.l.: s.n.], 2015. p. 363–372.

WENG, L. Attention? attention! **lilianweng.github.io/lil-log**, 2018. Available from Internet: <http://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html>.

WU, D.; HE, D. A study of query translation using google machine translation system. In: IEEE. **2010 International Conference on Computational Intelligence and Software Engineering**. [S.l.], 2010. p. 1–4.

ZHANG, L.; ZHAO, X. An overview of cross-language information retrieval. In: SPRINGER. **International Conference on Artificial Intelligence and Security**. [S.l.], 2020. p. 26–37.