Analytical Study of Traditional and Intelligent Textual Plagiarism Detection Approaches

Ayoob Ali[1*], Alaa Yassen Taqa[2]

[1*]Department of Mathematics, College of Basic Education, University of Mosul, Mosul, Iraq
[2]Department of Computer sciences, College of Pure science Education, University of Mosul, Mosul, Iraq

E-mail: [1*]ayobali-1980@uomosul.edu.iq, [2]alaa.taqa@gmail.com

(Received October 16, 2021; Accepted December 27, 2021; Available online March 01, 2022)

DOI: 10.33899/edusj.2021.131895.1192, © 2022, College of Education for Pure Science, University of Mosul.

Abstract:

The Web provides various kinds of data and applications that are readily available to explore and are considered a powerful tool for humans. Copyright violation in web documents occurs when there is an unauthorized copy of the information or text from the original document on the web; this violation is known as Plagiarism. Plagiarism Detection (PD)can be defined as the procedure that finds similarities between a document and other documents based on lexical, semantic, and syntactic textual features. The approaches for numeric representation (vectorization) of text like Vector Space Model (VSM) and word embedding along with text similarity measures such as cosine and jaccard are very necessary for plagiarism detection. This paper deals with the concepts of plagiarism, kinds of plagiarism, textual features, text similarity measures, and plagiarism detection methods, which are based on intelligent or traditional techniques. Furthermore, different types of traditional and algorithms of deep learning for instance, Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) are discussed as a plagiarism detector. Besides that, this work reviews many other papers that give attention to the topic of Plagiarism and its detection.

Keywords: Plagiarism, Plagiarism Detection, String matching, Deep learning , Text Similarity

دراسة تحليلية للأساليب التقليدية والذكية المستخدمة في كشف السرقة الاكاديمية

أيوب علي محمد سعيد[1*] ، الاء ياسين طاقة [2]

[1*] قسم الرياضيات, كلية التربية الأساسية, جامعة الموصل, الموصل، العراق

[2] قسم علوم الحاسوب, كلية التربية للعلوم الصرفة, جامعة الموصل, الموصل، العراق

الخلاصة :

توفر شبكة الويب أنواعًا مختلفة من البيانات والتطبيقات المتاحة بسهولة للاستكشاف و الاستخدام من قبل للمستخدمين. يحدث انتهاك حقوق النشر في مستندات الويب عند وجود نسخة غير مصرح بها من المعلومات أو النص من المستند الأصلي على الويب ؛ يُعرف هذا الانتهاك بالانتحال. يمكن تعريف اكتشاف الانتحال (PD) على أنه الإجراء الذي يجد أوجه التشابه بين مستند معين والمستندات الأخرى بناءً على الميزات النصية المعجمية والدلالية والنحوية. تعتبر مناهج التمثيل الرقمي (تحويل النص) مثل

Vector Space Model (VSM) ودمج الكلمات Word Embedding جنبًا إلى جنب مع مقاييس تشابه النص مثل مقياس جيب التمام ضرورية للغاية لاكتشاف الانتحال النصي. تتناول هذه الورقة البحثية مفاهيم الانتحال ، أنواع الانتحال ، خصائص النصوص أو المستندات النصية ، مقاييس تشابه النصوص ، وطرائق الكشف عن الانتحال التي تستخدم تقنيات ذكية أو تقليدية. إذ تم مناقشة أنواع مختلفة من التقنيات التقليدية وخوارزميات التعلم العميق ، الشبكة العصبية التلافيفية (CNN) والذاكرة طويلة المدى (LSTM) في كشف النصوص الأدبية. إلى جانب ذلك ، يستعرض هذا العمل العديد من الأوراق الأخرى التي تهتم بموضوع الانتحال النصي وكشفه .

الكلمات المفتاحية: السرقة الأكاديمية ، كشف السرقة الأكاديمية ، مطابقة السلاسل النصية ، التعلم العميق ، تشابه النصوص

1. **Introduction**:

Plagiarism is defined as using all or some portions of another person’s ideas or works but without providing a reference or mention for them. Nowadays in this digital era, the abundance of resources available on the Internet lead to an increase in the problem of plagiarism [1].

Around since the 1990s , statistical or computerized methods of plagiarism detection (PD) have been used in natural language documents [2]. Over the past ten years , Recent advances in related fields such as information retrieval (IR), cross language information retrieval (CLIR), natural language processing, computational linguistics, artificial intelligence, and soft computing have aided research on automated plagiarism detection in natural languages [3].

The remainder of this paper intends to illustrate plagiarism and its various types, investigate the approaches towards plagiarism detection, demonstrate various textual features for the quantification of documents as a *proviso* in PD. Furthermore syntax-based (SYN) and semantic-based (SEM) plagiarism are discussed, and finally , a conclusion is presented.

The paper summarizes and explains several concepts related to the academic plagiarism and plagiarism detection approaches , text features and text similarity measures. Besides that, the study presents a comparative analysis among traditional and modern intelligent techniques for textual plagiarism detection based on a set of criteria including vector representation, similarity approach, and dataset. Each technique has advantages and disadvantages, but none of them is fully developed for semantic plagiarism detection.

2. Kinds of plagiarism

Using text, visual, audio data , or any part of these data in a work you presented, but without taking a permission or putting an explicit mention, is known as plagiarism [4].

With respect to text, which is the subject of this paper, plagiarism can appear in program code or a research article and it could be in different behaviors [5]:

a. Claiming that you take credit for someone else's work.
b. Utilizing someone else's effort and without providing credit.
c. Whether or not credit is given, consider the majority of someone else's contribution to be your own.
d. Restructuring someone else's work and appealing it as your own.
e. Incorrectly acknowledging others' work in your work.

Broadly, it is possible to categorize Plagiarism into two chief kinds namely (i) Literal Plagiarism and (ii) Intelligent Plagiarism as shown in Figure 1.

Plagiarism in the literal sense is often done through exact copy of text in whole or in part. Near copy is done from various sources with a little alteration like insertion, deletion, substitution, spliting or joining sentences, Modified copy is done through changing the syntax or reordering phrases in the original text [6].

Intelligent Plagiarism is difficult to detect. It is not just copying the text but modifying it in such a way that the meaning of the text remains same, and it appears as a new idea. It can be further classified into three types [7,8]:

- Text Manipulation: This includes modifying the text, using lexical or syntactic changes in paraphrasing or summarizing through sentence reduction, combination, or reconstruction.
- Translation: Translating a sentence into another language and converting it back to the original language will change the way the original sentence was written thus by-passing the detection in an intelligent manner regardless of it having been done manually or in automatically.

Idea Adoption: It is similar to the way of presenting an idea in a different manner so that it resembles a new one. This is mostly practiced in the business world or research where a competitor's idea is stolen to gain success.

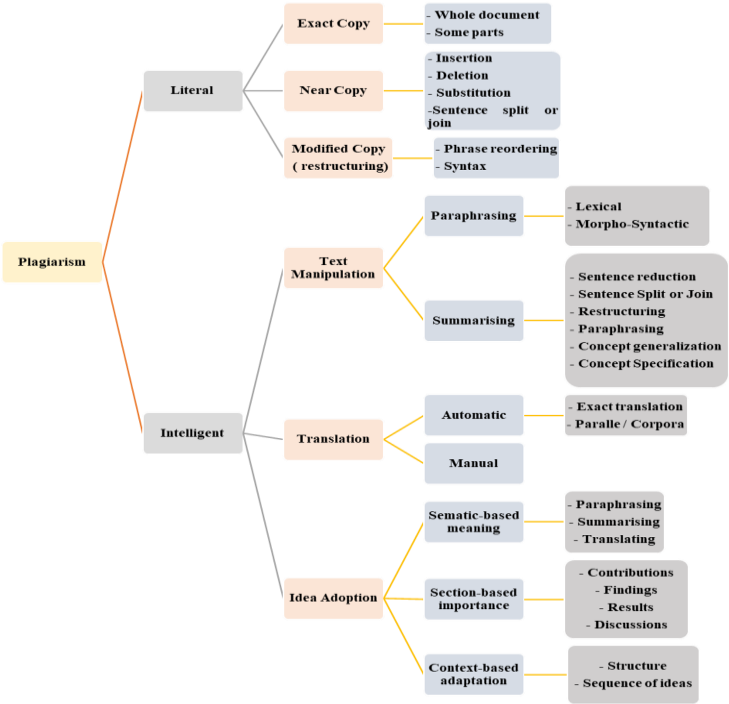

Figure 1. Plagiarism Classifications [8]

3. Cocepts of plagiarism detection

Plagiarism detection and paraphrase identification are hot topics for publishers, researchers, and educational institutions. Paraphrase identification is used in several other tasks beside plagiarism detection, like machine translation, information retrieval, question answering among others [9].

Figure 2 illustrates a system of plagiarism detection (PD) which is represented as a black box design that has three main components, one input being a set of query documents (QD) that are to be tested, and another input is a collection of documents (D) which help to detect plagiarism such as on the Web or an existing corpus. The output of this system is a set of suspicious sections that contains plagiarized text [10]. Plagiarism can occur in natural languages that are same or distinct.

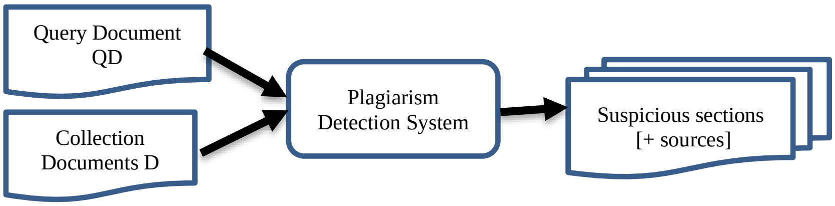

Figure 2. Black Box design for plagiarism detection system [10]

Detection approaches of plagiarism can be split into two fundamental types cross-lingual and monolingual. Depending on the textual sources being compared in terms of language homogeneity or heterogeneity [11].

Monolingual PD deals with homogeneous language settings for example English-English. It has two kinds[8].

- Intrinsic PD: This method examines the author's writing style or individuality and attempts to discover plagiarism using own-conformity or deviation between text parts. There are no external sources required for detection.
- Extrinsic PD: This method compares a submitted research article to a large number of additional comparable digital resources available in repositories or on the Internet. The extrinsic analysis has two subtasks:
  - Source Retrieval: Given a web search engine and a suspect document, the task is to identify all plagiarized sources while minimizing retrieval costs..
  - Text Alignment: The aim is to identify all contiguous maximum possible reused text passages between a given pair of documents.

Cross-Lingual PD technique can work in a variety of heterogeneous language environments, like English-Chinese.. In this approach, Finding closeness between two text fragments in different languages is very difficult.

3.1. Classes of text similarity

Text similarity is the comparison of two texts to find how ‘close’ the two texts are in meaning which is called semantic similarity or surface closeness, which is known as lexical similarity[12].

There are several measures to determine the text similarity as illustrated in Fig 3 [13].

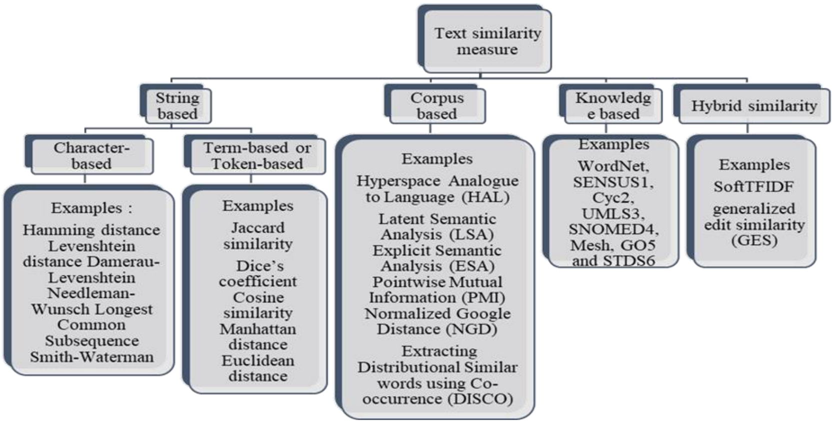
Text similarity measure
String based
Corpus based
Knowledge based
Hybrid similarity
Character-based
Term-based or Token-based
Examples : Hamming distance Levenshtein distance Damerau-Levenshtein Needleman-Wunsch Longest Common Subsequence Smith-Waterman
Examples Jaccard similarity Dice's coefficient Cosine similarity Manhattan distance Euclidean distance
Examples Hyperspace Analogue to Language (HAL) Latent Semantic Analysis (LSA) Explicit Semantic Analysis (ESA) Pointwise Mutual Information (PMI) Normalized Google Distance (NGD) Extracting Distributional Similar words using Co-occurrence (DISCO)
Examples WordNet, SENSUS1, Cyc2, UMLS3, SNOMED4, Mesh, GO5 and STDS6
Examples SoftTFIDF generalized edit similarity (GES)

Figure 3. Diagram of classes for text similarity measures [13].

- String based operates on string sequences and character composition. This class is further divided into two sub classes: Character-based detection is helpful for spotting typographical errors, and Term-based or Token-based which is useful for recognizing rearrangement of terms by breaking strings into substrings.
- Corpus based, depending on information taken from a vast corpus, it estimates the similarity between two concepts.
- Knowledge-based is a measure based on semantic similarity. The rules of conclusion, logical propositions, and network semantics such as taxonomy and ontology are all part of the knowledge representation schema.

- The thought of Hybrid class aims to uses the advantages of the previously established classes, such as similarity that is , corpus-based, string-based and knowledge-based, to create a better approach.

3.2. Similarity measures

It is important to determine the distance between the two texts which is needed in all classes of text similarity measures [14].

The query document DQ and source documents D are separated into smaller parts termed as fingerprints (or shingles) with length of K. Fingerprints like character or word k-grams are processed to generate hash values (or hash sums) by using a hash function. Afterthat , the hash values can be arranged and matched with hash values of other documents. Each document's vector comprises a set of its unique fingerprints (or hashes).

A text vector similarity is useful for identifying the similarity ratio between pair of texts [15]. Some distance measures are utilized for specifying the ratio of similarity among documents through measuring the distances between two vectors which represents the documents. [16].

String matching is a technique used to search a given pattern (substring) in a string. String editing techniques also known as approximate matching, compare two texts and perform required operations (add , edit , delete) to convert the first text into the second text [17]. Table 1 briefly summarizes some used text similarity metrics for vector based similarity, string matching and editing.

The first seven metrics are used with numeric and text representation as a vector and can be utilized for PD based on lexical and semantic features while the rest can capture only lexical features of the text that are applied in traditional approaches for PD. It is possible to combine two measures and specify thresholds to detect the similarity between texts for PD.

Table1. Text similarity metrics

Distance Measure	Description	Type	Ref.
Jaccard	The number of intersection between two vector divided by the union of the two vector	Vector	[18]
Dice	Equivalent to Jaccard, but with the added benefit of reducing the impact of common terms between vectors.	Vector	[13]
Cosine	Discovers the angle between two vectors that is cosine.	Vector	[18]
Euclidean	Calculates the euclidean distance between two vectors.	Vector	[13]
Manhattan	Measures the average differences across dimensions .	Vector	[13]
Rabin-karp	It searches the substring (m) in the string (n) utilizing a hash function	String matching	[17]
Jaro-Winkler Distance	For two strings, calculates the Jaro-Winkler Distance. The normal value is 0, which means there is no similarity, and 1, which means there are identical similarities.	String matching	[17]
Hamming	Define number of characters different between two strings	Character level editing	[8]
Levenshtein	Determine the minimal edit distance required to convert x to y.	Character level editing	[8]
Longest Common Sequence (LCS)	In terms of char order, this value represents the length of the longest char pairing that can be formed between x and y.	Word level editing	[8]

3.3. Text representation

Natural language processing (NLP) is an AI subfield that allows computers to read, understand, and interpret language of human.

In NLP, one of the active research areas is PD. Its goal is to detect text reuse, modification, and/or reproduction from one form to another. After acquiring the data (text) and before using NLP techniques, itis important to preprocess the text. This improves the accuracy of the approach of PD [19].

3.3.1 Text preprocessing

Text preprocessing is the action of cleaning and transforming the machine learning algorithms can perform better by converting content into a more consumable format. It is an essential step for NLP tasks [4].

There are several steps in the text pre-processing such as:

1- Text normalization: The characters in the input text that represent letters from other languages are turned to characters from the currently utilized language. All texts must be encoded in the same way.

2- Stop Word and Special Character Removal: The most frequently occurring words which slow down the processing of documents are called stop words. These words are irrelevant. Such words including articles, conjunctions, prepositions, and punctuations are removed.

3- Stemming and Lemmatization: Stemming is a heuristic procedure for removing word ends, which usually includes the elimination of affixes. Lemmatization is the process of morphologically analyzing words and returning the lemma, which is the dictionary or base form of the word.

4- Tokenization is a process of converting sentences into a chain of words so that processing word by word can be easily performed.

5- Text Representation: in this step the feature vector of text is represented numerically.

3.3.2 Textual features

Textual characteristics are utilized in PD methods , some of textual features are [9]:

1) Lexical Features**:** In any document, character and word are the simplest forms to represent. Lexical features are operated at level of word or character. For example N-gram representation of words is a collection of words in any document.

2) Syntactic Features**:** These are characterized by sentence-based representation. Text is fragmented into sentences utilizing full stops , question marks or other delimiters.

3) Semantic Features**:** Semantic features help in PD by providing insights into the meaning of text so that it can be compared semantically using semantic dependencies and POS tagging.

4) Structural Features**:** Structural features for PD deals with tree organizations of documents. Document is the collection of paragraphs and similar paragraphs that constitute a block having similar semantics. These blocks later form sections and sub-sections. Structural features can be divided into two types which are block-specific and content-specific structured features [20].

3.4 Representing textual features

There are several approaches to represent a text as numeric values, as illustrated in Figure 4 [4].

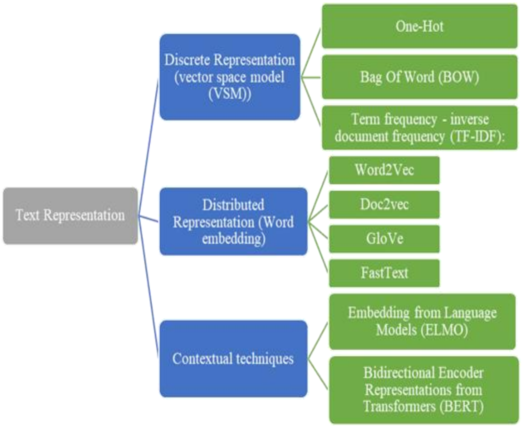
Text Representation
Discrete Representation (vector space model (VSM))
One-Hot
Bag Of Word (BOW)
Term frequency - inverse document frequency (TF-IDF):
Distributed Representation (Word embedding)
Word2Vec
Doc2vec
GloVe
FastText
Contextual techniques
Embedding from Language Models (ELMO)
Bidirectional Encoder Representations from Transformers (BERT)

Figure 4. Text representation approaches [4]

3.4.1. Discrete Representation or Vector Space Model (VSM)

The VSM is an algebraic paradigm for encoding text documents as a vector for identification [8]. This approach has several examples such as:

- One-Hot: Represents words as a vector with dimensions equal to the dictionary's length. One-Hot is a set of bits in which the only permitted value combinations are those with a single (1) bit and all others are (0).
- Bag of Word (BOW): The words are placed in a "bag," and the number of times each word appears is counted.
- Term Frequency - Inverse Document Frequency (TF-IDF): TF: is the frequency of a word that appears in the current text which is simply the sum of the one-hot representation of its constituent words. IDF: Inverse document frequency is a numerical statistic that indicates how important a term is in a corpus or collection of texts.

The TF-IDF is calculated according to the following formulas (1) and (2):

$$Idf(w) = \log\left(\frac{N}{n_w}\right) \qquad (1)$$

$$tf - idf = tf * idf(w) \qquad (2)$$

Where w is the word, n_w is the count of documents comprising the word and N is the total amount of documents.

3.4.2 Distributed Representation

Converts a word to a n-dimensional vector. Words which are associated with other words are transformed to similar n-dimensional vectors, while non associated words will have different vectors. In this manner the embedding of a word will reflect the semantic features or ‘meaning’ of that word [12].

- Word2Vec: An unsupervised shallow, two-layer neural network to create a distributed representation of words. Two word2vec models as shown in Fig 5.

o Continuous Bag of Words (CBOW): Predict the primary word using the words of context's.

o SKIP-GRAM: A neural network that predicts context by using the center word.

- Doc2vec: Doc2vec is an unsupervised technique for converting phrases, paragraphs, and documents into vectors. It is based on the Word2Vec algorithm.
- GloVe: Global Vectors for representations of words, it’s an algorithm for calculating the global word frequency statistic (count-based and overall statistics word representation).

- FastText: Builds on Word2Vec, character based by learning vector representations for each word and the n-grams found within each word.

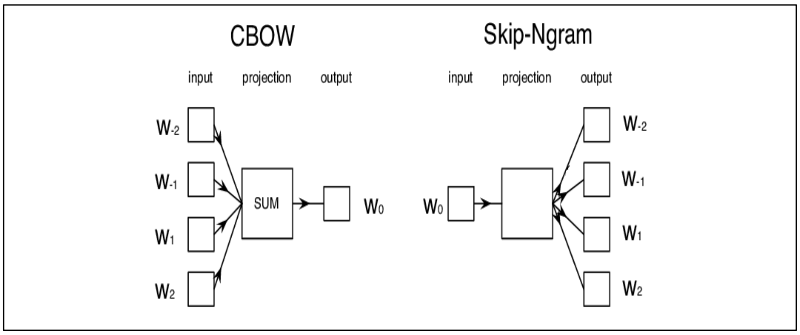

Figure 5. word2vec CBOW and Skip-gram [17].

3.4.3 Contextual techniques

Contextual techniques utilizes transformers and long and short-term memory techniques to convert a word into an n-dimensional vector [21]. The two common Contextual techniques are:

- Embedding from Language Models (ELMO): is a deep character-dependent bidirectional language model (biLM) composed of two-layer networks of long short-term memory (LSTM) layered on top of a convolutional layer with max-pooling.
- Bidirectional Encoder Representations from Transformers (BERT): utilizes of Transformer and an attention mechanism that learns contextual relationships between words in a phrases.

4. Study of plagiarism detection approaches

Authors have used several approaches for PD, some of them used only direct text matching without vectorization of text which is considered as a traditional approach. Diversely, intelligent approaches also utilized by others for text vectorization and PD.

Both the traditional and intelligent approaches utilizes an existing corpus for PD such as Plagiarism, Authorship, and Social Software Misuse (PAN), Conference and Labs of the Evaluation Forum (CELF) , Corpus of English Novels (CEN) and Open Source Arabic Corpus (OSAC). In the case where the PD is based on retrieved source text from the web or consider the web as corpus, the researchers used the source retrieval approach.

4.1. Traditional approaches

The traditional approaches for PD are based on direct text matching methods, word or term frequency-based methods, considering only lexical features of text. Table 2 illustrates several traditional approaches which have been used in PD.

Table 2. PD based on traditional approaches.

Author	Encoding	Similarity	Corpus	Remarks
[22]	Direct string Matching	Dice coefficients	Bahas	Generates a percentage of the similarity of the documents by calculating n-grams hash results with Dice coefficients.
[23]	BOW	Jaccard	PAN-2012	The number of participant authors and the length of the evaluated documents are two criteria that influence the accuracy levels.
[24]	VSM	Jaccard	PAN corpus of CLEF	Based on a combination of VSM and the improved Jaccard coefficient, a novel plagiarism detection model has been

				developed.
[25]	n-gram of BOW	Cosine similarity	Indian Dataset	A VSM for PD that uses trigram as a possible technique. Furthermore, the cosine similarity metric yields somewhat better results than the Jaccard similarity measure, making the cosine similarity measure the better option.
[26]	TFIDF VSM	cosine	E-homework	The e-homeworks with lower ratio in similarity than threshold value are marked as non plagiarism and accepted, while others are rejected and considered as plagiarism.
[27]	VSM TFIDF	Cosine + Jaccard	Persian Dataset	The application achieved both suitable accuracy and rapid speed. The temporal order is determined by the count of features in the VSM and the size of the collection of documents.
[28]	VSM	Cosine	Indian Dataset	In different documents, TF-IDF does not capture text semantic co-occurrences.

4.2. Intelligent approaches

Artificial intelligence (AI) has been used in a number of ways to deal with the difficulty of plagiarism detection. Similar to many other booming fields, AI plays a significant role in this regard, since stolen text often is altered in a great degree so as to evade even the strongest copy content scanning software. Therefore, several AI techniques such as Deep Learning and Machine learning were employed by researchers for PD [29] .

4.2.1 Machine learning

The intelligent approaches such as ML and word embedding for text representation or training can be used for PD. These approaches are used for PD taking into account semantic and lexical features of text. The following Table 3 illustrates some papers that performed PD using machine learning based approach [30].

Table 3. PD based on ML.

Author	Encoding	Intelligent technique	Similarity	Corpus	Text feature	Remarks
[6]	BOW, LSA	SVM Stylometr-y	Cosine	Corpus of English Novels (CEN)	Semant-ic Lexical	By combining LSA, which links words semantically, with Stylometry, which captures each author's writing style patterns, the technique offers a new mechanism.
[30]	VSM	NB , SVM	Manhatta-n	Lexical Semanti-c	160 trainin-g data	Using SVM algorithm showed better results It also outperforms Naive Bayes when it comes to tackling high-dimensional issues (which have a lot of features)
[31]	TF/IDF and Word2ve-c	Word2vec	Cosine + Euclidean	Lexical and Semanti-c	Open Source Arabic Corpus OSAC	Approach based on TF-IDF and word2vec to reduce computing complexity and increase the likelihood of correctly identifying words in context.

[32]	**TFIDF and Doc2vec**	**Doc2vec**	**Cosine**	**Lexical and Semanti-c**	**French novel from Gutenberg Project ,**	**Doc2vec document representation outperforms TF-IDF document representation,**
[33]	**Word2ve-c**	**Word2vec**	**longest common subsequence LCS**	**Lexical Semanti-c**	**PAN 2013**	**A The weight defined by a distributed representation is used to identify plagiarism using a document similarity algorithm.**

4.2.2 Deep learning

Deep learning (DL) is a novel approach in Artificial Intelligence and a branch of machine learning that uses deep multiple layer graphs to try to find more abstract features [34].

Some researchers have used DL for text feature extraction while others used it for text classification and detection of plagiarism.

Deep neural network topologies come in a variety of shapes and sizes such as [5]:

- Recursive neural network (RNN): is a kind of the best commonly used architectures in problems of NLP because their recurrent structure, it is well suited to processing texts that has variable-length, and can process a sequence of arbitrary length by recursively applying a transition function to the input sequence's internal hidden state vector.
- Siamese LSTM: is a particular type of RNN that can learn long-term dependencies. LSTMs are specifically developed to prevent the problem of long-term dependency. It is basically their default behavior to remember information for long periods of time. RNNs are capable of modeling and remembering the links between words and phrases.

Figure 6 illustrates an example of a LSTM network. The text pair is supplied into the input layer, where the Embedding layer embeds the text into low-dimensional vectors, the hidden layer learns high-level features, the attention layer generates a weight vector, and the output layer generates predicted similarity (or label) [35].

- Convolutional neural network (CNN): is a feed-forward, deep artificial neural network that uses multilayer perceptual variation with little preprocessing. The visual cortex of animals served as inspiration for them. CNNs are extensively used in computer vision. However, they have just lately been used to solve a range of NLP challenges, like text classification [36].

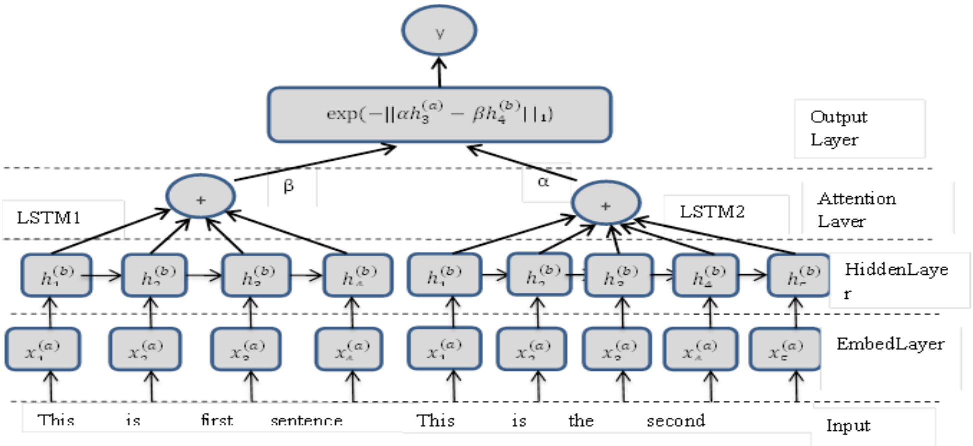

Figure 6. Siamese LSTM Network Structure and Mechanism of Attention [35].

Many authors have presented several approaches to deal with plagiarism detection using Deep learning algorithms. Table 4 displays some PD based on word embedding approaches that utilized the DL approach. While table 5 summarizes some of the research used LSTM type of DL approaches.

Table 4. PD based on word embedding

Author	Embedding	DL method	Similarity distance	Corpus
[37]	Word2vec	Word2vec	Softcosine	UCSC_Sinhala_News
[38]	Word2vec	Word2vec	Cosine	OSAC
[34]	Word2vec	Word2vec	Cosine+Jaccard	PAN2016 Persian

Table 5. PD based on LSTM

Author	Embedding	DL method	Similarity distance	Corpus
[39]	Word2vec	Bi-LSTM	Cosine	CCKS2018 QIM
[40]	Word2Vec	LSTM	Manhattan	SICK dataset
[36]	Glove	LSTM	Cosine	(SICK), (MSRVID), (STS2014), WikiQA, TrecQA
[41]	Word2vec	LSTM’s	Cos	Input sentences
[19]	Doc2Vec	LSA and Bidirectiona l LSTM	Cosine	MSRPC
[42]	word2vec	Siamese LSTM	Manhattan cosine	IMDB 20Newsgroups
[43]	Word2ec	LSTM	Cosine	MSR, Quora
[44]	GloVe	Bi-LSTM	cosine and jaccard	MSRP and Quora
[45]	Word2Vec Skip-Gram	Siamese LSTM	Manhattan	Stanford Web
[46]	LSTM	Siamese LSTM	Manhattan	SICK
[47]	word2vec	LSTM	Cos	SemEval 2016

Table 6 lists some papers that have utilized the CNN type of DL approaches for Plagiarism detection through different corpora. Some authors are mixed both CNN and LSTM for plagiarism detection purpose and Table 7 illustrates some papers that have used that mix.

Table 6. PD based on CNN

Author	Embedding	DL method	Similarity distance	Corpus
[29]	Glove	CNN	cosine and Euclid	MSRP,SICK, MSRVID
[48]	Glove	CNN	Cosine	Input sentences
[49]	word2vec	CNN	Cos	KSUCCA, AraCorpusa Wikipediab, Total number , Test model, OSACc

Table 7. PD based on CNN and LSTM

Author	Embedding	DL method	Similarity distance	Corpus
[12]	doc2vec	CNN and LSTM	Cosine	PAN
[50]	Word2Vec	CNN and LSTM	Cosine	SICK dataset
[51]	Word2vec	CNN and LSTM	cosine	SemEval, Microsoft Paraphrase

4.2.3. Evaluation

It is important to ensure that the project or design model has reasonable results and achieves its goal, therefore evaluation metrics used to evaluate the project.

The accuracy is one of standard evaluation metrics of classification results. The percent of the total number of correctly detected documents across all sets is called accuracy. Correctly clean classified texts (True Negatives: TN), correctly classified plagiarized texts (True Positives: TP), clean texts incorrectly classified as plagiarized (False Positives: FP), and plagiarized texts incorrectly classified as clean (False Negatives: FN) are all utilized in the typical calculation of accuracy [52].

The number of correctly categorized cases divided by the total number of cases is accuracy, as shown in the formula below :

$$\text{Accuracy} = TP + TN / (TP+TN+FP+FN) \qquad (3)$$

Many authors have used accuracy metric for evaluation purposes, after extracting the obtained accuracy by the authors as illustrated in Fig 7, Dima et al. [38] obtained the highest accuracy 98.5 %, using word2vec model and vectors' cosine similarity which utilized to detect plagiarism.

The OSAC corpus was utilized in the study, the quality of the corpus affects the accuracy of vector representation, which in turn impacts the accuracy of plagiarism. As a result, if the changes are confined to single word replacements or the order of verbs and nouns has altered significantly, this methodology can discover similarities across texts.

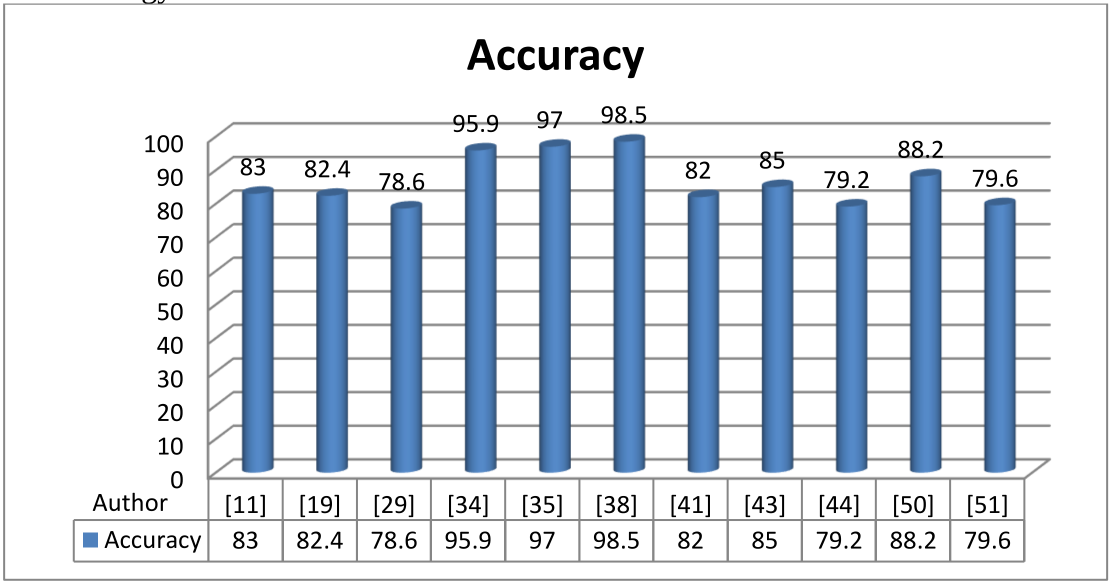

Figure 7. Compariosn among authors with respect of accuracy

4.3. Source Retrieval Approach

Several other authors have presented plagiarism systems that build the corpus through crawling related information from the web or source retrieval.

In [3] , to identify plagiarism, a web-based anti-plagiarism technique is used. The created text fragments (chunks) are saved in the database locally. Following that, all of these fragments (chunks) will be systematically matched with the Local database (local disks), Distributed database (LAN), and Global database (www)..

[53] Presented a web application that enables checking a bilingual text (English and Arabic). Through three famous search engines: Google, Bing, and Yandex SERP, the application looked for

duplicate content on the internet. It was discovered each search engine's top three results on the first page. Text vectorization was done with tf-idf. The machine next compares the suspect sentence to the recovered text fragment using the cosine text similarity methodology for all nine findings.

[54] Discussed plagiarism detection, text mining, web mining, and how to avoid plagiarism on the web. They utilized the TF- IDF approach for numeric representation of text with relevance to plagiarism detection. To indicate commonalities between papers, similarity measures were used. Purity determined the size of the total cluster. The accuracy of the assignment was established by counting the number of appropriately allocated documents and dividing by all documents N. Each cluster was assigned to the most often occurring class.

[48] 55 Introduced a text mining technique for detecting all of the common patterns between a suspect document and the documents in a reference database.

The technique is based on a pattern detection algorithm and a data structure that enables the computer to recognize all common patterns. The dataset is comprised of five different Wikipedia texts on various topics: near-copy (category B), light revision (category C), extensive revision (category D), and non-plagiarism (category E) (categories A and E).

In [49] 56 A plagiarism detection method is proposed, as well as a technique for source retrieval and text alignment. Plagiarism seeding was described as a measure of the similarity between phrases in suspect papers and sentences in source papers, while query generation was defined as rating terms in suspicious segments. Using the BM25 model for relevance ranking and the VSM, the created methodology used a method of source retrieval based on BM25 and a method of text alignment based on VSM, respectively. The methodology is utilized to develop a system for plagiarism detection.

[50] 57 Used the Google Search API, a plagiarism detection system was presented. The proposed framework was created to accommodate both Thai and English. Another significant distinction is that, rather than downloading entire Web pages, our methodology analyzes language patterns in search result snippets to enhance system response time.

5. Discussion

The authors used an existing corpus or dataset which has a large storage requirement with huge repository size of documents. This represents an obstacle for method efficiency when finding matching with many stored documents.

The used corpus or dataset, computer specifications, such as speed of CPU and RAM, along with programming language, such as Python and Java, all have an especially important effect on the obtained accuracy and precision of the PD.

Most of the papers do not consider the required time for PD since there are several factors that affect the PD such as size of the corpus, the used algorithm, the specifications of the used computer and other factors which all affect the system and make it difficult to measure the exact time.

It is obvious that authors who used string matching or term frequency-based methods could only find lexical text similarity-based PD. While the authors that used word embedding approach for text encoding can find both lexical and semantic text similarity-based PD.

Also, many authors have used the cosine length normalization because it is extremely popular and has had remarkable success.

With respect to the techniques for vector representation or encoding of a text, the study reveals that the majority of approaches use either word2vec or doc2vec for vector transformation, implying that these two representations are the best for maintaining the semantic component of a given text.

Despite the fact that each approach treats the text differently, the former converts texts into a list of words and the latter into a list of sentences, these representations produce results that differ from one approach to the next, the transformation of a text to a list of sentences, in our opinion, remains the most relevant because the meaning of the text being treated is kept in mind.

In terms of the methodologies applied for similarity classification, the previous sections discuss the various approaches applied to determine whether the studied texts are similar or not. Many systems include RNN and CNN in their architecture for detection of plagiarism, However, for their vector representation, the majority of them utilize the word level, hence they are only utilized to detect similarity between sentences, not texts.

It was discovered that almost all of these methodologies utilize the cosine measure to indicate semantic plagiarism and the Jaccard to indicate lexical plagiarism to assess plagiarism between any two texts or documents. Because it is possible to locate two papers that have the same words or sentences but are not semantically related. Furthermore, when texts are viewed as a collection of sentences or words, the semantic aspect can be lost. So, it is needed to come up with a solution that deals with this problem and represents a text as a set of sentences that will finally be converted into a set of vectors, as well as a process that preserves the semantic feature of this set of sentences, so it will be a manipulation that uses an algorithm like the DL algorithm to analyze a set of texts in order to find similarities.

Both the traditional and string-matching approaches for PD use only lexical features of the text and are inflexible when the size of the corpus is excessively large. However, these approaches are considered simple in implementation. On the other hand, intelligent approaches can utilize the text's lexical, syntactic, and semantic properties to detect plagiarism and are hence suitable for large sized corpora. However, they require special hardware and software specifications for implementation. The PD based on novel DL approaches provide more accurate results than other approaches.

Regarding authors that build a corpus through web crawling, one can use intelligent techniques such as clustering to speed up search, also using word2vec or doc2vec to encode text (vectorize) and deep learning algorithms for plagiarism detection process.

The PD systems based on results of search engines such as Google, Yahoo!, and others, need small storage requirements since they will not store thousands of documents on every detection, but they will require considerable time for every search if they not limit the count of search results that will be used for PD.

Each of the two approaches, corpus-based or web search result-based have their advantages and disadvantages. One can propose a hybrid approach that tries to make use of the advantages of both approaches, like using a dynamic updatable dataset which can be added to or delete from. Detecting the plagiarism is first done at the dataset, then the web search results are obtained and related text which does not exist in the dataset is added. Furthermore, using an intelligent technique for the web search and then adding the results to the dataset.

6. Conclusion

The review concludes that many authors have produced PD systems using different approaches based on the lexical, semantic, and other text features. Most of the authors have not considered the required time for PD because there are several factors that affect the PD, such as size of the corpus, the used algorithm, the specifications of the used computer, and other factors, all affecting the system and making it difficult to measure the exact time.

The traditional approaches for PD only deal with lexical features of the text, while the intelligent and DL approaches can capture the text's lexical, syntactic, and semantic properties for detecting plagiarism. That makes it better than the traditional approaches, especially when the corpus is extremely large in size.

Most papers have considered accuracy metric as evaluation criteria and approaches of PD are applied to different corpora with different languages. Each one obtains result according to the used corpus. This research has shown which roads to go in order to build a strategy, taking advantage of the strengths of each method while avoiding the flaws.

Acknowledgment
The authors would like to express heartfelt gratitude to the College of Computer and Mathematic Sciences at the University of Mosul, Iraq for the help and support.

References

[1] B. Akanksha, A. Anukruti, V. Tarjni, S. Desai and A. Nair, “A Survey on Plagiarism Detection”, *Advances in Computational Sciences and Technology,* ISSN 0973-6107 Volume 10, Number 8 , pp. 2359-2365, 2017 .

[2] M. Jiffriya, M. A. Jahan, R. G Ragel and S. Deegalla, “AntiPlag: Plagiarism Detection on Electronic Submissions of Text Based Assignments”, *IEEE 7th International Conference on Industrial and Information Systems,* 2013.

[3] J. Khatri and V. Mohan, “An Approach for Implementing Web-Based Tool for Plagiarism Detection” , *International Journal of Engineering and Management Research*, Volume-6, Issue-3, p.p. 57-60, 2016.

[4] H. A. Chowdhury, D. K. Bhattacharyya, “Plagiarism: Taxonomy, Tools and Detection Techniques”, *19th National Convention on Knowledge, Library and Information Networking,* 2018.

[5] E. M. Hambi and F. Benabbou,”A deep learning based technique for plagiarism detection: a comparative study”, *IAES International Journal of Artificial Intelligence (IJ-AI)* ,Vol. 9, No. 1,, pp. 81~90 , 2020.

[6] M. AlSallal, R. Iqbal, S. Amin, A. James and V. Palade, “An Integrated Machine Learning Approach for Extrinsic Plagiarism Detection”, *9th International Conference on Developments in eSystems Engineering,* 2016.

[7] G. M.Adel and A. Ghallab, “Performance Comparisons on Online Plagiarism Detection Software in Arabic Theses” , *International Conference on e-Commerce, e-Administration, e-Society, e-Education, and e-Technology,* 2014.

[8] S. M. Alzahrani, N. Salim, and A. Abraham, “Understanding Plagiarism Linguistic Patterns ,Textual Features, and Detection Methods”, *IEEE transactions on systems MAN and CYBERNETICS*, Part C: Applications and reviews, vol. 42, No. 2, 2012.

[9] H. Sherwani and M. Bargadiya, “A Survey on Plagiarism Detection Techniques and Understanding Textual Features” , *International journal of scientific progress and research (IJSPR),* ISSN: 2349-4689 Volume-14, Number – 01, 2015.

[10] I.M. I. Subroto and A. Selamat, “Plagiarism Detection through Internet using Hybrid Artificial Neural Network and Support Vectors Machine”, *TELKOMNIKA,* Vol.12, No.1, pp. 209-218, 2014.

[11] A.Y. Ichida, F. Meneguzzi and D. Ruiz , “Measuring Semantic Similarity Between Sentences Using A Siamese Neural Network”, *International Joint Conference on Neural Networks (IJCNN), IEEE,* 2018.

[12] E. M. Hambi and F. Benabbou, “A New Online Plagiarism Detection System based on Deep Learning”, *(IJACSA) International Journal of Advanced Computer Science and Applications,* Vol. 11, No. 9, p.p. 470 , 2020.

[13] M. Farouk, “Measuring Sentences Similarity: A Survey”, *Indian Journal of Science and Technology*, Vol 12(25), 2019.

[14] D. D. Prasetya, A. P. Wibawa and T. Hirashima, “The performance of text similarity algorithms” , *International Journal of Advances in Intelligent Informatics,* ISSN 2442-6571 Vol. 4, No. 1, pp. 63-69, 2018.

[15] C. K. Kent and N. Salim , “Web based Cross Language Semantic Plagiarism Detection”, *Ninth IEEE International Conference on Dependable*, Autonomic and Secure Computing, 2011.

[16] K. Vani and D. Gupta, “Study on Extrinsic Text Plagiarism Detection Techniques and Tools”, *Journal of Engineering Science and Technology Review*, Vol 9 , No (4), p.p. 150 – 164, 2016.

[17] B. Leonardo and S. Hansun , “Text Documents Plagiarism Detection using Rabin-Karp and Jaro-Winkler Distance Algorithms”, *Indonesian Journal of Electrical Engineering and Computer Science,* Vol. 5, No. 2, February 2017, pp. 462 – 471, 2017.

[18] P. Mahdavi, Z. Siadati and F. Yaghmaee , “Automatic External Persian Plagiarism Detection Using Vector Space Model”, *4th International Conference on Computer and Knowledge Engineering (ICCKE),* 2014.

[19] Abdul Aziz, E. C Djamal and R. Ilyas , “Paraphrase Detection Using Manhattan's Recurrent Neural Networks and Long Short-Term Memory”, *Proc. EECSI* - Bandung, Indonesia, 2019.

[20] W. Lin , N. Peng, C. Yen and S. Lin , “Online Plagiarism Detection Through Exploiting Lexical, Syntactic, and Semantic Information”, *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics,* pages 145–150,, Association for Computational Linguistics, 2012.

[21] J. Rodina, Y. Trofimova , A. Kutuzov and E. Artemova, “ELMo and BERT in semantic change detection for Russian”, *The 9th International Conference on Analysis of Images, Social Networks and Texts*, 2020.

[22] W. G. S. Parwita and I. N. Wahyu, “String Matching based Plagiarism Detection for Document in Bahasa Indonesia”, *5th International Conference on New Media Studies*, 2019.

[23] N. Akiva , “Authorship and Plagiarism Detection Using Binary BOW Features”, *Notebook for PAN at CLEF*, 2012 .

[24] S. W. Ang, H. QI, L. Kong and C. Nu, “Combination of VSM and Jaccard Coefficient for external plagiarism detection”, *Proceedings of the International Conference on Machine Learning and Cybernetics*, 2013.

[25] M. Jiffriya, M. Akmal , J. Roshan and G. Ragel, “Plagiarism Detection on Electronic Text based Assignments using Vector Space Model”, *8th International Conference on Information and Automation for Sustainability (ICIAfS)* , 2014.

[26] Z. Xiaoping, M. Xiaoxuan, and S. Honghong , “Research on a VSM-based E-homework Anti-plagiarism System,” *International Conference on Information Management, Innovation Management and Industrial Engineering., IEEE*, 2012.

[27] P. Mahdavi, Z. Siadati and F. Yaghmaee, “Automatic External Persian Plagiarism Detection Using Vector Space Model,” *4th International Conference on Computer and Knowledge Engineering (ICCKE),* 2014.

[28] M. Umadevi, “Document comparison based on TF-IDF metric,” *International Research Journal of Engineering and Technology (IRJET),* e-ISSN: 2395-0056, Vol.,07 Issue., 02, 2020.

[29] H. He, K. Gimpel, and J. Lin, “Multi-Perspective Sentence Similarity Modeling with Convolutional Neural Networks”, *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* Association for Computational Linguistics, 2015.

[30] Z. F. Alfikri and A. Purwarianti, "Detailed Analysis of Extrinsic Plagiarism Detection System Using Machine Learning Approach (Naive Bayes and SVM)" , *TELKOMNIKA Indonesian Journal of Electrical Engineering*, Vol. 12, No. 11, pp. 7794 – 7804, 2014.

[31] A.Mahmoud and M. Zrigui , "Semantic Similarity Analysis for Paraphrase Identification in Arabic Texts", *Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation*, pp 274–281, 2017 .

[32] M. Elamine , F. Bougares, S. Mechti and L. Hadrich Belguith, "Extrinsic Plagiarism Detection for French Language with Word Embeddings", *Intelligent Systems Design and Applications , Springer*, pp.217-224, 2020 .

[33] K. Babaa T. Nakatohb and T. Minamic "Plagiarism detection using document similarity based on distributed representation," *8th International Conference on Advances in Information Technology, IAIT2016,* Macau, China, Elseiver, 2016.

[34] E. Gharavi, K. Bijari, K. Zahirnia and H. Veisi, "A Deep Learning Approach to Persian Plagiarism Detection", *Conference FIRE* (Working Notes), p.p. 154-159, 2016.

[35] W. Bao_, W. Bao , J. Du , Y. Yang and X. Zha, " Attentive Siamese LSTM Network for Semantic Textual Similarity Measure", *International Conference on Asian Language Processing (IALP), IEEE*, 2018.

[36] H. He and J. Lin , "Pairwise Word Interaction Modeling with Deep Neural Networks for Semantic Similarity Measurement", *Proceedings of NAACL-HLT*, Association for Computational Linguistics, pages 937–948, 2016.

[37] T. K. Arachchi and E. Y. Charles, "Deep Learning Approach to Detect Plagiarism In Sinhala Text", *IEEE 14th International Conference on Industrial and Information Systems (ICIIS)*, Peradeniya, Sri Lanka, 2019.

[38] D. Suleiman , A. Awajan and N. Al-Madi , "Deep Learning Based Technique for Plagiarism Detection in Arabic Texts", *International Conference on New Trends in Computing Sciences, IEEE*, 2017.

[39] Z. Zhu , Z. He , Z. Tang , B. Wang and W. Chen, "A Semantic Similarity Computing Model based on Siamese Network for Duplicate Questions Identification", *CCKS Tasks*, 2018.

[40] J. Mueller and A. Thyagarajan , "Siamese Recurrent Architectures for Learning Sentence Similarity", *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*, 2016 .

[41] P. Neculoiu, M. Versteegh and M. Rotaru, "Learning Text Similarity with Siamese Recurrent Networks", *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 148–157, Association for Computational Linguistics, 2016.

[42] C. Shih, B. Yan, S. Liu and B. Chen , "Investigating Siamese LSTM Networks for Text Categorization", *Proceedings of APSIPA Annual Summit and Conference*, Malaysia, 2017.

[43] L. Yao, Z. Pan , and H. Ning, "Unlabeled Short Text Similarity, With LSTM Encoder", *IEEE Access* (Volume: 7),p.p.: 3430 – 3437, 2018.

[44] H. S. hammadi , M. H. Dezfoulian and M. M. zadeh, "Paraphrase detection using LSTM networks and handcrafted features", *Multimedia Tools and Applications, Springer* Science Business Media, LLC, 2020.

[45] A. Ichida, F. Meneguzzi and D. Ruiz , "Measuring Semantic Similarity Between Sentences Using A Siamese Neural Network," International Joint Conference on Neural Networks (IJCNN), IEEE, 2018.

[46] Z. Chi and B. Zhang, "A Sentence Similarity Estimation Method Based on Improved Siamese Network," *Journal of Intelligent Learning Systems and Applications*, Vol 10, pp. 121-134, 2018.

[47] N. Afzal , Y. Wang and H. Liu., “MayoNLP at SemEval-2016 Task 1: Semantic Textual Similarity based on Lexical Semantic Net and Deep Learning Semantic Model,” *Proceedings of SemEval*, pages 674–679, 2016.

[48] S. Zhang, Z. Liang, and J. Lin, "Sentence similarity measurement with convolutional neural networks using semantic and syntactic features", *Computers, Materials & Continua*, vol. 63, no.2, pp. 943–957, 2020.

[49] A.Mahmoud and M. Zrigui, “Sentence Embedding and Convolutional Neural Network for Semantic Textual Similarity Detection in Arabic Language”, *Arabian Journal for Science and Engineering*, https://doi.org/10.1007/s13369-019-04039-7, 2019.

[50] E. L. Pontes, S. Huet, A. C. Linhares and J. Torres-Moreno., “Predicting the Semantic Textual Similarity with Siamese CNN and LSTM”. *Traitement Automatique des Langues Naturelles (TALN)*, pp.311-319, 2018.

[51] B. Agarwal, H. Ramampiaro, H. Langseth and M. Ruocco, "A Deep Network Model for Paraphrase Detection in Short Text Messages". *In Information Processing & Management Journal (IPM)*, Elsevier, 54(6), pp. 922-937 , 2018.

[52] A.Chitra and A. Rajkumar, “Plagiarism Detection Using Machine Learning-Based Paraphrase Recognizer”, *Journal of Intelligent Systems*.; 25(3), p.p. 351–359, 2016.

[53] M. Alabbas, R. S. Khudeyer, M. Radif and H. K. Hameed, “Online multilingual plagiarism detection system using multi search engines”, *Journal of southwest JIAOTONG University*, DOI : 10.35741/issn.0258-2724.54.6.30, vol. 54 No. 6., 2019.

[54] D. B. Dasari and V. G. Rao. K, “Detecting The Plagiarism For Text Documents On The World Wide Web”, *International Journal of Social Relevance and Concern*, Vol 2, Issue 10, p.p. 6-10, 2014.

[55] K. Xylogiannopoulos , P. Karampelas and R. Alhajj, “Text mining for plagiarism detection: Multivariate pattern detection for recognition of text similarities”, *IEEE/ACM ASONAM*, Barcelona, Spain, August 28-31, 2018.

[56] L. Kong, Z. Zhao, Z. Lu, H. Qi and F. Zhao, “A Method of Plagiarism Source Retrieval and Text Alignment Based on Relevance Ranking Model”, *International Journal of Database Theory and Application*, Vol.9, No.12 , pp.35-44, 2016.

[57] S. Thaiprayoon and C. Haruechaiyasak , “Web plagiarism detection based on search result Snippets”, *In Proceedings of the 25th International Technical Conference on Circuits/Systems*, Computers and Communications (ITC-CSCC 2010), 2010.