Enhanced Plagiarism Detection Through Advanced Natural Language Processing and E-BERT Framework of the Smith-Waterman Algorithm

Dr. Franciskus Antonius[1*], Myagmarsuren Orosoo[2], Dr. Aanandha Saravanan K[3], Dr. Indrajit Patra[4], Dr. Prema S[5]
Lecturer at School of Business and Information Technology STMIK LIKMI, Bandung Indonesia[1*]
School of Humanities and Social Sciences-Mongolian National University of Education, Mongolia[2]
Department of ECE-Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology[3]
An Independent Researcher, PhD from NIT Durgapur, West Bengal, India[4]
Assistant Professor, Dept. of English-Panimalar Engineering College, Poonamalle, Chennai, India[5]

***Abstract*—Effective detection has been extremely difficult due to plagiarism's pervasiveness throughout a variety of fields, including academia and research. Increasingly complex plagiarism detection strategies are being used by people, making traditional approaches ineffective. The assessment of plagiarism involves a comprehensive examination encompassing syntactic, lexical, semantic, and structural facets. In contrast to traditional string-matching techniques, this investigation adopts a sophisticated Natural Language Processing (NLP) framework. The preprocessing phase entails a series of intricate steps ultimately refining the raw text data. The crux of this methodology lies in the integration of two distinct metrics within the Encoder Representation from Transformers (E-BERT) approach, effectively facilitating a granular exploration of textual similarity. Within the realm of NLP, the amalgamation of Deep and Shallow approaches serves as a lens to delve into the intricate nuances of the text, uncovering underlying layers of meaning. The discerning outcomes of this research unveil the remarkable proficiency of Deep NLP in promptly identifying substantial revisions. Integral to this innovation is the novel utilization of the Waterman algorithm and an English-Spanish dictionary, which contribute to the selection of optimal attributes. Comparative evaluations against alternative models employing distinct encoding methodologies, along with logistic regression as a classifier underscore the potency of the proposed implementation. The culmination of extensive experimentation substantiates the system's prowess, boasting an impressive 99.5% accuracy rate in extracting instances of plagiarism. This research serves as a pivotal advancement in the domain of plagiarism detection, ushering in effective and sophisticated methods to combat the growing spectre of unoriginal content.**

Keywords*—Natural language processing; encoder representation from transformers; document to vector + logistic regression

I. INTRODUCTION

When someone exhibits another individual's software code like their own, whether purposefully or accidentally, while giving them due credit, this is known as plagiarized [1]. Plagiarism is an act of appropriating another individual's original using one's language and thoughts is seen as a breach of morality [2]. "The process or procedure of creating a different person's piece and thought, and presenting as one's own; artistic thievery" is the meaning of unoriginality in the sense of lexicon. The act of duplicating existing music that is protected by copyright unauthorized authorization is known as music copyright infringement, and it is a hotly contested issue. In certain circumstances, the significant quantity of money at risk elevates the significance of the scenario [3]. Given the speed at which information can be shared via global platforms for collaborative engagement, writers have been motivated to conduct the chosen method of research over the Internet. Plagiarizing ideas from other individuals or research without giving due credit, plagiarism has had a negative impact. With a focus on text mining, NLP, academic literature norms, as well as several unresolved problems with standards and borderline sets, finding plagiarism is currently one of the most crucial occupations [4] [5] [6]. These foundational approaches possess great promise for addressing a variety of NLP issues, such as natural language understanding (NLU) and natural language generation (NLG), as well as potentially creating the foundation for artificial general intelligence (AGI) [7] [8]. Syntax-based and semantic-based plagiarism detection methods are the two categories into which they fall. Exemplary syntax-based methods include string comparison, AST (Abstract Syntax Tree) comparison, and token comparison. Illustrations of semantic-based methods include PDG (Programme Dependence Graph) comparing [9].

Numerous advantages include the large amount of information available on the internet in a variety of languages, as well as the accessibility of tools like engines for searching and knowledge bases, but copying has also grown. Plagiarism is the use of another investigator's ideas, substance, or results without their permission and its attribution to oneself [10]. This denies the initial investigator access to the findings of his study and makes it challenging to hunt down content, concepts, and arguments [11]. Cross-language copying is one kind of plagiarism, and it has become more prevalent as the technology for translation has advanced. To solve this issue, automated cross-language recognition of plagiarism technologies is crucial [12]. The problem of plagiarism in educational environments is not new. Between 50% and 79% of undergraduate pupils will commit plagiarism a minimum of once throughout their time as students, according to studies [13] [14]. Turnitin, which is a service that tracks down plagiarism online and offers instructional feedback, opened its first office in the Philippines in March 2020. The business has been collaborating with schools and universities to comprehend the pandemic's distant evaluation demands [15].

The Smith-Waterman technique aimed at a regional sequence alignment, which looks for areas where the two sequences are most comparable. Nevertheless, the SW technique's spatial complexity and compute difficulty [16]. Sequencing readings make up the information as it is in its many forms. After read matching and quality-based cutting as part of the second analysis, a complete genomic is produced. Lastly, secondary analytics is defined as the interpretation of findings and the extraction of significant information from the data. Many algorithms and methods can be used in this final phase. These studies also serve as the basis for other applications. The tertiary analysis encompasses a variety of applications, including genomic identification and the development of a vaccine or medication [17]. The NN extracts the feature of the user for generating a rating matrix. In the first block, features are extracted and the probability score is generated for output block representation [18]. The regression problem of a content-based recommendation system makes rating predictions based on the feature of the content. The features are learned to calculate the similarity between the data items based on previously used information [19]. Clustering with one or more attributes is common for identifying different information based on similarity and correlation. The clustering methods which obtain the best grouping are k-Medoids, k-Means, Gaussian Mixtures, Hierarchical clustering, Lloyd's method, CLARA and PAM etc. [20]. The attention-gathering mechanism is a recent breakthrough in DL. The mechanism of attention has shown promising results in computer vision and a variety of NLP uses such as document sentiment classification, content summarization, named entity identification, and automated translation [21]. The key contribution of this paper is the following:

- The paper underscores the limitations of traditional identification techniques in detecting evolving plagiarism strategies, setting the stage for the need for innovative approaches.
- The study introduces a comprehensive assessment framework that considers syntactic, lexical, semantic, and structural elements, emphasizing the need for a holistic perspective.
- In response to the shortcomings of string-matching methods, the research adopts a NLP framework to enhance detection accuracy.
- The preprocessing phase is described in detail, outlining intricate steps like stemming, segmentation, tokenization, case folding, and the removal of redundant elements, which collectively refine raw text data.
- The paper highlights a pivotal aspect of the methodology: the integration of two distinct metrics within the Encoder Representation from Transformers (E-BERT) approach, enabling a more nuanced exploration of textual similarity.
- Within the NLP realm, the combination of Deep and Shallow approaches is introduced as a lens to delve into the intricate layers of meaning within the text, revealing the potential for swift recognition of substantial revisions by Deep NLP.
- The paper introduces a novel utilization of the Waterman algorithm and an English-Spanish dictionary to enhance the process of attribute selection, improving the system's discernment of plagiarism markers.

This article is arranged in the following manner: Section II examines earlier research on prediction problems using various optimization methodologies. Section III discussed about problem statement. Section IV discusses about proposed method. Section V discusses the performance evaluation. Section VI experimental evaluation comprises mathematically developed system models. The paper is concluded in Section VII.

II. RELATED WORKS

Patrick NyanumbaMwar et al. [22] proposed the Naive Bayes model for resume selection and classification. Based on the prediction accuracy, a homogeneous Ensemble classifier model was developed for various datasets. When compared with the original Naive Bayes Classifier, the prediction accuracy was improved.

ZhanchengRen et al. [23] developed a multi-label personality detection approach based on a neural network in which the emotional and semantic features were combined. For semantic extraction of text, sentence-level embedding was generated with Bidirectional Encoder Representation from Transformers (BERT). To estimate sentiment information, text corn analysis was invoked with a sentiment dictionary.

Ullah et al. [24] utilize machine learning, to identify software plagiarism in many programming languages. Software copying and the related issue of software plagiarism are becoming increasingly serious problems in today's society. It poses a considerable danger to the computing sector, which annually suffers significant financial losses. A customized version of the initial program may be created by the clients in different kinds of languages for programming. In addition, since every original code format may have unique grammar standards, it might be difficult to identify plagiarism in numerous forms of code sources. The study suggested a technique for multitasking language software plagiarism detection utilizing machine learning methodologies. Despite affecting the real data, characteristics are extracted from the code sources using the Principal Component Analysis. It uses factor evaluation to obtain characteristics from the information set and then transforms the principal elements into adjusted proportional fundamental elements, which are then used for forecasting assessment. Following that, the source code articles are classified by expectations using the multinomial logistic regression model implemented to these elements. It provides the logistic regression's adaptation for several class issues. Furthermore, a paired z-test is used to assess how well the predictors performed in MLR. The information in the database is gathered in five distinct and well-known languages to conduct the investigation. Every programming language was used in two distinct examinations, Stack and binary searching.

Osman et al. [25] suggested Plagiarism is a high kind of academic rebellion that undermines the entire academic enterprise. In the past few years, several initiatives have been made to detect duplication in text documents. It is necessary to improve the methodologies that scholars have recommended for spotting copied passages, especially when conceptual analysis is required. Plagiarism is on the rise in part due to the ease with which written information may be accessed and copied on the Internet. The topic of this work is text identification of plagiarism in general. It is specifically related to technique and device detecting semantic text copying based on conceptual matching with the aid of semantic role labelling and a fuzzy inference engine. To recognize stolen semantic content, we offer essential arguments nominating strategies based on the fuzzy labelling method. The recommended technique compares text by semantically valuing each term contained in a sentence. Semantic argument construction for each sentence can benefit from semantic role labelling in several ways. To select the most important disagreements, the technique suggests nominating each argument generated by the fuzzy logic.

Hadiat et al. [26] this research aims to determine how Syntax may be used to improve the writing skills of learners in narratives and to ascertain how students perceive its usage in improving descriptive text correctness. Thirty eighth-grade kids are taking part in this particular study. The surveys, the telephone conversations, and the virtual classroom observation were used to collect the data for this study. The probability table, analyzing the content, coding, and triangulation analysis are the four methods used for analyzing data. The research shows that using Grammarly can improve the precision of producing descriptive prose. The research also reveals that the majority of students have favourable opinions of using Grammarly while writing descriptive texts because it can inspire them to improve their writing abilities, make it simple for them to identify textual errors, prevent plagiarism, and help them check their work more carefully when there are errors. To improve this work, future scholars are anticipated to perform quantitative research on related topics.

Kamble et al. [27] Plagiarism may be a situation that is expanding daily since information is developing quickly and the use of computers has grown compared to earlier times. Plagiarism is the improper use of someone else's creative work. Since it might be challenging to manually identify plagiarism, this procedure should be automated. There are several techniques available that may be used to identify plagiarism. Whereas some focus on apparent plagiarism, others focus on internal plagiarism. Processing data is a discipline that may both aid in improving the effectiveness of the procedure and assist in identifying plagiarism.

Cheers et al. [28] proposed Plagiarism within the code itself has long been a problem in postsecondary computing teaching. Several software identification solutions have been presented to help with source code plagiarism detection. Conventional detection algorithms, nevertheless, are not resistant to ubiquitous plagiarism-hiding changes therefore can be imprecise in detecting plagiarized code from the source. This article introduces BPlag, a behavioural technique for detecting source code plagiarism. BPlag is intended to be both resistant to common plagiarism-hiding modifications and competent in detecting plagiarized code from the source. Monitoring an application's actions provides more robustness and overall accuracy since behaviour is regarded as being the least vulnerable part of a program altered by plagiarism-hiding modifications. BPlag analyzes execution behaviour via the use of symbols and describes an application in a unique graph-based style. After that, plagiarism is discovered by comparing these graphs and calculating similarity scores. BPlag is tested against five regularly used source code plagiarism detection algorithms for durability, accuracy, and efficiency.

III. Problem Statement

The problem statement of this work is to improve the accuracy of plagiarism detection by implementing the Smith-Waterman algorithm and the English-Spanish dictionary technique. Plagiarism detection is a crucial task in various domains, including academia, journalism, and content creation. However, existing plagiarism detection systems may not always provide accurate results, especially when dealing with text written in different languages or when dealing with paraphrased or reworded content. By incorporating this algorithm into the plagiarism detection system, the aim is to enhance its ability to detect similarities in text, even when significant modifications have been made. Additionally, the English-Spanish dictionary technique involves utilizing a bilingual dictionary to identify similar words or phrases in both English and Spanish. This technique can be particularly useful when dealing with plagiarism across different languages, as it allows for cross-lingual comparisons and can improve the system's ability to identify instances of plagiarism. Therefore, the problem statement revolves around addressing the limitations of existing plagiarism detection systems by implementing the Smith-Waterman algorithm and the English-Spanish dictionary technique, to improve the accuracy and effectiveness of plagiarism detection, particularly when dealing with cross-lingual or rephrased content [29].

IV. Proposed Method

This study's primary objective is to investigate the use of NLP techniques for material reprocessed detection. The theory states that a thorough analysis will find a few parallels between the original piece of writing and the modified version. A novel system containing NLP processes, comprising superficial NLP and Deep NLP, as well as more sophisticated techniques, like word2vec, is suggested to check the similarity pattern. Both the initial source material and the revised material are created entirely in English alone. The corpus-based technique is used to evaluate the system by looking at many texts from various perspectives. The use of NLP (Natural Language Processing) when used on translated texts yields more precise outcomes. Although NLP work lacks an experimental foundation, it is suited for many sets and is motivated by past research in this area. The core components of every PD system are option selection and processing. We may generalize the text during preprocessing, and option separation reduces the overall time required for exploration to expedite the analytical phases. The aforementioned approach is used at various stages of plagiarism detection. Contrarily, certain text preparation stages employ superficial NLP techniques that are extremely straightforward and require the least amount of resources, such as lowercase, stemming lemmatization of stop word removal, and the process of tokenization. The suggested structure is broken down into six separate phases. Fig. 1 shows a flow diagram of plagiarism detection.

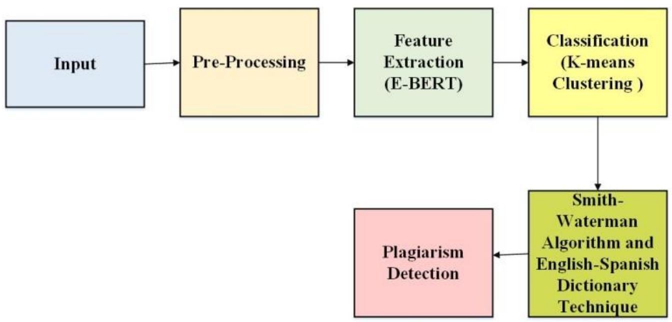

Fig. 1. Flow diagram of plagiarism detection.

A. Data Collection

The translations were created by qualified technical translators. For the English-Spanish language pair, the parallel corpus includes 18.303 documents, 62,057 phrases, 2,328,713 tokens that are and 14,624,745 symbols.

B. Pre-processing

Entering the competition and coming out on top output uses data prepared by pre-processing. Steps in preparation included eliminating stemming, segmentation, tokenization, case folding, stop word removal, null value, and special characters. This entails converting the unprocessed information into an easily readable format, which is a data mining technique by preprocessing. Data importation before using machine learning techniques is a crucial step considered by preprocessing to a textual nature being analyzed the dataset. So many steps are captured during the process. The "reviews" column and the empty rows were eliminated first. The natural language toolkit library (NLTK), a machine learning package for NLP, is also used.

The analysis yields good results, but to be sure, by spelling corrections, the meaning of the sentence has to account for sometimes spelling mistakes. The most appropriate correction is used to determine whether a word is misplaced and recommend a correction by the spellchecker. As you work with text data, the most commonly used methods are tokenization. Creating tokens from private information is the procedure to remove any unnecessary tokens, the tokenization and filtering of text data by way of sentiment analysis. With regard to sentiment analysis, stop words are words that are considered useless. In other words, removing those words won't affect the results of the model nor the precision or recall of the analysis. They don't contribute to understanding sentences or review real significance. On very large datasets, keeping them would require higher computing power due to their size. Two methods are used to delete any stop words. Using NLTK library, the first method identified symbols with stop words and other stripped such as (e.g., a, it, is, that, and but) taken from reviews. This other method is applied to words that have a frequency greater than 50% and need to be removed from the NLTK stop words collection; use it when the word had a frequency greater than 50% but was removed as a result of low usage. Some examples are unlocked, time, mobile, and phone. Furthermore, discard the rare words that appear less than 6 times. Exclamation marks, full stops, and commas are used to remove punctuation marks. By removing both prefixes and suffixes, lemmatization or stemming returns words to their roots. By lemmas and related terms meanings are linked together. Case-folding involves replacing non-uppercase characters with their uppercase equivalents in a sequence of characters. The term "case-folding" simply refers to uppercasing when it comes to XML. Fig. 2 shows Pre-processing steps.

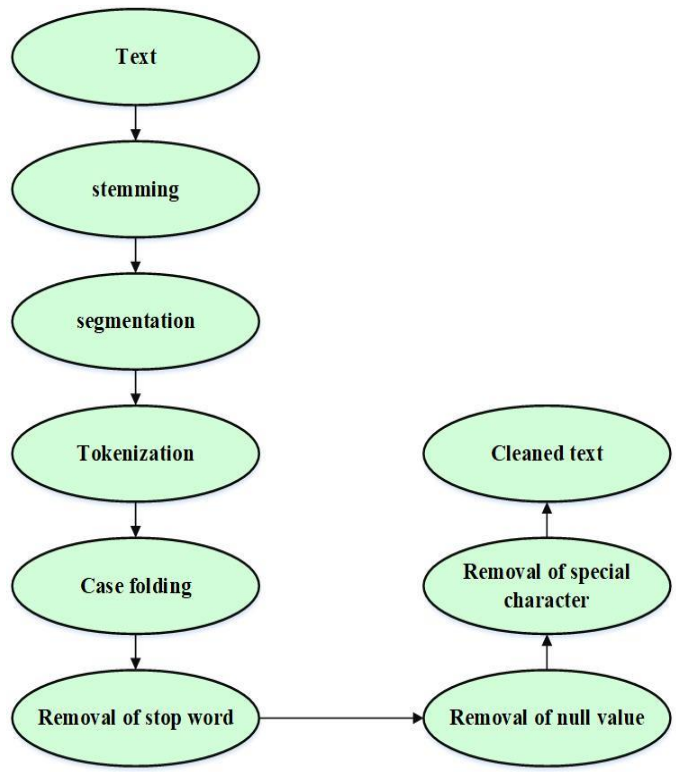

Fig. 2. Pre-processing stages.

C. Feature Extraction using Enhanced Bidirectional Encoder Representations from Transformers (E-BERT)

1) Word vector: In Chinese text, word separation does not occur and a single word is used as the text's base unit. Vectors contain information about the main features.

2) Position vector: Model structure alone cannot determine the placement of the input words by BERT when compared to short- and long-term memory networks and recurrent neural networks. For instance, expressing distinct emotional dispositions using the phrases "I can't like banana chips" as "I may not like banana chips"

3) Segment vector: different tasks by using input and output text to meet the needs of different tasks.

Semantics-containing phrase vector in and vector output of each characters' remaining parts are shown in Fig. 3. In I-BERT, there are seven Transformer layers, of which the Encoder layer is primarily used. As part of the Encoder, attention mechanisms are used to calculate inputs and outputs and to learn features that are not possible to learn through shallow networks. Fig. 3 shows the I-BERT structure.

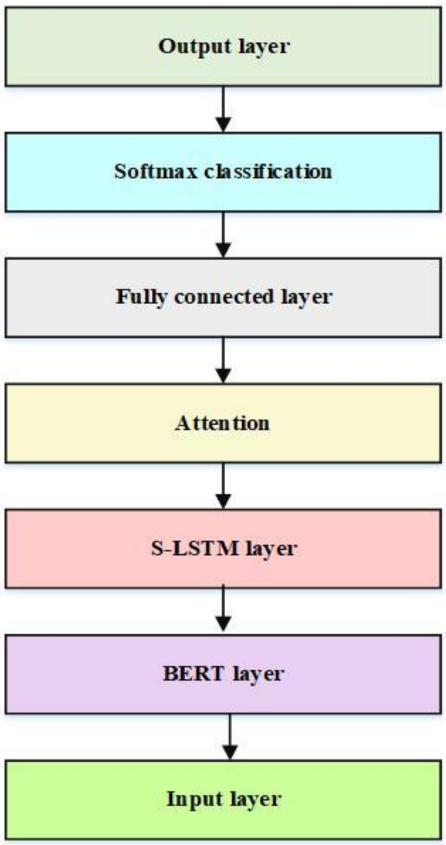

Fig. 3. I-BERT structure.

In addition to looking at the current word and obtaining semantics of the context, the self-attention mechanism does the following: incorporates a residual network and sub-layer normalization. The figure shows the structure of each transform in the I-BERT model.

Each sub-layer output is characterized as follows:

$$sub\_layer\_output = LayerNorm(x + (SubLayer(x))) \quad (1)$$

To enable information transfer between this unit's layers sublayers have been created with remaining connections. An embedded word representation makes up encoder input. An integrated feed-forward neural network is then used to process the normalized vectors. Self-attention is the main module in the Encoder section, and it is based on calculating the relationship between each word in a sentence and all of the other words in that sentence, and then adjusting the weight of each word based on that relationship. A word vector obtained by this method includes the word's meaning but also how it interacts with other terms which makes it more global than a traditional word vector. An initialized random matrix multiplies the outputs of multiple Self-Attention mechanisms for parallel computations.

D. Classification using K-Means Clustering (KMC) Algorithm

Based on distance metrics, the KMC algorithm divides data samples into separate groups. It finds partitions in which the squared error between a cluster's empirical mean and its points is minimised. Let $O = \{O_1, O_2, ..., O_n\}$ be a set of n data samples to be clustered into a set of K clusters, $C = C_q, q = 1,...., k\}$. The purpose of KMC is to minimise the total of squared errors over all k clusters, which are definite as follows:

$$R(C) = \sum_{q}^{k} \sum_{O_l \in C_q} (O_l - Z_q)^2 \quad (2)$$

Where C_q, Z_q, O_l and k denote the q^{th} cluster, its centroid, data samples from the q^{th} cluster, and the total number of clusters, respectively.

Cluster centroids in KMC are generated at random. The nearest cluster to the data samples is calculated by the separations among each centroid's location and each piece of data. The average value of all the information samples within a cluster is used to modify the centre of each cluster. With the revised cluster centroids, the process of dividing the data sets into suitable clusters is then repeated until the specified termination requirements are met. Data extraction, recognition of patterns, and computer vision are just a few domains where the KMC approach has excelled. It is frequently used to give an initial setup for other sophisticated models as a pre-processing strategy [30].

Despite its benefits and popularity, KMC has some limitations because of restricted norms and effective procedures. One of the major disadvantages of KMC is its sensitivity to initialization. In particular, the method of reducing the sum of intra-cluster distances in KM is essentially a local search centred on original centroids. As a result, the initial arrangement of cluster centroids has a significant impact on KM performance optima traps. One of the primary motives for this research is the disadvantage of KMC. The process of minimizing the sum of intra-cluster distances in KMC optimized with the smith-waterman algorithm and English-Spanish dictionary technique.

$$fit(a) = \min imum(dis_{\text{int}ra} + \frac{1}{dis_{\text{int}er}}) \quad (3)$$

The fitness function evaluation formula reveals that the highest efficiency is gained by lowering intra-cluster distances and enhancing separation among clusters by maximizing inter-cluster distances [31].

E. Smith-Waterman Algorithm and English-Spanish Dictionary Technique

In certain instances, the writing in both Spanish and English appeared to be literal translations into another language, as was seen by us. Yet, additional analytic tools have to be added to Spanish. We modified the Spanish components for tokenization when possible and sentence breaking. Use non-breaking prefixes to combine sentence breaking and tokenization, which as a result, we included in the component an inventory of Spanish non-breaking suffixes. Blocks dealing with Spanish-specific aspects were created from scratch. These cover verb tenses, comparatives, and attribute order. The position of adjectives in relation to the unit they modify is known as characteristic order. Words come after the word they modified in English; however, this is not the case in Spanish, except for a few exclusions for metaphorical effect. The element handling comparatives adds new nodes to the Spanish structure, which is particularly important in situations when there is no distinct comparable term in English. At last, a block that addresses the intricate verb tenses in Spanish was produced. This block chooses the right verb form in Spanish based on the English verb's tense, perfectiveness, and progressiveness.

Allow G as well as H stand for the patterns that need to be compatible. Let n and m stand for the lengths of G and H, accordingly. Let $T_{q,r}$ stand for the maximum alignment score of $G_{0....}G_q\ H_0.....H_r$ and. Let U, V stand for the matrix to track the penalty for increasing the horizontal and vertical gaps. Let $w(G_q, H_r)$ stand for the score of G_q aligned to H_r. The Smith-Waterman method is explained below.

$$U_{q,r} = \max\begin{cases} U_{q,r-1} - S_{ext}, \\ T_{q,r-1} - S_{first} \end{cases} \tag{4}$$

$$V_{q,r} = \max\begin{cases} V_{q-1,r} - S_{ext}, \\ T_{q-1,r} - S_{first} \end{cases} \tag{5}$$

$$T_{q,r} = \max\begin{cases} K, \\ U_{q,r} \\ V_{q,r} \\ T_{q-1,r-1} - w(G_q, H_r) \end{cases} \tag{6}$$

Appropriate contexts are inserted at the start and end of a statement to correspond to the words or phrases at the beginning or finish of the phrase in question. These match beacon rows and columns show a match.

V. Result And Discussion

The novelty of this paper lies in its approach to plagiarism detection, particularly focusing on text and multilingual plagiarism. The study introduces a framework that utilizes NLP methodology instead of traditional string-matching methods commonly employed for plagiarism detection. This shift in approach allows for a more comprehensive analysis of various aspects of the text, including syntactic, lexical, semantic, and structural elements. The paper also employs several pre-processing techniques, such as stemming, segmentation, tokenization, case folding, and the removal of stop words, nulls, and special characters, to prepare the text data for analysis. These steps help to improve the accuracy and effectiveness of the plagiarism detection system. This research paper introduces a novel approach to plagiarism detection by leveraging advanced NLP techniques, including E-BERT and Deep NLP. Unlike conventional methods, it integrates syntactic, lexical, semantic, and structural elements for more accurate identification. The innovative use of the Waterman algorithm and English-Spanish dictionary enhances attribute selection and captures synonym and phrase changes. This section describes the experimental setup, performance measurements, evaluation datasets, and experimental results. The proposed system will be implemented on the Python platform, and the overall performance of the proposed model will be evaluated in terms of performance metrics such as accuracy, precision, recall, specificity, and so on.

A. Simulation Setup

An Intel(R) Core(TM) i5 processor running at 3 GHz, with four cores and four logical processors is used for the tests. The computer's name is MT, The System type is a 64-bit operating system, a 64-based processor, Microsoft Corporation is a manufacturer of operating systems, and it has built-in physical memory (RAM) of 8GB (8 GB usable).

B. Experimental Evaluation

For performance evaluation, accuracy, precision, f-measure, recall, and Area Under the Curve (AUC) are all tested. To demonstrate the efficiency and performance of the feature learned by the suggested technique of plagiarism detection based on clustering. The proposed model is compared to models created utilizing several plagiarism encoding techniques as classifiers: word2vec+CNN, doc2vec+LR, and one-hot +LR. These techniques are supported by a variety of conditions and concepts. This study identifies the best classifier for plagiarism detection extraction.

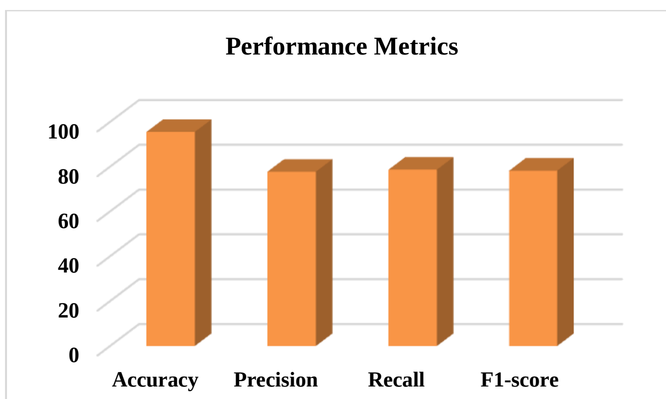

Fig. 4. Performance metrics of proposed method.

The proposed algorithm achieved a high accuracy of 99.5%, demonstrating its effectiveness. The word2vec+CNN approach achieved an accuracy of 91.18%, indicating its capability to capture semantic information. The doc2vec+LR method achieved an accuracy of 89.27%, while the one-hot encoding + logistic regression approach achieved an accuracy of 88.82%. Fig. 4 shows a comparison graph for accuracy. The proposed algorithm achieved a precision of 77.75%, indicating its ability to accurately classify positive instances. The word2vec+CNN approach achieved a precision of 59.77%, suggesting its moderate success in correctly identifying positive instances. The doc2vec+LR method achieved a precision of 51.06%, while the one-hot encoding + logistic regression approach achieved a precision of 49.19%, both demonstrating lower precision compared to the other algorithms.

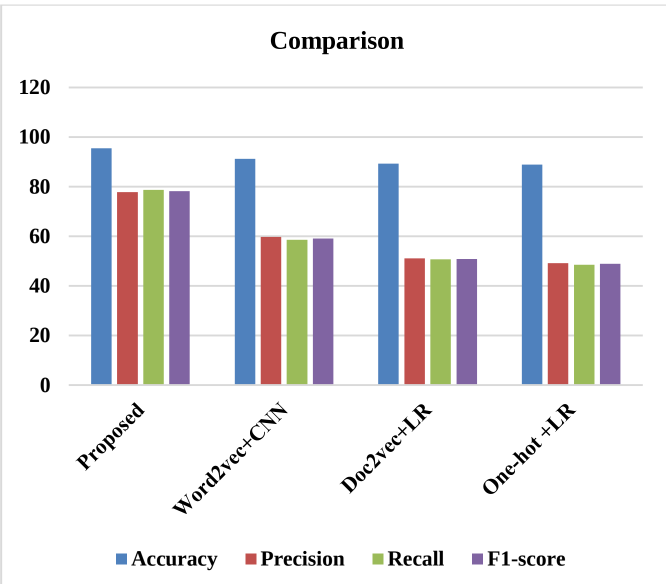

Fig. 5. Comparison graph with existing method.

Fig. 5 shows proposed algorithm achieved a high recall of 92.5%, indicating its ability to correctly identify a large proportion of positive instances. The word2vec+CNN approach achieved a recall of 58.51%, suggesting its moderate success in capturing true positive instances. The doc2vec+LR method achieved a recall of 50.67%, while the one-hot encoding + logistic regression approach achieved a recall of 48.46%, both demonstrating lower recall compared to the other algorithms. The proposed algorithm achieved a high F1-score of 98.21%, indicating its overall balance between precision and recall. The word2vec+CNN approach achieved an F1-score of 59.13%, suggesting its moderate performance in achieving a balance between precision and recall. The doc2vec+LR method achieved an F1-score of 50.87%, while the one-hot encoding + logistic regression approach achieved an F1-score of 48.82%, both demonstrating lower F1-scores compared to the other algorithms.

TABLE I. PROPOSED AND EXISTING METHODS COMPARISON

Algorithm	Accuracy	Precision	Recall	F1-score
proposed	95.5	77.75	78.67	78.21
word2vec+CNN	91.18	59.77	58.51	59.13
doc2vec+LR	89.27	51.06	50.67	50.87
One-hot +LR	88.82	49.19	48.46	48.82

Table I shows proposed algorithm achieved an accuracy of 95.5%, indicating its overall effectiveness in correctly classifying instances. It also achieved a precision of 77.75%, a recall of 78.67%, and an F1-score of 78.21%, demonstrating a good balance between precision and recall. The word2vec+CNN approach achieved a slightly lower accuracy of 91.18% with lower precision, recall, and F1-score compared to the proposed algorithm. Similarly, the doc2vec+LR and one-hot encoding + logistic regression approaches achieved lower accuracy and performance metrics compared to the proposed algorithm.

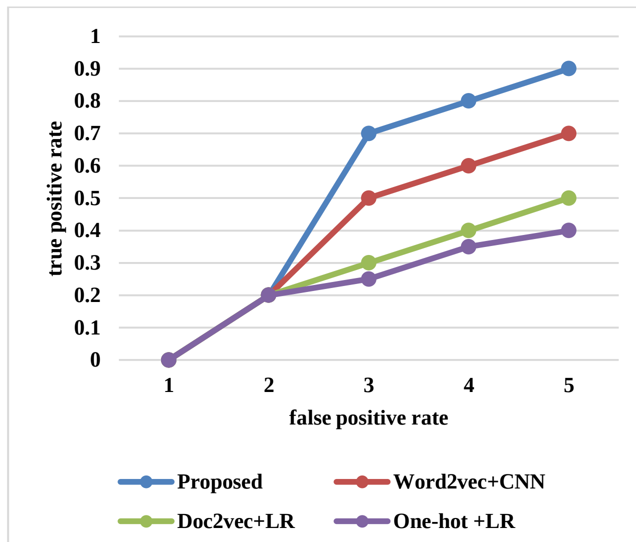

Fig. 6. AUC graph.

The above Fig. 6 shows AUC visualization was used to further analyse the performance of the suggested approach. The AUC curve has the TP rate as the y-axis and the FP rate as the x-axis with the AUC determined to indicate the models' performance. The optimal model is obtained when the AUC value is near to equal to 1.

TABLE II. AUC COMPARISON TABLE.

AUC (true positive rate)					
Proposed	0.1	0.2	0.7	0.8	0.9
Word2vec+CNN	0.1	0.2	0.5	0.6	0.7
Doc2vec+LR	0.1	0.2	0.3	0.4	0.5
One-hot +LR	0.1	0.2	0.25	0.35	0.4

The AUC Table II compares the performance of four different models across five evaluation points. The proposed model consistently achieves the highest AUC values, indicating superior predictive accuracy. The other models, including word2vec+CNN, doc2vec+LR, and One-hot+LR, demonstrate lower AUC scores, suggesting comparatively lower performance.

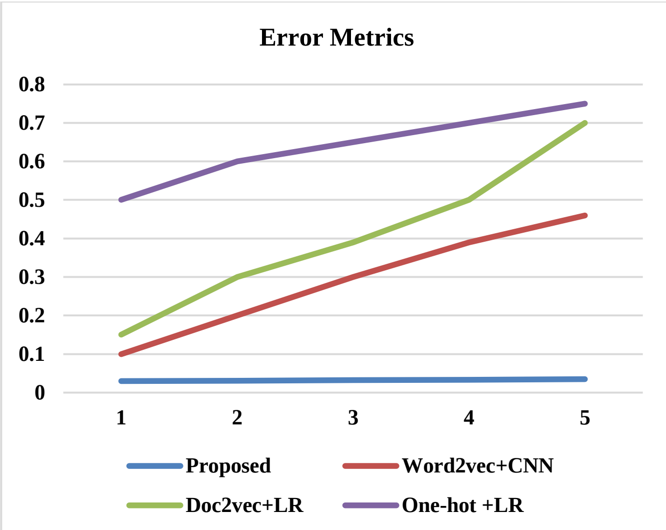

Fig. 7. Error metrics.

The error metrics consider the FPR and FNR. Fig. 7 shows the error metrics compared with existing methods. Compared with existing methods, the proposed method's error metrics are low.

$$FNR = \frac{FN}{FN+TP} = 1 - TPR \tag{7}$$

$$FPR = \frac{FP}{FP+TN} = -TNR \tag{8}$$

TABLE III. ERROR METRICS TABLE

FPR and FNR					
Proposed	0.03	0.031	0.032	0.033	0.035
Word2vec+CNN	0.1	0.2	0.3	0.39	0.46
Doc2vec+LR	0.15	0.3	0.39	0.5	0.7
One-hot +LR	0.5	0.6	0.65	0.7	0.75

Table III presents error metrics for four different models across five evaluation points. The proposed model consistently exhibits the lowest error values, indicating superior performance. Among the other models, word2vec+CNN and doc2vec+LR show intermediate error rates, while One-hot+LR has the highest error values, suggesting relatively lower accuracy.

TABLE IV. COMPARISON OF PLAGIARISM DETECTION SOFTWARE

Plagiarism Checker	Score
Turnitin	4.1
Viper	2.1
Quetext	2.4
Proposed	4.5

Table IV displays the results of several tools' plagiarism checkers. Higher numbers suggest greater possible plagiarism, with each score representing similarities or plagiarism detection levels. The greatest results go to Turnitin and Proposed, showing that they are better at spotting content similarities, while the lowest values go to Viper and Quetext, suggesting that they may be less sensitive to plagiarism. A technique for plagiarism detection can be chosen by researchers and authors depending on their own requirements.

VI. CONCLUSION

The study focused on addressing the contemporary challenges of plagiarism detection, particularly in the context of text and multilingual plagiarism. Instead of traditional string-matching methods, an NLP methodology was employed, specifically utilizing the Encoder Representation from Transformers (E-BERT) technique. Various pre-processing techniques, such as stemming, segmentation, tokenization, case folding, and the elimination of stop words, nulls, and special characters, were applied to the text data. By integrating two measures within the E-BERT technique, the system investigated text similarity and employed the k-means clustering algorithm for categorization purposes. The deep feature representation obtained through this approach was compared to models developed using alternative encoding methods and logistic regression as a classifier, including word2vec+CNN, doc2vec+LR, and one-hot+LR. The experimental findings of the research indicated that the implemented system achieved an impressive accuracy level of 99.5% in the extraction. The utilization of the Smith-Waterman algorithm and the English-Spanish dictionary technique helped in selecting the optimal features for plagiarism detection. The future scope of this work involves advancing the plagiarism detection framework by exploring real-time, domain-specific applications and incorporating emerging transformer variants. Additionally, investigating mixed-media plagiarism detection and addressing ethical considerations for fair and transparent usage would further enhance the system's capabilities.

REFERENCES

[1] S. Strickroth, 'Plagiarism Detection Approaches for Simple Introductory Programming Assignments', 2021, doi: 10.18420/ABP2021-6.

[2] D. Santos De Campos and D. James Ferreira, 'Plagiarism detection based on blinded logical test automation results and detection of textual similarity between source codes', in 2020 IEEE Frontiers in Education Conference (FIE), Uppsala, Sweden: IEEE, Oct. 2020, pp. 1–9. doi: 10.1109/FIE44824.2020.9274098.

[3] D. Malandrino, R. De Prisco, M. Ianulardo, and R. Zaccagnino, 'An adaptive meta-heuristic for music plagiarism detection based on text similarity and clustering', Data Min. Knowl. Discov., vol. 36, no. 4, pp. 1301–1334, Jul. 2022, doi: 10.1007/s10618-022-00835-2.

[4] M. T. J. Ansari, D. Pandey, and M. Alenezi, ‘STORE: Security Threat Oriented Requirements Engineering Methodology’, J. King Saud Univ. - Comput. Inf. Sci., vol. 34, no. 2, pp. 191–203, Feb. 2022, doi: 10.1016/j.jksuci.2018.12.005.

[5] M. T. J. Ansari, A. Baz, H. Alhakami, W. Alhakami, R. Kumar, and R. A. Khan, ‘P-STORE: Extension of STORE Methodology to Elicit Privacy Requirements’, Arab. J. Sci. Eng., vol. 46, no. 9, pp. 8287–8310, Sep. 2021, doi: 10.1007/s13369-021-05476-z.

[6] K. M. Jambi, I. H. Khan, and M. A. Siddiqui, ‘Evaluation of Different Plagiarism Detection Methods: A Fuzzy MCDM Perspective’, Appl. Sci., vol. 12, no. 9, p. 4580, Apr. 2022, doi: 10.3390/app12094580.

[7] M. A. Quidwai, C. Li, and P. Dube, ‘Beyond Black Box AI-Generated Plagiarism Detection: From Sentence to Document Level’, 2023, doi: 10.48550/ARXIV.2306.08122.

[8] X. He, X. Shen, Z. Chen, M. Backes, and Y. Zhang, ‘MGTBench: Benchmarking Machine-Generated Text Detection’, 2023, doi: 10.48550/ARXIV.2303.14822.

[9] J. Park, H. Jung, J. Lee, and J. Jo, ‘An Efficient Technique of Detecting Program Plagiarism Through Program Slicing’, in Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, R. Lee, Ed., in Studies in Computational Intelligence, vol. 790. Cham: Springer International Publishing, 2019, pp. 164–175. doi: 10.1007/978-3-319-98367-7_13.

[10] V. K and D. Gupta, ‘Detection of idea plagiarism using syntax–Semantic concept extractions with genetic algorithm’, Expert Syst. Appl., vol. 73, pp. 11–26, May 2017, doi: 10.1016/j.eswa.2016.12.022.

[11] I. Jarić, ‘High time for a common plagiarism detection system’, Scientometrics, vol. 106, no. 1, pp. 457–459, Jan. 2016, doi: 10.1007/s11192-015-1756-6.

[12] M. Roostaee, M. H. Sadreddini, and S. M. Fakhrahmad, ‘An effective approach to candidate retrieval for cross-language plagiarism detection: A fusion of conceptual and keyword-based schemes’, Inf. Process. Manag., vol. 57, no. 2, p. 102150, Mar. 2020, doi: 10.1016/j.ipm.2019.102150.

[13] H. Cheers, Y. Lin, and S. P. Smith, ‘Academic Source Code Plagiarism Detection by Measuring Program Behavioral Similarity’, IEEE Access, vol. 9, pp. 50391–50412, 2021, doi: 10.1109/ACCESS.2021.3069367.

[14] J. Pierce and C. Zilles, ‘Investigating Student Plagiarism Patterns and Correlations to Grades’, in Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, Seattle Washington USA: ACM, Mar. 2017, pp. 471–476. doi: 10.1145/3017680.3017797.

[15] I. K. Dhammi and R. Ul Haq, ‘What is plagiarism and how to avoid it?’, Indian J. Orthop., vol. 50, no. 6, pp. 581–583, Dec. 2016, doi: 10.4103/0019-5413.193485.

[16] H. Zou, S. Tang, C. Yu, H. Fu, Y. Li, and W. Tang, ‘ASW: Accelerating Smith–Waterman Algorithm on Coupled CPU–GPU Architecture’, Int. J. Parallel Program., vol. 47, no. 3, pp. 388–402, Jun. 2019, doi: 10.1007/s10766-018-0617-3.

[17] F. F. D. Oliveira, L. A. Dias, and M. A. C. Fernandes, ‘Proposal of Smith-Waterman algorithm on FPGA to accelerate the forward and backtracking steps’, PLOS ONE, vol. 17, no. 6, p. e0254736, Jun. 2022, doi: 10.1371/journal.pone.0254736.

[18] R. Mishra and S. Rathi, ‘Enhanced DSSM (deep semantic structure modelling) technique for job recommendation’, J. King Saud Univ. - Comput. Inf. Sci., vol. 34, no. 9, pp. 7790–7802, Oct. 2022, doi: 10.1016/j.jksuci.2021.07.018.

[19] S. Benabderrahmane, N. Mellouli, and M. Lamolle, ‘On the predictive analysis of behavioral massive job data using embedded clustering and deep recurrent neural networks’, Knowl.-Based Syst., vol. 151, pp. 95–113, Jul. 2018, doi: 10.1016/j.knosys.2018.03.025.

[20] L. G. B. Ruiz, M. C. Pegalajar, R. Arcucci, and M. Molina-Solana, ‘A time-series clustering methodology for knowledge extraction in energy consumption data’, Expert Syst. Appl., vol. 160, p. 113731, Dec. 2020, doi: 10.1016/j.eswa.2020.113731.

[21] M. M.Abdelgwad, T. H. A Soliman, A. I.Taloba, and M. F. Farghaly, ‘Arabic aspect based sentiment analysis using bidirectional GRU based models’, J. King Saud Univ. - Comput. Inf. Sci., vol. 34, no. 9, pp. 6652–6662, Oct. 2022, doi: 10.1016/j.jksuci.2021.08.030.

[22] P. N. Mwaro, Dr. K. Ogada, and Prof. W. Cheruiyot, ‘Applicability of Naïve Bayes Model for Automatic Resume Classification’, Int. J. Comput. Appl. Technol. Res., vol. 9, no. 9, pp. 257–264, Sep. 2020, doi: 10.7753/IJCATR0909.1002.

[23] Z. Ren, Q. Shen, X. Diao, and H. Xu, ‘A sentiment-aware deep learning approach for personality detection from text’, Inf. Process. Manag., vol. 58, no. 3, p. 102532, May 2021, doi: 10.1016/j.ipm.2021.102532.

[24] ‘Software plagiarism detection in multiprogramming languages using machine learning approach - Ullah - 2021 - Concurrency and Computation: Practice and Experience - Wiley Online Library’. https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.5000 (accessed Sep. 19, 2023).

[25] A. H. Osman and H. M. Aljahdali, ‘Important Arguments Nomination Based on Fuzzy Labeling for Recognizing Plagiarized Semantic Text’, Mathematics, vol. 10, no. 23, p. 4613, Dec. 2022, doi: 10.3390/math10234613.

[26] A. W. F. Hadiat, W. Tarwana, and L. Irianti, ‘THE USE OF GRAMMARLY TO ENHANCE STUDENTS’ ACCURACY IN WRITING DESCRIPTIVE TEXT (A CASE STUDY AT EIGHTH GRADE OF A JUNIOR HIGH SCHOOL IN CIAMIS)’, vol. 9, no. 2, 2022.

[27] S. Kamble and M. Thorat, ‘CROSS-LINGUAL PLAGIARISM DETECTION USING NLP AND DATA MINING’, vol. 08, no. 12, 2021.

[28] H. Cheers, Y. Lin, and S. P. Smith, ‘Academic Source Code Plagiarism Detection by Measuring Program Behavioral Similarity’, IEEE Access, vol. 9, pp. 50391–50412, 2021, doi: 10.1109/ACCESS.2021.3069367.

[29] T. M. Tashu, M. Lenz, and T. Horváth, ‘NCC: Neural concept compression for multilingual document recommendation’, Appl. Soft Comput., vol. 142, p. 110348, Jul. 2023, doi: 10.1016/j.asoc.2023.110348.

[30] H. Xie et al., ‘Improving K-means clustering with enhanced Firefly Algorithms’, Appl. Soft Comput., vol. 84, p. 105763, Nov. 2019, doi: 10.1016/j.asoc.2019.105763.

[31] C. Mageshkumar, S. Karthik, and V. P. Arunachalam, ‘Hybrid metaheuristic algorithm for improving the efficiency of data clustering’, Clust. Comput., vol. 22, no. S1, pp. 435–442, Jan. 2019, doi: 10.1007/s10586-018-2242-8.