JCBI
Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 05 Issue 01
2023

Research Article
https://doi.org/10.56979/501/2023

Optimized Feature Extraction and Cross-Lingual Text Reuse Detection using Ensemble Machine Learning Models

Muhammad Sajid Maqbool[1*], Israr Hanif[1], Sajid Iqbal[1] , Abdul Basit[1] and Aiman Shabbir[2]

[1]Department of Computer Science, Bahauddin Zakariya University, Multan, Pakistan.
[2]Department of Computer Science, Muhammad Nawaz Sharif University of Agriculture, Multan, Pakistan.
[*]Corresponding Author: Muhammad Sajid Maqbool. Email: sajidmaqbool7638@gmail.com

Received: December 12, 2022 **Accepted:** May 21, 2023 **Published:** June 05, 2023.

Abstract: With the availability of digital data in different languages, cross-lingual plagiarism detection has gained more importance. Cross-lingual plagiarism is difficult to detect because suspicious and source texts can be written in different languages and processing of digitized text in different languages presents varying types of challenges. In this work, we propose a cross-lingual plagiarism detection method using machine learning algorithms. In this work, we have created an ensemble of machine learning algorithms and to evaluate the designed methodology, a corpus focusing Urdu-English language pair titled CLPD-UE-19 is used. The corpus is a collection of 2398 documents where the source text is written in Urdu language and the suspicious text is presented in the English language. Using NLP methods, optimal features are extracted and fed to designed ensemble method for document classification. A number of aggregating techniques are employed which include majority voting, stacking, averaging, boosting, and bagging. Among these models, the stacking has performed the best achieving accuracy of 96 percent.

Keywords: Cross-lingual Plagiarism Detection, Urdu English Plagiarism Detection, Plagiarism Detection, Machine Learning, Ensemble machine learning methods.

1. Introduction

Plagiarism is the act of using another author's words or works without properly citing them. The use of computer systems for this kind of issue has grown in importance as the number of virtual documents on the Web has increased. Plagiarism detection is of particular significance. There are two varieties of plagiarism: (1) mono-lingual plagiarism, in which the original text and the copied material are both written in the same language, and (2) cross-linguistic plagiarism (CLP), in which the original text and copied text are both written in two distinct languages. Because it is so simple to translate content from one language into another using online translation tools, CLPD research is now being done instead of traditional plagiarism detection research. CLP is difficult to detect because the suspicious text and source can be written in different languages. In this regard, distinct algorithms were proposed to carry out the challenge of PD in textual content documents. We proposed an improved version of CLPD in which different ML algorithms are used to detect plagiarism by applying different Ensemble methods on language pair Corpus CLPD UE 19 [1]. CLPD UE 19 Corpus is a collection of 2398 documents in Urdu-English pairs. In the language pair corpus (CLPD-UE-19) the source text is written in the Urdu language and the suspicious text is presented in the English language. We use NLP techniques in python language to extract optimized features from the Corpus and create a dataset that understands by the ML tools to train the ML models. Our created dataset is in CSV format, and it includes source-specific attributes as well as suspected content such as Jaccard similarity, Cosine similarity, and LCS. To get the result, we used the N-gramme of the preprocessed text to Measure comparability between documents. We use the PyCharm tool to build Ensemble ML models from various classifiers. The ML Models have been trained by utilizing all features. Five Ensemble ML techniques [40,41] as Voting, Stacking, Average, Boosting, and Bagging are used to Measure the Accuracy of CLDP. The Voting and Stacking method have better Performed when the model is trained using all the features. According to the precision of the obfuscate level, the highest score is given by Class NP and the lowest score is given by class HR. In the Recall, NP has the best score, while LR has the lowest, and in the F-1 Score, NP has the best score, while HR has the least. This research work has following contributions

1. A data set from CLPD-UE 19 dataset is extracted and number of features computed from it to come up with required data for classification.
2. Ensemble based CLPD model is designed and evaluated on extracted dataset.
3. The proposed method produces better results than existing research [2,10].

Rest of the document consists of following parts: Section two presents the literature survey on plagiarism detection using different techniques and Section three introduces the methodology of the proposed work. The results of the proposed work are discussed in section four and finally, section fifth section presents the conclusion of the work.

2. Related Work

The Due to the huge rise in availability of online digital data, cross lingual plagiarism has become a norm and with every passing day, reproduction of information is increasing. Detection of cross lingual plagiarism is a challenging task. Scientists and technologists are focusing to solve the problem of CLPD [1,10,11] to ensure the availability of original data and information. In this section, we review some of the prominent works done in this field.

A cross-lingual text-reuse detection method for Hindi and English language pairings was put up by Basant Agarwal [2]. The researcher has built a model that translates the Hindi text into English using the Google translator before comparing it with the suspected text. In this study PAN-CLEF corpus (https://pan.webis.de) is used. The author has used cross language paraphrase detection method to determine the similarity. In this study, two similarity metrics are used (1-semantic similarity and 2-alignment Based similarity) to compare the documents. The created model gives accuracy of 92 percent.

A plagiarism detection system based on cross-lingual (Arabic-English) document pairs was introduced by Mokhtar Al-Suhaiqi et al. [4,5]. This work includes single language plagiarism detection as well as cross lingual plagiarism detection. Machine learning algorithms have been used for CLPD and proposed method consists of five-stages which include 1) pre-processing of documents, 2) keyword extraction, 3) candidate document retrieval, 4) monolingual PD, and 5) ML based cross lingual PD. Three machine learning models 1) Support Vector Machine (SVM), 2) Naïve Bayes and 3) Linear Logistic Regression are used to determine the precision and F-1 score of the system. Different features are extracted from the translated source and the suspicious text such as Longest Common Subsequence, N-gram Similarity, Finger Print based Jaccard similarity, Dice coefficient, and Finger Print Based Containment similarity. The dataset used in this research is custom built and consists of 314 documents of Arabic and English language. The best result is given by the SVM Classifier with F-1 Score and Presicion of 92% and 85% respectively.

A deep neural network-based text classification method is introduced by Salha Alzahrani and Hanan Aljuaid [6] for Arabic-English plagiarism detection. This study focused on achieving two research goals: 1-detecting the plagiarism and judgment and, 2-determining cross-lingual semantic text similarity (CL-STS) [6]. A large size custom developed dataset is used in this study which is collected from different sources and consists of 71910 document pairs of Arabic-English languages. This study extracts the semantic features from the CL textual pairs which include the semantic role labeling, topic similarity, bag of meanings, named entity similarity, bag of stop words, number of most frequent terms, spatial role labeling and the combination of these features. In this research work both classification and regression methods are used to generate predictions for the binary and multiclass dataset which contains different similarity features. They have addressed two classification tasks 1: binary and multi-classification. Classification model produced the best results of 91% of binary classification and 75% of multi-classification whereas accuracy score for regression model is 71%.

A cross-lingual plagiarism detection system for Russian-English documents was suggested by Bakhteev et al. [7]. The proposed method in this research contains three stages which includes 1) machine translation, 2) source retrieval, and document comparison. First of all, the source text (Russian) is translated into English by using Transformer which was presented by [8] and is an open source neural machine translator. At stage 3, the analysis of both (source and destination) documents is started. Secondly Shingle-Based Approach (N-Grams) is used to retrieve the documents and their IDs. Finally, the comparison of source translated text and English suspected text is done to determine the plagiarism. The results of this plagiarism detection are given as, the Precision is 83%, Recall is 79% and the F1-Score of 80 percent. A new plagiarism detection was proposed by Kensuke Baba [9] for Spanish-English language documents. In this study the document similarity is defined by distributed representation of words. The maximal and minimum value of Longest Common Subsequence (LCS) is used as the features to compare the similarities between document pairs. PAN 2013 corpus [10] is used for the evaluation of plagiarism detector, the corpus contains 5185 pairs of documents. The selected corpus contains 1000 documents pairs of having zero plagiarism and 4185 documents pairs containing some degree of plagiarism. The plagiarized document pairs are divided among four classes 1) No obfuscation, 2) Random Obfuscation, 3) Translation and 4) Summarization. The scores reported in this study are 89.62% accuracy, 97% Recall, 92.08% Precision and 90% F1-score.

A Crosslingual textual similarity method is proposed by Yigit Sever and Gonenc Ercan [15] for seven different languages (Albanian, Bulgarian, English, Greek, Italian, Romanian and Slovene) by using word embedding technique. The used words are collected from Open Multilingual Wordnet (OMW) [13,14] definitions and used to compare the semantic textual similarity between the wordnet glosses of synonyms groups in the 7 different languages. The wordnets used in this study contains 4681 definitions from Albanian, 4959 definitions from Bulgarian, 117,659 definitions from English, 18,136 definitions from Greek, 12,688 definitions from Italian, 58,754 definitions from Romanian, and 3144 definitions from Slovene. In this study a supervised deep learning model (Siamese) [11,12] is compared with three unsupervised learning methods for text similarity (1-Wasserstein distance (WS), 2-Sinkhorn distance (SD) and 3-cosine similarity (CS)). The experiments of this study show that supervised learning give the average weightage of seven languages is 19.07 percent matching score and unsupervised learning is better performed as compare to the supervised learning (Wasserstein distance, Sinkhorn distance and cosine similarity) with matching score of (38.33 percent, 46.49 percent, and 45.63 percent). A candidate retrieval approach for plagiarism detection is used for German-English and Spanish-English by Meysam Roostaee et al. [15]. This method consists of two steps. In first step, a cross-lingual candidate retrieval model that reduced the number of documents which is used in the next step, and the second step is to compare with the state-of-the-art cross lingual plagiarism detection. In this study English, German and Spanish Wikipedia dump files since January 2018 are collected. Each record in the dump files contains the features such as id, title, and content of the article. The source text is translated by using the google translator and the preprocessing of the text is done by using the Stanford CoreNLP Toolkit [22]. After the preprocessing the T+MA-BOW [26] retrieval model is used to compare the textual similarity between the source and suspicious documents. Three datasets are used to measure the performance of the proposed model. First dataset contains the PAN-PC-12, second dataset is collected from three different sources (RC-Acquis [29], PAN-PC-11 and Wikipedia) and the last one is ClueWeb09 [26] corpus. The experiment results of plagiarism detection for second dataset are best one in this study. The scores are presicion 90%, Recall of 88%, F1 of 70%, and F2 of 76%.

A text alignment-based approach for plagiarism detection is used by Meysam Roostaee et al. in [16] for German English and Spanish English language pairs. Their proposed model is based on two stages of similarity matching approach for syntactic-semantic similarity to identify the plagiarized portion. A multilingual word embedding based dictionary and a vector space model is used with local weighting technique to find the candidate fragments from the source and the target text document. At the second stage, sentence matching is performed on the fragments that identified in the first stage and performed the similarity on the words and their relationship by presenting the text as graph of words. Three standard benchmarks corpuses (PAN-PC-11 [17,18] and PAN-PC-12 12 [19] and SemEval 2017 [27] are used in this study to train and evaluate the proposed model. Their proposed model well performed on the PAN-PC-12 corpus with accuracy of 66.33% for the German-English and 70.80% for the Spanish-English. The Recall is 92.28%, Presicion is 51.77 and Granularity is 100% on the German-English and 97.46%, 55.60% and 100% on the Spanish-English.

In the reviewed work, it can be observed that majority of researchers have employed conventional NLP techniques and classification algorithms for detection of plagiarism, which undoubtedly have limitations in terms of performance and addressing of varying aspects of the task. A good number of researches have used different machine learning and deep learning-based algorithms. This shows the usability and popularity of latest classification methods for CLPD. A variety of similarity detection methods exists in literature such as Semantical, Alignment Based, Vector Based, Syntactic, Fragment level, Conceptual Based, Length Based, Jaccard Similarity, Longest Common Subsequence, Cosine similarity, Key Words Based and Sentence Based. This variation sometimes makes it difficult to conduct a comparative study. Correctly annotation of curpus increase the accuracy of models by evaluating new annotated dataset. Optimized selection of features and Cleaning of dataset performed well on the machine learning models. Selection of ensemble machine learning models give more accurate results as compare to the simple models. Precision, Recall and F-1 score of each class is increased in the new annotated dataset. Stacking and Voting classifier are performed but better as compare to the other ensemble classifiers. Precision of the obfuscate level, highest score is given by the Class NP and lowest score is given by class HR. In the Recall, NP has the best score, while LR has the lowest F-1 Score, NP has the best score, while HR has the least. Our proposed methodology gives better result as compare to [6, 7, 25]. The presicion of these studies is less than our proposed model. Our proposed model achieved presicion of 95 percent that higher than the mentioned studies. Our proposed majority voting classifier achieved accuracy of 96 percent.

Table 1. Different Research Works on Plagiarism Detection

Ref.	Technique	Algorithm	Language	Dataset	Similarity metrics	Results
Proposed Work (2022)	Machine Learning	Voting, Stacking, Bagging, Average, and Boosting	Urdu-English	CLDP-UE-19, 2198 document pairs	CS, JS and LCS	Accuracy=96 % Presicion=95% Recall=94% F-1 Score=94%
Formal work (2020)	Machine Learning	RFC, DTC, NN, KNN	Urdu-English	CLDP-UE-19, 2198 document pairs	CS, JS, and LCS	Accuracy=77% Precision=72% Recall=60% F-1 Score=57%
Basant Agarwal (2019)	Machine learning	Semantic space	Hindi-English	PAN- CLEF	Semantical similarity, alignment-based similarity	92 %
Mokhtar Al-Suhaiqi et al. (2018)	Machine learning	Linear logistic regression, naïve Bayes, SVM,	Arabic-English	Real dataset that contain 314 documents	N-Grams Similarity, LCS, DC, Fingerprint based JS, and Fingerprint based CS	F1- Score=92% Presicion=85%
Salha Alzahrani and Hanan Aljuaid (2020)	Deep learning, NLP	deep neural networks	Arabic-English	handmade data which contain 71910 pairs	Semantic text similarity	71% = regressor 80%=Classification
Bakhteev et al. (2019)	Deep learning	Transformer	Russian-English	OPTUS dataset, 30 million sentences	Semantic similarity	Precision = 83%, Recall = 79% and F1 = 80%
Kensuke Baba (2017)	Deep learning, NLP	A smith-Waterman algorithm, word2vector	Spanish-English	PAN 2013 5185 pairs of documents	length of (LCS), min (LCS) and max (LCS)	Accuracy=89.62%, Recall=97%, Presicion=92.08%, F1-Score=90%.
Yigit Sever and Gonenc Ercan (2020)	Deep learning	A Siamese deep learning model	Spanish-English	Wordnet dataset	Wackerstein distance, Sinkhorn distance	WD=38.33% SD =46.49%, CS =45.63%
Meysam Roostaee et al. (2020)	Machine learning	CL-ASA, CL-CTS, CL-BOC, T+MA-BOW	German English & Spanish English	PAN-PC 11 & PAN-PC 12 and ClueWeb09	Conceptual-based and keyword based	Presicion =90% Recall=88% F1=70% F2=76%

Meysam Roostaee et al. (2020)	Machine learning	multilingual word embeddings (MWEs) and Babel Net	German English & Spanish English	PAN-PC 11 & PAN-PC 12 and SemEval2017	fragment-level and sentence-level	Accuracy=66% Recall=97.46%, Presicion=55.60%

3. Proposed Framework

We present an approach for identifying CLP for the UE language pair in this section. A little work can be found in literature for UE CLP. The dataset used in this study is extracted from CLPD-UE-19 [1] corpus and was developed using the methodology depicted in (Figure 1).

3.1. Feature Extraction Methodology

The steps for the creation of the dataset are illustrated in (Figure 1). Thorough analysis of dataset is required in order to optimize the performance of selected classification models. The dataset extracted from CLPD-UE-19 is improved first then high value features are identified and filtered using PCA which are then employed for training of classifiers. The extracted dataset is divided into training and validation (also called test) sets. The CLPD-UE contains document in the form of pairs where query or source text is given in Urdu language and suspicious or target text is provided in English language. A number of features about source text are provided in CLPD-UE like 'text domain' which describes the field from which document is prepared like science or management sciences, and size of the document which could be small (<50 words), large (50-100 words) or medium (>100 words). This corpus is given in XML format.

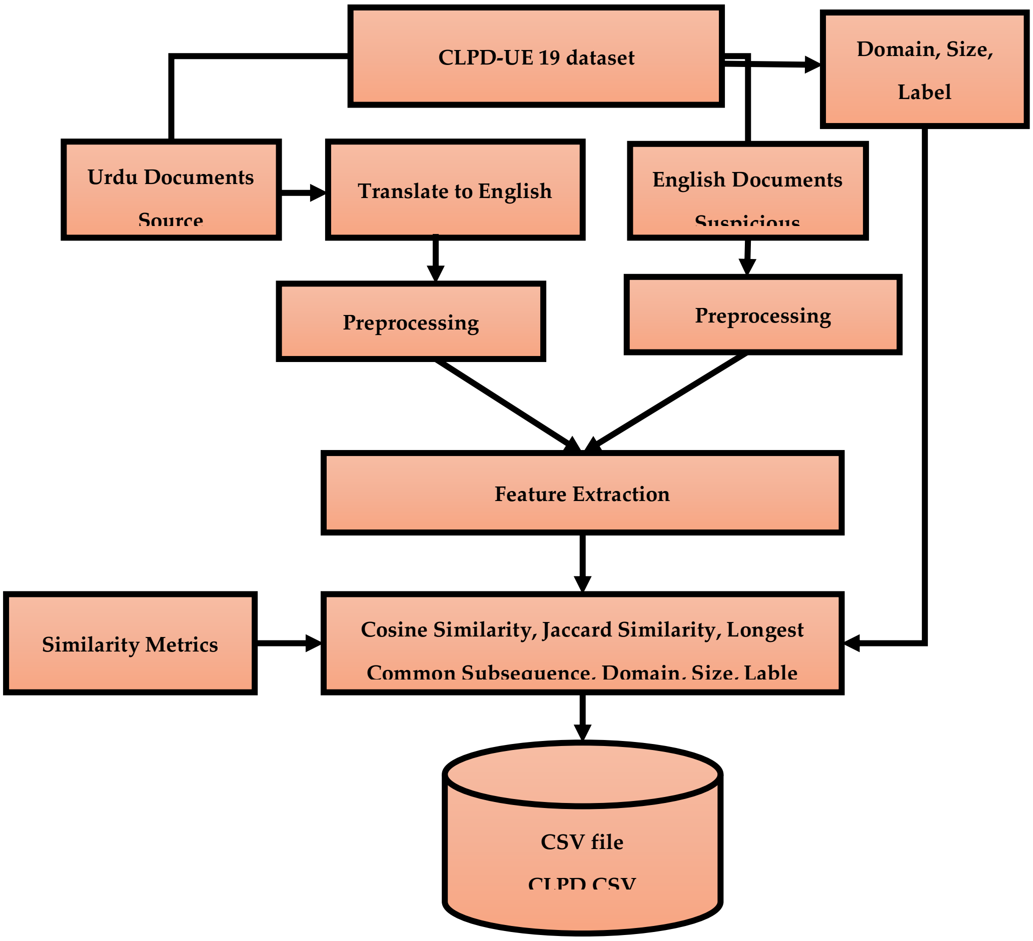

Figure 1. Feature Extraction Methodology

There are 2395 documents about different topics, including 540 automatic translations (generated through Google translate), 539 artificially paraphrased texts (by using paraphrasing tools), 508 documents written by human experts, and 808 texts free of plagiarism [1]. Four level of plagiarism are defined in this corpus that include near copy (NC) having around 100% plagiarism, heavy revision (HR), the documents having plagiarism more than 68 percent, light revision (LR) having plagiarism of 52%, and 22% for non-plagiarized (NP) documents. Here is an example taken from the CLPD-UE corpus in XML format (Table 2).

Table 1. Example of XML file

```
<features domain="gt" size="small" obfuscation="APC">
     <source> عالمی حرارت سے مراد دنیا اور بحار کے درجہ حرارت میں انیسویں صدی سے اضافہ ہے جو آج تک جاری
     ہے-</source>
     <suspicious> Global warming refers to the rise in temperature of the planet and oceans of the
     nineteenth century that continues nowadays. </suspicious>
</features>
```

This xml data is then converted to a CSV file to use it as input for the classifiers. Following (Table3) shows the sample from csv file.

Table 2. CLPD-UE-19 CSV file sample

Source Text (translated from Urdu)	Suspicious Text	Domain	Size	Class
computers computer electronic device question complex statistical issue set program instructions easily solve display result calculations take safe car make calculations enumerator estimate count habit calculator generally speak write computer conflict say	computer computer associate degree device apply math calculations complicate downside question per directions give therefore program well resolve result calculations show either offer take shop automobile they calculate count calculate compute calculator generally speak write pc write speak	Computer Science	Medium	LR

3.2 Preprocessing

To produce clean text, few transformations as preprocessing are applied:

Tokenization: Base on hard spaces, the source and suspicious texts are tokenized.

Stop word and punctuations removal: In this phase, English stop words and punctuations are removed.

Stemming: To covert tokens to their root forms, text stemming is performed using porter stemmer. This phase removed all suffixes and prefixes.

Following (Table 4) lists the processing steps of given text.

Table 3. Preprocessing of Text data

Urdu Source Text مادری زبان کے طور پر دنیا کی سب سے بڑی زبان جدید چینی ہے جسے 70 کروڑ افراد بولتے ہیں، اس کے بعد انگریزی ہے جو اکثر لوگ ثانوی یا رابطے کی زبان کے طور پر بولتے ہیں جس کی بدولت دنیا بھر میں انگریزی بولنے والے افراد کی تعداد ایک ارب سے زیادہ ہوگئی ہے-
English Suspicious Text 70 million people are speaking Chinese as their mother language, larger in its category and then English most spoken language as a second language has more than 1 billion people all over the world.

Translated Source Text

The largest language in the world as a mother tongue is modern Chinese, which is speaking 70 million people, followed by English, which is often spoken as a secondary or contact language that makes people of English speaking worldwide. The number has increased to over one billion.

Tokens of Translated Source Text

'The', 'largest', 'language', 'in', 'the', 'world', 'as', 'a', 'mother', 'tongue', 'is', 'modern', 'Chinese,', 'which', 'is', 'speaking', '70', 'million', 'people,', 'followed', 'by', 'English,', 'which', 'often', 'speak', 'as', 'a', 'secondary', 'or', 'contact', 'language', 'that', 'makes', 'people', 'of', 'English', 'speaking', 'worldwide. The', 'number', 'has', 'increased', 'to', 'over', 'one', 'billion.'

Tokens after removal of stop words

'largest', 'language', 'world', 'mother', 'tongue', 'modern', 'Chinese,', 'speaking', '70', 'million', 'people,', 'followed', 'English,', 'often', 'speak', 'secondary', 'contact', 'language', 'makes', 'people', 'English', 'speaking', 'worldwide', 'number', 'increased', 'one', 'billion'

Tokens after removal of punctuations

'largest', 'language', 'world', 'mother', 'tongue', 'modern', 'chinese', 'speak', '70', 'million', 'people', 'follow', 'english', 'often', 'speak', 'secondary', 'contact', 'language', 'make', 'people', 'english', 'speak', 'worldwide', 'number', 'increase', 'one', 'billion'

Stemmed Text

largest language world mother tongue modern Chinese speak 70 million people follow English often speak secondary contact language make people English speak worldwide number increase one billion

3.3 Feature extraction

Following is the set of hand-crafted features extracted from processed text in addition to features provided in CLPD-UE:

Jaccard Similarity: This measure is calculated for unigram and trigram words. In this measure, the number of common words in both documents are divided by the number of total terms in both documents.

$$\boldsymbol{J(A,B)} = \frac{|\boldsymbol{A}\cap\boldsymbol{B}|}{|\boldsymbol{A}\cup\boldsymbol{B}|} \tag{1}$$

Cosine Similarity: For cosine similarity, our source and suspicious documents are converted into vectors using $\boldsymbol{tf - idf}$ function. The dot product of both vectors is divided by the magnitude of both vectors. This is also calculated for unigram and trigram words.

$$\boldsymbol{cos\theta} = \frac{\overrightarrow{A.B}}{\|\vec{A}\|.\|\vec{B}\|} \tag{2}$$

Longest Common Subsequence (LCS): LCS is used to obtain similarity values between two documents. In which the length of the common sub-string is divided by the document that has the minimum size measured in characters.

$$c[\boldsymbol{i},\boldsymbol{j}] = \begin{cases} \boldsymbol{0} & \boldsymbol{if\ i = 0\ or\ j = 0} \\ \boldsymbol{c[i-1,j-1]+1} & \boldsymbol{if\ j,j > 0\ and\ x_i = y_i} \\ \mathbf{max}(\boldsymbol{c[i,j-1]},\boldsymbol{c[i-1,j]}) & \boldsymbol{if\ i,j > 0\ and\ x_i \neq y_i} \end{cases} \tag{3}$$

We create another file containing the 8 features for our dataset. These features include Jaccard Score for unigram and trigram (JS_1 and JS_3), Cosine similarity for unigram (CS_1) and for trigram (CS_3), longest common subsequence (LCS), domain of document, size of document and its similarity measure with source document i.e. Label. (Table 5)contains some examples from this file.

Table 4. Extracted Features

index_doc	$\mathbf{JS_1}$	$\mathbf{CS_1}$	LCS	$\mathbf{CS_3}$	$\mathbf{JS_3}$	Domain	Size	Label
Dacument-0001 (2).xml	0.19	0.29	0.64	0.00	0.00	CS	Medium	LR
Dacument-0001 (3).xml	0.24	0.38	0.63	0.00	0.00	MS	Medium	LR
Dacument-0001 (4).xml	0.50	0.53	0.73	0.07	0.07	CS	Medium	NC

Dacument-0001 (5).xml	0.49	0.66	0.72	0.01	0.01	MS	Medium	NC
Dacument-0001 (6).xml	0.16	0.54	0.51	0.00	0.00	CS	Medium	NP
Dacument-0001 (7).xml	0.14	0.21	0.55	0.00	0.00	GT	Medium	NP
Dacument-0001 (8).xml	0.12	0.34	0.51	0.00	0.00	CS	Medium	NP

The dataset extracted from CLPD-UE 19 is improved by removing inconsistencies found among data annotation. The features which are given in nominal form are then converted to numeric form in order to use them in machine learning algorithms. These features include domain, size and document labels. After applying these preprocessing steps, data features frequency distribution is visualized in figure-2.

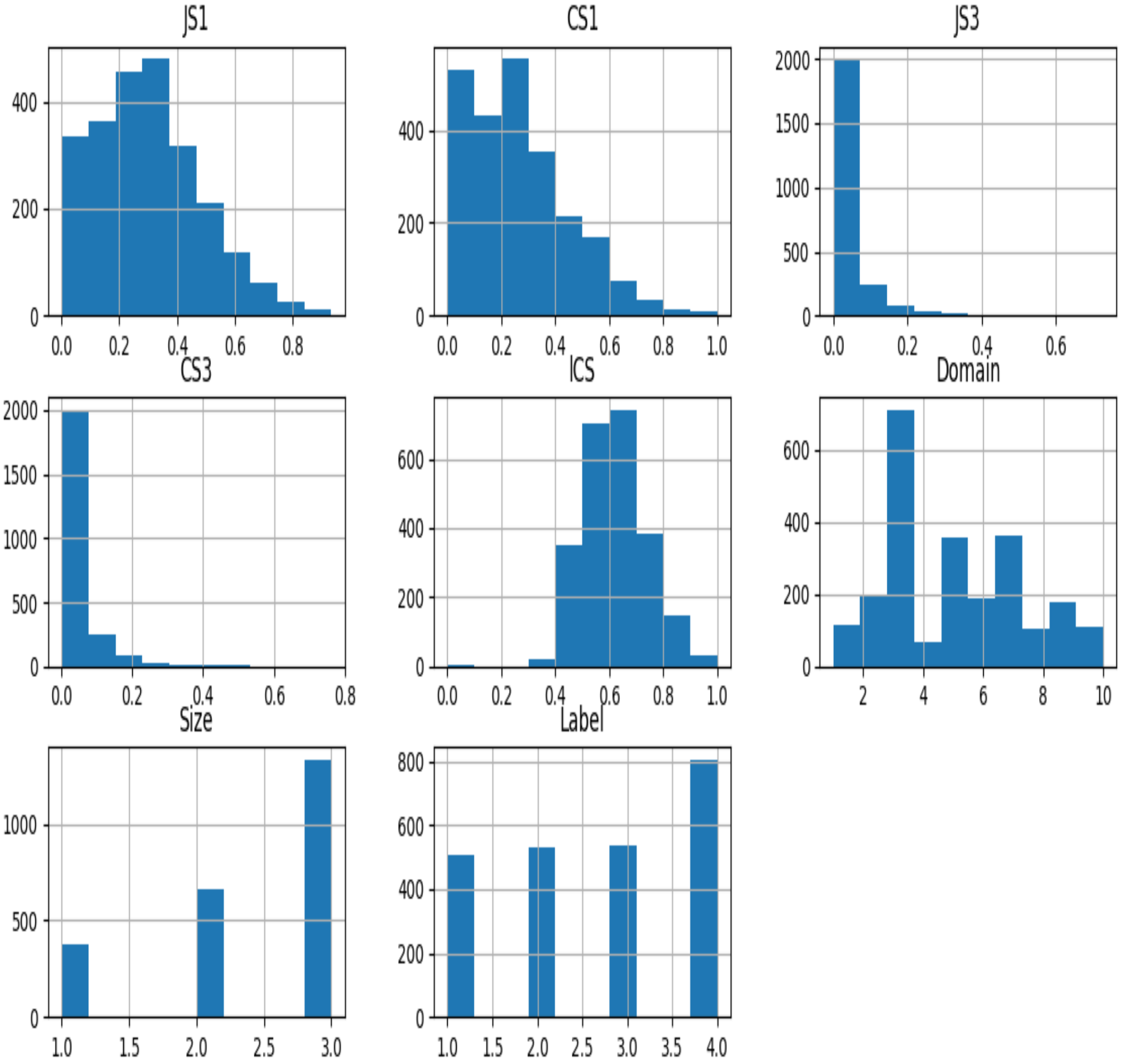

Figure 2. Data Features Frequency Distribution

3.4 Performance Evaluation Framework

(Figure 3) explain the Framework how to use the CSV file to evaluate the performance of the CLPD system and explain each step. we innovatively introduced ensemble meta technique on the dataset to explore their impact on performance of the classifiers. All these steps resulted in improvement of accuracy ratio in certain ML models.

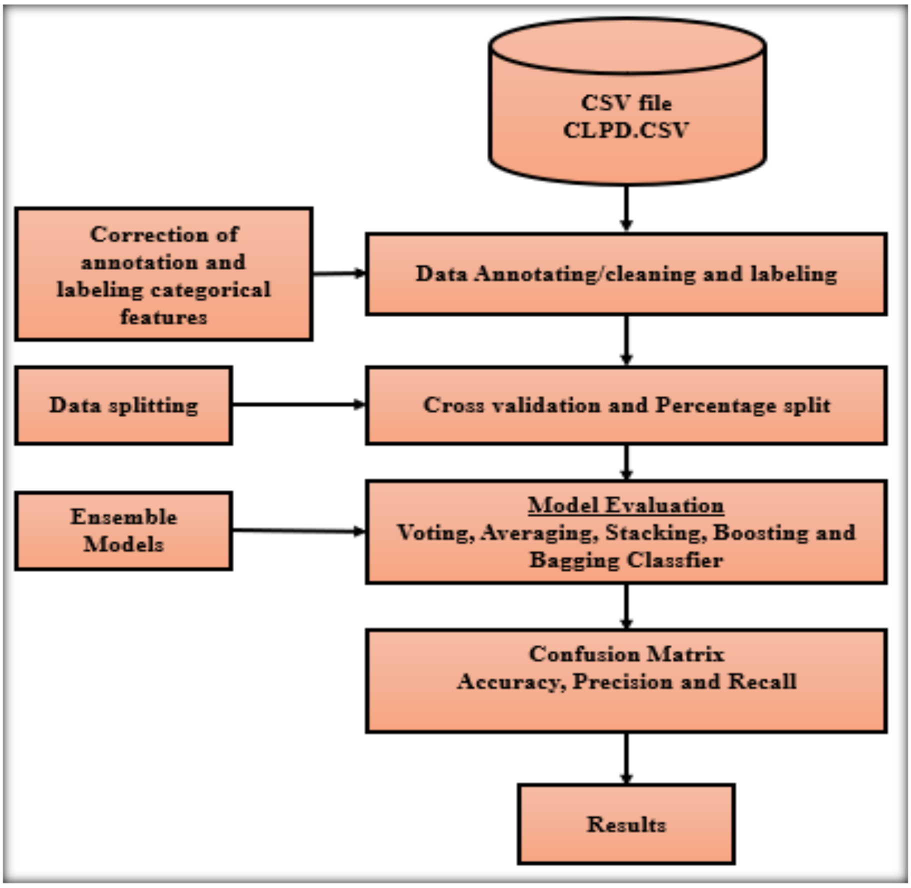
CSV file
CLPD.CSV
Correction of annotation and labeling categorical features
Data Annotating/cleaning and labeling
Data splitting
Cross validation and Percentage split
Ensemble Models
Model Evaluation
Voting, Averaging, Stacking, Boosting and Bagging Classfier
Confusion Matrix
Accuracy, Precision and Recall
Results

Figure 3. Performance Evaluation Framework

3.5 Classification models design and experimental setup

We have used three machine learning models to create ensemble models for document classification. These base models are decision tree, random forest, and Naïve Bayes. Following (Figures 5) shows the general view of ensemble models.

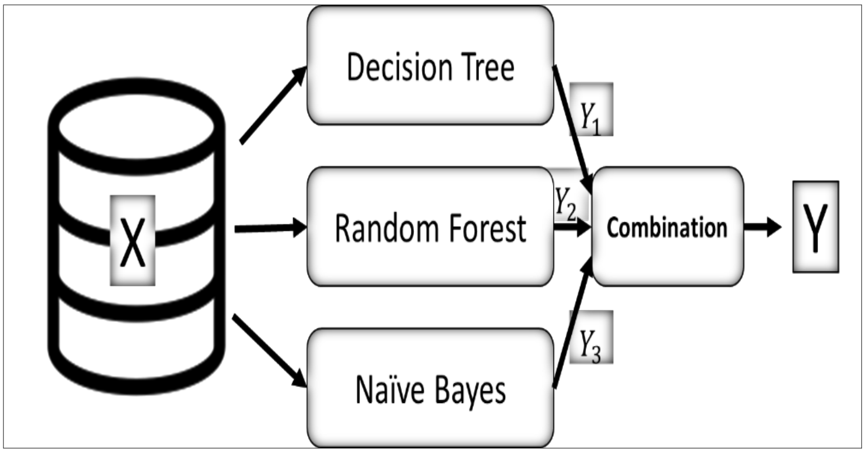

Figure 4. General Ensemble model architecture

In this general model, predictions (Y_1, Y_2, Y_3) made by each base model are combined using a combination method that include majority voting, averaging, stacking, boosting, and bagging. X input is provided to each classifier which generates its prediction as Y_i .

$$\boldsymbol{Y_1 = DT(X)}$$

$$\boldsymbol{Y_2 = RF(X)} \tag{4}$$

$$\boldsymbol{Y_3 = NB(X)}$$

Where DT refers to Decision Tree, RF for random forest and NB for Naïve Bayes.

Configuration of base models are as given below:

Decision Tree Classifier: Maximum depth of DT is kept as 6 with minimum split at each internal node as 2.

Random Forest Classifier: Number of estimators are taken 1000 with Gini Impurity as measure for feature quality. The depth of each tree is taken as maximum as possible. Minimum split of each internal node is kept as 2.

Naïve Bayes: We have used multi-class Naïve Bayes classifier to classify data among different classes.

In majority voting, a prediction made by maximum number of classifiers is chosen as final prediction of ensemble. In case each model predicts a different output, prediction made by random forest is chosen. Research has shown that random forest comparatively performs better than other two classifiers (Esmaily et al., 2018; Bin Chen et al., 2012).

$$Y_{ensemble} = \begin{cases} \text{count}_{unique}(Y_1, Y_2, Y_3) & if(\exists\ count_{unique} > 1) \\ Y_2 & if\ each\ count_{unique} = 1 \end{cases} \tag{5}$$

For second ensemble model, predictions by each model (which are in numeric form) are averaged which are then rounded to nearest integer to predict the exact category of paper.

$$Y_{ensemble} = round(average(Y_1, Y_2, Y_3)) \tag{6}$$

In stacking based ensemble model, the predictions of each base model are stacked and then a meta-classifier is applied on these predictions to create ensemble final prediction. This meta-classifier could be chosen arbitrarily and, in our study, we have used logistic regression. Following equation describes the process:

$$Y_{ensemble} = logistic\ regression\ (Y_1, Y_2, Y_3) \tag{7}$$

Boosting method is comparatively complex method. In this model, output of first base model is given as input to next model, output of second model is given as input to next model. Output of final model is considered as prediction of ensemble model. The configuration of boosting method for our study is as follows:

$$Y_{ensemble} = NB\left(DT\big(RF(X)\big)\right) \tag{8}$$

A bagging is an ensemble meta-estimator that trains base classifiers using random subsets taken from given dataset. The selection of subsets is performed with replacement. The predictions made by each classifier are then aggregated and either majority voting or averaging is performed on it. In our configuration, we have used majority voting mechanism.

$$Y_1 = DT(X_1), \qquad Y_2 = RF(X_2), \qquad Y_3 = NB(X_3) \tag{9}$$

$$Y_{ensemble} = \begin{cases} \text{count}_{unique}(Y_1, Y_2, Y_3) & if(\exists\ count_{unique} > 1) \\ Y_2 & if\ each\ count_{unique} = 1 \end{cases} \tag{10}$$

The data set is divided into train and test sets with 80:20 ratio respectively. Each ensemble is separately trained on trained dataset. We performed 10-fold cross validation with each model and average performance is reported.

4. Experimental Results

Confusion matrix is popular method to measure the performance of classification models. A confusion matrix consists of four statistics as shown below:

		Actual Class →	
		Class A	Class B

Predicted	Class A	**TP**	**FP**
Class ↓	Class B	**FN**	**TN**

Considering the statistics from class A perspective, TP is the number of instances whose actual class is A in our dataset and the classification models has also predicted the same class. FP is the count of instances whose actual class is B but our classification model has predicted them as class A which is an error and known as type-1 error. Similarly, FN is the statistic which counts the number of input instances which are actually in class A but erroneously our classifier has classified them in class B and it is known as type-2 error. Finally, those input data instances which are neither in class A nor the classifier has predicted them in class A. Hence our classifier has truly filtered out unwanted input instances.

As our task is multi-classification task so for the ease of reader, we describe the multiclass confusion matrix here too.

		Actual Labels →			
		NC	HR	LR	NP
Predicted Class ↓	NC	TP	FP	FP	FP
	HR	FN	TN	TN	TN
	LR	FN	TN	TN	TN
	NP	FN	TN	TN	TN

Here again, we will take NC class as reference class. TP are the input documents which are classified as NC and actually they have the label NC. The documents having actual label NC and are classified in any other class are counted as FN. FP is the count of documents which are not in class NC but our classifier has predicted them in NC class. Finally, all those documents whose label is not NC and they are not predicted in NC. Similarly, confusion matrix for other calluses can be computed.

Following metrics are used to evaluate the ensemble models.

Accuracy: we use the following formula to calculate the accuracy. This is simple performance metric that tells that how many instances are correctly predicted. It is given by

$$Accuracy = \frac{TP+FP}{TP+FP+TN+FN} \quad (11)$$

Precision: we used This statistic to describes that how much the predictions are near to each other. In other words, smallest the standard deviation, higher the precision.

$$Precision = \frac{TP}{TP + FP} \quad (12)$$

Recall: Recall is used to find the true positive rate or sensitivity. Recall looks at the number of false negatives that included in confusion matrix. Recall is a metric that quantifies the number of correct positive predictions made out of all positive predictions that could have been made.

$$Recall = \frac{TP}{TP+FN} \quad (13)$$

F1-Score: The F1 score of the models helps to measure precision and recall at the same time.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (14)$$

The accuracies of the five ensemble models are listed in (Table 6). For cross-validation and training testing split, respectively, each model has two different types of accuracy.

Table 5. Accuracy Models

Model	**Cross Validation Accuracy (%)**	**Testing Accuracy (%)**
Majority Voting	95	95.3
Stacking	96	95.3
Average	91	94.3

Boosting	95	95.2
Bagging	94.7	94.74

With a 98 percent accuracy rate on training and testing data, the model Voting Classifier has the highest accuracy, while the model Bagging Classifier has the lowest accuracy at 94.74 percent. In the graph above, two metrics—Cross-Validation and Training Testing—are used to compare the accuracy of five models. The model Average Classifier has the lowest accuracy of Cross-Validation with an accuracy of 91 percent, while the Boosting Classifier and Stacking Classifier have the highest accuracy of 96 percent. The accuracy values for each class are represented in Figure-4 graph, which demonstrates that the Voting model calculates with the highest score while the Stacking model calculates with the lowest score for each class. In every model, the class NP has high performance, while the class HR has poor performance because the number of instances in the NP class is higher than the number of instances in the other (NC, LR, HR) classes.

(Table 7) shows the class wise precision score for each model. The majority voting method performs best for class NP, where it is 99 percent, and poorly in class HR. With an accuracy of 73 percent, the Stacking method provides the highest precision for the class LR, while providing low precision for the class NP with score of 45 percent. Class NC has low precision and an accuracy score of 88 percent according to the Average method, whereas class NP has the highest with a precision score of 96 percent. The highest Precision is provided by Boosting method at class NP, where accuracy is 99 percent, while the lowest Precision is generated for class HR, which is 91 percent. Bagging method performs well at class NP, where accuracy is 99 percent, and poorly at class HR, where accuracy is 85%.

Table 7. Precision of Models

Models	Class HR (0)	Class LR (1)	Class NC (2)	Class NP (3)
Majority Voting	92	97	97	99
Stacking	72	73	50	45
Average	90	91	88	96
Boosting	91	96	96	99
Bagging	85	93	95	99

(Table 8) depicts the recall for each class and each classifier. Class NP has the highest recall of the voting method with an accuracy of 99 percent, and class HR has the lowest recall with an accuracy of 93 percent. The stacking method has an accuracy of 100% and the highest recall for class NC while having a recall accuracy of just 19% for class LR. Class NP receives the highest accuracy from the average method, with a recall score of 97 percent, while class HR receives a low recall score of 82 percent. The recall of the boosting method is 99 percent at class LR, where it provides the best recall, and 93 percent at class HR, where it provides the lowest recall. Bagging method performs well at class NP with 93 percent accuracy and poorly at class HR with 88 percent accuracy. The graph in Figure-6 displays the values of Recall for each class and demonstrates that the Majority Voting Classifier, with accuracy of 93%, 96%, 96%, and 99 percent, calculates the highest Recall. The lowest Recall is produced by the Stacking method on every Class. In every model, the class NC has high recall, while the class HR has low recall.

Table 8. Recall of Models

Models	Class HR (0)	Class LR (1)	Class NC (2)	Class NP (3)
Voting	93	96	96	99
Stacking	26	19	100	99

Average	82	93	94	97
Boosting	93	99	88	97
Bagging	88	91	93	93

In (Table 9), each class is listed along with the f-1 Score of each method. Class NP has the highest F-1 Score of the voting method, with an accuracy of 99 percent, and the class HR has the lowest F-1 Score, with an accuracy of 92 percent. The class NC receives the greatest. F-1 Score from the stacking method, with an accuracy of 66 percent, whereas the class LR receives a low F-1 Score, with an accuracy of 30 percent. Class NP has the highest accuracy provided by the average method, with an F-1 score of 97 percent, whereas class HR has a low F-1 score and an accuracy score of 86 percent.

Table 9. F-1 Score of Models

Models	Class HR (0)	Class LR (1)	Class NC (2)	Class NP (3)
Voting	92	97	97	99
Stacking	39	30	66	61
Average	86	92	91	97
Boosting	92	97	92	98
Bagging	86	92	94	97

Analyzing the results of all methods and observing the confusion matrix, we get the following outcomes from our results.

Correctly annotation of curpus increase the accuracy of models by evaluating new annotated dataset. Optimized selection of features and Cleaning of dataset performed well on the machine learning models. Selection of ensemble machine learning models give more accurate results as compare to the simple models. Precision, Recall and F-1 score of each class is increased in the new annotated dataset. Stacking and Voting classifier are performed but better as compare to the other ensemble classifiers. Precision of the obfuscate level, highest score is given by the Class NP and lowest score is given by class HR. In the Recall, NP has the best score, while LR has the lowest F-1 Score, NP has the best score, while HR has the least.

5. Conclusion

Nowadays plagiarism has crossed the geographical and language barriers. Hence the research work in plagiarism detection is expanded to the cross-lingual domain. In this domain, the text of one language can be easily transformed into another language by using online translation tools and then reuse to commit CLP. Cross-Lingual Plagiarism is difficult to detect because the suspicious text and source can be written in different languages. In this regard, we have presented for CLPD to address this challenge using Urdu English (UE) language pair. For PD task Urdu English language pair Corpus CLPD UE 19 is used and evaluated the performance of PD. Five Ensemble ML techniques Voting, Stacking, Average, Boosting, and Bagging are used to measure the accuracy of CLDP. We have found that Voting and Stacking methods have better performance for the text document classification.

In future research, this topic can be expanded by utilizing several translators from Urdu to English, with various similarity criteria applied to each pair. The CLPD-UE-19 [1] corpus can be used to generate a multilingual dictionary. The corpus can be expanded or introduced to more classes for finetuning in the CLDP domain.

Disclosure: This work is available in Research Square as a preprint article; it offers immediate access but has not been peer-reviewed [45].

Data availability: The dataset created in this study is available on demand.

Competing interests: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Agarwal, B. (2019). Cross-lingual plagiarism detection techniques for English-Hindi language pairs. Journal of Discrete Mathematical Sciences and Cryptography, 22(4), 679-686.
2. Al-suhaiqi11, M., Hazaa22, M. A., & Albared33, M. (2018). Arabic English Cross-Lingual Plagiarism Detection Based on Keyphrases Extraction, 2 Monolingual and Machine Learning Approach 3.
3. Alzahrani, S., & Aljuaid, H. (2020). Identifying cross-lingual plagiarism using rich semantic features and deep neural networks: A study on Arabic-English plagiarism cases. Journal of King Saud University-Computer and Information Sciences.
4. Arabi, H., & Akbari, M. (2022). Improving plagiarism detection in text document using hybrid weighted similarity. Expert Systems with Applications, 207, 118034.
5. Ataman, D., Camargo de Souza, J. G., Turchi, M., & Negri, M. (2016). FBK HLT-MT at SemEval-2016 Task 1: Cross-lingual semantic similarity measurement using quality estimation features and compositional bilingual word embeddings. In 10th International Workshop on Semantic Evaluation, SemEval@ NAACL-HLT (pp. 570-576). The Association for Computer Linguistics.
6. Baba, K., Nakatoh, T., & Minami, T. (2017). Plagiarism detection using document similarity based on distributed representation. Procedia computer science, 111, 382-387.
7. Bakhteev, O., Ogaltsov, A., Khazov, A., Safin, K., & Kuznetsova, R. (2019, September). CrossLang: the system of cross-lingual plagiarism detection. In Workshop on Document Intelligence at NeurIPS 2019.
8. Bond, F., & Foster, R. (2013, August). Linking and extending an open multilingual wordnet. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1352-1362).
9. Chicco, D. (2021). Siamese neural networks: An overview. Artificial Neural Networks, 73-94.
10. Clarke, C. L. (2010). ClueWeb09 and TREC Diversity. In NTCIR (p. 13).
11. Chen, Bin, Robert P. Sheridan, Viktor Hornak, and Johannes H. Voigt. "Comparison of random forest and Pipeline Pilot Naive Bayes in prospective QSAR predictions." Journal of chemical information and modeling 52, no. 3 (2012): 792-803.
12. Da Costa, L. M., & Bond, F. (2015, July). Omwedit-the integrated open multilingual wordnet editing system. In Proceedings of ACL-IJCNLP 2015 System Demonstrations (pp. 73-78).
13. Di Vito, S. (2007). Les ressources en français pour la linguistique de corpus.
14. E. Gharavi, K. Bijari, K. Zahirnia, and H. Veisi, “A deep learning approach to Persian plagiarism detection,” CEUR Workshop Proc., vol. 1737, pp. 154–159, 2016.
15. E. M. Hambi and F. Benabbou, "A Multi-Level Plagiarism Detection System Based on Deep Learning Algorithms" IJCSNS International Journal of Computer Science and Network Security, VOL.19 No.10, October 2019.
16. Esmaily, Habibollah, Maryam Tayefi, Hassan Doosti, Majid Ghayour-Mobarhan, Hossein Nezami, and Alireza Amirabadizadeh. "A comparison between decision tree and random forest in determining the risk factors associated with type 2 diabetes." Journal of research in health sciences 18, no. 2 (2018): 412.
17. F Jérémy, L. Besacier, Ferrero, Laurent Besacier, Didier Schwab, and Frédéric Agnes “Deep Investigation of Cross-Language Plagiarism Detection Methods,” pp. 6–15, 2017.
18. Franco-Salvador, M., Gupta, P., & Rosso, P. (2013, March). Cross-language plagiarism detection using a multilingual semantic network. In European Conference on Information Retrieval (pp. 710-713). Springer, Berlin, Heidelberg.
19. Haneef, I., Adeel Nawab, R. M., Munir, E. U., & Bajwa, I. S. (2019). Design and development of a large cross-lingual plagiarism corpus for Urdu-English language pair. Scientific Programming, 2019.
20. Hanif, R. M. A. Nawab, A. Arbab, H. Jamshed, S. Riaz, and E. U. Munir, “Cross-language Urdu-English (CLUE) text alignment corpus,” CEUR Workshop Proc., vol. 1391, 2015.
21. H. A. Bouarara, A. Rahmani, R. M. Hamou, and A. Amine, “Machine learning tool and meta-heuristic based on genetic algorithms for plagiarism detection over mail service,” 2014 IEEE/ACIS 13th Int. Conf. Comput. Inf. Sci. ICIS 2014 - Proc., pp. 157–162, 2014.
22. Ikae, C., Nath, S., & Savoy, J. (2019). UniNE at PAN-CLEF 2019: Bots and Gender Task. In CLEF (Working Notes).
23. Koch, G., Zemel, R., & Salakhutdinov, R. (2015, July). Siamese neural networks for one-shot image recognition. In ICML deep learning workshop (Vol. 2, p. 0).
24. M. Franco-Salvador, P. Gupta, P. Rosso, and R. E. Banchs, “Cross-language plagiarism detection over continuous-space- and knowledge graph-based representations of language,” Knowledge-Based Syst., vol. 111, pp. 87–99, 2016.
25. M. Roostaee, S. M. Fakhrahmad, and M. H. Sadreddini, “Expert Systems with Applications Cross-language text alignment: A proposed two-level matching scheme for plagiarism detection,” Expert Syst. Appl., vol. 160, p. 113718, 2020.
26. Magliacane, S., & Groth, P. (2013, May). Repurposing Benchmark Corpora for Reconstructing Provenance. In SePublica (pp. 39-50).
27. Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014, June). The Stanford CoreNLP natural language processing toolkit. In Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations (pp. 55-60).
28. Mohammad, S., Bravo-Marquez, F., Salameh, M., & Kiritchenko, S. (2018, June). Semeval-2018 task 1: Affect in tweets. In Proceedings of the 12th international workshop on semantic evaluation (pp. 1-17).

29. N. N Chaubey, 2022. "automatic plagiarism detection and extraction in a multilingual: a critical study and comparison," no. 01: 284–304.
30. N. N. Chaubey, "automatic plagiarism detection and extraction in a multilingual: a critical study and comparison," no. 01, pp. 284–304, 2022.
31. Yousaf, F., Iqbal, S., Fatima, N., Kousar, T., & Rahim, M. S. M. (2023). Multi-class disease detection using deep learning and human brain medical imaging. Biomedical Signal Processing and Control, 85, 104875.
32. Potthast, M., Eiselt, A., Barrón Cedeño, L. A., Stein, B., & Rosso, P. (2011). Overview of the 3rd international competition on plagiarism detection. In CEUR workshop proceedings (Vol. 1177). CEUR Workshop Proceedings.
33. Potthast, M., Hagen, M., Stein, B., Graßegger, J., Michel, M., Tippmann, M., & Welsch, C. (2012, August). ChatNoir: a search engine for the ClueWeb09 corpus. In Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval (pp. 1004-1004).
34. Rangel, F., Rosso, P., Koppel, M., Stamatatos, E., & Inches, G. (2013). Overview of the author profiling task at PAN 2013. In CLEF Conference on Multilingual and Multimodal Information Access Evaluation (pp. 352-365). CELCT.
35. Rangel, F., Rosso, P., Koppel, M., Stamatatos, E., & Inches, G. (2013). Overview of the author profiling task at PAN 2013. In CLEF Conference on Multilingual and Multimodal Information Access Evaluation (pp. 352-365).
36. Roostaee, M., Fakhrahmad, S. M., & Sadreddini, M. H. (2020). Cross-language text alignment: A proposed two-level matching scheme for plagiarism detection. Expert Systems with Applications, 160, 113718.
37. Roostaee, M., Sadreddini, M. H., & Fakhrahmad, S. M. (2020). An effective approach to candidate retrieval for cross-language plagiarism detection: A fusion of conceptual and keyword-based schemes. Information Processing & Management, 57(2), 102150.
38. Rosenthal, S., Farra, N., & Nakov, P. (2019). SemEval-2017 task 4: Sentiment analysis in Twitter. arXiv preprint arXiv:1912.00741.
39. R. Damaševičius, A. Venčkauskas, J. Toldinas, and Š. Grigaliūnas, "Ensemble-based classification using neural networks and machine learning models for windows pe malware detection," Electron., vol. 10, no. 4, pp. 1–26, 2021.
40. Rasool, S., Khan, A. H., Rasool, Q., Abbas, S., & Hussain, S. K. (2023). A Systematic Analysis for the Detection of Skin Disease Using Deep Learning Methodologies. Journal of Computing & Biomedical Informatics, 4(02), 66-75.
41. S. Kulkarni, Sagar Kulkarni, Dr. Sharvari Govilkar D. Amin. n.d. "Analysis of Plagiarism Detection Tools and Methods," 1–7.
42. Safi, Faramarz & Rakian, Sh & Nadimi-Shahraki, Mohammad H.. (2017). English-Persian Plagiarism Detection based on a Semantic Approach. 5. 275-284.
43. Vaswani, A., Bengio, S., Brevdo, E., Chollet, F., Gomez, A. N., Gouws, S., ... & Uszkoreit, J. (2018). Tensor2tensor for neural machine translation. arXiv preprint arXiv:1803.07416.
44. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.
45. Maqbool, M. S., Hanif, I., Iqbal, S., Basit, A., & Shabbir, A. (2022). Optimized Feature Extraction and Cross-Lingual Text Reuse Detection using Ensemble Machine Learning Models.