PlagCaps: Prediction of Plagiarised Text on a Corpus Dataset using Deep Learning Algorithms

Prathmesh Shukla

x19231644

Abstract

Plagiarism detection in the field of education and research is a challenging and tedious task. Recently created machine learning algorithms are mainly focused on string-level analysis and comparison. An untrustworthy and faulty model used for plagiarism detection is not beneficial to the education and research sector. Advancement in the field of deep learning is helping in reading huge text and image-based datasets in no time with the help of its pre-trained models. This research is proposing state-of-art Capsule Networks (PlagCap) and Long Short-Term Memory (LSTM), deep learning models for text classification purposes on IMDB movie reviews and Quora Question Pair datasets. Natural Language Processing Tool Kit (NLPTK) is also used for data pre-processing, and Glove a pre-trained word vector model is used for word embedding. These two models are performing well with huge corpus base text data. An accuracy of 85.70% by LSTM and 94.99% by Capsule Network model is achieved and these models are outperforming all the previously done researches. These models can be used in real-world applications to improve the accuracy of plagiarism detection techniques.

1 Introduction

Plagiarism is the unacknowledged usage of someone else's content without providing them appropriate credit and manifesting it as their work by altering the meaning of idioms or paraphrasing the original data. Nowadays, every single piece of information is available on social media and data sites. This can be disseminated over the Internet; people have begun to search for literature online, resulting in other academics, particularly in the education area, plagiarizing others' work without attribution. Addition topic of interest is semantic identity identification, which is concerned with determining how similar two texts are (Harispe et al. 2015).

1.1 Research Motivation and Background

Since the Internet's popularity and usefulness have expanded, the ability to analyze textual similarities in a given set of sentences has proven helpful for a wide range of applications. Machines must understand human language and implicit meanings in various applications, including question answering, suggestion systems, and conversational bots, to name a few. Over several years and a vast number of tests, paraphrase detection algorithms have proven reliable and effective, particularly for clean text corpus datasets. Because paras are expected to have a high level of semantic similarity, systems utilized for para identification should also distinguish analytical similarity in text inputs (Altheneyan and Menai 2020). Sentence similarity can't always be studied using conventional Information Retrieval (IR), or Natural Language Processing (NLP) approaches. As, there are no related words in the texts, or the word regeneration is unusual. In this regard, a number of brief text similarity treatments have lately been proposed.

While preventing such an issue is essential, especially for educational reasons, it is also necessary to discover plagiarized situations. Over the years, many techniques for automatic plagiarism detection have emerged, all of which are based on the use of natural language processing text detection. Turnitin (iParadigms 2021) is the most widely used plagiarism detection tool in education, while CopyCatch is also used in companies. MOSS is used to detect plagiarism in computer source code. Current sentence similarity procedures are largely string-based and are based on theory. They are unsuitable for frequent reuse owing to their limitations. Moreover, scalability issues are rarely explored. This study aims to provide a fast technique for recognizing paragraphs in texts that fit the semantic meanings of the text.

Table 1: Research Objectives and Contribution

Obj.	Description	Evolution Metrics
1	In Dept analysis of existing text-classification techniques used for Plagiarism Detection(2019-2021)	
2	Data Cleaning, Feature Selection using NLPKT, Data Transformation using Word Embedding	
3	Implementation of deep learning models	Accuracy
3.1	Implementing LSTM Model on Quora Question Pair Dataset	Loss
3.2	Implementing Capsule Network Model on Quora Question Pair Dataset	Precision
3.3	Implementing Capsule Network Model on IMDB Movie Reviews Dataset	Recall
4	Comparing the results of each model with respect to each dataset	f1-Score
5	Comparing the results of each model	

Using a word vector from a pre-trained embedding model is also a useful technique to deal with complex texts.

Deep learning models are the current state-of-art for studying models which are used in computer vision, huge text datasets, and image classification. Capsule networks are the recent hit in the research sector as it working smoothly with all kinds of the dataset. It learns on spatial content of a text using its multilayer capsule architecture and provides the best models which basic convolution networks used to fail. Similarly, LSTM network-based models are advancements of RNN models. Since the RNN model can't keep a piece of information for a longer time due to its network layer architecture and hence LSTM networks are also used here for model training so that an accurate plagiarism detection model can be built.

1.2 Research Question, Objectives and Contribution

This research's Plagiarism Detection model will help educational institutions and business sectors where large sets of raw text data are generated daily. This will further help in increasing the data security and integrity on cloud platforms.

RQ: How can Plagiarism Detection Accuracy can be improved by making use of deep learning techniques (*LSTM and Capsule Networks*)? Different sizes of datasets will be used for testing the performance of the models and a comparative study will be done.

Sub-RQ: To what extent a paraphrased sentence can be identified with the help of deep learning models, and how effective is NLPKT techniques are on massive text classification datasets? Performance of Glove, a pre-trained word vector model, will be used for feeding the data inputs to deep learning models, and based on their performance metrics, results will be evaluated.

The following research purpose in table 1 will be reviewed and performed in detail so that the proposed research question and the sub-research question can be acknowledged also, all the objectives are implemented, evaluated and relevant results are given in section 6.

Apart from this, section 2 is dealing with all the related work and literature reviews done on recent deep learning models which were used for plagiarism detection (2019-2021). Section 3 represents all the text methodologies that were used while developing the plagiarism detection models. Next section 5 is used for illustrating the results and a final discussion was done over the objectives in section 7 which are proposed in section 1.2.

2 Related Work

One of the most challenging jobs is applying and detecting the proper degree of plagiarism on a given corpus dataset. In the business world, there are several strategies and tools to choose from, and selecting the right ones that will work together to provide the greatest results is critical. Text categorization has benefited from advances in deep learning and machine learning technology during the past year. Referring to these technologies will aid in understanding text data insights and building a better model based on that analysis for this study.

2.1 Role of Machine Learning in Paraphrase Detection

Not only the regular text but also the programming codes, researchers use each other's programming code and logic to build their projects without giving proper credits to the original author. A machine learning model built by (Viuginov, Grachev, and Filchenkov 2020) deals with plagiarism detection on the programming code level. They suggested a newly created feature set that permits Association for Computing Machinery (ACM) solutions to be acknowledged feature vectors. This feature set includes both Abstract syntax tree and C++ source code disassembly capabilities. They used feature extraction methods to select features from around ninety solutions to five distinct challenges of varying complexity. They suggested a dataset collection technique and cost function to reduce plagiarism detection challenge to a binary classification issue. They organized ten coding events to prepare the dataset and formulated 62 C# and C++ programming challenges. They developed classifiers and presented a method for calculating the significance of specific characteristics. They demonstrated that even utilizing only the top 10% of suggested features, a classifier for plagiarism detection can be well-trained. However, the accuracy of the model was good, but their code was restricted to only two programmings languages.

A hybrid approach for intelligent plagiarism detection that divides the process into three steps. The first step is clustering, the second step is vector formulation in each group based on grammatical roles, normalization, and homogeneity index computation. The last step is summary creation by applying encoder-decoder. They used K-means, calculated on the equivalent set for the stated word; an effective weighing strategy was proposed to choose terms utilized to create vectors. The next semantic argument was examined only once the value determined in the previous stage exceeds a preset level. A summary for paraphrased papers is produced when the similarity score for two documents surpasses the threshold. Tests indicate that, in addition to identifying literal plagiarism, the vector space model with the RNN technique can detect essence and hide employed in concept plagiarism (Nazir, Mir, and Qureshi 2021).

Apart from copying the code, business text data, there are cases where researchers copy abstracts, methodology, and other contents from a scientific research article that is not theirs. To track down such problems of plagiarism, a machine learning, and programming-based model was created by (Helmiawan et al. 2020). Named Entity Recognition (NER) is a technique for recognizing people's personal information like address, names, and database structure. It may, however, be generalized to distinguish DNA, proteins, and other things as specified. NER is profitable in many NLP (Natural Language Processing) purposes, such as QA, summarizing a given topic, and conversation delivery systems, since it may reduce complexity. Link classification, event disclosure, and temporal examination are among the other knowledge removal tasks completed by NER (Plank 2020). Their algorithm will examine and evaluate the harmony of words and phrases in the research papers with other document databases, resulting in a substance for assessment, forecast, and resolution of whether or not the text is copied.

Machine learning methodologies are frequently used for text classification datasets. Sentimental analysis can also be done on such text datasets using machine learning models. (Manna, Pascucci, and Monti 2020) identified a strategy for dealing with the Profiling Fake News Spreaders on the Twitter mission at PAN 20201. The job aims to recognize users who spread fake news (consciously or unintentionally) in English and Spanish. They combined solely stylometric characteristics, emoji kinds, and a plethora of linguistic features linked to the language of false news headlines with various machine learning techniques. They employed 1000 tweeter users in two distinct languages, 500 of every, for model training. They separated it into two groups (The first group with 300 tweets whereas the second was 200 with tweets). To diagnose and recognize counterfeit news spreaders, they included two sorts of features in their templates. The first is linked to stylometric elements directly. The second is about lexical features divided into two sections: 1) lexical components showing personal impression in online connection and 2) clickbait verbs and expressions in misleading news headers. For the Spanish sub-task, their models produce an accuracy of 73% percent (using the Logistic Regression). In contrast, for the English sub-task, they got an accuracy of around 60% using Random forest.

2.2 Role of Natural Language Processing in Paraphrase Detection

Improving the accuracy of word similarity metrics can improve various NLP tasks such as Question Answering System and Document Processing, ending in more intelligent and better IoT-based applications like Healthcare. (Zhang et al. 2021) presents a new technique for determining word similarity that incorporates knowledge graphs and word embedding. For each word pair, the primary concept is to combine the similarity values computed by Knowledge Graph-based techniques and Word Embedding-based methods like glove and word2vec, resulting in diverse combinations. The weights of the constituent techniques are allocated to these combinations depending on their entropy. The experiment's baseline was constructed of combinations of techniques from the same class (KG-based methods or WE-based methods). The outcome of the experiment proofs that the Knowledge Combined and Word Embedding approach perform substantially better than other methods for evaluating word similarity, with the most prominent Spearman correlation coefficient on three of the five datasets. As a result, the KCWE word similarity scores are the most analogous to human evaluations.

An NLP problem is identifying if two text parts have the same meaning. Several NLP applications rely on a solution to this problem, including automatic plagiarism discovery, manuscript summarization, computer translation, and puzzle answering. Two types of programs for defining paras in the discussion: similarity-based classifications and analysis methods. (Altheneyan and Menai 2020) examined and reviewed by developing strategies for paraphrase identification and how these techniques can be used to automated plagiarism detection. Word overlap, structural designs, and machine translation metrics were function subsets that contributed to the highest achievement outcomes for assist vector machines in both summary identification and piracy detection on corpora, according to the conclusions. The effectiveness of deep learning procedures demonstrates that they were the most impressive research maneuvering in this field. Different machine learning and deep learning models were used after data pre-processing in NLP. They used MS Research Paradigm as well as the Twitter corpus dataset; this study founded that Space Vector Machine-based methods and deep learning models produce the greatest results when working on text datasets.

String-matching algorithms do not function effectively in an NLP-based approach when the text has undergone major semantic and syntactic changes. To identify some modifications, linguistic approaches capable of a more thorough analysis of the text are required. Only a superficial amount of investigation has been done utilizing linguistic techniques to detect plagiarism to date. The notion is that original and revised texts have useful but noticeable changes. The intent of this study was to put a deep dive and assess the effect of text pre-processing, mathematical, superficial, and deep linguistic approaches on plagiarism detection utilizing NLP on a corpus dataset. Experiments determined that combining lightweight and deep methods enhances plagiarized text categorization by overcoming the frequency of false negatives. Moreover, the test on distinguishing plagiarism paths explains that altered texts may be detected using statistical and morphological features. The outcomes of the study indicate future research topics and applications for solving the challenges of text reuse detection (Chong 2013).

2.3 Role of Deep Learning models in Paraphrase Detection

Parsipayesh, a framework designed by (Lazemi and Ebrahimpour-Komleh 2020) which was used to identify the plagiarized text in a given scientific article based on the Persian language. They had utilized statistical characteristics to create candidate data, and they had applied semantic analysis together with structural retrievals from the data while texting data alignment. They had observed the structure of the plagiarized text utilizing a deep learning parser and later used a dominion tree to measure the dimension of similarity between the text compositions. For this study, they used the AAIC and PAN datasets from the year 2015. They had erected a keyword dataset, which they compared each page to. This document is deemed to be plagiarized when the similarity score exceeds a certain outset value. A Bi-LSTM based network uses distributive maxims as input in the first stage, and an LSTM based network takes paired input in the second stage. A masked layer in the third stage highlighted a corrected activation tool that combined input from the previous two stages and the output layer in the last stage for generating the scores for each pair. Due to a paucity of data in the applicant retrieval data for the Persian language, this research was not fully evaluated. To get a commending outcome in future work, a richer dataset and a solid deep learning design would be necessary.

Human plagiarism detection is a time-consuming, inefficient, and confusing process. In this work, (El Mostafa Hambi 2021) proposed a plagiarism detection method based on several deep learning models. The Data Pre-processing Layer, which incorporated word embedding, the Deep Learning Layers, and the Final Detection Layer made comprised this substructure. The two documents will be preprocessed and converted into a collection of vectors using the doc2vec paradigm. The framework will next conclude if the input papers have been recognized using the S-LSTM Model, and it will reveal the possibilities of each type of plagiarism taught in their arrangement using the Convolutional NN Model. They performed research on plagiarism detection systems in the educational field and confronted them using a set of criteria to assess this method. In contrast to other efforts, their method has a precision of around 98%, can recognize many types of plagiarism, allows you to blueprint another dataset, and can compare a text from an internet exploration.

A comparative study centered on a variety of factors such as vector representation model, level processing, comparison technique, and the dataset was done practicing high-quality vector illustrations of words or phrases through a deep learning algorithm. As a result of this study, most experimentation employs the word2vec method and concentrates on word granularity, which isn't necessarily ideal for retaining the meaning of entire sentences. When employed with deep learning models, they claim Mikolov designs, rather than w2vec and glove, are the best techniques for maintaining the semantic component of the supplied text. They utilized a variety of datasets to train the model and discovered that most techniques employ cosine to calculate text similarity. Word similarity analysis was done on a word-by-word, either sentence-by-sentence basis, it was revealed. This might be untrustworthy; for example, two papers may have the same words or sentences but are not significantly comparable (El Mostafa and Benabbou 2020).

2.4 Role of Capsule Networks in Text-Classification

Academics have been watching at ways to deal with loathsome language since it has shifted a rising solicitude for online social media platforms and creating algorithms to detect various sorts of it, such as cyberbullying, loathe speech, brutality, etc. The majority of comments on this issue had thus far accumulated in English, with a few meaningful exceptions. The availability of English language aids is to blame for this. To approach this, the Offensive Greek Tweet Dataset, the first Greek annotated dataset for foul language detection, was presented in this practice. It's a professionally annotated dataset that holds around 5000 tweets that have been classified as abusive or not abhorrent. In addition to giving a detailed description of the dataset, they examined and evaluated different theoretical models and came out with their results. Deep learning models from several other suppliers were considered after data preprocessing. Many of these templates were utilized in stimulus apprehension work. Among the templates employed are Merged GRU, Piled LSTM with Caution, L-STM and GRU with Caution, two-D CNN with Pooling, Gated Recurrent with Capsule, L-STM with Capsule and Attention, and a BERT as well. In parallel to all of the machine training models, the L-STM and GRU with Attention models produced more reliability (Pitenis, Zampieri, and Ranasinghe 2020).

Text classification is a time-consuming and challenging process that combines text recognition and categorization. The Capsule Network is an interface design competent for seizing fundamental characteristics. (Jacob 2020) suggested a Capsule Networks-based multitasking framework because it allows for the separation of insights across various activities and indirectly enhances data utilized in training. In such learning, the proposed architecture, which comprises Capsule-Networks, efficiently classifies the text and eliminates interference among many jobs. To ensure the viability of the suggested framework, the proposed approach is tested on a variety of classification datasets. It is possible to eliminate the capsule network using cluster function prediction using vectors rather than scales, resulting in better text identification. It also makes use of the gated distribution unit (Ballakur and Arya 2020) to prevent users from giving irrelevant material. Multitask learning is used in the suggested technique to increase text categorization efficiency in NLP. It uses gated multitask to sift out the useless, as well as the Caps-Net with task-based routing to avoid task interference. Three types of data sets collected from the internet were used to fit and prove the proposed system. Out of all the model capsule network performed well as compare to other models.

Language conversion, object identification and classification, natural language processing, and image processing are examples of modern computer vision-based jobs that demand cutting-edge solutions. Because symbolic AI is unable to tackle today's challenges due to its complicated coding patterns, deep learning models such as CNN and RNN have been introduced. CNN, on the other hand, requires a lot of training data and is unable of comprehending object position and deformation, which is why Capsule Networks were created. (Kwabena Patrick et al. 2019) performed a study to verify the newly built capsule network-based architectures and techniques utilized in the current application. The MNIST dataset is employed to create capsule network architecture since it is a vast and complicated dataset, and a model built on it may be utilized on a broad measure. All given sets of the capsules utilized in this study were examined for model assessment after the model had been trained. Margin loss was computed, and elements that detracted from performance were eliminated, allowing the specific design or posture to be determined.

3 Research Methodology

3.1 Introduction

This research is building a deep learning model to identify a plagiarised text in a given corpus. Multiple datasets are being used for this research. Starting with data loading, pre-processing, transformation and at the end model building and testing the accuracy of the model of train dataset. Exact structure of the designed system can be seen in section 4. To evaluate the model's overall performance a thorough testing on the test dataset is done and accuracy, precision recall and f1-score is calculated.

3.2 Text Classification Methodology

Plagiarism detection methodology figure 1 is explaining the knowledge and data discovery stages referred as KDD data mining techniques. It consists of multiple stages as given below:

1. Raw data loading and processing
2. Feature selection: Selecting 3 target columns
3. Data cleaning: Using NLP techniques
4. Data transformation from text to vector
5. Data Mining: Building architecture and implementing deep learning models using Keras and Tensorflow and various Python libraries
6. Evaluation: Using performance metrics for the model evaluation

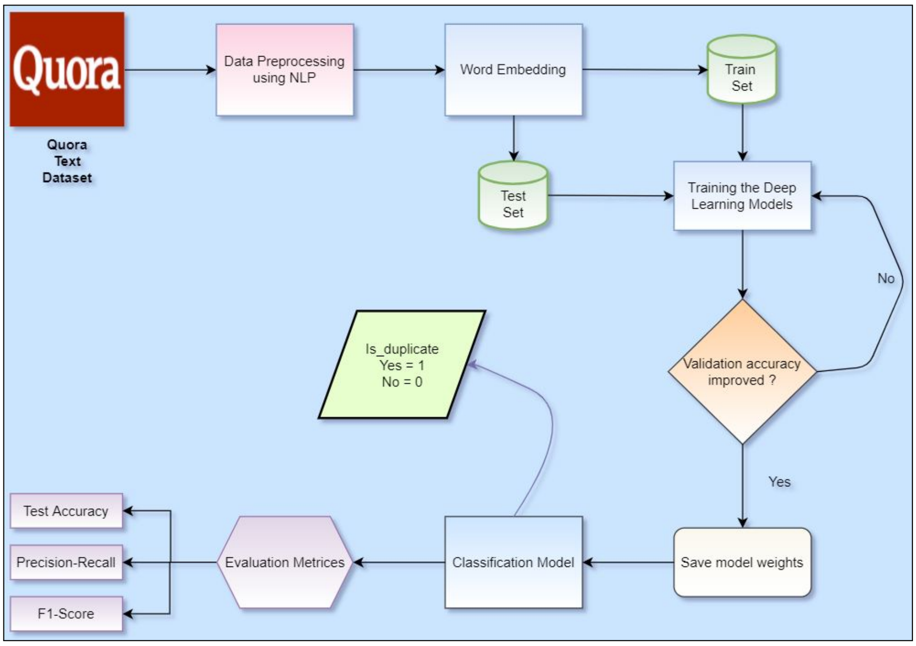


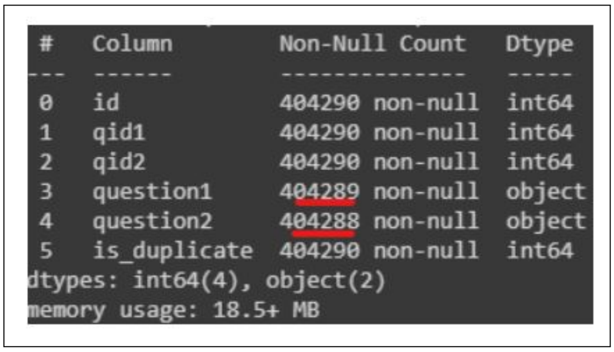
Figure 1: Design Flow Structure

3.3 Research Understanding

At the beginning of the research, objectives are defined based on the real-world example of detecting plagiarism in the education system. After that, all the planning and goals are presented correctly in an organized manner. The research is aiming to build a novel deep learning model to identify paraphrased sentences on the basis of their semantics and to do so the model should not consume huge operation costs but provides better outcomes in terms of good accuracy. A better plagiarism detection model not only helps the education sector but also helps other sectors where researchers do an immense amount of work to formulate a great idea into consideration.

3.4 Data Understanding

Data gathering starts in the beginning and lays a foundation for future outcomes. For this research, multiple datasets will be used for training the model. But, the main dataset which will be used for performance evaluation is Quora Question and Answer (QA) pair dataset. This dataset has 5 columns and contains 404290 QA pairs in the form of rows as shown in figure 3.4.



```
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   id            404290 non-null  int64
 1   qid1          404290 non-null  int64
 2   qid2          404290 non-null  int64
 3   question1     404289 non-null  object
 4   question2     404288 non-null  object
 5   is_duplicate  404290 non-null  int64
dtypes: int64(4), object(2)
memory usage: 18.5+ MB
```

3.5 Feature Selection

Applying the model to the raw data will yield poor results and that model can't be used for business purposes. Hence, feature selection is an important step. There are a variety of feature selection tools available in the NLP module as shown in figure 2. NLP is used to extract the correct semantics of the words from a given sentence. Later on, the semantics of two different questions will be compared and similarity evaluation will be carried out on that. A set of NLP data cleaning techniques will be applied to the raw text to generate the cleaned text of the corpus dataset. These techniques are applied in the Python platform with the help of in-built libraries of Natural Language processing.

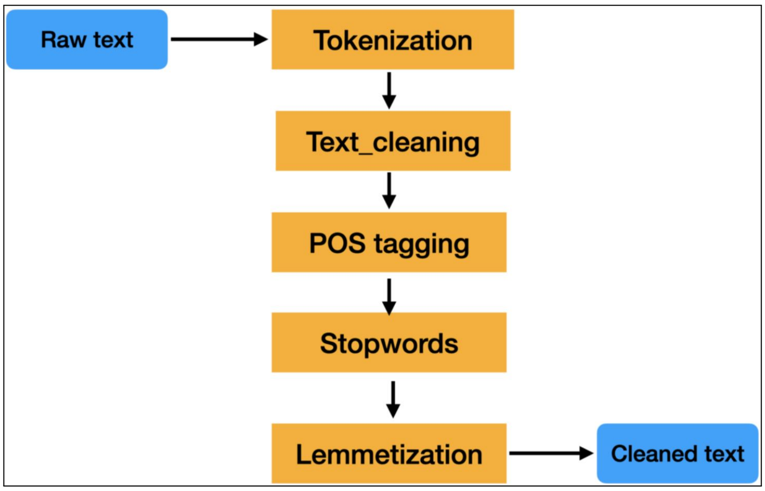


Figure 2: Data Pre-Processing and Feature Selection

- **Tokenization:** Creating tiny chunks or tokens of a phrase is called Tokenization. The text may be tokenized into paragraphs, with each section being divided into terms, and each sentence is tokenized into words, and so on. Because all datasets are already split into sentences, just token splitting is required.

- **Normalization:** Normalization is another crucial stage in the pre-processing process. Normalization entails changing all text to the same case (upper or lower), removing punctuation, and turning any symbols to lower case. Normalization equalizes all words and allows processing to proceed in a consistent manner.

- **Stemming:** The method of eradicating affixes (circumfix, suffix, infix, prefix) in order to obtain a term stem is known as stemming.

  eat, eats, eating, eatable → *eat*

- **Lemmatization:** Normalization includes lemmatization, which captures canonical forms depending on a word's lemma. Stemming the word "better," for example, would not return its citation form (another word for lemma); as, lemmatization can do following:

  better → *good*

- **Stop words removal:** Removing the language's most common terms from the original input text (i.e., the, at). Because such words do not have a specific meaning, they might be overlooked.

3.6 Data Transformation

Cleaned text which is obtained in the previous step, can be seen in the table2. Here, question1 and question2 are a pair of questions, and is_duplicate represents whether they are semantically similar or not. Now, NLP has done the magic of removing all the unwanted texts using its various data cleaning techniques. Hence, q1 and q2 columns represent the clean text data which will be given as input for the data transformation stage. At this stage, embedding is used to convert words to their relevant vector equivalent form so that size of the dataset can be further reduced and data will be ready for feeding into deep learning models for training.

Table 2: Cleaned Text by NLPTK

Sr.No.	Question1	Question2	q1	q2	is_duplicate
1	Why do girls want to be friends with the guy they reject?	How do guys feel after rejecting a girl?	girl want friend guy reject	guy feel after reject girl	0
2	Why do rockets look white?	Why are rockets and boosters painted white?	do rocket look white	rocket booster paint white	1

Data must be separated into training, testing, and validation sets because it was categorized into various categories. Different python functions were used to complete this assignment, using a 60%, 20%, and 20% split for training, testing, and validation, individually.

3.7 Embedding

The most widely used NLP approach is word embedding, which aims to detect low-dimensional vector representations of words from text (Li et al. 2017). Word embedding techniques like Glove and Word2Vec have been shown to help with NLP tasks like aspect removal, POS tagging, and opinion analysis because they can capture semantic and syntactic word connections. The distributional principle central concept, which says that words that appear in analogous situations have related meanings.

The hidden association between words that may be exploited during data training is represented by word embedding. Word embedding approaches look at how a corpus of text with a specific vocabulary size may be described as a real-valued vector. Text categorization can be learned using a neural network task or an unsupervised technique based on text statistics.

The size of the vector space may be adjusted between 50 and 300 dimensions, with random values being used by default. An example of vector space is shown in figure **??**. Glove along with Word2Vec pre-defined word vectors are utilized in the subsequent section, which will be explained in the following sections.

3.7.1 Glove

Pennington at Stanford created the Global Vectors for Word Representation, or GloVe, technique as an addition to the word2vec approach for effectively learning word vectors (Pennington, Socher, and Manning

2014). This is nothing but a method for obtaining vector illustrations for words in an unsupervised corpus of text. The glove offers a variety of models with dimensions of 25, 50, 100, 200, and 300 on 2, 6, 42, 840 billion tokens. Word word co-occurrence probability is utilized to create the embedding, i.e., two words have co-existed for a long time; hence, both comments have similar meanings; so, the pattern will be resembling.

4 Design Specification

The three-tier design architecture for this project is shown in below figure 3. This architecture is comprised of three layers which is given as:

1. **User Interface layer:** Here, the user will make a request to check if a given text is plagiarized or not.
2. **Data Presentation Layer:** This layer holds the inserted data from an excel file source and based on the user request a pre-processed data by an NLPTK getting inserted into the business layer.
3. **Business Logic Layer:** This layer has the logic set by the business for model training using TensorFlow and Keras libraries. A deep learning NN model of Capsule Networks and LSTM Networks is used on the processed data to predict if the inputted QA pair is similar or not.

The overall code execution is done in google collab with a professional version as to execute deep learning models on a huge dataset requires large CPUs, GPUs, and RAMs. Colab provided 25 GBs of GPUs and 69 GBs of disk space along with 25 GBs of RAMs.

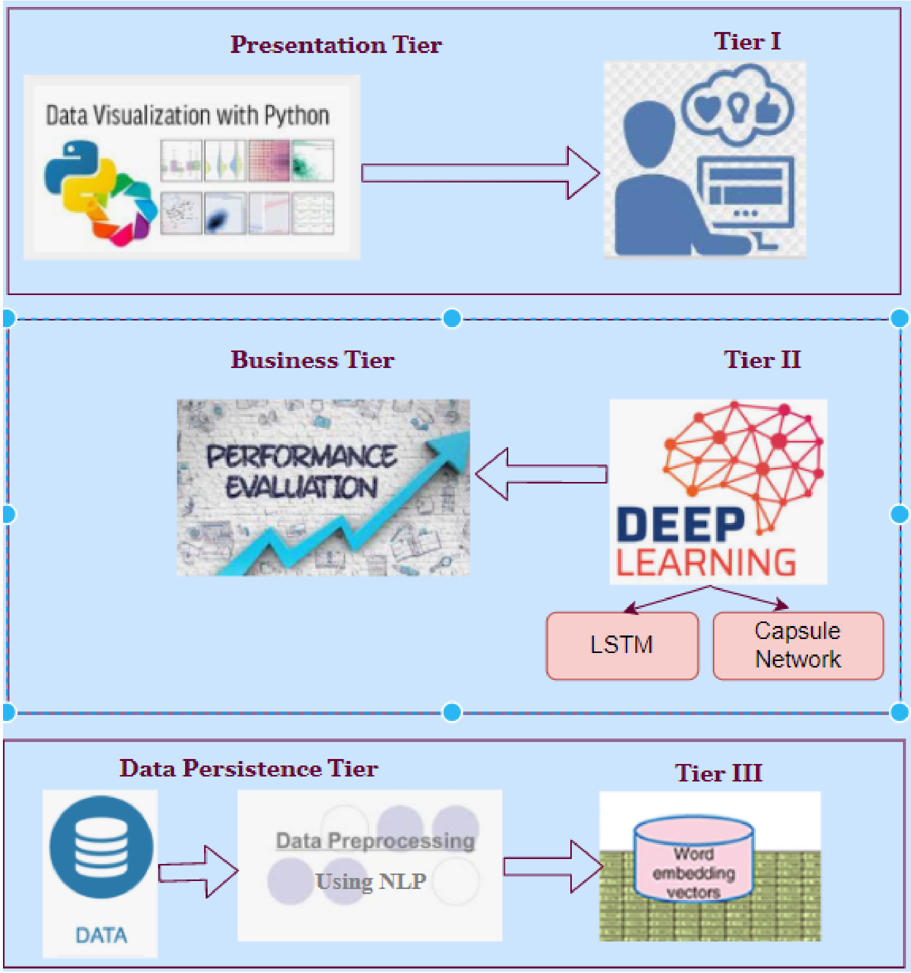


Figure 3: Three Tier Design Architecture for Paraphrase Detection Tool

5 Implementation of Deep Learning Models for Plagiarism Detection

This section gives a description of all the steps taken during the implementation of the paraphrase detection model. In the beginning, an environmental setup was done and later different python libraries like Keras, panda, TensorFlow, scikit-learning, and NLP were used for data pre-processing model building. Furthermore, data splitting and transformation and model evaluation parameters are discussed in detail.

5.1 Environment Setup

For this research, google collab pro was used along with 16 GB windows OS. Pythons' 3.8.5 version was used, and for fast data access, a SAMSUNG SSD of 650 GB was used.

5.2 Data Pre-processing and Transformation

In this research, Quora Question Pair (404290 rows) and IMDB's movie reviews (25000 rows) datasets were used. Both these datasets are available in Kaggle. These were text datasets and hence, word2vec and Glove embedding techniques were used for converting the text data into vector form.

5.2.1 Text Pre-processing & Splitting

Data pre-processing was done in NLPTK using different NLP data cleaning techniques and then it was divided into three sets in the form of ratios for model testing and evaluations. Post-model implementation test data was used for predictions.

5.3 Classification Techniques

In this section, all the deep learning models used in this project to detect similar text are discussed in detail. All the models produced state-of-art outcomes compared with previous research as previous research was done only on small datasets. For this project, two huge text classification datasets were used.

5.3.1 LSTM Model

This was the first model used for plagiarism detection. As there were two question pairs in the dataset for the comparison, the model had seven layers for each pair, comprised of embedding, LSTM, Dropout, 2-LSTM, and 2-Dense layers. After merging these two question pairs into 1, a merged out model was created with 6 more layers. Finally, the model had 20 layers in total with a sigmoid activation function in the end. This model was applied to the Quora dataset and all the defined parameters were used correctly and achieved better performance metrics which indicates, using such a model in plagiarism detection is useful for future implementation.

5.3.2 Capsule Networks Model

This was the second model used in this project to predict whether a given Question pair is identical or not in terms of semantics. It is a 6 layer architecture. Starting with the input layer with a maximum length of 1000, 2-embedding layers, Bi-Directional GRU layers with GRU length of 256, Capsule network layer with 10 capsules and 3 routings, and at the end, a dense layer with sigmoid activation function was created. Finally, the model was created with binary cross-entropy and Adam optimizer function. Accuracy was used for performance metrics. However, this model worked correctly with the IMDB dataset but due to dataset complexity, time, and resource availability, this model has not been applied to the quora dataset. This was applied twice on the IMDB, initially, the dataset was imbalanced and hence a remedy was used to correct the imbalance class, and later on this produces better results.

6 Evaluation and Results

In this section, a thorough study was done on both the models created for this research. Various performance metrics like Precision recall, Accuracy, and F1-score were calculated and later on compared to select the best model out of these two.

6.1 Results

Multiple iterations were performed in the group of two experiments by varying the tuning parameters to achieve a good fitting model among the two-deep learning models. Based on their performance metric score best model was selected.

6.1.1 Experiment 1

In this experiment, the Capsule network was used on the IMDB dataset and after completion of the 10 epochs confusion matrix showed imbalanced data, and some weights were added to improve the model fitting and this can be seen in the experiment section 6.1.2. Now, here as the epochs were increased the overall performance of the model also increased. From the graph figure 4, it is easy to observe the pattern of the graph. An accuracy of 98.83% with 0.0393 training loss was calculated. While training this model, the loss function was binary_crossentropy and adam optimizer was selected.

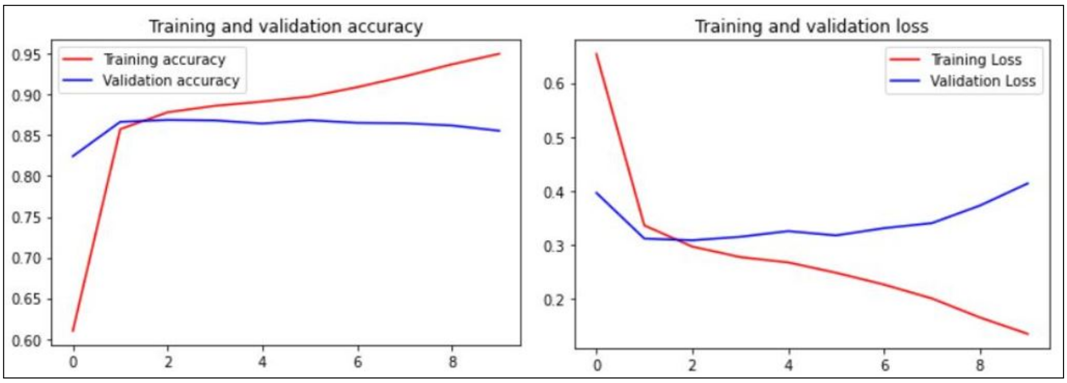


Figure 4: CapsNet: Accuracy and Loss Curve on IMDBs Imbalanced Dataset

6.1.2 Experiment 2

In this experiment same set of configurations were used with some added class weights so that class imbalance can be removed. As seen in the graph figure 5, the model performed well with weights and a smoother graph curve for training was observed. An accuracy of 92.56% with a training loss of 0.1876 was calculated in the last epoch. Although the confusion matrix produced the same number of True positive records the balance class accuracy is less as compare to the imbalance class which indicates improvements in the model fit. Although at the very end of the experiment, the validation loss started to increase and reached up to 0.3632 but on the other hand training loss followed a path towards zero.

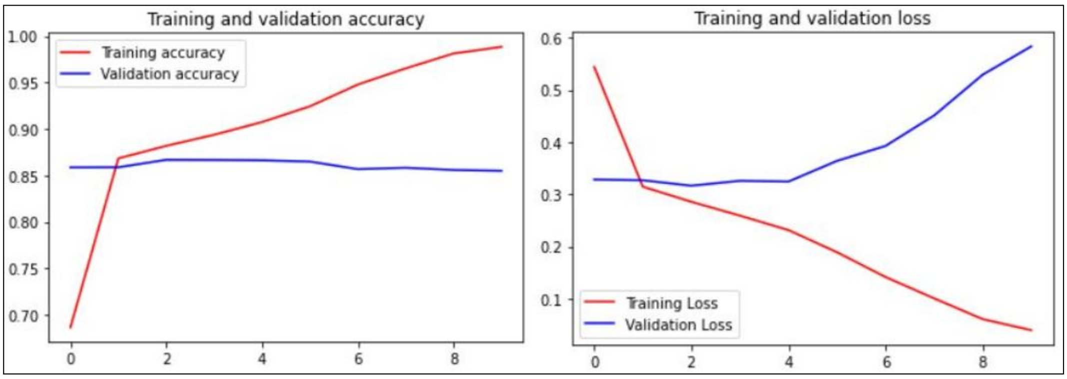


Figure 5: CapsNet: Training and validation accuracy on IMDBs Balanced Dataset

6.1.3 Experiment 3

This was the last experiment that was conducted on a different dataset as the previous two experiments were conducted on the IMDBs dataset using Capsule Networks. Now, for this experiment Quora's Question pair dataset was used with 23 layers consists of input, embedding, LSTM, dropouts, flatten and a multiply layer as for this dataset there were two input parameters and hence two separate models were created and at the very end, both of them were merged to form a merged output model. This model was trained with a batch size of 2000, and 25 epochs. As the data rows in this dataset are more the 40000 and hence, the batch size is kept a bit high, and later on various inputs were tried on this to observe the overall outcomes. Here, Adam optimizer was used with loss function sparse categorical cross-entropy. After a set of executions, the model achieved the best results at the loss of 0.2458 and accuracy of 91.47%. Training rate can be observed from the below graph figure 6 where a training and loss plot is shown. As the epochs were increased, the loss function is gown down and accuracy is following an upward exponential curve.

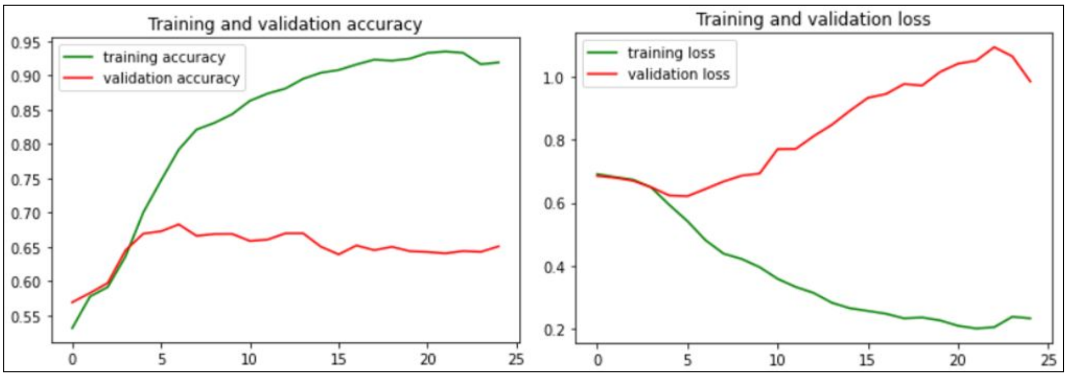


Figure 6: LSTM Model: Training and validation accuracy on Quora Dataset

7 Discussion and Comparison of Results

After a successful model implementation in Capsule and LSTM networks, Now, it is time to compare them with the help of accuracy and loss factors.

7.1 Comparative Study of Classification Models in Terms of Accuracy

Below table 3 is showing accuracy scores of all the models along with their loss factors. Out of all the model's Capsule networks were performing best as they achieved the best accuracy on huge text classification datasets like IMDB Movie Reviews. Capsnet not only received better accuracy but their loss rate is also very low as compared to the LSTM model on text data. As mentioned in table 1, all the models are implemented. All the objectives are achieved except for implementing capsule network model on Quora dataset due to dataset complexity as the dataset had 2 question pairs and merging them into a single field was not allowed in capsule networks.

Table 3: Deep Learning Models Comparative Study in terms of Accuracy

Model Name	**Dataset Name**	**Accuracy**	**Loss**
Capsule Networks (class imbalance)	IMDB	98.83%	0.0393
Capsule Networks	IMDB Movie Reviews	92.56%	0.1876
LSTM	Quora Question Pair	91.47%	0.2458

8 Conclusion and Future Work

A text classification-based research, the aim was to identify plagiarism in a given text corpus dataset using deep learning algorithms. Capsule networks and LSTM were used to train and build the model in a python environment using Keras and TensorFlow libraries. Capsule networks were applied on the IMDB dataset and the model performed exceptionally well and have achieved an accuracy of identifying a paraphrased sentence was 92.56% with a minute loss of information of 0.1876. Initially, the dataset was imbalanced and hence, the same class weights were added to achieve this improved accuracy and loss factor. LSTM was applied on the Quora question pair dataset and after training the model it achieved an accuracy of 91.47% with a loss of 0.2458. This indicates, the performance of a capsule network with a huge and complex text dataset is recommendable and future projects can be done using this. Both these models are state-of-art in the text classification field. Using these in real-world applications will help institutions and other research group sectors in finding data integrity and security.

Due to limited time and resource availability capsule model implementation with the Quora dataset was not done correctly and hence one of the objectives for this research was not completed. Better python libraries and classes are required to build a capsule network model.

Due to this objectives of building a capsule network in the Quora dataset can be also taken as a future work as capsule networks can be beneficial for such kinds of datasets and can yield good results in plagiarism identification.

Acknowledgement

I want to thank my supervisor and guide, Dr. Catherine Mulwa, for monitoring and motivating me throughout the study and to address all of my questions by discussing them in-depth and demonstrating various excellent and poor instances. Her suggestions aided me in writing a more substantial report, and as a result, I was able to complete a cutting-edge solution project. I'd also like to express my thankfulness to my family for their unwavering support during my project's lengthy working hours and at the conclusion. I want to express my appreciation for friends who are always willing to engage in constructive debates that have helped me develop better ideas.

References

Altheneyan, Alaa and Mohamed El Bachir Menai (2020). "Evaluation of state-of-the-art paraphrase identification and its application to automatic plagiarism detection". In: *International Journal of Pattern Recognition and Artificial Intelligence* 34.04, p. 2053004.

Ballakur, Amulya Arun and Arti Arya (2020). "Empirical Evaluation of Gated Recurrent Neural Network Architectures in Aviation Delay Prediction". In: *2020 5th International Conference on Computing, Communication and Security (ICCCS)*. IEEE, pp. 1–7.

Chong, Man Yan Miranda (2013). "A study on plagiarism detection and plagiarism direction identification using natural language processing techniques". In:

El Mostafa, Hambi and Faouzia Benabbou (2020). "A deep learning based technique for plagiarism detection: a comparative study". In: *IAES International Journal of Artificial Intelligence* 9.1, p. 81.

El Mostafa Hambi, Faouzia Benabbou (2021). "A New Online Plagiarism Detection System based on Deep Learning". In:

Harispe, Sébastien et al. (2015). "Semantic similarity from natural language and ontology analysis". In: *Synthesis Lectures on Human Language Technologies* 8.1, pp. 1–254.

Helmiawan, Muhammad Agreindra et al. (2020). "Improving The Detection of Plagiarism in Scientific Articles Using Machine Learning Approaches". In: *ICONISTECH-1 2019: Selected Papers from the 1st International Conference on Islam, Science and Technology, ICONISTECH-1 2019, 11-12 July 2019, Bandung, Indonesia*. European Alliance for Innovation, p. 99.

iParadigms (2021). "iParadigms". In: URL: http://turnitin.com/.

Jacob, I Jeena (2020). "Performance evaluation of caps-net based multitask learning architecture for text classification". In: *Journal of Artificial Intelligence* 2.01, pp. 1–10.

Kwabena Patrick, Mensah et al. (2019). "Capsule Networks – A survey". In: *Journal of King Saud University - Computer and Information Sciences*. ISSN: 1319-1578. DOI: https://doi.org/10.1016/j.jksuci.2019.09.014. URL: https://www.sciencedirect.com/science/article/pii/S1319157819309322.

Lazemi, S. and H. Ebrahimpour-Komleh (2020). "ParsiPayesh: Persian Plagiarism Detection based on Semantic and Structural Analysis". In: *2020 10th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 525–533. DOI: 10.1109/ICCKE50421.2020.9303672.

Li, Yang et al. (2017). "Learning word representations for sentiment analysis". In: *Cognitive Computation* 9.6, pp. 843–851.

Manna, Raffaele, Antonio Pascucci, and Johanna Monti (2020). "Profiling fake news spreaders through stylometry and lexical features. UniOR NLP@ PAN2020". In: *CLEF*.

Nazir, Azra, Roohie Naaz Mir, and Shaima Qureshi (2021). "Idea plagiarism detection with recurrent neural networks and vector space model". In: *International Journal of Intelligent Computing and Cybernetics*.

Pennington, Jeffrey, Richard Socher, and Christopher D Manning (2014). "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Pitenis, Zeses, Marcos Zampieri, and Tharindu Ranasinghe (2020). "Offensive language identification in greek". In: *arXiv preprint arXiv:2003.07459*.

Plank, Barbara (2020). "Neural cross-lingual transfer and limited annotated data for named entity recognition in Danish". In: *arXiv preprint arXiv:2003.02931*.

Viuginov, Nickolay, Petr Grachev, and Andrey Filchenkov (2020). "A Machine Learning Based Plagiarism Detection in Source Code". In: *2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence*, pp. 1–6.

Zhang, Dehai et al. (2021). "A novel word similarity measure method for IoT-enabled Healthcare applications". In: *Future Generation Computer Systems* 114, pp. 209–218.