A

Minor Project Report

on

Sentence Similarity Based Framework Using RNN For Plagiarism Detection

Submitted for partial fulfillment for the degree of

Bachelor of Technology

(Computer Science and Engineering)

in

Department of Computer Science and Engineering

by

Arihant Jain
189301107

Under the Guidance of
Ms. Bali Devi

(July-2021)

SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY

MANIPAL UNIVERSITY JAIPUR

CERTIFICATE

Date: 18/06/2021

This is to certify that the project titled **Sentence Similarity Based Framework Using RNN For Plagiarism Detection** is a record of the bonafide work done by **Arihant Jain** (189301107) submitted in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology (B.Tech) in **Computer Science and Engineering** of Manipal University Jaipur, during the academic year 2020-21.

Ms. Bali Devi
Project Guide, Dept of Computer Science and Engineering
Manipal University Jaipur

Dr. Sandeep Joshi
HOD, Dept of Computer Science and Engineering
Manipal University Jaipur

ABSTRACT

In the world of academia, one of the worst crimes one can commit is plagiarism. It is a very serious offence that can thoroughly discredit any scholar or researcher as it shows not only their dishonesty, but also their lack of respect for the institution they belong to. Despite the harsh nature of the offence, with the advent of the information age, it has become increasingly harder to determine if and where a document has been plagiarized from.

As a result, we now require an automated system that is able to detect plagiarism between pairs of documents with a high degree of accuracy and with short enough checking times that we may compare several thousand documents in a reasonable amount of time. My model uses a semantic similarity framework to generate sentence scores which allows a deep neural network to make predictions and determine whether a document in plagiarized or not.

LIST OF TABLES

LIST OF FIGURES

Contents

1. INTRODUCTION

1.1 Introduction

Since the advent of the internet, we have access to more and more information every day. Not only does the amount of information at our fingertips increase every day, but the speed at which we can access it is also improving daily. Never have we had petabytes of data just a simple search away. Despite the untold benefits of this, we also face a growing problem of plagiarism. Some may call plagiarism the worst offense in an academic institution. It undermines the hard work of others and is a very dishonest thing to do.

As such, we must develop methods of allowing an unsupervised system to check whether a submitted document is plagiarized from a list of documents. Furthermore, if it is plagiarized, the system should also be able to determine which sections of the document are plagiarized. If a sentence's vocabulary has been changed yet its meaning remains the same it's still considered to be plagiarized and thus, a simple one to one comparison of sentences will not be sufficient. Due to the large amounts of data that would be required to check the document against, it is physically impossible for a human operator to perform these checks.

My proposed model shall combine the powers of both Recurrent Neural Networks as well as Deep Neural Networks in order to make accurate estimations of where and how much plagiarism exists in each pair of documents. Hopefully by the end of the project my system will be able to compete and even outperform current state of the art systems that are being used in professional settings.

1.2 Motivation

As somebody who has been interested in deep learning since my early teens, I knew I wanted to do a project in deep learning. After going through all the problem statements relating to the topic, I finally settled on this project because it allows me to learn about Recurrent Neural Networks as well as allow me to prepare a project that pertains to a very serious real life issue. While there are several methods that exist already for detecting plagiarism in documents, I feel I can bring something new to the table and put forth a solution that is novel, but also builds upon the successful results of others. As a result I decided that this would be a good place to start implementing and learning about various different networks and help provide a better and more accurate solution to this serious problem.

1.3 Literature Review

Table 1.3.1: Literature Review

SR. NO.	***PAPER***	***PROS***	***CONS***
1.	A Multi-Level Plagiarism Detection System Based on Deep Learning Algorithms[1]	Determines The Overall Degree Of Plagiarism Using SoftMax Classifier	Unable To Provide Exact Location And Amount Of Plagiarism
2.	Semantic Similarity Between Sentences[2]	Determines Similarity Between Pairs Of Sentences Using Several Methods	Only Provides A Partial Picture Regarding Plagiarism Due To Complexity of Natural Language
3.	Semantic Plagiarism Detection System For English Texts[3]	Detects Paraphrasing Between Documents using BiLSTM RNN	Uses Linear Thresholding To Recognize Plagiarism Which May Fail To Recognize Split Sentences, Reshuffled Content, etc.

As a result of reviewing the literature mentioned above, I discovered some pros and cons about each paper and that helped me in coming up with a solution that tries to work around these issues. For example:

1. Allowing only the Recurrent Neural Network to make predictions is not very useful as it cannot deal with the complexities of Natural Language.
2. While SoftMax classifiers can neatly and easily provide a category of plagiarism, it cannot provide specificity and clarity on the location of plagiarism.
3. Due to the complex nature of Natural Language, we may face issues with false negatives if we decide to use simple linear thresholding between pairs of document vectors.

These are just a few of the conclusion that were derived from reviewing literature and they clearly show certain issues that must be avoided by the system such as using a more complex thresholder than a simple linear one in order to reduce the number of false negatives.

1.4 Problem Statement

This application presents a recurrent adaptation of the neural network for labeled data comprised of pairs of variable-length sequences. This model is applied to assess semantic similarity between sentences, where we exceed state of the art, outperforming carefully handcrafted features and proposed neural network systems of greater complexity. This model provides word embedding vectors supplemented with synonymic information to the network artifacts. This is a framework-oriented application for maintaining the plagiarism of any artifacts or documents.

1.5 Project Objectives

After reviewing literature, I have come up with certain objectives that I wish to achieve in the development of this project :-

1. To determine a score for each sentence using a Semantic Similarity framework
2. To compute a reduced form of documents for quick retrieval and comparison
3. To determine whether plagiarism exists between a pair of documents

2. BACKGROUND OVERVIEW

2.1 Conceptual Overview

Unlike humans, computers are incapable of understanding anything other than binary. When we look at a pair of documents, we are able to understand their meaning and easily discern whether they contain plagiarism. Computers, however, are incapable of performing this task. In order for a computer to be able to understand a text document, it needs to be converted into a vector of number such that the computer can use those numbers to "understand" the text. This where the concept of word vectors comes up. Word vectors are a *(1, n)* dimensional array that a computer uses to identify and draw relationships between words. These word vectors are at the core of any processing a Recurrent Neural Network performs on a sequence of words. The word vectors utilized in this project were provided courtesy of Stanford University. Once we have these word vectors, we can utilize them to build an array of word vectors corresponding to every word in a sentence, and every sentence in a document. With this three-dimensional array, our model can now begin learning to discern the existence of plagiarism. This process however is lengthy and time consuming to execute for every document on the fly. Hence, we may convert documents into a vector of scores for each sentence by teaching the recurrent neural network to generate unique scores depending on the words in a sentence. This drastically reduces storage size and improves processing time significantly. Armed with these scores, a deep neural network is able to compare two such vectors and determine the presence of plagiarism in them.

2.2 Technologies Used

This project uses the power of Recurrent Neural Networks and Deep Neural Networks in order to achieve its objectives. These models were built, tested and executed on a Windows 10 operating system using Python v3.8.8 and TensorFlow v2.5.0. Python and TensorFlow were run on the Spyder IDE v5.0.3.

3. METHODOLOGY

3.1 Detailed Methodology

The proposed system shall follow the steps depicted above in order to produce a final output of where and how much plagiarism exists in a pair of documents.

1. **Input:** When checking for plagiarism in a suspicious document, we will input both the suspected as well as a number of source documents as pairs. Each document may be of whatever length and must be in English following ASCII conventions.
2. **Split Document:** Both the documents will split into a vector of sentences and will be sent forward to step 3.
3. **Vectorize Sentences:** Once an array of sentences has been received, each sentence will be individually cleaned and vectorized using word vectors to prepare them for score generation in the next step.
4. **Score Prediction:** Each sentence vector will be passed through the trained recurrent neural network in order to produce an output vector which will further have all of its elements squared and added to produce a final score.
5. **Similarity Matrix:** Once the system has created a vector containing scores for each sentence in a document, the vectors will be used to compute a similarity matrix for each pair of documents.
6. **Predict Plagiarism:** When the similarity matrix is passed to the system, it will be sent through a Deep neural network which will output a 1 or a 0 where 1 represents plagiarism and 0 represents no plagiarism.

3.2 Block Diagram

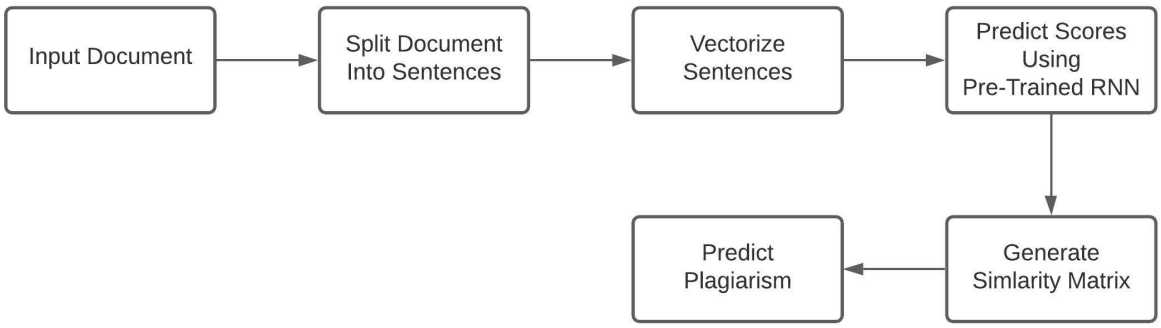

Figure 3.2.1: Methodology Flowchart

4. IMPLEMENTATION AND RESULTS

4.1 Recurrent Neural Network

The STSBenchmark dataset is being used to train the RNN in my project and to do so I had to clean and restructure the data. It presents sentence similarity between pairs of sentences on a scale from 1.0 to 5.0 which I converted to a scale of 0.0 to 1.0 and converted the file into a .csv file for quick read and write operations.

Table 4.1.1: RNN Dataset Metrics

Type Of Example	Number Of Examples	Percentage Of Total
Train	8,935	90%
Dev	496	5%
Test	496	5%
Total	9,927	100%

After obtaining and preprocessing the dataset, I designed and trained a Recurrent Neural Network to be able to generate the scores for a sentence. The Structure of the Neural Network used is shown below.

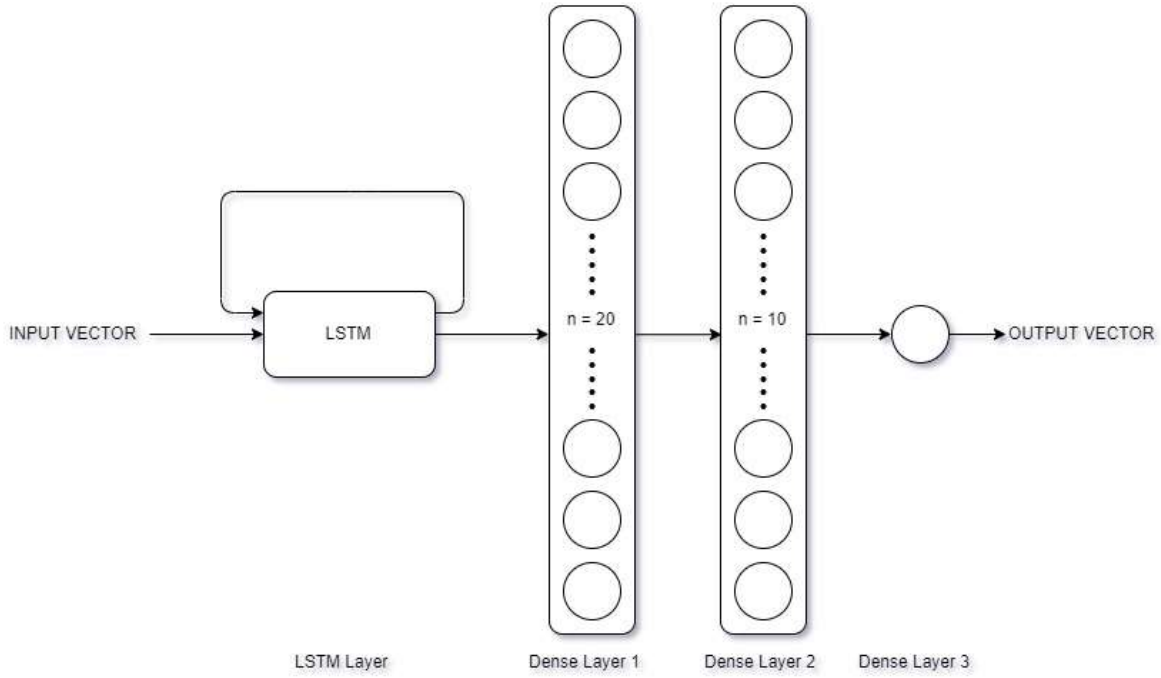

Figure 4.1.1: RNN Architecture

The above network was trained for 100 epochs with a batch size of 32 and a learning rate of 0.001 on an Adam Optimizer. The resultant data is given below.

Table 4.1.2: RNN Training Metrics

Data Type	**Accuracy**
Training	80.15%
Dev	79.69%
Test	79.49%

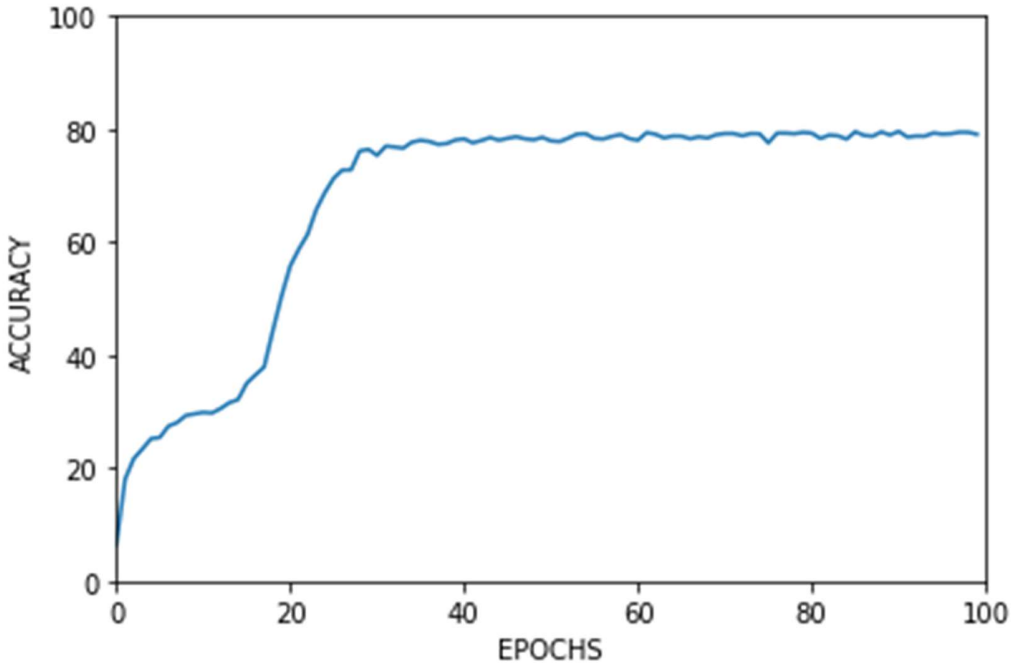

Figure 4.1.2: Accuracy Graph

Due to the difference between all three datasets being less than 1 percent, we can confidently say that the system is not overfitting and is working as intended. The STSBenchmark website also lists several other models developed by both the team who built this dataset as well as independent researchers and my model seems to be competing with some of the highest accuracies.

4.2 Deep Neural Network

The second step of the project was to take these predicted scores and use to determine the occurrence of plagiarism. The Dataset used here was taken from the PAN Corpus, used for detecting plagiarism in documents. Due to time and hardware constraints, I was only able to use a total of 1780 files for the formation of my dataset and thus I procured 4,855 total data pairs. Those pairings were split into training and testing data as given below.

Table 4.2.1: DNN Dataset Metrics

Type Of Example	Number Of Examples	Percentage Of Total
Train	3,884	80%
Test	971	20%
Total	4855	100%

The Network used to determine whether plagiarism is occurring or not here, is a simple deep neural network that utilizes a dropout probability of 20% in order to prevent overfitting.

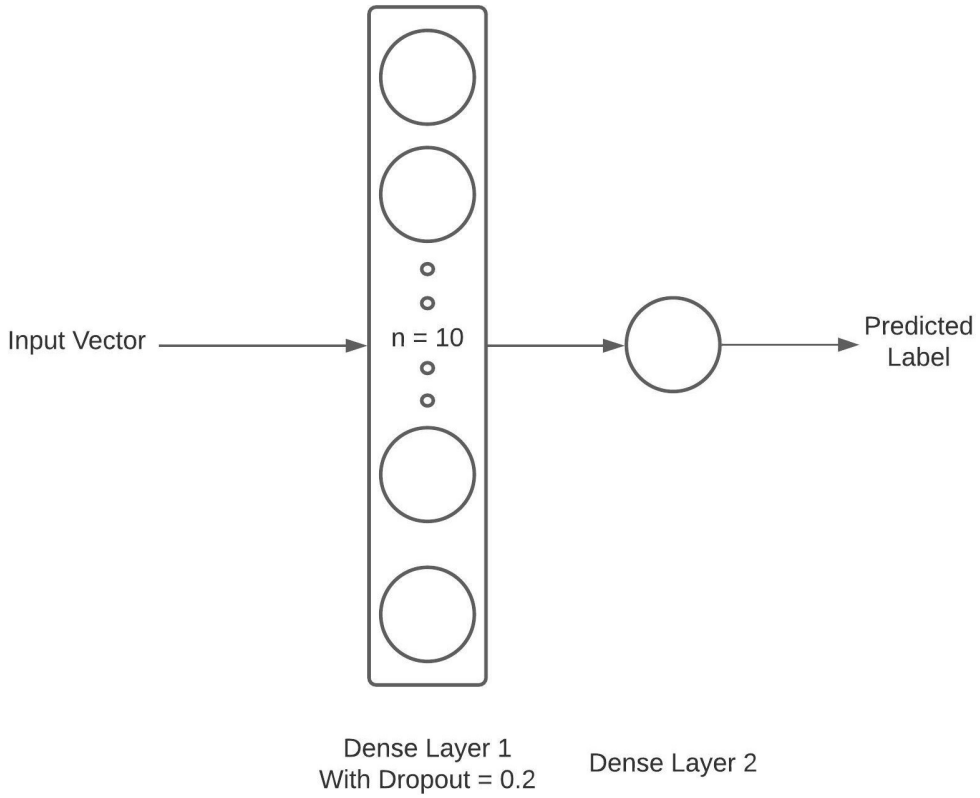

Figure 4.2.1: DNN Network Architecture

As you can see there was a large difference between the results when dropout is introduced, as it drastically reduced overfitting and made our model more robust.

Table 4.2.2: DNN Training Metrics Without Dropout

Data Type	**Accuracy**
Training	91.39%
Test	68.21%

Table 4.2.3: DNN Training Metrics With Dropout

Data Type	**Accuracy**
Training	79.79%
Test	80.64%

The above network was trained for 20 epochs with a batch size of 32.

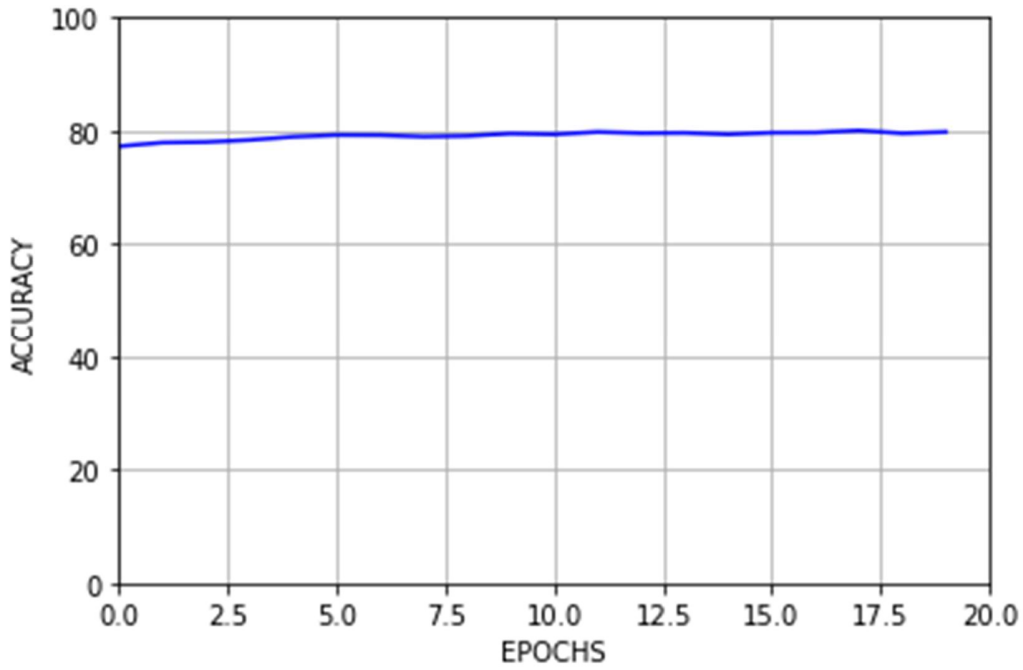

Figure 4.2.2: DNN Accuracy Graph

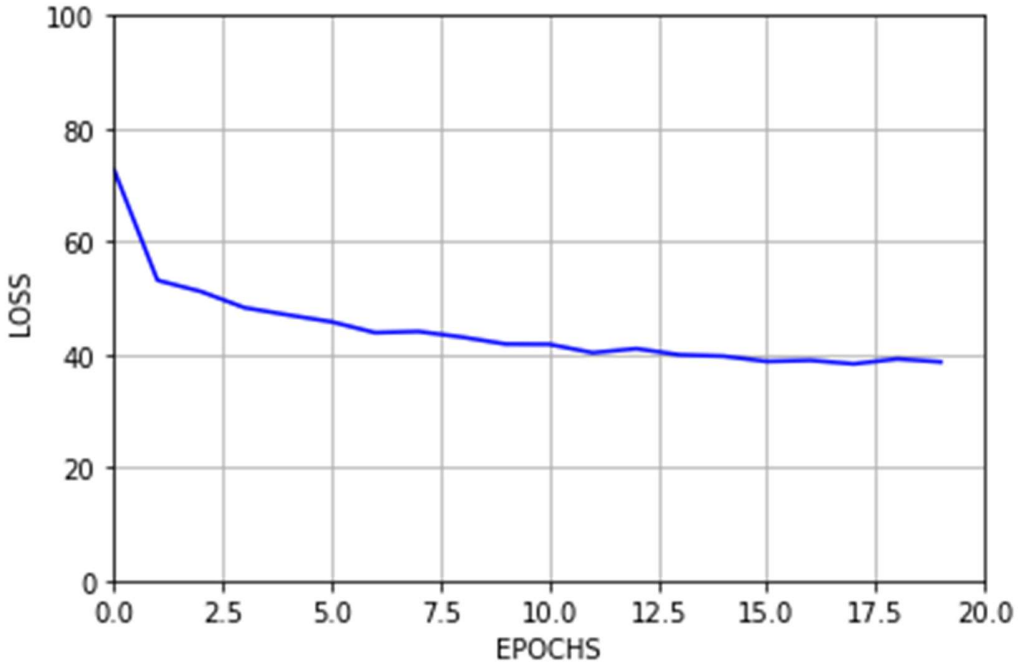

Figure 4.2.3: DNN Loss Graph

The plot clearly shows that due to the simplicity of the model there wasn't much improvement in accuracy over the training period despite the remarkable change in the loss.

5. FUTURE WORK AND CONCLUSIONS

5.1 Gantt Chart

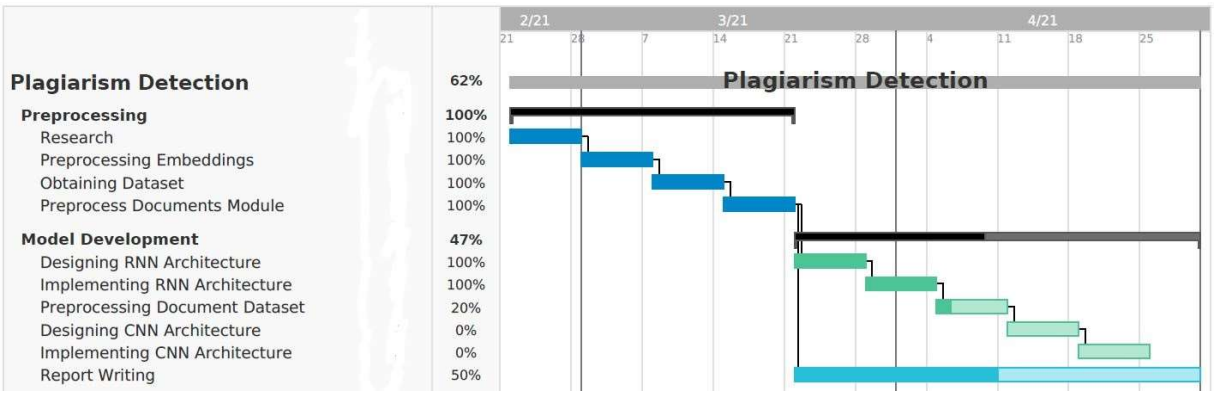

Figure 5.1.1: Gantt Chart

5.2 Conclusion

The project developed in this project is but the beginning. As of now all it can do is detect plagiarism, however I already have plans to expand it further to allow it to recognize the location and degree of plagiarism. I have learnt a lot while making this project and I will continue to work on it in the future. Through this project I hope you can see the difficulty of the issue regarding plagiarism. There is no perfect solution to this problem and all we can do is make it more and more difficult for anyone to take advantage of the lack of manpower to use fraudulent means of achieving academic success.

REFERENCES

[1] El Mustafa HAMBI and Faouzia 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, Pg 110-117

[2] Pantulkar Sravanthi and Dr. B. Srinivasu, “Semantic Similarity Between Sentences”, International Research Journal of Engineering and Technology (IRJET), VOL. 4, ISSUE: 01, Jan 2017, Pg. 156-161

[3] Anupama Nair, Asmita Nair, Gayatri Nair, Pratiksha Prabhu and Prof. Sagar Kulkarni, “Semantic Plagiarism Detection System For English Texts”, International Research Journal of Engineering and Technology (IRJET), VOL. 7, ISSUE: 05, May 2020, Pg. 532-537