**NATURAL LANGUAGE AND PROCESSING**   **DIGITAL ASSIGNMENT**

**TOPIC: NEURAL NETWOKS FOR NAMED ENTITY RECOGNITION**

**TEAM MEMBERS:**

**SHAIK AHMMED BASHA-19BCT0252**

**AMIT KUMAR-19BCE0424**

**SLOT:G2**

**CODE LINK:** <https://colab.research.google.com/drive/11ZtGtIdP_QMzpz7X6JDBargFH3lzFhWj?usp=sharing>

1. **PROBLEM STATEMENT:**

Named Entity Recognition is a popular task in Natural Language Processing (NLP) where an algorithm is used to identify labels at a word level, in a sentence.

The algorithm learns to identify labels from data and predicts them for unseen sentences.

Named entity recognition involves annotating words in a sentence as named entities. More formally, given an input sequence *S*=[*w*1,*w*2,...,*wN*], we predict corresponding labels *Y*=[*y*1,y2,...,*yN*].

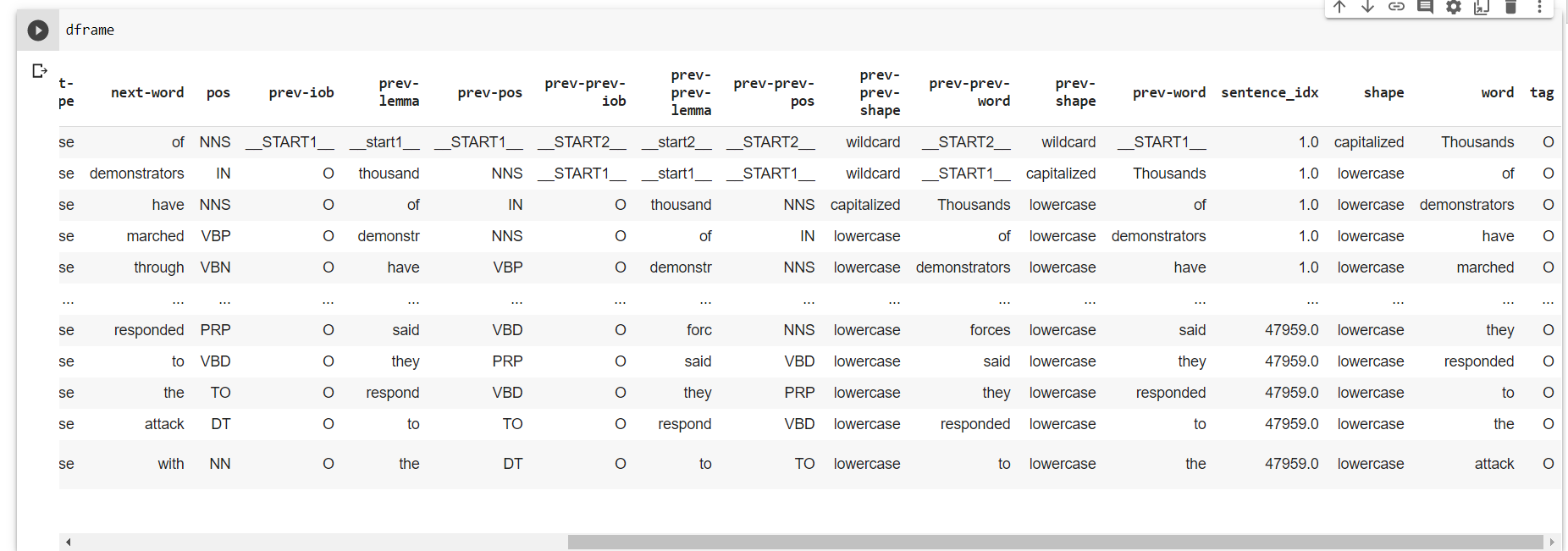
NER becomes very useful when we are handling unstructured data and we want to know about the sentiments of the data or different entity. Also when we are handling biomedical data NER plays a very important role.

NER also works as subproblems of various NLP problem.

1. **UNDERSTANDING DATASETS**

Dataset (ner.csv) has been taken from kaggle.com. Dataset is in csv format containing 140 sentences consisting of 1050795 words. It has 1050795 rows with 25 columns. Columns consist lemma POS etc for each words. After data pre-processing sentences will be divided and each word will be given a sentence index associated with the following sentence number. Every column will be dropped except sentence\_idx word and tag because in our given problem other column of data is not needed for training the model.

RAW DATA

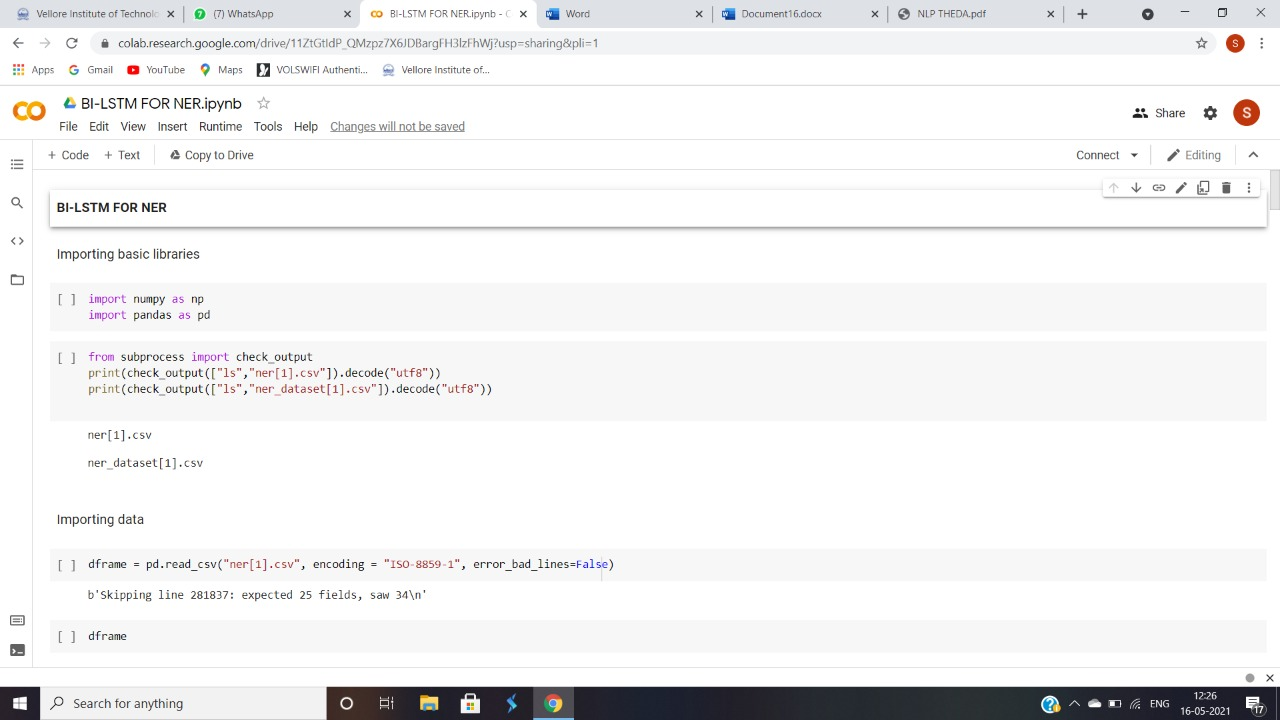


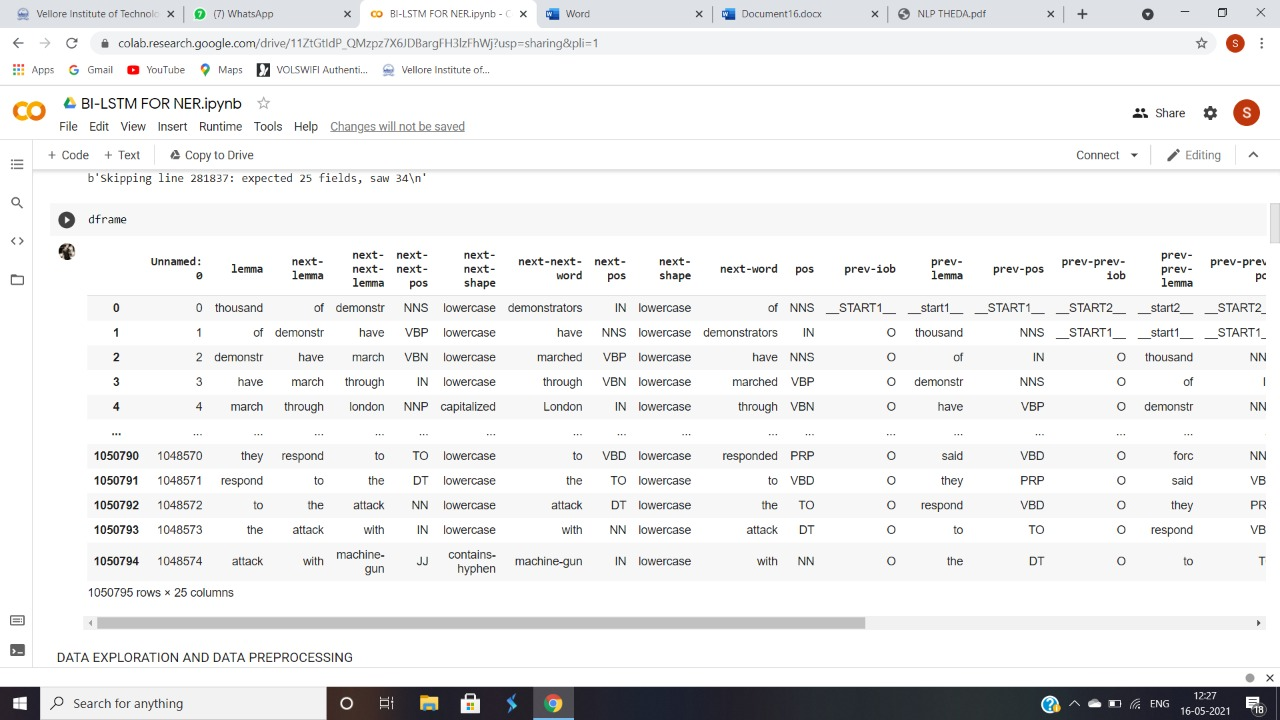
DATA AFTER PR-EPROCESSING WHICH IS GOING TO BE USED FOR TRAINING THE MODEL

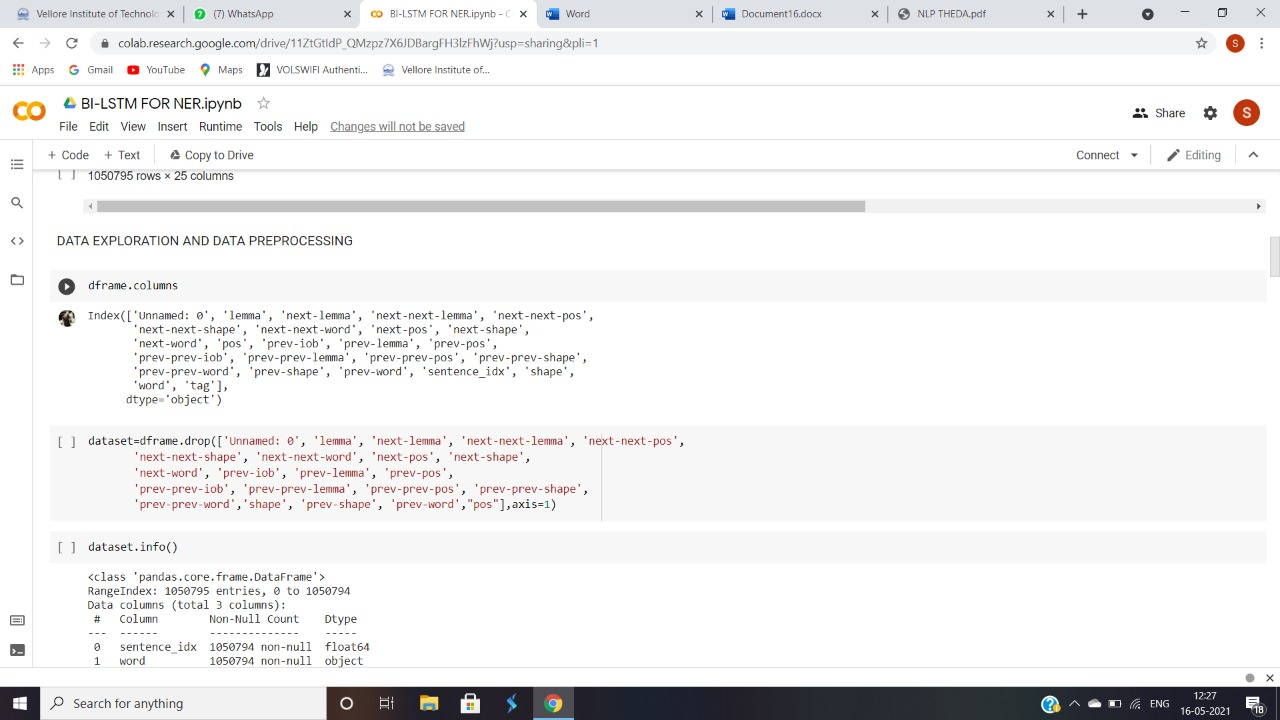


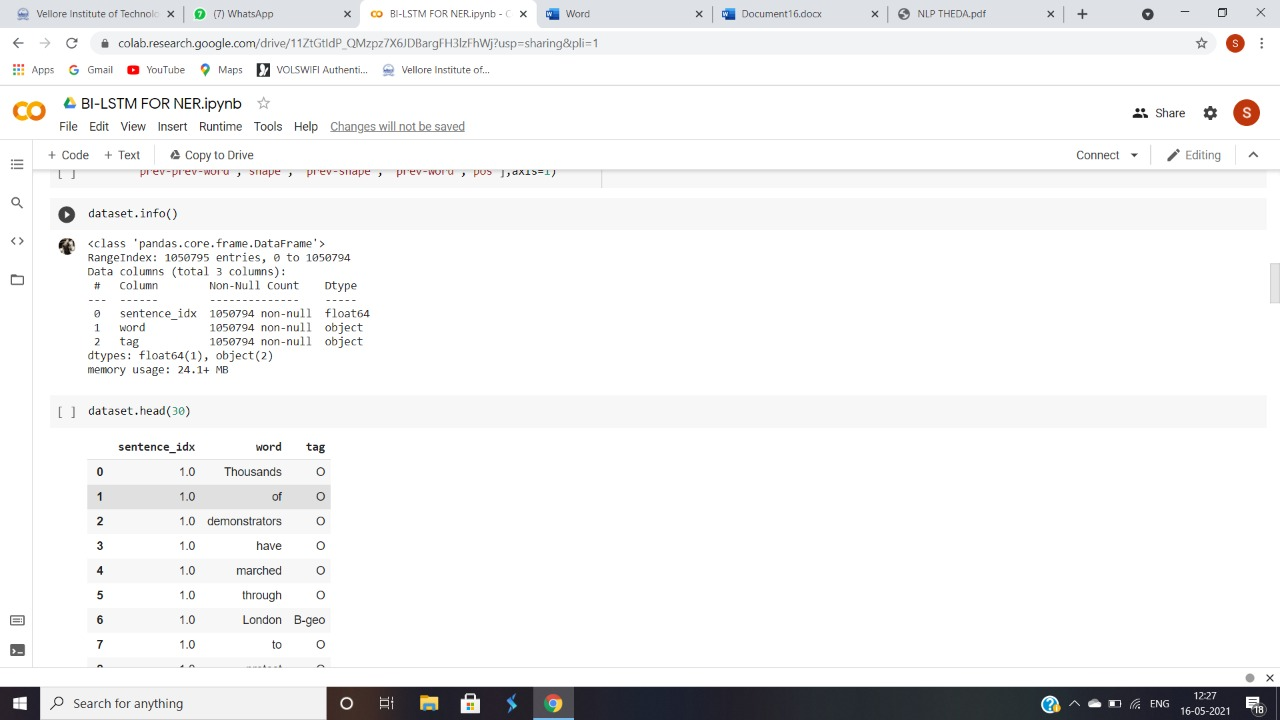
1. **IMPLEMENTATION AND OUTPUT**

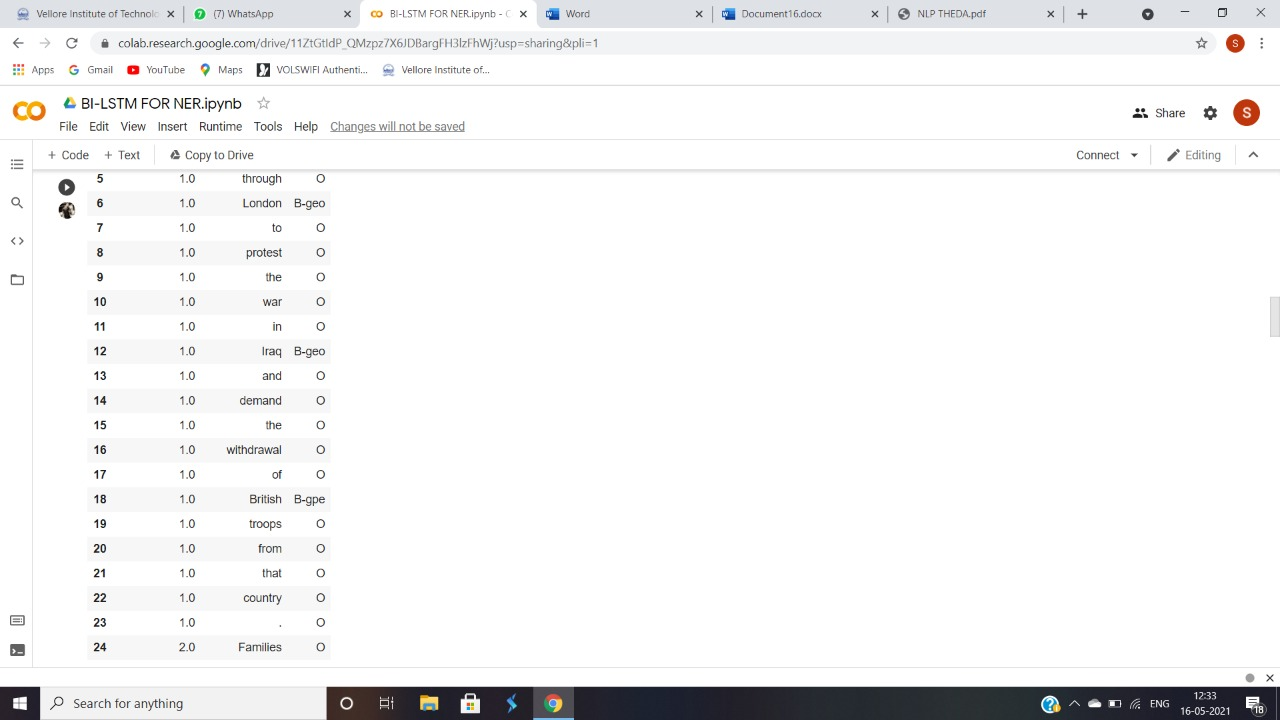
**CODE LINK:** <https://colab.research.google.com/drive/11ZtGtIdP_QMzpz7X6JDBargFH3lzFhWj?usp=sharing>

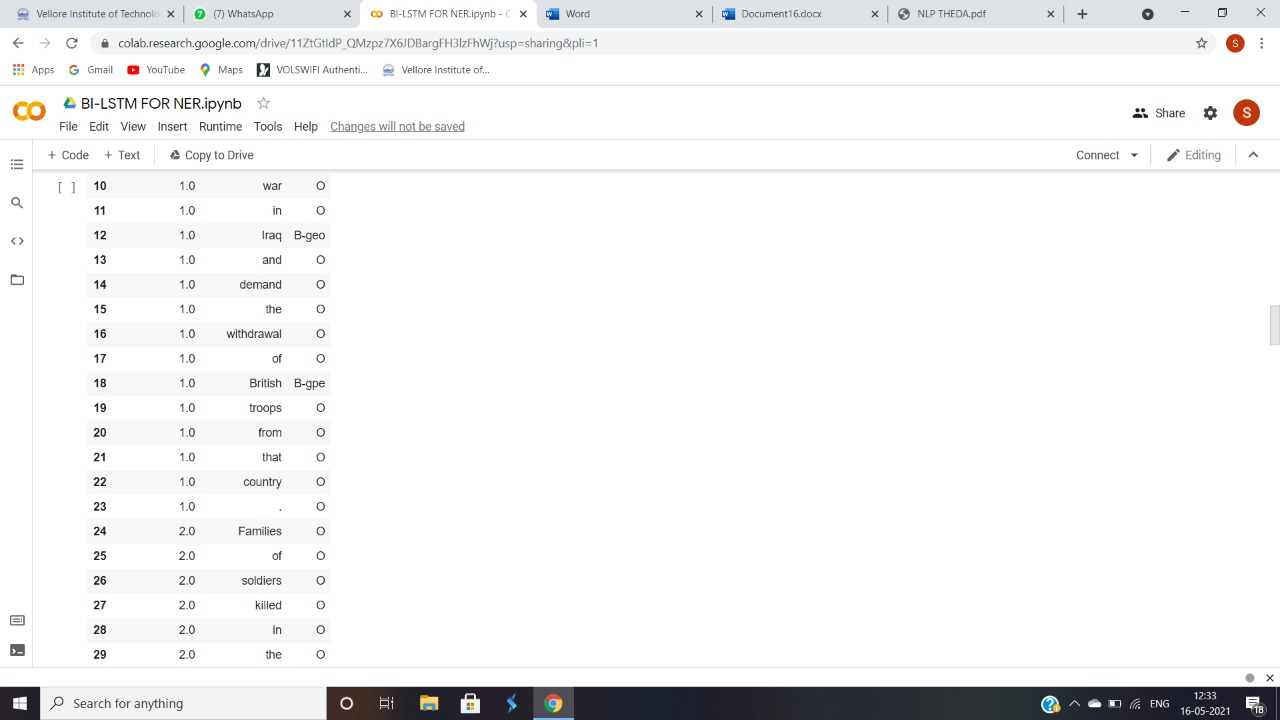


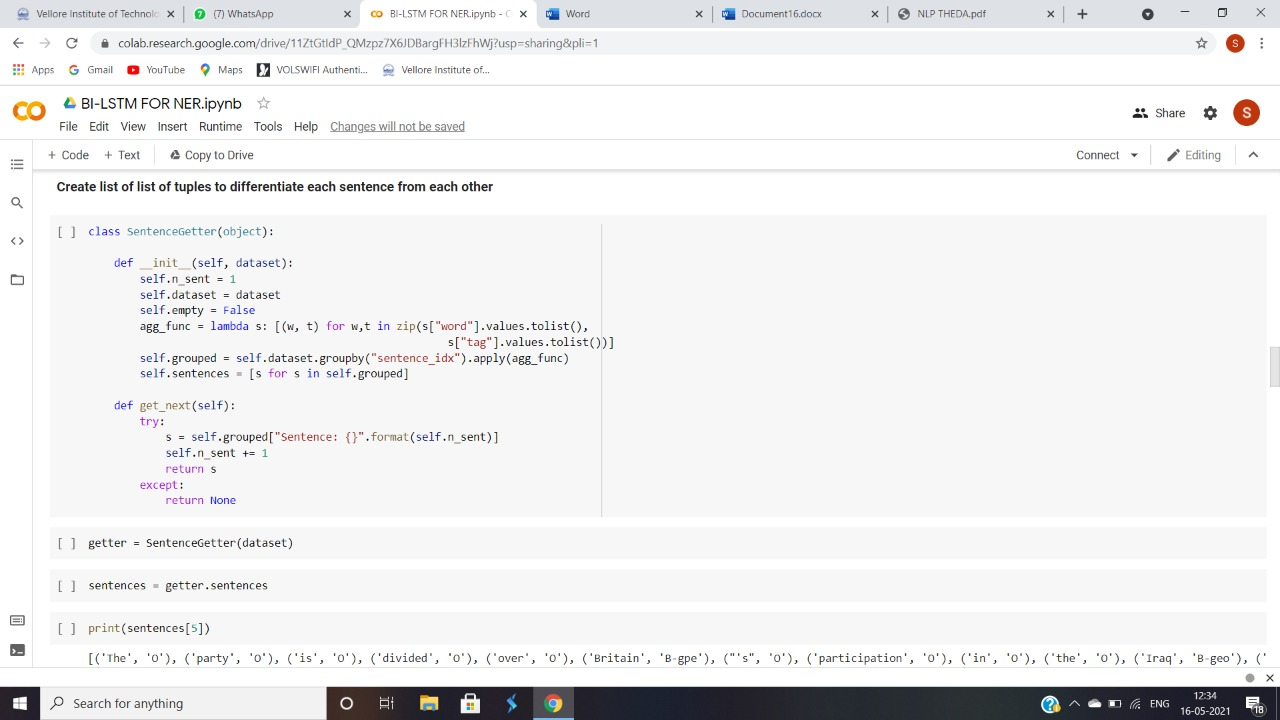


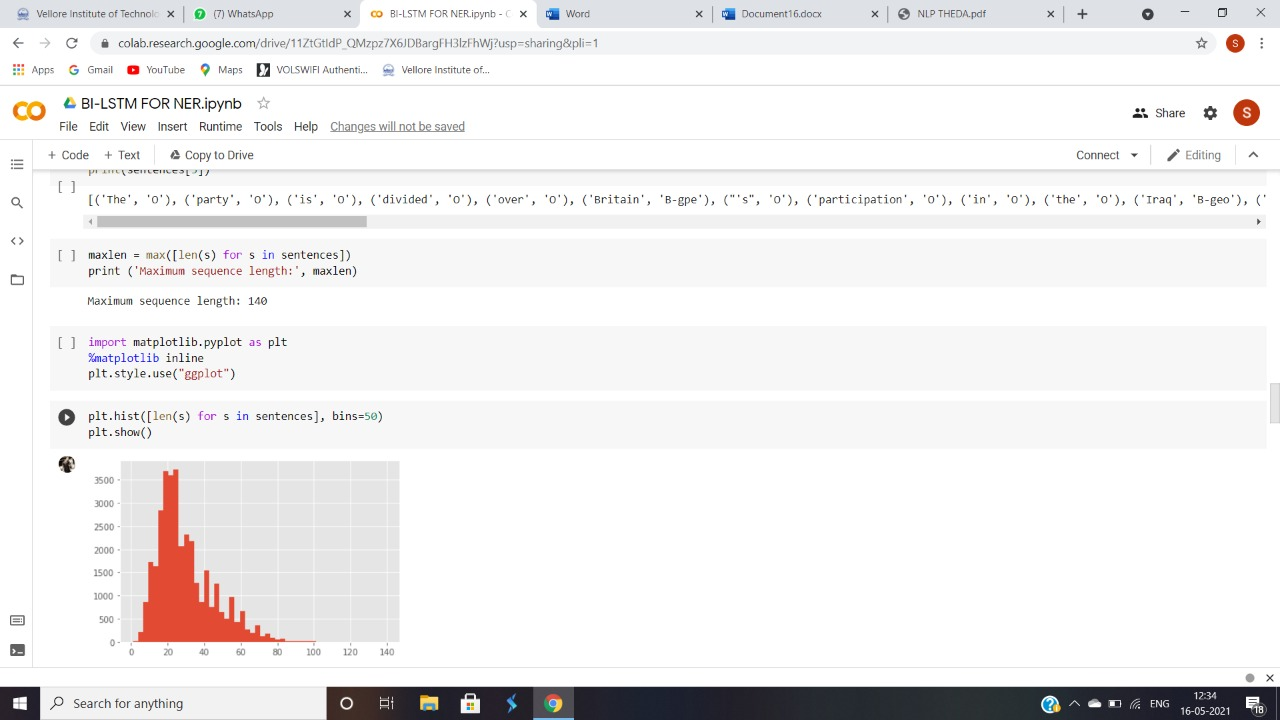


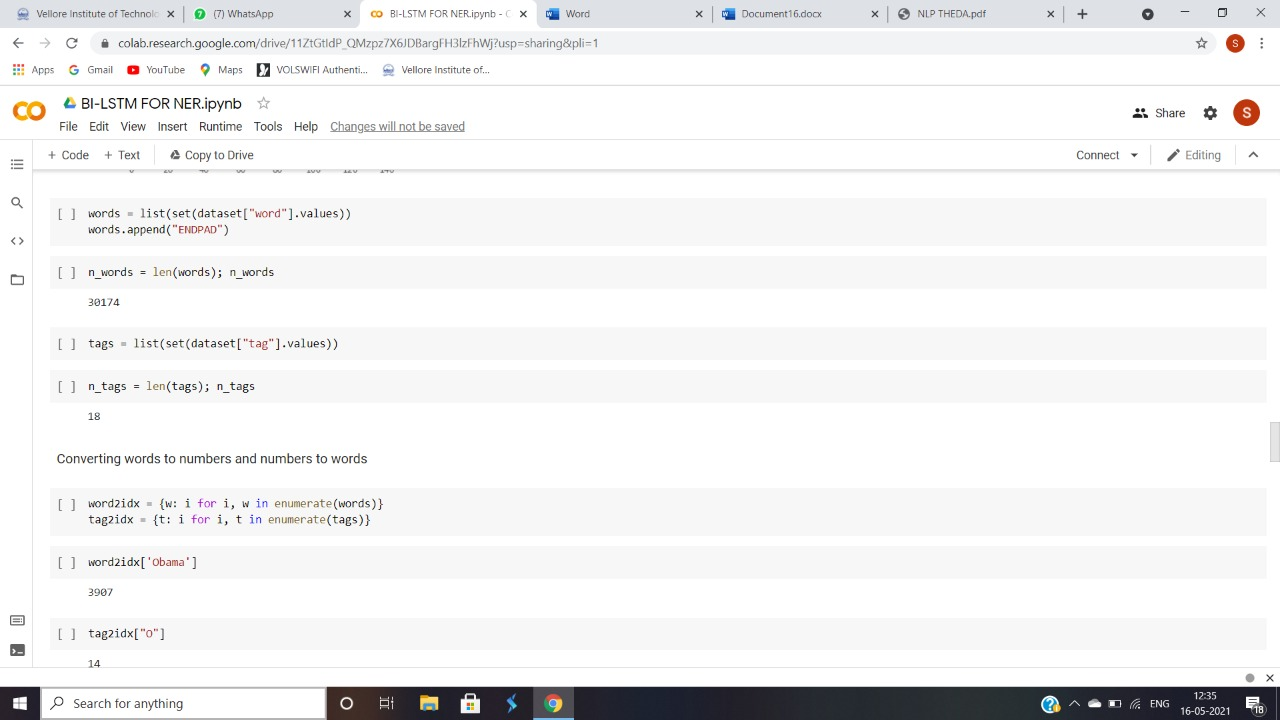




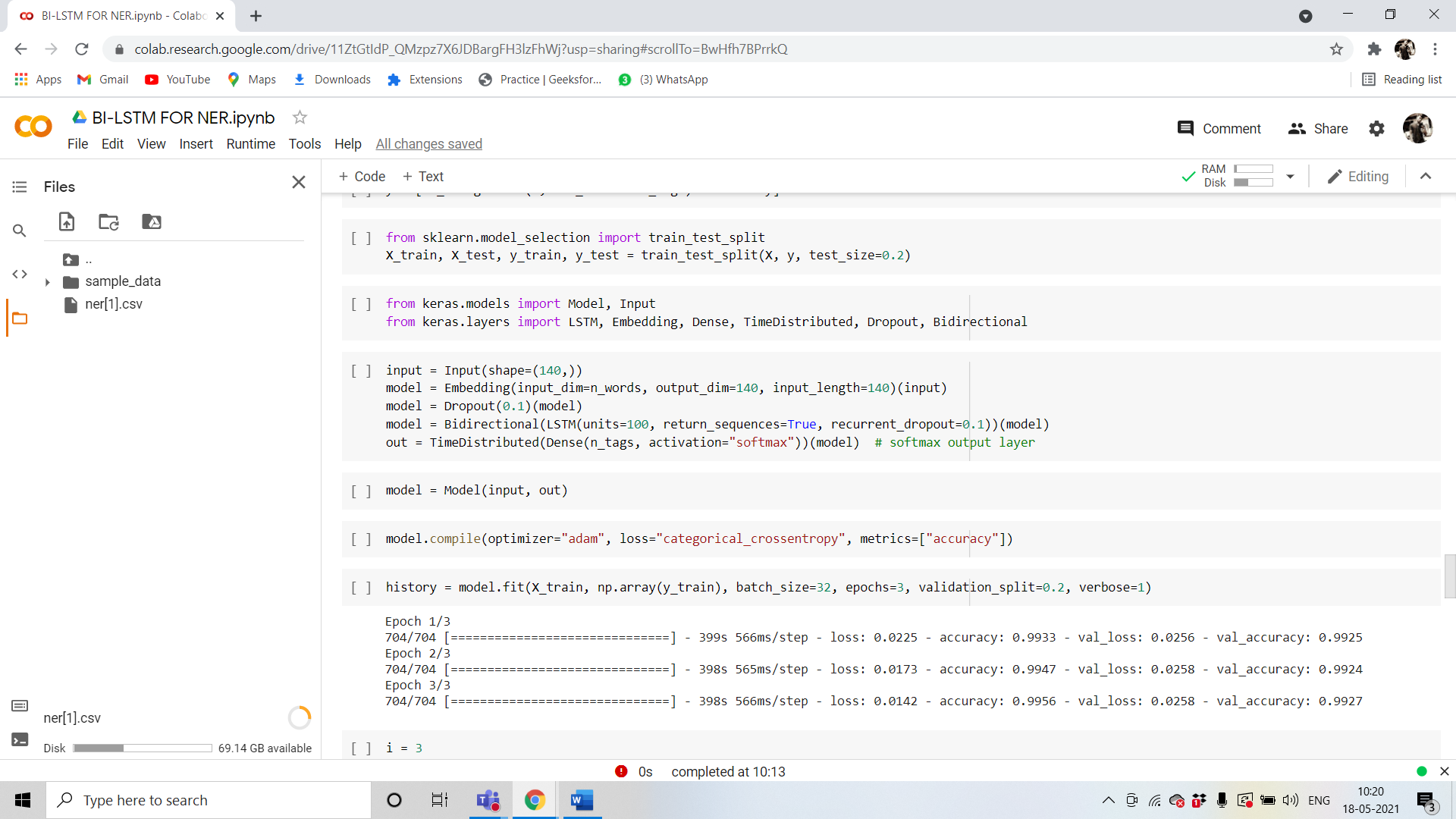








**TRAINING MODEL**



**5.OUTPUT:**

