Chapter 1 Introduction

**Chapter 1**

**Introduction**

* 1. **Description**

In Today's era where the world has never been more connected and informed about the world, It is natural to have a stream of news flowing in rapidly. This news could get overwhelming to people as most of the people tend to read only a particular genre of news. As nowadays smart news generator apps tend to accumulate news from various sources which might not be verified by an editor. It's important to classify this news for better user reach.

* 1. **Problem Formulation**

During pre Election days, BBC News could use a fast and simple system that differentiates between Political and non-political news as it needs to further needs to analyze the sentiment of the general public about various candidates. So given a small Punjabi news article the problem to solve is to classify it into Political and non-Political News.

**1.3 Motivation**

Reading an article that has been extracted from various sources by smart web crawlers and APIs would take on an average of 2-5 min to determine the sentiment of the article. Also, It would be interesting to know what makes an article political. Are there a few special keywords that are most often used in the article that are political etc.

Chapter 1 Introduction

* 1. **Proposed Solution**

As we know for any machine learning algorithm to perform well it needs a large and good quality of data. Moreover, for NLP tasks we need a good Language model that understands the semantics of the words in the language used in a particular context. English is the most common language worldwide and hence there are some powerful language models built on billions of textual data by various tech giants.

Punjabi is a regional language in India that is not very popular Internationally and there has not been much work on this language and hence there is no pre-existing state of the art language model.

We will be using a Text corpus of Punjabi language extracted from Wikipedia articles to build our Punjabi Language model.

We will be using a BBC new article dataset from Kaggle to use the Punjabi language model to classify news articles.

**1.5 Scope of The Project**

As we are building a very decent Language model in this project it can be utilized to various other tasks in the Punjabi language.

Also, the BBC news articles classifier can be extended for multiple other categories such as sports, finance, etc.

Chapter 2 Review Of Literature

**Chapter 2**

**Review of Literature**

Processing textual information involves passing it to various pre-processing phases.

Stop-word elimination is one such sub-phase. In this paper, the author has identified 184 stop words. This paper provides a base for others in this community to develop a better list for stopwords elimination. We are using one such work done in this community which identifies 463 stopwords.[1]

Chapter 3 System Analysis

**Chapter 3**

**System Analysis**

**3.1 Functional Requirements**

**Interface Requirements:**

1. Field accepts one Textual Input in Punjabi language.

**Business Requirements:**

1. Data must be entered before a request is submitted.
2. Clicking the submit button must request to the Approval Workflow.

**3.2 Non Functional Requirements**

**3.2.1 Performance**

The computer running the software must have a powerful CPU and GPU so that the data can be processed faster and there won’t be lag in the entire process of clustering the financial news articles.

**3.2.2 Reliability**

The system takes in the inputs without any error and predict the class accurately so that the users of this system can rely on the system for the classifying new articles.

**3.2.3 Usability**

The system is easy to handle and navigates in the most expected way with no delays. The system interacts with the user in a very friendly manner making it easier for the users to use the system.

Chapter 3 System Analysis

**3.3 Specific Requirements**

**3.3.2 Hardware Requirements**

* CPU Type : Intel i5 or above
* Clock speed : 2.1GHz
* Ram size : 8GB and above
* Hard Disk capacity : 100GB and above
* Working keyboard

**3.3.2 Software Requirements**

* Operating System : Any
* Python 3.5 or above

**3.3.3 Communication Interfaces**

This system will be completely based on a local system of the user.

Chapter 3 System Analysis

**3.4 Use-Case Diagrams and Description**

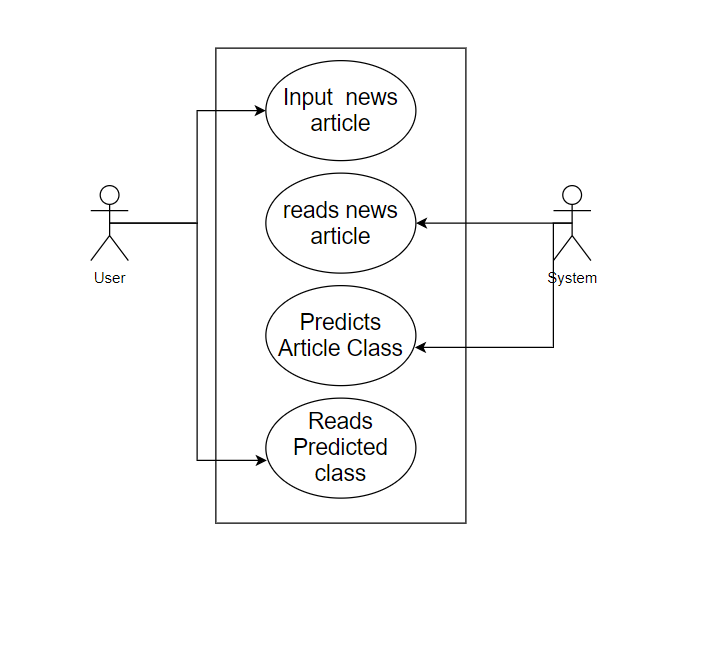


Fig 1. Use Case Diagram for Classifying of BBC News Articles.

User:

Methods:

1. Enter\_News\_Article()

System

Methods:

1. Read\_News\_article()
2. Classify()
3. Display()

Chapter 4 Analysis Modeling

**Chapter 4**

**Analysis Modeling**

**4.1 Class Diagram and Activity Diagram**

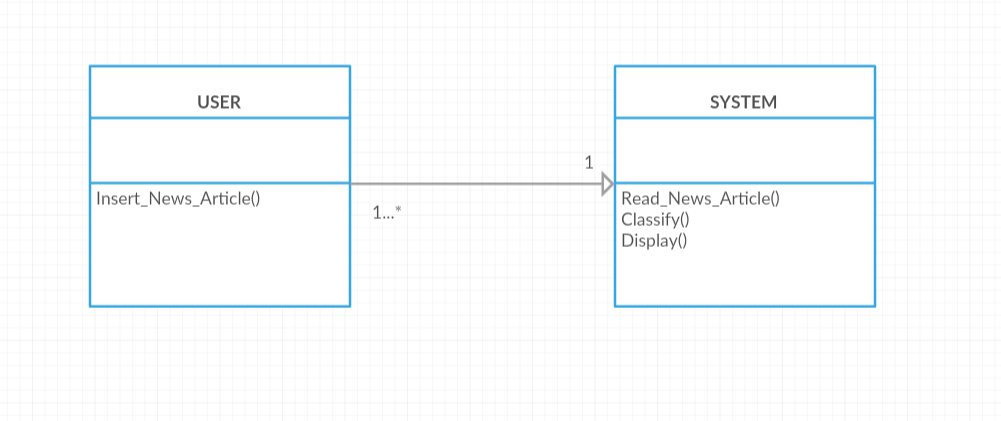


Fig 2. Class Diagram for Classifying BBC News Articles.

Class: User

Methods:

1. Enter\_News\_Article()

Class: System

Methods:

1. Read\_News\_article()
2. Classify()
3. Display()

Chapter 4 Analysis Modeling

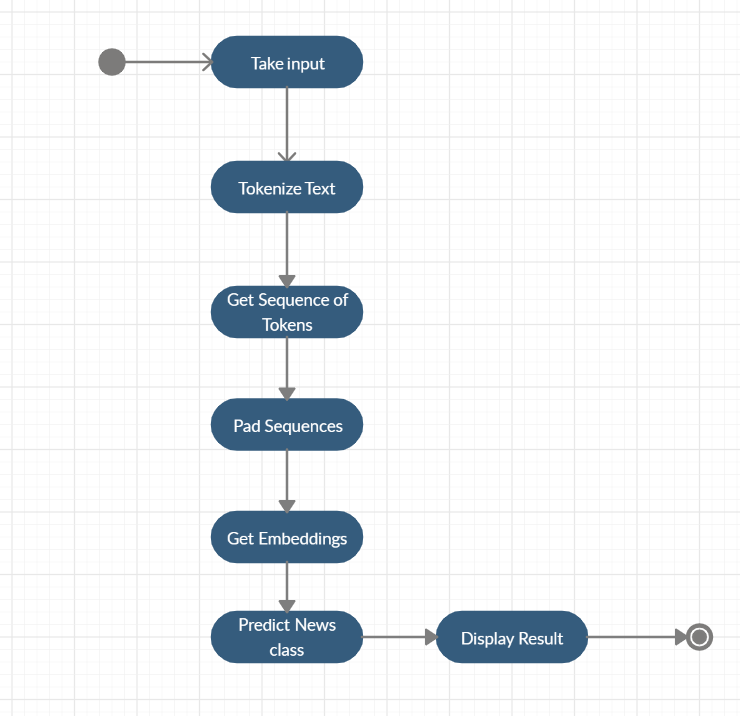
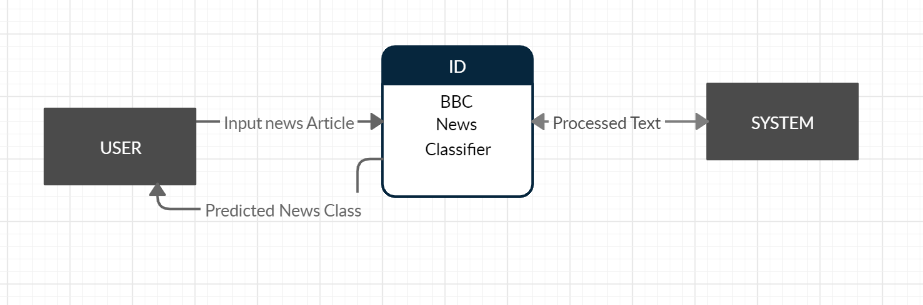


Fig 3. Activity Diagram for Classifying BBC News Articles.

Chapter 4 Analysis Modeling

**4.2 Functional Modelling**

**Data Flow Diagram**



Contextual Level

Fig 3. Data flow Diagram for Classifying BBC News Articles.

Chapter 5 Design

**Chapter 5**

**Design**

**5.1 Architectural Design**

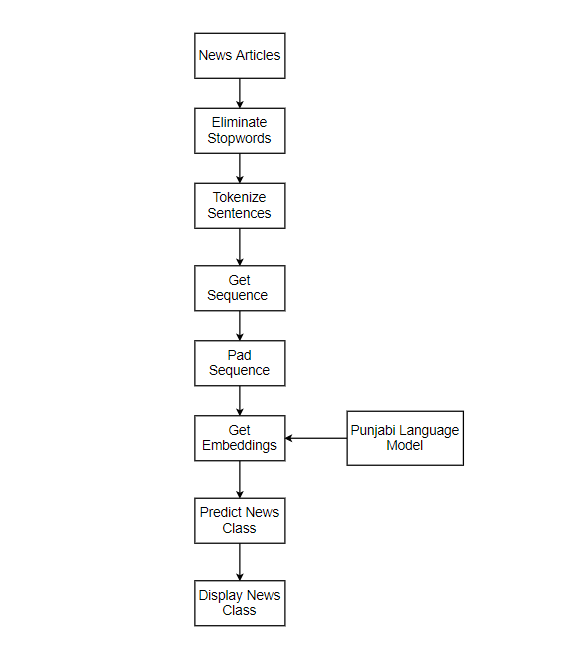


Fig 7. Architecture for Classifying BBC News Articles.

Chapter 5 Design

**5.1.1 News Articles**

Suitable dataset with headline and content is Inserted.

**5.1.2 Data Preprocessing**

In this step, we removed the unwanted data. Also checked, if there were any null values present in the dataset as these articles were taken from e-news.

**5.1.3 Data Transformation**

As our dataset contained textual data, we had to convert it into a numerical form where the features were easily understood.

Certain Steps followed were-

--Removal of Stopwords:

Stopwords are the words that do not play a major role in identifying the text Stopwords contained pronouns,articles and conjunctions which were removed.

--Removal of Punctuation: Punctuations were also removed as they did not pass any significant message.

--Natural Language Processing:

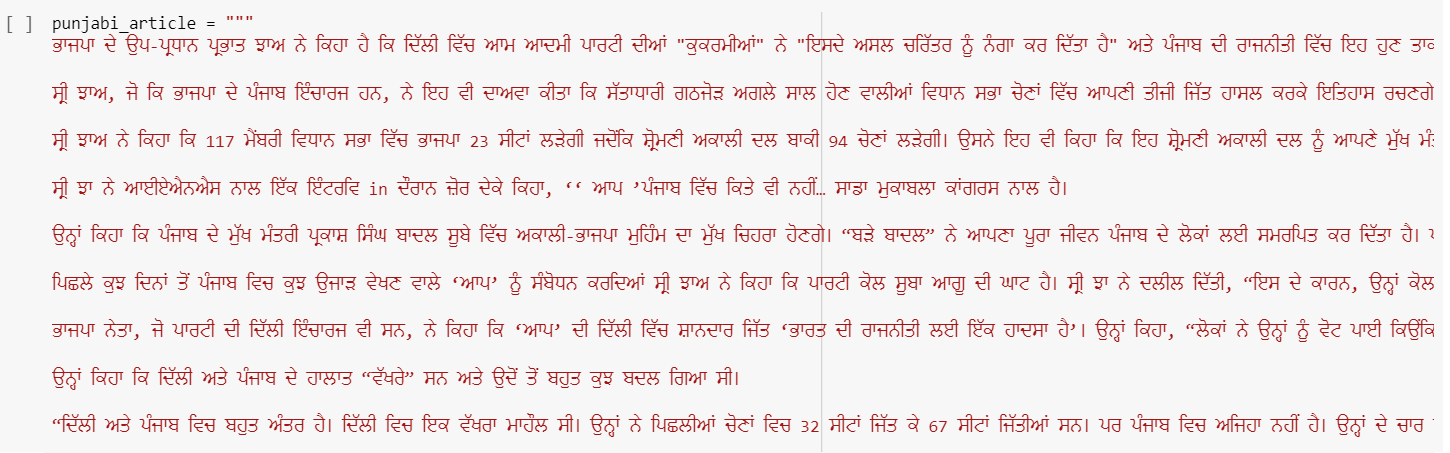
a) Tokenization - Tokenization describes splitting paragraphs into sentences, or sentences into individual words

b) Padding – In recent times Deep Learning Algorithm with fixed size Input performs better and hence Padding helps achieve the sequence to be of a fixed length.

Chapter 5 Design

**5.2 User Interface Design**

User Input:



Visualize Keywords in 3D:



Chapter 6 Implementation

**Chapter 6**

**Implementation**

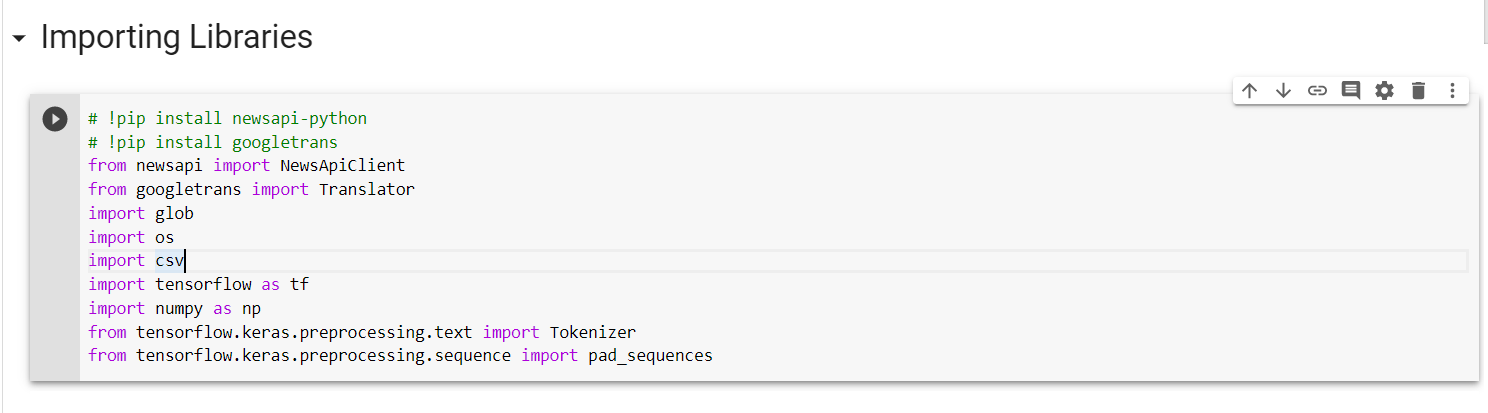
**6.1 Algorithms Used**

1. **Deep Neural Network(DNN)**

* Takes input a Fixed Size Sequence of vector.
* Learn Trainable parameters of the Neural network during the training process by using Adam optimizer and Sparse categorical crossentropy to calculate loss in the training purpose.

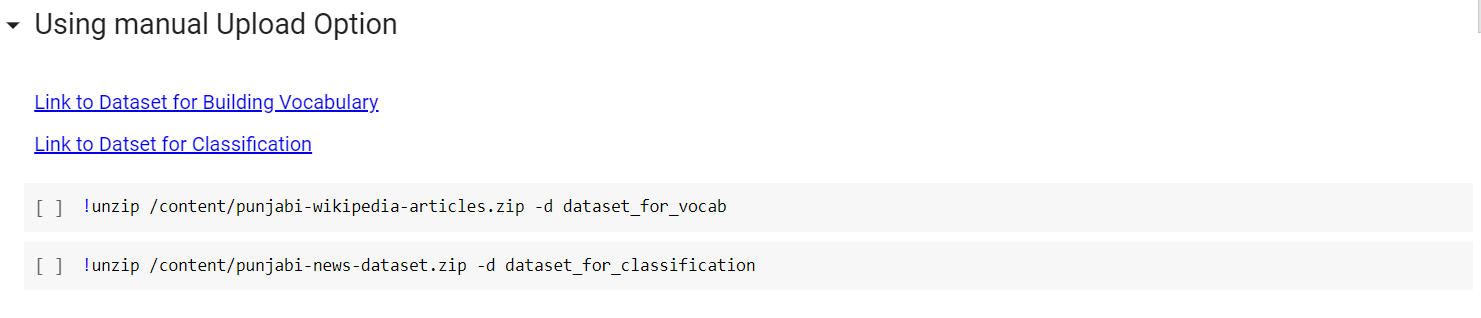
**6.2 Working of the project**

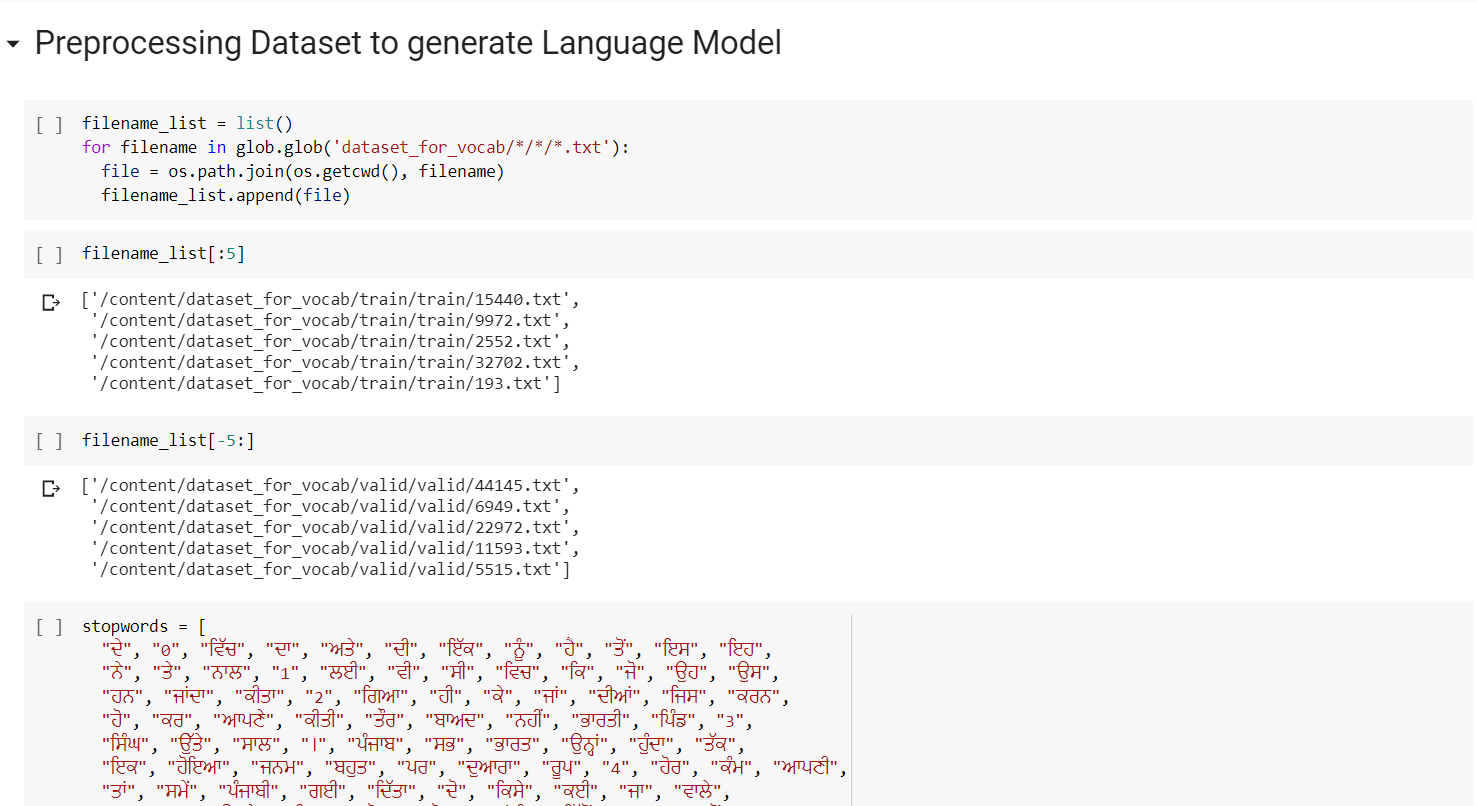
**CODE SNIPPETS**

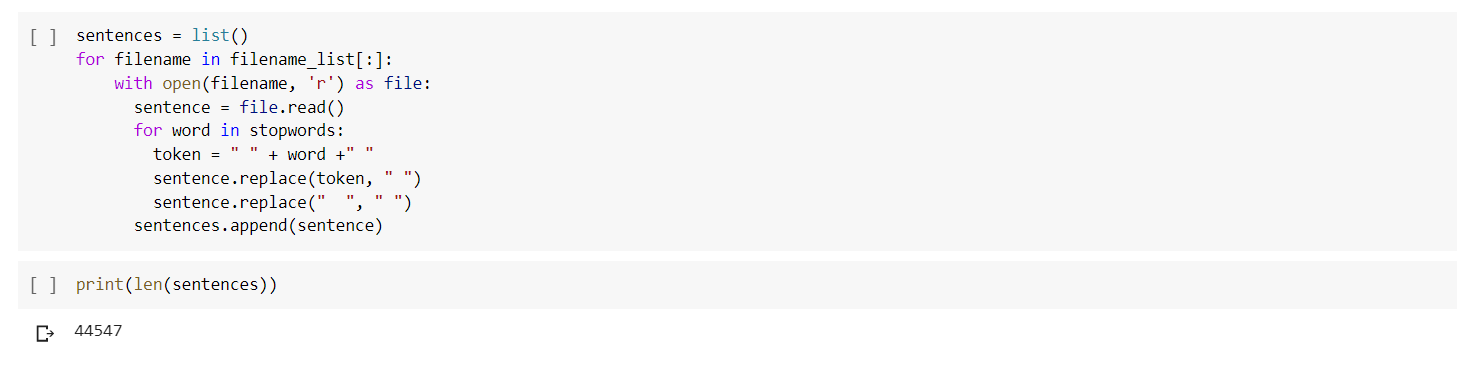


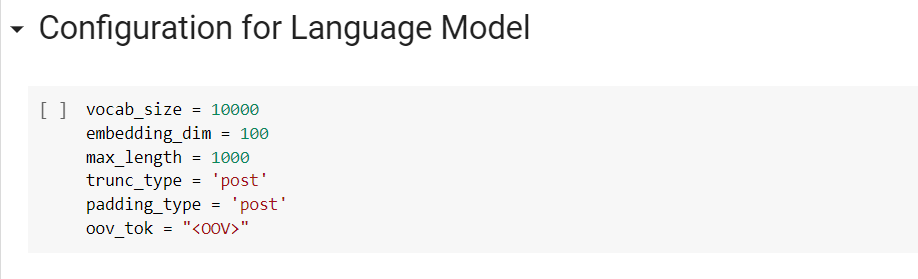
Importing Dataset



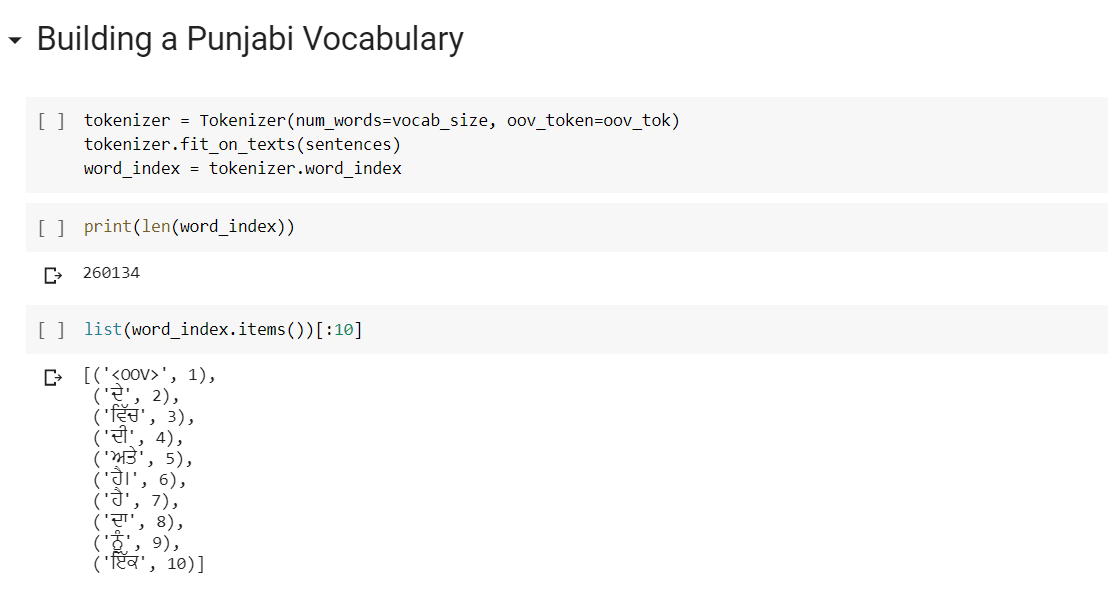


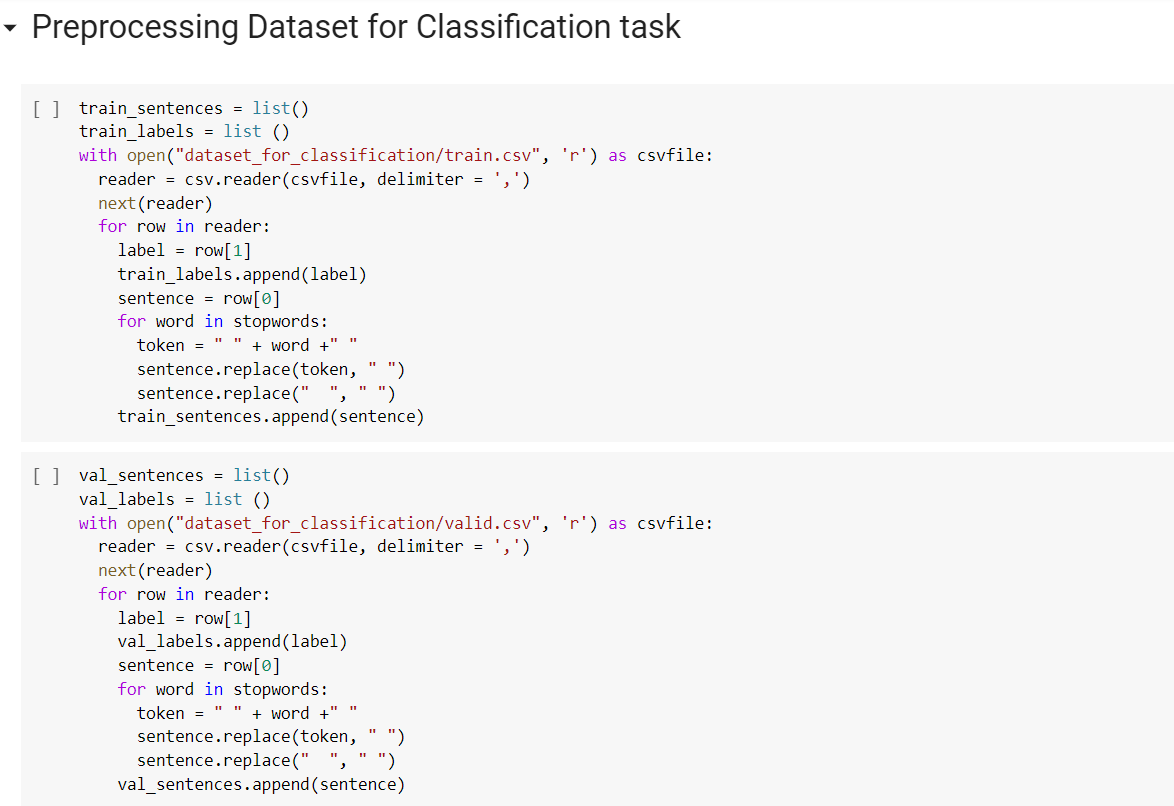




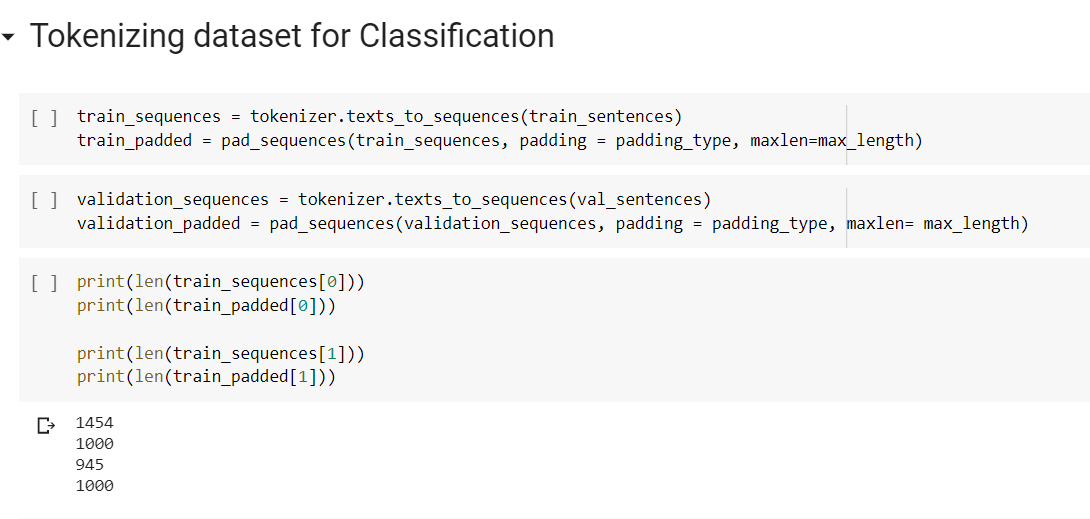


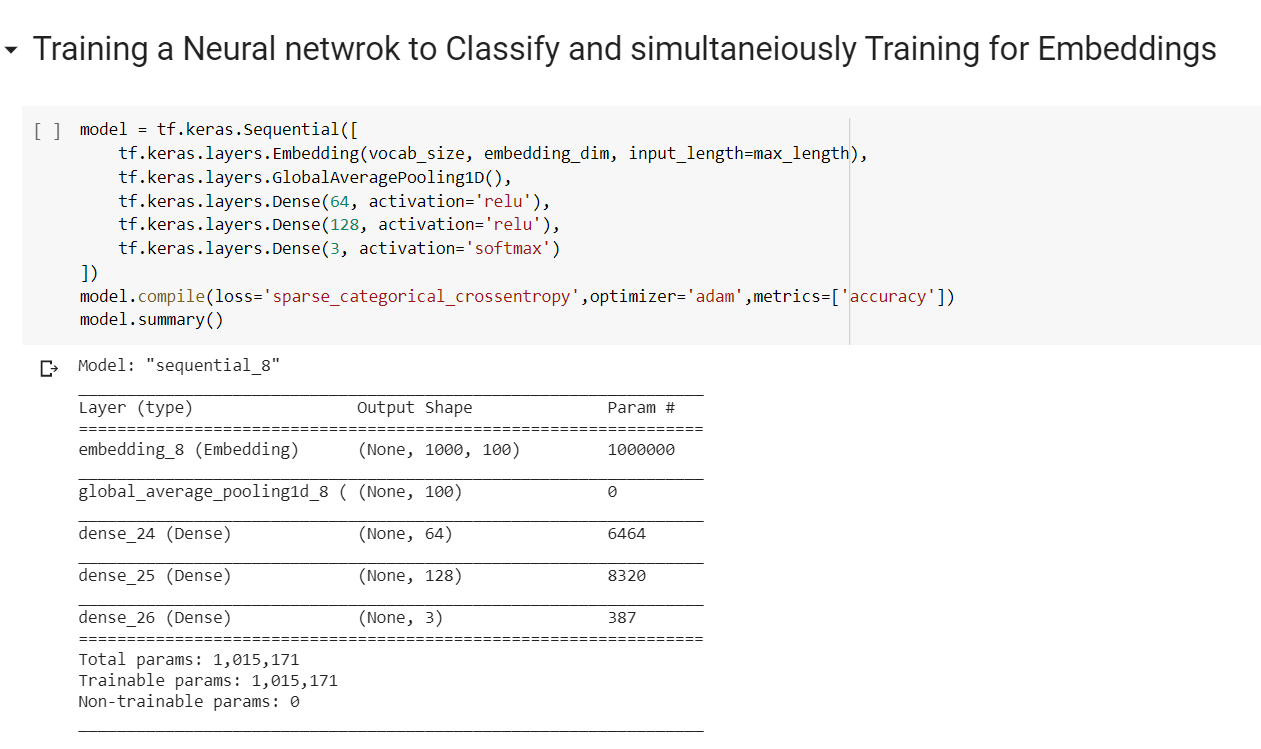
Chapter 6 Implementation





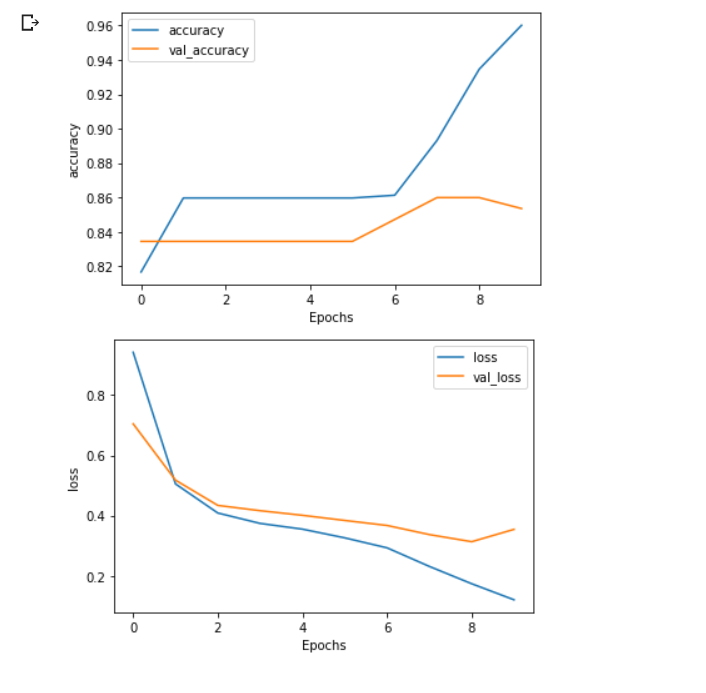
Chapter 6 Implementation





Chapter 6 Implementation

# Visualizing Training Process



Chapter 6 Implementation



Chapter 6 Implementation



Chapter 7 Conclusion

**Chapter 7**

**Conclusion**

The main requirement of Classifying news articles in Punjabi Language is the data preprocessing and building a Language model. Building a good Language model helps a getting a better accuracy. For a Textual non English NLP task such as classification of articles require us to build a Language model on our own as there is no pre trained model that we can use in most cases. We conclude that on having a good enough language model for a particular language it can be used to solve any NLP tasks using transfer learning and other methods.

**References**

[1] Kaur, Jasleen & Saini, Jatinderkumar. (2016). Punjabi Stop Words: A Gurmukhi, Shahmukhi and Roman Scripted Chronicle. 32–37. 10.1145/2909067.2909073.

**Acknowledgements**

We take the opportunity to thank all those people who have helped and guided us through this

project and make this experience worthwhile for us. We wish to sincerely thank our reverend

**Bro. Jose Thuruthiyil** and principal **Dr. Sincy George** for giving us this opportunity for

making a project in Final Year of Engineering. We would also like to thank HOD of Computer

department **Dr. Kavita Sonawane** and all teaching and non-teaching staff for their immense

support and cooperation.

Last but not the least we would like to thank **Mrs.** **Vincy Joseph** for guiding us throughout

the project and encouraging us to explore in this domain.