

High Level Design Document(HLD)

BBC News Articles Sorting

Revision Number: 1.4

Last date of revision: 24/07/2022

Rushikesh Darge

Contents

Abstract	5
Why this High-Level Design Document?	6
Scope	6
General Description	7
Product Perspective	7
Technical Requirements	7
Constraints	8
Design Details	11
Process Flow	11
LSTM Model	11
Deployment Process	12
Performance	13
Reusability	13
Application Compatibility	13
Deployment	13
References	14

Abstract

Recent trends are to build tall buildings in big cities as a way out of the current housing overpopulation problem. These new structures unveil problems that if not addressed in time could cause catastrophes of unimaginable impact. Some of those problems are the incidence of a fire threat happening upstairs in one of those buildings, medical emergencies due to any road accidents or mob that may cause threat to the human kind. This work discusses the implementation of the unmanned ground vehicles to spot the real location of the medical emergencies due to road mishap, mob or illegal activities such as hooliganism, snatching, robbery and the fire emergency and accordingly channelize or route them to the concerned helpline for quick mitigation and avoid disaster.

1 Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
 - o Security
 - o Reliability
 - o Maintainability
 - o Portability
 - o Reusability
 - o Application compatibility
 - o Resource utilization
 - o Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

1.3 Definitions

Term	Description
TF-IDF	Method to convert text into numbers Term Frequency - Inverse Document Frequency
AWS	Amazon Web Services
LSTM	It is a variety of RNN models, Long Short Term Memory Network
RNN	Deep neural network architecture Recurrent neural network which can be use in text or time series data

General Description

Product Perspective

The Model to sort news to his category reduces time to sorting images or we can also this is for sorting other text data.

Problem statement

To create an AI solution for separating each news item between categories.

Proposed Solution

The solution proposed here is to create an api or website that segeratgate the given news text to his specific categories and give the correct categories to the user.

Further Improvements

This model can be extended to other categories of news like religious news, locality news, etc., for that we require more data. We can also use the same model to segregate the type of articles that are available on the internet or to separate the books categories.

Technical Requirements

To complete this project we required Internet connection, Laptop/ Desktop, GPU to train deep learning model, Python, librairies like Pandas, numpy, Tensorflow, scikit learn etc.

Data Requirements

Data required for this is text data, we need to scrap the news from any news website and also need to store that news category. More data, more accuracy of our model.

Tools used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, TensorFlow, Keras and Roboflow are used to build the whole model.



- Google colab used as IDE.
- For visualization of the plots, Matplotlib, Seaborn and Plotly are used.
- AWS is used for deployment of the model.
- Pandas is used to read, insert, delete, and update the CSV file.
- Front end development is done using HTML/CSS.

Hardware Requirements

- Laptop / Desktop
- GPU
- Multicore CPU
- Disk Storage

Constraints

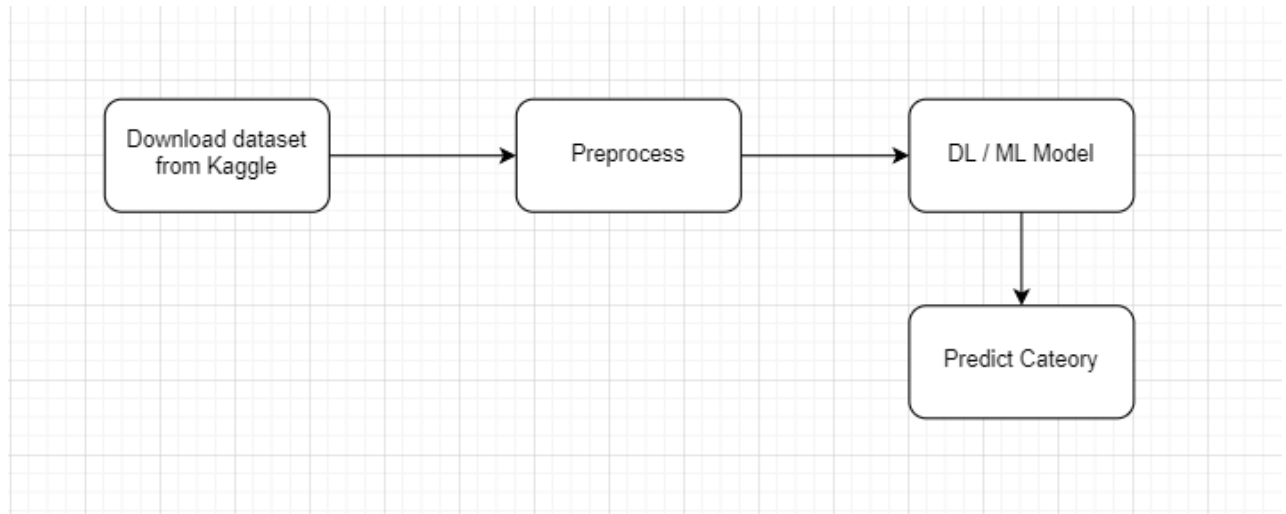
Our model should give higher accuracy as well as give output in less time, which means low latency.

Design Details

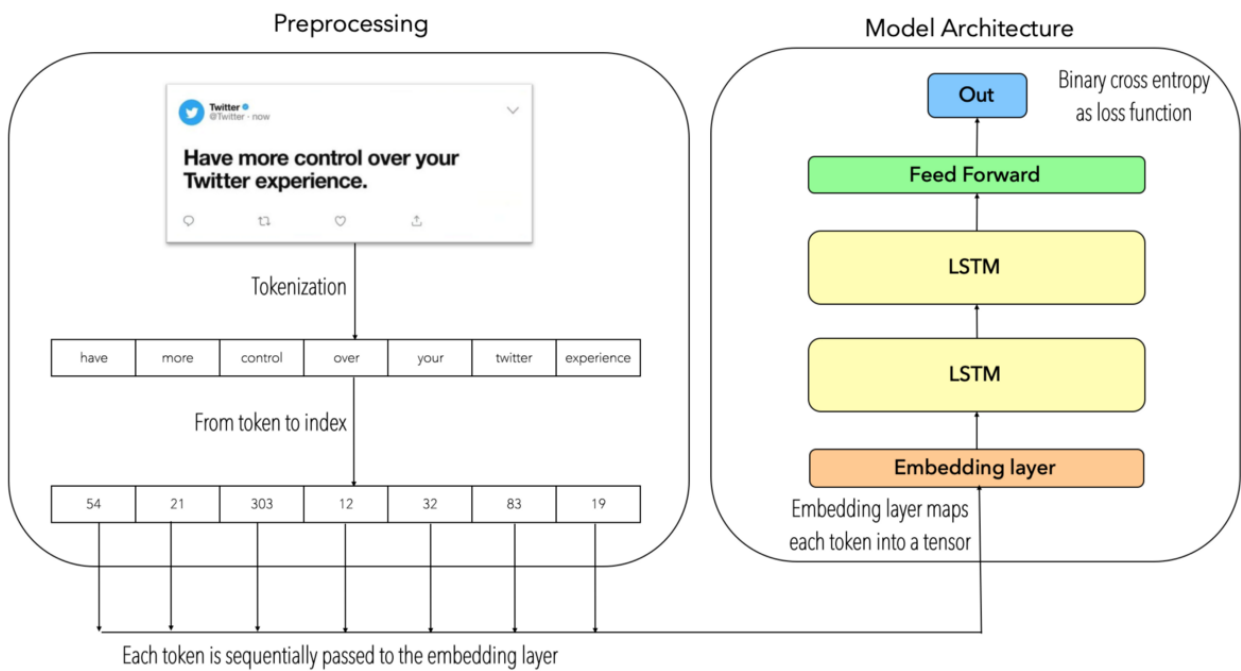
Process Flow

For identifying the different types of anomalies, we will use a deep learning base model. Below is the process flow diagram as shown below.

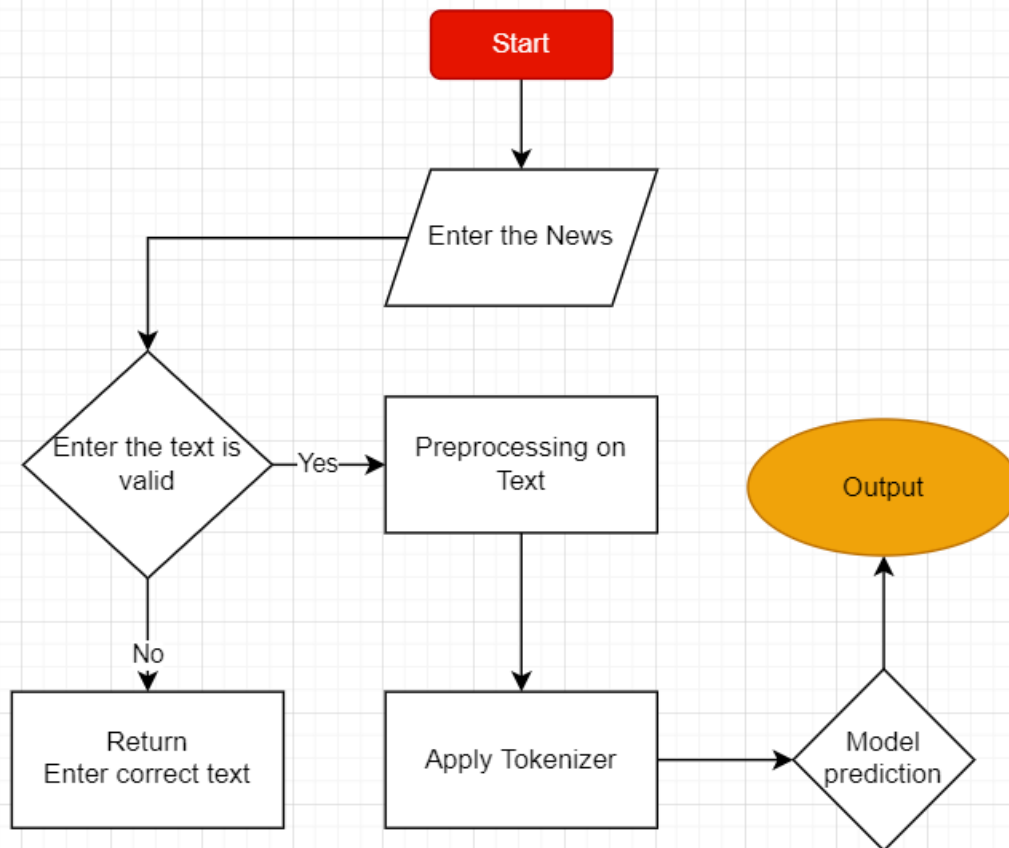
Proposed methodology



LSTM Model



Deployment Process



Performance

Performance we can calculate using speed of model prediction and accuracy of our model.

Reusability

The code written and the components used should have the ability to be reused with no problems.

Application Compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

Deployment



References

1. <https://colab.research.google.com/#create=true>
2. <https://streamlit.io/cloud>
3. https://www.tensorflow.org/api_docs/
4. <https://app.diagrams.net/>