Low Level Design

News Article Sorting

Written By	Abhishek Nayak, Biswajeet Padhi
Document Version	0.2
Last Revised Date	20/11/2022

Document Control

Change Record:

Version	Date	Author	Comments
0.1	19 – May - 2021	Abhishek Nayak	Introduction & Architecture defined
0.2	20 – May - 2021	Biswajeet Padhi	Architecture & Architecture Description appended and updated

Approval Status:

Version	Review Date	Reviewed By	Approved By	Comments

Contents

1. Introduction

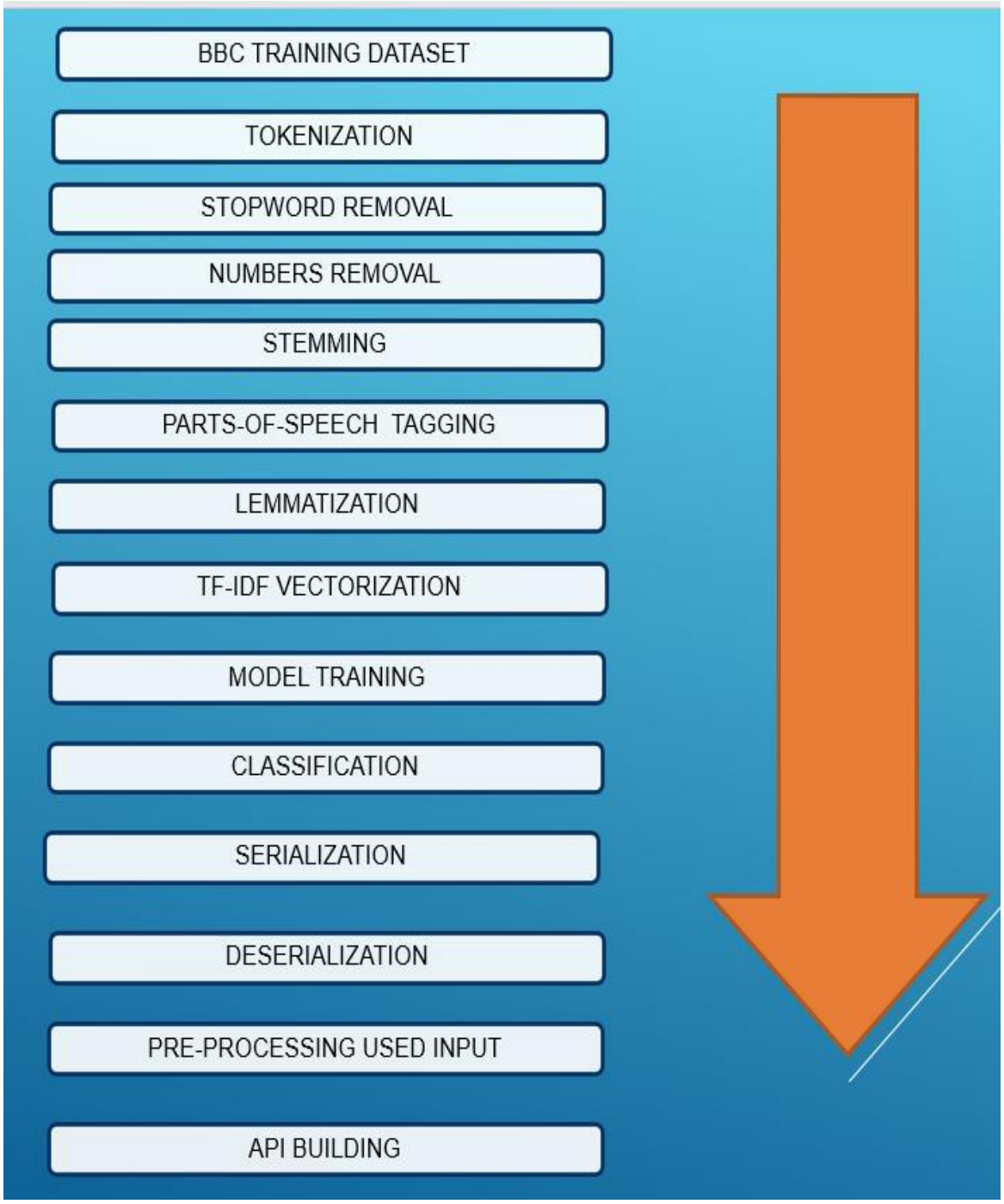
1.1. What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Food Recommendation System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

1.2. Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

2. Architecture



3. Architecture Description

3.1. Data Description

- The Dataset is a public dataset from the BBC comprised of 2225 articles, each labeled under one of 5 categories: business, entertainment, politics, sport or tech.
- The Dataset has 3 columns.
- Column 1 is the Article id
- Column 2 is the Article Text
- Column 3 is the Category
- The dataset for training is in csv format with 1490 rows. The dataset for testing has 790 rows
- The size of the dataset is 3.2 mb

3.2. Tokenization

Tokenization is used in natural language processing to split
paragraphs and sentences into smaller units that can be more easily assigned meaning. The first step of the NLP process is gathering the data (a sentence) and breaking it into understandable parts (words).

We have applied tokenization on the Article text data.

3.3. Stop Words Removal

Here we have removed the words that occur commonly across all the documents in the corpus.
For that we utilized the nltk library stopword corpus.

3.4. Numerical entry removal

Here we have removed the numbers that occur across all the documents in the corpus. For this we used the Regex library

3.5. Stemming

Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words.
For this we used the porter stemmer from the nltk library.

3.6. Parts of Speech Tagging

Part-of-speech (POS) tagging is a process of converting a sentence to forms – list of words having a tag that signifies whether the word is a noun, adjective, verb or adjective. For this we used the POS tagging module from NLTK

3.7. Lemmatization

Lemmatization is the grouping together of different forms of the same word. For this purpose, we used the Wordnet Lemmatized from the nltk library

3.10. TF-IDF VECTORIZATION

Term frequency-inverse document frequency is a text vectorizer that transforms the text into a usable vector. for TF-IDF VECTORIZATION we have used the sklearn library

3.11. Model Training

MODEL TRAINING is a process in which a machine learning (ML) algorithm is fed with sufficient training data to learn from. For Model Training we have used the sklearn library Multinominal Naïve Bayes Algorithm

3.12. Classification

Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data. We have classified the test data after preprocessing

3.13. Serialization

Serialization is the process of converting an object into a stream of bytes to store the object or transmit it to memory, a database, or a file. For Serialization we have used the pickle library

3.14. Deserialization

Deserialization is the reverse of that process, taking data structured from some format, and rebuilding it into an object. For Deserialization we have used the pickle library.

3.15. Pre-Processed user input

After taking the input from the user we have preprocessed the article text and the used the classification model to classify the news.

3.16. API building

After classification to represent the model in an api, we have used the fastapi framework of the python with the uvicorn server