Architecture Design

Spelling Corrector

Project By: Apurv Bhusari

Document Version control

Date issued	Version	Description	Author
22/04/2023	1	Initial Architecture	Apurv Bhusari
29/06/2023	2	Updated Dependencies	Apurv Bhusari
30/09/2023	3	Final Updated Architecture	Apurv Bhusari

Contents

ABSTRACT

The goal is to build a NLP based system that will correct the spelling mistakes of the words that are given as an input using the state of the art algorithm model.

1. Introduction

1.1 What is Architecture design document?

Any software needs the architectural design to represents the design of software. IEEE defines architectural design as “the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.” The software that is built for computer-based systems can exhibit one of these many architectures.

The goal of Architecture Design (AD) is to give the internal design of the actual program code for the `Spelling Corrector`. AD describes the class diagrams with the methods and relation between classes and program specification

1.2 Scope

Architecture Design (AD) 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. And the complete workflow.

1.3 Constraints

We only predict those words which are in the latest updated version of english dictionary. Any new word added to the dictionary will not be known to the system.

2. Technical Specification

2.1 Dataset

The dataset is comprised of 10000 plus words converted into embeddings using text cleaning ,preprocessing.

2.2 Logging

We should be able to log every activity done by the user

- ✓ The system identifies at which step logging require.
- ✓ The system should be able to log each and every system flow.
- ✓ The system should not be hung even after using so much logging. Logging is just because we can easily debug issuing so logging is mandatory to do.

2.3 Deployment

To deploy a model, you can use the following steps:

- ✓ Save the trained model as a model.h5 file using Python's pickle library.
- ✓ Create a Flask app in Python, which will act as the server for your model.
- ✓ Define the routes for the Flask app, which will determine the behavior of the server when it receives different HTTP requests.
- ✓ In the routes, you can load the h5 file and use it to make predictions based on the input received in the request.
- ✓ You can also create HTML templates to display the results of the predictions on a website.
- ✓ Test the Flask app using Postman or a similar API testing tool to ensure it is working correctly.

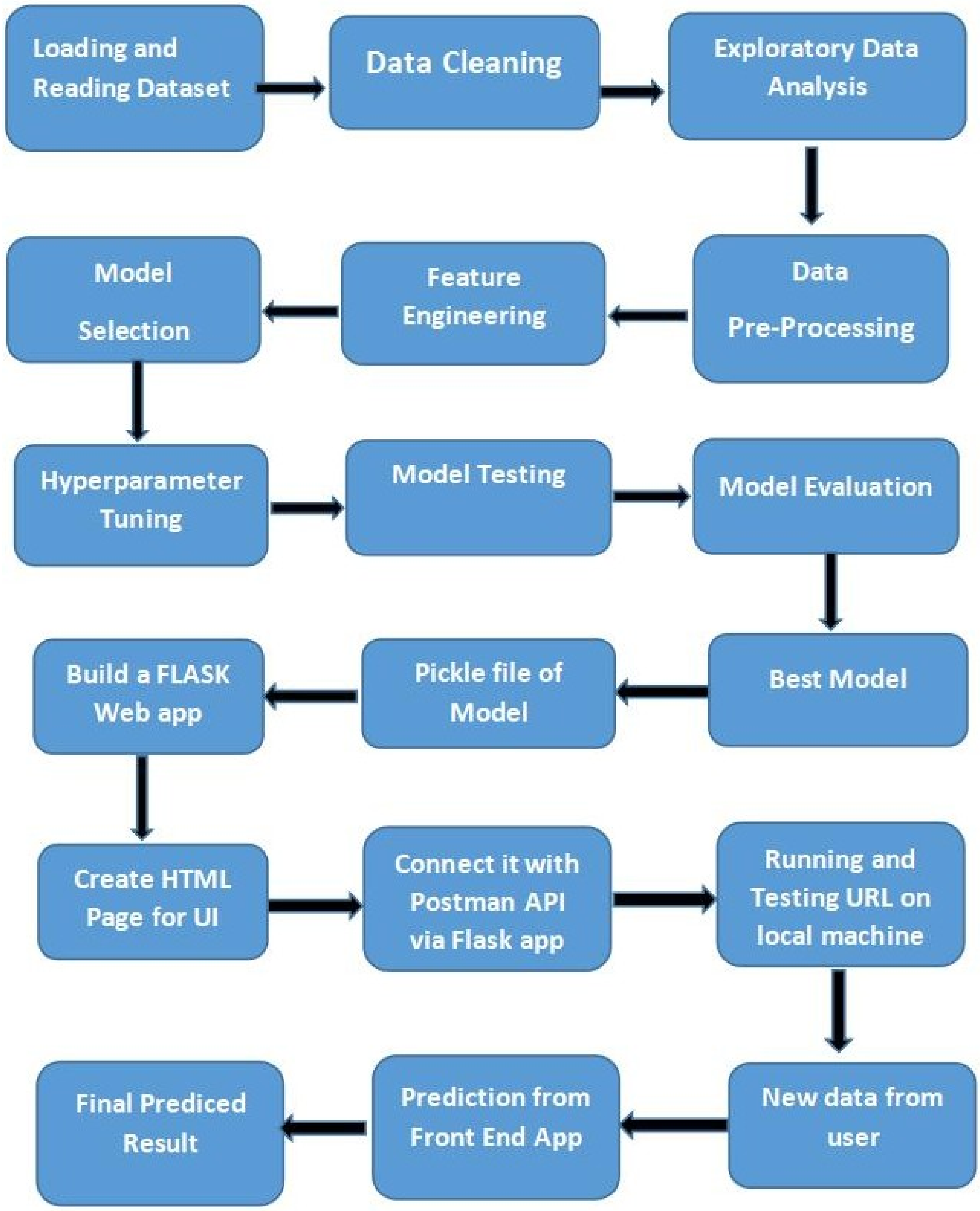
3. Technology Stack

Front End	HTML/CSS
Backend	Python/ Flask
Deployment	AWS CICD using Github actions

4. Proposed Solution

- ✓ To accurately predict the spelling mistake and correct it.
- ✓ The client will input relevant information through our web application, which will be passed to the backend for validation and preprocessing.
- ✓ From there, the data will be fed into a transformer model that has been optimized through fine tuning on our dataset.
- ✓ The model's output will be the final prediction for correcting the spelling mistake, which will be displayed to the client through the web application.

5. Architecture



5.1 Data Gathering

Data source: https://github.com/neuspell/neuspell#Datasets

5.2 Data Cleaning

Data/ Text cleaning is the process of detecting and correcting words, removing punctuations, removing incorrect words, . It is an essential step in data pre-processing and is often the most time-consuming part of the data analysis process.

5.3 Feature Engineering

After pre-processing, like in ml , in nlp we create tokens like embeddings on which our model we will train.

5.4 Model Selection

In This step we use different transformer state of the art model to see which one performs both when it comes to computational cost and is optimized and respond faster .

5.5 Model Evaluation

We evaluate our elmo model(transformer model) using Word Error Rate(WER)

5.6 Deployment Process

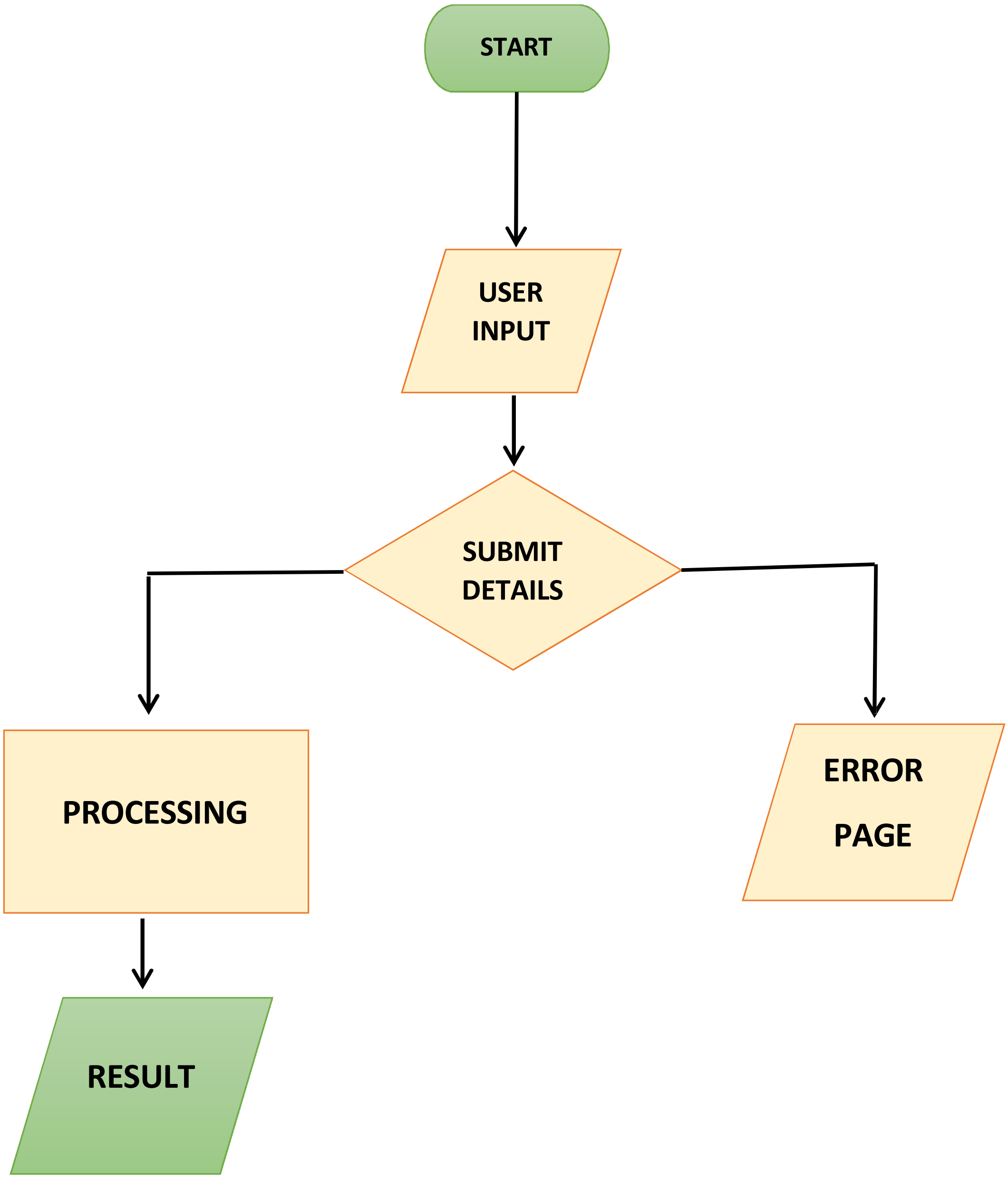
To deploy a model, you can use the following steps:

- ✓ Save the trained model as a pickle file using Python's pickle library.
- ✓ Create a Flask app in Python, which will act as the server for your model.
- ✓ Define the routes for the Flask app, which will determine the behavior of the server when it receives different HTTP requests.
- ✓ In the routes, you can load the pickle file and use it to make predictions based on the input received in the request.
- ✓ You can also create HTML templates to display the results of the predictions on a website.

5.7 GitHub

The whole project directory will be pushed into the GitHub repository

6. User Input / Output Workflow.


START
USER INPUT
SUBMIT DETAILS
PROCESSING
ERROR PAGE
RESULT