HIGH LEVEL DOCUMENT (HLD)
Spelling Corrector

Project By :

Apurv Bhusari

Document Version control

Date issued	Version	Description	Author
29/04/2023	1	Initial HLD	Apurv Bhusari
21/06/2023	2	Updated HLD	Apurv Bhusari
12/09/2023	3	Final HLD	Apurv Bhusari

Contents

ABSTRACT

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

1. Introduction

1.1 Why this High-Level Document?

The purpose of this High-Level Design (HLD) Document is to add 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:

1. Present all of design aspects and define them in detail

2. Describe all user interface being implemented

3. Describe the hardware and software interfaces

4. Describe the performance requirements

5. Include design features and architecture of the project

List and describe the non-functional attributes like:

1. Security
2. Reliability
3. Maintainability
4. Portability
5. Reusability
6. Application compatibility
7. Resource utilization
8. Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as 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.

2. General Description

2.1 Definitions

TERM	DESCRIPTION
SPP	Spelling Corrector
IDE	Integrated Development Environment
API	Application Programming Interface
Postman	Postman is an API development tool

2.2 Product Description

The Spelling Corrector is a nlp system which helps us to correct words which are mistaken.

2.3 Problem Statement

The goal is to build a end to end nlp system which corrects the spelling mistakes of the words that are given as an input using state of the art model

2.4 Proposed Solution

Using all the standard techniques used in the life-cycle of a Data Science project (nlp)starting from Data Exploration, Data Cleaning, Feature Engineering, Model Selection,Model Building and Model Testing and also building a frontend where a user can fill the information in the form input and receive instant output based on our predictionmodel. Throughout the project, we will utilize various techniques and methods to optimize and improve the accuracy of our model.

2.5 Further Improvements

The SC can be easily embedded inside any website or an application and can be used to find out the wrongly given words as an input

This can also be improved further by feeding the model with more data, this data can be from various data repositories, Import-Export agencies, data scraped from internet, etc.

2.6 Data Requirements

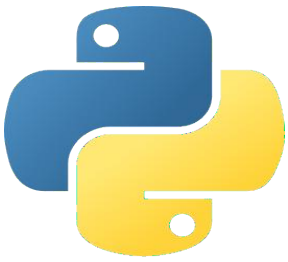
Data requirement completely depend on our problem statement. We need the dataset from past few years, to train our model. Each record in the dataset must have the following features which are important to determine Spelling Corrector:

1. Textual data consisting of more than 10000 plus records
2. Libraries like symspell, dictionaries oriented libraries

1. **Line-item insurance (USD)**: Line-item cost of insurance, created by applying an annual flat rate to commodity cost.

2.7 Tools used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, Matplotlib, Seaborn, Flask ,Jupyter Notebook, Visual Studio Code and a few other libraries were used to build the whole model.








1. Google colab and Visual Studio code were used as IDE.
2. For text preprocessing ,we use nltk,
3. Flask were used for building the web application and server to run the code.
4. GitHub is used as version control system..
5. Hugging face was used to fine tune ,cross validate and compare different models.

2.7.1 Hardware Requirements

1. Windows Server, Linux, or any operating system that can run as a webserver, capable of delivering HTML5 content.
2. Minimum 1.10 GHz processor or equivalent.
3. Between 1-2 GB of free storage
4. Minimum 512 MB of RAM
5. 3 GB of hard-disk space

2.8 Constraints

The front-end must be user friendly and should not need any one to have any prior knowledge in order to use it.

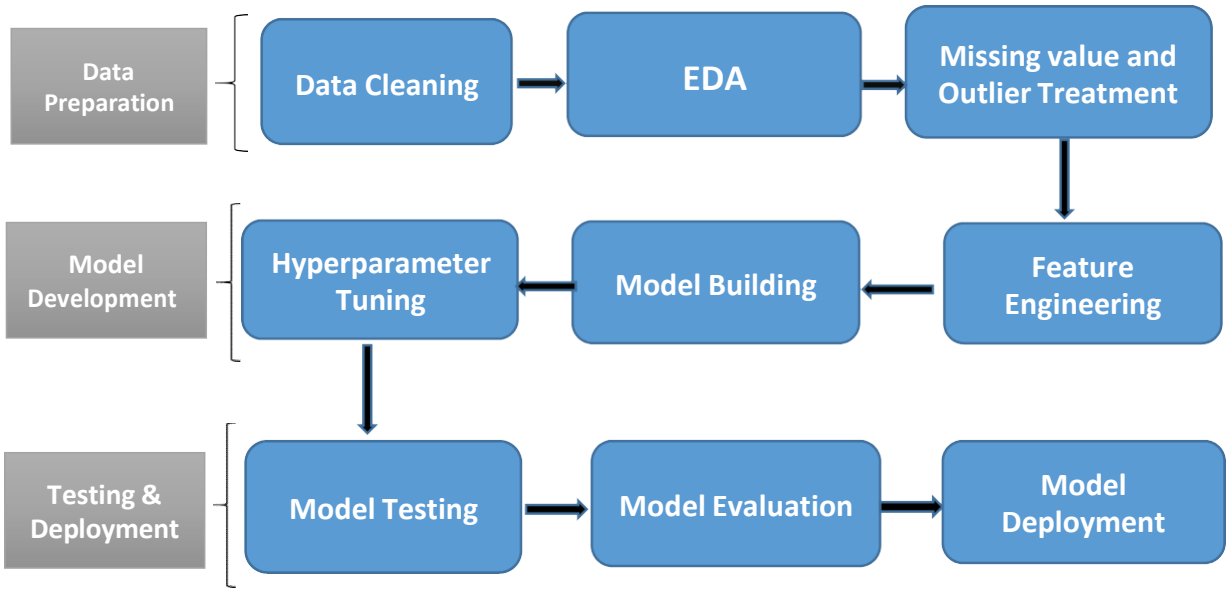
2.9 Assumptions

The main objective of this project is to implement the use case as previously mentioned (2.3 problem statement) for new dataset that comes through the UI. It is assumed that all aspects of this project have the ability to work together as the designer is expecting and also the data on which our model is trained is as correct as possible.

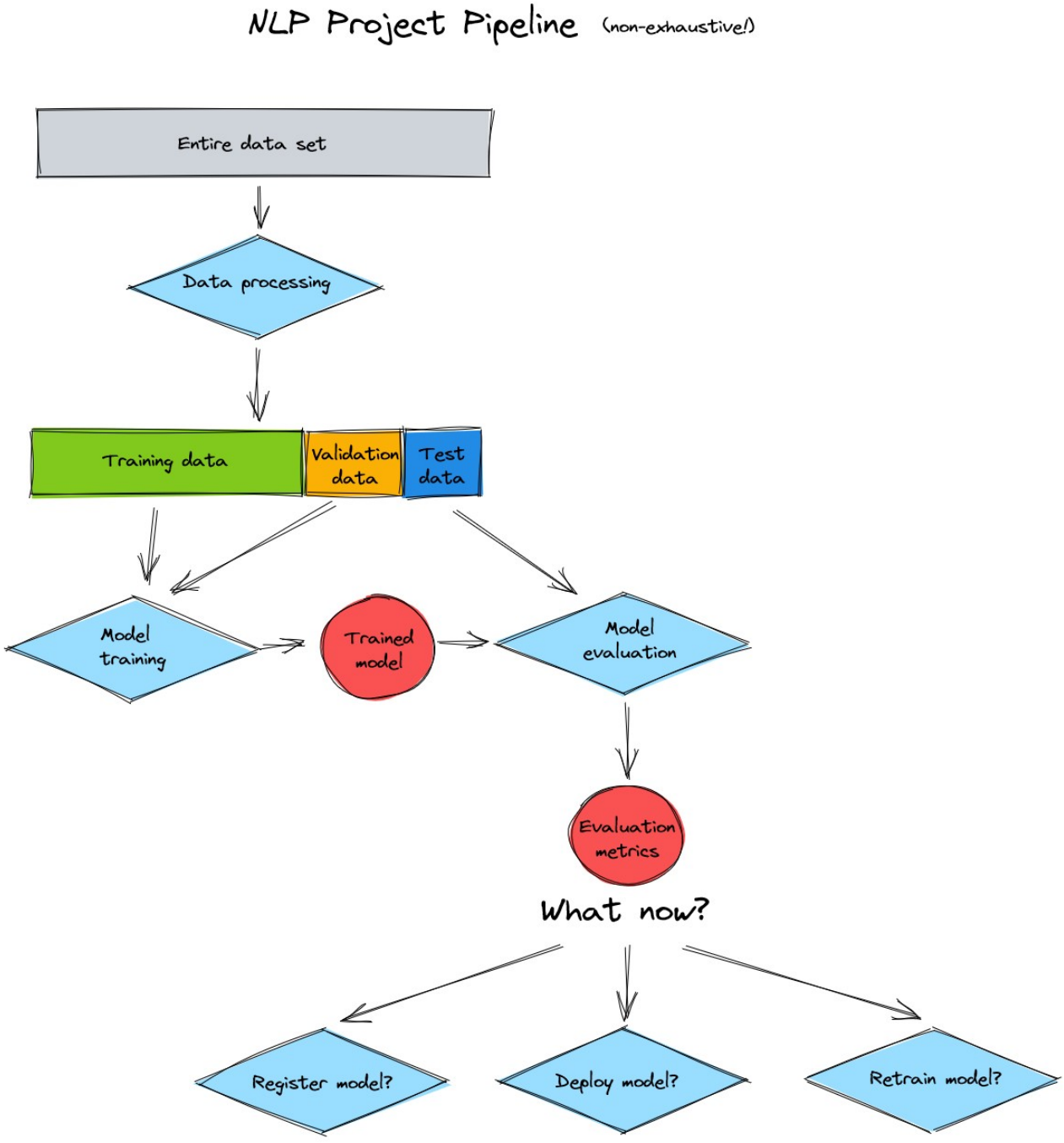
3. Design Details

3.1 Process Flow

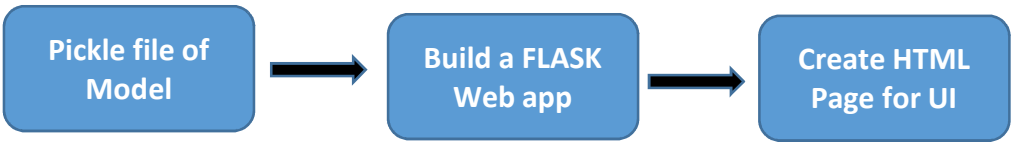
For accomplishment of the task, we will use a trained transformer model. The process flow diagram is shown below :

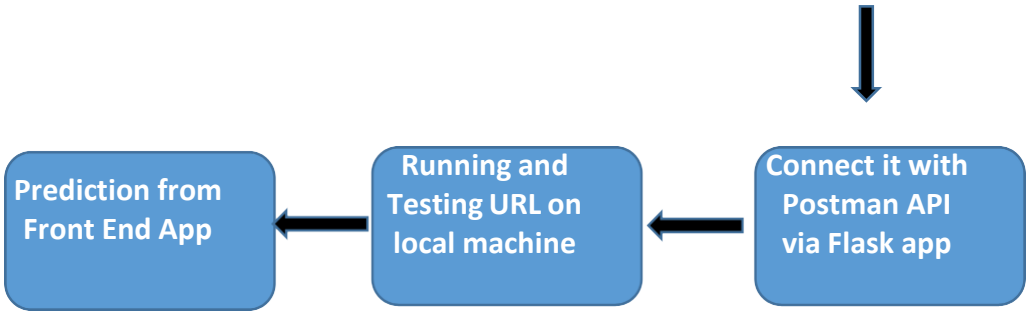


3.1.1 Model Training and Evaluation



3.1.2 Deployment Process




Connect it with Postman API via Flask app
Running and Testing URL on local machine
Prediction from Front End App

3.2 Event Log

The system should log every event so that the user will know what process is running internally.
Initial step-by-step description :
1. The system identifies at what level logging is required
2. The system should be able to log each and every system flow
3. Developer can chose logging method. You can chose database logging/ File logging as well
4. System should not hang even after so many loggings. Logging just because we can easily debug issues so logging is mandatory to do

3.3 Error Handling

Errors should be encountered, an explanation will be displayed as to what went wrong ? An error will be defined as anything that falls outside the normal intended usage.

4 Performance

The SC tool is used to predict whether a word is correct or not based on the spelling given in the input by the end user. You can also type a sentence with wrong words and it will correct it accordingly.

4.1 Reusability

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

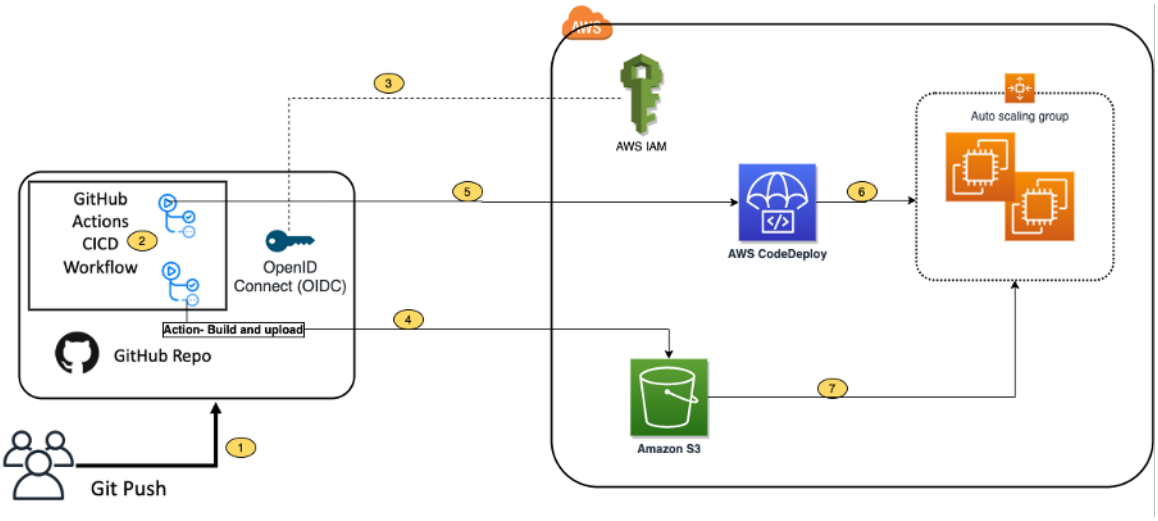
4.2 Application Compatibility

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 Python to ensure proper transfer of information.

4.3 Resource Utilization

When any task is performed, it will likely use all the processing power available to it until finished. Deployment

4.4 Deployment



5 Conclusion

The SC will give the corrected spelling of the not so correct words.