TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

KATHMANDU ENGINEERING COLLEGE

DEPARTMENT OF COMPUTER ENGINEERING

Major Project Proposal

On

Q&A SYSTEM WITH DUPLICATE QUESTION DETECTION

[Code no: CT 755]

Submitted By:

Anurag Shukla – KAT075BCT013

Ayush Shrestha – KAT075BCT018

Bharat Adhikari – KAT075BCT019

Diwash Adhikari – KAT075BCT030

Kathmandu, Nepal

Ashar, 2079

ABSTRACT

The number of questions asked on question and answer (Q&A) forums like Stack Overflow, Quora and Twitter is increasing rapidly. Millions of users visit these sites each month and post their questions. It is no surprise that many of these questions are duplicates. Users may have to wait for a long time to get answers to their questions even though related questions have already been answered. So, it is important to have an automatic way of identifying duplicate threads. Quora, on the other hand, uses a XGBoost model to identify duplicate questions. In this project, we will use TF-IDF for vectorization and for training datasets we will use Machine learning algorithms like Decision Tree, Random Forest, etc.

Keywords: *Stack overflow, Quora, Twitter, XGBoost.*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATION

Acronym	Full Form
AI	Artificial Intelligence
CNN	Convolution Neural Network
CSS	Cascaded Style Sheet
GloVe	Global Vector
GPU	Graphics Processing Unit
GRU	Gated recurrent Unit
GUI	Graphical User Interface
HTML	Hypertext Markup Language
ID3	Iterative Dichotomizer
IG	Information Gain
LSTM	Long-term Short-term Memory
ML	Machine Learning
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
ORM	Object Relational Mapping
Q&A	Question and Answer
RNN	Recurrent Neural Network
SDLC	Software Development Life Cycle
SNLI	Standford Natural Language Inference
SVM	Support Vector Machine
SQL	Structured Query Language
TF-IDF	Term Frequency-Inverse Document Frequency

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND THEORY

Social media platforms are a great success as can be witnessed by the number of the active user base. In the age of internet and social media, there has been a plethora of social media platforms, for example, we have Facebook, for user interaction, LinkedIn, for professional networking, WhatsApp for chat and video calling, Stack Overflow for technical queries, Instagram for photo sharing. Along the line, Quora is a Question & Answer platform and builds around a community of users to share knowledge and express their, opinion and expertise on a variety of topics. Question Answering sites like Yahoo and Google Answers existed over a decade however they failed to keep up the content value of their topics and answers due to a lot of junk information posted; thus, their user base declined [1].

Quora is a social media website where questions are posted by users and answered by experts who provide quality insights. Other users can cooperate by editing questions and suggesting more accurate answers to the submitted questions. According to statistics provided by the Director of Product Management at Quora on 17 September 2018 [4], Quora receives 300 million unique visitors every month, which raises the problem of different users asking similar questions with same intent but in different words. Multiple questions with similar wording can cause readers to spend more time to find the best answer, and make writers answer multiple versions of the same question. Therefore, Quora has an important principle for having a single question thread for logically different questions. For example, questions like ‘‘How can I be a good photographer?’’ and ‘‘What should I do to be a great photographer?’’ are identical because both have the same meaning and should be answered only once. Some questions, like ‘‘How old are you?’’ and ‘‘What is your age?’’ do not have same wording. However, the context remains the same. Therefore, such questions are also considered duplicate. It can be an overhead to have different pages for such questions. Thus, identifying the duplicate questions at Quora and merging them makes knowledge sharing more efficient and effective in many ways. This way, the seekers can get answers to all the questions on a single thread and writers do not need to write the same answer on different locations for the same question. They can get larger number of readers than if the readers are divided in several threads [2].

Identification of duplicate questions is a crucial task in Natural Language Processing (NLP) with many applications such as Recognizing Textual Entailment (RTE) classifying text, retrieving information and detecting plagiarism and Paraphrase Recognition. It measures the degree of similarity between two interrogative fragments. If the fragments are semantically similar, they can get the same answer and are considered duplicate. The task of identification of duplicate questions can be a great challenge because the true meaning of sentence cannot be known with certainty due to the ambiguous language and synonymous expressions.

1.1.1 Natural Language Processing:

Natural Language Processing (NLP) is a way for computers to analyze, understand, and derive meaning from human language in a smart and useful way. By utilizing NLP, developers can organize and structure knowledge to perform tasks such as automatic summarization, translation, named entity recognition, relationship extraction, sentiment analysis, speech recognition and topic segmentation.

1.1.2 Stemming:

Stemming is a process of linguistic normalization, which reduces words to their word root word or chops off the derivational affixes. For example, likes, liking, liked reduce to a common stem word "like".

1.1.3 Lemmatization:

Lemmatization is similar to stemming, difference being that lemmatization refers to doing things properly with use of vocabulary and morphological analysis. For example, Lemmatization clearly identifies the base form of "troubled" to "trouble" denoting some meaning whereas Stemming will cut out "ed" part and convert it into "trouble" which has the wrong meaning and spelling errors.

1.2 PROBLEM STATEMENT

When there is multiple user active in any Q&A system, it raises the problem of different users asking similar questions with the same intent but in different words. Multiple questions with similar wording can cause readers to spend more time to find the best answer, and make writers answer multiple versions of same question. So, in order to solve this problem, we are developing a web application which has an important principle for having a single question thread for logically different questions.

1.3 OBJECTIVES

The objective of the project is to develop an application that:

- Allows users to ask question and checks whether similar question already exists in the database using machine learning and if so, then it suggests the existing similar questions to the user and if the question is new, then it is added to the database.

1.4 SCOPE OF THE PROJECT

This project is primarily useful in question answering system where there is need of finding the duplicate questions to make the system clutter free and also helps to find the good answers easily which would otherwise distribute over multiple questions.

1.5 APPLICATIONS

Our system can be applied to various use cases like:

- Information Management systems can use it for grouping of various kinds of data based on semantic similarity.
- Due to the advancement in plagiarism techniques adopted by plagiarists, it has become challenging to detect plagiarism using traditional string-matching systems. Our system can be tuned to detect plagiarism because newer machine learning models can effectively capture the phrase changes also.
- Advertising agencies can also use the similarity detection in user interests profiling to show personalized contents.

CHAPTER 2: LITERATURE REVIEW

In this there are some papers which were taken references for making this project. From those papers some conclusions were made and from which we made those decisions what features to take or what features to be used. Some papers were similar to the project to be made while some had the algorithms which were to be implemented, while some had the features which were to be implemented, the others had idea about pre-processing the data and how to clean the data before processing it through the algorithms.

The previous work to detect duplicate question pairs using Deep learning approach [5], shows that deep learning approach achieved superior performance than traditional NLP approach. They used deep learning methods like Convolutional Neural Network (CNN), Long-term Short-term Memory Networks (LSTMs), and a hybrid model of CNN and LSTM layers. Their best model is LSTM network that achieved accuracy of 81.07% and F1 score of 75.7%. They used GloVe word vector of 200 dimensions trained using 27 billion Twitter words in their experiments.

The method proposed in paper [6] makes use of Siamese GRU neural network to encode each sentence and apply different distance measurements to the sentence vector output of the neural network. Their approach involves a few necessary steps. The first step was data processing, which involves tokenizing the sentences in the entire dataset using the Stanford Tokenizer. This step also involved changing each question to a fixed length for allowing batch computation using matrix operations. The second step involves sentence encoding, where they used both Recurrent Neural Network (RNN) and Gated Recurrent Unit (GRU). The next step was determining the distance measure that are used in combining the sentence vectors to determine if they are semantically equivalent. There were two approaches for this step, the first being calculating distances between the sentence vectors and running logistic regression to make the prediction. The paper has tested cosine distance, Euclidean distance, and weighted Manhattan distance. The problem here is that it is difficult to know the natural distance measure encoded by the neural network. To tackle this issue, they replaced the distance function with a neural network, leaving it up to this neural network to learn the correct distance function. They provided a row concatenated vector as input to the neural network and also experimented using one layer and two- layer in the neural network. The paper utilized data augmentation as an approach to reduce overfitting. They also did a hyperparameter search by tuning the size of the neural network hidden layer (to 250) and the standardized length of the input sentences (to 30 words) which led to better performance.

In the literature [7], authors have used word ordering and word alignment using a Long-Short-Term-Memory (LSTM) recurrent neural network, and the decomposable attention model respectively and tried to combine them into the LSTM attention model to achieve their best accuracy of 81.4%. Their approach involved implementing various models proposed by various papers produced to determine sentence entailment on the SNLI dataset. Some of these models are Bag of words model, RNN with GRU and LSTM cell, LSTM with attention, Decomposable attention model. LSTM attention model performed well in classifying sentences with words tangentially related. However, in cases were words in the sentences have a different order; the decomposable attention model [8] achieves better performance. This paper [8] tried to combine the GRU/LSTM model with the decomposable attention model to gain from the advantage of both and come up with better models with better accuracy like LSTM with Word-by-Word Attention, and LSTM with Two-Way Word By-Word Attention.

CHAPTER 3: METHODOLOGY

3.1 PROCESS MODEL

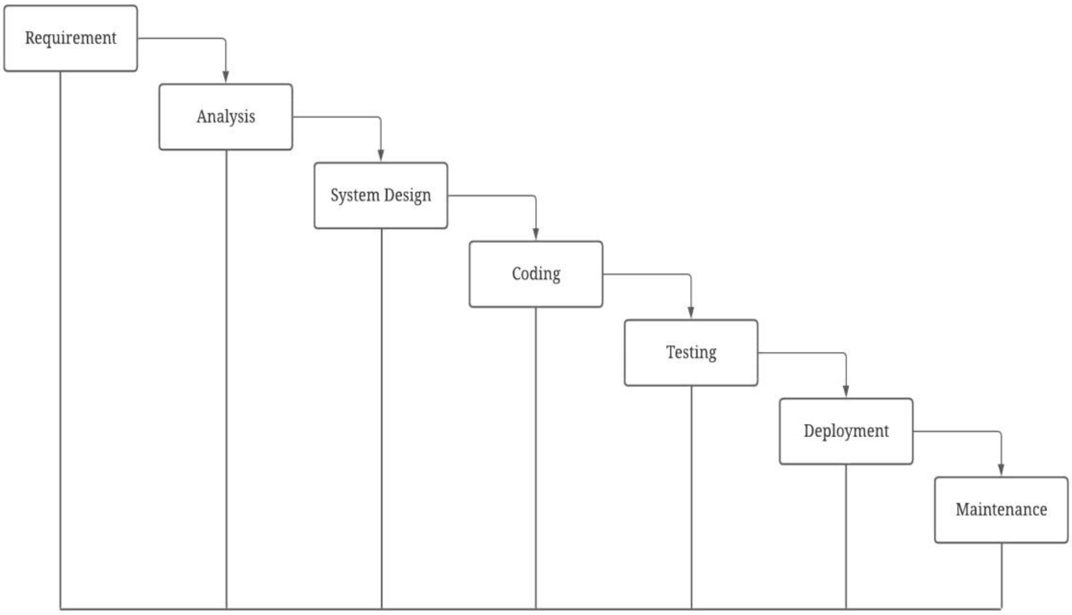

Figure 3.1: Waterfall Model

Waterfall model is the earliest SDLC approach that was used for software development. Since the requirement and the output of our project is known initially, we use waterfall model for development.

The sequential phases in waterfall model are:

- **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System design:** The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Coding/Implementation:** With inputs from the system design, the system is first developed in small programs called units, which are integrated in next phase. Each unit is developed and tested for its functionality, which is referred as unit testing.

- **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system:** Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance:** There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

3.2 BLOCK DIAGRAM

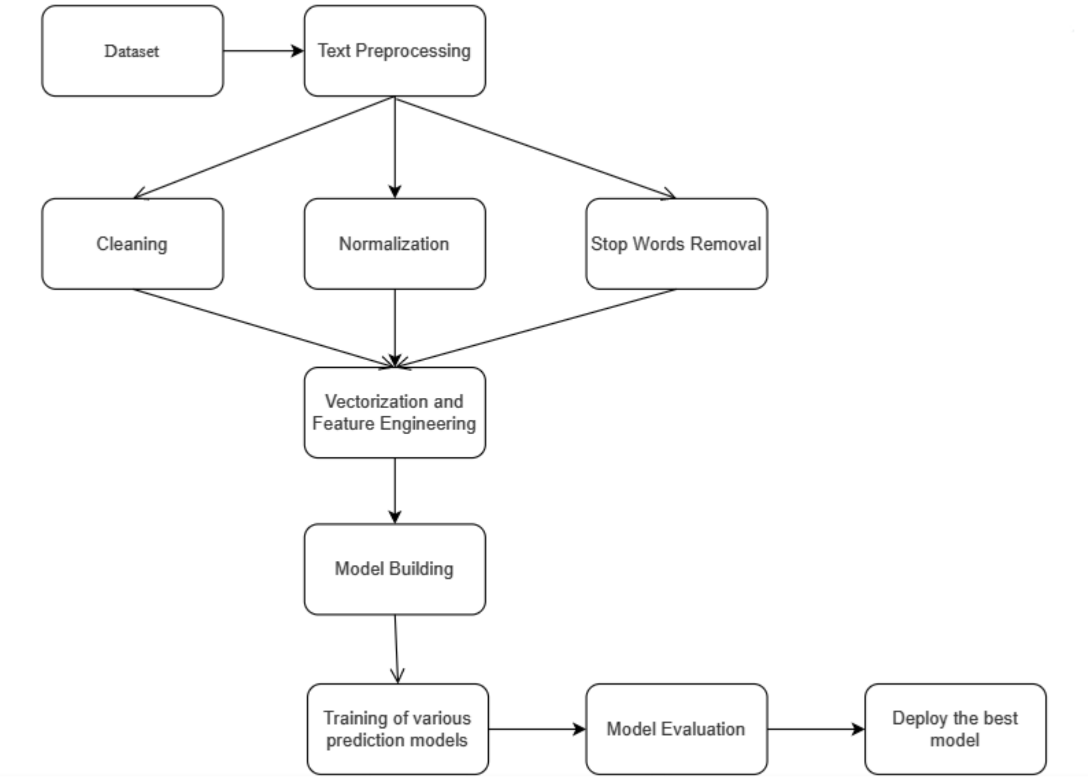

Figure 3.2: Block Diagram

The block diagram consists of several different components.
They are:

i) Datasets:

The first step to sole any Machine Learning problem is to collect quality data. The data we want can be collected from various datasets repositories or can be created manually.

ii) Text Preprocessing

The aim of this preprocessing step is to clean the messy data so that it is ready for further processing. In NLP, preprocessing consists of several steps as outlined below:

- Cleaning: This includes removing hashtags, HTML tags, etc.
- Normalization: This includes converting the text into lowercase, removing punctuation symbols and extra spaces.
- Stop words removal: Stop words includes words that do not carry much meaning but are important grammatically. For example, words such as “to”, “is”,” but”, etc. can be counted as stop words.

iii) Vectorization and Feature Engineering

The process by which we convert text into numbers is known as vectorization. So, basically in this process we are taking the text data and creating a vector of numbers. Vectorization is an important task of NLP. Without vectorization we won’t be able to apply any machine learning models on the text data.

iv) Model Building

In this step, we choose the ML models that will be used to train the data. This may take into consideration the problem being solved.

Commonly used ML models for NLP tasks are Naive Bayes, SVM and Random Forest although other models like Neural Networks can also be used.

v) Training the model

The next step after initializing the model is to begin training the model on the train data. The model will now learn all parameters from the train data.

vi) Model Evaluation

Now, we test the model on a dataset that the model has not yet seen. This will help us to evaluate the performance of the model.

vii) Deployment

After evaluating the model, we will deploy the model as per the need.

3.3 ALGORITHM

3.3.1 TF-IDF:

TF-IDF stands for Term Frequency-Inverse Document Frequency. It is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. TF-IDF uses a weighting system that assigns a weight to each word in a document based on its term frequency (TF) and the inverse document frequency (IDF). The words with higher scores of weights are deemed to be more significant. It is important because Machine learning with natural language is faced with one major hurdle – its algorithms usually deal with numbers, but natural language is text. So, we need to transform that text into numbers, otherwise known as text vectorization. It's a fundamental step in the process of machine learning for analyzing data, and different vectorization algorithms will drastically affect end results, so we need to choose one that will deliver the higher results. After transforming words into numbers, the TF-IDF score can be fed to various machine learning algorithms for training.

Its score calculation steps are:

For a word(t) in a document(d),
TF = frequency(t,d)/total no. of words in document
where, t = word or term,
d = one document
IDF = log(N/n)
where, N = total number of documents
n = number of documents containing the word(t)
Finally,
weight (TF-IDF) = TF * IDF

3.3.2 Random Forest:

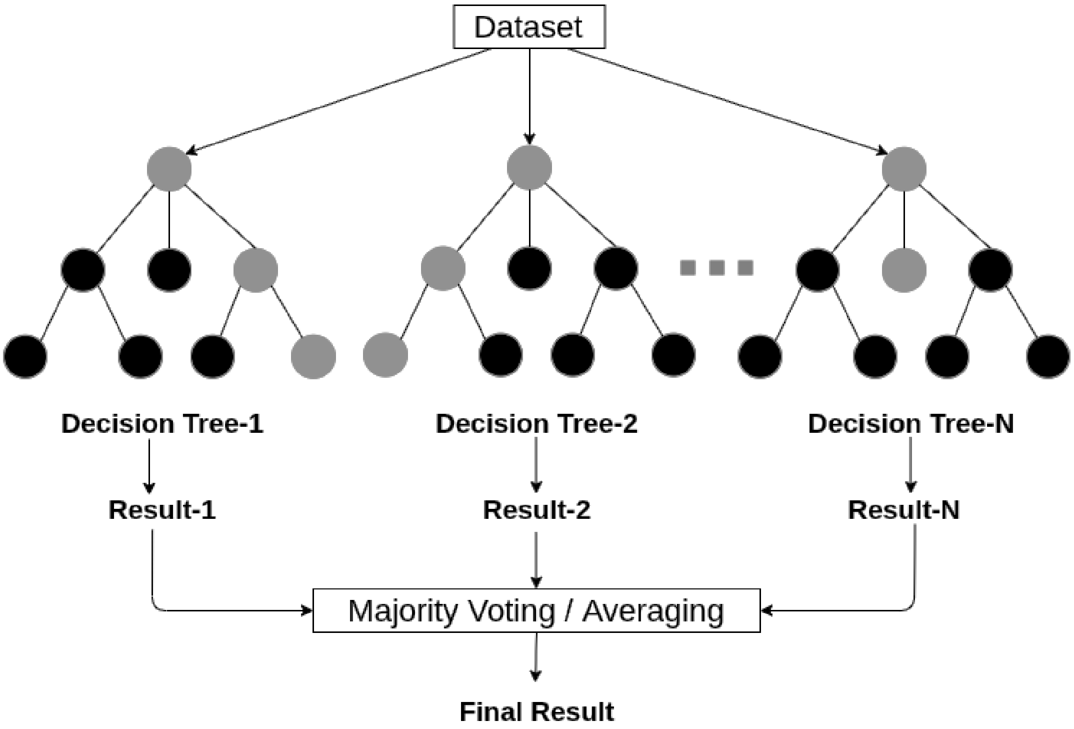

Figure 3.3: Random Forest

Random Forest is an algorithm based on the concept of ensemble learning. It can be used for both classification and regression task. It is an extended form of decision tree. This algorithm builds multiple decision trees and combines them to produce more accurate and stable results. Instead of relying on one decision tree, it takes the prediction from each tree and finds the final output based on majority vote.

Its steps are:
Step-1: Select random certain data points from the training set.
Step-2: Build the decision trees associated with the selected data points (Subsets).
Step-3: Choose the number N for decision trees that we want to build.
Step-4: Repeat Step 1 & 2.
Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

3.3.3 Decision Tree:

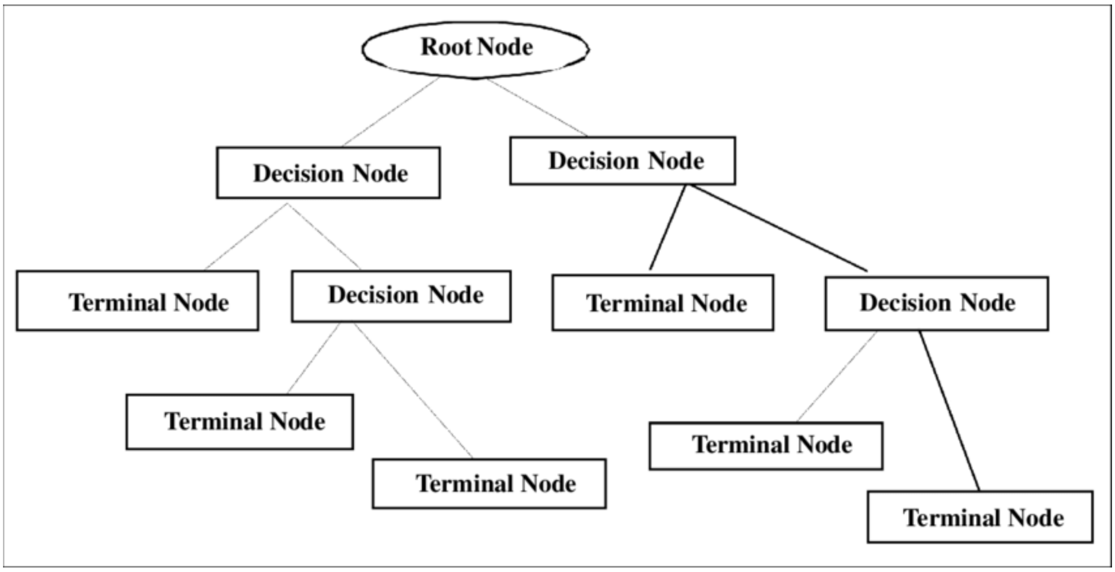

Figure 3.4: Decision Tree

It is a tree structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. A decision tree simply asks a question and based on the answer (Yes/No), it further split the tree into subtree. ID3 is a popular method to construct a decision tree.

Its steps are:

Step-1: Calculate the Information Gain of each feature before splitting.

$$IG_{before} = -\frac{P}{P+N}log2\frac{P}{P+N} - \frac{N}{P+N}log2\frac{N}{P+N}$$

Step-2: Calculate Entropy of each class

$$E_{class} = \Sigma(\frac{Pi+Ni}{P+N})\ IG_{subclass}$$

N → Negative, P → Positive

Where,

$$IG_{subclass} = -\frac{Pi}{Pi+Ni}log2\frac{Pi}{Pi+Ni} - \frac{Ni}{Pi+Ni}log2\frac{Ni}{Pi+Ni}$$

Step-3: Calculating gain for each class

$$Gain = IG_{before} - E_{class}$$

Step-4: Compare all gain value & select the root node having the highest gain.

Step-5: Again, start from step 1 to 5 for root node. Continue this unit it covers all value.

3.4 FLOWCHART

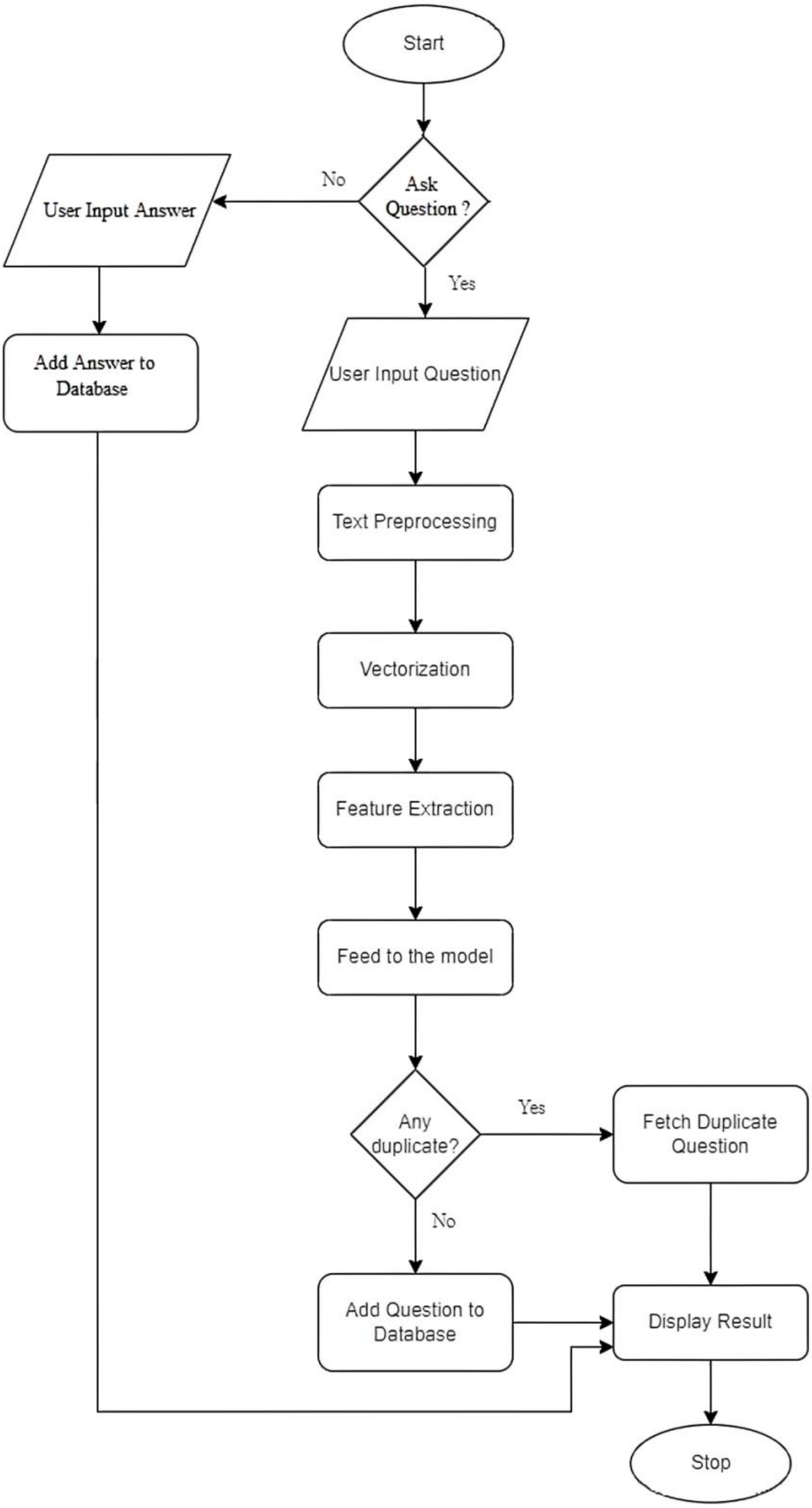

Figure 3.5: Flowchart

3.5 TOOLS TO BE USED

3.5.1 Python:

Python is a high-level, general-purpose programming language. Its design emphasizes on code readability. Its object-oriented and language construct helps the programmers to write clean code for small and large projects.

Python supports multiple programming paradigms, including both functional and object-oriented programming. It is extensively used in data science and machine learning purposes due to the availability of high-standard libraries and frameworks.

3.5.2 Django:

Django is a free and open source, full stack web application framework written in Python. Its design philosophies are loosely coupled due to its each element of stack independent on others, less coding, don't repeat yourself, fast development and clean design. Its features are Object Relational Mapping (ORM) support, administrative GUI and lightweight web server for end-to-end application development and testing.

3.5.3 Numpy:

Numpy is the fundamental package for scientific computing in python. It provides multidimensional array object, various derived objects like matrices and other mathematical and logical operations. At the core of Numpy package, is the ndarray which encapsulates n-dimensional arrays of homogeneous data types.

3.5.4 Pandas:

Pandas is an open-source library that is made mainly for working with relational or labeled data intuitively. It provides various data structures and operations for manipulating numerical data and times series. This library is built on top of Numpy library. Its features are easy handling of missing data, data merging, reshaping, data split, etc.

3.5.5 Matplotlib:

Matplotlib is a plotting library for the Python programming language. It helps to generate histograms, bar charts and other types of charts and plots with just few lines of code. It consists of module called pyplot which provides simple functions for adding plot elements such as lines, images to the axes in the current figure.

3.5.6 NLTK:

The Natural Language Toolkit, or more commonly NLTK is a suite of libraries and programs for symbolic and statistical natural language processing for English language. It also includes graphical demonstrations and sample data.

3.5.7 Sklearn:

Sklearn is a python library which consists of a lot of efficient tools or machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction. It contains supervised learning algorithms, cross-validation, unsupervised learning algorithms, feature extraction, etc.

3.5.8 Google Colab:

It is a free Jupyter notebook environment that runs entirely on cloud and doesn't require a setup. It helps in documenting the code. It also allows to create, upload and share notebooks. Colab integrates frequently used libraries and provides cloud service with GPU for free.

3.5.9 HTML & CSS:

HTML stands for HyperText Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. HTML is a markup language that is used by the browser to manipulate text, images, and other content to display it in the required format.

CSS (Cascading Style Sheets) is a stylesheet language used to design the webpage to make it attractive. The reason for using CSS is to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.

3.5.10 JavaScript:

JavaScript is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document. It is used for creating web and mobile apps, building web servers, game development and adding interactive behavior to web pages.

3.5.11 SQLite:

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is a database, which is zero-configured, which means like other databases you do not need to configure it in your system. SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its storage files directly.

CHAPTER 4: EPILOGUE

4.1 EXPECTED OUTPUT

Our project is expected to give following outputs:

- Develop a web application of Q&A with duplicate questions detection.
- Detection of duplicate or repeated questions and suggest related questions.
- Allows user to read answers to their questions.
- Allows user to write answer to the questions.

4.2 GANTT CHART

Key Activities	Duration					Duration					
	SEMESTER 1					SEMESTER 2					
	June	July	Aug	Sept	Oct	Nov	Dec	Jan	Feb	Mar	Apr
Requirement	▨	▨									
Analysis			▨								
System Design				▨							
Implementation					▨	▨	▨	▨	▨	▨	
Testing										▨	▨
Deployment										▨	▨
Documentation	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

Figure 4.1: Gantt Chart

REFERENCES

[1] Ansari, Navedanjum, and Rajesh Sharma. "Identifying semantically duplicate questions using data science approach: A Quora case study." *arXiv preprint arXiv:2004.11694* (2020).

[2] Imtiaz, Zainab, et al. "Duplicate questions pair detection using Siamese Malstm." *IEEE Access* 8 (2020): 21932-21942.

[3] Patel, Uday, Amol Dattu, Pritam Patil, Renuka Khot, and Sujit Tilak. "Quora Question Duplication Problem." (2020).

[4] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, ''Signature verification using a 'siamese' time delay neural network,'' in Proc. 6th Int. Conf. Neural Inf. Process. Syst. (NIPS), San Francisco, CA, USA, 1993, pp. 737–744.

[5] Travis Addair. 2017. Duplicate question pair detection with deep learning. Stanf. Univ. J (2017).

[6] Y Homma, S Sy, and C Yeh. 2016. Detecting Duplicate Questions with Deep Learning. 30th Conf. Neural Inf. Process. Syst. (NIPS 2016), no. Nips (2016), 1–8.

[7] A Tung and E Xu. 2017. Determining Entailment of Questions in the Quora Dataset., 8 pages.

[8] A Parikh, O Tckstrm, D Das, and J Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. Proc. 2016 Conf. Empir. Methods Nat. Lang. Process (2016), 2249–2255.