FAST NUCES PESHAWAR

DEPARTMENT OF SOFTWARE ENGINEERING

Word Prediction with Bigram and Trigram Models

Noman Yousaf (p200614), Afaq ur Rehman (p200034), Khawaja Fawad (p200058)

April 1, 2024

Contents

1 Introduction

The Word Prediction project aims to develop an intelligent system capable of predicting the next word in a sentence using bigram and trigram models. This system is designed to assist users in typing and enhance their experience with natural language interfaces.

1.1 Technology Stack

Our project utilizes a diverse technology stack to achieve accurate word predictions:

- Python: The core programming language for model development and data processing.
- HTML and CSS: For building a user-friendly web interface.
- Flash: A Python library for fast and scalable deep learning model training.

Additionally, we leverage bigram and trigram models to analyze patterns in English sentences. The project's dataset consists of English sentences stored in a chat.txt file.

2 Approach Overview

Our approach involves several key steps in building and training the word prediction model:

1. Data Acquisition: Collecting a dataset of English sentences from various sources, including chat logs.
2. Data Preprocessing: Cleaning and tokenizing the text data to prepare it for model training.
3. Model Training: Training bigram and trigram models using the preprocessed data.
4. Prediction Function: Implementing a prediction function to generate the next word based on the input sentence.
5. User Interface: Designing a user-friendly web interface using HTML and CSS to interact with the prediction model.

This systematic approach ensures the development of an effective word prediction system.

3 Implementation Details

The implementation phase involves the following detailed tasks:

3.1 Data Preprocessing

During data preprocessing, we perform the following operations:

- Tokenization: Splitting sentences into individual words.
- Cleaning: Removing punctuation and special characters.
- Lowercasing: Converting all text to lowercase for consistency.

These steps prepare the text data for model training.

3.2 Model Training

We train bigram and trigram models using the processed data. The models analyze the frequency of word combinations to predict the next word in a sentence.

3.3 Prediction Function

The prediction function takes an input sentence and generates the next word based on the trained models. It calculates the probability of each word given the context of the input sentence.

3.4 User Interface

We develop a user-friendly web interface using HTML and CSS to allow users to input sentences and receive word predictions in real-time.

4 Usage and Scope

The Word Prediction project aims to provide users with a convenient tool for predicting the next word in a sentence based on the context. The scope of the project includes:

- Improving typing efficiency: Users can benefit from word prediction to speed up their typing by receiving suggestions for the next word as they type.
- Enhancing user experience: The project aims to enhance user experience in natural language interfaces by providing accurate and contextually relevant word predictions.
- Supporting various applications: Word prediction can be integrated into various applications such as text editors, messaging platforms, and virtual keyboards to assist users in composing text more efficiently.

By providing accurate predictions and a user-friendly interface, the project aims to fulfill the diverse needs of users across different platforms and applications.

5 Challenges and Considerations

Several challenges and considerations were encountered during the project:

- Data Quality: Ensuring the dataset contains a diverse range of English sentences.
- Model Complexity: Balancing model complexity with computational resources.
- User Interface Design: Designing an intuitive interface for easy interaction.

Addressing these challenges effectively is crucial for the success of the word prediction system.

6 Team Segregation and Roles

Our team members are assigned specific roles and responsibilities to ensure a smooth and efficient development process:

- Noman Yousaf (p200614): Research, documentation, and testing of the software with AI-powered tools.
- Afaq Ahmad (p20123): Development of the latest framework for frontend.
- Fawad Ahmed (p200058): Lead developer responsible for model architecture design, training, and optimization using Flash and N-grams model.

Name	**Role**
Noman Yousaf (p200614)	Project Manager and Testing Engineer
Afaq Ahmad (p20123)	Frontend Engineer
Fawad Ahmed (p200058)	Backend Engineer

Collaboration and communication among team members are essential for successful project completion.

7 Test Case Diagram

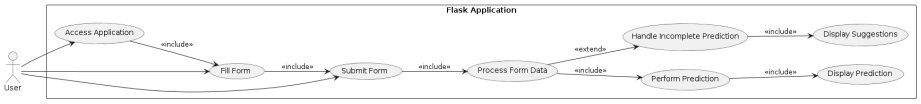

Figure 1: Test Case Diagram

8 Class Diagram

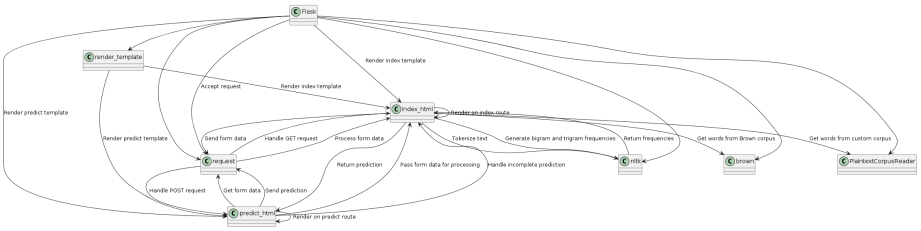

Figure 2: Class Diagram

Architecture Diagram

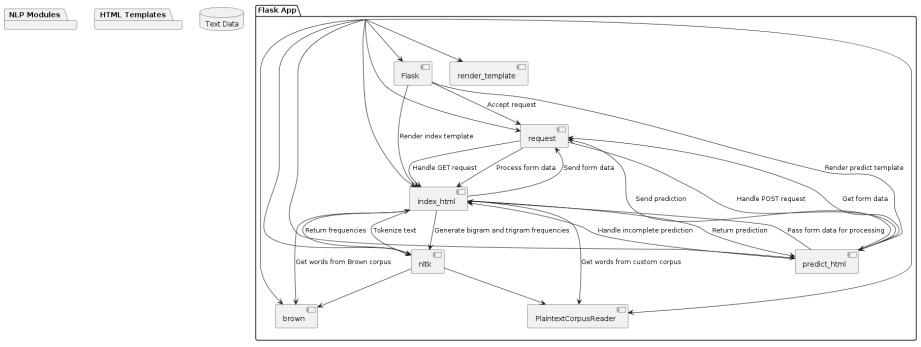

Figure 3: Architecture Diagram

9 Activity Diagram

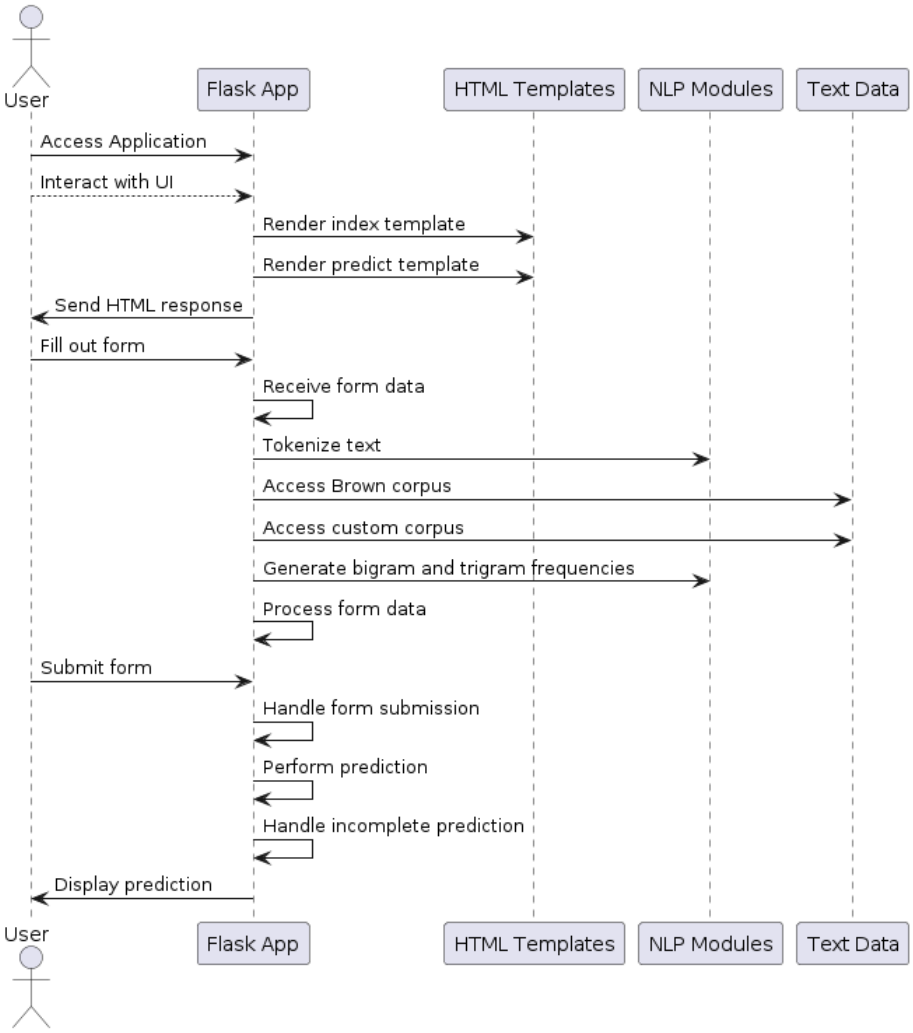

Figure 4: Activity Diagram

10 Test Case Diagram

Basic Input Test:
Input: "The quick brown"
Expected Output: "fox"
Single Word Input Test:
Input: "Hello"
Expected Output: "world"
Empty Input Test:
Input: ""
Expected Output: (No prediction)
Input with No Previous Words Test:
Input: "apple"
Expected Output: (No prediction)
Input with Limited Training Data Test:
Input: "I love"
Expected Output: "you"
Input with Rare Words Test:
Input: "The cat"
Expected Output: "sat"
Input with Punctuation Test:
Input: "I am"
Expected Output: "going"
Input with Multiple Possible Predictions Test:
Input: "I want"
Expected Output: "to" (or "a", depending on training data)
Long Input Test:
Input: "The quick brown fox jumps over the lazy dog"
Expected Output: (Prediction for the next word after "dog")
Case Insensitivity Test:
Input: "THE QUICK BROWN"
Expected Output: "FOX"
These test cases cover various scenarios such as basic input, edge cases, punctuation handling, and handling of limited training data.

10.1 Diagram for better vision

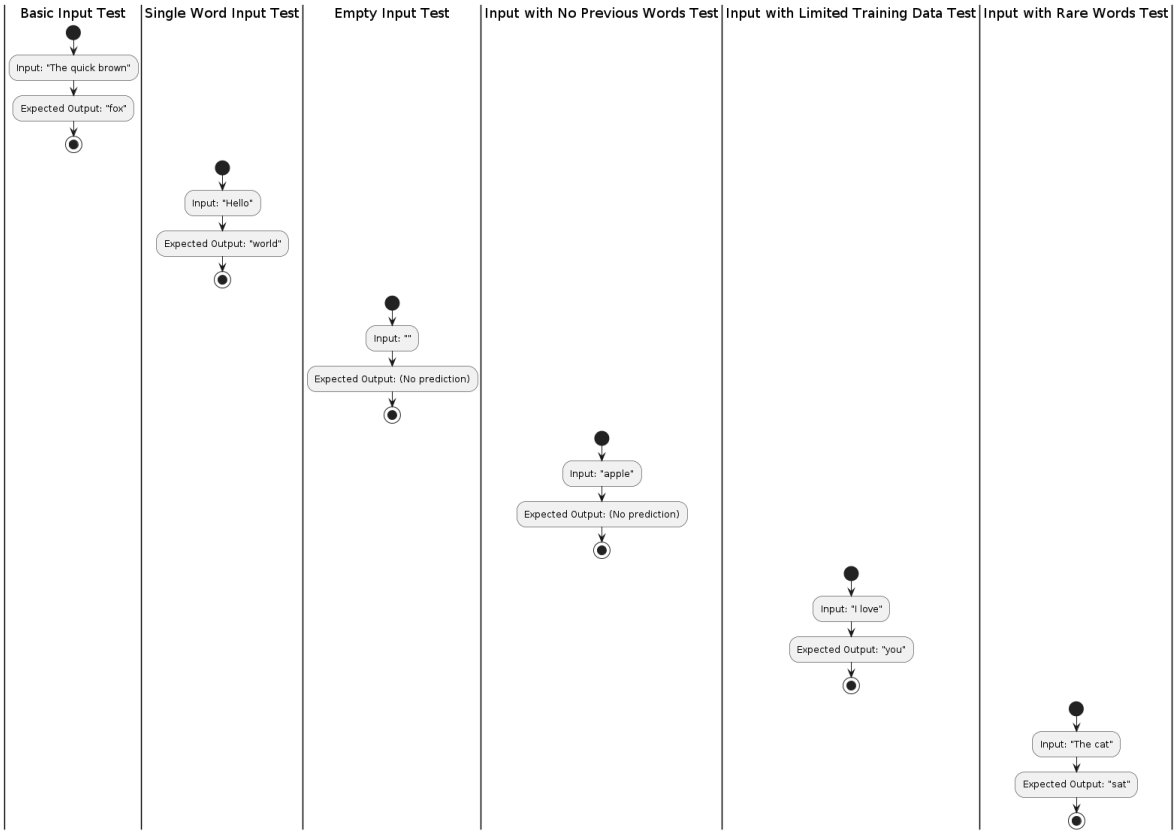

Figure 5: Test Case

11 Conclusion

The Word Prediction project leverages bigram and trigram models along with Python, HTML, CSS, and Flash to develop an accurate and efficient word prediction system. By analyzing patterns in English sentences, the system enhances user typing experience and improves natural language interfaces.