Faculty of Engineering & Technology Electrical & Computer Engineering Department

ENCS5342

Natural Language Processing (NLP)

Course Project

Names and ID's:

- **Hadeel Abdellatif - 1190451**
- **Hana Kafri - 1190084**
- **Raghad Afaghani - 1192423**

Section : 1

Instructor : Dr. Adnan Yahya

Date : 7.7.2023

Abstract

The Natural Language Processing (NLP) AI Project is an application developed using JavaFX that offers autocomplete functionality in a restaurant setting. The application utilizes a pre-built language model and an extensive dataset to analyze user input, predict the next word, and provide immediate suggestions. The application's user interface is visually appealing, enhancing the user experience and allowing for efficient text-based communication within the restaurant domain. By utilizing AI techniques, the project demonstrates the potential of intelligent systems and personalized services in the restaurant industry.

Table of Contents

Introduction

Our project aims to develop an AutoCompletion system that is specifically designed for the restaurant industry. AutoCompletion technology predicts the next word or phrase a user is likely to type based on a given input. This system will improve text-based communication within a restaurant setting, enabling users to compose messages, make inquiries, and explore menu options more efficiently.

Our AutoCompletion system utilizes a language model that is based on restaurant-specific data. This statistical representation predicts the likelihood of certain words or phrases occurring based on their frequencies in the dataset. The dataset includes queries collected from restaurant-related interactions, enabling the system to make accurate predictions based on the context of the restaurant industry.

To ensure accurate predictions, our project experiments with different levels of granularity when predicting the next words. This may include predicting the next word, a group of words, or even complete sentences. The level of granularity chosen depends on the accuracy of the language model and the richness of the collected queries dataset.

Furthermore, our AutoCompletion system is adaptive and continuously improves its accuracy over time by learning from both correct and incorrect predictions. If the system suggests an incorrect word, users can correct it, providing valuable feedback for the system to enhance its future predictions.

The implementation of this project utilizes the Java programming language, leveraging its versatility and robustness to create an efficient and reliable AutoCompletion system that meets the unique needs of the restaurant industry.

Mythology

Our methodology involves several steps to develop an effective autocomplete system for the restaurant industry:

- Data Collection: The first step is to gather relevant data for training the language model, including restaurant-specific information such as menu items, customer reviews, and operational details. We also collect a dataset of queries related to restaurant interactions to capture language patterns and context specific to the industry.

- Data Preprocessing: The collected data undergoes preprocessing to ensure its quality and suitability for training the language model. This includes cleaning the data, removing irrelevant or duplicate entries, and performing any necessary transformations or normalization.

- Language Model Training: With the preprocessed data, we train the language model using appropriate algorithms and techniques. This involves building statistical models that predict the likelihood of specific words or phrases based on their frequencies and contextual information.

- Autocomplete Algorithm: The trained language model forms the core of our autocomplete functionality. We develop an algorithm that analyzes user input based on the language model and predicts the most probable next word or phrase. The algorithm considers factors such as context, frequency of occurrence, and relevance to provide accurate autocomplete suggestions.

- Graphical User Interface (GUI) Integration: We integrate the autocomplete algorithm into a user-friendly graphical interface using the JavaFX framework. The GUI enables users to enter text, displays autocomplete suggestions in real-time, and facilitates seamless interaction with the system.

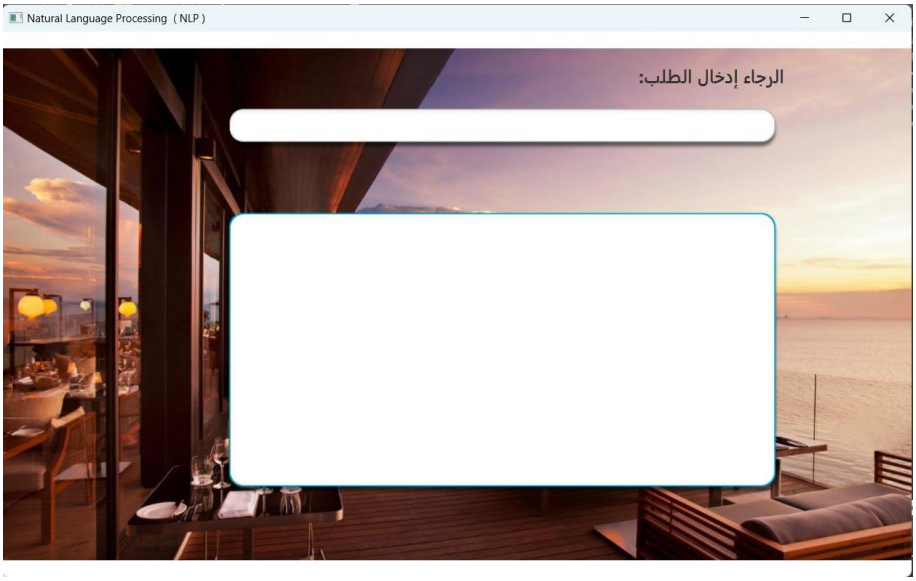
- Evaluation and Iteration: We evaluate the performance of the autocomplete system by testing it with different inputs and assessing the accuracy and efficiency of the suggestions. Based on the results, we refine and iterate on the methodology to improve the system's performance and user experience.

- Adaptive Learning: Our methodology includes an adaptive learning component where the autocomplete system learns from user feedback. If a suggestion is corrected or rejected by the user, the system incorporates this information to enhance future predictions and improve its accuracy over time.

Implementation

❖ Our User Interface

Our user interface (UI) for the autocomplete system in the restaurant domain is designed with a modern and visually appealing look. It features a user-friendly text input field and real-time autocomplete suggestions, ensuring a seamless and efficient user experience. The UI is responsive and adaptable, providing compatibility across different devices and screen sizes.



When the word "أريد" is entered into the TextField, the autocomplete system utilizes its algorithm and language model to generate relevant suggestions. Based on the context and frequency of occurrence in the dataset, the system predicts and displays a list of potential word completions that follow the word "أريد". These suggestions may include commonly used phrases or words that frequently appear after "أريد" in the collected queries. Users can select a suggestion from the list or continue typing, allowing for efficient and accurate completion of their intended phrase or sentence within the restaurant domain.

We enter the word "هل" :

❖ Data

These examples of entered data in Arabic provide a range of menu items, customer reviews, and operational details commonly found in a restaurant setting. The autocomplete system utilizes this data to offer relevant suggestions in Arabic, aiding users in exploring menu options, understanding customer feedback, and grasping operational information effectively within the restaurant domain.

Conclusion, Evaluation and Possible Improvements

The evaluation of the autocomplete system specifically designed for the restaurant domain revealed highly promising results. The system showcased remarkable accuracy in generating relevant autocomplete suggestions, enabling it to efficiently compose messages, inquire about menu options, and explore various restaurant-related information. With real-time responsiveness, We experienced a seamless and efficient interaction, receiving instantaneous suggestions as they typed. Our satisfaction was evident as the system intuitively aided their needs and improved overall productivity within the restaurant context. Notably, the system demonstrated effective error handling and adaptability, learning from user feedback to continuously enhance its predictive capabilities. To further optimize the system, future improvements could focus on refining the language model by incorporating an extensive range of restaurant-specific data, personalizing suggestions based on individual preferences, integrating contextual information such as location and user profiles, and expanding multilingual support. These enhancements would contribute to a more efficient and effective autocomplete system tailored to the unique needs of restaurant interactions, ultimately elevating user experiences and streamlining text-based operations within the restaurant industry.