Closed-Domain Natural Language Question-Answering System for advising London Restaurant

Author: Xiaofeng Fu

Degree: BSc Computer Science

Supervisor: Prof. Anthony Hunter

Submission Date: 30th April 2018

This report is submitted as part requirement for the BSc Degree in Computer Science at UCL. It is substantially the result of my own work except where explicitly indicated in the text. The report may be freely copied and distributed provided the source is explicitly acknowledged.

Abstract

Nowadays, restaurant recommendation applications have been phenomenally successful. For example, TripAdvisor and Yelp, both are well-known applications in the world. They provide exhaustive restaurants information and users can easily search for a specific restaurant in their applications or websites. However, little has been developed to provide a question-answering (QA) system for restaurant recommendation. Different from reading the discrete restaurant information on the websites, a natural language QA system can provide a reader-friendly textual format of restaurant information that translated from the data on website for users. Yet that QA system involves both natural language understanding (NLU), natural language processing (NLP) and natural language generation (NLG). The development of the natural language system has always been a challenging issue. This project aims to develop a simple natural language QA system that is able to answer questions about London restaurants with well-organized textual format using Python.

A deep research of natural language system and QA system was done to better understand the architecture and the algorithms to implement the system. And a literature review of Natural Language Toolkit (NLTK) for Python was conducted to learn the method to process natural language in Python. Moreover, various meeting with supervisor was organized to sketch the blueprint of the project and decide the core programming language and technologies to be used in this project. Based on the preliminary activities, the QA system for advising London restaurant was designed to process and understand user’s question and then answer the question in reader-friendly textual format. The system was implemented using Python, JavaScript, CSS and HTML with the Django web framework. A testing process and user feedback questionnaire was used to insure the practicability and the stability of the application.

After the implementation of the system, user feedback questionnaires were provided to people that had used the QA system. And the result from the questionnaires has a positive feedback that shows the natural language QA system provides a convenient method for people to search for restaurants information in London.

Contents

1. **Introduction** 
   1. Motivation
   2. Initial Brief
   3. Overview of Approach
   4. Report Structure
2. **Background**
   1. Introduction
   2. Project Context
   3. Programming Language and Frameworks
3. **Requirements and Analysis**
   1. MoSCoW requirements
   2. Use Cases
   3. User Interviews
   4. Risk Assessment
   5. Discussion
4. **Design and Implementation**
   1. System Overview
   2. System Architecture
   3. Database
   4. Natural Language Processing System
   5. Natural Language Generation System
   6. User Interface Design
   7. Challenges and Design Decisions
5. **Testing**
   1. Testing Strategy
   2. Unit Testing
   3. Use Case Testing
6. **User Feedback**
   1. Introduction
   2. Findings and Discussion
7. **Conclusion and Evaluation**
   1. Conclusion
   2. Evaluation
   3. Future work
8. **Reference**
9. **Bibliography**
10. **Appendices**

Appendix A: System Manual

Appendix B: User Manual

Appendix C: Interim Report

Appendix D: Project Plan

Appendix E: User Feedback Questionnaires

Appendix F: Use Cases

Appendix G: Code Listing

Appendix H: Example Video

1. Introduction
   1. Motivation

With the development of Internet, people can easily find a specific restaurant information through a search engine or restaurant recommendation applications. According to the study of the Telmetrics-xAd, restaurants are always the top mobile and local search category on search engines based on sizable query log data from different sources (Greg, 2012). Moreover, from the study, the most popular applications that used by mobile users to search for a restaurant are multi-purpose online restaurant review aggregators like TripAdvisor and Yelp (Kevin, 2016).

For these application, the method to show the information of restaurants are not direct enough. People must enter the name of the restaurant first and then find the information they need from the search result page, which may contain many useless information for them and it is ineffective. Compare with the strong review aggregation system and the huge restaurants database of these applications, few of them has been developed to provide a natural language question-answering (QA) system for restaurant recommendation. Different from the traditional method to search for a restaurant in these restaurant review aggregators, a QA system can provide a reader-friendly textual format of restaurant information that translated from the data on website for users. However, the implementation of that QA system involves both natural language understanding (NLU), natural language processing (NLP) and natural language generation (NLG). It has always been a challenge for people to develop a natural language system. In this project, the implementation of a simple natural language QA system is the primary objective.

* 1. Initial Brief

This section includes the main challenge to implement the system and the aims & goals of this project.

* + 1. Main Challenge

The main challenge of the project is to find a feasible method to implement the natural language process system and the natural language generation system for this project based on limited searchable resources. In addition, this project involves new programming technologies. Not only a new programming language Python, but also Natural Language Toolkit (NLTK), a platform for building Python programs to work with human language.

* + 1. Aims and Goals

Aims:

* Collect enough London restaurants information from the Internet and put them into a local database.
* Understand background of NLP and NLG and be familiar with the core implementation process of them.
* Learn Python and Django web framework and apply them to build the system.
* Design and implement a QA system that translates the discrete restaurant information into reader-friendly textual format.

Goals:

* **NLP system:** The QA system must ‘understand’ users questions (the system should take the question as an input and then identify the question type and process the question to a set of keywords
* **MySQL database:** The system must contain a MySQL database to put the restaurants data. And the system must have the ability to search and operate the database.
* **NLG system:** The system must generate a paragraph of the restaurant information with well-organized reader-friendly textual format. And this system must have the ability to provide different answers for different types of question.
* **User Interface:** The QA system must have a simple user interface for users to ask questions. This user interface should contain at least a chat window and an input field.
* **Usability:** The system must achieve a high efficiency and user satisfaction in a specified context of use.
  1. Overview of Approach

In this project, several weekly meetings with supervisor were organized first to create ideas of the implementation of the system. These meetings not only gave the opportunity to discuss about the main challenges, but also to receive feedback from the supervisor. Moreover, email was used to daily communicate with the supervisor.

Throughout the project, the implementation approach was step-by-step. In each work period, a part of the QA system had to be implemented. In the project plan document (Appendix D), the work plan of the completion of the project is shown. This work plan outlined objectives and works that had to be completed in each period. As all works were completed, testing process and user feedback questionnaire were used to improve and evaluate the quality of this project.

* 1. Report Structure

This report consists of seven chapters. In chapter 1 and 2, the background information and the overview of the project will be introduced. This section will include the introduction of related programming language and web frameworks used in the project. Chapter 3 covers the detailed requirements and related analysis. The requirements will be shown as a MoSCoW table. Furthermore, use cases and user interviews will be covered in this section. Chapter 4 will discuss the design and implementation process of the system. In this section, an overview and the architecture of the system will be given. And this report will provide the implementation detail of each component of the system separately. In chapter 5, testing strategy of the project will be descripted. Chapter 6 will cover the introduction and the result of the user feedback questionnaire. And the result will be analyzed in this report. Finally, chapter 7 contains the conclusion and the evaluation of the project.

1. Background
   1. Introduction

At the start of the project, learning the basic natural language knowledges from the NLTK textbook and the Internet was an important process. Although there are numerous example NLP and NLG systems on the Internet, these systems are complex, and it was difficult for a beginner to read the source code and learn the algorithms. To implement the project, much time was spent on the preliminary process and the period to get familiar with these knowledges. Background information below are the main technologies and resources that are learnt and used during the implementation of this project.

* 1. **Project Context**

This section provides the contextual knowledge required to understand the project and it involves the technologies that were used during this project.

* + 1. **Question-answering (QA) System**

Question-answering is a computer science technology within the field of NLP. It is concerned with developing applications that can answer human questions automatically in the form of short natural language texts. Open domain and closed domain are two main methods to implement the QA system. For the open domain QA system, it involves the knowledges of computational linguistics, information retrieval and knowledge representation. It is hard to implement an open domain QA system since there are a huge number of question types and this system must identify the correct one to return an answer.

For this project, a closed domain question-answer system was the objective to implemented. By contrast with an open domain system, this system has a restricted domain, which means it has fewer question types to identify. And it is easier for a beginner to implement it. Especially for the project, the system only has to answer the questions about London restaurants. Moreover, it is also easier for the system to find the correct answer.

To have a deeper understanding about the implementation process of a closed domain QA system, a review of literatures and reading materials from the Internet were conducted, a **Keywords Based Closed Domain Question Answering System** report written by Rohini P. Kamdi and Avinash J. Agrawal gives a systematic introduction about how to implement a closed domain QA system. And this report also introduces the overview of the system architecture and several core algorithms in this system. Furthermore, there are several reading materials on **Quora**, which provide useful information of the development of a QA system.

* + 1. **Natural Language Processing (NLP)**

Natural language processing is a computer technology that concerns with the interactions of natural languages and computer. It focuses on the processing procedure of natural language data and translate these data into an organized format. The processing of natural language involves both syntax part and semantics part.

For the syntax part, it covers part-of-speech (POS) tagging, parsing and word segmentation. POS tagging aims to determine the POS for each word in a sentence. In this project, the POS tagging tool provided by NLTK was used. This tool not only tag each word with POS, but also consider the context of the word to eliminate the ambiguity of many words that have multiple POS. Word segmentation part aims to separate a chunk of continuous text into words in a sentence. For the semantics part, named entity recognition, question-answering and natural language generation are included. Named entity recognition is used to determine which items in a sentence map to proper names. In this project it is used to identify the name of the restaurant.

**The book of NLTK** was used to learn the natural language processing. This book includes the knowledges of NLP and also the related functions that can be used in Python programming.

* + 1. **Natural Language Generation (NLG)**

Natural language generation is the process that takes input data and translate these input into a sensible reader-friendly text. Four core elements are important in a NLG system: knowledge source, communicative goal, user model and discourse model. Knowledge source is the knowledge that the system has, for example, a database. And communication goal is the expected text the system output. User model is the group of people that will read the output text and discourse model is the style of the output text (Swizec, 2012).

In a traditional architecture of a NLG system, content determination is the first step. The system will determine what content should be included in the output, more generally, what the system going to say. Secondly, the system will plan the sentences semantically. Finally, specific words that used in the sentences will be determined. And the system will check the connections between each sentence to make sure all connections are smoothly.

* + 1. **Natural Language Toolkit (NLTK)**

*“NLTK is a leading platform for building Python programs to work with human language data.” (NLTK 3.2.5 documentation, 2017)*

As the introduction on the website of NLTK website, it is a powerful programming library for Python application developers to process natural language data. In involves more than 50 corpora and related resources. Throughout the project, NLTK documents was used as a reference and NLTK library was used to implement the functionalities of the QA system.

* 1. **Programming Language and Frameworks**

This section involves the programming language and related web frameworks that used in the project.

* + 1. **Python**

Python is used as the programming language of this project since it has many powerful libraries like NLTK and PyDictionary. NLTK is one of the leading platform for NLP programming and PyDictionary is a similar library as WordNet to get meaning, synonyms, translations and Antonyms of words (PyDictionary 1.5.2 documentation, 2018).

* + 1. **Django**

The system was designed to develop as a web application. To combine Python and web development together, Django is a reliable platform to implement it. This web framework provides a platform for developers to write Python, HTML and JavaScript together. Moreover, Django encourages agile development and clean design and it can prevent much of the problems of web development to give developers a better programming environment without the reinvent the wheel (Django, 2018).

* + 1. **Materialize**

Because of the related development experiences before, Materialize was used to build the user interface of the project. Materialize is a responsive front-end framework based on Material Design. It focuses on a better user experience and it is easy for developers to use.

1. Requirements and Analysis

In Chapter 3,

* 1. MoSCoW requirements
  2. **Use Cases**
  3. **User Interviews**
  4. **Risk Assessment**
  5. **Discussion**

1. References

Greg Sterling, 2012, <https://searchengineland.com/study-64-percent-of-mobile-restaurant-searchers-convert-immediately-or-within-an-hour-135188>

Kevin Kent, 2016, <https://www.reviewtrackers.com/mobile-customers-food/>

Swizec Teller, 2012, <https://swizec.com/blog/natural-language-generation-system-architectures/swizec/4535>

NLTK 3.2.5 documentation, 2017, <http://www.nltk.org/>

PyDictionary 1.5.2 documentation, 2018, <https://pypi.python.org/pypi/PyDictionary/>

Django, 2018, <https://www.djangoproject.com/>

1. Bibliography

Rohini P. Kamdi, Avinash J. Agrawal, 2015, <http://www.mecs-press.org/ijisa/ijisa-v7-n12/IJISA-V7-N12-6.pdf>

Amit Gunjal, 2017, how do I develop a closed-domain natural language QA system. <https://www.quora.com/How-do-I-develop-a-closed-domain-natural-language-QA-system>

Brad Miller & David Ranum, 2018, Problem Solving with Algorithms and Data Structures using Python, <http://interactivepython.org/runestone/static/pythonds/index.html>