

Module: CMP-6040A Artificial Intelligence

Assignment: CW2: Developing an Intelligent Conversational System

Set by : Dr. Wenjia Wang e-mail: wenjia.wang@uea.ac.uk

Checked by : Dr. Hane Aung

Date set : 02 November 2021

Value : 80%

Date due : 19 January 2022 by 15:00

Returned by : 28/02/2022

Submission : The Blackboard

Learning outcomes

1. Good understanding of basic concepts, theories and modern methods in AI.
2. Gained essential practical skills for applying AI technologies to real world problems.

Specification

Overview

The aim of this coursework is to develop an intelligent conversational system, i.e. a chatbot in short, for a specified business service. It will be done in the form of a grouped mini-project.

Group formation: A group will be formed with 3 students either by your voluntary choice or assigned by the module organiser and responsible lecturer. Grouping should be completed by the deadline - Friday of Week 7.

If you have formed a group yourselves with the required number of the members, and you may create a name for your group, then your group leader needs to input the names of the members into an Excel file available on OneDrive (the link will be given later separately) as soon as possible before the above deadline.

If you cannot find enough members to set up a group or you cannot find a group to join by the deadline, you should also send an email to Dr. Hane Aung (min.aung@uea.ac.uk) with whoever members you have or none, to inform him that you do not have a group yet for the coursework. We will then try to assign you to a group or someone else to your group.

Description

The task is to design and implement an intelligent chatbot for a train operating company with two main objectives:

1. to help their customers in finding the cheapest available ticket for their chosen journey;
2. to improve customer service satisfaction by applying some appropriate AI techniques.

The chatbot system should start by greeting a user who has just accessed your system and asking a basic question, such as “Hello, how can I help you?”, to start a conversation. Your chatbot should try to keep the conversation going in the intended context until it gets all the necessary information from the user, and then perform whatever necessary actions to meet the needs of the user.

Below are the topics of conversation and their associated specific scenarios that your chatbot should be able to complete.

1. Finding the cheapest train ticket

The objectives of this task are to get the necessary information from a user/passenger for their intended train journey through a series of dialogue and then find the cheapest available train ticket for the journey from any appropriate train ticket booking web site.

The basic requirements and procedure for this task should be:

- (1) Respond to a user who initiates a conversation or by detecting a user who is surfing your business web site or any other related social media;
- (2) Carry on the conversation until enough information (such as travel date and time, departure station, destination station, etc.) has been obtained;
- (3) Access a train ticket booking web site, e.g. National Rail's or Greater Anglia's web sites, to search for the available tickets that meet the requirements of the user, and
- (4) Choose and present the cheapest ticket(s) to the user, with a hyperlink to the web site where you found the ticket;
- (5) If the user wishes to book it, he or she can click the URL. Your chatbot should then direct him or her to the ticket booking web site. After this, the conversation can be considered to have been successfully completed.

A test example: As a student at UEA, you plan to travel from Norwich to London on the 15th January by train, and come back on the 17th January. Using this information, talk to your chatbot to see if it can find the cheapest train ticket for you.

A bit more complicated case is that a passenger intends to travel from Norwich to Swansea on the 20th January, hoping to depart before 11am, and come back, departing from Swansea in the afternoon (after 1pm) of 30th January.

2. Improving Customer Service

This task is to extend your chatbot system with more functions and more knowledge in the Knowledgebase(KB) to improve customer services related to transit operations. There is no limit on the services your chatbot system can provide through conversations as long as they are related to train services. Below is just one specific scenario you need to take as your starting point (a minimal requirement).

Scenario: For Task 2, you may focus on the train services between Weymouth and London (Waterloo Station) provided by the South Western Railway Company (TOC). Imagine you are a passenger on a train travelling from Weymouth to London (Waterloo Station), when the train has just arrived at a station, e.g. Southampton, an announcement is made that your train has

been delayed for 10 minutes, so now you want to find out the time that the train shall arrive at your destination, e.g. London Waterloo Station.

This should be done through another conversation with your chatbot. It should first ask you - the passenger, for necessary information, including which train you are on, where it is now, how much time the train has been delayed, and where is your destination, etc. Your chatbot should then present the gathered information (data) to a predictive model (or some models) to let it predict the arrival time of the delayed train at your destination.

To complete this task, you need to build some predictive models and use one or some to predict the delay time (or arrival time) and integrate them into your chatbot system.

Hints for developing the chatbot system

Analysing the tasks and scenarios described above, you should consider your chatbot as an expert system that is capable of answering questions or providing some services through conversations.

As you have learned from the lectures, an Expert System (ES) in this coursework should contain some of the components as follows:

- (i) A User Interface (UI), this may be done with graphics or text,
- (ii) A Natural Language Processing and Understanding (NLPU) component,
- (iii) A Knowledge-Base (KB),
- (iv) A Reasoning Engine (RE),
- (v) A Prediction Model (PM), or some models
- (vi) A database (DB),
- (vii) A Knowledge Acquisition (KA) component - optional

The knowledge-base (KB) stores the domain knowledge and expertise, which can be represented by Questions and Answers (QnA), rules and/or other ways. The knowledge in the KB can be retrieved by the Reasoning Engine (RE) to help deriving the most appropriate answers to the questions in a conversation. You should build the KB by searching and collecting relevant QnAs, and/or rules, and/or models. The KB should be updated or adapted through learning.

The NLPU component is to process the questions asked in text in order to understand what they really mean. For this purpose, you may use some Python or Java-based or any other NLP libraries or any existing NLP systems as a black box as long as they meet your requirements, but you must give the appropriate references.

The RE will be needed when you cannot find the direct answers to the asked questions. You can apply / implement any reasoning methods taught in the module or not in the module, to infer the answers.

If your RE cannot find or infer a good or reasonable answer for a question, your chatbot has to say something, e.g. "Sorry, I have no answer."

For the prediction model (PM), it is your choice what type of models you want to build for your chatbot system. It could be any type of the models you have learned in this module, e.g. k-Nearest Neighbours (KNN), Bayesian model, Markov model or neural networks, or any other types of the models, e.g. regression, that may have been provided in the machine learning library, e.g. Python's Scikit-learn.

The historical performance data of trains will be provided through a database (more instructions will be given later in a seminar) for training and testing your predictive models.

Note: Although you should not worry too much about the accuracy of your predictive models as long as you can build and use them to perform the intended tasks, you should try to train some models of the same type or different types, assess their accuracy with appropriate measures and then choose the most accurate and reliable one to be implemented in your chatbot system. Or you may implement a number of them to improve the accuracy and reliability of prediction. You are expected to write at least a section in your report specifically describing this part of the work.

There are some train performance information web sites that may provide an estimated or predicted arrival time for a train for any station along its journey. You may use their estimated time as a comparison baseline to your predictive models.

A Database (DB) should be designed to save some data, such as the details of previous conversations, train station names and their official abbreviations, historical train running data for training prediction models etc. It is your choice - whatsoever type of database and structure you want to use as long as it meets your requirements.

A knowledge acquisition (KA) component is an additional function that your chatbot system may provide for the system developers to add more (or update) knowledge, e.g. more Qs and As, or rules, to the knowledge base.

The User Interface (UI) can be designed and implemented with or without graphics. The most basic requirement is that the conversation is carried out through text in the command-line.

Multimedia input and output are optional: If you wish, you may use any existing speech recognition (speech-to-text and text-to-speech) software or toolkit, as a black box in a speech-based conversation. It is not necessary to do this; however, if you do and do well, you can get some bonus marks.

The chatbot can be developed as either a web-based system, a mobile application, or a standalone system. It may be deployed through a web site, a social media platform, e.g. Skype, Twitter, Facebook, etc. or a chatbot emulator or running directly as a standalone system.

The conversation should be carried out in the English language.

There is no restriction on the choice of development platform and programming language. You can decide what you want to use but you must justify your decision.

If you have to, you may use an existing framework to develop your chatbot (e.g. Microsoft's LUIS), but you may not get as much credit as you would for assembling your own framework using libraries for each of the chatbot components.

Due to our limited teaching resources, we can only provide technical support for using Python.

Project Groups: you can form a group by finding partners yourself. Each group should have a group leader responsible for organising and coordinating the coursework. You should also give a name to your group for convenience.

If you cannot form or find a group, let Dr. Aung know and we will then try to allocate you into a group.

Relationship to formative assessment

Formative assessments will be carried out between Weeks 10 and 12 to check each group's progress.

Week 10: A short report (2 to 4 pages) which outlines the requirements, initial design and the work plan. Submit the outline's pdf file of your group report to the dropbox on the module blackboard.

Week 12: An updated interim report (3-6 pages) and practical progress check.

Each group should extend the outline of their report by adding some more contents or details on the requirements, some review of methods, tools and packages for developing chatbot, improved design, revised workplan etc. and submit it to the dropbox by 15/12/21.

For practical progress check, we will use the lab sessions to check your practical development up to date. More details will be given in due time.

Deliverables

1. A Chatbot system:

- (a) Presentation and Demonstration: The chatbot implemented by each group needs to be presented (with some PowerPoint slides describing its development) and demonstrated within 15 minutes live on-line to the whole class. The dates and time will be scheduled later, usually within the same week of the deadline.
- (b) Source code submission: All the source code must be submitted in a zipped file to a dropbox in Blackboard after your demonstration.

2. Group report: each group must write a technical report that documents all the key development work. It should be structured with sections such as Introduction, Related Work, Methods, Design, Implementation and Testing, Evaluation and Discussion, Conclusion or Summary, and also references. In your Introduction Section, each group must describe how you have distributed the work to each member and how much (in percentage) each member has actually contributed to the coursework. There is no restriction on the numbers of words and pages for the group report, but it must not be formatted in two columns.

3. Individual report (one page): each student must write an individual report to indicate and justify your contribution to the coursework. Specifically, you should clearly describe the parts of the chatbot development work you have done and/or have been involved with, and how much (in percentage, e.g. 33%) you have contributed to the group work in total — this figure should be agreed on by the other members in your group.

You should be honest and fair when estimating your own and other members' contributions, and compromise a bit if necessary to maintain a good work relationship among the group members. If for whatever reasons, an agreement in your group can not be reached, you can write down your own estimates of the contribution from yourself and other members in your individual report. The module organiser will make the final decision based on the evidence presented in the individual reports.

4. A short (2–3 minutes) video that shows the best parts of your chatbot.

Note: you are NOT allowed to upload any of your work (code, reports and video) to any public media, e.g. GitHub or Youtube, etc., without getting a permission from the module organiser.

Marking scheme

1. Presentation and Demonstration of your Chatbot system (50%):

Each group is required to present and demonstrate your Chatbot in front of the entire class to show that it can complete

- Task 1 for finding a cheapest train ticket smoothly for a journey given through a conversation.
- Task 2 for predicting the delay of a given train.

The marks will be distributed to smooth completion of these two tasks, and implementation of other features, such as NLP, GUI, Prediction and other additional functions, and also accuracy of your answers to the questions asked. The detailed marking scheme for this part will be given separately on the mark sheet.

Please note: this demonstration is very different from the ones you did in other modules, it is NOT a PC lab bench demo just to your markers but a combination of presentation and live demonstration to your peers and markers live on-line. Then the members in each group will be asked some questions by the markers and the students.

The purpose is to give you an opportunity to show your chatbot to all other students and also see and judge the chatbots developed by other groups. More details on how the demonstrations are conducted will be given in due time.

2. The reports (40%):

- Group Technical Report (35%): Logically structured and clearly written with sufficient technical details and figures (e.g. screen shots, etc.).
- Individual Report (5%): A clearly stated justification of your contributions.

3. The video (10%):

- As a part of your documentation, this short video (2 - 3 minutes) should show how your chatbot completes major tasks smoothly.
- Good visual quality and verbal explanation.

The final mark for an individual student is calculated by combining the marks from above elements, and multiplied by his/her contribution weight. More details on the marking scheme and calculating individual's mark will be explained in the coursework briefing.

In addition, as this coursework is of a team work, the leader of a group is very important in organising and managing the entire team to work on the coursework effectively and efficiently. So, if a leader has led his or her group very well to complete the coursework successfully, he or she may be given a few extra marks as a reward for his or her good leadership at the discretion of the module's organiser.

Resources

There are plenty of resources on the internet on developing a chatbot, natural language processing and understanding, and train service information. You should research the alternatives and decide what to use yourselves. Below are some that I found to use as a starting point.

On Developing a ChatBot and Expert Systems:

1. IBM: developing a chatbot guide

<https://developer.ibm.com/code/topics/chatbot/>

2. Create a bot with Azure Bot Service

<https://docs.microsoft.com/en-us/azure/bot-service/bot-service-quickstart>

3. How I developed my own learning chatbot in Python from scratch and deployed it on Facebook Messenger!

<https://chatbotlife.com/how-i-developed-my-own-learning-chatbot-in-python-from-scratch-and-deployed-it-on-facebook-88bc828be0a8>

4. ChatBots — The Rise of Conversational UI

<https://chatbotlife.com/chatbots-the-rise-of-conversational-ui-8a59078e2f95>

5. Create your own chat bot in Java using Apache OpenNLP

<http://itsallbinary.com/create-your-own-chat-bot-in-java-using-apache-opennlp-artificial-intelligence-natural-language-processing/>

6. Python rules engines

<http://pyke.sourceforge.net/index.html>

<https://github.com/jruizgit/rules>

On Natural Language Processing and Understanding:

7. Python and Java libraries for NLP

<http://www.nltk.org/>

<https://spacy.io/>

<https://opennlp.apache.org/>

8. Microsoft: Language Understanding Intelligent Service (LUIS)

<https://www.luis.ai/>

<https://azure.microsoft.com/en-us/services/cognitive-services/language-understanding-intelligent-service/>

9. Amazon AWS Comprehend

<https://aws.amazon.com/comprehend/>

10. IBM Watson

<https://console.bluemix.net/developer/watson/dashboard>

On Train Services, Timetable and Ticket Searching, and Booking:

11. Train ticket booking and other services

<https://www.greateranglia.co.uk>

<https://www.southwesternrailway.com>

<https://www.thetrainline.com>

12. Train Timetable and Running Information

<http://www.nationalrail.co.uk/>

<https://www.realtimetrains.co.uk/>

13. Open Rail Data Wiki

<https://wiki.openraildata.com/index.php>

For making a video

You can use any tools you prefer.

There is one available in UEA Applications called Camtasia. If you are using a UEA PC attached to the Network, you can download Camtasia straight onto your computer from the UEA Application Catalogue. It is easy to record your screen movements and actions, narrated PowerPoint or to import HD video from a camera or other source.

<https://portal.uea.ac.uk/learning-technology/technologies/camtasia>