COMP3074 Coursework

AN INTERACTIVE NLP-BASED AI SYSTEM

Coursework – an Interactive NLP-based AI system	
Version	2023
Release date	13/10/2023 @ 11h00 (Nottingham time)
Hand-in date	14/12/2023 @ 15h00 (Nottingham time)
Late penalty	5% per working day

In this coursework you will be required to design and implement your own user-facing **interactive NLP-based system**, evaluate its performance and usability, write it up in a report, and make a demo video showing it in action. Using Python and a restricted selection of libraries, you will produce the following three (3) deliverables:

- **Deliverable 1**: an interactive NLP-based AI system, worth **30%** of the grade.
- **Deliverable 2**: a report of up to a **maximum of 3,000 words** (excluding citations, but including tables and figures) describing the architecture, implementation, and evaluation of the system, worth **60%** of the grade.
- **Deliverable 3**: a short video of **no more than 5 minutes** showing your system working, worth **10%** of the grade.

This document will describe each of those three deliverables, explain what they should contain, and how they will be evaluated.

General notes:

1. Submissions need to be in English.
2. The use of offensive terminology in your code, report, or video will not be tolerated and will be penalised by reduction of marks and may be reported for misconduct.
3. The code and report you submit needs to be written by you, or properly cite its sources, and the video also needs to be recorded and narrated by you.

Deliverable 1. Interactive NLP-based AI system

You should build a dialogue-based system which allows a user to complete a **transaction** (see below) with an AI, commonly known as a **chatbot**, or a **conversational AI**. The work will be based on what you have learned, using Python and a restricted set of allowed libraries[1]. A lot of Python libraries can automate much of the process, so it is of the utmost importance that you do not use them. Any part of the assessment that is performed by a library will not be counted towards your grade.

The two key objectives for this coursework are:

- **Natural language processing architecture**: the system needs to include the architectural features that show what you have learned about processing text in lectures and labs.
- **Conversational design**: the system should implement the **conversational design principles** as covered in lectures to ensure a usable and satisfactory user experience of your chatbot.

The system's main functionality should be focused on supporting the user in making a *transaction*, i.e., it should support users in completing a task, such as booking and ordering in a specific domain of your choice. Some examples (not an exhaustive list) of acceptable tasks include:

- Travel booking
- Restaurant booking
- Cinema/theatre/gig ticket booking
- Groceries and home delivery ordering
- Cooked food home delivery ordering
- Cook-along recipe provision
- Uni timetable management
- Calendar management
- Kitchen inventory management
- Playlist management
- Your turn, be creative! If you are in doubt whether your idea satisfies the requirements, ask us in the lab.

1.1 Architecture

This section contains a non-exhaustive list of common tasks in interactive NLP-based systems (chatbots) that you are expected to implement. They are not all equal in complexity or importance, e.g., intent matching is necessary for any multi-usage chatbot while small talk is not.

Expected features:

1. *Intent matching*

Intent matching is a necessary part of any system which aims to do more than one task. It consists in taking in a user input and predicting the intent of the user, i.e., which function of the system the user

[1] Python standard library, numpy, scipy, scikit-learn (and dependencies), nltk **except for** nltk.chat

is intending to use. Such functions could be storing/showing the user's name, outputting the current date and time, starting a transaction dialogue, or anything else described in this document. Intent detection can be done at different degrees of complexity, from simple text matching to similarity-based and machine learning-based approaches.

2. Identity management

Identity management refers to being able to acquire, memorise, and use the name of the person currently using the system. It consists in functions such as detecting name statements (e.g., "my name is X", "call me X", etc.) and storing the name to be used either in basic statements addressing the user, or in explicit name output functions (e.g., "what is my name?").

3. Transactions

A transactional dialogue system is a tool that allows for an artificial system to dialogue with a human operator **in order to achieve a specific set of goals**, which are **transactional** in nature, such as booking a train/plane ticket, a table in a restaurant, ordering an item from a store, etc.

Here is an example of a turn-by-turn dialogue between an operator and a travel booking system, to illustrate the range of tasks that such system would need to accomplish to properly function.

Turn	Author	Text
1	Bot	Hello, welcome to the Skynet Travel Agency, how may I help?
2	User	I would like to book a flight from London to Paris
3	Bot	Would that be a return trip or a single flight?
4	User	A return trip please
5	Bot	When were you thinking of going?
6	User	From the 10th to the 19th?
7	Bot	I'm sorry, I am going to need more details than that – please use the precise dates of your trip and whether you are flexible in your dates
8	User	10/11/2021 to 19/11/2021, and I am flexible
9	Bot	Thank you, I will search my database for available tickets. Please wait.
10	Bot	I have one flight available on the 10th from London to Paris, but no flight from Paris to London on the 19th, would you like to change the return date or book a single flight?
11	User	Change the date to the 20th

4. Information retrieval & question answering

A conversational information retrieval system is an information retrieval tool that interprets a *query* from a dialogue with a user and returns a set of one or more results which are deemed to be relevant, such as **direct answers**, **webpages**, **books**, **recipes**, **songs**, **movies**, etc. It requires having some form of database of documents and an indexing system. A question answering system can be seen as a special case of information retrieval where the documents are question-answer pairs, and only the answer is retrieved.

5. Small talk

A small talk function refers to the ability to have short discussions about shallow topics, such as "hello", "how are you", "how's the weather", etc. There is no expectation of long conversations due to them being one-off topics, but they can either be single turn (question, answer) or multi-turn (question, answer, and one or more follow-ups).

1.2 Conversational Design

Your chatbot should implement the following **conversational design principles** (as covered in lectures). You should also comment on the principles where appropriate in your report and video.

1. **Prompt design**

Have conversational markers and key phrases been used? Are the prompts clear and concise? Are prompts structured to support the user/task?

2. **Discoverability**

Is contextual help provided (e.g., what can the user do next?), and help to discover functionality and content?

3. **Error handling**

What error handling strategies are being used? Are they used effectively?

4. **Personalisation**

Is the experience somehow personalised to the user?

5. **Confirmation**

What confirmation strategies are used? Are these effective?

6. **Context**

Is context tracking used effectively?

1.3 Marking criteria – system

Your system will be marked using mainly the following criteria:

1. **Core architecture / features** (50%)
    a. Structure and commenting
    b. Effect of bugs
    c. Presence of non-referenced code
    d. Quality of implementation
2. **Conversational design** (50%)
    a. Quality of design principles implementation
    b. Conversational user experience

If your system does not work on a standard installation such as the A32 computers or the virtual desktop, or if it does not make use of natural language processing as taught in the lectures and labs, you will get 0 marks.

Deliverable 2. **Report**

You should write a report explaining and critically analysing your system, using the following structure:

1. **Introduction** – introduce your system briefly.
2. **Chatbot Architecture** – technical description of your system. At least three points should be discussed: **functionality** (what it does), **implementation**: how you built it, **justification**: why you built it that way. Necessary images, flowcharts, and general illustrations[2] are welcome. Reference relevant parts of code in the report (if short), or in the project files (if long).
3. **Conversational Design** – describe how you implemented the Conversational Principles in your chatbot.
4. **Evaluation** – how you tested your project with users and technically, i.e., **usability testing** and **performance testing**.
5. **Discussion** – make use of the Responsible Research and Innovation framework to reflect on the **results of your evaluation** and **your chatbot's application area**.

I strongly encourage you to follow this structure if you are not an experienced writer of scientific reports (and you probably are not).

2.1 Length and format

The report should me a maximum of 3,000 words (+/-10%). We recommend that the sections on Chatbot Architecture and Conversation Design together should have about 2/3 of the words overall.

There is no strict minimum number of words, but the report should probably be near the 3,000 words +/- 10% limit to get a good mark.

Use the single column ACM template available for Microsoft Word, LaTeX (use `sample-manuscript.tex` for submissions), and Overleaf.

Your submission should be a single PDF document.

2.2 Marking criteria – report

Your report will be evaluated across the following criteria.

1. **Description of the system (40%)**
    a. Software architecture
    b. Conversational design
2. **Evaluation (30%)**
    a. Usability
    b. Performance
3. **Discussion (20%)**
    a. Reflection on your results using Responsible Research and Innovation (RRI)
    b. Reflection on your application area using RRI
4. **Writing quality (10%)**
    a. Sensible structure and presentation

[2] https://www.diagrams.net is a great tool to make all sorts of charts. Hand drawn diagrams are not acceptable in 2023.

b. Writing clarity
c. Grammar and typos
d. Sensible use of diagrams and screenshots

If your report does not open or is not readable in a standard PDF reader or Microsoft Word, you will get 0 marks. It is your responsibility to check that your uploaded document is readable.

If your report describes a system which is not what you actually implemented, you will also get 0 marks.

Deliverable 3. Demonstration video

You should produce a short demonstration video of a up to five minutes, that demonstrates your system in use. This can be achieved by screen capture of you using the chatbot live. The video should have some form of audio narration of you explaining what the chatbot is doing. In the video, you should demonstrate the different features you implemented, and the conversational design principles you followed.

It is not sufficient to show/narrate your code in a text editor or using slides to explain your chatbot. Show the system as you use it, with you explaining what is happening in your own words. Make sure the audio of the video is understandable and not sped up to the point of unintelligibility.

3.1 Marking criteria – video

Your video will be evaluated in a pass/fail/distinction manner with penalties. If your video adequately shows your project working along with a narrated explanation of how it works, you will get 10 marks, subject to the following penalties:

3. You did not cover all features: -5 marks
4. Your video was hard to follow, or hard to understand, or there is no sound: -5 marks.

If your video does not run, you will get 0 marks. Please use a sensible video format.

Offensive language in your video will not be tolerated and lead to a penalty of -10 marks.

Marking scheme

Deliver-able		Poor (<30%)	Lacking (>=30%)	Passable (>=50%)	Good (>=60%)	Very good (>=70%)	Outstanding (>=80%)
Report (60%)	Description of the architecture and conversational design (40%)	Missing section(s) or poor, insufficient description.	The description of the architecture and/or conversational design is lacking (for example, it is unclear or not explained what has been done).	Both architecture and conv. design are adequately described, but there are shortcomings in description of functionality, implementation, or justification.	Both architecture and conv. design are well described, but there may be some improvement of the description of functionality, implementation, or justification.	Both architecture and conv. design are very well described, and no improvement are needed of the description of functionality, implementation, and justification.	Very effective description of the functionality, implementation, and justification of the architecture and conversational design written to the standard of a scientific paper.
	Evaluation (30%)		The usability and/or performance testing is lacking.	Both usability and performance testing is adequate, but there are clear shortcomings.	Both usability and performance testing is good, but there but there is room for some improvement.	Both usability and performance testing is very good, there is no need for any improvement.	Both usability and performance testing has been done to near-publishable standard.
	Discussion (20%)		The discussion does not make use of RRI or does not reflect on the results of your evaluation or your chatbot's application area.	The discussion makes use of RRI, attempts to reflect on the results of the evaluation and the application area but has clear shortcomings.	The discussion makes uses RRI effectively to reflect on the results of the evaluation and the application area but has some room for improvements.	The discussion makes uses RRI effectively to reflect on the results of the evaluation and the application area and there is no need for improvements.	The discussion makes uses RRI effectively to reflect on the results of the evaluation and the application area and is of near-publishable quality.
	Quality of writing and presentation (10%)			Passable attempt with multiple mistakes or typos, or structural issues. There might be misuse of jargon or citations.	Good writing with a few mistakes or typos, or structural issues. There might be misuse of jargon or citations.	Excellent writing, few, or no typos/mistakes. Well-presented and structured paper, using scientific jargon correctly, using citations correctly.	Writing quality is comparable to a peer-reviewed research paper

Chatbot in Python (30%)	Core architecture (50%)	The system contains code from unreferenced sources, or does not run.	The code is poorly structured and lacking your comments, the system has many bugs.	The architecture is basic but functional, it implements the expected features and follows basic but adequate coding practices.	The architecture is good, it implements the expected features well and generally follows good coding practices.	The architecture is very good, exceeding expected features, and following good coding practices throughout.	The architecture is excellent, with a high degree of originality, the coding practices are professional.
	Conversational design (50%)	Conversational principles have not been implemented.	Conversational principles have been implemented poorly and some are missing.	Conversational principles have been mostly implemented but in a basic or lacking way.	Conversational principles have all been implemented well.	Conversational principles have all been implemented very well.	Conversational principles have all been implemented very well, and some in a creative or comprehensive way.
Demo video (10%)	Demo video quality (10%)	The video is missing, corrupt, irrelevant, or incomprehensible	The video is poor or missing required content	The video is basic or lacking in some respects	The video is done satisfactorily or well, and provides a good overview of the system and the design principles	The video is very well done, provides a very good overview of the system and how it effectively implements the design principles	The video is very creative, provides a comprehensive overview of the system and how it excels at the design principles

Each of those criteria will be evaluated on a scale of 0-100, with the following indicative bands:

5. 0-29 -> Failing attempt.
6. 30-39 -> Marginal fail.
7. 40-49 -> Low pass (UG) / soft fail (PGT).
8. 50-59 -> Adequate to good.
9. 60-69 -> Good to very good.
10. 70-79 -> Very good to excellent.
11. 80-89 -> Outstanding, potentially publishable.
12. 90-100 -> Exceptional, publishable ‘as is’.