NOTTINGHAM TRENT UNIVERSITY

School of Science and Technology

COURSEWORK ASSESSMENT SPECIFICATION

MODULE CODE	:	ISYS30221
MODULE TITLE	:	Artificial Intelligence
MODULE LEADER	:	Dr Benjamin Inden
TUTOR(S)	:	N/A
TITLE	:	Coursework
LEARNING OUTCOMES ASSESSED	:	K1, K2, K3, S1, S2, S3
CONTRIBUTION TO ELEMENT	:	1 of 2 (50% of module marks)
DATE SET	:	Wednesday 09th October 2019 (Teaching Week 11)
DATE OF SUBMISSION	:	Friday 27th March 2020 (Teaching Week 35, final part of assessment)
SUBMISSION METHOD	:	Online (NOW / TurnitIn)
FEEDBACK DATE	:	Wednesday 22nd April 2019 (Teaching Week 39, final part of assessment)
FEEDBACK METHOD	:	Online (NOW)
NOTE	:	The usual University penalties apply for late submission and plagiarism. Please consult your student handbook for further details.

I. Assessment Requirements

PLEASE NOTE: THIS IS STILL A DRAFT COURSEWORK BRIEF UNTIL THE EXTERNAL EXAMINER APPROVES IT. SHOULD ANY CHANGES BE NECESSARY AT A LATER STAGE, THEN THESE WILL BE MANAGED SUCH THAT NO DISADVANTAGES ARISE FOR STUDENTS BECAUSE OF THAT.

You are expected to submit a "lab book", that is a collection of design documents, software, and test results, related to the design of a chatbot. The chatbot must be modular and demonstrate the use of several major AI techniques, as well as their integration into one user friendly system.

The lab book is to be submitted on NOW (electronic submission only) in several stages over the academic year as detailed in the following table.

Stage	Target completion date	Tasks
1	Tue 05/11/2019	Chatbot design planning document Chatbot with rule-based and similarity-based component Conversation log
2	Tue 10/12/2019	Chatbot extended with image classification component Conversation log
3	Tue 25/02/2020	Chatbot extended with toy world reasoning system Conversation log
4	Fri 27/03/2020	Chatbot extended with sequence to sequence NN and reinforcement learning demonstration Conversation log Final chatbot design specification and reflective document

Make sure that the chatbot topic, as well as the individual functionalities, are very specific, so it's unlikely anyone else in the class would create exactly the same functionality. You can use other Python packages if they are on pypi.org and you find them useful for your chosen domain of application, but you must not use them as a replacement for any of the specified parts of the assignment (for example, you must use AIML, not any other method to program chatbot responses as a replacement for AIML). Your code must not violate the NTU's Computer Use Policy, NTU's Internet Content and Access Policy, and other relevant laws or university policies (see https://www4.ntu.ac.uk/information_systems/policies/index.html).

II. Assessment Scenario/Problem

Chatbot design planning document / Final chatbot design specification

This should specify the chosen topic area for the chat bot, a more detailed list of requirements, a brief description of the individual modules you intend to create, and a flow diagram showing how these work together to produce an answer to the user's input. Limit this document to two to three pages. For the final specification, update your planning document. The final specification should also a reflective document (about one page), which analyses what went well or not so well, puts the work in context, and mentions some ideas for what could be done if there was more time.

Chatbot with rule-based and similarity-based component

This submission should consist of one Python file that implements the chat bot, one AIML file that implements the rules, and one file in a suitable format (such as plain text or CSV) that has the question-answer pairs. If you wish, you can create more files than that. You can use the Python and aiml files provided on NOW as a starting point, but you should extend and customise them towards your design specification. The entire program might be as compact as 110 lines but can be double that size depending on your programming style and how many features you implement. The similarity-based component should be based on the bag-of-words model, tf/idf, and cosine similarity. The AIML file could be as compact as 80 lines but will probably somewhat longer, maybe even significantly longer depending on the domain you intend your chatbot to be for.

Conversation log

For each conversation log, record an actual conversation between you (and/or your peers) and the chatbot that demonstrates the implemented features. Annotate it with comments that explain which feature / component generated this, and how, for any particularly remarkable output. If the conversation included non-textual input (e.g., images), this should also be included to the degree possible. The conversation logs will not receive a grade by themselves, but are there to demonstrate the respective chatbot components, so their quality influences the component grade.

Image classification component

If the user inputs a question such as "What is img01.jpg"?, the chatbot should invoke a pre-trained convolutional neural network to provide the answer. You decide yourself which network architecture and training data set to use. The possible classes of objects that the network can recognise depend on the domain of the chatbot that you have chosen. If you wish, you can decide to use a different source of images instead of files, e.g., camera input, and also decide to trigger the network in other ways. You will need to train your neural network using a data set with class labels. There are web sites that serve as a repository for many good data sets such as Kaggle.com or the UCI Machine Learning Repository. If you find it hard to find a fitting data set, you might have to change the function specification of this component --- don't worry too much if it doesn't fit that well with the remainder of the chatbot, the priority is to get a working system without spending too much time on selecting a data set. For creating the convolutional neural network, you must use the keras library. For loading and pre-processing the data, scikit-learn is recommended. It is also recommended that you follow closely one of the tutorials available on the keras or tensorflow websites (with proper referencing). One key challenge is to make it so that the trained model can be saved and later used for classification when the chatbot is running.

Toy world reasoning system

Use the provided NLTK based code that translates natural language sentences into first-order logic, and does some reasoning based on its knowledge. Adapt it to your chosen domain by adding new objects and relations (and possibly removing old ones). Connect it to the chatbot such that it is activated when certain key words or phrases appear, adds knowledge from user input to its knowledge base, and does some reasoning, for example, as response to a user question. Note that this is only a toy world related system, in other words, the number of objects and relations can remain small, there should be at least as many as in the example provided, but not normally more than twice those numbers.

Sequence to sequence network extension

Build a sequence to sequence network, and use it in some way to generate chatbot responses (for example, if the other modules cannot produce detect any pattern in the input. Follow the provided tutorials to build the network, and train it with one of the data sets of conversations that you will be pointed to. Make sure that the trained network is saved, and loaded again when the chatbot runs. Note that as there is only a limited set of

conversation data sets available for training, some overlap with other students' work is unavoidable, but you must still make your own decisions about the architecture and parameters of the neural network and training procedure.

Reinforcement learning extension

Use a reinforcement learning algorithm of your choice (either based on deep learning, or on evolutionary algorithms) to train an agent that then becomes part of your chatbot. Depending on the chatbot you are building, this agent could be playing a game via console or a separate window (if you want to use OpenAI Gym), or solve some optimisation problem (this might be most suitable if you use an evolutionary algorithm). It is again recommended that you follow a tutorial (with proper referencing), and/or use the examples provided in this module as a starting point. Again, as there is only a limited set of OpenAI environments available for training, some overlap with other students' work might be unavoidable, but you must still make your own decisions about the architecture and parameters of the neural network and training procedure.

III. Assessment Criteria

Task	**1st**	**2:1**	**2:2**	**3rd**	**Fail**
Chatbot planning document 10%	All information for a design specification is present and completed to an excellent standard; (highly) original design.	All information for a design specification is present and completed to a very good standard; a design that seems well adapted to a particular domain of application.	Most information for a design specification is present and completed to a good standard; the design has been adapted to a particular domain of application with some errors / emissions	Some meaningful information for a design specification is present; the design has some merits but is not well adapted to a particular domain, or lacking many details.	(Very) few information about the design, design severely flawed and/or incomplete.
Chatbot with rule-based and similarity-based component 15%	Excellent programming style, functionality, and user interface design as evidenced by both source code and conversation log. The components make a perfect fit for the chosen domain. Methods used go beyond taught methods; excellent knowledge and understanding of the area of study is revealed.	Very good programming style, functionality, and user interface design as evidenced by both source code and conversation log. The components make a very good fit for the chosen domain. Methods used may sometimes go beyond taught methods; very good knowledge and understanding of the area of study is revealed.	Good programming style, functionality, and user interface design as evidenced by both source code and conversation log. The components fit somehow to the chosen domain and have a good functionality with some errors or omissions. Good knowledge and understanding of the area of study is revealed.	Programming style, functionality, and/or user interface design are sufficient to achieve the basic goals as evidenced by both source code and conversation log. The components may not fit well to the chosen domain, contain a number of bugs, and/or have limited functionality. Knowledge and understanding is sufficient to deal with terminology, basic facts and concepts but fails to make meaningful synthesis.	Programming style, functionality, and/or user interface design are insufficient. Component may be severely incomplete or critical mistakes made. (Highly) insufficient knowledge and understanding of the area of study.

Image classification component 15%	Excellent programming style, functionality, and user interface design as evidenced by both source code and conversation log. The component makes a perfect fit for the chosen domain. Methods used go beyond taught methods; excellent knowledge and understanding of the area of study is revealed.	Very good programming style, functionality, and user interface design as evidenced by both source code and conversation log. The component makes a very good fit for the chosen domain. Methods used may sometimes go beyond taught methods; very good knowledge and understanding of the area of study is revealed.	Good programming style, functionality, and user interface design as evidenced by both source code and conversation log. The component fits somehow to the chosen domain and has a good functionality with some errors or omissions. Good knowledge and understanding of the area of study is revealed.	Programming style, functionality, and/or user interface design are sufficient to achieve the basic goals as evidenced by both source code and conversation log. The component may not fit well to the chosen domain, contain a number of bugs, and/or have limited functionality. Knowledge and understanding is sufficient to deal with terminology, basic facts and concepts but fails to make meaningful synthesis.	Programming style, functionality, and/or user interface design are insufficient. Component may be severely incomplete or critical mistakes made. (Highly) insufficient knowledge and understanding of the area of study.
Toy world reasoning system 15%	Excellent programming style, functionality, and user interface design as evidenced by both source code and conversation log. The component makes a perfect fit for the chosen domain. Methods used go beyond taught methods; excellent knowledge and understanding of the area of study is revealed.	Very good programming style, functionality, and user interface design as evidenced by both source code and conversation log. The component makes a very good fit for the chosen domain. Methods used may sometimes go beyond taught methods; very good knowledge and understanding of the area of study is revealed.	Good programming style, functionality, and user interface design as evidenced by both source code and conversation log. The component fits somehow to the chosen domain and has a good functionality with some errors or omissions. Good knowledge and understanding of the area of study is revealed.	Programming style, functionality, and/or user interface design are sufficient to achieve the basic goals as evidenced by both source code and conversation log. The component may not fit well to the chosen domain, contain a number of bugs, and/or have limited functionality. Knowledge and understanding is sufficient to deal with terminology, basic facts and concepts but fails to make meaningful synthesis.	Programming style, functionality, and/or user interface design are insufficient. Component may be severely incomplete or critical mistakes made. (Highly) insufficient knowledge and understanding of the area of study.

Sequence to sequence network extension 15%	Excellent programming style, functionality, and user interface design as evidenced by both source code and conversation log. The component makes a perfect fit for the chosen domain. Methods used go beyond taught methods; excellent knowledge and understanding of the area of study is revealed.	Very good programming style, functionality, and user interface design as evidenced by both source code and conversation log. The component makes a very good fit for the chosen domain. Methods used may sometimes go beyond taught methods; very good knowledge and understanding of the area of study is revealed.	Good programming style, functionality, and user interface design as evidenced by both source code and conversation log. The component fits somehow to the chosen domain and has a good functionality with some errors or omissions. Good knowledge and understanding of the area of study is revealed.	Programming style, functionality, and/or user interface design are sufficient to achieve the basic goals as evidenced by both source code and conversation log. The component may not fit well to the chosen domain, contain a number of bugs, and/or have limited functionality. Knowledge and understanding is sufficient to deal with terminology, basic facts and concepts but fails to make meaningful synthesis.	Programming style, functionality, and/or user interface design are insufficient. Component may be severely incomplete or critical mistakes made. (Highly) insufficient knowledge and understanding of the area of study.
Reinforcement learning extension 15%	Excellent programming style, functionality, and user interface design as evidenced by both source code and conversation log. The component makes a perfect fit for the chosen domain. Methods used go beyond taught methods; excellent knowledge and understanding of the area of study is revealed.	Very good programming style, functionality, and user interface design as evidenced by both source code and conversation log. The component makes a very good fit for the chosen domain. Methods used may sometimes go beyond taught methods; very good knowledge and understanding of the area of study is revealed.	Good programming style, functionality, and user interface design as evidenced by both source code and conversation log. The component fits somehow to the chosen domain and has a good functionality with some errors or omissions. Good knowledge and understanding of the area of study is revealed.	Programming style, functionality, and/or user interface design are sufficient to achieve the basic goals as evidenced by both source code and conversation log. The component may not fit well to the chosen domain, contain a number of bugs, and/or have limited functionality. Knowledge and understanding is sufficient to deal with terminology, basic facts and concepts but fails to make meaningful synthesis.	Programming style, functionality, and/or user interface design are insufficient. Component may be severely incomplete or critical mistakes made. (Highly) insufficient knowledge and understanding of the area of study.

Final chatbot design specification and reflective document 15%	All information for a design specification is present and completed to an excellent standard; (highly) original design. Very insightful reflections that connect the project to its scientific background, and the real world, including PSEL issues. Excellent critical evaluation.	All information for a design specification is present and completed to a very good standard; a design that seems well adapted to a particular domain of application. Insightful reflections that make some connections to the scientific background, and the real world, including PSEL issues. Very good critical evaluation.	Most information for a design specification is present and completed to a good standard; the design has been adapted to a particular domain of application with some errors / emissions. Some insightful reflections on how the project went, and its connection to the outside world.	Some meaningful information for a design specification is present; the design has some merits but is not well adapted to a particular domain, or lacking many details. Reflections mention some correct but perhaps relatively obvious points, and/or are rather brief.	(Very) few information about the design, design severely flawed and/or incomplete. Reflections (highly) superficial and/or incomplete.

IV. Feedback Opportunities

Formative (Whilst you're working on the coursework)
Your tutor will want to see your chatbot working at various points during the year. You will frequently be given informal verbal feedback regarding your performance on tasks relating to the coursework assessment during laboratory sessions. Attendance is therefore important for your development and thus coursework success.

Summative (After you've submitted the coursework)
You will receive specific feedback regarding your coursework submission together with your awarded grade when it is returned to you. Clearly, feedback provided with your coursework is only for developmental purposes so that you can improve for the next assessment or subject-related module.

V. Moderation

The Moderation Process
All assessments are subject to a two-stage moderation process. Firstly, any details related to the assessment (e.g., clarity of information and the assessment criteria) are considered by an independent person (usually a member of the module team). Secondly, the grades awarded are considered by the module team to check for consistency and fairness across the cohort for the piece of work submitted.

VI. Aspects for Professional Development Portfolio

The final chatbot design specification, software, and conversation log would also make god additions to your professional development portifolio, and/or to demonstrate to future employers. That way, you will evidence skills that are in very high demand both in industry and in academia. For example, a recent US-based survey found that there were around 144,000 AI-related job openings in industry, but only about 26,000 developers and specialists seeking work. The situation in the UK is very similar. You can read more details at https://www.nature.com/articles/d41586-019-01248-w.