Topics in Human Computer Interaction
Project (40%) – Small Chatbot
Due: July 31, 2015

The objective of this assignment is to implement a small chatbot using the dialogue markup language *AIML* (*Artificial Intelligence Markup Language*). Ideally, your chatbot should be a consultant who is helping people cut down their energy consumption. However, you can select a different topic. Regardless of the topic, the chatbot should have a personality and a life, e.g., in the energy domain, it should live in a house/apartment, have a family, pets, etc, have likes and dislikes, and have specific needs and costs.

The main focus of the project is on dialogue management issues, such as focus handling, clarification sub-dialogues, and possibly handling non-cooperative behaviour. While it is important to provide useful information, you shouldn't spend all your time acquiring domain expertise. Your chatbot should also be able to chit-chat, and both ask and answer questions. It should be able to handle at least 10 dialogue turns, and answer at least 20 different questions in a particular topic, and possibly provide links to useful web sites.

Below is a list of preliminary tasks you can do in order to get an early start on the project.

1. Install the environment

1. Install Python 3.5 (or higher) on your machine.
2. Make sure that Python is installed correctly by opening a Terminal and typing
   `python --version`

   If installed correctly, this will print the version number of your Python installation, which should be 3.5 or higher.
3. Download `pyaiml3-master.zip` from the course website.
4. Extract the downloaded zip file to a directory of your choice. A new sub-directory called `pyaiml3-master` will be created there.
5. Download the files `testTK.py` and `test.aiml` into the same sub-directory (`pyaiml3-master`).
6. Test whether `pyaiml` is installed correctly by typing in the terminal:

   `python testTK.py`

   If installed correctly, this will print "Loading test.aiml... done (X.XX seconds)". It will also print a second line containing a prompt like this:

   `aiml>`

   In this line, type "hello" and hit enter. If installed correctly, the response will be "world" followed by another prompt. Type CTRL-D to break out of the AIML prompt.

2. Get acquainted with AIML and the pyaiml interpreter

1. Read the AIML overview at `www.pandorabots.com/pandora/pics/wallaceaimltutorial.html`, and the AIML 1.0.1 specifications in `alicebot.org/TR/2011/`. Additional information appears in `www.alicebot.org/documentation/` (all these links are in the course website).

2. Read the contents of the file `test.aiml`. Make sure you understand what each line means. Just to really make sure, edit `test.aiml` so that if you save the file and repeat Step 6 above, the word "hi" is printed instead of "world". Then edit `test.aiml` again so that the system says "hi" as an answer to two additional greetings (other than "hello"). Use AIML's <srai> to implement that functionality.

Note: you can also run the system using `test-startup.py`, which calls `std-startup.xml` to load the functions in `standard/`. Unfortunately, some of these functions have bugs, so use at your own risk.

3. Design a dialogue system for a "save energy at home" domain (or a different domain)

Refer to the Iterative Development Cycle slide (# 7) and the Chatbot Building Tips slides (# 49-51) in LN2 (DS Development), and follow the steps:

1. **Task analysis:** If your chatbot is in the energy domain, visit the site `switchon.vic.gov.au/more-ways-to-save/top-10-ways-to-save-power`, and inform yourself about the domain. Extend your knowledge by reading further resources related to saving energy at home.

2. **Scenarios:** This is a suggestion to generate some scenarios, but you can envisage other options. Imagine an energy consultant whose job it is to help people save energy at home, and three people with different energy consumption patterns: (a) a student who likes video games; (b) a house-holder with three daughters aged 5, 6 and 11; (c) a self-employed engineer who has a workshop in his home.

   Write on paper what a short dialogue between each of these people and an energy consultant would look like. The setting should be that all three want to save energy but are unsure how. The consultant tries to find out by talking to them (through questions) how they can save energy at home. For instance:
   STUDENT: Hello.
   CONSULTANT: Hello, how may I help you?
   STUDENT: My last energy bill was way too high and now I'm trying to find a way to get the costs down.
   CONSULTANT: I see. No worries. We can help you with that.
   STUDENT: Great.
   CONSULTANT: Let's see. How many hours per day do you usually spend at home?
   ...

3. Use your imagined dialogues to **determine the objects and actions in the domain**, and map them to the AIML topics and the actions performed in each topic.

4. No simulation environment.

5. Employ **Distilled Human-Human dialogues or Wizard of Oz experiments:** You can conduct a conversation with some people about the topic you have picked. Ideally, you should do this over a chat channel, so that you can log the conversation. Feel free to guide the conversation in a direction your system will know about.

6. **Analyze the log files** to determine whether you have missed any obvious objects or actions.

7. You can now use your (very small) collected **corpus** to develop a lexicon and grammar. You may find it helpful to develop a formal dialogue grammar, but it is not strictly necessary. In AIML, the lexicon and grammar are implemented in the topics, categories, patterns and templates.
   Note: If you have a similar topic to that of other people, you can generate a collective corpus that everybody can use.

8. **Implement your prototype system (Sections 4 and 5).**

9. No additional data structures to those provided by AIML.

10. **Verify** that your chatbot is generally behaving as expected. Plan for some variations of utterances, but you don't need to plan for all possible combinations of things people could say. Plan for a catch-all phrase such as "tell me more about XXX".

11. **Test with real users** – no less than 5 people, but 10 people would be better.

12. **Perform error analysis** – this means that you should log things such as:

    - length of a dialogue (number of turns);
    - number of inappropriate utterances – you can detect them by letting the user enter a particular symbol, such as an exclamation mark, whenever a response doesn't make sense;
    - user satisfaction (with the obtained information for the task, and with the dialogue) – e.g., on a Likert scale from -1 for dissatisfied, 0 for neutral, 1 for satisfied. You can also ask specific questions regarding, for example, the naturalness and repetitiveness of the dialogue.

4. Implement the basic dialogue system

Follow the steps suggested in slides 52-54 of LN2 (DS Development). You should conduct at least two preliminary evaluation sessions with test subjects, so that you can make some changes based on the feedback you receive.

Your chatbot should have the following specific capabilities:

1. Pick up certain answers from its conversational partner, including his/her name, and use them in its own response.

2. Handle different ways of saying the same thing (use <srai> for this).

3. Add dialogue history using <that> or <topic>. The chatbot should be able to answer at least three consecutive questions in a domain.

4. Not repeat the same question after it has received an answer.

5. Not repeat the same information. Indicate that an answer has been provided to a repeated question.

6. Use conditional random responses that make sense.

7. Generate a log file.

5. Implement refinements to the dialogue system

You may decide yourself how to further improve the chatbot, e.g., handle non-cooperative behaviour; create a nice environment for the chatbot using, for example, an avatar; employ learning strategies to create the data base, make the chatbot more robust, or have the chatbot learn from the interaction. **One of these improvements is sufficient. Additional improvements will earn bonus marks.**

6. Evaluation

Part of the evaluation process has been described in Steps 11-12 in Section 3. In addition, you should write a brief report that includes the following:

- What is the initiative and complexity of the dialogue?
- Which dialogue acts can your chatbot generate and understand?
- Which grounding actions can your chatbot handle?
- Which types of sub-dialogues (e.g., clarification, digression) can your chatbot handle? To what extent does it maintain a coherent dialogue structure?
- Explain why you used or did not use a grammar. If you used a grammar, describe it.
- If you have enough users (10 or more), the insights obtained from PARADISE regarding the factors that influence your chatbot's performance. Otherwise, report on the individual performance factors.

7. Marking scheme

- (65%) Performance
  - (20%) Appropriateness of scenario and variety of dialogue phenomena being handled.
  - (45%) Dialogue behaviour (combines accuracy, variability and annoyance), likeability and habitability.
- (20%) Evaluation with users (with iterative development).
- (15%) Report, including error analysis.