



THE UNIVERSITY *of* EDINBURGH

THE LEARNING, TEACHING AND WEB SERVICES  
DIVISION (LTW)

TEACHERBOT PROJECT

---

# Digital Day of Ideas Workshop Exercises

---

*Author:* Kathrin HAAG

## Exercise 1: Creating a Teacherbot on the Playground

- (a) Go to the Pandorobot Playground website at <https://playground.pandorabots.com> and register for an account. Once you have done this, click on **My Bots** in the blue header menu and create a new bot by clicking on **Create Bot**. Bot names must be written in lowercase.
- (b) Click on the name of the Teacherbot you just created and inspect the editor that shows up. We will work within the **Files** and **Train** windows.

## Exercise 2: Adding knowledge to your Teacherbot

- (a) Go to the **Train** tab in the editor. Type in a question into the empty box next to **Ask** and click **Ask**. Inspect the result.
- (b) In order to change the Teacherbot's response, click on **Advanced Alter Response**. Remove the default answer from the **Template** box and add your own response instead. Click **Submit**. **Note:** We are building a bot for Twitter which only accepts a character count of up to 140. To check the length of your response you can use <http://www.twitter-character-counter.com/>.
- (c) Test your new response. Click **Reset Bot Memory** and ask the exact same question again. Can Teacherbot understand it now?
- (d) Click on the **Files** tab and look for the file named **pand\_learn**. It is written in a scripting language called **AIML**. Click on it and inspect the content of the file. Can you figure out how question/answer pairs are constructed?
- (e) Try to construct a question/answer pair within the AIML file. Notice that the question (i.e. pattern) should be written in uppercase at all times and it shouldn't contain any punctuation.

Advanced

Advanced

### Exercise 3: Wildcards

Wildcards allow for more flexibility in recognising user input. They act as placeholders for an arbitrary number of words. Let's look at the pattern

`^ WHO IS ^ TEACHERBOT ^`

as an example. The wildcard `^` is a placeholder for **zero** or more words (you can also use the wildcard `*` if you want to account for **one** or more words). This pattern will recognise user inputs such as:

- Who is Teacherbot?
- Who is this guy called Teacherbot?
- Tell me, who is this Teacherbot and what does it do?
- ...and many more!

There are two options to complete this exercise:

- (a) Add a new question using the **Train** interface and **Advanced Alter Response** as in Exercise 1, but this time include wildcards in your question.
- (b) Go to the AIML file **pand\_learn** and make your existing patterns more flexible using wildcards. Add new question/answer pairs including wildcards. Don't touch the text between the **template** tags.
- (c) Test your changes using the **Train** interface. **Note:** The user input has to follow the exact order of words as specified in your pattern. Phrases such as *Teacherbot, who is that?* won't be recognised and you have to make a separate question/answer pair for this.

Advanced

## Exercise 4: Synonym sets [Advanced]

Users will ask the same question using many different words. For example, instead of *How can I build a Teacherbot?*, a user might ask

- How can I make a Teacherbot?
- How do I create a Twitterbot?

We can create **synonym sets** and recognise all these questions with a single question/answer pair. **Note:** The pre-requisite for this exercise is that you're happy to code in AIML

- Go to the **Files** tab. Click on the + symbol in the upper left corner of the editor. Create a new file called **build** and select **Set** from the drop-down box. In the **Item** box, add as many words sharing the meaning of **build** as you can think of. You can do the same with **teacherbot** and add a new set with Teacherbot synonyms.
- Go to the AIML file **pand\_learn** and make a pattern using your synonym set(s). In order to refer to your synonym set, you have to use the tags `<set>synonym</set>`. Note the / before the second *set*. The text between the tags is written in lowercase but all other text remains uppercase.  
Example: *HOW CAN I <set>build</set> A TEACHERBOT*
- Make the above example more flexible using wildcards and another synonym set. Come up with your own question/answer pairs using wildcards and synonym sets!

## Exercise 5: Randomised responses

In order to make Teacherbot appear more intelligent, instead of always giving the same answer to a question we can have a list of various answers. The answer that will be given to a user query is picked randomly from this list. Alternative responses are between `<li></li>` tags and these are nested between the tags `<random></random>` within your template. For example, here is a set of randomised responses to the question *How are you?*:

```
<template>
  <random>
    <li>I'm very well. How are you doing?</li>
    <li>Glad to see you.</li>
    <li>Always cheered up when I see you.</li>
  </random>
</template>
```

Make sure that the randomised responses always follow this pattern.

- (a) Add sets of randomised responses to your existing question/answer pairs. Test your changes using the **Train** interface.