

COMS 493

AI, ROBOTS & COMMUNICATION

Agenda

- ▶ Review
- ▶ Chatterbot Competition
- ▶ Preview

Review

Natural Language Processing (NLP)

Computational Linguistics
D. G. BOBROW, Editor

Contextual Understanding by Computers

JOSEPH WEIZENBAUM
Massachusetts Institute of Technology, Cambridge, Mass.

A further development of a computer program (ELIZA) capable of conversing in natural language is discussed. The importance of context to both human and machine understanding is stressed. It is argued that the adequacy of the level of understanding achieved in a particular conversation depends on the purpose of that conversation, and that absolute understanding on the part of either humans or machines is impossible.

We are here concerned with the recognition of semantic patterns in text.

I compose my sentences and paragraphs in the belief that I shall be understood—perhaps even that what I write here will prove persuasive. For this faith to be at all meaningful, I must hypothesize at least one reader other than myself. I speak of *understanding*. What I must suppose is clearly that my reader will recognize patterns in these sentences and, on the basis of this recognition, be able to recreate my present thought for himself. Notice the very structure of the word “recognize,” that is, know again! I also use the word “recreate.” This suggests that the reader is an active participant in the two-person communication. He brings something of himself to it. His understanding is a function of that something as well as of what is written here. I will return to this point later.

Much of the motivation for the work discussed here derives from attempts to program a computer to understand what a human might say to it. Let it be understood, let me state right away that the input to the computer is in the form of typewritten messages—certainly not human speech. This restriction has the effect of establishing a narrower channel of communication than that available to humans in face-to-face conversations. In the latter, many ideas that potentially aid understanding are communicated by gestures, intonations, pauses, and so on. All of these are unavailable to readers of telegrams—be they computers or humans.

Work reported herein was supported (in part) by Project MAC, an MIT research program sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research Contract Number Nonr-4102(04).

474 Communications of the ACM
Volume 10 / Number 8 / August, 1967

Introduction to Communication and AI (Polity Press, 2019)
Uncorrected manuscript - ©2018 David Gunkel

5 Natural Language Processing

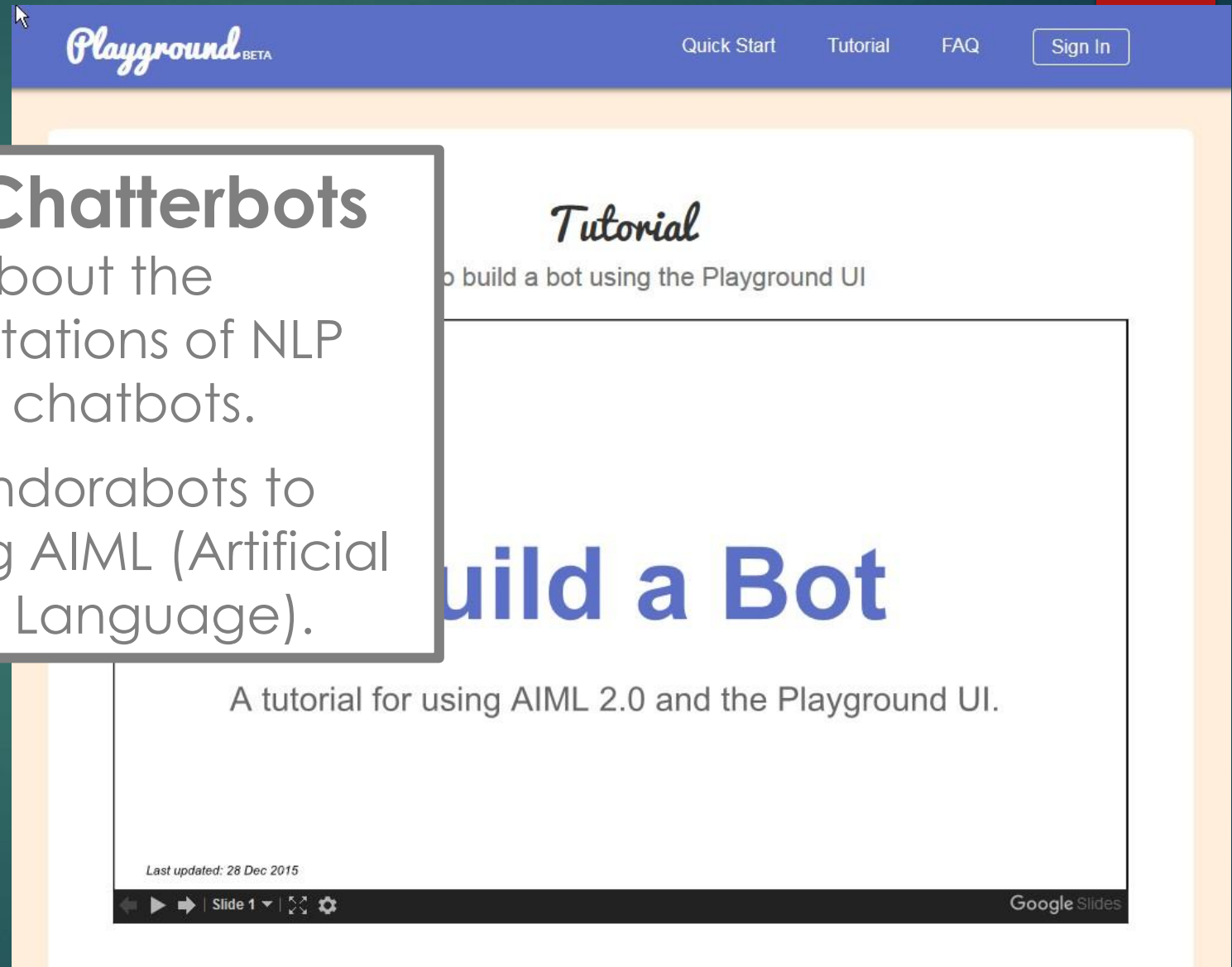
and robots talk. They communicate with us using natural human language. The computer of 2001: A Space Odyssey has conversations with the human crew, identifying and addressing each individual by using their first name. The computer (if the computer is gendered male) participates in a BBC interview, forming a "mutual relationship" with his human companions and even accomplishes some of their tasks. And when things do go wrong—and they do go wrong—dramatically explains himself and even pleads for his own life: "I'm afraid...I'm afraid. I'm afraid, Dave." The robots of science fiction are not only produces intelligible speech, but they also articulate these articulations with gestures, facial expressions, and other human-like behaviors. They are designed to assist humans in working and interacting with the device. Its "vocalizations" may not consist of what we would expect, but the trash-can-looking robot emits a series of electronic sounds that are (within the context of the narrative) clearly expressive of its intentions. It is understood and interpreted for us by the android C3PO.

Creating machines that can talk or communicate with humans using what is called "natural language," has been one of the goals of AI from the very beginning. It was the first item on the list of proposed goals for the Dartmouth summer conference of 1956—"an agenda for how to make machines use language"—it comprised the defining goal of "artificial intelligence" in Alan Turing's agenda-setting paper from 1950 and demonstrated in some of the earliest applications, like the Turing Test program and Terry Winograd's SHRDLU. For this reason, working with natural human language content is not one application of AI, but a general application. In this chapter we will look at Natural Language Processing with two particular implementations—chatbots and spoken dialogue systems.

chatbot, which, in turn, is derived from a concatenation of the words "chat" and "robot" (Ellis 2010, 77). Bots, therefore, consist of a chunk of software designed to accomplish some particular routine task automatically and autonomously. And the virtual spaces of the internet are crawling with them, so much so that bot activity now accounts for over 50 percent of all traffic on the internet (Zeifman 2017). There

Review

<https://home.pandorabots.com>

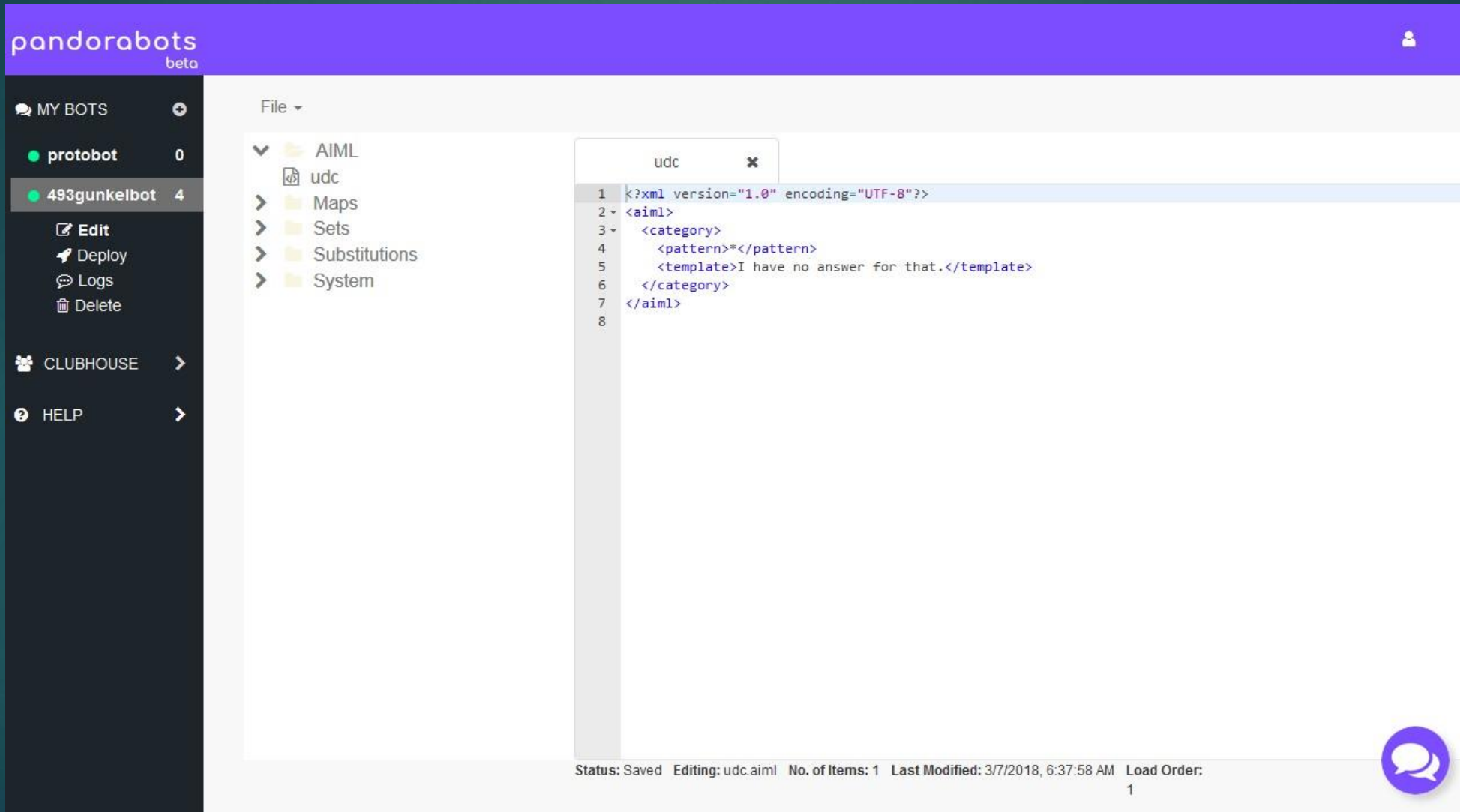


Experiment with Chatterbots

Objective – Learn about the capabilities and limitations of NLP by building our own chatbots.

Procedure – Use Pandorabots to program a bot using AIML (Artificial Intelligence Markup Language).

Review - AIML



The screenshot displays the 'pandorabots beta' web interface. On the left is a dark sidebar with navigation options: 'MY BOTS' (containing 'protobot' with 0 items and '493gunkelbot' with 4 items, plus 'Edit', 'Deploy', 'Logs', and 'Delete' actions), 'CLUBHOUSE', and 'HELP'. The main area shows a file explorer with a tree view containing folders for 'AIML', 'Maps', 'Sets', 'Substitutions', and 'System', and a file named 'udc'. The 'udc' file is open in an editor window titled 'udc', showing the following XML code:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <aiml>
3   <category>
4     <pattern>*</pattern>
5     <template>I have no answer for that.</template>
6   </category>
7 </aiml>
8
```

At the bottom of the editor, a status bar reads: 'Status: Saved Editing: udc.aiml No. of Items: 1 Last Modified: 3/7/2018, 6:37:58 AM Load Order: 1'. A purple chat bubble icon is located in the bottom right corner.

Review - AIML

Artificial Intelligence Markup Language (AIML)

History of AIML

The **XML dialect** called AIML was originally developed by [Dr. Richard Wallace](#) and a worldwide [free software community](#) between 1995 and 2002. AIML formed the basis for what was initially a highly extended [Eliza](#) called "[A.L.I.C.E.](#)" ("Artificial Linguistic Internet Computer Entity"), which won numerous awards. Because the A.L.I.C.E. AIML set was released under the [GNU GPL](#), and because most AIML interpreters are offered under a [free or opensource](#) license many "Alicebot clones" have been created based upon the original implementation of the program and its AIML knowledge base.

The Pandorabots platform implements and supports development of the AIML open standard. It is the most popular open standard scripting language for creating chatbots available on the market to date. A number of AIML interpreters have been written in a variety of languages and open sourced, and a number of other bot platforms support AIML explicitly or under the hood.

Review - AIML

AIML Basics

```
<category>  
  <pattern>HELLO</pattern>  
  <template>Hi there!</template>  
</category>
```

The basic unit of knowledge in AIML is called a **category**.

Each category consists of an input question, an output answer, and an optional context.

The question, or stimulus, is called the **pattern**. The answer, or response, is called the **template**.

Review - AIML

AIML Basics

```
<category>
```

```
  <pattern>HELLO</pattern>
```

```
  <template>Hi there!</template>
```

```
</category>
```

The AIML **pattern** language is simple, consisting only of words, spaces, and wildcard symbols like *.

The words may consist of letters and numerals, but no other characters (no punctuation).

The pattern language is not case sensitive.

Review - AIML

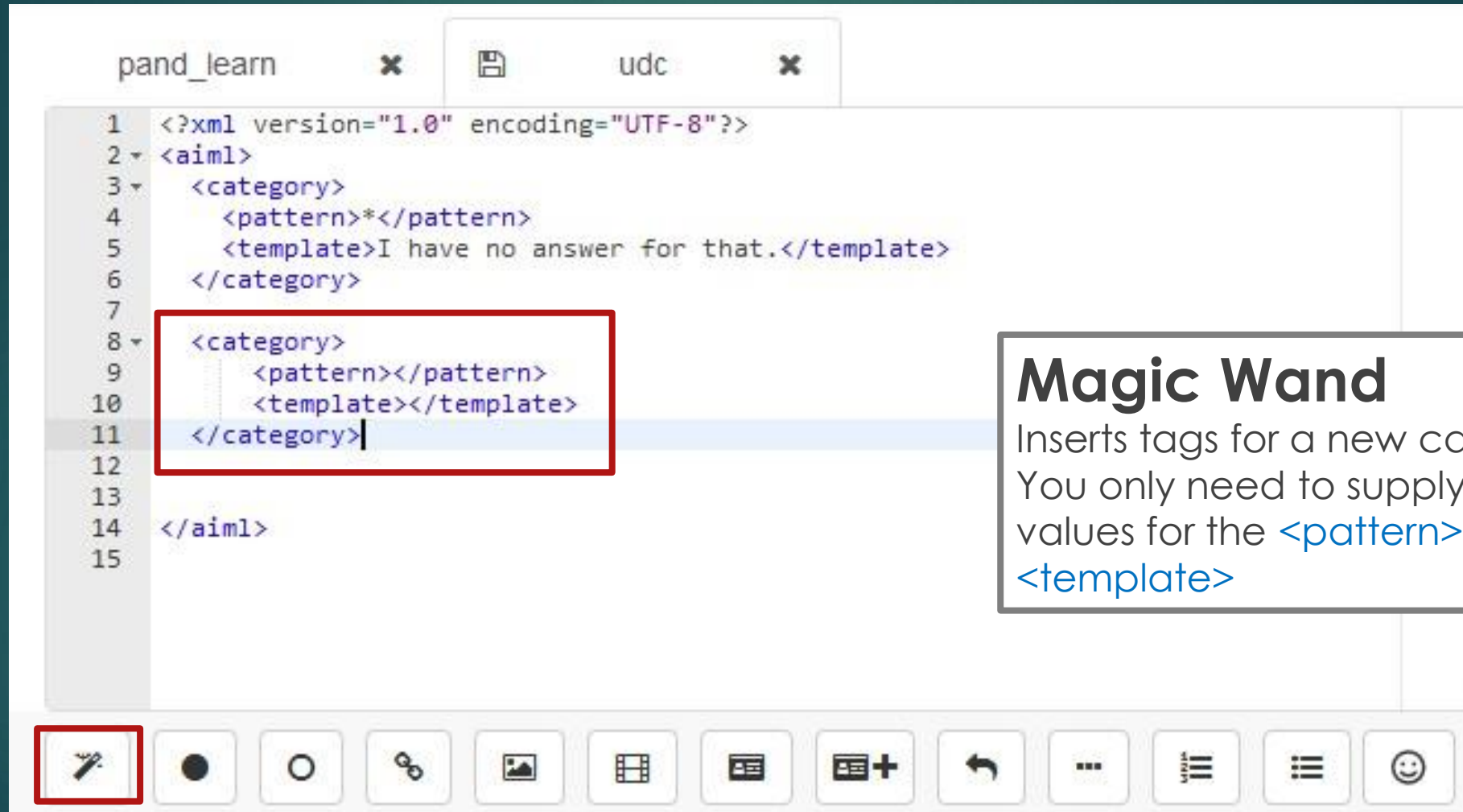
AIML Basics

```
<category>  
  <pattern>HELLO</pattern>  
  <template>Hi there!</template>  
</category>
```

Template defines the bot's response to the matched pattern.

Case does matter in the template! Your bot's response will be returned to the user exactly as written between the template tags.

Review - AIML



The screenshot shows a code editor with two tabs: 'pand_learn' and 'udc'. The code in the editor is as follows:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <aiml>
3   <category>
4     <pattern>*</pattern>
5     <template>I have no answer for that.</template>
6   </category>
7
8   <category>
9     <pattern></pattern>
10    <template></template>
11  </category>|
12
13
14 </aiml>
15
```

The code on lines 8-11 is highlighted with a red box. The toolbar at the bottom has a 'Magic Wand' icon (a wand) highlighted with a red box.

Magic Wand

Inserts tags for a new category. You only need to supply the text values for the `<pattern>` and the `<template>`

Review - AIML

```
<category>
  <pattern>*</pattern>
  <template>
    <random>
      <li>What was that?</li>
      <li>I don't understand</li>
      <li>Can you say that more clearly?</li>
    </random>
  </template>
</category>
```

Randomized Responses

Review - AIML

Wild Cards

```
<category>
```

```
  <pattern>MY NAME IS *</pattern>
```

```
  <template>                </template>
```

```
</category>
```

The * captures 1 or more words in the user input.

This pattern would match all of the following inputs:

Hello there!

Hello Kathy.

Hello my good friend.

The ^ captures 0 or more words in the user input. So it would capture all of the above plus "My name is"

Review - AIML

Using the Wild Card

```
<category>  
  <pattern>MY NAME IS *</pattern>  
  <template>Hi, <star/></template>  
</category>
```

Echo the word(s) captured by the * wild card with the **<star/>** tag.

This tag inserts what ever word or words were captured from the user input with the *

USER: My Name is Zenon.

BOT: Hi, Zenon.

Review - AIML

Variables (Predicates)

```
<category>  
  <pattern>MY NAME IS *</pattern>  
  <template>Hi, <set name="name"><star/></set></template>  
</category>
```

Set a Variable

```
<category>  
  <pattern>WHAT IS MY NAME</pattern>  
  <template>Your name is <get name="name"/></template>  
</category>
```

Recall a Variable

Review - AIML

Remembering Context with `<that>`

```
<category>  
  <pattern>^ COFFEE ^</pattern>  
  <template>Do you like coffee?</template>  
</category>
```

```
<category>  
  <pattern>YES</pattern>  
  <that>Do you like coffee</that>  
  <template>Do you prefer dark or medium roast?</template>  
</category>
```

User: I should drink less coffee.
Bot: Do you like coffee?
User: Yes.
Bot: Do you prefer dark or medium roast?

User: I could go for a coffee or espresso.
Bot: Do you like coffee?
User: Yes.
Bot: Do you prefer dark or medium roast?

Review - AIML

Conditional Statement (version 1)

```
<category>
  <pattern>TODAY IS *</pattern>
  <template>
    <think><set name="today"> <star/></set></think>
    <condition name="today">
      <li value="Monday">The start of a new week.</li>
      <li value="Tuesday">Tuesday already?</li>
      <li value="Wednesday">Halfway to the weekend!</li>
      .
      .
      .
    </condition>
  </template>
</category>
```

Using the **<condition>** tag, a bot can respond differently to the same input depending on the value of a variable.

The concept is the same as the **IF / ELSE** statements we were using with Javascript

If the User writes "Today is Tuesday" The bot responds with "Tuesday already."
If the User writes "Today is Monday" The bot responds with "The start of a new week"

Review - AIML

Conditional Statement (version 2)

```
<category>
  <pattern>WHAT IS MY NAME</pattern>
  <template>
    <condition name="name">
      <li value="*">Your name is <get name="name"></li>
      <li>You haven't told me your name yet!</li>
    </condition>
  </template>
</category>
```

Another way to use the conditional statement is to check the status of a variable to see whether it has been set or not.

If it has been set, then there is one response.

If it has not been set, then there is a different response.

Review - AIML

Learning <learn> and <learnf>

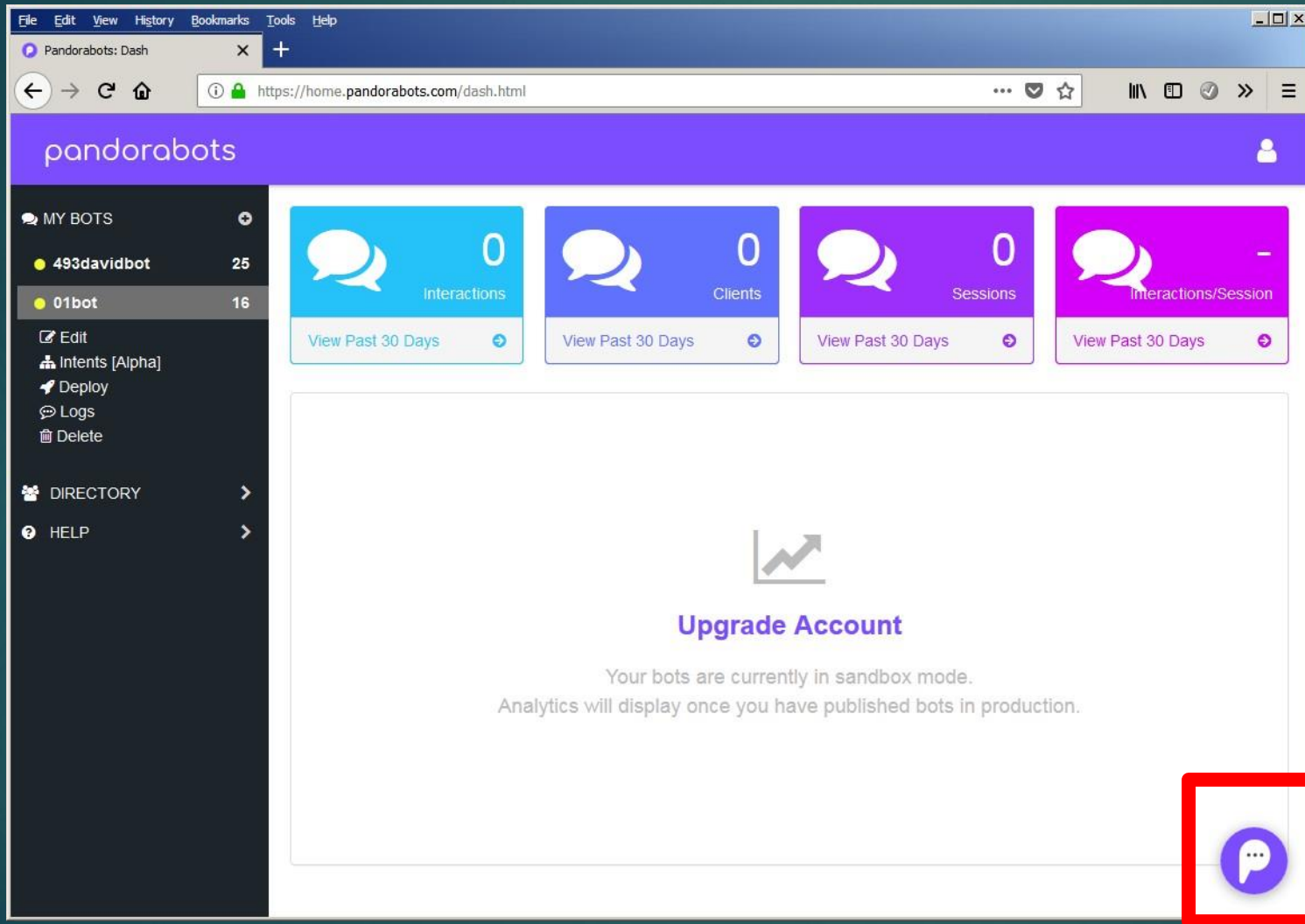
```
<category>
  <pattern>THE SKY IS BLUE</pattern>
  <template>Thanks, I will remember that information
  <learn>
    <category>
      <pattern>WHAT COLOR IS THE SKY</pattern>
      <template>The sky is blue</template>
    </category>
  </learn>
</template>
</category>
```

With <learn> users can generate new categories from a conversation.

Categories learned using <learn> will only be accessible during that conversation, and will eventually be cleared.

Categories learned using <learnf> will be written to a new AIML file and can be accessed by anyone talking with your bot.

Review - AIML



Bot Training via Chat Widget

Clicking on this icon will allow you to chat with your bot and make modifications to the bot's behavior without needing to write the code in AIML

Review - AIML

The screenshot displays the Pandorabots dashboard interface. At the top, a purple header contains the 'pandorabots' logo and a user profile icon. Below this is a dark sidebar with navigation options: 'MY BOTS' (with a plus icon), '493davidbot' (25), '01bot' (16), 'Edit', 'Intents [Alpha]', 'Deploy', 'Logs', 'Delete', 'DIRECTORY', and 'HELP'. The main content area features three analytics cards: 'Interactions' (0), 'Clients' (0), and a partially visible third card. Each card includes a 'View Past 30 Days' button. A modal window for '01bot' is open, showing a large empty space with a 'What are you' prompt at the bottom. A central message reads: 'Upgrade Account' with a line graph icon, stating 'Your bots are currently in sandbox mode. Analytics will display once you have published your bot.' The browser's address bar shows 'https://home.pandorabots.com/dash.html'.

Review - AIML

The screenshot displays the Pandorabots dashboard in a web browser. The browser's address bar shows the URL `https://home.pandorabots.com/dash.html`. The dashboard features a purple header with the 'pandorabots' logo and a user profile icon. On the left, a dark sidebar lists navigation options: 'MY BOTS' (with a plus icon), '493davidbot' (25), '01bot' (16), 'Edit', 'Intents [Alpha]', 'Deploy', 'Logs', 'Delete', 'DIRECTORY', and 'HELP'. The main content area contains three cards for 'Interactions', 'Clients', and another bot-related metric, each showing a count of 0 and a 'View Past 30 Days' button. A large 'Upgrade Account' banner is visible, stating 'Your bots are currently in sandbox mode. Analytics will display once you have published your bot.' A chat window for '01bot' is open, showing a user message 'What are you', a bot response 'I have no answer for that.', and a 'Show Metadata' link. The chat input field at the bottom contains the text 'Say something ...'.

Review - AIML



The screenshot shows a web browser window with the URL `https://home.pandorabots.com/dash.html`. The page title is "Pandorabots: Dash". The interface features a purple header with the "pandorabots" logo and a user profile icon. A left sidebar contains navigation options: "MY BOTS" (with a sub-menu for "493davidbot" and "01bot"), "Edit", "Intents [Alpha]", "Deploy", "Logs", and "Delete"; "DIRECTORY"; and "HELP". The main content area displays three cards for "Interactions", "Clients", and "01bot". An "Alter Response" dialog box is open, showing a conversation snippet: "You: What are you" and "Bot: I have no answer for that." Below this, there is a "New Response:" text input field, an "Add to File:" dropdown menu currently set to "pand_learn.aiml", and a purple "Add Category" button at the bottom right of the dialog.

Review - AIML

The screenshot shows a web browser window displaying the Pandorabots dashboard. The browser's address bar shows the URL `https://home.pandorabots.com/dash.html`. The dashboard has a purple header with the 'pandorabots' logo and a user profile icon. On the left, there is a sidebar menu with sections: 'MY BOTS' (containing '493davidbot' with 25 interactions and '01bot' with 17 interactions), 'Edit', 'Intents [Alpha]', 'Deploy', 'Logs', and 'Delete'; 'DIRECTORY'; and 'HELP'. The main area features several cards for 'Interactions' and 'Clients', both showing '0'. A chat window for '01bot' is open, showing a user message 'What are you' and a bot response 'I have no answer for that.'. An 'Alter Response' dialog box is overlaid on the chat window. The dialog contains the following fields: 'You:' with the text 'What are you'; 'Bot:' with the text 'I have no answer for that.'; 'New Response:' with a text input field containing 'I am a chat bot'; and 'Add to File:' with a dropdown menu showing 'pand_learn.aiml'. A purple 'Add Category' button is located at the bottom right of the dialog box.

Review - AIML

The screenshot displays the Pandorabots dashboard in a web browser. The browser's address bar shows the URL `https://home.pandorabots.com/dash.html`. The dashboard features a purple header with the 'pandorabots' logo and a user profile icon. A dark sidebar on the left contains navigation options: 'MY BOTS' (with a plus icon), '493davidbot' (25), '01bot' (17), 'Edit', 'Intents [Alpha]', 'Deploy', 'Logs', 'Delete', 'DIRECTORY', and 'HELP'. The main content area includes three cards for 'Interactions', 'Clients', and another bot-related metric, each with a 'View Past 30 Days' button. A large 'Upgrade Account' banner is visible, stating 'Your bots are currently in sandbox mode. Analytics will display once you have published your bot.' A chat window for '01bot' is open, showing a conversation with a bot that responds 'I have no answer for that.' and 'I am a chat bot'. The chat interface includes a 'Say something ...' input field at the bottom.

File ▾ Insert ▾ Publish Changes

AIML

- udc
- pand_learn
- Maps
- Sets
- Substitutions
- System

pand_learn ✕

udc ✕

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <aiml version="2.0">
3 <!-- insert your AIML categories here -->
4
5 <category>
6   <pattern>HI</pattern>
7   <template>Hello, how are you?</template>
8 </category>
9
10 <category>
11   <pattern>WHAT ARE YOU</pattern>
12   <template>I am a chat bot</template>
13 </category>
14
15 </aiml>
```

😊●○🔗🖼️🎬📄📄+↶⋮📄💬

These alterations to bot behavior are saved in another file under **AIML**. It is called **pand_learn**

When you chat with your bot, its operations will come from both the **udc** and **pand_learn** files

The screenshot shows the Pandorabots web interface. On the left is a sidebar with 'MY BOTS' (listing '493davidbot' and '01bot'), 'Edit', 'Intents [Alpha]', 'Deploy', 'Logs', 'Delete', 'DIRECTORY', and 'HELP'. The main area contains a code editor with two tabs: 'pand_learn' and 'udc'. The 'pand_learn' tab is active and contains the following XML code:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <aiml version="2.0">
3 <!-- insert your AIML categories here -->
4
5 <category>
6   <pattern>HI</pattern>
7   <template>Hello, how are you?</template>
8 </category>
9
10 <category>
11   <pattern>WHAT ARE YOU</pattern>
12   <template>I am a chat bot</template>
13 </category>
14
15 </aiml>
```

A 'File' menu is open, with options: Save (ctrl-s), New, Upload, Download Zip, and Delete. The 'File' menu title and the 'Save (ctrl-s)' option are highlighted with a red box. A 'Publish Changes' button is visible in the top right of the editor area.

Save all changes

And **save** often so you do not lose changes

Deploy your bot to the **Directory**

Test your bot to make sure it works

Competition

Quasi Loebner Prize

Procedure/Rules

1. Edit your bot; deploy it to the Directory
2. Interact with everyone's bot
3. Complete Evaluation Form

Timeframe

6:00-7:15 – Bot editing and deployment

7:15-8:15 – Chat with the bots

8:15-8:30 – Calculate results & Awards

COMS 493 – Quasi Loebner Prize

10 October 2018

Evaluator _____

Bot Builder	Bot Name	Score (5 = high score)
Nicholas Campo		1 2 3 4 5
Alexandra Cednick		1 2 3 4 5
Jasmine Chang		1 2 3 4 5
Michael Decker		1 2 3 4 5
Konstanze Fowler		1 2 3 4 5
Chris Grabowski		1 2 3 4 5
Keelen Holcomb		1 2 3 4 5
Breyana Jackson		1 2 3 4 5
McCoy Kent		1 2 3 4 5
Gabriel Leblanc		1 2 3 4 5
Ronald Luckett		1 2 3 4 5
Michael Madrid		1 2 3 4 5
Elizabeth McMurtrey		1 2 3 4 5
Travis Noonan		1 2 3 4 5
Jacalynn O'Donnell		1 2 3 4 5
Khalilah Rasheed		1 2 3 4 5
Jillian Sheehan		1 2 3 4 5
Nixon Urrutia		1 2 3 4 5
Nathan Veloz		1 2 3 4 5