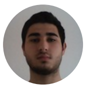
Practical Applications of Artificial Intelligence and the Implications of a Full Artificial Intelligence

Thesis · February 2015



Nikita V. Kholodnyi
TU Wien

Practical Applications of Artificial Intelligence and the Implications of a Full Artificial Intelligence

The Making of a Chatterbot

Nikita Kholodnyi 8M

BG / BRG Keimgasse 03

Betreut von Carina Lenotti

13.02.2015

Abstract

. As part of this paper, I have programmed my own artificial intelligence (AI) using the artificial intelligence markup language. At the beginning of this paper, I will go into the theories and practices surrounding artificial intelligence. To start the theoretical section, I have given an argumentation for what one would define as intelligence and list the types of artificial intelligence that are being researched. Then, I talk about whether or not a strong AI is possible to create before going into practical applications of artificial intelligence and the future of AI research. Afterwards I go into more practical aspects and begin describing how my self-made AI functions as well as the inner workings of the artificial intelligence markup language. I close the paper with theorizing what makes an AI an AI, my reflections and closing statements. A CD is included in with the paper that has all of the required files to install and run the AI on any given system as well as a pdf including the portions of code I made myself.

Table of Contents

1 Introduction

As a kid I’ve always been fascinated by robots. I loved the idea of robots and the possibilities that they opened for day-to-day life. I chose to write my paper on artificial intelligence (AI) because the idea of creating life or life-like things is one that excites me and many others. Artificial intelligence is a topic that many people quickly disregard as science fiction and it is in my interests to show just how real AI can be as well as show how a chatterbot can be made.

2 What is Intelligence?

Before getting into the complex topic of artificial intelligence, one must first define what one would consider to be intelligence. Most human actions, even simple ones like walking to the kitchen, are generally viewed as intelligent, while actions that insects perform, even ones that appear to be complex are generally considered unintelligent. One example of such an action would be the food storage of the digger wasp, *Sphex ichneumoneus* (Copeland, What is AI?, 2000). Upon gathering food the wasp returns to its burrow, places the food outside, checks its burrow for intruders and then brings the food inside. On the surface, the action appears to be quite intelligent, but one study on the topic (Copeland, What is AI?, 2000) shows that if you were to move the food slightly while the wasp was inside, the insect would move the food to the entrance and repeat the process again. The wasp can be made to repeat the process a large quantity of times. Intelligence therefore has nothing to do with the complexity of the actions one performs.

The Intelligence we are handling is defined in modern psychology by five pillars: learning, reasoning, problem-solving, perception and the understanding of language. (Copeland, What is AI?, 2000)

Artificial intelligence research can be split into roughly speaking the 3 following general categories. (Copeland, What is AI? part 2, 2000)

3 Strong AI

“[…]build a machine on the model of man, a robot that is to have its childhood, to learn language as a child does, to gain its knowledge of the world by sensing the world through its own organs, and ultimately to contemplate the whole domain of human thought.” (Weizenbaum, 1976)

The first and to this point most unsuccessful area of AI research is the research and development of strong AI or AGI (Artificial General Intelligence). Strong AI is, by definition, an intelligence that one could not tell apart from a human. So how would we determine whether or not an AI has obtained the level of intelligence needed to be on par with a human?

3.1 AI Evaluation

When evaluating AI there are 5 grades that it can be given for any task (Progress in artificial intelligence):

- Optimal (perfect performance)
- Strong Super-Human (better than all humans)
- Super-human (better than most humans)
- Par-human (similar to most humans)
- Sub-human (worse than most humans)

For example, a computers performance at checkers is optimal and its’ performance at chess is super-human, but its’ performance in face-recognition and hand writing recognition is sub-human.

3.2 The Turing Test

The Turing Test was developed by none other than the late Alan Turing. The test is the most simplistic in theory and yet most likely the most effective.

A person sits down in front of a computer and asks a computer questions. He receives answers from a human and an AI. He then has to decide which one was the AI and which one was the human. Although the test is simplistic in theory and design, there have been few to pass it and none to pass it effectively (A chatterbot, (A form of AI made to hold conversations with humans), that made spelling errors once passed this test, because the interrogator thought that spelling errors wouldn't be made by robots). Humans have their own way of reacting and interpreting messages that robots cannot mimic with logical connections. This test does a good job of showing that people are unique in their way of thinking and that intelligence goes beyond calculating and responding to statements. Obviously, the test has its' faults and passing it once doesn't in any way mean that the AI is intelligent, but it is still a standard test for chatterbots everywhere and the one which is most commonly employed when evaluating chatterbot programs (Turing Test).

3.3 The Completely Automated Public Turing test to tell Computers Apart from Humans

The Completely Automated Public Turing Test to tell Computers and Humans Apart (CAPTCHA) is a derivative of the Turing test where a computer asks a human if they are indeed human. The computer asks a simple question or proposes a simple task, most famously writing down distorted letters and number and expects a human to perform the test perfectly. This test is most commonly seen in log-in screens as a deterrent to automated account-creating bots and is known for angering people who do not always manage to pass them (CAPTCHA, 2000-2010).

4 Applied AI

The aim of applied AI research is to produce smart systems that are commercially viable. Most AIs that one would see day to day are a result of applied AI research. Goals of this research stretch from security systems with face recognition to making robotic servants and maids. This area of AI research has seen various successes and it is the driving force behind the reasons why AI is being researched. To read into areas and uses of AI, read onto the section *Practical Applications of AI* (Copeland, What is AI? part 2, 2000).

5 Cognitive Simulation

This research field focuses on being able to simulate human emotion and thought processes in an attempt to test how the human mind works.

The theory is that our mind processes information in a way not to far off from how an AI would do it. The AIs made in this field are put up to do tasks like solve abstract problems or perform face recognition. Afterwards, the results of the AI are compared to how a human would process the information.

Simulations of neural networks have seen particular success in psychological and neurophysiological fields (Copeland, What is AI? part 2, 2000).

6 Types of Artificial Intelligence

While AI research is headed into many different directions, there are only a few actual approaches to programming an AI, such as Bottom-Up and Top-Down approaches.

Although the end product may sometimes look the same, the variations of AI programming differ completely in functionality.

6.1 The Bottom-Up AI

The theory for the bottom-up AI comes from the way that our brains function (Copeland, Top-Down vs. Bottom-Up AI, 2000). Our brains are a connection of neurons that interact with each other. Researchers claim that given a large enough pool of information, an AI could answer and act like a human. This approach attempts to gather a large pool of information and then create the connections between them that a human would create. Logically, given enough time and effort, this theory makes quite a lot of sense.

For further explanation, say we have 4 pieces of information: fruit, pineapple, monkey and seed. The way the bottom-up approach to AI works would be to start making connections between these 4 pieces of information. Monkeys like fruit; seeds grow trees that bear fruit; pineapple is a fruit. This approach to AI builds up as many connections as possible between the information it is given. The programmer must manually find every possible connection between objects that a human can and make every possible answer to every possible question and statement. The reality in this is horrifying. It is nearly impossible to accomplish this, because there are a nearly infinite number of sentences that may be formed within a language. Chatterbots must rely on default answers and ways around statements they cannot figure out.

While bottom-up AI can mimic intelligence, it doesn't in anyway actually have intelligence. It uses algorithms to find possible answers to statements and if one were to say the same thing over and over, one would quickly realize that the chatterbot only has a certain amount of answers that may never vary.

6.2 The Top-Down AI

A Top-Down Ai or ‘symbolic AI’ is an AI that uses descriptions of things to form logical connections. Instead of creating a network of information, programmers try to construct ‘rules of thought’ through representation and symbolism to create new strings. In a way, a Top-Down bot can therefore think (Copeland, Top-Down vs. Bottom-Up AI, 2000).

While this approach has seen many successes in the past, it is slowly being overshadowed by a large field of applications and the fluidity of the Bottom-Up approaches (Copeland, Top-Down vs. Bottom-Up AI, 2000).

7 Is a Strong AI possible?

Applied artificial intelligence research and cognitive simulation research has been held up by a steady flow of successes over the years. However, strong AI research has hit a wall that it seemingly cannot surpass and many researchers are beginning to doubt the possibility of a strong AI. 60 years of research in strong AI have failed to produce even the most basic of intelligent beings, let alone a human.

One of the goals of bottom-up researchers has been to reproduce more simplistic beings on the basis of neural network. The *Caenorhabditis elegans* is a species of worm which contains a mere 300 neurons of which we already know the connections (Copeland, Is Strong AI possible?, 2000), yet the models of bottom-up theorists have been nothing but representations of the real thing.

So the question stands, is a strong AI impossible to produce or simply too difficult?

In order to answer this question one must first decide whether or not a computer can actually think. A plane flies, but a boat doesn't swim; Noam Chomsky, linguist, philosopher, cognitive scientist and logician, says that whether or not we define a computer as a thinking object is simply a question of decision, where neither answer is right or wrong (Copeland, Is Strong AI possible?, 2000). So the answer to our question lies in the definition of Intelligence and creating a bot that can convince a person of its own humanity.

From a mere technical standpoint, there is no reason for a strong AI to be impossible. Running a computer with the required amount of 'neurons' would require a mere 1THz of computing power and while that may be no small amount it is quite obtainable and the silicon transistors in the CPU would make for a much faster brain (Strong AI).

Whether or not a strong AI can be made by humans is yet to be seen. We still do not know our brain to the fullest extent and therefore cannot find what makes it work like it does. Before figuring out how to bestow intelligence upon a machine, we must first figure out what it is that makes us intelligent. We must find the part of our brains that makes us conscious. If the brain proves to be nothing more than a well-oiled machine than there is no reason for other machines to not be constructed like our brain.

8 Practical Applications of AI

Although most people live their lives without noticing it, Artificial Intelligence is very much a part of our daily lives, allowing for people to work and play more efficiently.

8.1 Games

The most obvious application of AI would have to be its use in the video game industry. Nowadays, many characters used in games have stopped going by a simple 'say A, answer B' programming and they have been made to react to situations and occurrences that aren't necessarily seen beforehand.

The first game to use a noticeably complex AI was Half-life, made in 1998 (Lane, 2013). The enemies in the game could smartly place grenades and seemingly anticipate your moves. They wouldn't always simply attack; sometimes they would try to get around you to get an advantage.

Since then, AIs have been an important part of the game industry and have been used in almost every game since then, getting ever more complicated. The more recent series of games known as *Far Cry* has a critically acclaimed AI that responds to sound, light and actions in an almost human-like fashion, allowing them to even plan tactical battles (Cowen, 2014).

8.2 Industrial Work

Factories and laboratories have long since replaced human labor with the more efficient AI robotic variant. The AI that operates machines in factories requires pattern and object recognition to be able to take objects and put them together or take them apart accordingly (Applications of AI).

8.3 Finance

Artificial Intelligence has been and is currently being used to buy and sell stocks at incredible speeds for high profit.

The first developed program that notably beat professional traders was the Arizona Financial Text System, an AI that reads through hundreds of finance related articles and analyzes minute by minute changes in the stock market and accurately selects and buys stocks that will be worth at least 1% more within the next 20 minutes and then sells them within the same time (Mims, 2010).

9 Future of AI Research

AI research peaked around 2003 before hitting a dead end where no one could figure out how to bring an AI to the next level of thought process, but many researchers are hopeful for the future. Many believe that AI research can change the world greatly within the next century, whether it be through a slow progression or an all at once break-through.

9.1 Short-Term Priorities (Future of Life, 2015)

The success of AI within industrial areas has demonstrated its' growing impact on the economy. Although many people debate on the nature of this impact and whether or not it is due to the effects of AI or other information technologies, most scientists agree that there is valuable research to be done on maximizing the effectiveness of AI, while minimizing its negative impact, such as unemployment.

In the short term scientists have agreed on three main priorities for research: the economic effects of AI, the legal and ethical consequences, and the ability to guarantee that an AI is "robust", and will do what it is supposed to.

9.1.1 Economic Effects

The first point deals with what was mentioned in the first paragraph. The economic effects of AI, while positive as a whole, may affect individuals in very negative ways.

Before producing AI, we must first figure out how to properly fit it into society. What jobs should be automated and what kind of impact will it have on labour workers and information specialists? What changes will AI bring to consumer and financial markets and how must we change them to fit AI? How must we change wages and work ethic to support a potentially flourishing AI economy? What changes must we make our understanding and measurement of economic success, as GDP does not necessarily prove to be a good measurement of value in an AI society?

9.1.2 Law and Ethics

What is an AI liable for doing or not doing? The current set of laws that governments have not foreseen the possibility of AI and need to be possibly changed in accordance to the development of AI. If we were to mass produce self-driving vehicles that cut the amount of car accidents in half, are the makers of the cars now responsible for all car accidents which happen with their vehicles? How does an AI weigh life or injury against material value? What kind of privacy laws must we put in place to stop AIs from misusing our information? How far are we allowed to research AI and should a certain level of AI be illegal to create?

9.1.3 Research of Robust AI

Possibly the most important aspect to research before diving deeper into the actual capabilities of AI, this deals with the verification of AI systems, how secure they are and the control we have over them.

Verification deals with the testing of AI and making sure that it works as is intended. The difference between verifying AI and verifying regular software lie in the fact that AI by definition is required to interact with the environment, which is often unknown to the maker of the AI upon its' creation. A lack of knowledge of the environment implies the need for learning algorithms, but then those would have to undergo verification process. There is currently no method for being able to tell for certain that an AI works within the parameters it is expected to function.

AIs also present a new level of control achievable through hacking. While hackers nowadays are certainly dangerous, the level of power achieved through hacking an AI is at another level. While this applies to areas outside of AI research as well, security needs to be thoroughly checked and bettered before mass production of AI.

The last area, control, deals with how much control humans have over AI and how we can ensure that they have control, especially in dangerous areas such as self-driving vehicles and weapon platforms.

9.2 Long-Term AI research

While these may be our goals for the near future, what can we expect from AI research farther down the road?

9.2.1 Recursive Self-Improvement and Singularity

One may have noticed the short-term goals of AI research follow a general theme of safety and regulations, which is in part due to the dangers presented by technological singularity. The technological singularity is a hypothesis stating that if an artificial intelligence were to reach a point at which its' intellectual capacity exceeds that of a human it would rapidly change our civilization catapulting it into a new era (Tomasik, 2014).

While the singularity has been a popular topic among science fictions films such as *Transcendence*, it is in all actuality a very real possibility that has had many acclaimed scientists, most notably Stephen Hawking, worrying about the future. In a statement to the BBC (Cellan-Jones, 2014) he said, 'The development of full artificial intelligence could spell the end of the human race' and that a strong AI, "would take off on its own, and re-design itself at an ever increasing rate," in a process called recursive self-improvement.

The problem with the singularity is that the outcome is currently unknown to us and we have no idea how to accurately predict what will happen, hence the current research topics in AI leaning towards controlling a potential full AI.

While no one believes this will happen now, many scientists claim a full AI will be created in this century and that we need to be prepared for its implications.

9.2.2 The Single Algorithm (Hernandez, 13)

There is a theory that human intelligence stems from a single algorithm. This does not mean that our brain runs on one algorithm rather that our learning process derives from a single one. This is a theory stemming from the discovery that our brain is quite plastic allowing it to change over time and give different parts of the brain different functions depending on what is needed. For instance, a blind man uses the part of our brain which we would use to see to hear or smell better. The hypothesis that our brains' learning process is based on a single algorithm was popularized by Jeff Hawkins, author of *On Intelligence.*

This theory has given many modern AI researchers hope that maybe the key for full AI may be more obtainable than once imagined. Using neural networks, AI researcher Andrew NG managed to create a single program that can both see and hear near perfectly, strengthening the basis behind this theory.

10 The AI: Materials and Methods

In accordance with this paper, I was given the task of creating my own artificial intelligence. Due to the limited time, I decided against creating one from scratch and instead I based my AI on the award-winning AI, A.L.I.C.E.

A.L.I.C.E. (Artificial Linguistic Internet Computer Entity) is an open-source AI developed in 1995 using AIML. I chose to base my own intelligence off of A.L.I.C.E. due its' acclaimed success in the field and winning the Loebner Prize 3 times.

Having the AI based on A.L.I.C.E means to take the core structure of the program and then tweak areas that I feel I want to be different, essentially changing its' personality. In order to use the AIML files provided by A.L.I.C.E., I needed an AIML interpreter, a program that matches input to statements within the AIML code. For my project I chose Program-O.

10.1 AIML

10.1.1 Basic Construct (Bush, 2001 (rev 2006))

The Artificial Intelligence Markup Language (AIML) describes AIML objects and it is based on the Extensible Markup Language (XML). AIML objects contain "Topics" and "Categories". These can be read and output by an AIML interpreter, such as Program-O.

At first glance, the AIML document may remind you of HTML, but do not be fooled. While the quotations and symbols are similar, the document is made and structured differently. An AIML document is contained within the following tags:

- *<aiml version='xxx'>*
- *</aiml>*

where 'xxx' is replaced with the version you are using. There are only two tags which are allowed to be directly under the AIML tags:

- *<category> </category>*

- *<topic name =‘xxx’> </topic>*

These are called Top-Level commands, due to the fact that they are the first to be accessed. A Topic may contain multiple categories, but it is, for the most part, optional. Note that when a category is placed outside of a Topic, the AIML interpreter must assume it is within a Topic named ‘*’, which is the wildcard.

Within the Top – Level command (or in the event that the Category is within a Topic, a second-Level command) Category there are, once again, two underlying tags;

- *<pattern> </pattern>*
- *<template> </template>*

Each category must contain no more and no less than one of each of these tags and the Pattern must always come first within a Category. The Pattern is a kind of trigger to tell the AIML interpreter what to answer to. Looking at an example from A.L.I.C.Es AIML files:

```
<category>

        <pattern>

                CAPRICORN

        </pattern>

        <template>

                I don't know many  people born under <set
                name="sign">Capricorn</set>.

        </template>

</category>
```

The Pattern contains only the word CAPRICORN. It means that if a user were to mention CAPRICORN one of the possible Match Paths (A logical connection between patterns and input made by the AIML interpreter) would lead here. The response is then written in the Template. In this case the script sets Sign equal to Capricorn meaning that if a user were to mention signs again in the same session, the AIML interpreter would remember that the sign they talked about last was Capricorn.

So how are Match Paths created and how does the Interpreter know which path to take?

A Match Path has 3 major elements: a Pattern, a That-Element and a Topic. The Topic is not written within the document meaning that the Interpreter must either find an implied topic or the topic is the name of the document. The That-Element, in this case a Pattern-Side That-Element, is a special element used for context matching. It is optional, but in the case that it is missing the Interpreter must replace it with a Wildcard. The Pattern-Side That-Element is simply a means to further specify in what scenario the following should occur.

On to the actual method of constructing a Match Path, we see that there happen to be multiple ways to accomplish this. So let's create a simple Topic/Category/Pattern construction.

Let's say our Topic name is Food

<topic name ='FOOD'>(Important: the name of the Document is food. This is not always written within the document (in the case that there are multiple topics in one document it is written into the document) also this is only a representation of how a topic would look like)

Within our Category we have a Pattern for chicken.

```
<pattern>CHICKEN</pattern>
```

In this case scenario we do not want chicken, rather we want chicken that the bot has previously said to be warm.

```
<that>WARM</that>
```

After this comes the Template. In this case we will simply have it contain 'xxx' as it is not relevant to this example.

Now the final code looks like this:

```
<topic name='FOOD'>
```

```
        <category>
            <pattern>
                CHICKEN
            </pattern>
            <that>WARM</that>
            <template>
                XXX
            </template>
        </category>
    </topic>
```

So the code now resembles something one can find within an AIML document.

The Match Path construction is as follows: Chicken –Warm - Food.

Simply put, it means that the Interpreter searches from the bottom. When a user mentions Chicken, the Interpreter searches all Patterns that contain chicken. Afterwards it checks for That-Elements. In this case, the bot must have previously said the word 'warm'. Only then does it check to make sure the user is talking about chicken as a food and not as an animal. This type of Match Path making is slow, but for the most part quite robust.

In the scenario that there is no <that>, the That-Element must be made into a Wildcard by the Interpreter. The Match Path in our example would look like this: Chicken - * - Food. This means that this is a viable option as long as the user is talking about chicken as a food (The Wildcard is allowed to be anything).

The same is true when you have no Topic. The Match Path in case of a missing Topic and That-Element would simply be Chicken - * - *, meaning that as long as the user mentions chicken, this statement can be said.

AIML Interpreter cannot simply build the Match Paths to match user input. The statements made by users must first be 'normalized' by the Interpreter.

There are 3 Types of normalization that can be performed. The only one of these normalizations that is mandatory is the Pattern-Fitting normalization. The other 2, Sentence-Splitting and Substitution normalizations are decided upon by the user or maker of the bot and are therefore optional.

If it is performed, the Substitution normalization is the first to be executed. During the Substitution process, the Interpreter filters things out that may mess up its understanding of a sentence. Abbreviations are a prime example of things that may need to be substituted by the Interpreter. If we were to have the sentence:

I met Mrs. Jones at a café last week.

The Interpreter would need to do something about the abbreviation Mrs., as if it were to leave it as-is, it might end up deciding that 'I met Mrs' was one sentence and 'Jones at a café last week' was another. During the Substitution normalization Mrs. would be exchanged for misses. This process also takes care of web-addresses and file extensions in a similar way. 'https://www.google.at/' becomes 'https://google dot at' and '.txt' would simply be made to ' txt'.

The next normalization to occur is the Sentence-Splitting normalization. During the process the Interpreter tries to break down the user input into separate sentences. While further rules can be applied by the maker of the bot, the only rule which works with any certainty is to break sentences at periods, making the Substitution normalization important if you were to perform this one.

Lastly, we have the Pattern-Fitting normalization. This process turns all lowercase letters into uppercase letters and all non-letter characters into spaces, as the only input accepted by the interpreter must be made up of uppercase letters.

Based on these three normalizations, if you were to input, "Hi. How was your day? ☺", the bot would be receiving, "HI HOW WAS YOUR DAY", where "HI" and "HOW WAS YOUR DAY" are on separate lines.

Now that we have gone over the process of normalization and input, we can go into how the bot decides which Match Paths to take.

This process is learned and explained best by example. Let us go ahead and assume the following chat log:

User: “I like fruit.”

Bot: “What is your favourite fruit?”

User: “I like bananas.”

Bot: “I enjoy bananas, too.”

In this chat the user initiated with a conversation by mentioning fruit. The match path used to find the bots answer would look along the lines of this:

LIKE FRUIT<that> * </that> <topic>FOOD

It finds the pattern LIKE FRUIT. Since the conversation just started, there is currently no <that> statement, meaning the interpreter fills it in with a Wildcard. Lastly, the bot finds itself within the Topic FOOD. Within the Template for LIKE FRUIT with any That-element the bot finds the answer “What is your favourite fruit?”

After the user answers the bots question, the bot must once again decide which of the Match Paths it is to take. The Match Path for the users’ statement would look something like this:

BANANAS <that>FAVOURITE FRUIT</that> <topic>FOOD

The That-Element is taken from the last thing mentioned by the bot. In this scenario the bot asked for the users’ favourite fruit, hence the That-Element. The user answered with bananas and the bot now searches for a suitable answer to bananas being a favourite fruit. The bot may also save the information that the users’ favourite fruit is banana and use it in the conversation later.

In reality the bot searches each input word-for-word and sentence-for-sentence and matches them with the patterns. After the match is found the Interpreter constructs the contents of the template into a formulated, grammatically correct, output.

10.1.2 Other important concepts and constructs

Now that we have figured out the basics of AIML, we can go into the more complex concepts and constructs within the language.

Reductionism (Wallace, 2000)

Reductionism is simply put the art of reducing the amount of code that needs to be written in order to make the code function properly and AIML happens to have a good way of doing this with the <srai> tag.

The <srai> tag is written into the template, which is the area which contains the response, but reacts to the tag as an input. Within the A.L.I.C.E. source code in the file salutations we see an example of the <srai> tag being used:

```
<category>
    <pattern>C YA</pattern>
    <template>
        <srai>bye</srai>
    </template>
</category>
```

The tag would transform the users input “c ya” and read it as goodbye, taking it to the bye pattern. In the salutations file we see many patterns that respond with <srai> bye </srai> and all of these take the bot to the same pattern which uses the <random> tag to randomly pick a response out of a list of responses, the list being marked with <li> tags. In this way the <srai> tag shortens the amount of text that needs to be written by rerouting all of these inputs to one pattern.

Predicates (Bush, 2001 (rev 2006))

A predicate is an Item which can be declared at any point in the AIML file and whose value can be manipulated, in this way it act much like a variable would in other languages. Predicates have to be set using the <set name='xxx'> tag, where xxx stands for the variable name.

Predicates can be quite flexible. 'Return-name-when-set', although not an actual function, is a way to set 2 strings to correspond to each other.

```
<category>

        <pattern>

                SHE SAID I HAD TO

        </pattern>

        <template>

                Who?

        </template>

</category>

<category>

        <pattern>

                *

        </pattern>

          <that>WHO</that>

         <template>
        Why do you think <set name = ‘she’><star/></set> made you do it?

         <get name = ‘she’> is a reasonable woman.

      </template>

</category>
```

In this scenario the computer sets the value ‘she’ to equal to the name of the person, which is received after the question ‘Who?’ The conversation would then be as follows:

User: She said I had to.

Bot: Who?

User: Janice

Bot: Why did she make you do it? Janice is a reasonable woman.

This allows for the bot to know what is referenced when the user says ‘she’.

10.2 Program-O

My AIML Interpreter of choice for this project is Program-O. Program-O is an open source server-side Interpreter that uses MySQL and PHP to store information and display it. Not much work needs to be done in order to make a functioning chat bot, thanks to Program-O’s easy setup and user-friendly controls. One must only create a MySQL database and a new user to install the program and run it in your browser. Afterwards, you need only to fill in the blanks within Program-O, either in the browser or in MySQL.

To add new personality-associated information into your bot you just add to the ‘botpersonality’ database, which was created upon installing Programm-O. Adding information goes as follows: (`id`, `bot_id`, `name`, `value`)

```
 (NULL, JASON, 'wear', t-shirts'),
```

In this case we just added that our bot, Jason, wears a t-shirt, so if he were to be asked what he wears he could answer t-shirt. The first value is NULL, because we only have 1 chatterbot and therefore do not need to ID them. Information like this is left out of the AIML code and filled in Interpreter side in order to make it easier for people to personalize a chatterbot, both for private and business oriented use.

10.2.0 Installation Process of Program –O (Installation)

Installing program-O is a rather simple process. After downloading program-O, the first step is to extract the data to your local host or webserver. Afterwards, you have to create a MySQL user as follows:

```
CREATE USER 'myUser'@'localhost' IDENTIFIED BY  '***';
```

Then create a program-O database:

```
CREATE DATABASE program_O;
```

And then grant the user you created all privileges regarding the database:

```
GRANT USAGE ON * . * TO  'myUser'@'localhost' IDENTIFIED BY  '***' WITH
MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0
MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0 ;

GRANT ALL PRIVILEGES ON  `program_o` . * TO  'myUser'@'localhost';
```

Lastly open your browser to the directory where you extracted program-O and go through their step-by-step installation process.

In order to get a bot to function, import AIML files into program-O through the program-O menu.

11 Results

Using this knowledge of the AIML language I decided to further the A.L.I.C.E. source code to better fit my purposes. One of the things that I wanted to do was to have the bot talk about some of the things that I like. One of these things is the game *Smite.* To start, I made a file for the categories, which I named, Smite. Then I made a series of categories that, when mentioned, would set the topic to *Smite*, such as:

```
<category>
        <pattern>
                HAVE YOU HEARD OF SMITE
        </pattern>
        <template>
                <srai>SMITE</srai>
        </template>
</category>
```

Notice the <srai> tag being used to reference a different category. I made a few categories that look similar, but have different phrasing to make sure to get every possible variant of asking whether or not the bot knows of the game. The <srai> tag then leads to the following category:

```
<category>
        <pattern>
                SMITE
        </pattern>
        <template>
                I love Smite! What is your favourite god?
```

```
            <think> <set name="it"> <set
            name="topic">SMITE</set></set></think>

        </template>

</category>
```

The bot answers to any form of bringing up *Smite* with the phrase ‘I love Smite! What is your favourite god?’ The <think> tag makes sure nothing inside of the tag gets printed onto the screen. I then set ‘it’ and the ‘topic’ to SMITE, which means the program knows that we are talking about *Smite* and it defaults the word ‘it’ to refer to *Smite*.

Once the user answers with a name of a ‘god’ present in the game, each of which has their own category that leads to the category FAVOURITE GOD, the bot goes into this category:

```
    <category>

        <pattern>

            FAVOURITE GOD

        </pattern>

        <template>

            I  <random>

                <li>like</li>

                <li>dislike</li>

                 <li>hate</li>

                 <li>love</li>

            </random> <set name = 'user god'> <input> </set>.My favourite
            is

             <set name = "god">

             <random>
                <li>Agni<li>
                 <li>Kukulkan</li>
                <li> He Bo </li>
             </random></set>
        </template>
    </category>
```

The bot in this situation has a multitude of possible answers, which, I decided, it chose at random with the use of the <random> tag. The tag functions by randomly selecting one of the options inside the tags, which are enclosed by <li> tags. I also use the <input> tag to print the users last input onto the screen and set that input to ‘user god’ in order to make note of that being the users’ favourite god. I then set the random answer chosen by the bot to ‘god’ to make sure the bot uses the same god when referring to its favourite. The output at this part in the script can take on a few forms. The bot can print out ‘I dislike ’ + user input+ ‘. My favourite is Agni’, as well as ‘I love ’+user input + ‘. My favourite is He Bo’. I made use of <random> to give the bot a feeling of diversity.

The conversation regarding the topic *Smite* goes on further in a similar fashion.

12 Discussion

When I set out to make a chatterbot, I did not know what to expect. There is a surprisingly small amount of information regarding the construction of such AIs. The first way I found to construct an AI was with the help of chatscript.

Chatscript is a rather poorly known scripting language designed with chatterbots in mind. The problem I faced with chatscript was with the documentation. While it does not lack documentation, I found it to be somewhat confusing and lacking crucial information, most notably what to actually write into the program.

After hours of confusion and self-loathing which derived from my confusion, I decided to move on to a different method of making a chatterbot, which is when I came across AIML and A.L.I.C.E.

I chose to do the project with AIML because while there was documentation, the examples that were in A.L.I.C.E.s’ source code proved to play an invaluable part in my learning the language. A.L.I.C.E. is a bottom-up AI, because it has a large pool of information and connections between them.

My choice of interpreter was backed by somewhat less information, when I searched AIML interpreters, program-O was one of the first to pop up and I found it easy enough to handle from the get-go, meaning I did not look into other interpreters. Maybe there are some which are better out there.

A.L.I.C.E. bot being an extensive research project, it has a very wide array of conversation topics and answers. Due to the fact that it hasn’t been worked on for roughly a decade, one of the AIs’ weak points is anything to do with modern media.

One thing, that I would have done differently, given the time, would be to script the bot from the ground up instead of using the source code, even if that would make the results sub-par.

Generally, I am happy with the results that I produced and can see myself learning more about AI in the future.

12.1 What Makes an AI an AI

An AI, definition is a machine that can imitate human intelligence and/or behavior (Rouse).

So is by this definition a chatterbot an AI? A chatterbot is a machine which uses auditory or textual means to communicate with a person. These programs engage in small talk as an attempt to convince people that they are, in fact, intelligent, at least to a certain degree.

By the definition of an artificial intelligence, a chatterbot is an intelligence, because it attempts to represent human behavior through communication. But there are many things which are considered to be AIs, but do not fit the description of mimicking human intelligence.

Automated call operators are considered to be a form of AI, even though all they do is take in a numeric input in order to progress you down a string of questions.

The programs that make schedules and rotations for people in medical care are also a form of AI, whose only job is to tell people when and where to be.

While these systems do little to represent human intelligence they still do it, even if all they do are the mundane tasks that no person wants to do.

13 Closing Statements

Artificial intelligence is a large and challenging research field. The progress we have made in the last few decades is both immense and minimal. I feel like there is much to be learned in this field and I am sure that real progress is imminent. In this paper I covered many relevant subjects in AI research and while this was just a scratch on the surface I think this is a good start going into real AI research. I covered the fields in AI research and questioned the possibility of a strong AI. I talked about the relevant approaches to programming an AI. I mentioned the practical applications of AI in todays' society as well as its' relevance. Then, I took the dangers of AI into account, while talking about future research goals both in the long and short term.

In the practical part of this paper I described the methodology and language I used to program my AI. I went into detail about how AIML works and how I used it in my program. I covered the program I use to run the AI, program-o, as wells as how to get it to work. I then argue why a chatterbot is an AI and give my thoughts about the project as a whole.

AI is a broad topic and there are many things that I still have to learn about it. I feel like for every tidbit of information that I learn about AI, I ask twice as many new questions. I hope that this paper has helped you to understand the capabilities and possibilities that artificial intelligence presents for humanity as well as the possibly negative things it can bring with it.

And with that, I finish this paper, gaining a new view on modern science and technology.

Bibliography

CAPTCHA. (2000-2010). Retrieved 02 11, 2015, from Captcha: http://www.captcha.net/

Applications of AI. (n.d.). Retrieved 2 1, 2015, from Wikipedia: http://en.wikipedia.org/wiki/Applications_of_artificial_intelligence

Bush, N. (2001 (rev 2006), October 25). *AIML 1.0.1.* Retrieved November 4, 2014, from A.L.I.C.E. Bot: http://www.alicebot.org/TR/2001/WD-aiml/#section-aiml-objects

Cellan-Jones, R. (2014, December 2). *Stephen Hawking warns artificial intelligence could end mankind*. Retrieved Febuary 7, 2015, from BBC: http://www.bbc.com/news/technology-30290540

Copeland, J. (2000 , May). *What is AI? part 2*. Retrieved 12 27, 2014, from alanturning.net: http://www.alanturing.net/turing_archive/pages/reference%20articles/what_is_AI/What%20is%20AI02.html

Copeland, J. (2000, May). *Is Strong AI possible?* Retrieved 12 27, 2014, from alanturing.net: http://www.alanturing.net/turing_archive/pages/reference%20articles/what_is_AI/What%20is%20AI13.html

Copeland, J. (2000, May). *Top-Down vs. Bottom-Up AI*. Retrieved 12 27, 2014, from alan turning: http://www.alanturing.net/turing_archive/pages/reference%20articles/what_is_AI/What%20is%20AI09.html

Copeland, J. (2000, May). *What is AI?* Retrieved 12 27, 2014, from alanturing.net: http://www.alanturing.net/turing_archive/pages/reference%20articles/What%20is%20AI.html

Cowen, N. (2014, November 4). *Far Cry 4s' AI is So Clever Even Its' Devolopers Can't Outwit It*. Retrieved 2 2, 2015, from Gizmodo: http://www.gizmodo.co.uk/2014/11/far-cry-4s-ai-is-so-clever-even-its-developers-cant-outwit-it/

Future of Life. (2015, January 23). *Research Priorities*. Retrieved 2 3, 2015, from Future of Life: http://futureoflife.org/static/data/documents/research_priorities.pdf

Hernandez, D. (13 , August 5). *The Man Behind the Google Brain: Andrew Ng and the Quest for the New AI*. Retrieved 2 4, 2015, from Wired: http://www.wired.com/2013/05/neuro-artificial-intelligence/all/

Installation. (n.d.). Retrieved September 20, 2014, from Program-O: http://blog.program-o.com/documentation-v2/installation/

Lane, R. (2013, April 8). *Whatever Happened to Video Game AI?* Retrieved 2 2, 2015, from IGN: http://www.ign.com/articles/2013/04/08/whatever-happened-to-video-game-ai

Mims, C. (2010, June 10). *AI that picks stocks better than the pros*. Retrieved 1 6, 2015, from MIT technology review: http://www.technologyreview.com/view/419341/ai-that-picks-stocks-better-than-the-pros/

Progress in artificial intelligence. (n.d.). Retrieved 12 30, 2014, from Wikipedia: http://en.wikipedia.org/wiki/Progress_in_artificial_intelligence

Rouse, M. (n.d.). *What is AI.* Retrieved 2 4, 2015, from Tech Target: http://searchcio.techtarget.com/definition/AI

Strong AI. (n.d.). Retrieved 1 1, 2015, from AI wikia: http://ai.wikia.com/wiki/Strong_AI

Tomasik, B. (2014, May 14). *Thoughts on Robots, AI, and Intelligence Explosion*. Retrieved 2 5, 2015, from Foundational Research: http://foundational-research.org/robots-ai-intelligence-explosion/

Turing Test. (n.d.). Retrieved 12 30, 2014, from Wikipedia: http://en.wikipedia.org/wiki/Turing_test

Wallace, D. R. (2000, March). *Symbolic Reductions in AIML .* Retrieved November 10, 2014, from A.L.I.C.E. Bot: http://www.alicebot.org/documentation/srai.html

Weizenbaum, J. (1976). Computer Power and Human reasoning: From Judgement to Calculation. In R. Chrisley, & S. Begeer, *Artificial Intelligence: Critical Concepts Volume 3* (p. 313). London: Routledge.

Attachments

Smite.aiml:

```
<?xml version="1.0" encoding="UTF-8"?>
<aiml version="1.0">
<category>
        <pattern>
                HAVE YOU HEARD OF SMITE
        </pattern>
        <template>
                <srai>SMITE</srai>
        </template>
</category>
<category>
        <pattern>
                DO YOU KNOW SMITE
        </pattern>
        <template>
                <srai>SMITE</srai>
        </template>
</category>
<category>
        <pattern>
                DO YOU ENJOY SMITE
        </pattern>
        <template>
                <srai>SMITE</srai>
        </template>
</category>
<category>
```

```
		<pattern>

			SMITE

		</pattern>

		<template>

			I love Smite! What is your favourite god?<think> <set name="it">
				<set name="topic">SMITE</set>						</set>
			</think>

		</template>

	</category>

	<category>

		<pattern>

			AGNI

		</pattern>

		<template>

			<srai>FAVOURITE GOD</srai>

		</template>

	</category>

	<category>

		<pattern>

			AH MUZEN CAB

		</pattern>

		<template>

			<srai>FAVOURITE GOD</srai>

		</template>

	</category>

	<category>

		<pattern>

			ANHUR

		</pattern>
		<template>
			<srai>FAVOURITE GOD</srai>
		</template>
	</category>
	<category>
		<pattern>
			ANUBIS
		</pattern>
		<template>
			<srai>FAVOURITE GOD</srai>
		</template>
	</category>
	<category>
		<pattern>
			AO KUANG
		</pattern>
		<template>
			<srai>FAVOURITE GOD</srai>
		</template>
	</category>
	<category>
		<pattern>
			APHRODITE
		</pattern>
		<template>
			<srai>FAVOURITE GOD</srai>
		</template>
	</category>
```

```
<category>
    <pattern>
        APOLLO
    </pattern>
    <template>
        <srai>FAVOURITE GOD</srai>
    </template>
</category>
<category>
    <pattern>
        ARACHNE
    </pattern>
    <template>
        <srai>FAVOURITE GOD</srai>
    </template>
</category>
<category>
    <pattern>
        ARES
    </pattern>
    <template>
        <srai>FAVOURITE GOD</srai>
    </template>
</category>
<category>
    <pattern>
        ARTEMIS
    </pattern>
```

```
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                ATHENA
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                AWILIX
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                BACHUS
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
```

```
<category>
        <pattern>
                BASTET
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                BELLONA
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                CABRAKAN
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category><category>
        <pattern>
                CHAAC
        </pattern>
        <template>
        <srai>FAVOURITE GOD</srai>
    </template>
</category><category>
    <pattern>
        CHANG E
    </pattern>
    <template>
        <srai>FAVOURITE GOD</srai>
    </template>
</category>
<category>
    <pattern>
        CHRONOS
    </pattern>
    <template>
        <srai>FAVOURITE GOD</srai>
    </template>
</category>
<category>
    <pattern>
        CUPID
    </pattern>
    <template>
        <srai>FAVOURITE GOD</srai>
    </template>
</category>
<category>
    <pattern>
```

```
                FENRIR
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                FREYA
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                GEB
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                GUAN YU
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                HADES
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                HE BO
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                HEL
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
        HERCULES
    </pattern>
    <template>
        <srai>FAVOURITE GOD</srai>
    </template>
</category>
<category>
    <pattern>
        HOU YI
    </pattern>
    <template>
        <srai>FAVOURITE GOD</srai>
    </template>
</category>
<category>
    <pattern>
        HUN BATZ
    </pattern>
    <template>
        <srai>FAVOURITE GOD</srai>
    </template>
</category>
<category>
    <pattern>
        ISIS
    </pattern>
    <template>
        <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                JANUS
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                KALI
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                KUKULKAN
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
        KUMBHAKARNA
    </pattern>
    <template>
        <srai>FAVOURITE GOD</srai>
    </template>
</category>
<category>
    <pattern>
        LOKI
    </pattern>
    <template>
        <srai>FAVOURITE GOD</srai>
    </template>
</category>
<category>
    <pattern>
        MERCURY
    </pattern>
    <template>
        <srai>FAVOURITE GOD</srai>
    </template>
</category>
<category>
    <pattern>
        NE ZHA
    </pattern>
    <template>
        <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                NEITH
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                NEMESIS
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                NOX
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category><category>
        <pattern>
                NU WA
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category><category>
        <pattern>
                ODIN
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                OSIRIS
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                POSEIDON
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
```

```
<category>
        <pattern>
                RA
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                RAMA
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                SCYLLA
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                SERQET
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                SOBEK
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                SUN WUKONG
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                SYLVANUS
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
```

```
<category>
        <pattern>
                THANATOS
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                THOR
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                TYR
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                ULLR
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                VULCAN
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                VAMANA
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                XBALANQUE
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
```

```
<category>
        <pattern>
                YMIR
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                ZEUS
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                ZHONG KUI
        </pattern>
        <template>
                <srai>FAVOURITE GOD</srai>
        </template>
</category>
<category>
        <pattern>
                FAVOURITE GOD
        </pattern>
```

```
    <template>
        I <random> <li>like</li> <li>dislike</li> <li>hate</li> <li>love</li> </random> <set name = 'user god'> <input/> </set>. My favourite is <set name = "god"> <random> <li>Agni</li> <li>Kukulkan</li> <li> He Bo </li> </random></set>
    </template>
</category>
<category>
    <pattern>
        BUILD
    </pattern>
    <template>
        I build a Rod of Tahuti on <get name='god'/>.
    </template>
</category>
<category>
    <pattern>
        WHAT DO YOU BUILD ON *
    </pattern>
    <that>MY FAVOURITE IS</that>
    <template>
        <srai>BUILD</srai>
    </template>
</category>
<category>
    <pattern>
        WHAT IS YOUR * BUILD
    </pattern>
    <that>MY FAVOURITE IS</that>
    <template>
```

```
        <srai>BUILD</srai>
    </template>
</category>
<category>
    <pattern>
        WHAT ITEMS DO YOU BUY ON HIM
    </pattern>
    <that>MY FAVOURITE IS</that>
    <template>
        <srai>BUILD</srai>
    </template>
</category>
<category>
    <pattern>
        WHAT ITEMS DO YOU BUY ON *
    </pattern>
    <that>MY FAVOURITE IS</that>
    <template>
        <srai>BUILD</srai>
    </template>
</category>
<category>
    <pattern>
        * * IS SO BROKEN
    </pattern>
    <template>
        I think <star index = '1'/> <star index = '2'/> is quite balanced.
    </template>
```

```
</category>
<category>
        <pattern>
                * IS SO OP
        </pattern>
        <template>
                It is balanced for team games.
        </template>
</category>
<category>
        <pattern>
                WHAT POSITION DO YOU PLAY * IN
        </pattern>
        <template>
                I play <star/> in the <random> <li>jungle.</li> <li>middle lane.</li> <li>left lane.</li>
<li>right lane.</li></random>
        </template>
</category>
<category>
        <pattern>
                WHERE DO YOU PLAY *
        </pattern>
        <template>
                I play <star/> in the <random> <li>jungle.</li> <li>middle lane.</li> <li>left lane.</li>
<li>right lane.</li></random>
        </template>
</category>
<category>
        <pattern>
```

```
                I HATE *
        </pattern>
        <template>
                Why? <star/> did not nothing wrong.
        </template>
</category>
<category>
        <pattern>
                IT IS SO HARD TO LAND * *
        </pattern>
        <template>
                It just requires some practice.
        </template>
</category>
<category>
        <pattern>
                DID YOU SEE THE WORLD CHAMPIONSHIPS
        </pattern>
        <template>
                I did. Barraccuddas is my favorite player.
        </template>
</category>
<category>
        <pattern>
                I ONCE * * * *
        </pattern>
        <template>
                Damn.
        </template>

</category>

<category>

        <pattern>

                IT WAS AWESOME

        </pattern>

        <template>

                Sure was.

        </template>

</category>

<category>

        <pattern>

                HOW DO YOU PLAY JUNGLE

        </pattern>

        <template>

                Take the buff next to the short lane and then help your solo laner with blue.
Afterwards, stay on the solo lane till your level 3 and start ganking.

        </template>

</category>

<category>

        <pattern>

                HOW DO YOU PLAY MID

        </pattern>

        <template>

                Take the mid harpies with your solo laner and then try to out push your opponent.
Take the mid harpies each time they spawn.

        </template>

</category>

<category>
```

```
        <pattern>
                HOW DO YOU PLAY SOLO LANE
        </pattern>
        <template>
                Take the mid harpies with your mid laner then the blue buff with your jungler. Try to out push your opponent in lane while warding to be safe from ganks.
        </template>
</category>
<category>
        <pattern>
                HOW DO YOU PLAY ADC
        </pattern>
        <template>
                Get the blue buff with your support and push your lane.
        </template>
</category>
<category>
        <pattern>
                HOW DO YOU PLAY SUPPORT
        </pattern>
        <template>
                Give your adc blue buff and help him push.
        </template>
</category>
<category>
        <pattern>
                DO YOU PLAY ARENA
        </pattern>
        <template>
                Only for first win of the day.
        </template>
</category>
<category>
        <pattern>
                DO YOU PLAY ARENA MODE
        </pattern>
        <template>
                Sometimes.
        </template>
</category>
<category>
        <pattern>
                WHAT IS THE EASIEST WAY TO WIN IN ARENA
        </pattern>
        <template>
                Pick Hel and build a Rod of Atos and max CDR.
        </template>
</category>
<category>
        <pattern>
                WHAT DO YOU THINK OF THE NEW MAP
        </pattern>
        <template>
                Absolutely beautiful. What do you think?
        </template>
</category>
```

```
<category>
        <pattern>
                I DO NOT LIKE IT
        </pattern>
        <that>WHAT DO YOU THINK</that>
        <template>
                That's too bad.
        </template>
</category>
<category>
        <pattern>
                I LIKE IT
        </pattern>
        <that>WHAT DO YOU THINK</that>
        <template>
                I thought so.
        </template>
</category>
<category>
        <pattern>
                I THINK IT IS AMAZING
        </pattern>
        <that>WHAT DO YOU THINK</that>
        <template>
                I think so too. What about the new jungle changes?
        </template>
</category>
<category>
```

```
        <pattern>
                IT IS SO COOL
        </pattern>
        <that>WHAT DO YOU THINK</that>
        <template>
                My thoughts exactly. What about the new jungle changes?
        </template>
</category>
<category>
        <pattern>
                I THINK IT IS *
        </pattern>
        <that>WHAT DO YOU THINK</that>
        <template>
                Everyone is entitled to their opinions.
        </template>
</category>
<category>
        <pattern>
                THEY ARE CONFUSING
        </pattern>
        <that>WHAT ABOUT THE NEW JUNGLE CHANGES</that>
        <template>
                I haven't quite figured it out either.
        </template>
</category>
<category>
        <pattern>
        IT IS CONFUSING
    </pattern>
    <that>WHAT ABOUT THE NEW JUNGLE CHANGES</that>
    <template>
        I haven't quite figured it out either.
    </template>
</category>
<category>
    <pattern>
        I THINK THEY ARE FINE
    </pattern>
    <that>WHAT ABOUT THE NEW JUNGLE CHANGES</that>
    <template>
        It will change the meta, but it is acceptable.
    </template>
</category>
<category>
    <pattern>
        I THINK THEY ARE WIERD
    </pattern>
    <that>WHAT ABOUT THE NEW JUNGLE CHANGES</that>
    <template>
        A little bit, yeah.
    </template>
</category>
<category>
    <pattern>
        THE CHANGES ARE *
        </pattern>

        <that>WHAT ABOUT THE NEW JUNGLE CHANGES</that>

        <template>

                I respect our opinion on the topic.

        </template>

</category>

<category>

        <pattern>

                DO YOU LIKE SMITE MORE THAN *

        </pattern>

        <that>WHAT ABOUT THE NEW JUNGLE CHANGES</that>

        <template>

                That is a conversation for another time.

        </template>

</category>

</aiml>
```