

# **Python Applications in Artificial Intelligence Research**

By Daniel G. Taylor

I've been interested in artificial intelligence for as long as I can remember. While we obviously don't have fully functional cognitive entities to interact with yet, I thought I would research a bit about what is available. I've chosen to specifically limit my search to applications of artificial intelligence research implemented in the Python programming language. I will first explain a bit about Python and then get into the projects I have come across, including projects dealing with semantic networks, natural language processing, knowledge bases, and robotics.

## What is Python?

Simply stated, "Python is an interpreted, interactive, object-oriented programming language." It can be compared to other languages like Perl, Scheme, Java, and Microsoft's .NET languages. Python is free software, compatible with the GNU General Public License.<sup>1</sup>

Python is extremely powerful while having very clear syntax. It supports modules, classes, exceptions, high level dynamic data types, etc. It also has many built in modules to access underlying system calls from within Python programs, and can easily be extended using C or C++. Python can also be embedded into other applications to allow for scripting. Applications such as The GIMP, Blender, Xchat, Inkscape and others do this.<sup>1</sup>

Python is very portable. I regularly write Python applications in GNU/Linux and run them in Mac OSX and Windows without changing a thing. Python is available for GNU/Linux, Mac OSX, Microsoft Windows, Solaris, OS/2, AIX, AS/400, BeOS, Palm OS, QNX, VMS, Sony Playstation, Sharp Zaurus, Windows CE and Pocket PC, among others.<sup>2</sup>

## Why use Python for AI?

Python is an excellent language for text processing and string manipulation. It is also very extensible and can be made very fast by incorporating modules written in C/C++, which gives you the speed of C/C++ with the amazing syntax and other features of Python. Python also has many high level structures like lists, tuples, dictionaries, sets, and more built right in, so you can get right down to your program logic instead of messing with writing all sorts of extra code. Python is also very easy to learn, and is many times suggested as a "beginner's programming language," because of its clean and simple syntax.

Python is included in every major operating system distribution except for Microsoft Windows, and there is hope that that will change with IronPython, a Python implementation that runs in the .NET Common Language Runtime. Write once - run anywhere can be a reality with Python!

Python also happens to be my favorite programming language, which is why I chose to limit my research to it.

## ELIZA revisited

Remember ELIZA, the virtual psychologist developed in the 1960's? One of the first examples of artificial intelligence in Python that I ran across is an implementation of an ELIZA-like bot called Therapist.<sup>3</sup>

The original ELIZA program is described as "a program [...] at MIT which makes certain kinds of natural language conversation between man and computer possible. Input sentences are analyzed on the basis of decomposition rules which are triggered by key words appearing in the input text. Responses are generated by reassembly rules associated with selected decomposition rules."<sup>4</sup>

While it may be somewhat antiquated the therapist does show simple language processing and interaction. Below is a sample session I had with the therapist.

Hello. How are you feeling today?

> Not bad. I'm working on my AI project at the moment.

I see. And what does that tell you?

> Well, probably that I should finish my assignments sooner!

I see.

> I'm not sure you do.

Why do you tell me you're not sure me do?

> I'm not so sure that you are intelligent at all!

How does being not so sure that me am intelligent at all make you feel?

> Somewhat disappointed, sadly.

How does that make you feel?

> quit

Thank you, that will be \$150. Have a good day!

Gee, that therapist sure charges like a real psychologist! It is, however, quite obvious to anyone who plays with the program for a while that it is far from perfect. Even in the short example above we can see a few grammatical mistakes and repetitions (such as the "How does that make you feel" question right after one just like it). Of course, for its time ELIZA was an amazing accomplishment.

## Semantic networks and knowledge bases

Semantic networks describe semantic relations between concepts. One such example is WordNet, a lexical reference system. "English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets."<sup>5</sup>

I've come across a simple semantic network implementation in Python called SemNet that uses entities, relations, and facts to manage information and provides a way to query the system. In practice it reminds me of Prolog. The included demo builds a semantic network of facts and rules about a few animals and produces a few example queries. Below are some example queries from the demo network.<sup>3</sup>

ape is a fish? 0

minnow is a fish? 1

ape bigger than minnow? 1

what can swim? [fish]

Note that in the examples a zero denotes false and a one denotes true. The brackets around "fish" show that it is a list (with one element). This is all very similar to expert systems, which can be used for things like diagnosing patients or managing stocks given enough semantic knowledge to base its decisions off of. With this Python module quite a lot is possible, and the syntax is very simple.

This of course brings us to knowledge bases, which are a special kind of database for knowledge management. This is of interest because I found an extensive English and French knowledge base of words, phrases, concepts, and common sense assertions called ThoughtTreasure which has an implementation written in Python. Not only that, but their system can be queried online so you can try it out without having to download anything (the online query system uses Python CGI scripts).<sup>6</sup>

Some example queries posted on the Thoughttreasure site conclude that:

Soda is a drink.

People have necks.

Excellent food is called quality food

A play lasts about two hours.

One hangs up at the end of a phone call.

After a few of my own experiments I've concluded that the system is actually quite good, though I was able to find many words it did not have in the database. One of my queries returned that the result of eating is satiation. Another returned that a meal in a restaurant will cost about \$20. Yet another returned that a doctor does checkups and medical examinations of body parts such as the head, lungs, chest, throat, and eyes, and that there are many different types of doctors.<sup>6</sup>

Semantic networks are quite useful in real world situations. Having Python implementations of semantic networks, knowledge bases, and expert systems that can be embedded in other software easily is great, because it makes otherwise extremely complicated systems easily accessible by many.

## Language Identification

Moving into the concept of natural language recognition and processing I came across an interesting utility called GuessLang that will guess the language used on any web page using various methods. It is a simple command line utility that will return the language it has guessed as a two character language code (en for English, de for German, etc). Below are some of the results I get using the included library (that the command line client also uses) in Python's interactive shell.<sup>7</sup>

```
>>> from logilab.guesslang.guess_language import *
>>> identify_language_from_url("http://programmer-art.org")
'en'
>>> identify_language_from_url("http://www.fsep.org")
'en'
>>> identify_language_from_url("http://www.heise.de")
'de'
>>> identify_language_from_url("http://www.logilab.fr")
'fr'
```

As you can easily see GuessLang is pretty good at guessing languages. Before anyone thinks it is using the URL or language meta tags of pages I will assure you it does not. Feel free to look at the code, as it is free software just like the rest of these projects.

GuessLang is actually an offspring of the Narval project on the same site. The Narval project is a free software agent framework that lets anyone setup intelligent personal assistants that can help you in your daily work. You can apparently contact such agents via email, web, telnet, phone, and graphical interfaces, though I haven't created anything with the framework. It is supposed to be pretty easy to create actions and plugins for Narval using Python. Because of the project's huge scope I won't go into it any further, but it may be worth investigating if you are interested. It certainly sounds interesting to me.<sup>7</sup>

## Natural Language Processing

The above examples didn't really process natural languages to the fullest in my opinion. I did happen to find a Python library in my research that is extremely powerful for language processing called the Natural Language Toolkit. It is a "suite of Python libraries and programs for symbolic and statistical natural language processing. NLTK is ideally suited to [...] natural language processing, empirical linguistics, cognitive science, artificial intelligence, information retrieval, and machine learning." There is a lot of useful documentation with this project, so don't worry too much about the large scope of the project.<sup>8</sup>

Examples are given in the NLTK documentation of how to classify parts of sentences, how to write a simple application that finds the next best word for a sentence being typed using the included English data sets and how to generate whole stories from nothing to a few key words. There are tutorials for tokenizing, tagging, chunk parsing, classification, regular expressions, and building probabilistic systems.<sup>9</sup>

Another natural language processing system I came across is PyAIML, which is a Python interpreter for the Artificial Intelligence Markup Language (which is what powers AliceBot as well).<sup>10</sup> Perhaps the biggest project built on top of PyAIML is Howie, a chat bot by the same developer that made PyAIML (in fact Howie is the reason PyAIML was created). Howie supports natural language communication over instant messenger protocols like Jabber/XMPP and the proprietary AIM and MSN protocols (and even has an XML-RPC interface!). Other protocols like IRC will be supported in the future. This stuff is seriously cool! Below you will see a chat I just had over Jabber/XMPP with a bot using Howie's code.<sup>11</sup>

Dan: hey  
DanBot: Can I help you?  
Dan: I'm not sure  
DanBot: Is there anything I can say to help you make up your mind?  
Dan: What's your name?  
DanBot: My name is DanBot.  
Dan: what time is it?  
DanBot: Time does not exist.  
Dan: what?  
Danbot: Perhaps I'm just expressing my own concern about it.  
I've been waiting for you.  
A deeper algorithm is needed to respond to that correctly.  
What is your occupation?  
Dan: I'm a student  
DanBot: Where do you go to school?  
Dan: Edinboro University  
DanBot: How do you like school at Edinboro?  
Dan: It's nice.

From my tests Howie seems to be the best Python chat bot around. Howie's responses are generally much more complex than those of e.g. ELIZA and other such bots and he rebounds much better when confronted with topics which he doesn't understand. Howie also seems to remember things while still running.

## **Intelligent robotics design**

The next project I came across, and last project I will be writing about, is called Pyro. "Pyro is a library, environment, graphical user interface, and low-level drivers to explore artificial intelligence and robotics using the Python language." There is even a LiveCD available so that you don't have to install anything to get started with Pyro - just boot from the CD and start hacking!<sup>12</sup>

Pyro is probably the project I know the least about of all the projects I have written something about. I just had to write something, even if it isn't much. Pyro just has that high level of coolness.

I've never been big into robotics - not by choice, I've just never had the resources available. Pyro comes with several simulators, including fully 3D OpenGL environments that can simulate multiple physical robots running your programs. It also comes with a lot of documentation, much of which is in a class or tutorial like style, and many even include ideas or exercises you can try to further your knowledge and experience. This by itself is an amazing starting point on which to base your robotics research.<sup>12</sup>

Coincidentally, this summer there will be a free Pyro workshop in Philadelphia (3rd - 5th August 2005). Anyone interested in Pyro should try to attend. They will even pay for your room to stay there! I will be in

Germany so I won't be able to make it, but I will get to attend GUADEC and LinuxTag while in there, so not all is lost!<sup>12</sup>

## **Conclusion**

In conclusion there appears to be a healthy community of artificial intelligence related software and developers that use Python. I have just dipped into the pot; for each project I've talked about I am sure I have missed dozens of others. It is left as an exercise for the reader to find the rest!

I would say Python is a great language for artificial intelligence development and hope to see it used for such things in the future. I've always had an interest in chat bots and will probably play with some of the code I found. I hope I've inspired a few others to do so as well! Thank you for reading this far!

## References

1. What is Python? <<http://www.python.org/doc/Summary.html>>
2. Python Downloads <<http://www.python.org/download/>>
3. Artificial Intelligence in Python <<http://www.strout.net/python/ai/>>
4. ELIZA - A Computer Program For the Study of Natural Language Communication Between Man and Machine <<http://i5.nyu.edu/~mm64/x52.9265/january1966.html>>
5. About WordNet <<http://wordnet.princeton.edu/>>
6. ThoughtTreasure <<http://www.signiform.com/tt/htm/tt.htm>>
7. Logilab Artificial Intelligence Projects <<http://www.logilab.org/categories/ai>>
8. Natural Language Toolkit <<http://nltk.sourceforge.net/>>
9. NLTK Tutorials <<http://nltk.sourceforge.net/tutorial/index.html>>
10. PyAIML - A Python AIML Interpreter <<http://pyaiml.sourceforge.net>>
11. Howie the Chatterbot <<http://howie.sourceforge.net>>
12. Pyro - Python Robotics <<http://emergent.brynmawr.edu/~dblank/pyro/>>

## Copies of this document

Please note that the XHTML version of this document is enhanced with hyperlinks, acronym explanations, and other goodies.

- XHTML - <http://programmer-art.org/dan/python-ai.html>
- PDF - <http://programmer-art.org/dan/python-ai.pdf>