

Lab: Wolfram Alpha Automated Queries

CSI 3305: Introduction to Computational Thinking

February 16, 2011

1 Introduction

Wolfram Alpha seeks to be nothing less than the world's premier *knowledge engine*, a search engine that parses natural language queries and computes results on the fly. As a young scientist, you have an idea for a new method of artificial intelligence and feel that perhaps Wolfram Alpha can help your idea get off the ground. Your AI engine needs a core set of facts to begin with and since Wolfram Alpha has an automated API protocol, you decide to use the engine to create a small set of facts for your AI program.

2 Problem Statement

Your task is to use the Wolfram Alpha python interface to submit a set of queries via the internet, do some preliminary parsing on the returned results, and collect the output to a text file.

3 Programming

4 Tools

You will be using the Python programming language and Python interpreter for this lab. The code will be written using a text editor, and the interpreter will be called using the command line.

5 Setup and Programming

1. To begin, download the Wolfram Alpha API files from the course website.
2. Next, create a new text file and label it `wolfram_query.py`.
3. Open the text file in the text editor of your choice (such as Notepad or Vim.)
4. Enter the following code at the top of your file:

```
import sys
import json
import wap
```

This code will import python libraries as well as the Wolfram Alpha wap module.

5. Next, we need to define some variables that will be used to query the Wolfram Alpha website. Type the following code into your file:

```
server = 'http://api.wolframalpha.com/v1/query.jsp'  
appid = 'YJXA5P-5XU5684VYJ'
```

The first is the server address, and the second is a unique appid key allowing us to send queries.

6. Next, we will create our main function. Enter the following code, beneath what you've already entered:

```
''' Functions '''  
def main(filename):  
    global server, appid  
    queries = open(filename, "r").readlines()  
    for query in queries:  
        waeo = wap.WolframAlphaEngine(appid, server)  
        wa_query = waeo.CreateQuery(query)  
        result = waeo.PerformQuery(wa_query)  
        waeqr = wap.WolframAlphaQueryResult(result)  
        jsonresult = waeqr.JsonResult()  
        j = json.loads(jsonresult)  
        if bool(j[2][1]):  
            numpods = int(j[5][1])  
            for i in range(11, 11 + numpods):  
                try:  
                    print j[i][2][1],":", j[i][7][2][1] \  
                        if len(j[i][7][2]) > 1 else j[i][7][2]  
                except:  
                    pass  
        else:  
            print "No results returned."
```

This method takes in a filename that includes a set of queries, one on each line, passed in as a parameter from the command line. It then opens the file, extracting the individual queries. Next, the queries are iterated over, and sent to the Wolfram Alpha website, using the API. After results are returned, we check if the query was successful. If so, we get the number of sub-results returned (**numpods**), and iterate over those, attempting to parse them as we go. Everything is printed to the screen, which we will capture using redirection.

7. Lastly, we enter code that calls the main function:

```
if __name__ == "__main__":  
    if len(sys.argv) < 2:  
        print "USAGE: wolfram_query.py input_file_with_queries.txt"  
        sys.exit()  
    else:  
        main(sys.argv[1])
```

8. Save the file and open the command line (From the Windows menu bar: Start > Run... > cmd) Navigate to the folder where you created your file (use the following command: cd "C:\Folder\where file\is".)

9. From the command line, type the following to run your program and hit enter:

```
python wolfram_query.py queries.txt
```

The file queries.txt is a simple text file that includes a few queries, one on each line, to be sent to the Wolfram Alpha website.

10. If queries.txt does not exist, create the file in the same folder and enter the following three queries:

```
pi
GDP of India
Who was Napoleon?
```

11. To capture the data to a file, run the program and redirect to a text file names results.txt. To do so, from the command line, type the following to run your program and hit enter:

```
python wolfram_query.py queries.txt > results.txt
```

The results.txt file should now contain your data.

6 Questions

1. What was the birthplace of Napoleon Bonaparte, returned fro Wolfram Alpha?
2. In terms of world GDP, what is India ranked?
3. What are the first four entries of the “Continued Fraction” for pi?
4. What is India’s 2004 unemployment rate estimate?