edbullen / **NLPBot**

Simple ChatBot introducing NLP and Machine Learning for Classification of Sentences

22 commits | **1** branch | **1** release | **1** contributor

Branch: **master** | New pull request | Find file | Clone or download

botuser and **botuser** fix to chat_flow() in chatbot.py to exclude anwers with weight < 2 — Latest commit `0d32dae` on Feb 19

__pycache__	first release	a year ago
analysis	first release	a year ago
config	updated NLP bot first commit	a year ago
slides	minor slides update	a year ago
README.md	wording tweaks for README and slides	a year ago
RFmodel.ml	First release of RF model	a year ago
all_triples.py	first release	a year ago
botserver.py	removed some BOT> prompts	4 months ago
chatbot.py	fix to chat_flow() in chatbot.py to exclude anwers with weight < 2	4 months ago
dataDump.py	first release	a year ago
dataLoad.py	first release	a year ago
features.py	tidy up redundant comments	a year ago
featuresDump.py	first release	a year ago
getConfig.py	first release	a year ago
mlClassGenerateRfModel.py	added RF ml gen util and all test utils	a year ago
pingDB.py	first release	a year ago
pwdutil.py	first release	a year ago
python_server_config.md	fix omissions in server build doc	a year ago
setupDatabase.py	updated NLP bot first commit	a year ago
simpleclient.py	first release	a year ago
testBulkGrammar.sh	added RF ml gen util and all test utils	a year ago
testClassifyModel.py	added RF ml gen util and all test utils	a year ago
testGetAnswer.py	added RF ml gen util and all test utils	a year ago
testGetGrammar.py	added RF ml gen util and all test utils	a year ago
testStoreStatement.py	added RF ml gen util and all test utils	a year ago
utils.py	Fixed botserver bugs for NLP change and added logging	a year ago

README.md

This is a simple Chatbot written in **Python 3.5** with a MySQL database backend. The code builds on my other SimpleBot demo (https://github.com/edbullen/SimpleBot) and introduces some NLP with Python NLTK and basic Machine Learning capabilities to demonstrate Sentence Classification using the NLTK and scikit-learn. The Stanford CoreNLP package, written in Java, is also used to parse grammar and extract sentence topics, subject, object etc.

The ChatBot conversation operates in 3 modes

- Chat (just learned responses from previous exchanges)
- Statement (receive a statement of fact and store it away)
- Question (receive a question and attempt to answer it based on previously stored statements)

As it currently stands (May 2017), the functionality is far from perfect, but it demonstrates the concepts of Natural Language Processing, Sentence Classification and a very basic level of Natural Language Grammar processing.

This version is still at a basic experimentation level - there is no concept of authentication, security etc.

Python Library Dependencies

- pymysql
- nltk
- numpy
- pandas
- scipy
- scikit-learn
- Stanford CoreNLP Parser This is a Java package that needs to be download and located in suitable dir for future ref
- Java - tested with Java 8, java version "1.8.0_131"

Files and Components

Core Functionality

- `chatbot.py` - main ChatBot library
- `utils.py` - generic function utilities used by ChatBot (config, DB conn etc)
- `features.py` - library for extracting features from sentences using NLTK
- `botserver.py` - Multi-Threaded server to allow multiple clients to connect to the ChatBot via network sockets
- `simpleclient.py` - Simple network sockets client to connect to `botserver`

Default `botserver.py` logging location is

```
./log/bostserver.log
```

Setup and Test

- `./config/config.ini` - template config file, requires editing before starting the chatbot for the first time.
- `pwdutil.py` - store an encoded password for connecting to the database schema.
- `setupDatabase.py` - drop and recreate the database tables (existing data gets lost).
- `pingDB.py` - test the database configuration: create test table, insert data, query it, drop the test table.

- `mlClassGenerateRfModel.py` - generate a scikit-learn Random Forest Model for sentence classification based on input CSV. Default output file-name is "**RFmodel.ml**"

Tools and Utilities

All Dump and Load utilities use the `./dump` subdirectory by default.

- `dataDump.py` - Dump out a database table in CSV format
- `dataLoad.py` - Load a database table in CSV format
- `featuresDump.py` - read in a CSV of sentences, dump out features using `features.py` into a CSV
- `testClassifyModel.py` - test the Sentence Classification logic and Model for a given sentence (using the pre-built Random Forest `RFmodel.ml` model)
- `testGetAnswer.py` - test retreiving an Answer from the chatbot for given sentence
- `testGetGrammar.py` - extract grammar structure for given sentence (using Stanford CoreNLP package)
- `testStoreStatement.py` - parse and store a given Statement sentence in the database.

Install and Setup

Details of installing dependancies for the NLPBot to function are documented in the python_server_config.md note in this repo.

In summary, for a new install, the following steps are required:

1. Install Python 3.5
2. Install pyMySQL
3. Install NLTK
4. Install Machine Learning Libs
5. Create the Linux BotUser
6. Install GIT
7. Install Java 8
8. MySQL Database Server Configuration
9. Install Bot Code and Configure
10. Install the Stanford CoreNLP Package
11. Configure the botuser ./config/config.ini file
12. Start BotServer

These steps are documented in detail here

Sample config.ini Configuration File Contents

```
[MySQL]
server: localhost
dbuser: nlpbot
dbname: nlpbot
dbcharset: utf8mb4

[Server]
listen_host: 0.0.0.0
tcp_socket: 9999
listen_queue: 10

[Java]  #required for Stanford CoreNLP
bin: /usr/bin/java

[StanfordNLP]
corejar: /home/botuser/StanfordParser/coreNLP/stanford-corenlp-3.7.0.jar
modelsjar: /home/botuser/StanfordParser/coreNLP/stanford-corenlp-3.7.0-models.jar
```

```
[DEBUG]
assoc: False
weight: False
itemid: False
match: False
server: False
answer: False
```

Starting the BotServer

The chatbot can be started with a multi-threaded server scheduler (`botserver.py`) that listens for connections on a TCP port. This is a very simple "bare-bones" multi session framework with no authentication and just relying on TCP sockets for connection.

Remote TCP Socket Connection requests are given a thread and their own session connection.

The botserver gives each session a connection to the shared database server.

```
nohup python3.5 botserver.py &
```

Logging

Bot server output is logged to

```
~/NLPBot/log/botserver.log
```

Stopping the Server

```
$ ps -ef | grep botserver
botuser  **22915** 22854  0 08:15 pts/0    00:00:01 python3.5 botserver.py
botuser  23165 22854  0 08:54 pts/0    00:00:00 grep botserver
kill -9 22915
```

Local Client Connect

```
python3.5 simpleclient.py -a localhost -p 9999
```

Remote Client Connect

Make sure the botserver port is allowed through the firewall.

```
python3.5 simpleclient.py -a 192.168.10.101 -p 9999
```

Sample Conversation

This sample conversation shows the capabilities of the NLPBot ChatBot as of 1 May 2017. There are some good examples of limitations which are most noticeable around the very basic Question mode (looking for a previously learnt Statement). It does however generally determine whether it is "just chatting", being given a Statement to store or being asked a question.

```
$ python simpleclient.py -a 192.168.10.1 -p 1234
Connecting to server 192.168.10.1  at port  1234
>>> Hi
Bot> hello
>>> The latest Python source distribution is always available from python.org at https://www.python.org/downloads/
Bot> Shall I store that as a fact for future reference?  ("yes" to store)
>>> yes
```

```
Thanks, I've made a note of that.
>>> There is a Python newsgroup comp.lang.python and a mailing list: python-list                          .
Bot> Shall I store that as a fact for future reference?  ("yes" to store)
>>> yes
Bot> OK, I've stored that information.
>>> Where can I get the latest Python source code from?
Bot> Sorry, I can't find an answer to that.
>>> Where is the latest Python source distribution?
Bot> The latest Python source distribution is always available from python.org at
https://www.python.org/downloads/
>>> Thanks.
Bot> How are you?
>>> I am fine thanks
Bot> How are you?
>>> Good
Bot> Please train me - enter a response for me to learn (or "skip" to skip)'
>>> skip
Bot> OK, moving on...
>>> Is there a Python newsgroup or mailing list?
Bot> There is a Python newsgroup comp.lang.python and a mailing list: python-list.
>>> quit
Connection closed
```

PyData 2017 Slides

GitHub Preview Viewer for HTML Slides