

# **The Chatbot For Stocks' Searching**

Xu Tianhong  
Electrical Engineering  
Beihang University  
1.31.2019

## Abstract:

This Program aims at building a chatbot which can help the user get information about stocks. The chatbot should interpret the colloquial or literary sentences sent from users and get what they mean and what they need to know, then create appropriate response to answer them. The responses are also based on the statements of the conversation, which enables user to search in steps. Technologies used in the chatbot contain RASA NLU, SPACY and other tools for Natural Language Recognition. The code is based on PYTHON.

## 一、Background

As Artificial intelligence and especially machine learning is becoming a popular method to deal with all kind of problems, Natural language processing(NLP) technology has also been a hot method used in every fields. NLP is a subfield of computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data.[1] On the other hand, trading stocks online has been a daily activity in the contemporary lives, so does searching informations about stocks on the internet. In this program, we built a chat robot to mix the two field above together, create a tool with NLP technology to search stocks online.

## 二、Python module required

### lexfinance:

lexfinance is a Python module to get stock data from IEX API 1.0. It provides real-time financial data from the various IEX endpoints, as well as historical data. This data includes stock quotes, fundamentals, actions, and information. In addition, support for IEX market data and statistics is provided. [2]

In this program, two functions in iexfinance are required:

`get_available_symbols()`

`get_market_tops()`

They are used to get the symbols, names, prices, shares and volumes of stocks.

### Spacy:

Spacy is a library for advanced Natural Language Processing in Python and Cython. It's built on the very latest research, and was designed from day one to be used in real products. spaCy comes with pre-trained statistical models and word vectors, and currently supports tokenization for 30+ languages. It features the fastest syntactic parser in the world, convolutional neural network models for tagging, parsing and named entity recognition and easy deep learning integration. It's commercial open-source software, released under the MIT license. [3]

In this program, the Spacy are used to draw words from sentences and recognize their parts of speech, in order to gets entitles.

## **RASA\_NLU:**

Rasa NLU is a tool for understanding what is being said in short pieces of text. It is primarily used to build chatbots and voice apps, where this is called intent classification and entity extraction. To use Rasa, some training data have to be provided first, which means a set of messages which already labelled with their intents and entities. Rasa then uses machine learning to pick up patterns and generalise to unseen sentences. [4]

In this program, we use this tool to interpret intention of sentences. There are five intentions set: Asking, Affirming, Deny, Greeting and chatting.

## **Fuzzywuzzy:**

Fuzzy string matching like a boss. It uses Levenshtein Distance to calculate the differences between sequences in a simple-to-use package. Using this tool, we can input two characters string easily get the rate showing their Similarity. [5]

In this program, the tool is used to get the name of stocks from the sentences.

## **Re(Regular Expression):**

Regular expressions are a special sequence of characters that can help you easily check whether a string matches a pattern. [6]

In this program, this module is used everywhere.

# **三、 Explanation for principles**

## **Context&Input:**

To enable the chatbot to create its answer based on context so that the user could ask his question in steps, there are two dynamic parameter showing the information of the conversation before between the user and the chatbot, the State and the Name. The State shows the objective of the last response from the chatbot, there are 3 State set: INIT, CONFIRM\_NAME and ASKING\_INFO. The Name storage the symbol of the Stock they are talking about, it can be "None". Every time when the user input a sentence, the two dynamic parameter are sent together to the response function to create a response and create new dynamic parameter.

## **Interpret intentions:**

The working process of the robot can be split into two part, Interpreting process and Response process. The two process are not separated in the code, but interlude logically instead.

When the first step when the response function in the chatbot get message from the user is interpreting the intention. This process is based on the Python module RASA\_NLU, we first write a sample involving about 100 sentence, 20 sentence for each intentions (Asking, Affirming, Deny, Greeting and chatting). Trained with the sample, the module could create a interpret function to get the properest intention from the set five of the sentence input. All the actions the chatbot would do are based an the intention and the dynamic parameter "State".

## Find stock's name&fuzzy recognition:

If the dynamic parameter "State" is "INIT" and the intention is "Asking", or the "State" is "CONFIRM\_NAME" and the intention is "Deny", the chatbot would try to find a name of stock in the message the user input.

The process of finding name is based on two Python module, Fuzzywuzzy and Spacy. Use Spacy to split sentences into words and find out the words which have specific Part of speech, meanwhile use Regular Expression tools to find words or phrases with specific pattern.

After finding out the strings which could be the name or symbol of stock, use Fuzzywuzzy tools to compare the strings with every stocks' name and symbol.

There are three possible results:

If string totally matches a stock's name or symbol, the chatbot would know "I have got the stock's name".

If the string has a Similarity rate over 75 with a stock's name, or the string has only case differences with a stock's symbol, the chatbot would know "I got a name, but not sure".

Otherwise, the chatbot would know "I haven't find a name".

The next step of the chatbot would based on the result.

To enable the chatbot to recognize some familiar name or abbreviation of stocks, the chatbot would first delete some meaningless suffix of the names and storage them as a set. The strings would also compare to that set. This step is through Regular Expression.

## Response with states:

After interpret the intentions, the chatbot would then act based on the intentions and "State".

If State is INIT and the intention is "Asking", the chatbot would try to find a name of stock in the message the user input and get a result. If the result is "I have got the stock's name", the chatbot would then try to find entitles from the message. Finding entitles is also applying Regular Expression, three entitles are searched: Price, Volume and Share. If at least one entitle are found, the chatbot would find the information answer the user, the state would come back to "INIT" and this question come to an end, otherwise the chatbot would ask for entitles and the State would change to "ASKING\_INFO". If the result is "I got a name, but not sure", the chatbot would answer with a sentence asking if the stock is actually what the user want and the state would change to "CONFIRM\_NAME". If the answer is "I haven't find a name", the chatbot would response to ask for name, the State is still "INIT".

If the State is "CONFIRM\_NAME" and the intention is "Affirm", the chatbot would then try to find entitles from the message. If at least one entitle are found, the chatbot would find the information answer the user, the state would come back to "INIT" and this question come to an end, otherwise the chatbot would ask for entitles and the State would change to "ASKING\_INFO".

If the State is "CONFIRM\_NAME" and the intention is "Deny", the chatbot would try to search name in the message. The way to deal with the result is the same as the situation when the State is "INIT".

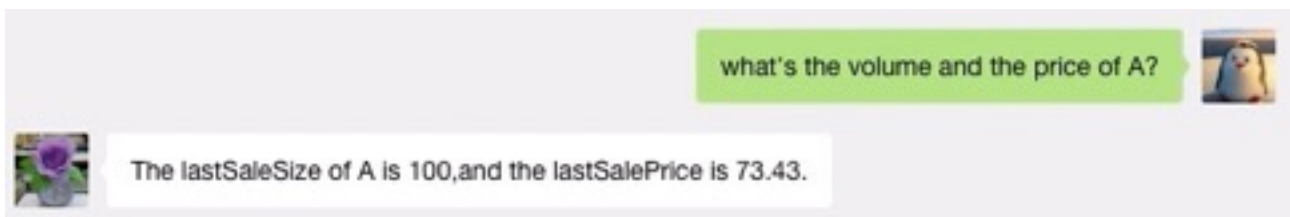
If the State is "ASKING\_INFO" and the intention is "Asking", the chatbot would try to find entitles from the message. If at least one entitle are found, the chatbot would find the information answer the user, the state would come back to "INIT" and this question come to an end, otherwise the chatbot would ask for entitles again and the State would still be "ASKING\_INFO".

## Greeting with policy:

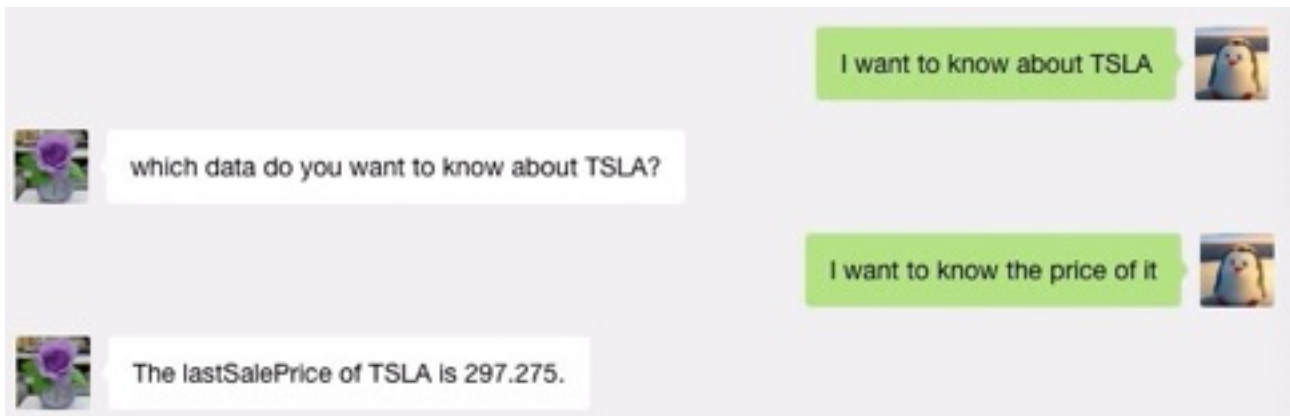
In other combination of State and Intention, the response and the new state would be created with Policy. the Policy is a dictionary, input State and intention and output response and new State. All the response which don't need logical Calculation could be create through it.

## 四、Application Demo

If the user could input a concise statement including clear stock's name and entitles, the chatbot would complete the inquiry in one step.



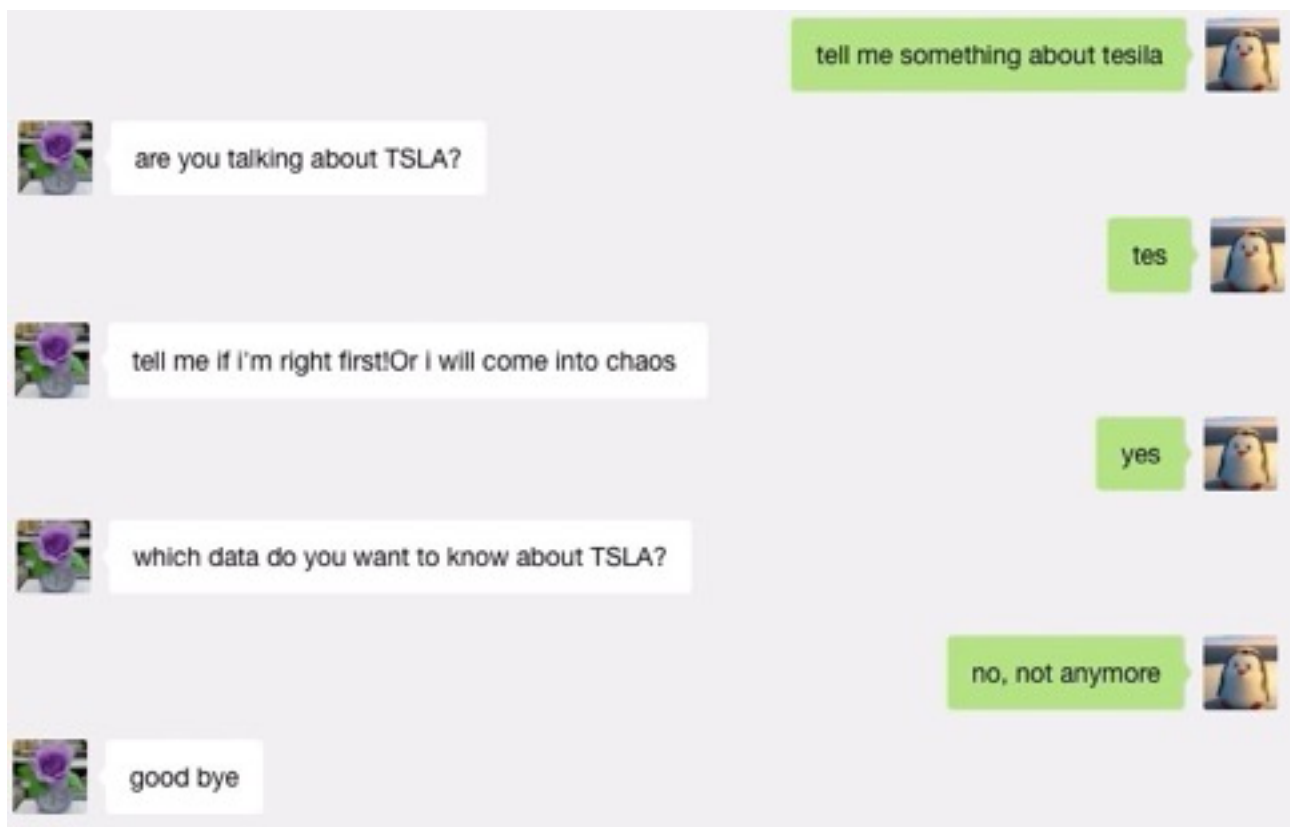
Normally, the user could ask for information in two steps.



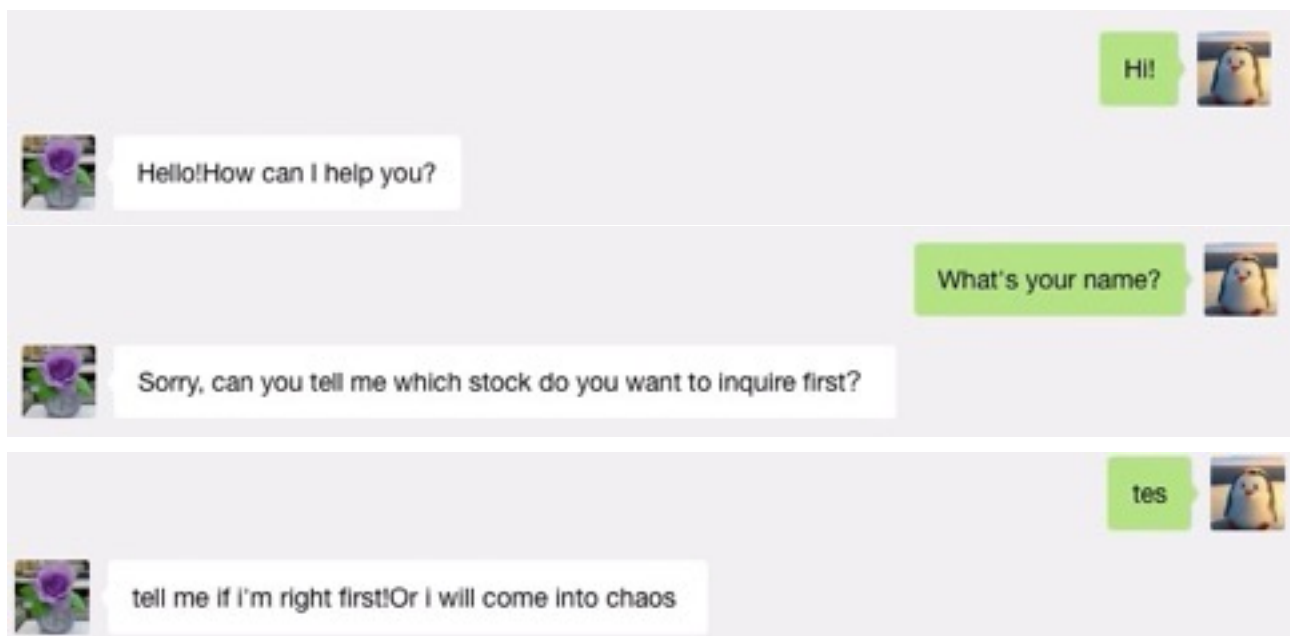
If the user don't know the symbol or the whole name of the stock, just input the abbreviation is ok. Besides, he can also ask more than one entitles in one step.



Even the spelling have something wrong, the chatbot could still recognize it sometimes. The asking process could be end up at anytime when the user input deny message.



The chatbot have few intelligence on gossip, when the user input message which has nothing to do with stocks, the chatbot would lead the topic to stocks.



## 五、Summary and extension

Through this program, I have learned several technologies about natural language processing and a general understanding has been established on it. Professor Zhang taught me many knowledges on it involving:

- Multiple alternative answers to the same question and a default solution.
- Answering questions through regular expressions, pattern matching, keyword extraction, syntactic transformation, etc.
- Interpreting users' intentions by regular expressions, nearest neighbor classification or one or more schemes of support vector machines.
- Recognizing entitles by pre-built named entity type, role relationship and dependency analysis.
- The construction of basic chat robot system based on Rasa NLU;
- Query database and use natural language to explore database content (extract parameters, create queries, respond).
- Single-round multiple incremental query technology based on incremental filter and screening negative entity Technology.
- Multi-round multiple query technology of state machine and provide explanations and answers based on contextual questions.
- Multi-Round Multiple Query Technology for Dealing with Denial, Waiting State Transition and Undetermined Action.

Though I haven't use all of them in my program, they helped me a lot on learning the principles of natural language processing, and I believe I would use them in the future.

This chatbot is still having a huge distance from perfection, and It may be improved in the future in a few way:

- The sample used in the intention interpreting step is too small, so the intentions I set are general. If we got a sample big enough, we could split those intention into concrete and clear ones. For example, "Asking for stocks", "Asking for entitles", "Affirm with nothing", "Deny with response".

- The chatbot don't have a function to claver with the user now, but the intention interpreting step can distinguish chatting and asking. So the chatbot could be joint up into other big chatbot, like Seri. In this situation, the program is a module in a big chatbot program, and only applied when the user is asking about stocks.

## References:

[https://en.wikipedia.org/wiki/Natural\\_language\\_processing](https://en.wikipedia.org/wiki/Natural_language_processing) [1]

<https://pypi.org/project/iexfinance/0.3.1/> [2]

<https://pypi.org/project/spacy/> [3]

<https://pypi.org/project/rasa-nlu/> [4]

<https://pypi.org/project/fuzzywuzzy/> [5]

<http://www.runoob.com/python/python-reg-expressions.html> [6]