Natural language process and chatbot application

Abstract:

In this report, we seeks to build a chatbot which are able to help user to check stock information. In this project, I mainly used five approaches to accomplish this task.

1.Use predefined dictionary to randomly select response
2.Use regular expression to extract entity
3.Use rasa nlu to parse user input, get intents and entities
4.Use multi turn response to get enough information for certain operation
5.Use state machine to well organize the process

Started from 1955, the first digital computer was invented with only 3600 transistors equipped to nowadays over five hundred million transistors for common personal computer, the capacity and speed of computer are even powerful than human brain in certain aspects. So some applications like chat bot are used in customer support, ordering food, and many other areas. But there are some concerns like the accuracy of understanding human language, the inflexibility of robot. In the future, maybe all these flaws will be fixed and more human-liking chat bots will be invented.

Introduction:

The first so called Artificial Intelligent was invented by the British computer scientist Alan Turing during the Second World War. Even the “Bombe” was the most powerful machine at that time, it could take months to crack the “Enigma” code which was used by German forces to send messages securely. With the great success of “Bombe”, many organizations and government stared to take AI seriously, and many programming languages were developed for machine learning. However, as the whole world’s computer scientists trying to exploit the potential of Artificial Intelligent, they found it incredibly difficult to make machine “intelligent” without processing enormous amount of data. But back to that time, big data base is not easily accessible and computers were not developed enough to deal with these data. Therefore, ‘AI Winters’ came and the whole development of AI was stoped for a while. Until in the past 15 years, with the advance of Internet and well developed computers, people can easily get data and train them in their programs. Nowadays many big companies like Amazon, Google, Baidu are taking a huge commercial advantages from it. In this report, we seeks to build a chatbot which are able to help user to check stock information.

Tools used:

REGULAR EXPRESSION:

A regular expression is a sequence of characters that define a *search pattern*. Usually this pattern is used by string searching algorithm for "find" or "find and replace" operations on strings, or for input validation. Regexes are useful in a wide variety of text processing tasks, and more generally string processing, where the data need not be textual. Common applications include data validation, data scraping(especially web scraping), data wrangling, simple parsing, the production of syntax highlighting systems, and many other tasks.

In this project, I use regular expression to extract user inputs' entities, so that the program will do different operations according to the entities.

RASA NLU:

Rasa NLU is an open-source natural language processing tool for intent classification and entity extraction in chatbots. Its main purpose is, given an input sentence, predict an intent of that sentence and extract useful entities from it. Intent dictates how the chatbot should respond to an input from a user. Entities are used to make responses more customized. Training data is the most important part of using rasa. A huge training dataset with accurate intent and entities can make parsing more accurate. In this program, I used RASA NLU TRAINER to customize my training data, so that it would fit in the context.

STATE MACHINE:

A finite state machine is a mathematical abstraction used to design algorithms. In simple terms, a state machine will read a series of inputs. When it reads an input it will switch to a different state. Each state specifies which state to switch for a given input. In this program, I used state machine to react differently according to the current state and user's input.

IEX FINANCE:

IEX finance is a powerful tool that enables user to access financial data for multiple asset classes. In this program, I used IEX finance to get real time price, historical data, volume and other informations of stocks.

WXPY:

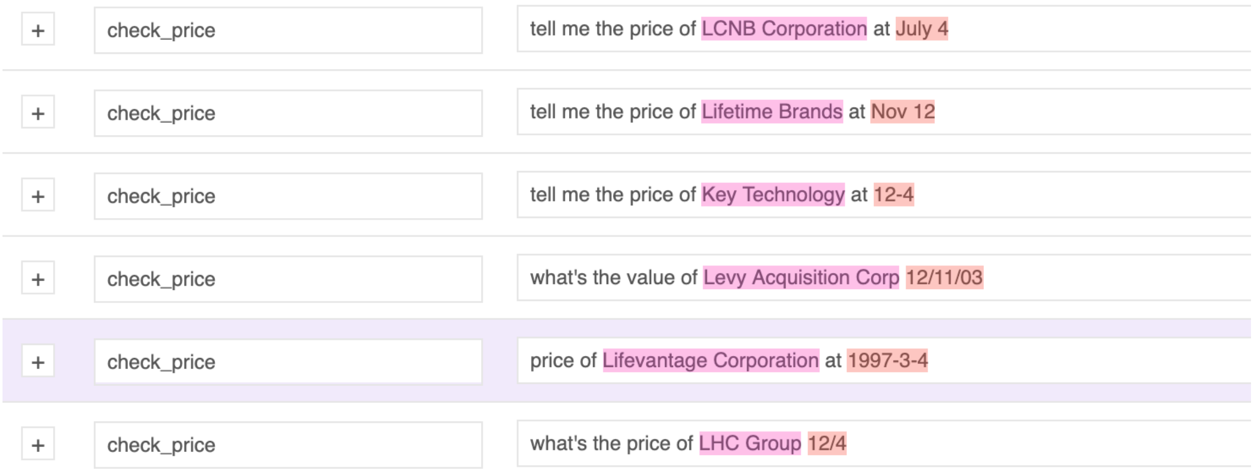
WXPY allow us to link our python program with a WeChat account. Any messages received from WeChat are able to be sent to the python program, and the returned values will be sent to user through WeChat. In this program, I used it to receive user input, parse messages and provide corresponding responses to user through WeChat.

program design:

First I create the whole structure using state machine. Four different states are set.
INIT = 0: initial state, this is the where the program is when it starts.
PEND = 1: pending state, at this state, user are asked to provide some information, otherwise it will not move to the next state.
DO_OPERATION = 2: at this state, the program has received all required information and is able to generate answer and send it to user.
END = 3: conversation end, the program is stoped unless it moves to the INIT state.

After all states are set up, I concluded five intents that could summarize all users' inputs, and in different states, even the same intent were given, responses will be different.
greet: state unchanged
bye: move to END state
pickStock: if user provide ambiguous target stock, move to PEND state, else move to DO_OPERATION state.
check: if previous state was DO_OPERATION and target stock didn't change, program will use the same information about the target and provide required response, state unchanged. Else it will specify a new target, if ambiguous target were provided, move to PEND state, otherwise do response.
unrelated: all other intents will be considered unrelated, and no concrete response will be returned.

In the next step, I need to parse the user's input. I used rasa_nlu to interpret input, get intent and entities. The most important part of getting an accurate answer is the proper training data. To fit in the context, I used RASA NLU Trainer to make my own training data.

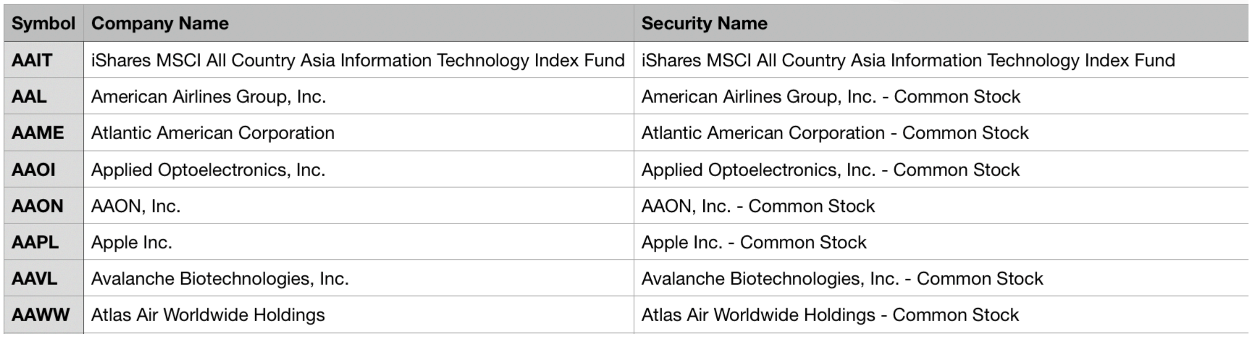


+	check_price	tell me the price of LCNB Corporation at July 4
+	check_price	tell me the price of Lifetime Brands at Nov 12
+	check_price	tell me the price of Key Technology at 12-4
+	check_price	what's the value of Levy Acquisition Corp 12/11/03
+	check_price	price of Lifevantage Corporation at 1997-3-4
+	check_price	what's the price of LHC Group 12/4

……

I wrote a function getTrainer() which will return interpreter. The interpreter was trained by training data and by calling interpret(message, interpreter) we can use it to get intent and entities from a string.
Next, we have received user's input and successfully parsed message, we'll give users their desired information or response.

If the intent is "greet", we'll call greeting_respond(entities), which will randomly select one greeting response and return it. If user introduced his name, the name will be stored in a global variable.
If the intent is "goodbye", we'll call goodbye_respond(user_name). It will randomly return one goodbye response from response dictionary, if user introduced his name, the name will be mentioned.
If the intent is "pickStock", we are going to specify an exact target, a response will be returned by target_respond(entities). User can provide the company name or stock symbol, even part of the company name of the company they want to know and the program will store the formal company name and stock symbol for further use. I used a csv file of all company names and symbols and a function get_symbol(name) to search from the file.



Symbol	Company Name	Security Name
AAIT	iShares MSCI All Country Asia Information Technology Index Fund	iShares MSCI All Country Asia Information Technology Index Fund
AAL	American Airlines Group, Inc.	American Airlines Group, Inc. - Common Stock
AAME	Atlantic American Corporation	Atlantic American Corporation - Common Stock
AAOI	Applied Optoelectronics, Inc.	Applied Optoelectronics, Inc. - Common Stock
AAON	AAON, Inc.	AAON, Inc. - Common Stock
AAPL	Apple Inc.	Apple Inc. - Common Stock
AAVL	Avalanche Biotechnologies, Inc.	Avalanche Biotechnologies, Inc. - Common Stock
AAWW	Atlas Air Worldwide Holdings	Atlas Air Worldwide Holdings - Common Stock

……

If ambiguous company name are provided, like "Apollo", there are 11 company names contain this word, the function user_selection(symbols) will show user all satisfied company names and symbols, also state will move to PEND. The function get_selection(user_choice) will identify the users choice. I used regular expression, so that it support different format and the program becomes more personalized.
If the intent is "check", it will specify the target first like what we did above. The it will check wether the target is changed and time or specific operations are provided and use iexfinance.stock package to get information about the target. Different time format are supported here by using dateutil.parser package. formal_time(times) was called to get date object which can be used in get_price(symbol, times) function to check the information at the given time.

The final step is to get user's input and show output throw WeChat.
I used wxpy package to connect the program with my WeChat account.

user = bot.friends().search(XXX) was used to make user XXX able to access to the chat bot. A decorator @bot.register(user, TEXT) makes the following function able to get user input and the function returned value will be sent to the user directly.
Finally, the chat bot is done.

Reflection:

During this project, I realized some limitations about regular expression and rasa.
For regular expression, the each pattern has to be built manually, in order to extract entities accurately, we need to compile many patterns to cover all situations, which is not a trivial test.
As for rasa, a big data base is required for accurate information extraction. In some situation, the interpreted message is very similar to training data, but may have totally different intent. In this case a huge amount of training data will play a pivotal role for machine to learn the difference. Training data in appropriate context is also valuable. A small amount of training data in perfect context can be much more efficient than many unrelated datas.

Conclusion:

As stated by Moore’s law, the cost and speed of computer will be cheaper and cheaper. But in contrast, human resources becomes more and more expensive. So AI like chat bot is a good replacement of human in many work place. Compared to human labor, there are some advantages that you cannot ignore. The first one is time saving, for both company and their clients. No line busy situation happens in customer support, no human resources allocation problem anymore. The second one is great customer satisfaction. Robot never have temper. No matter what customer asked or did, it will always be patient and positive.
That’s why many tech giants like Google, Facebook, Microsoft, IBM, and Amazon are developing their own chatbots, and chat bot will play a key role in the future.