Chatbot Project

09/13/2019

Jiacheng He

Index

1. Project Overview

1.1. Abstract

In recent years, natural language processing(NLP) has become more and more important in the field of human-computer interaction, it lets people work directly on the device or software through the human natural language. However, the human language is often ambiguous, contextual and implicit and the computer program itself could not clearly recognize the true semantics of the human language. Therefore, we need to analyze and process natural language through a series of methods, let the machine understand the meaning that human users want to express and make feedback automatically.

1.2. Project Introduction

After a month of learning, this project to use the API to get the card information of Hearthstone and design a chatbot to allow users to find the all of cards information from Hearthstone. The project use the telegram to implement the interaction UI for user, and extract the user-entered keywords to match the data through various methods learned this month to make the proper feedback.

2. Knowledge Review

2.1. Simple Echo Chatbot

A simple Echo chatbot will give users a very direct feedback. For instance, if a user type in “hello”, the echo chatbot could directly make the response “hello” as well. If you want to make the echo chatbot more personalized and interested, the program can build a dictionary containing many alternative responses and randomly select one response for the user. Furthermore, the program could create rules of dictionary to match the input statements through the regular expression to get corresponding response for users. In this project, I use this simple concept to deal with the simple feedback from the user to say hello or goodbye to the chatbot and so on.

2.2. Intention Recognition

The task of Intention Recognition is to figure out the intention from human languages. It seems very simple, but in practice, human language often contains implicit, contextual or ambiguous statements. It is very difficult to extract intent from these statements.

This research mainly introduces three methods for intention recognition. The first method is really simple, it let the program creates a dictionary with some intents as keys and corresponding regular expressions as values to identify the intent of a statement by searching through the keywords to match the correct intent. The second and third methods use the word vector. The second method uses the word vector to extract the words in each sentence to compare the similarities between the two words to determine the intent of the sentence. The third method uses the Support Vector Machine (SVM) to analyze the word vector features using a classifier and uses the data and corresponding intent to train the Support Vector Classifier (SVC), and then, use the SVC to predict the intent of the target statements. These three methods have their own advantages and disadvantages, but at present the third method has a relatively more accurate intention. I used SVC in this program to extract the intent of the user input statement.

2.3. Named Entity Recognition

The task of Named Entity Recognition (NER) is to locate and classify name entity from unstructured text into predefined categories such as three major classes (entity, time and number) and seven subclasses (name, institution name, place name, time, date, currency, and percentage). The reason of why NER is used to extract the entities from the statement is that the keywords in the statement are an important basis for our judgment of semantics.

Similarly, NER has three different methods. The first method uses spacy to directly extract all the entities in a sentence. The second method is to analyze the relationship between entities through the "ancestor" words in front of the entities in the statement to determine the class of the entities. The third method can also use the regular expression to directly match the desired entity type, but this method has a lot of limitations. It cannot directly extract multiple different types of entities and is prone to errors.

2.4. Database and API

Building a database allows us to easily deal with more different kinds of questions without being limited to a fixed format. By creating a database with corresponding data, we can more easily match all the entities in the user-entered statements with the keywords in the database to get the most appropriate results. The same is because the choice we give to the user after the database is built depends entirely on the size of the data inside the database. SQL is the most commonly used and the simplest database class.

The API is another way to extract data. It is similar to a database, but the source of the data is different. The API searches for data on the network through a network link, and the database is built locally according to the program needs. For this project, we mainly use API to get data, because the main function of our dialogue robot is to provide a query for a certain kind of thing, and it is often not comprehensive enough to build a database by ourselves.

2.5. Multiple rounds of dialogue

Because of the database, we can implement more chat conversation functions, such as multiple rounds of dialogue, negation, etc. The incremental slot filling and negation method is based on the existing database to finally find the best-fit answer for the user in the database by accepting the user's multiple input requirements to continuously update the restrictions.

In human-computer interaction, the negation semantics are very difficult to extract. At present, we decide whether an entity is negative or not only by checking whether there is "not" or "n't" before the entity. However, many words that negate the meaning, such as hate, cannot be identified as negative, and cannot be judged as negative if the negative word is no longer an entity. Multiple rounds of dialogue involving affirmation and negation can more accurately identify the result for the same intent.

2.6. State Machine

Through the state machine, we can easily distinguish the different responses of chatbot to the same sentence in different states. At the same time, we can also determine whether the status is currently changed by the user's input. Through a variety of different state transitions, chatbot can respond to users' feedback after changing their intent in the Duolun dialogue.

3. Conclusion

Before signing up for this research, I couldn't imagine that I could make a simple chat bot and provide a certain level of search. I am very grateful to our mentor, Fan Zhang, for guiding us this month.