

|  |  |  |  |
| --- | --- | --- | --- |
| Course Code | : | SOF106 | |
| Course Name | : | Principles of Artificial Intelligence (Group 3) | |
| Lecturer | : | Shamini Raja Kumaran | |
| Academic Session | : | 2021/04 | |
| Assessment Title | : | Project(Final Assessment) | |
| Submission Due Date | : | 02/07/2021 | |
| Prepared by |  |  | |
| : | Student ID | Student Name |
|  | AIT2009357 | Bi Xiaoyang |
|  | AIT2009376 | Wang Qipeng |
|  | AIT2009380 | Zhang Yichi |
|  | AIT2009364 | Hu Yaqi |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Date Received | : | 01/07/2021 | |
|  |  |  | |
| Feedback from Lecturer:  Mark: | | | |

**Intelligent Companion**: a rule-based system using advanced self-learning algorithm

**Abstract**:

Chatbots are intelligent agents with which users can engage in conversations using natural language, usually via text or voice. After decades of evolution, highly developed chatbots like Siri, Cortana, and Alexa have left people surprised with their intelligence and capabilities. However, these super-smart chatbot requires significant experience and expertise, However, their perfection relies on the long-term collaboration and refinement of a large number of professionals, hard to be replicated. Therefore, in this paper, our group focused on designing a chatbot that can have daily conversations with us and make itself our chat partner under the COVID-19 situation. We named it Intelligent Companion.

Intelligent Companion is a program written in python and based on the application of artificial intelligence knowledge such as Natural Language Processing and Neural Network. Compared to the earliest chatbots such as ELIZA, our Intelligent Companion uses a similar rule-based system and a more advanced self-learning algorithm. Improved on the restricted conversation content and the inability in generating meaningful natural language conversations, it can generate richer, meaningful, and special conversational responses.

Keywords: Chatbot, Neural Network, Python, Natural Language Processing (NLP), daily conversation

Table of Contents

[SOF106 1](#_Toc7120)

[1. Introduction 4](#_Toc29175)

[2. Methodology 5](#_Toc18972)

[2.1 Research Approach 5](#_Toc5966)

[2.1.1 Neural Network(删掉了但不能完全删掉，再看看) 7](#_Toc16007)

[2.2 Research Process 7](#_Toc11613)

[2.2.1 Dataset 7](#_Toc16519)

[2.2.2 Encoding 8](#_Toc119)

[2.2.3 Training details 8](#_Toc19212)

[3. Experiment 8](#_Toc10252)

[3.1 Dataset details 8](#_Toc1797)

[3.2 Training details 8](#_Toc17312)

[3.3 Running configuration 9](#_Toc3816)

[3.4 Experiment results 9](#_Toc16566)

[Figure1 9](#_Toc27533)

[Figure2 9](#_Toc30902)

[Figure3 10](#_Toc31971)

[Figure4 10](#_Toc12117)

[Figure5 11](#_Toc21828)

[4. Conclusion 11](#_Toc32020)

[5. Reference 11](#_Toc1142)

1. **Introduction**

In recent years, chatbots have been empowered to engage in social conversations with humans and have the potential to elicit people to disclose their personal experiences, opinions, and emotions. Nowadays, the chatbots we commonly use in daily lives mainly perform two functions: daily conversation and client service, and thus we classify chatbots as Social Chatbot and Task-oriented Chatbot. Advanced chatbots such as Siri and Cortana have been put into use in people's daily life, the reason that they have become important helpers in our office and life is their success in combining these two functions. However, their perfection relies on the long-term collaboration and refinement of a large number of professionals, far beyond our capabilities. So we aim at building a virtual companion that can handle the easy everyday conversation with users and giving it a human tone to the best of our ability. In this work, our team designed a chatbot named Intelligent Companion with the basic knowledge of Neural Network, Natural Language Processing, as well as Python programming.

The key capabilities that Intelligent Companion has implemented is selecting the closest matched response by searching in the range of all known statements that are related to the input, it then chooses a response from the selection of known responses to that statement. In addition, we have equipped the chatbot with some mathematical knowledge, so that it can respond to some mathematical questions.

In the learning of the earliest chatbots such as ELIZA, we recognize the application of a Rule-based system in chatbot: conversations are processed by predefined rules (keywords, if-else, machine learning methods, etc.), and then the corresponding actions are executed to generate replies. With the same approach, a set of ground rules are established and Intelligent Companion can only operate on these rules in a constrained manner. As for applying the self-learning algorithm, on the other hand, Neural networks are used to train the chatbots to reply to a user, based on the training set of interaction. Rule-based approach and self-learned approach are applied to the task part and interactions respectively, According to our research, this combined approach is found to have the promising capability to generate daily conversation.

1. **Methodology**

During the selection phase, we set out to learn about different types of chatbots, and due to our limited capabilities, we chose a rule-based system, a relatively more simple one, to our underlying principles and improved on it to try to give it the ability to understand natural language and generate richer responses. In this section, we will elaborate on our project preparation and coding process.

**2.1 Research Approach**

Our purpose is to build a chatbot that can make them respond to our questions and generate conversation. The two main challenges are understanding natural language and generating meaningful responses. As Python was chosen to be our programming language, the Natural Language Toolkit (NLTK), is one of our basic methods to realize our goal. In addition, the basic neural network was also applied to reach the same. In this section, we will list the knowledge based on which we were able to start the building process so that we can illustrate the approach of this research.

the Natural Language Toolkit (NLTK)

NLTK, the Natural Language Toolkit, is a suite of Python modules providing many NLP data types, processing tasks, corpus samples, and readers, together with animated algorithms, tutorials, and problem sets (Loper and Bird, 2002). ( NLTK: The Natural Language Toolkit Steven Bird)

NLTK, in this project, was used to realize 5 functions:

**Morphological and Lexical Analysis**

**Syntactic Analysis**

**Semantic Analysis**

**Discourse Integration**

**Pragmatic Analysis**

**Morphological and Lexical Analysis:** Lexical analysis depicts analyzing, identifying, and description of the structure of words. It includes dividing a text into paragraphs, words, and sentences.

**Syntactic analysis:** The words are commonly accepted as being the smallest units of syntax. The syntax refers to the principles and rules that govern the sentence structure of any individual language. Syntax focus on the proper ordering of words which can affect their meaning.

**Semantic Analysis:** This component transfers linear sequences of words into structures. It shows how the words are associated with each other. Semantics focuses only on the literal meaning of words, phrases, and sentences.

**Discourse Integration:** It means a sense of the context. The meaning of any single sentence depends upon those sentences. It also considers the meaning of the following sentence.

**Pragmatic Analysis:** Pragmatic Analysis deals with the overall communicative and social content and its effect on interpretation. It means abstracting or deriving the meaningful use of language in situations.

After elaborating on the above five purposes we wanted to realize by NLTK，the methods or functions used to implement these five components are listed as follows.

**Tokenization:** Tokenization is the process by which a big quantity of text is divided into smaller parts called tokens. It takes in a sentence and decomposes it into the smallest extractable units or words.

**Speech Tagging:** Various parts of speech like a verb, noun, adjective often assume different components in the sentence structure of natural language. Therefore, if can tag to which parts of speech do a word belongs, it will be easy for the chatbot to understand the context of a sentence.

**Lemmatization:** Using lemmatization, the words which have the same meaning but have some variation according to the context or sentence are brought down to their root word. For pattern matching and rule-based approaches, this is an indispensable step.

Since all these facilities are provided by python's NLTK libraries, We can tentatively conclude that our research is feasible.

### Neural Network

Neural network-based dialogue systems have two advantages over the traditional four approaches.

1. Converting sentences into word vectors and then generating semantic matching vectors. By decomposing the semantic matching vectors, the similar and dissimilar components of the sentences can be found and parsed separately, making the comparison of sentences with a higher The decomposition of the semantic matching vectors can find out the similar and dissimilar components of the sentences and parse them separately, which makes the comparison of sentences more accurate.

2.The neural network algorithm is used to integrate the similar and dissimilar components to obtain the feature vectors, which can extract the features in the sentences better than the traditional methods and has strong robustness and powerful self-learning ability, especially suitable for solving such problems with complex internal mechanisms.

## Research Process

After a detailed introduction to the principles and techniques we used, in this section, we will elucidate the general procedure of our experiment.

### Dataset

We created a JSON file named intents, containing corresponding answers to the tags and patterns. There are 56 tags, representing various topics used in daily conversation. And under each tag, we have patterns and responses, which correspond to the Q & A in daily conversation. This intent.json file is the dataset we used for training, and according to our experiment, the more topic we set in the file, the longer training time it takes. These 56 tags covered all the topics our teammates were interested in in our daily conversations.

### Encoding

We set up 3 files and got 3 new ones generated in the training:

**intent.json**: It is the training data as we mentioned above

**training\_chatbot.py**: This is the file in which can we start the training.

**main.py**: In main.py, we defined 3 main functions: clean\_up\_sentence(), bag\_of\_words() and predict\_class(). Based on the NLTK facilities we have mentioned in 2.1.1, they enabled the chatbot to deconstruct sentences into words to understand the meaning and generate answers that are logical in natural language.

**chatbot\_model.h5, words.pki, classes.pkl**: These are the 3 output when we run the training\_chatbot.py, without which our main.py cannot run.

### Training details

# Experiment

We applied our asking to our well-trained Intelligent chatbot. Then we can see the

response from the chatbot. We can also see the accuracy and loss after running the

training file, which they are visualized and represented in the python console.

**3.1 Dataset details**

We created a JSON file called intents.json, which is used in training our models. We

separate them into 56 tags, each tag represents a topic when we chat with this chatbot. And under each tag, we have "patterns" and a "responds", which on behalf of Q&A. These 56 Q&A topics are some conversations we think it is common in our daily life.

**3.2 Training details**

Neural network is easy for training our models. And to build our neural network, we use TensorFlow to reach it. we create a simple sequential model and add a couple of layers. The first layer is the input layer, which is a dense layer with 128 neurons and an input shape that is dependent on the size of the training data for x. And we specify the activation function "relu". We again add a dense layer with 64 neurons and a Dropout layer as output layer then our drop out as 0.5. And we equal the activation function to softmax because this is the function that sums up or scales the results. Last, we set the learning rate as 0.01, a decay of 1e minus 6, a momentum of 0.9, and set 200 epochs for training.

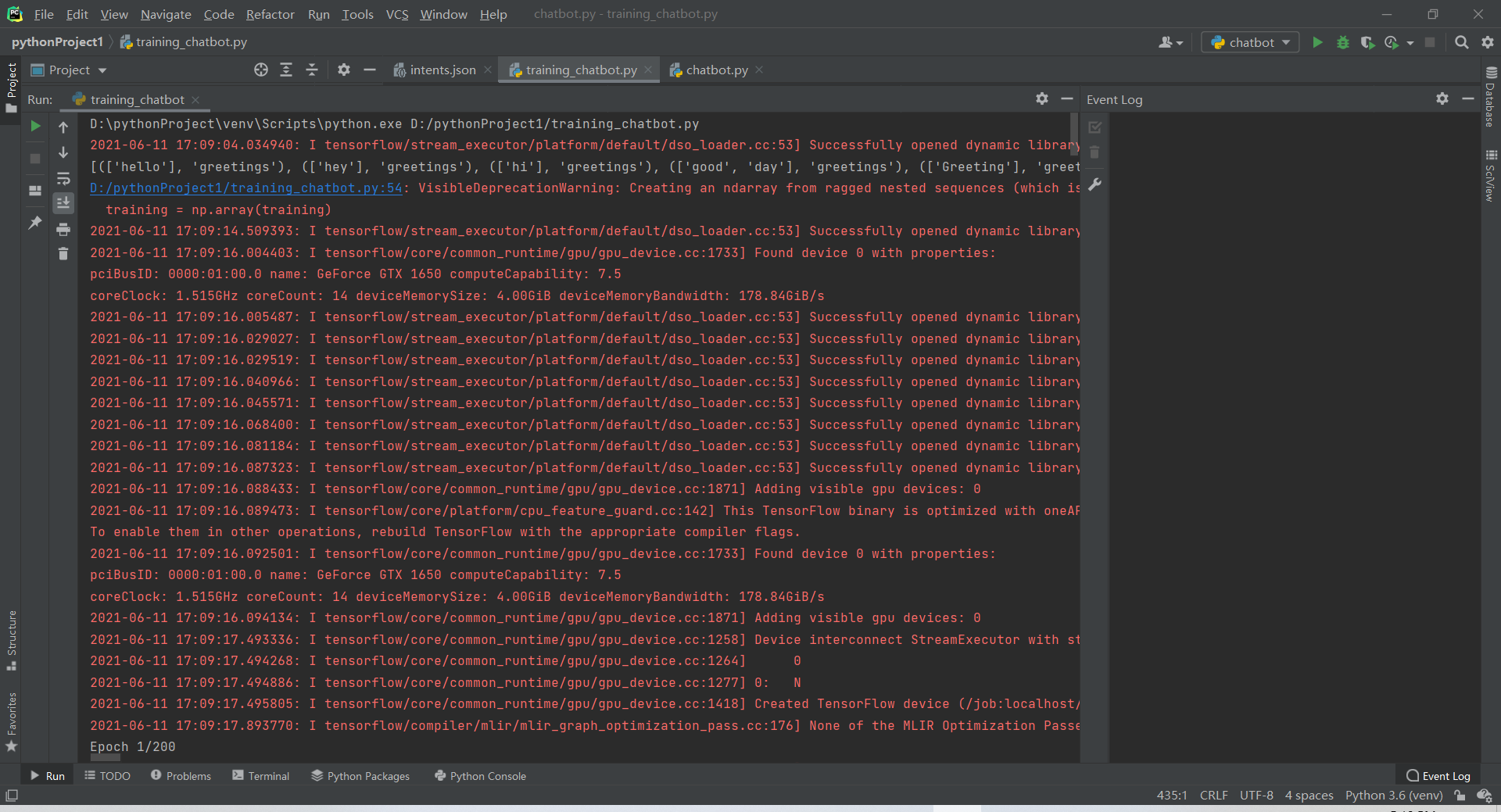
**3.3 Running configuration**

For IDE, we choose Pycharm 2021.1 to edit it. And we also download some packages such as nltk, numpy, TensorFlow, and some modules from TensorFlow and python

standard library.

**3.4 Experiment results**

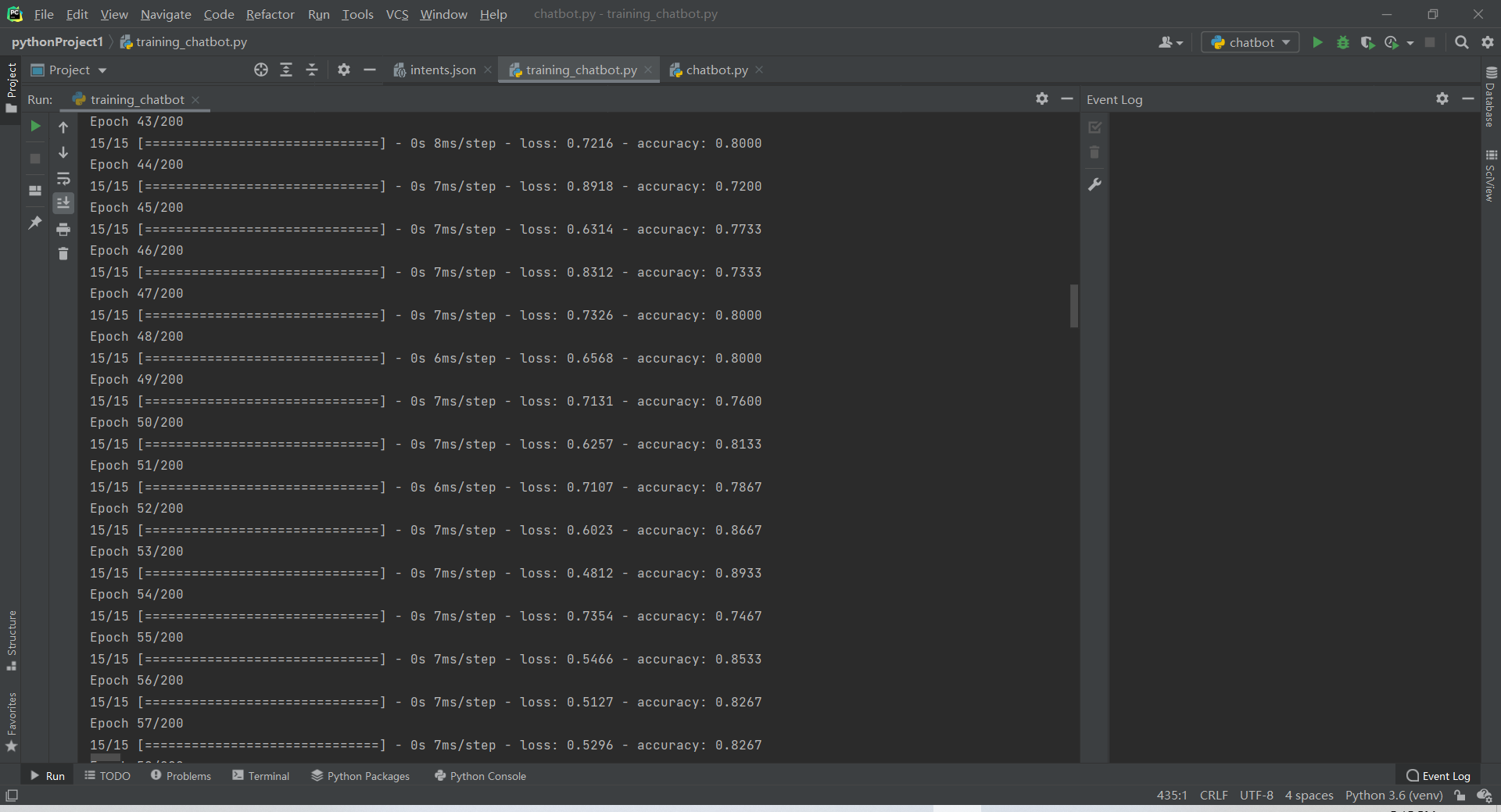
**Figure1**



In the Image1, we execute the training file and start to train.And it shows Tensorflow

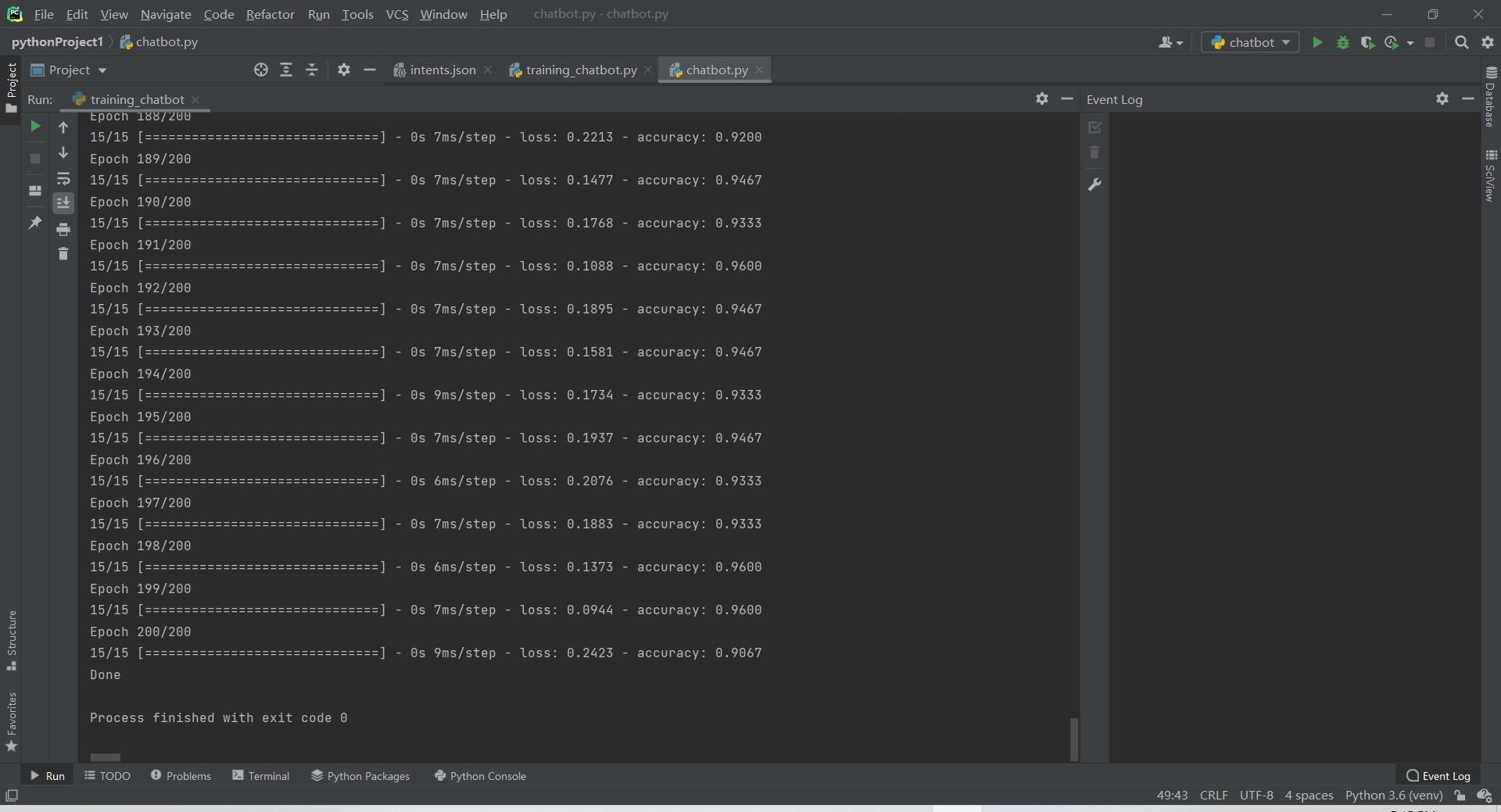
as been opened successfully

**Figure2**



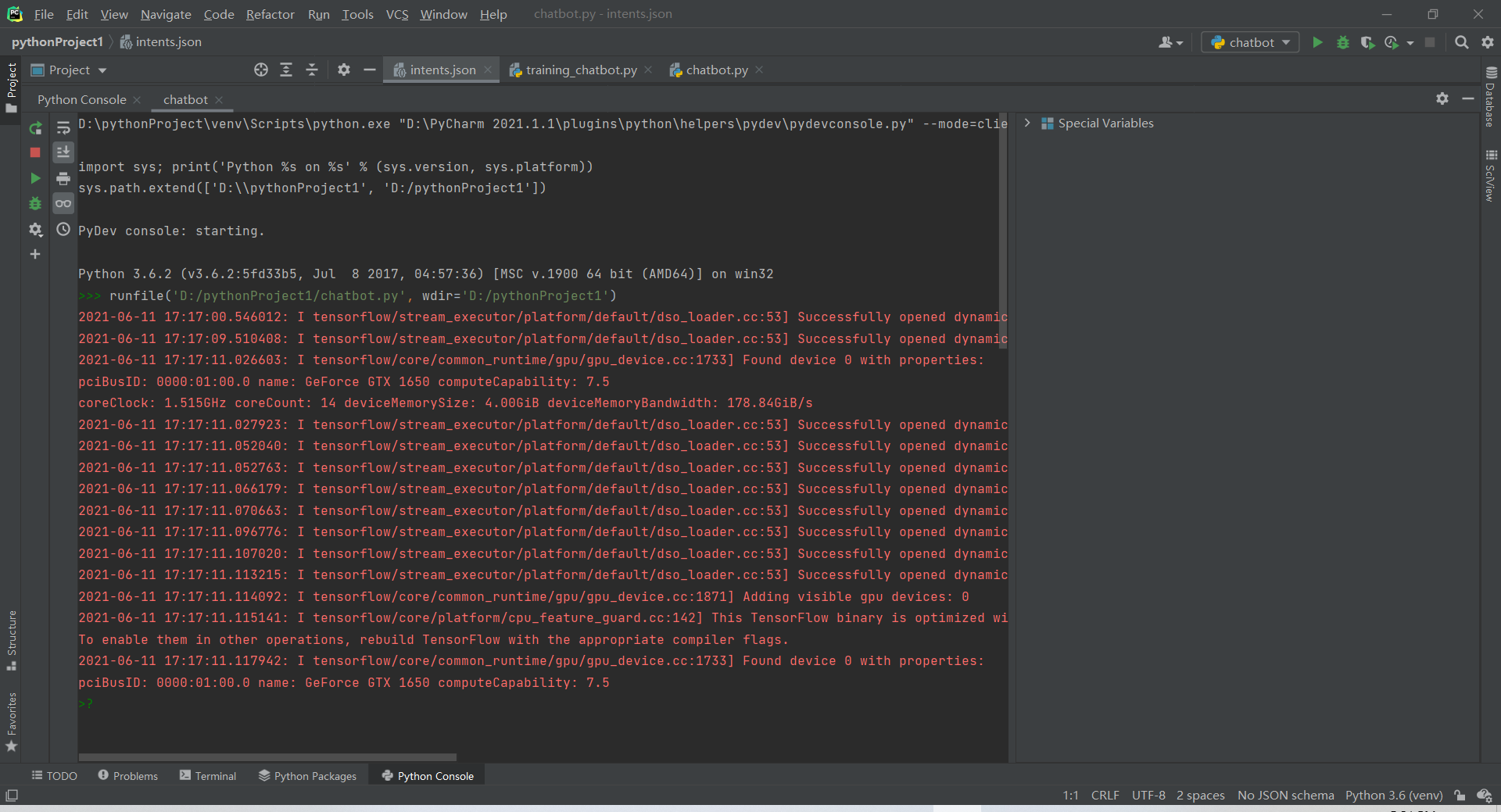
In the image2, we can see the data is training and shows the accuracy and loss of each epoch’s training.

**Figure3**



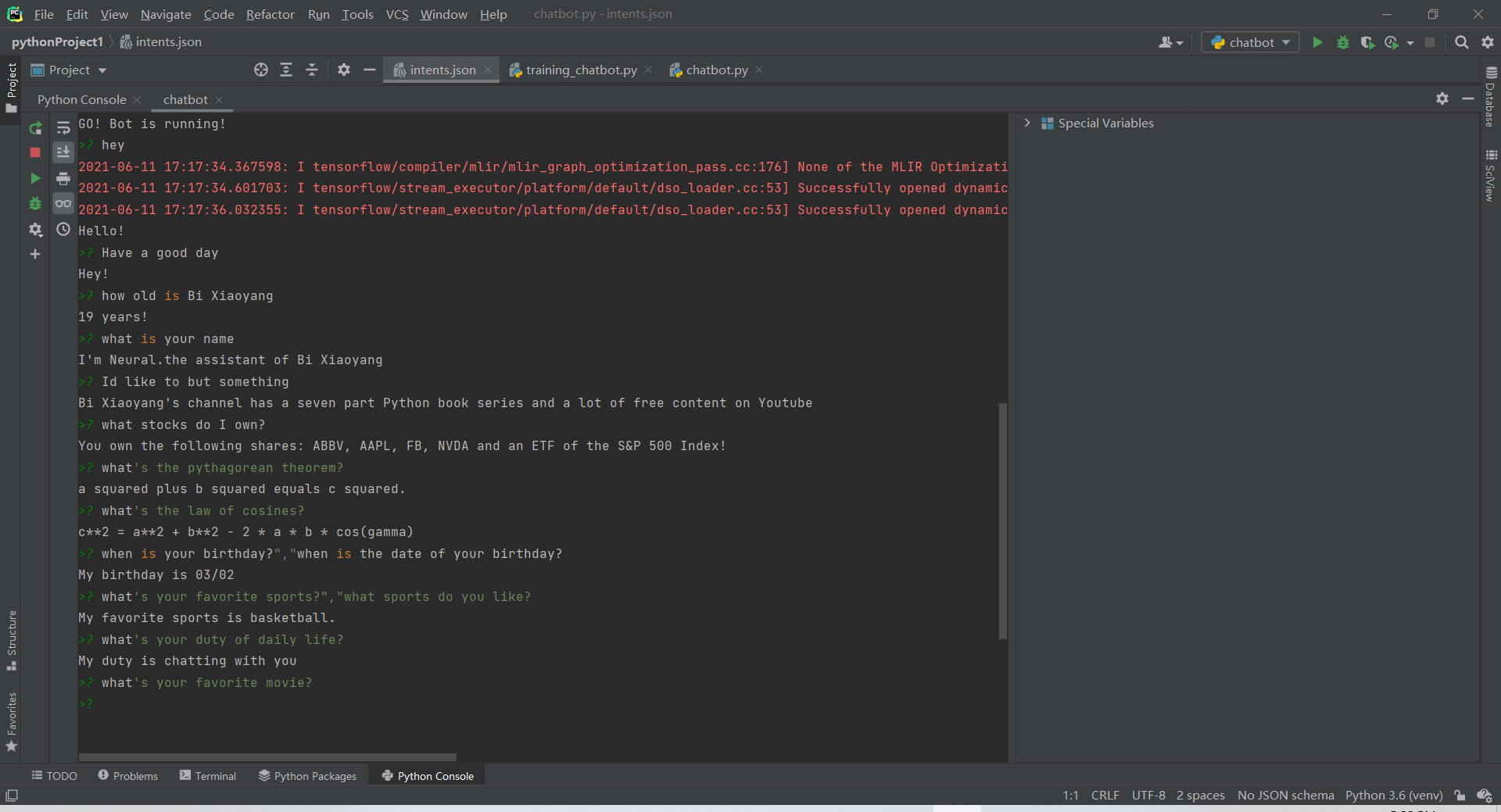
After 200 epochs training ,when we see the hint word “Done!”, then we know the

training work is done and our training model is built.

**Figure4**

In this image, we can see the chatbot is executing, and when we see a green sign “>?” it hints us we can start to chat with our chatbot.

**Figure5**



When we ask the question in the intents.json, then it will make a respond.

# Conclusion

Chatbot, one of the most promising areas of **AI** today, has been highly developed, but there are still many problems that are not solved well enough, such as the ability to link contexts and complete a real conversation as people do. This study, as a rather shallow attempt, clearly does not give a feasible direction for these remaining problems. At the same time, **Intelligent Companion** has even more limitations: since the underlying code is derived from a rule-based system, the more complex the system is, the more rules there are, and thus it cannot be improved to a higher level. But overall, we have made our effort to design this Intelligent Companion under the COVID-19 situation, which is the first step that we have taken in our academic career to researching chatbot.

# Reference

Joseph Weizenbaum. 1966. “ELIZA a computer program for the study of natural language communication between man and machine.” Communications of the ACM, 9(1):36–45.

Christopher D Manning, Christopher D Manning, and Hinrich Schutze. 1999. ¨ Foundations of statistical natural language processing. MIT press.

Maali Mnasri Opla / Clermont-Ferrand, 2019, “ France Recent advances in conversational NLP : Towards the standardization of Chatbot building”

Ilya Sutskever, Oriol Vinyals, Quoc V. Le,2014, “Sequence to Sequence Learning with Neural Networks”