

Project: Eliza based chatbot

Course: Artificial Intelligence and Natural Language Processing

Author: Adam Malinowski ID/65425

1. Introduction
2. Description of techniques used
3. Performance and evaluation
4. Summary

1. Introduction

I created Eliza based chatbot. Unlike Eliza it is not a therapist, but a friend. This person is somebody well known by the user, as we can see from the responses of bot during the chat. This characteristics was made in order not to make the conversation boring and so unnatural.

PyCharm IDE was used to create the code.

2. Description of techniques used

The techniques I used:

- Non-trivial use of Regular Expressions - code contains many regular expressions which are used to discard all the words which are less important for the response choosing. Before each block of possible responses, there is a regular expression which determines the words, or the structure of the user's input in order to classify it to the appropriate responses block. Then the response of bot from this block is picked and given as output to the user. The responses are randomly chosen from the block so there is smaller probability of getting the same automatic response after writing the same sentence.
- Classification - is made by assigning the user's input to the specific response block. Blocks are dependent completely on the input so the content of it is checked using regular expressions every time user writes something.
- Natural language understanding - chatbot mimics natural language understanding. It is strongly dependent on the input but generally it is able to determine and classify what was written by using hand-coded rules. The mechanism for the change of the forms of the words representing the subject, reduces the possible grammar mistakes made by bot. For example if it finds "I" in the input, then in the response it generates the appropriate form "You" when talking about the user.

- I also used the assumption from the Naive-Bayes technique which says that the position of the words does not matter. The input of the user is checked sometimes using this assumption, because it just looks if there is a specific word somewhere in the input, which would straightaway tell us what topic the user suggests without the need to check the whole sentence.

3. Performance and evaluation

The performance is strongly dependent on the user's input which cannot be accurately predicted. Thus the solutions I used will perform differently each time. However we can state that most of the topics which are present in the conversation of two people who are friends, are included in the code so the conversation will seem less unnatural. Thanks to that coverage, the code performs better than in the more general case. Providing the context for the user generally brings better results, because the user knows that he is expected to play a role while chatting with a bot, in this case it is a role of a close friend. It also helped to reduce the number of possible topics and the need to both analyze the input thoroughly and the need to generate very complex output.

4. Summary

Miracle was not expected when creating a bot based on Eliza, but I think that I created something which performs better than Eliza and reduces the impression that we are chatting with a bot in a better way than Eliza. Responses are less automatic and less repetitive. The code's main advantage is that it is easily expandable so we can always write more responses blocks for other topics without the need to change something else in the code. Thus it can be expanded to a version where rules and regular expressions will cover so many topics that the impression of chatting with a bot will be very small.