**Final project MIPT-2018**

Zheglov Dmitry

[demonzheg@gmail.com](mailto:demonzheg@gmail.com)

**Abstract**

In this paper I’d like to tell you about constructing telegram stackoverflow assistent based on intent classifier and chatterbot model for dialogue system. This chatbot will give you link to stackoverflow connected with your question also it will talk with you about anything.

**1 Introduction**

We will construct a dialogue chat bot which will be able to:

1) answer programming-related questions (using StackOverflow dataset[1]);

2) chit-chat and simulate dialogue on all non programming-related questions

For a chit-chat mode we will use a pre-trained neural network engine available from [ChatterBot](https://github.com/gunthercox/ChatterBot).[2] Also we will try to train own models for chit-chat.

For those questions, that have programming-related intent, we will proceed as follow predict programming language (only one tag per question allowed here) and rank candidates within the tag using embeddings. For the ranking part, we will need:

word\_embeddings.tsv — word embeddings, that we will train with StarSpace.

We should obtain the following new objects that we will then use in the running bot:

1) intent\_recognizer — intent recognition model;

2) tag\_classifier — programming language classification model;

3) tfidf\_vectorizer— vectorizer used during training;

4) thread\_embeddings\_by\_tags — folder with thread embeddings, arranged by tags.

## **1 Intent and language recognition**

We want to write a bot, which will not only answer programming-related questions, but also will be able to maintain a dialogue. We would also like to detect the intent of the user from the question. So the first thing we need to do is to distinguish programming-related questions from general ones.

It would also be good to predict which programming language a particular question referees to. By doing so, we will speed up question search by a factor of the number of languages (10 here), and exercise our text classification skill a bit.

**1.1** **Intent recognition**

We will do a binary classification on TF-IDF representations of texts. Labels will be either dialogue for general questions or stackoverflow for programming-related questions. First, prepare the data for this task:

1) concatenate dialogue and stackoverflow examples into one sample

2) split it into train and test in proportion 9:1, use random\_state=0 for reproducibility

Train size = 360000, test size = 40000

3) transform it into TF-IDF features

We will train the intent recognizer using LogisticRegression on the train set with the following parameters: penalty='l2', C=10, random\_state=0.

We have test accuracy is equal to 0.99

### **1.2 Programming language classification**

We will train one more classifier for the programming-related questions. It will predict exactly one tag (programming language) and will be also based on Logistic Regression with TF-IDF features.

We will train the tag classifier using OneVsRestClassifier wrapper over LogisticRegression. Use the following parameters: penalty='l2', C=5, random\_state=0.

We have test accuracy = 0.78

## **2 Ranking questions with embeddings**

To find a relevant answer (a thread from StackOverflow) on a question we will use vector representations to calculate similarity between the question and existing threads.

We will use tf-idf weighted mean over words from sentence.

However, it would be costly to compute such a representation for all possible answers in online mode of the bot (e.g. when bot is running and answering questions from many users). This is the reason why we will create a database with pre-computed representations. These representations will be arranged by non-overlaping tags (programming languages), so that the search of the answer can be performed only within one tag each time. This will make our bot even more efficient and allow not to store all the database in RAM.

Let’s train StarSpace[3] embeddings model based on Stack Overflow[1] posts in supervised mode on the same corpus that is used in search. We can account on that these representations will allow us to find closely related answers for a question. It contains more appropriate embeddings.

For each tag we create two data structures, which will serve as online search index:

1) tag\_post\_ids — a list of post\_ids. It will be needed to show the title and link to the thread;

2) tag\_vectors — a matrix where embeddings for each answer are stored.

**3 Chit chat model**

We will train seq2seq model with LuongAttention mechanism and dropout on Cornell movies dataset[4]. We will use symbol level to train our model because it allow us to reply on slang and emotion messages like “gooooood morning”. Encoder is LSTM, Decoder is also LSTM with greedy policy.

General parameters:

embedding\_size = 100, hidden\_size = 500, batch\_size = 60, n\_epochs = 10, learning\_rate = 0.001, dropout\_prob = 0.5, max\_len = 50

We have sequence loss is equal to 1.2.

X is request, Y is true answer, O is output of our NN, C is output from chatterBot.

Some examples:

X: the drugstore

Y: miz cooper i never went to sewin lessons all them

O: the drugstore is the drugstore

C: Which ones?

X: i know you areonly it aint goin to be that wayshe

Y: livin with comanches aint bein alive

O: i dont know what you do to that way

c: i have always thought whoever did it could have done a better job

X: im fine thanks

Y: no thanks honey

O: you dont have to be a little bit of the store

C: It's not your powers of observation I doubt, but merely the paradoxical nature of asking a masked man who is. But tell me, do you like music?

X: ill take care of my end

Y: stay clean

O: you dont know what you were going to do that was

C: I like to laugh as much as the next.

X: what is your name?

O: i dont know what you do to me

C: I don't have any number

Result of chatterBot is better than our model so we will use chatterBot.

**4 Conclusion**

So we have model for intent recognition, model for tag classifier, database for ranking questions and chat model for dialogue system. Bot have run in Amazon and you could use it in telegram (@DeGoRuKa\_bot).

**References**

[1] <https://github.com/hse-aml/natural-language-processing>

[2] <https://github.com/gunthercox/ChatterBot>

[3] <https://github.com/facebookresearch/StarSpace>

[4] <https://github.com/Conchylicultor/DeepQA/raw/master/data/cornell>