Software project:Developing a decision-making assistant

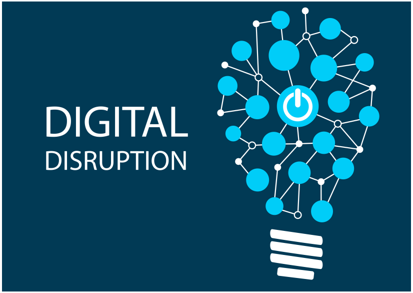
Morozova Milena

July 26, 2022

Table of Contents

Basic Tasks

- Creating a training sample
- Preprocessing training sample
- Building and training the model
- Testing the Model

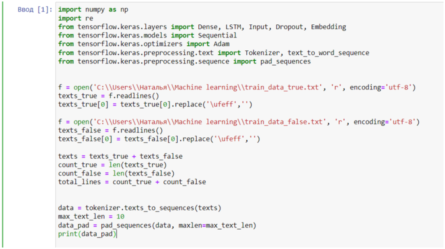


Additional Tasks

- Rating calculation
- Creating GUI
- Testing the App on different platforms
- Setting the application icon

Preprocessing the sample

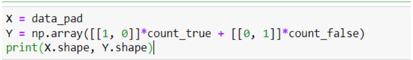
Any text data in its raw material form cannot be analyzed by NLP libraries. This data must be cleaned using various data processing techniques.



```
import numpy as np
import re
from tensorflow.keras.layers import Dense, LSTM, Input, Dropout, Embedding
from tensorflow.keras.models import Sequential
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.text import Tokenizer, text_to_word_sequence
from tensorflow.keras.preprocessing.sequence import pad_sequences


f = open('C:\\Users\\Наталья\\Machine learning\\train_data_true.txt', 'r', encoding='utf-8')
texts_true = f.readlines()
texts_true[0] = texts_true[0].replace('\ufeff','')

f = open('C:\\Users\\Наталья\\Machine learning\\train_data_false.txt', 'r', encoding='utf-8')
texts_false = f.readlines()
texts_false[0] = texts_false[0].replace('\ufeff','')

texts = texts_true + texts_false
count_true = len(texts_true)
count_false = len(texts_false)
total_lines = count_true + count_false


data = tokenizer.texts_to_sequences(texts)
max_text_len = 10
data_pad = pad_sequences(data, maxlen=max_text_len)
print(data_pad)
```

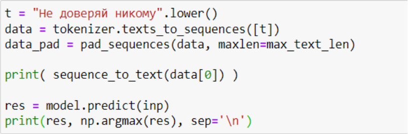
Figure: Preprocessing the training sample

Our neural network will have 2 neurons at the output - the upper neuron will be responsible for the positive text, the lower - for the negative. For positive statements at the output, we will require the vector [1.0], respectively for negative - [0.1].



```
X = data_pad
Y = np.array([[1, 0]]*count_true + [[0, 1]]*count_false)
print(X.shape, Y.shape)
```

Figure: Training



```
t = "Не доверяй никому".lower()
data = tokenizer.texts_to_sequences([t])
data_pad = pad_sequences(data, maxlen=max_text_len)

print( sequence_to_text(data[0]) )

res = model.predict(inp)
print(res, np.argmax(res), sep='\n')
```

Figure: Testing

Figure: Working with a mathematical model

Bibliography

- Machine Learning, Neural and Statistical Classification
  `https://www1.maths.leeds.ac.uk/~charles/statlog/`
- Cleaning Preprocessing Text Data for Sentiment Analysis
  `https://towardsdatascience.com/`
- Generating WordClouds in Python Tutorial
  `https://www.datacamp.com/tutorial/wordcloud-python`
- TripAdvisor
  `https://www.tripadvisor.ru/`
- Sentiment Analysis of Review Datasets
  `https://www.researchgate.net/publication`
- Keras Documentation
  `https://ru-keras.com/recurrent-layers/`