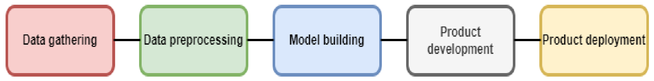
Text classifier - a way from obtaining data to model deployment

Stanislav Zámečník† and Vojtěch Šindlář†

†Department of Mathematics and statistics, Faculty of Natural sciences, Masaryk University University

Text classifier

Our project is web based app - text classifier, based on Neural Network, which enable user to categorize his text into three separate categories, namely: sport, travel and science category. The app is built in Python programming language, with various tools, which we would like to also present as so the whole process of obtaining data, i.e. which tools we used, till deployment model into "production".



Data gathering

Sometimes, when modeling we might miss data. This problem can be due to many reasons: lack of finance, time to obtain them, no available data sets. To solve this problem, we can use the process of web scraping technique. One of very popular scraping tools in Python is package beautifulsoup.

Beautiful Soup

Very easy to use package, where with few lines of code , we are able to gather text, pictures, web-links and possibly everything what we could find on html page. We developed a scraper to get a text for model training from web page `https://www.dailymail.co.uk/home/index.html`.

Data preprocessing

In reality, the longest period in product development. Since in our model, we train NN, we cleaned text from url links, removed blank spaces from text, unknown symbols and put the whole text into lower letters so it's easier for model to learn. In this process we used packages as numpy and pandas, which are popular within data science community.

NumPy

pandas

Model building

Our text-classifier is built upon very popular concept of Neural networks. Since our data are pure text, we used specifically Recurrent neural networks (RNN). The winning model architecture is composed of embedding layer, three LSTM (long-short term memory) layers, and one fully-connected dense layers. We obtained 88% accuracy on testing set with the loss 0.863.

```
def create_model():

    model = tf.keras.Sequential()

    model.add(Embedding(vocab_size,240, input_length = max_length))
    model.add(Bidirectional(keras.layers.LSTM(150, return_sequences=True)))
    model.add(Bidirectional(keras.layers.LSTM(64, return_sequences=True)))
    model.add(Bidirectional(keras.layers.LSTM(32, return_sequences=True)))
    model.add(Flatten())
    model.add(Dense(3,activation = 'softmax'))


    model.compile(loss = 'categorical_crossentropy',optimizer='adam',metrics = ['accuracy'])
    model.summary()

    return model
```

Since you can see from the previous picture, the model is built in tensorflow and keras packages.

TensorFlow

K Keras

Both of these tools are among the most popular for building NN models. They are very easy to use, and from the newest version they incorporate the GPU of computer.

colab

Our model was originally developed in Google Colab - another cloud tool for colaboration. We decided for this platform, because Colab let users use GPU (graphical processing units) and TPU (tensor processing units developed by Google) for performance improvement (although time per project is limited).

Product development 1

We built our web-app with the python package called Streamlit.

Streamlit

It is a tool that allows quickly build highly interactive web applications with fastly growing base. Also strong advantage is, that it is much more "Pythonic" friendly way of building apps. Another popular python packages are Django, Flask and R tool Shiny. Especially first tool are widely used in commercial sphere and they are much more customizable,

Product development 1

The whole process of development was made via Git and GitHub, which is a provider o hosting for software development and version control using Git. If you don't want to r all the Git commands, we strongly recommend you to use one of the git clients - in o was Fork.

Product deployment

To finally put our model into "production" so other users can explore all features of ou we need to deploy it into some server (so far, we could only use it locally on our machi comes Heroku into game.

HEROKU

Heroku is cloud platform which now supports many programming languages. Current host up to five web-app projects for free (here are some limitations about size of app, also deployed our model to this platform, although the limitation about size prevent final stage of our project.

Features of our web-app

Among features of our web-app we include:

- Categorizing copy of any text article.
- Possibility to add .txt file for categorization.
- It's also possible to use combination of copy of text and text from .txt file.
- Use of webscraper to obtain own data from page `https://www.dailymail.co.u index.html`
- Possibility to train your own model.

Source code

Our repositary with whole source code and presentation can be found here: `https:// com/stazam/Datamining2-project`.