ONDEWO

Challenge 022 - Challenge Data Scientist (Daniel Kuddes)

The overall challenge goal is to implement and evaluate model random combinations for a simple text-based machine learning problem based on standard NLP features for a “moped renting chatbot”. The idea come from a recently published paper to randomly combine various models depending on the performance: https://arxiv.org/pdf/1805.01890.pdf

Here the goal is to implement an much easier version of this, by using 3 types of models (CNN, DNN, LSTM), and 1,2,3 or 4 layers per model:

- First use single models (CNN, DNN, LSTM), then combine two (CNN+DNN, CNN+LSTM, CNN+LSTM, DNN+LSTM,,....) and then three (CNN+DNN+LSTM) etc. Of course not only the models
- Variate the number of layers for each combination: 1 layer then 2 then 3 and then 4 layers

This is a “brute force” approach to finding the best combination for the highest prediction. Due to the very small data set this should run also on a laptop computer.

Preparation steps

- Understand code in zip file for loading data, classification and evaluation
- Make code work (watch out for bugs or missing parts) *Note: we want to test how fast you read, understand and fix machine learning programming code which is not in a Jupyter notebook*

Task 1: Create NLP features

Read “Section 2.4 Text / NLP based features” about the standard NLP based features. Implement these in the code and use only these features for training the models
https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guide-to-understand-and-implement-text-classification-in-python

Task 2: Implement, train and evaluate algorithm and layer combinations

The goal of this task is to find the best combination of models and layers for the given data set. Implement the algorithms with Keras, Tensorflow and NLTK.

1. Implement the algorithms for CNN, DNN, LSTM with Keras and Tensorflow 1.10.1
2. Develop an algorithm for building all the combinations based on “brute force” method
3. Develop an “smarter” algorithm for finding the best combination faster for the models and layers
4. Evaluate the performance of the algorithms and algorithm combinations for prediction performance (accuracy, precision and recall) and training time to find best model

Task 3: Make chatbot talk

Based on the best model identified make the chatbot talk based on the existing logic with the best “model”.

Wrap up: Python executable program

Write a python executable program that (i) trains the models, and (ii) evaluates the models with the different techniques. Answer briefly (max. 2-3 sentences) being aware there is very little training data:

- Which method, combination or technique performs best?
- What else would you add or change for increasing prediction performance? Why?
- How could an algorithm look like instead of the “brute force”-based best model combination finding?

Please provide us with a link to a **private GitHub repository** (grant access for andreas.rath@ondewo.com) where we can download the source code for your tasks, a requirement.txt file with all the python libraries used for easy install and the trained models**.** Feel free to also provide screenshots where you think it is helpful. The download link and a short “Readme/How to” you should sent to office@ondewo.com. We will review your code and the results, and decide if you get a final interview.

Let me know if you need anything or have any questions regarding the challenge. Happy to do a short hangout or a call.

Have fun and a great learning experience!