**Assignment 3 – Practical Deep Learning Workshop**

\* this assignment can be done in teams of up to 3 students please state next to each section who was in charge of that work.

In this task we will work with the dataset of the Home depot product search relevance competition that took place some 3 years ago. Please download all data files [here](https://www.kaggle.com/c/home-depot-product-search-relevance/data)

Notice that the test labels are also available [here](https://www.kaggle.com/c/home-depot-product-search-relevance/discussion/20587) to compute the loss and other metrics for your test

Our goal is to state the relevance of a search phrase to a corresponding item, this can help ranking results of a search engine for Home-Depot’s items

We will use this exercise to learn about a new type of architectures “Siamese networks” that is useful when comparing inputs from similar domains (and also for multi modal problems but we will not discuss it here) you can find a reference paper for the text domain (among many other references) [in this paper](http://www.aclweb.org/anthology/P16-1036) or [in this paper](https://www.aclweb.org/anthology/W16-1617.pdf)

**Character level LSTM**

1. in this section we will use character level processing to try and predict search relevance.

**35 pts**

* 1. Preprocess the training and testing data to contain sequences of single characters.

Note that you should do the above for both the search phrase and the item description.

* 1. Construct and train a Siamese network as described in the above liked paper. The input to the network should be the character sequences constructing the search phrase and the item description, and the output should be the predicted search relevance score
  2. As in previous tasks we would like to create either a naïve or model-based benchmark – use count vectorizer on the character sequences you generated on (a) along with any modeling or statistical method to create such a benchmark score – remember, only by comparing your results with a strong benchmark will truly serve your will to asses how good your model is
  3. Use the model you have got, either last layer or any other layer as a feature extractor for other ML model (please select at least two of: xgboost, lightgbm, catboost, random forest, knn, svm)

**word embeddings and word level LSTM**

1. in this section we will use word/character-combination level processing to try and predict search relevance.
   1. Preprocess the data to create tokens of words/character-combinations (e.g. “¾”, “90°” or “#SC” \* I’ll refer those as words onwards)

**35 pts**

* 1. Use either genism word2vec, or any other embedding training method, to create an embedding with all the desired words in our new dictionary.
  2. Construct and train a Siamese network (you may reuse the one from section (1) above), the input to the network should be the words/character-combinations constructing the search phrase and the item description, and the output should be the predicted search relevance score
  3. Again, use the best model you have got, either last layer or any other layer as a feature extractor for other ML model (please select at least two of: xgboost, lightgbm, catboost, random forest, knn, svm)

1. compare the results you got for each of the methods in a table:
   * + naïve benchmark model from subsection 1c,

**15 pts**

* + - LSTM with character level input
    - LSTM with word level input
    - LSTM as feature extractor with character level input (at least two models)
    - LSTM as feature extractor with word level input (at least two models)

The header should look as follows:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Model type | runtime | Train RMSE | Val-RMSE | Test-RMSE | Train MAE | Val-MAE | Test-MAE |
| Character level LSTM | xxx sec. | 0.4 | 0.45 | 0.45 | 0.385 | 0.395 | 0.397 |

1. Write a report that summarizes your research.

Your report should include:

**15 pts**

* A preface that describes the problem and the methods that were tried to solve it, along with the conclusions you derived from the research.
* Examples of the tokenization methods that you used
* Plots of the training process metrics (train-val loss across epochs or training steps) for the best run that you got for each subsections 1b,1c,1d,2c,2c
* The table of results you got in 2e
* Final remarks on what has been learned through the process and what else you think might be interesting to further explore

Make sure you read and comprehend all that is needed prior to starting the work so that you can keep a record of the things you need to include in the reports. It will be best if you use a github or bitbucket repository to track your advancement and a research diary (or a framework such as Neptune.ml or trains.ai as described in previous task) that keeps track on what experiment did you run, what were the selected parameters, and the results that it yielded.

I’m sure you’ll learn a lot, and hope you’ll enjoy it too

Good luck!

Nati