Kaggle: Home Depot Product Search Relevance

Aqeel Labash, Lisa Yankovskaya

University of Tartu, Institute of Computer Science

Introduction

Home Depot is a huge American on-line shop of home improvement and construction products. The task of this Kaggle competition is to develop a model that can correctly predict the relevance of search queries. The given data consists of real customer search terms from Home Depot's website (training and test sets), attributes and description of goods [1].

Recurrent Neural Networks and Long Short-Term Memory

It is a class of artificial neural network. It is renowned for its dependency on previous information (sequences)[2][3]. Which traditional neural networks lack because they treat all inputs and outputs as independent from each other which is not suitable for all cases [3]. Especially when we are working with natural languages where each word depends on series of previous words.

Long Short Term Memory is a Recurrent neural network that introduced memory cell. Memory cell prevents gradient vanishing and explosion by using four gates Shown in the memory cell Figure. Those gates modulate environment-memory cell interaction.

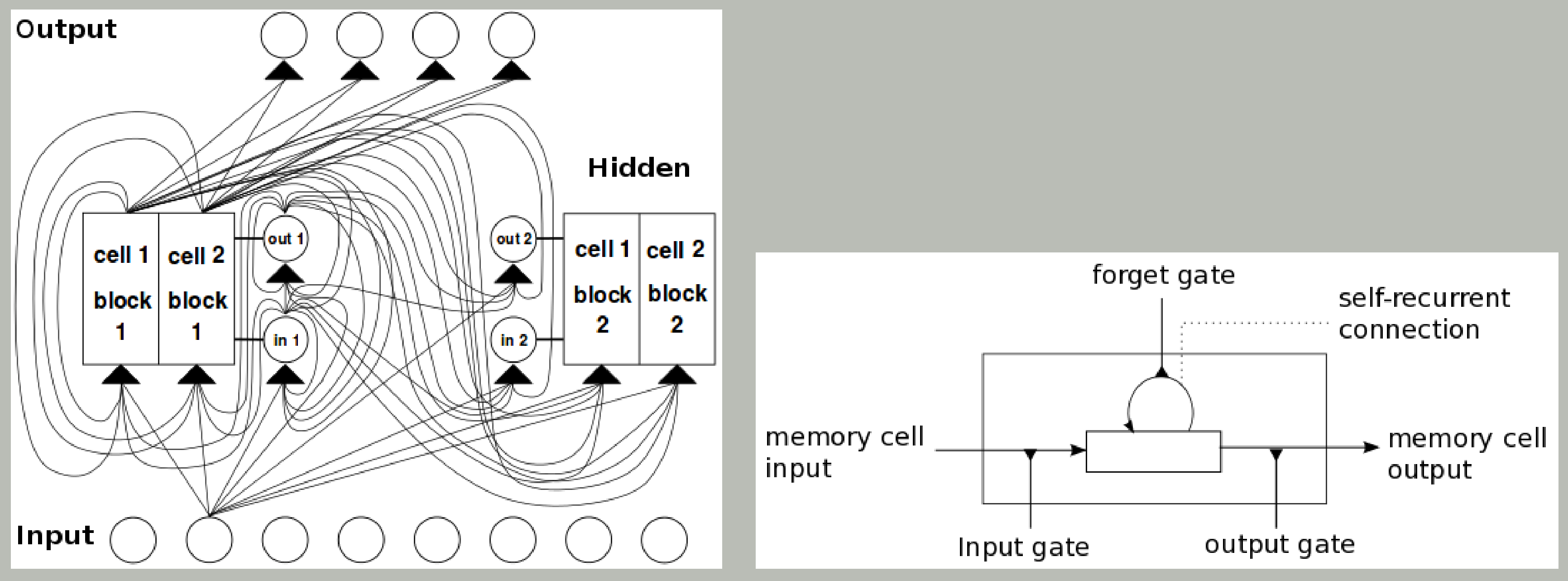
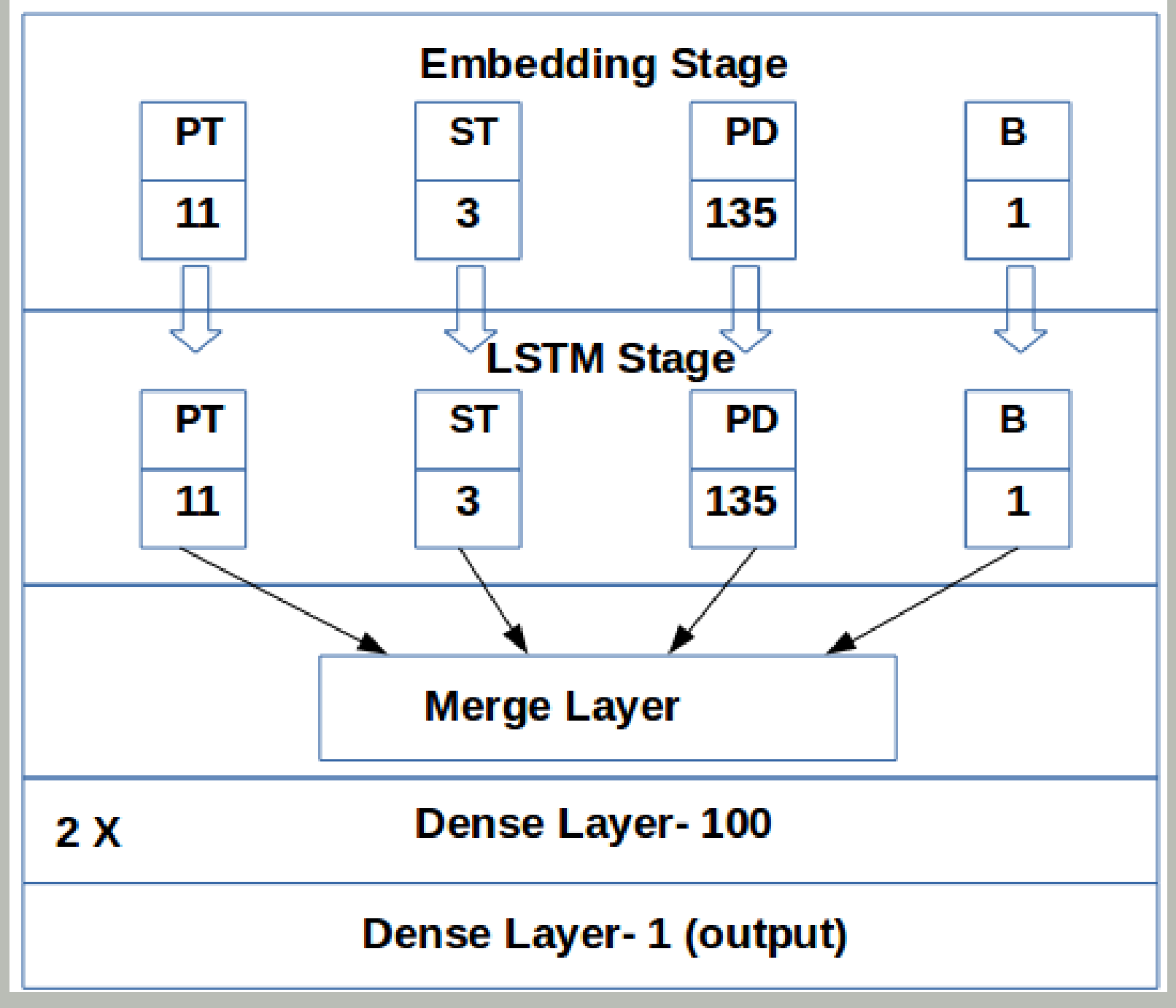


Figure: Example of a net with 2 memory cell blocks of size 2 [4], next to it memory cell [5]

Deep learning model using Keras

The Figure on the right shows the model we used. Embedding Stage contains four Embedding layers one for each group of features. The hidden size for each of them determined by the mean number of words. The same goes for LSTM layer. Two dense layers used sequentially each of them have output dimension of 100. At the end a dense layer with a single output to get the relevance.



Embedding Stage

PT 11 | ST 3 | PD 135 | B 1

LSTM Stage

PT 11 | ST 3 | PD 135 | B 1

Merge Layer

2 X Dense Layer- 100

Dense Layer- 1 (output)

Figure: The model we used for deep learning. PT = Product Title, ST = Search Term, PD = Product Description, B = Brand

Keras: Parameters Optimization and results

To optimize the parameters, we depended on the validation loss. The next two Figures show EPOCH vs training loss and EPOCH vs validation loss.

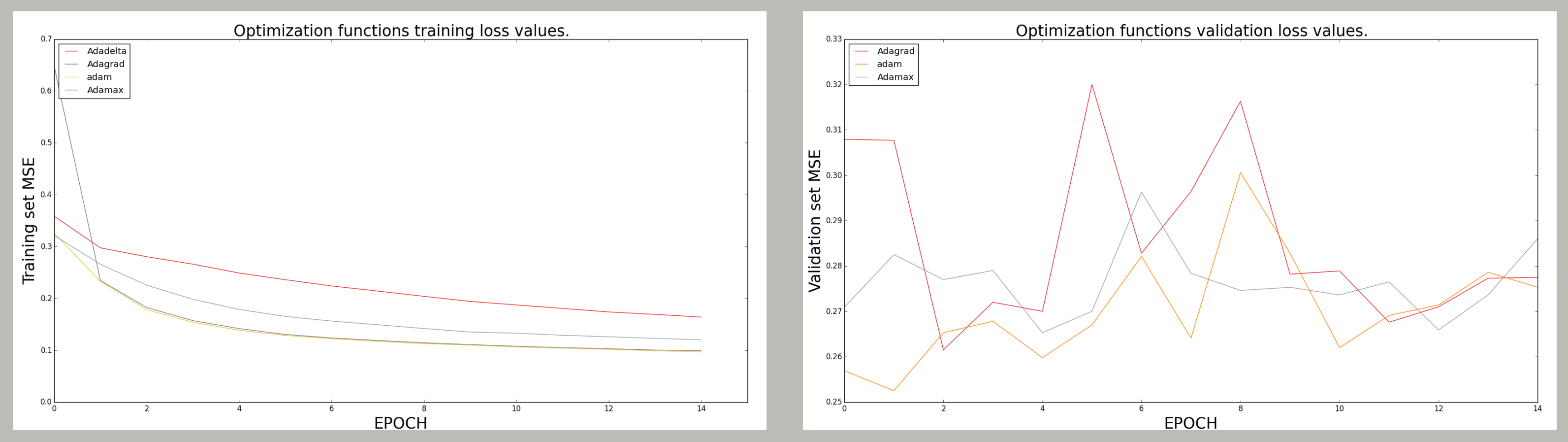
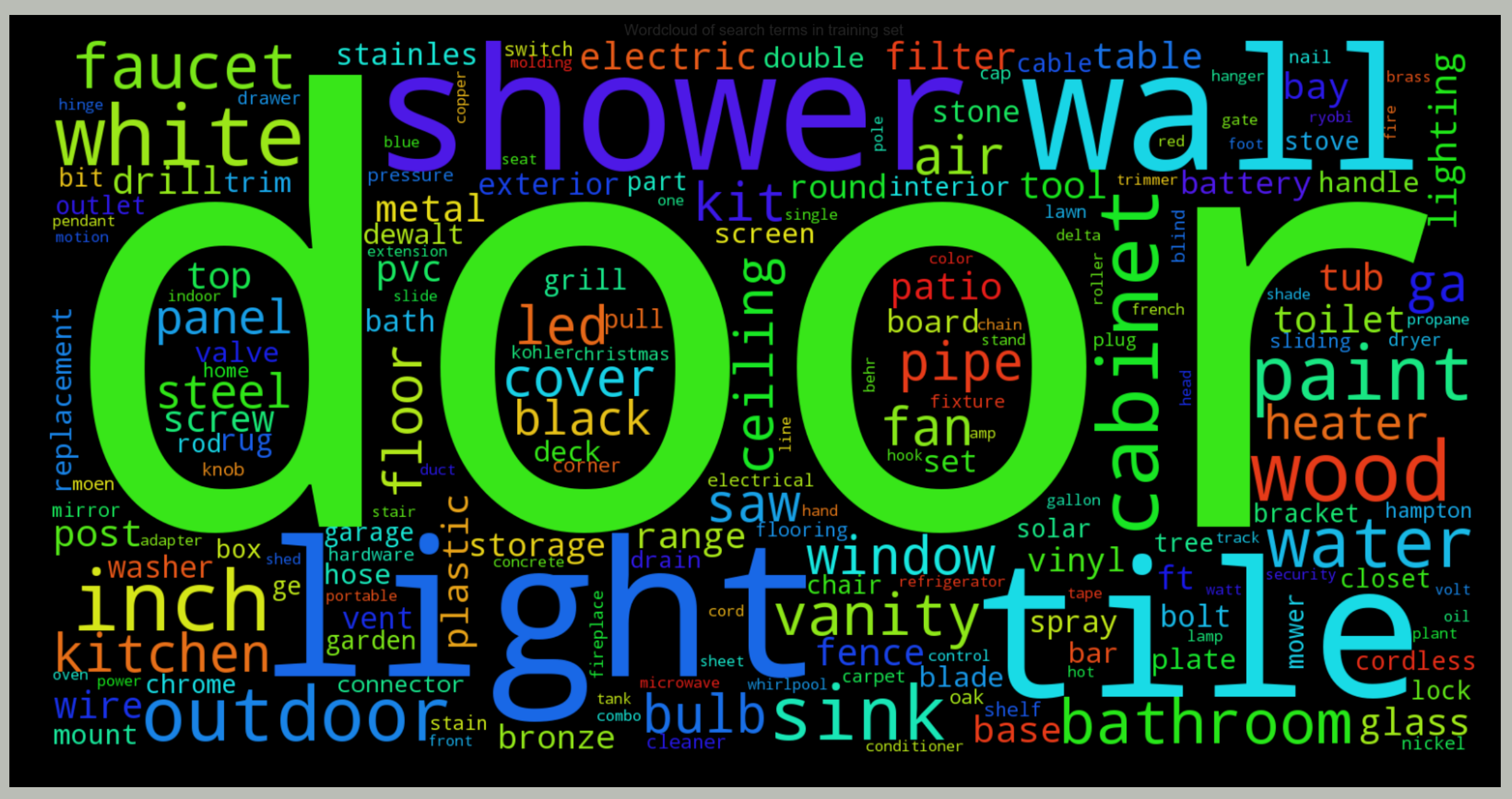


Figure: Left: Training set MSE over many optimization functions,Right: Validation set MSE over many optimization functions

We can notice that the best result is at the begining with adam optimization function where $\boldsymbol{Validation_{RMSE}} = \mathbf{0.5024}$. Using these settings we got $\boldsymbol{Train_{RMSE}} = \mathbf{0.49}$ and $\boldsymbol{Test_{RMSE}} = \mathbf{0.54}$.

Some words about data

At first, we have created word clouds of search items for training and test sets. As well, we have counted words in search queries. The most popular length is three words and about 95**%** of search queries have from one to six words. Also, we discovered that the most popular attributes of goods are brand and material.



Scikit learn: Random Forest and AdaBoost

We have used a library *scikit learn* for Random Forest and AdaBoost. We used the whole training data set for training models and then we predicted the result for the whole test data set.

The preparation steps were cleaning and stemming of data. Then we have chosen different models with different features. Our features: all attributes, brand, material, color, dimensions, Jaccard's distance, and length of the query.

As benchmarks models, mean, random and linear models were chosen.

We fitted both methods for different combinations of features at least three times and calculated RMSE of training and test datasets.

By using AdaBoost method we got almost the same results for all models: min value of $\boldsymbol{RMSE_{test}} = \mathbf{0.5315}$ and max value of $\boldsymbol{RMSE_{test}} = \mathbf{0.5464}$. Also, we got almost the same RMSE for the test set. The main advantage of AdaBoost is the time of calculation, it took about 10 minutes.

By using Random Forest we got the following result for all models: min value of $\boldsymbol{RMSE_{test}} = \mathbf{0.5227}$ and max value of $\boldsymbol{RMSE_{test}} = \mathbf{0.5437}$. RMSE for training set was significantly lower: about 0.179.

So, the best $\boldsymbol{RMSE_{test}} = \mathbf{0.5227}$ was obtained for the model: product title + search term + brand + material + color + length query.

Possible improvements

- enhance number of features;
- distinguish the most important words and add them weight;
- improve text processing;
- another methods

Conclusion

To predict product search relevance we have applied different methods, such as Random Forest, AdaBoost and Long Short-Term Memory. All methods gave the good result and, in our opinion, we should focus on new features and processing text to improve the result.

References

1. https://www.kaggle.com/c/home-depot-product-search-relevance
2. A. Graves, M. Liwicki, S. FernAandez, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for unconstrained handwriting recognition.
3. A. Karpathy. The unreasonable effectiveness of recurrent neural networks. Technical report, May 2015.
4. http://deeplearning.net/tutorial/lstm.html
5. M. Creaney-Stockton. Isolated Word Recognition Using Reduced Connectivity Neural Networks with Non-linear Time Alignment Methods. University of Newcastle upon Tyne, 1996.

lisa.yankovskaya@gmail.com, aqeel.labash@gmail.com