Peer review

16th May 2019

1 Evaluate Existing Word Embeddings Models

1.1 Warm-up

1. **1.1 Load Google News word embeddings**

   OK

2. **1.2 5 Nearest neighbours**

   You found the 5 nearest neighbors to given words correctly using cosine similarity.

   Now, it is not so hard to implement Euclidean distance as well and compare the results.

   It would be more readable to display the results in a table in both Jupyter notebook and in the report.

3. **1.3 SVD**

   It would be nice to just mention how you have implemented SVD (used some package/library or by your own function). The result seems to be correct and you have sufficiently answered how the words are clustered.

   For presenting the plot I would recommend following edits:

   1. Label axes.

   2. Some words are overflowing the boundaries, consider adjusting the graph that everything fits within the frame.

   3. Numbers/text is not readable. Make it more readable and get rid

of unnecessary stuff - e.g. text indicating words. Is it actually needed? It will not be readable at all and we are just curious about relationships between the words, not about exactly where is each word located.

4. You named your graph Figure 1, so it would be nice to reference it in the text where you mentioned it.

1.2 Main Task

It seems that you have gotten correct result for word Iraq and you well explained how you computed it.

When it comes to report, you left almost whole third page empty, which does not look nice.

Also, I would appreciate some comments in code and how you divide logical parts into separate cells.

Altough, I would suggest 2 following improvements.

First, the code where you do your predictions starts to be not readable and is missing some comments. Also, when you look into the two cells they have some things common, so maybe you could make a function which will be then called twice. Also, I needed to read a code to understand what you are actually doing, so maybe comment or a good name for function would solve it. :)

Second, I like to have all imports at one place, usually at the beginning of the notebook not in the cells where there are needed for the first time. When you will have longer notebook and you want to import something you may forgotten if you has not used it before and now you need to check whole notebook for imports. But, it's just a detail and it is up to you.

Lastly, you have showed accuracy score, but maybe you could name the table with small description of the results which you have gotten.

2 Word Embeddings for Text Classification

I miss some explanation why you have done steps which you state in report. For example *"I went at it by only taking the vectors of the embeddings which are actually showing up in the texts"* and *"I went at it by only taking the vectors of the embeddings which are actually showing up in the texts"*.

Then you create vectors for each document by taking the mean over all the words from the document to have a single vector for each document. It would be nice to explain why you actually can do it (do you not loose some information

about the data by this step?) and what it actually solves, again why you need to do this step.

Small typo of missing % sign after your 0.824 accuracy.

The main task of this part is to create and explain in detail model which you have used and how it was trained.

It's nice to see that sometimes Pareto 80/20 rule applies. Even tough only 20% of your work is creating and training model (the rest is loading and preprocessing the data) you should try to answer the question from the task. As you used *AdaBoostClassifier* and had not to adjust it much, there is not so much to describe. Maybe you can explain why you have chosen this particular model and explain your choice of it's initial parameters.

Suggestions for code:

1. Maybe again, you can create a function which loads the data and use it for training and test sets.

2. You load data of one class first and then append the data of the other class. Is it a good practise to have first "half" of the data set of one class and the other of second class?

3. You have done some data preprocessing, which you have not mentioned in report at all. You could mention why is it needed and describe the steps you have done.

(again, you can create a function and do not repeat yourself for train and test data)

4. I would move imports to the beginning of the notebook.

3 Extend Word Embeddings Models to the State of the Art

It would be nice to mention that you used Reuters data set. Again, you are doing twice the same for training and testing sets, consider writing one more general function and call it to load both data sets.

It seems that you read the paper and got familiar with the task. You implememnted TF-IDF baseline which is implemented correctly. You correctly implemented KNN method, but be aware that the best k parameter should be found with the experimental cross validation from {1,...,19}.

Next task is to implement in papers described CPTW method and compare the results with TF-IDF.

However, the results should not be calculated by just simply using f1 micro/macro scores, but we should implement experimental evaluation as described in the paper and present the results for both CPTW and TF-IDF.

4 General notes

You have done work for each task. First two tasks are mostly done, they just need small improvements. There are still some parts of task 3 left to be implemented, but you have good start with baseline.

In overall, all the parts which you have done are correct and being enough explained in the report.

In report, there are few typos, some sentences could be rewritten to be more formal. In some parts, which I have mentioned above I am missing more details and explanations of your actions which can help reader to better understand your solution and you can better demonstrate your knowledge of your solutions.

Except of this everything is clear.

There is still a lot of time and little less space for improvements.

Good luck :)