Information Retrieval

Silvan Robert Adrian
zlp432

June 12, 2019

1 Assignment 1

1.1 5 nearest neighbors

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

  + Euclidean distance calculations added in the report and the jupyter notebook

- 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.

  + Added mention of numpy svd

- 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.

  + Axes labeled
  + Changed it that there is no overflow anymore
  + Substituted the words by points and only added a few example words to see how they are clustered
  − Figure 1 is the caption doesn't necessarily need to be referenced rather just an explanation what is on the image

1.2 Main Task

- 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. :)

  + Code refactored, improved readability by reusing code and reducing duplicated code.

- 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 for- gotten 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.

  + moved the imports up to have a clear overview what modules are used

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

  + Added a small explanation to the table

1.3 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".

  + Added an text explanation

- 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

  + Explained disadvantages of it

- Small typo of missing % sign after your 0.824 accuracy.

  + Wouldn't really say that's a typo but added 82.4 as accuracy now with percent sign

- 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.
    + Described AdaBoost Classifier in short and the parameter I pass (n_estimators) which is pretty much the most important one
- Maybe again, you can create a function which loads the data and use it for training and test sets.
    - Have done so
- 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?
    + Added a shuffle to bring some randomization in to it
- 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.
    + Now wrote a short text about the text preprocessing I did
- I would move imports to the beginning of the notebook.
    + Have done so

1.4 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.
    + Have done so and created a own data reading function
- 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.
    + Now implemented the right kNN method
- 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.
    + Implemented the cross-validation and 5 fold splitting

2 Assignment 2

2.1 Abstract

- Use abstract to create an abstract in LaTeX
  - Already do so, but the IEEEtran document class might not be the best choice for it.

2.2 Introduction

- You introduced ad-hoc retrieval but don't mention the difficulties of the task.
  + Added more introduction and hopefully enough information
- *Missing/not detailed enough* What is the purpose of using text embedding for this task?
  + Added more info, hopefully enough to male a reader understand the report.
- Very short
  + Extended the introduction with more background information and other things.

2.3 BM25

- Using left(and right) will increase size of parantheses in LATEX
  + Thanks, used them now
- You mention relevance information r_i but it is not part of your equation?
  + Yes that slipped into it even though it shouldn't be there
- I don't believe there is a $k2$ in BM25? Where did you find this BM25 equation?
  + That is the equation from the slides of the first or second lecture to Bm25, but changed it now to the one I actually have implemented (Anserini).
- Your implementation differs from your reports equation, e.g. you do not have a k2
  + Since now switched to Anserini, k2 is not an issue anymore and equation in report has been changed

- I'm pretty sure the majority of the code is from `https://github.com/fanta-mnix/python-bm25/blob/master/bm25.py`, you should probably credit this to avoid issues.

  + I have used a BM25 implementation from pip module (rank-bm25) package, now resolved by using Anserini.

- Instead of implementing this yourself you could simply use Anserini which was linked to in the assignment text

  + Which I have done now and using Anserini for doing the BM25 ranking, which also imporves the performance by far.

2.4 QE

- Short - Could go more into detail, e.g. what is *qCent*? what is t?

  + Now explaining the function in more detail, hopefully detailed enough

- Missing exponential in your implementation of S_{Cent}

  + I also implemented a version with exponential but the running time got so bad (by calculating all the cosine similarieties between the documents and a vector), and the calculating exp on it, which as far as I see ended up on the same results. Only for negative cosine similarity it poses an issue but for the sake of running time scraped.

- Gensim does not specify whether similar_by_vector is using cosine similarity or not. I assume it is so i reckon that is fine.

  + As far as I read online it uses the cosine similarity

- Your fusion is incorrect, seems like you implemented Lq but didn't do the $p(t|q_i)$ part, at least i can't find any division or exponential operators

  + That's right the softmax was missing overall and the max fusion method wasn't implemented right, now changed

- Missing interpolation

  + Interpolation added

2.5 BM+WE

- Multiply (not multiplicate)

  + Rewrote the text, changed now

- What is BM25 with word embeddings? You briefly explain how you've implemented but you aren't describing what it is

  + Rewrote the whole section, hopefully good enough explained now

2.6 Experiments

- I would like a brief look at some of the files you create here, seeing that i don't have them available, like the first 5-10 lines of each file in an appendix

  + Removed all the unneeded files and doing it now with Anserini and this way don't need to compile a trec txt file first to run BM25 on it.

- Very brief - Can go more into detail, like, which trec_eval paramaters did you pass?

  + For once moved it into the jupyter notebook and also added a better explanation in the report.

2.7 Results

- "mthod"

  + Typo, text rewritten

- You seem to get very poor results, if you look at the Anserini github page you'll find that they get a MAP of 0.2531 on the robust04 dataset so it seems weird that you get a third of that. This leads me to believe that your BM25 implementation is incorrect.

  + Yes I seem to have done something totally wrong or the BM25 implementation was very bad, now with Anserini the MAP results seem to be similar to the anserini experiments page.

2.8 Discussion

- Compare the results to the theoretics, did the results match your expectations? Your results are lower than the original BM25, was this the intention?

  + Added a more detailed discussion with going into more detail as suggested

2.9 For final handin

- Finish up with BM25 + WE

  + I did implement BM25 + WE now

- Go more into detail in the report based on what i have written above

  + Duly noted and added more details to the report

- Fix your code. You'll probably find that Anserini can make your life easier for this task.

+ Yes Anserini made quite a few things easier, should have read the full manual of Anserini otherwise I wouldn't have skipped the part of how to use Anserini in Python