Altegrad 2015 : Final project

Text Categorization

Sammy Khalife, Oussama Ennafii, Shuyu Dong

09/04/2015

1 Graphs

Graph of words approach

We subdivided the task into three stages. First, we load the data and construct the document-term matrix, we reduce its dimensions and then we apply the learning method on it.

To get a document term matrix, we use the TW-IDF measure presented in [1]. So we made a function $'extractGraph.py'$ that extracts a graph from a document using the *networkx* library. This function can constructs a graph that can be weighted or not and directed or not. We can also tune in the window size through the argument *window*. That is when we introduce the function $'documentWordMatrix'$ that constructs obviously a document word matrix. In this matrix, we use the previous function to get the weights of each word depending on the graph and using the same approach as in [1]. Finally, we use the TW-IDF measure to calculate the document term matrix.

For the dimensionality reduction, we had two choices. Either, we use the Chi-square method or the Latent Semantic Indexing(LSI) in order to get a tractable matrix. Indeed, there are 14575 words in the train corpus. Both methods are already implemented on the $'sci-kitlearn'$ library.

In the third step, we choose two ways to learn. We compared the SVM and the AdaBoost algorithms. We used cross-validation on the train data to choose between the different parameters. SVM performed naturally better than Adaboost , as did the LSI. We reduced the dimension to 100 since it yielded the best results.

There were two ways to test: either we construct a document term matrix for the train set and the test set separately, or we could of constructed the matrix for the whole corpus and then divide it into a train matrix and a test one. The last approach is done so as to make sure that words that are not common for the test and train data sets are all taken into account.

We state here the results for a directed unweighted graph with a window of size 4. We reduce the dimension to 100 and we apply SVM: for the approach where we get the matrices :

- separatly:
  - Microaveraging
    - precision: 0.640475102787

    - ∗ recall: 0.640475102787
  - Macroaveraging
    - ∗ precision: 0.170622860994
    - ∗ recall: 0.203406087815

- jointly:
  - Microaveraging
    - ∗ precision: 0.667428049338
    - ∗ recall: 0.667428049338
  - Macroaveraging
    - ∗ precision: 0.631236989331
    - ∗ recall: 0.229378369

We did not have time to test other centrality measures for the graph construction. We could of used the eigenvector centrality which captures global properties of the document. It would of course take more time to compute since the $'networkx.eig_centrality'$ matrix powers to get the eigenvalues but it would be interesting to compare it to our results with a method that captures local properties of the document.

2 Bag of words approach:

We kept nearly the same configuration as in the first part: we kept Latent semantic indexing, which performs SVD to the Document-term matrix. The main question here was to find the number of dimensions to keep. In Reuters-21578 R8 dataset, there are 5485 samples on the train set and 2189 on the test set. Building the document term matrix and applying TF-IDF method yield features of size 14575 (total number of word). We kept SVM for classification following [2]. The SVD has been performed in two steps : one time on the train set and a second one on the test set.

Results

With cross-validation (10 values for C and γ with a radius basis function kernel), keeping a dimension of 100 after dimensionality reduction yields :

Micro-averaging precision : 0.781635449977
Macro-averaging precision : 0.198438628628
Micro-averaging recall : 0.781635449977
Macro-averaging recall : 0.242023076066

Conclusion:

In conclusion, the bag of words representation did better in general comparing with the graph of words representation except for the macro-averaging precision. We do not know how to

interpret this last remark. Otherwise, we should investigate in the future, the effect of a global property graph like PageRank for instance.

References

[1] Rousseau, François and Vazirgiannis, Michalis
Graph-of-word and TW-IDF: new approach to ad hoc IR,. Proceedings of the 22nd ACM international conference on Conference on information & knowledge management, pages 59-68, 2013, ACM

[2] Thorsten Joachims
Text categorization with support vector machines: Learning with many relevant features In Proceedings of the 10th European Conference on Machine Learning, ECML '98, pages 137–142, London, UK, 1998. Springer-Verlag.