Data Visualization Midterm

Jack Hirsh, Matthew Hoffman,
Daniel Silver

October 20, 2019

1 Introduction

Our project seeks to analyze the 20 Newsgroup dataset in many different ways, which we will go in depth on throughout this document. We first preprocessed the dataset and visualized some general information of the dataset such as the different categories and their frequency relative to the entire corpus and the most common words. In the Results & Analysis Section, we detail how we visualized this data and describe our visualization criteria. We then built two vocabularies after understanding the general idea of the data behind this corpus. One vocabulary was the unfiltered original 20 Newsgroups dataset, and the other filtered out headers and footers.

The continuing work we are about to describe, we applied everything to both vocabularies separately. We processed these vocabularies and filtered out punctuation and stop words, the lemmatized this data to get the bag of words and TFIDFs for each document in the corpus. We then created two topic models using LDA and visualized this after getting the topic distribution for each document. We then trained two Doc2Vec models upon these vocabularies and visualized this.

We later used the unsupervised learning clustering method, K-means for the following representations: Bag of Words, TFIDF, Topic Distribution, and Doc2Vec and compared the results from all of these representations to find how they stacked up against one another. We visualized this data as well. We then performed experimental analysis on many of the approaches listed above and will describe these analyses in the section below.

2 Results & Analysis

2.1 Visualizing Statistical Information of Dataset

We chose to use a pie chart to represent the amount of documents in each category of the cluster. In summary, there were 20 different topics and each topic had roughly the same representation in quantity of documents. We also graphed the top 20-most common words which led to some interesting results. We found out that the most common words were stop words, which shows to us the importance of removing the stop words from documents during preprocessing as this adds useless noise to the dataset as a whole.

2.2 K-means discussion

We tested K-means on TF-IDF and Bag of Words and found that the TF-IDF vectorizer significantly out performed bag of words as its completeness score was 3x as good. We found that the Normalized Mutual Information returned a score of 0.28623017866935624.

2.3 Experimental Analysis

We found the impact of different preprocessing methods. We found that our models generated better visualizations when we filtered out stop words and lemmatized the words. The only drawback of lemmatization is that when you produce word clouds, you may get word steps, which are not as pleasing to see. We also found that keeping in headers and footers are beneficial information and seemingly extraneous data should be tested before filtering it out (more on this in the Bonus Work section). We also found a relationship between number of perplexity and scoring of the model and the number of topics in unsupervised models such as LDA. We found that as the number of topics increases, the perplexity decreases and the score increases.

Different training factors for Doc2Vec include the number of epochs used for training the model, the alpha values (i.e. weight of values allocated to retaining previous epochs), and the size of vectors used to visualize docs. The key factor for document visualization is fully understanding the statistical intricacies and scope of the data, such that you may clearly represent associativity between terms and documents.

3 Bonus Work

The bonus work we had done was to generate word clouds for each category using the mask of an image that represents said category. Additionally we used the labels of this data for supervised learning as we trained a Naive Bayes' classifier to try to learn how to classify the documents. We also tried a standard multinomial classifier which performed significantly worse so we chose to include only the results of the Naive Bayes' classifier. We used a 80-20 testing to training split and were able to achieve 82.9% accuracy using the entire vocabulary. When we attempted this with filtering out the headers and footers, which we previously believed to be useless and noisy information, we only achieved an accuracy of 79.4%, which was surprising as we believed that filtering out this data would improve our accuracy. This was one of the greatest takeaways of this assignment.

4 Conclusion

In conclusion, this was a very successful project with many takeaways, both in the form of lessons and in the form of useful visualizations. We were able to compare 2 vocabularies and understand how different preprocessing methods affects the end machine learning results, through both supervised and unsupervised types of methods.

5 References

1. Gensim
2. scikit-learn
    - http://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html
3. WordCloud
    - https://www.datacamp.com/community/tutorials/wordcloud-python
- https://github.com/amueller/word$_c loud.git$
4. Natural Language Toolkit
5. RegExr
6. David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. J. Mach. Learn. Res. 3 (March 2003), 993-1022.