COMP-551 Final Project

T1-10: Distributed Representations of Sentences and Documents

Team: Linear Kernel Mustard

Daniel Cimento
daniel.cimento@mail.mcgill.ca
260679318

Adam Edery
adam.edery@mail.mcgill.ca
260691043

Howie Zhao
haoyi.zhao@mail.mcgill.ca
260631984

I. Objective

For this project, we were tasked with recreating the baselines reported in the paper "Distributed Representations of Sentences and Documents" (Le & Mikolov, 2016). Our objective was two-fold: first we aimed to achieve the same basline performances as the paper, then if possible, to tune those baselines further, achieving more competitive performance. Though we anticipated that we would be able to beat the reported baselines, we did not have much expectation that we would outperform the main model of the paper, the Paragraph Vector model.

II. Methodology

Our methodology for this project was divided into several distinct phases. First, we applied some pre-requisite processing to the data set. Then, we implemented the simpler linear models that were reported in the paper. After this, we applied further tuning to these models, to optimize the performance wherever possible. We then explored alternative linear models, to see if any could outperform those chosen by the paper's authors. Finally, we explored the performance of the other, more complex models like word vector averaging and neural networks, though these were not intended to be the focus of our efforts.

A. Data Pre-processing

The pre-processing and feature selection composed a non-negligible part of our methodology, so we will cover the steps taken to process the raw data sets.

1) Manual Data Cleaning: In the first place, the sentence files provided in the Stanford sentiment analysis data set were encoded in Latin-1. Since our code would benefit from having a consistent encoding standard throughout the development process, we needed to handle these files with the UTF-8 encoding standard. This resulted in parts of the file being inconsistently encoded, with mojibake appearing occasionally throughout our dataset, which proved troublesome for Python's default IO implementation. Consequently, the first step of our data pre-processing involved cleaning the sentence file by replacing all Latin-1 encoded characters with their corresponding unicode equivalent.

Furthermore, the Stanford sentiment analysis data set is formatted with the intention of being used for syntax parsing models, as the sentences are structured in a tree-like format, and their corresponding sentiments are stored across several files. For our application, we wanted to handle the full text of each sentence, so we wrote the code necessary to convert the tree-like structure into simple "sentence-sentiment" pairs.

2) Vectorization: Once we had our data loaded in the proper format, we needed to extract the features from each sentence to create vectors for our models. The simplest representation would be bag-of-words, so we started with that, using sklearn's built in CountVectorizer.

However, we felt that just using bag-of-words wouldn't accurately capture the semantic similarity between inflected words (e.g. "film" and "films"), so we replaced CountVectorizer's built-in tokenizer with a tokenizer that tagged the parts of speech for each word and retrieved that word's lemma from the WordNet database. To do so, we took advantage of several features built into the nltk package. We also used a CountVectorizer that included both unigrams and bigrams (for our Bigram Naive Bayes implementation), which used the same tokenization scheme.

Similar to the paper under consideration, we labelled each sentence in the Stanford sentiment analysis dataset based on its sentiment score (using the labels Very Negative, Negative, Neutral, Positive, and Very Positive). We likewise created a set with coarser label classifications (with labels Negative and Positive), ignoring all neutral data points for our coarse-grain performance evaluations.

The IMDB data set was vectorized in a similar way, but since it used a binary label approach, we didn't need to make any modifications to the labels.

B. Simple Linear Models

For the simple linear model methodology, there is little to discuss that hasn't already been covered by the paper under consideration, but we will summarize the main points of each.

1) Naive Bayes:

2) Linear SVM:

3) Bigram Naive Bayes:

C. Improving Linear Models

1) Naive Bayes:

2) Linear SVM:

D. Alternative Simple Models

For each of the alternative simple models we tested, we used the default sklearn implementations. We will briefly touch upon our rationale for choosing these models, as well as the hyperparameters we tested, and what effect we anticipated they would have on the model's performance.

1) Random Forests: Sentiment analysis with bag-of-words can be ultimately thought of as an entropy based task. Since certain words are significantly more positive or negative than others, the use of these words is liable to heavily influence the overall sentiment of that review. Other words are only somewhat charged, so they will have only a mild influence on the overall sentiment.

Because of these facts, decision trees seem to be an excellent candidate for sentiment analysis. However, decision trees come with non-negligible downsides, not least of which are the long training times and the risk of overfitting.

The Random Forest classifier serves as an excellent compromise to these problems. The classifier creates multiple small trees based on random subsets of the features, so the training time is kept low overall, since no single tree is too deep and not as much consideration needs to be given to features that have low overall impact on the impurity.

For hyperparameters, we experimented with the maximum depth of each tree, the criterion used for deciding the split at each level of the tree, and the number of trees in the forest.

The impact of the maximum depth is easily understandable: if there are many words in our vocabulary with high entropy, then allowing our trees to use more of those words will create a more accurate ensemble of trees.

The number of trees in the forest has a similar rationale. Since results are averaged from the results of the trees, having more trees can reduce misclassifications that are caused by the random feature selection.

We didn't expect to see much variance in performance by changing the split criterion, since entropy and gini impurity are known to perform similarly well. We had expected entropy to perform slightly better, since entropy is more commonly used for classification into discrete classes. Nevertheless, there are only two criteria to choose between, so we felt as though it was worth considering in the event that our hypothesis was incorrect.

2) Logistic Regression: Logistic Regression, similar to Random Forests, attempts to identify "explanatory variables" that have an impact on determining the classification of a given vector, and it then fits a logarithmic distribution to that data.

In effect, since we anticipated that certain words would have a high impact on determining class, the Logistic Regression would be able to prioritize certain features in determining overall sentiment. We thought as well that it might perform even better than the Random Forests, since the Random Forests only indirectly prioritize important features through chance repitition. Since we have so few categories, and since our lemmatization approach to vectorization would likely increase the number of events per explanatory variable, we anticipated Logistic Regression to be one of our best performing models.

The hyperparameters we experimented with were the norm function used in the penalization, the tolerance for the stopping criteria, and the regularization strength.

The tolerance for stopping only has a real impact when we struggle to further improve our model, so we would expect a low tolerance to perform better, since it would keep training until it finds the optimal value.

The regularization strength determines how much cost we would attribute to increasing the magnitudes of our parameters, which means that lower regularization strength would allow us to better classify data sets without clear delineations between classes, and high regularization strength would reduce overfitting if there are fairly clear delineations between classes. As a result, we anticipated that a higher regularization strength (low C in sklearn) would give better performance for our coarse-grain evaluation models, whereas a lower regularization strength (high C in sklearn) would give better performance for our fine-grain evaluation models.

The norm function for penalization wouldn't have much impact on its own, but when coupled with high regularization strength, we would expect L2 to provide more accurate solutions.

3) K-Nearest Neighbors: K-Nearest Neighbors is notoriously ineffective for problems such as these, since the distance-based metrics used for calculating "nearness" fall apart at high dimensionalities. Nevertheless, the training time is effectively non-existent, and if it performed well, it would have the potential to prove the null hypothesis for the task at hand, so we thought it worthwhile to at least entertain this model.

The hyperparameters we considered were the number of neighbors queried for each testing point and the weight of each neighbor.

Varying the number of neighbors to a certain value k would expect to improve the model's performance if we anticipated data points of any given class to be found in clusters of size k. Changing the weight of the neighbors to scale based on the distance from the testing point would improve the performance of the model if we anticipated points to be found in tight, compact clusters.

Since we couldn't intuit the layout of the vectors in any meaningful way, our approach to tuning these parameters was more or less arbitrary.

E. Complex Models

1) Word Vector Averaging:

2) Neural Networks:

III. Results

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi nec orci sed turpis placerat dapibus ac non justo. Phasellus volutpat sem felis. Nam id tincidunt massa. Quisque diam ante, mattis et molestie porttitor, rhoncus a dolor. Aliquam sed diam porta turpis dapibus ultricies. Nam sit amet vulputate felis. Vestibulum commodo placerat neque, et feugiat odio. Mauris tristique malesuada leo, nec finibus orci varius non. Sed quis mattis velit. Morbi rhoncus ante vel sodales ullamcorper. Duis feugiat accumsan metus id pellentesque. Nulla pellentesque, leo eget vehicula malesuada, turpis diam egestas neque, non vulputate purus nibh lobortis odio. Nunc maximus aliquam nisl vitae molestie. In hac habitasse platea dictumst. Fusce ut luctus diam. Aliquam aliquam laoreet felis, sed dictum nulla facilisis et.

In interdum suscipit mattis. Sed porta imperdiet purus, at imperdiet leo condimentum eget. Pellentesque at velit mauris. Phasellus venenatis arcu eget lorem efficitur, ut pulvinar purus tincidunt. Aliquam vel nibh tempor, blandit lacus vel, ullamcorper sem. Nam porttitor, ipsum tempus varius elementum, quam neque ultricies nisi, sed commodo tortor eros in nibh. Proin aliquet placerat erat in varius. Proin semper odio ut eros condimentum vestibulum. Mauris id porttitor ipsum. Sed tincidunt enim enim, sed dapibus urna aliquam sed. Sed a ex elementum, hendrerit eros ac, euismod dui. Etiam ligula ligula, scelerisque vitae lectus et, ultrices semper erat.

Vivamus id dictum nulla. Praesent consectetur pellentesque dui quis scelerisque. Nunc sed augue vehicula, scelerisque felis ut, viverra nulla. Curabitur aliquet ipsum non tincidunt congue. In pretium nec nisl et euismod. Sed vitae euismod lorem, vitae auctor est. Mauris varius orci ac augue semper, sed sollicitudin mi porttitor.

Nullam efficitur ante in bibendum rutrum. Suspendisse congue pretium metus vitae scelerisque. Maecenas vel justo finibus, bibendum nibh non, egestas metus. Vivamus sed mollis purus. Mauris interdum a ligula vitae tristique. Nam vel varius metus, a tincidunt elit. Curabitur convallis augue lacus, eget dapibus risus pretium non. Proin ut consectetur quam, ut lobortis nibh. Cras dui elit, suscipit euismod laoreet et, interdum id nulla. Cras varius augue eget enim semper, sed tempus enim sagittis.

Phasellus mattis maximus dui, id vestibulum mi finibus ut. Maecenas commodo fringilla magna quis dignissim. Praesent eget felis vulputate, eleifend dolor eu, finibus elit. Aliquam rhoncus mi ut tellus laoreet sodales. Sed sed luctus purus. Praesent hendrerit tortor sit amet nibh dignissim, sed gravida magna dictum. Nam mollis fermentum dui placerat auctor. Pellentesque vel quam tellus. Mauris gravida, leo eu commodo euismod, velit ante rutrum sem, sed sollicitudin mi tellus vel diam. In mattis malesuada sagittis. Vivamus vel fringilla tortor. Nulla ac sem in turpis rutrum sodales. Nullam accumsan dolor ac auctor pellentesque. Cras suscipit, eros at lobortis commodo, erat diam sollicitudin augue, interdum ullamcorper turpis dolor sit amet magna.

IV. Discussion

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi nec orci sed turpis placerat dapibus ac non justo. Phasellus volutpat sem felis. Nam id tincidunt massa. Quisque diam ante, mattis et molestie porttitor, rhoncus a dolor. Aliquam sed diam porta turpis dapibus ultricies. Nam sit amet vulputate felis. Vestibulum commodo placerat neque, et feugiat odio. Mauris tristique malesuada leo, nec finibus orci varius non. Sed quis mattis velit. Morbi rhoncus ante vel sodales ullamcorper. Duis feugiat accumsan metus id pellentesque. Nulla pellentesque, leo eget vehicula malesuada, turpis diam egestas neque, non vulputate purus nibh lobortis odio. Nunc maximus aliquam nisl vitae molestie. In hac habitasse platea dictumst. Fusce ut luctus diam. Aliquam aliquam laoreet felis, sed dictum nulla facilisis et.

In interdum suscipit mattis. Sed porta imperdiet purus, at imperdiet leo condimentum eget. Pellentesque at velit mauris. Phasellus venenatis arcu eget lorem efficitur, ut pulvinar purus tincidunt. Aliquam vel nibh tempor, blandit lacus vel, ullamcorper sem. Nam porttitor, ipsum tempus varius elementum, quam neque ultricies nisi, sed commodo tortor eros in nibh. Proin aliquet placerat erat in varius. Proin semper odio ut eros condimentum vestibulum. Mauris id porttitor ipsum. Sed tincidunt enim enim, sed dapibus urna aliquam sed. Sed a ex elementum, hendrerit eros ac, euismod dui. Etiam ligula ligula, scelerisque vitae lectus et, ultrices semper erat.

Vivamus id dictum nulla. Praesent consectetur pellentesque dui quis scelerisque. Nunc sed augue vehicula, scelerisque felis ut, viverra nulla. Curabitur aliquet ipsum non tincidunt congue. In pretium nec nisl et euismod. Sed vitae euismod lorem, vitae auctor est. Mauris varius orci ac augue semper, sed sollicitudin mi porttitor.

Nullam efficitur ante in bibendum rutrum. Suspendisse congue pretium metus vitae scelerisque. Maecenas vel justo finibus, bibendum nibh non, egestas metus. Vivamus sed mollis purus. Mauris interdum a ligula vitae tristique. Nam vel varius metus, a tincidunt elit. Curabitur convallis augue lacus, eget dapibus risus pretium non. Proin ut consectetur quam, ut lobortis nibh. Cras dui elit, suscipit euismod laoreet et, interdum id nulla. Cras varius augue eget enim semper, sed tempus enim sagittis.

Phasellus mattis maximus dui, id vestibulum mi finibus ut. Maecenas commodo fringilla magna quis dignissim. Praesent eget felis vulputate, eleifend dolor eu, finibus elit. Aliquam rhoncus mi ut tellus laoreet sodales. Sed sed luctus purus. Praesent hendrerit tortor sit amet nibh dignissim, sed gravida magna dictum. Nam mollis fermentum dui placerat auctor. Pellentesque vel quam tellus. Mauris gravida, leo eu commodo euismod, velit ante rutrum sem, sed sollicitudin mi tellus vel diam. In mattis malesuada sagittis. Vivamus vel fringilla tortor. Nulla ac sem in turpis rutrum sodales. Nullam accumsan dolor ac auctor pellentesque. Cras suscipit, eros at lobortis commodo, erat diam sollicitudin augue, interdum ullamcorper turpis dolor sit amet magna.

V. Conclusion

VI. Statement of Contributions

VII. References