Unsupervised Learning on Doc2Vec Embeddings for Text Classification

Nakul Gupta, Nghi Le, John D'ambrosio, Gary Zhou, Sanghun Kim
University of Southern California
3601 Trousdale Parkway,
Student Union 301 Los
Angeles, CA
nakulgup@usc.edu, nghile@usc.edu,
jfdambro@usc.edu, garyz@usc.edu,
sanghunk@usc.edu

1 Motivation

Current web technologies produce an unprecedented amount of data, the majority of which is unstructured and unlabeled. Many powerful machine learning algorithms for the classification of this data have been developed in recent years, yet the vast majority of these are supervised methods, which require labeling. As a result, research has increasingly focused on developing unsupervised machine learning methods.

In text classification, the ability to accurately classify texts typically requires a Naive Bayes or logistic regression approach at a minimum, although neural network and deep learning approaches have become more powerful and more popular. In this paper, we look at an alternative method for classification: clustering algorithms trained on document embeddings. The goal was to ascertain whether embeddings generated from the text naturally cluster into groups that are similar to the sentiment groups in the dataset.

The hypothesis was that since word vectors cluster naturally based on the word (i.e king and queen have similar coordinates in feature space), documents of similar sentiment may also cluster in similar locations. For example, positive sentiment documents may contain the same set of positive words (*good, great, amazing, love*, etc) and thus a document vector generated from such words may come to represent a generalized "positive" location in the vector space of document embeddings. Conversely, a similar argument can be made for the negative polarity.

In the literature, this avenue has not been extensively explored, with some authors exploring a method of clustering documents by both topic and sentiment (Huang, 2017), while others have treated sentiment as a noise variable in the clustering of documents (Gutierrez-Batista, 2016).

2 Design

Our project aims to study whether unsupervised learning can be used for classification tasks when a popular language model vector embedding is used as a text representation. To answer this question, we first used a pre-trained language model to generate document embeddings from a sentiment dataset. A standard clustering algorithm was then used to cluster the document embeddings. Lastly, we evaluated the clusters by comparing them to the labels in the dataset in a Monte Carlo simulation to obtain the final accuracy of the cluster assignment.

2.1 Materials

We used the IMDb movie review dataset to conduct our unsupervised sentiment analysis. The dataset contains 50,000 training and testing samples with binary sentiment labels (positive and negative), where each row represents a unique movie review. The data is high-polar, which means that it is easy for humans to distinguish between positive and negative sentiment movie reviews.

2.2 Methods

We used Doc2Vec (Le et. al, 2014) to encode each movie review into a numerical vector. Doc2Vec is a method to generate numerical representations of text documents regardless of their length. While Word2Vec generates word vectors that represent the machine-learned concepts of a single word, Doc2Vec generates document vectors that represent the concept of the document as a whole. Thus, we wanted to find out if Doc2Vec can generate document vectors that form clear cluster groups to help us identify positive and negative sentiment documents.

The document vector generated with Doc2Vec captures the learned concepts of a document. We were hoping that Doc2Vec can generate document vectors that form clear cluster groups to help us identify positive and negative sentiments. To see if our hypothesis is correct, we initially used a method called T-distributed Stochastic Neighbor Embedding (Laurens and Hinton, 2008) (t-SNE) to reduce the Doc2Vec documents vectors to a visualizable 2-d space and then graph them. We hoped to see clear patterns of separability between the two sentiments in the t-SNE visualization.

Then, a Doc2Vec embedding model was trained using the first 10,000 rows of the shuffled IMDB sentiment dataset. The Doc2Vec model was trained on the following hyperparameters:

- learning rate: 0.025
- minimum learning rate: 0.00025
- feature vector size: 500
- max epochs: 10

Then, we put all 50,000 rows of the dataset into the Doc2Vec model to generate the document vectors. Among many clustering algorithms such as Gaussian Mixture Models and DBSCAN, we selected the K-means clustering algorithm as our unsupervised clustering method.

The Doc2Vec vectors were clustered using the K-means algorithm with two main clusters, as the dataset had only two sentiment labels (positive and negative). The predicted labels on the document vectors were compared against the document vectors' true sentiment labels to calculate a measure of clustering accuracy.

3 Results

First, we sampled a random set of 1000 Doc2Vec vectors for the t-SNE visualization. The visualization is shown in Figure 1.

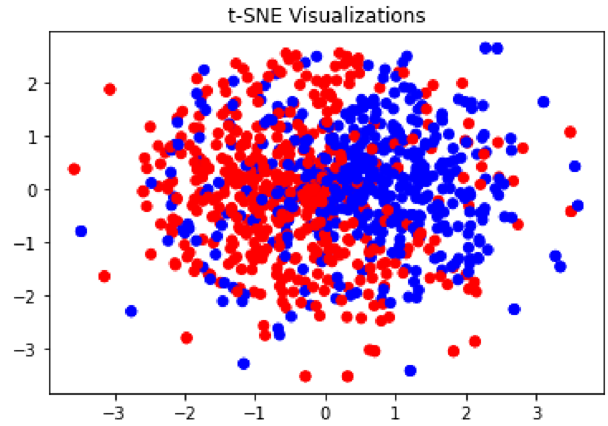

Figure 1: t-SNE visualization of a random 1000 Doc2Vec embeddings. The red and blue colors indicate two different sentiments.

As Figure 1 shows above, the data did not appear to readily separate into distinct clusters in this reduced, low dimensionality representation. This was the first indication that the proposed hypothesis may be incorrect. However, it was also possible that due to dimensionality reduction, the relationship between clusters had collapsed. As a result, we continued with the proposed analysis of clustering the documents in the high dimensional space.

Due to the randomness of clustering algorithms like K-means, we needed a method to assess the accuracy of each cluster found. We used the Monte Carlo simulation method to run 50 trials to measure the average performance of the K-means clustering algorithm on predicting the correct sentiment labels.

The Monte Carlo simulation reduced the variance found on individual runs and allowed us to find an average accuracy as well as the standard deviation. Table 1 shows these results, with both clusters having close to 54% mean accuracy with a standard deviation of .021. These results prove that across all runs, the clusters found were consistent in prediction.

Cluster	Mean	Standard Deviation
0	.535	.021
1	.536	.021

Table 1: Mean accuracies for each cluster with the respective standard deviations

The cluster means are similar, yet this table does not paint the entire picture. Clusters 0 and 1 both alternated between two possible groupings of sentiment data points. The means above are the average between these two groupings.

Below, Figure 2 is a histogram that contains all 100 accuracies found throughout the Monte Carlo simulation (50 for each cluster).

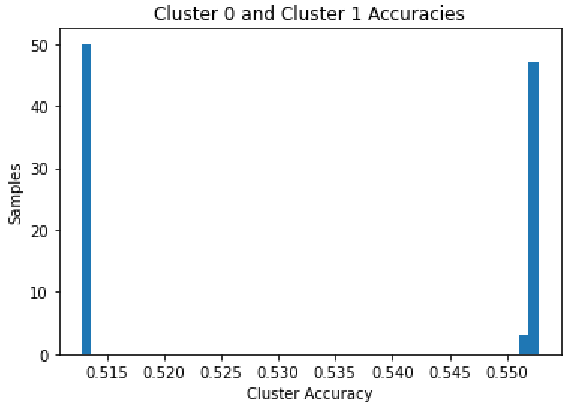

Figure 2: Histogram containing 100 samples from the Monte Carlo simulation. Two peaks around found representing the two sentiment clusters

Two peaks are seen in Figure 2, representing the two sentiment clusters that are found by K-means. The peak that is found centered close to .552 will be labeled Sentiment A while the peak found centered at .513 will be labeled Sentiment B.

Since both clusters found both sentiments several times, we need to examine the accuracy of each cluster for both sentiments A and B. For more insight, these two peaks were isolated for both clusters. These results can be seen in Figures 3 through 6.

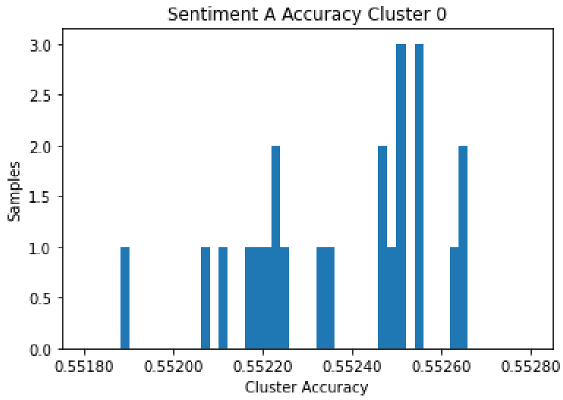

Figure 3: Histogram containing all samples and accuracies when Cluster 0 found Sentiment A. The average accuracy was found to be around .5523

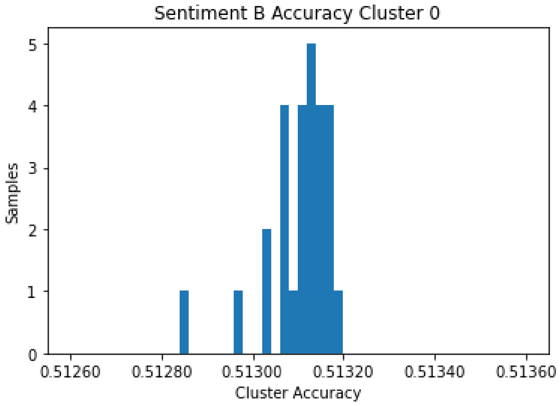

Figure 4: Histogram containing all samples and accuracies when Cluster 0 found Sentiment B. The average accuracy was found to be around .5131

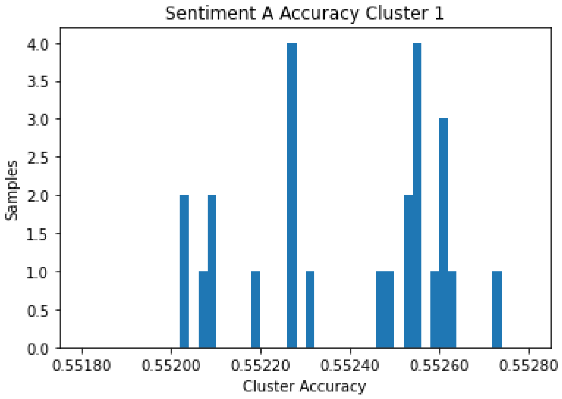

Figure 5: Histogram containing all samples and accuracies when Cluster 1 found Sentiment A. The average accuracy was found to be around .5523

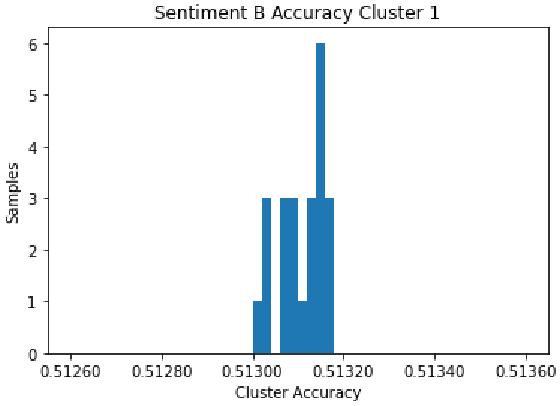

Figure 6: Histogram containing all samples and accuracies when Cluster 1 found Sentiment B. The average accuracy was found to be around .5131

Taking a closer look at the results for each cluster, we see that Sentiment A was always found with higher accuracy when compared to Sentiment B, regardless of which cluster found it. This can be explained when looking at the size of cluster sentiment pairing. During a single run of the Monte Carlo simulation, the size of whichever cluster found sentiment A had approximately 20,000 samples while sentiment B had approximately 5,500 samples.

We can attribute this to the random nature of the clustering algorithm. As Figure 1 showed, since the data did not appear to readily separate, this will lead to poor performance by the K Means clustering algorithm, which will tend towards "discovering" a main cluster and small spurious cluster that does not truly correlate with any real sentiment. In these situations, the clustering algorithm will fail to distinguish any true groupings.

In summary, the clustering algorithm failed to truly distinguish between any two clusters, instead continuously finding a main cluster and a spurious, "trash" cluster. The histograms showed the alternating nature of these clusters and their classification accuracy, with the main cluster showing a 56% accuracy and the spurious cluster showing a 51% accuracy. Averaged together, the overall clustering method produced a 53% accuracy, which is only slightly better than random guessing.

4 Conclusion & Future Work

We can see that using Doc2Vec with K-means clustering results in just barely better prediction accuracy than guessing. We can attribute this to two possible hypotheses.

The first is that Doc2Vec is not an appropriate language model for our objective or that we are using unoptimized hyperparameters. We see that the Doc2Vec text embeddings of document vectors do not cluster naturally based on sentiment. We do see, however, that particular clusterings are found, specifically two. However, these clusters consisted of the main cluster and a random, spurious "trash cluster" that was not particularly distinguishable from the main.

Determining what these clusters represent would help illuminate what hyperparameters can be adjusted to achieve our goal. The motivation behind using Doc2Vec was based on the similar model, Word2Vec, where similar words would share similar embeddings. It appears that while Doc2Vec may have embedded documents that are similar in syntax and diction close together, the semantic relations found within each document were not preserved. To further experiment with document embeddings, we will need to work with other methods such as bag of words, bag of n-grams, and tf-idf.

The second hypothesis is that K-means is an inappropriate method of unsupervised learning for this task. While K-means is simple and interpretable, we may need to use projection methods to find more separable classes, such as PCA or Fisher's Linear Discriminant Analysis. For future work here, experimenting with additional unsupervised learning methods, such as Gaussian Mixture Models, DBSCAN, or Spectral Clustering may improve prediction accuracy.

After the models have been tested and validated with sentiment analysis, future work can be performed on other types of text classification. These categories include detecting truthful and deceptive articles, documents that contain hate speech, and sarcasm.

Lastly, sentiment embeddings can also be studied for the unsupervised sentiment classification task. While current research suggests generating these sentiment embeddings in a supervised manner, a novel task formulation or inductive bias needs to be introduced to learn embeddings that contain sentiment information in an unsupervised way. This is an interesting area of research that can be pursued.

Division of Labor

The entire team together finalized the project direction, procedure, and paper as well as presented the poster to the class. John performed the Monte Carlo simulation and then plotted and discussed the findings. Sanghun researched different potential clustering methods such as K-means and GMM. Nghi did t-SNE visualization and explored the literature for other venues such as sentiment embeddings. Nakul was responsible for much of the main code of the experiment and running the algorithms on his machine, while conducting remote pair programming sessions with the team. Gary prepared the IMDb dataset and wrote the setup portion of the main code.

References

Le, Q. V. and Mikolov, T., "Distributed Representations of Sentences and Documents", 2014.

J. -J. Huang, "Using Topic and Subjectivity Analysis for Overlapped Co-clustering Documents," 2017 IEEE Third International Conference on Multimedia Big Data (BigMM), 2017, pp. 105-108.

Gutierrez-Batista, K., Campaña, J.R., Martinez-Folgoso, S., Vila, M.A., Martin-Bautista, M.J. (2016). "About the effects of sentiments on topic detection in social networks." International

Journal of Design & Nature and Ecodynamics, Vol. 11, No. 3, pp. 387-395.

Maaten, Laurens, and Geoffrey Hinton. “Visualizing Data Using T-Sne,” Journal of Machine Learning Research 9 (2008) 2579-2605

“Imdb · Datasets at Hugging Face.” *Imdb · Datasets at Hugging Face*, https://huggingface.co/datasets/imdb.

Mishra, Deepak. “Doc2vec Gensim Tutorial.” *Medium*, Medium, 30 Aug. 2018, https://medium.com/@mishra.thedeepak/doc2vec-simple-implementation-example-df2afbbfbad5.