B455 Final Project

Apply Word Embedding and Dimensionality Reduction on B455 Assignment Sheets

Emma Cai

**Introduction:**

This project aims to learn the vector representations of words taken from B455 homework sheets and the project sheets as well as to try several ways to reduce dimensions and visualize it in a 2D graph to see if there are clustering on homework words and assignment words. This project has three parts. First, convert every word showed in the assignment sheet into a 32-dimensional vector use word embedding. Second, three ways (PCA, t-SNE, LDA) are applied to further reduce the dimension to 2D and plot all the words. Lastly, I used the graph convolutional network to learn the feature further and reduce the dimension after that.

**Word Embedding and preprocessing:**

Word Embedding enables the words with similar meanings to come closer in a vector space. It is widely used in natural language processing. In this project, I implement it myself. I build my corpus, construct the input words, and output words with one-hot encoding and build a skip-gram model from starch to train. Eventually, after 1200 epochs of training, all the words showed in the assignment sheets were represented by 32 real numbers. To compare, I also use the genism library to convert words to vectors. By using the library, I spent less time to train and were able to choose a larger window size and discard the words showed in a less frequency.

**Dimensionality Reduction:**

A picture containing screenshot

Description automatically generatedA screenshot of a cell phone

Description automatically generatedThe first model I tried is PCA. I also implemented it myself. I find two major components of my words vector space and discards the other 30 eigenvectors to maximize variance in my 2D graph. Here are my graphs.

The second one I tried is t-SNE. t-Distributed Stochastic Neighbor Embedding is an unsupervised, non-linear technique used for visualization. It is an intuitive way to visualize the relationship of points in their original high dimension. I used to sklearn library to achieve this.

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedHere are the graphs.

The last way for dimensionality reduction is LDA. Linear discriminant analysis (LDA) aims to find a linear combination of features that can separate two or more classes of objects.

This is a supervised learning method so I labeled the words as homework words, project words, and words showed in both homework and project. Since I have 3 classes, I can reduce it to 2D space. Here is the graph.

A screenshot of a cell phone

Description automatically generated

**Graph Convolutional Network:**

Graph Convolutional Network (GCN) is a neural network perform a convolution on a graph, instead of an image in pixels. This is a fairly new while popular idea proposed by Thomas N.Kipf in 2017. I try this method because words are like nodes on the graph and the embedded vectors are their feature. The edge can be the word with their neighbor. As always, I choose t-SNE to visualize it. Here is the graph.

A screenshot of a cell phone

Description automatically generated

**Evaluation**

This is an unsupervised learning project (expect for LDA), therefore there are no such criteria like accuracy score. In this project, I successfully represent words as vectors and then reduce the dimension using PCA, LDA, and t-SNE. PCA supposed to maximize the variance, but it didn’t do a good job in those words vectors. The reason for that is all the eigenvalues are too close to average. In the ipynb file, you can see all my 32 components, the biggest one is 0.0549 and the smallest one is 0.0148. A possible way to solve this is to reduce the dimension from 32 to 16 and then apply t-SNE to visualize it. From the graph, we see that the LDA method didn’t do well in classify three classes, the blue, yellow and red nodes are mixed. From the t-sne graph, we can see nodes have been separated pretty well. However, there is no obvious clustering. The good news is, using GraphConv and t-sne, we can see 5 or 6 clusterings.

**Conclusion**

We spend a lot of time in the classification methods in B455 and only have one homework about unsupervised learning and dimensionality reduction. Therefore, I decided to explore more about those algorithms used for clustering and reduce dimension. It is interesting to apply these methods in the B455 assignment sheets. It turns out the best approach is to first represent words using vectors then using GraphConv to learn features from its neighbors, and finally generate the map using t-sne.