Machine Learning
hw4 report

Han Hao Chen, r05922021

December 6, 2016

1 TF-IDF Stopwords Removal

Stopwords Removal The stopword list is decided by the TF-IDF value of vocabularies. First, I calculate the TF-IDF value for words in each document. Then, calculate the average value for each vocabulary from all documents. Finally, I choose top 35 from the sorted average TF-IDF list in ascending order as stopwords.

Table 1: Top 5 averaged TF-IDF

is	0.142818749521
how	0.157200672237
in	0.158715468052
a	0.174704362809
the	0.207745522403

2 TF-IDF Method

TF-IDF I use TF-IDF method to represent the document vector, each document can be represent as $|V|$ dimensions' vector with TF-IDF feature. In prediction part, since TF-IDF vector have no spacial meaning, using k-means to do the clustering is not a good idea. I use dot value to predict each pair, if value is greater than threshold, the result will be 1, and vice versa. To improve the TF-IDF performance, I remove some hand-craft stopwords, and compare the results' performance.

Table 2: TF-IDF Comparision

TF-IDF	f1-score
K-means(k=20)	0.09142
Dot value	0.32710
Dot value, SW removal	0.33149

3 Word2Vec Method

Word2Vec To create meaningful word vectors, I use skip-gram method to train my word vector on title_StackOverflow corpus. Each title can be represent as summation of word vector in the sentence and divided by length of the sentence. Finally, We can use k-means clustering to cluster our document vectors and do the prediction.

Table 3: Word2Vec Clustering Comparision

Word2Vec	f1-score
K-means(k=20, no SW removal)	0.43150
K-means(k=20)	0.53886

4 Different Feature Comparision

Comparision TF-IDF feature can extract title feature within few minutes, but the performance is not very good(can only achieve Baseline). Since it can't capture latent meaning of words by using TF-IDF, we can't use k-means clustering on this feature. Word2Vec method will take some time to train word vector on corpus, but it can capture latent meaning of words and has some spacial meaning. According to these characteristics, we can use k-means to do the clustering and achieve good performance. To demonstrate this characteristic, I visualize document vectors by these two feature extraction method to show the difference.

Figure 1: Document Vector - Word2Vec

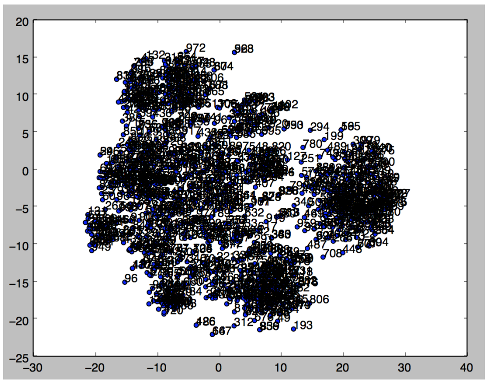
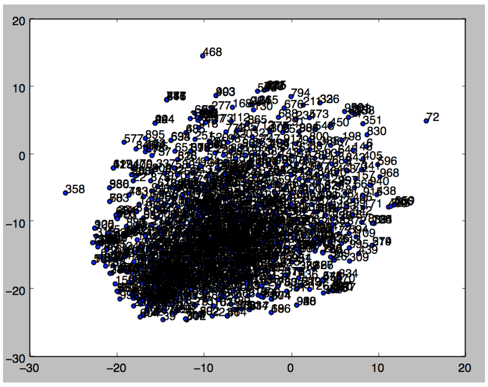


Figure 2: Document Vector - TF-IDF



5 Different Cluster Number Comparision

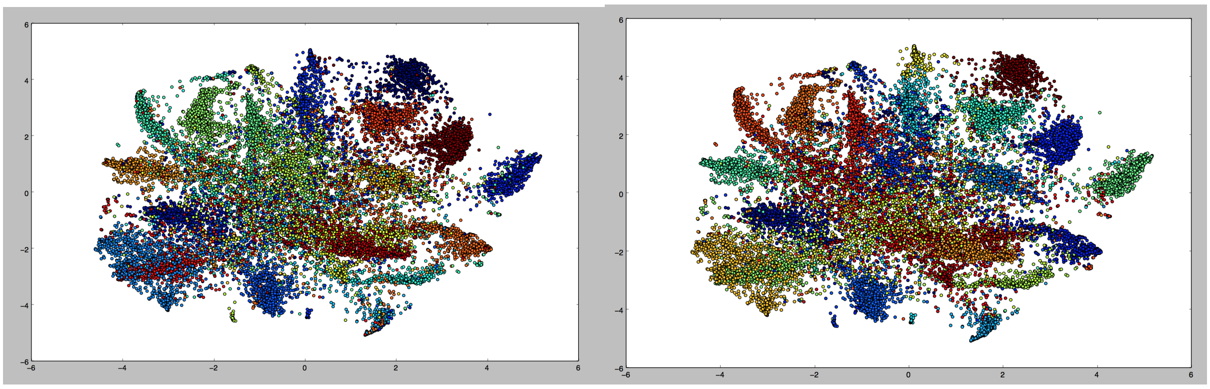
Cluster Number In this section, I will compare k-means clustering results on word2vec feature with different cluster number. The result shows that the best result is the cluster number k=24. The possible reason might be that the document vector is not good enough to cluster vectors into 20 categories. Therefore, using extra categories to classify those arbitrary vector into some category will have higher performance.

Table 4: Different Cluster Number On Word2Vec Feature

Cluster Number	f1-score
K= 15	0.42800
K= 20	0.53886
K= 24	0.64774
K= 25	0.64659

6 Training Tips

- Split camel case word will achieve higher perfomance.
- Stopwords removal also has great influence on performance.
- Using semi-supervised training on word vector will do better on clustering.(can't be use in this assignment)



(a) with ground truth labeled

(b) with prediction labeled

Figure 3: Document Vector - Word2Vec