COEN240 Final Project Report

Guohao Sun, Jiahong Li, Xuwei Pan

March 11, 2022

0.1 Introduction

This is an NLP project, we implemented different feature embedding methods and use k-means clustering to visualize the document cluster. The purpose of this project is to observe the performance of four methods (BOW, TF-IDF, LDA, Doc2V). After feature embedding, we feed the input into Transformer module for a classification task.

0.2 Experiment Steps

0.2.1 Preprocess the dataset

In this project, we used 20newsgroups dataset, which includes 18846 documents. We tokennize each document; remove stopwords; remove digits and one-character word. After preprocess the whole dataset, we plot the term-frequency distribution using the len(new token)/len(old token). See the result in Figure 1.

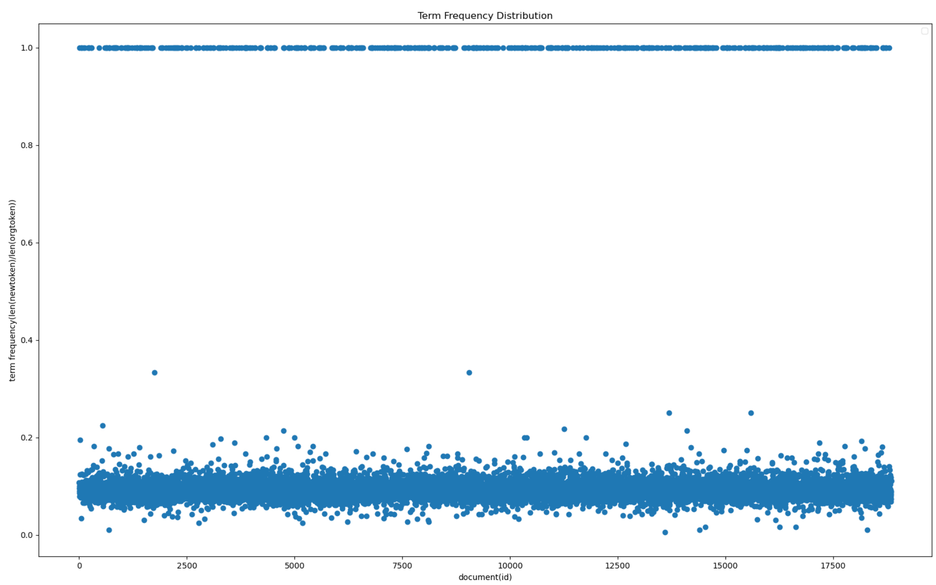

Figure 1: Term frequency distribution.

0.2.2 Build dictionary

We build a dictionary based on the tokenization of dataset. By using function filter-extremes(no-below, no-above, keep-n), we first set no-below = 5, no-above = 0.5, this generated a dictionary with feature dimension 24759, we define this dictionary as Vocab-v1. Then we set no-below = 5, no-above = 0.5, keep-n = 2000, this generated a dictionary with feature dimension 2000, we define this dictionary as Vocab-v2.

0.2.3 Generate feature embedding models

BOW We use sklearn built in package: CountVectorizer to generate BOW of dataset.

TF-IDF We use gensim TfidfModel, the input is a bow vector generated by Vocab-v1. Topic distribution visualization by TSNE in Figure 2. Use Vocab-v2, the topic distribution visualization by TSNE in Figure 3.

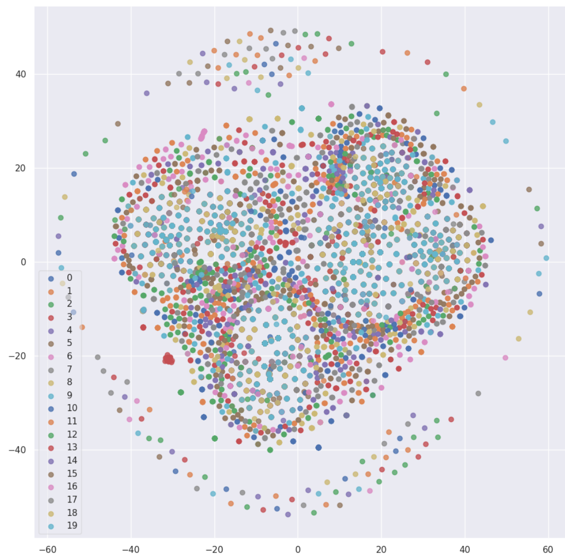

Figure 2: Topic distribution using TF-IDF model with with Vocab-v1.

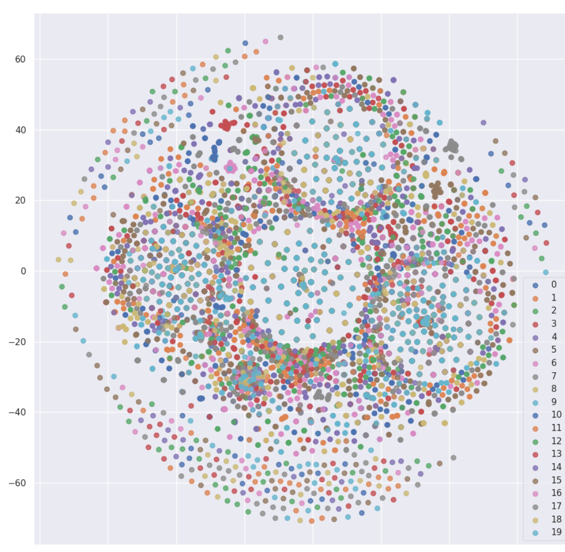

Figure 3: Topic distribution using TF-IDF model with Vocab-v2.

LDA We set TopicNUm = 10, eval-every = 5. Topic distribution visualization by pyLDAvis in Figure 4, TSNE in Figure 5.

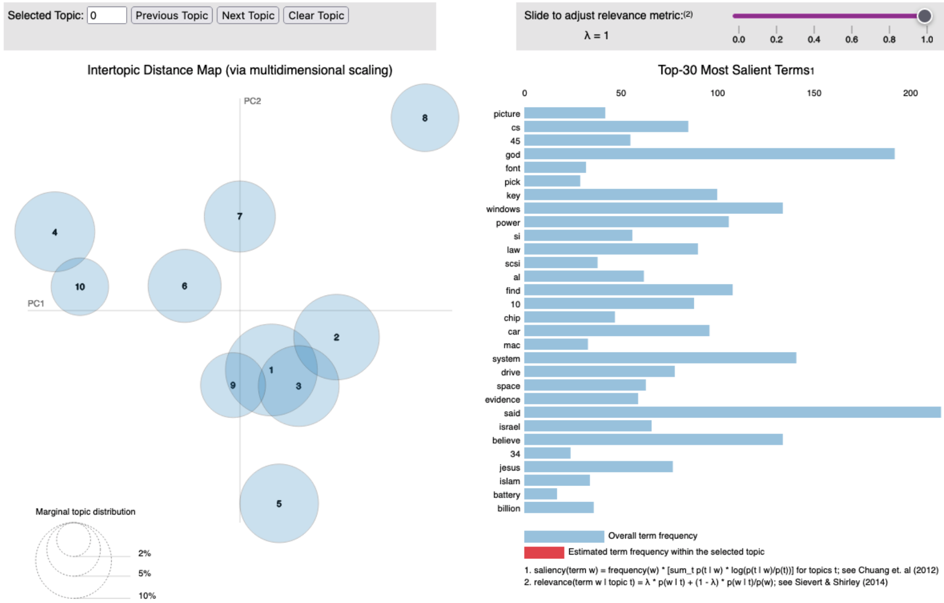

Figure 4: Topic distribution using LDA model.

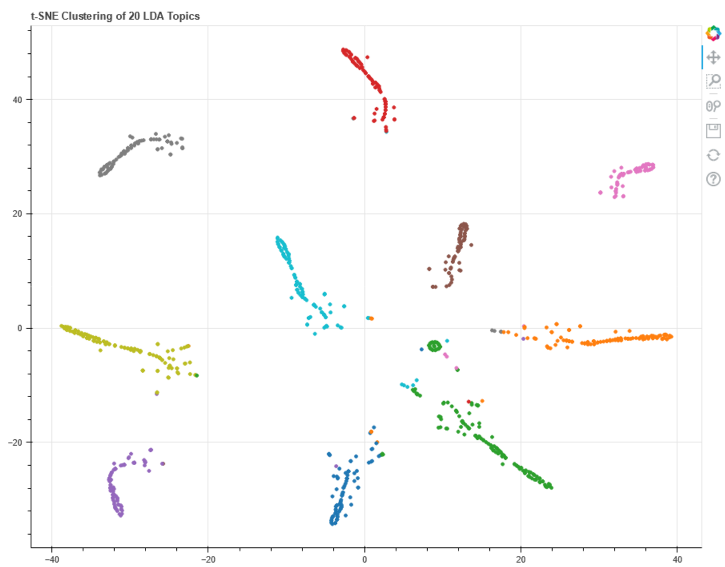

Figure 5: Topic distribution using LDA model.

Word2Vec We set vector-size = 100, min-count = 3, epochs = 40. Train on Vocab-v1. The topic distribution visualization by 3D axes TSNE in Figure 6.

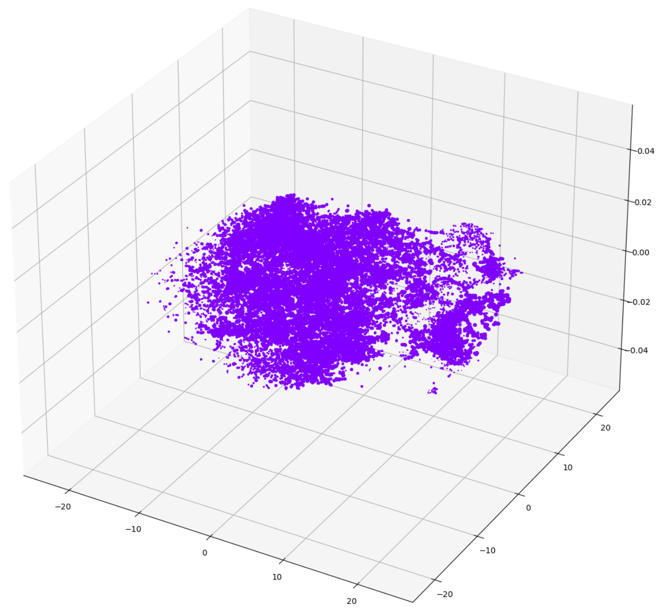

Figure 6: Topic distribution using Word2Vec model.

Doc2Vec We set vector-size = 100, min-count = 3, epochs = 40. Train on Vocab-v1. The topic distribution visualization by TSNE in Figure 7.

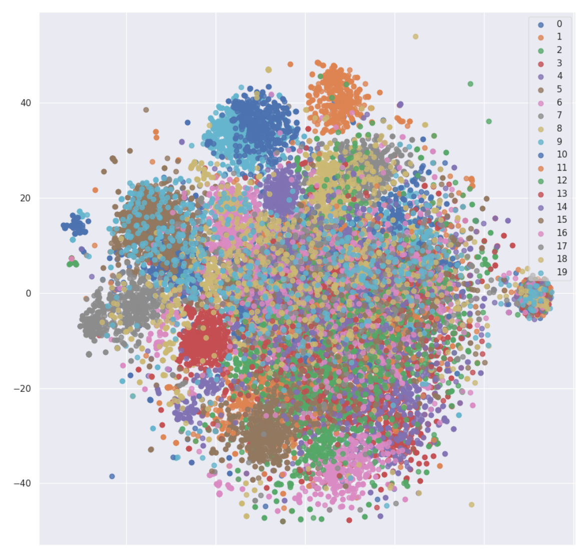

Figure 7: Topic distribution using Doc2Vec model.

0.2.4 K-means clustering with four different doc

In k-mean clustering, we use Vocab-v2, we set iteration step to 30. After iteration, we calculate the highest NMI score for each doc. See NMI results in below table .

NMI Results				
Method	BOW	TF-IDF	LDA	Doc2Vec
NMI	0.016	0.269	0.226	0.316

Table 1: NMI table.

0.2.5 Visualization of K-means clusters

BOW Figure 8.

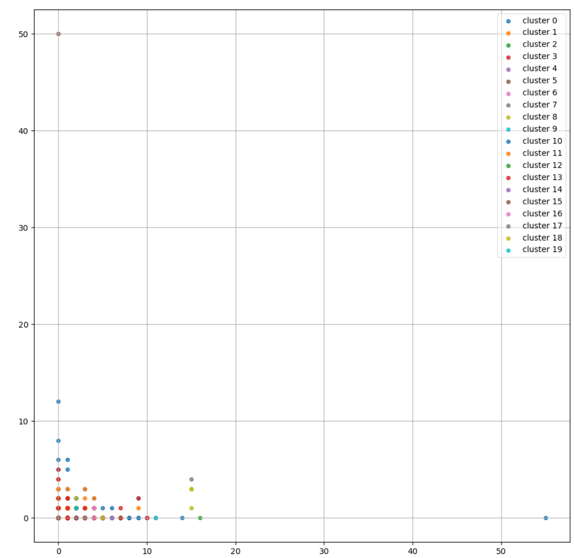

Figure 8: K-mean cluster visualization use BOW.

TF-IDF Figure 9.

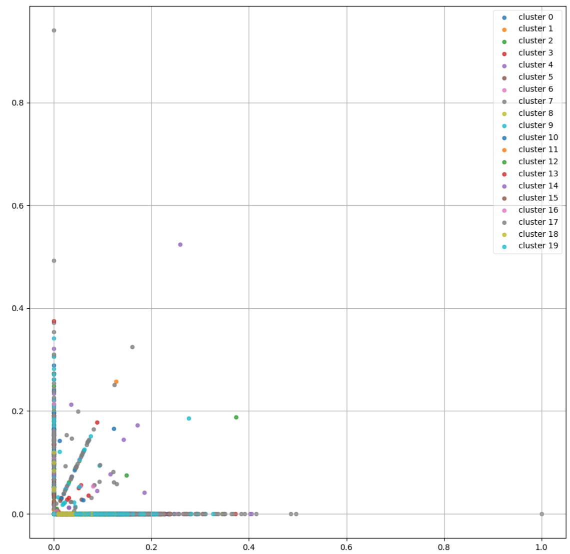

Figure 9: K-mean cluster visualization use TF-IDF.

LDA Figure 10.

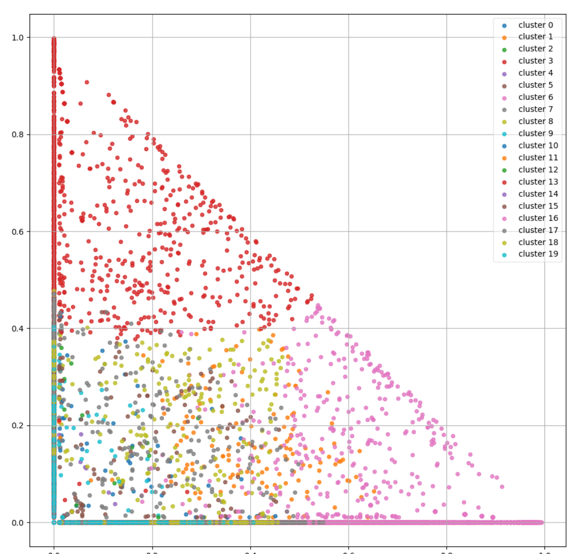

Figure 10: K-mean cluster visualization use LDA.

Doc2Vec Figure 11.

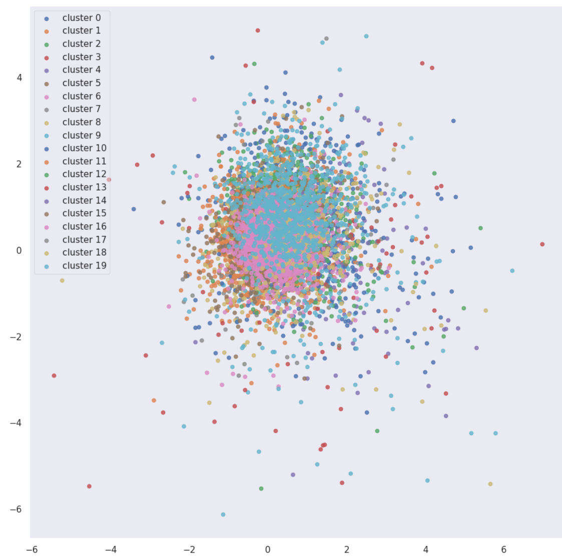

Figure 11: K-mean cluster visualization use Doc2Vec.

0.3 Comparison experiment

In comparison experiment, we observe the k-mean cluster with Doc2Vec model use Vocab-v1 Figure 11 vs. Vocab-v2 Figure 12. We observe the NMI of vocab-v1 is better then vocab-v2. From the cluster, we could see vocab-v1 cluster the document more accurate.

NMI Results		
Dictionary	Vocab-v1	Vocab-v2
Feature Dim	24759	2000
NMI	0.316	0.16

Table 2: NMI table.

0.4 Further Task

We proposed one supervised classification task upon the documents. In this part, we trained a SVM classifier based on TF-IDF as feature embedding. For dataset, use 20newsgroups', we split the whole dataset into train-set and validation-set, train=set includes 11314 documents and validation-set include 7532 documents. The final accuracy score we got for this topic classification task is 0.78. We output the accuracy for each topic, the result shows in Figure 13.

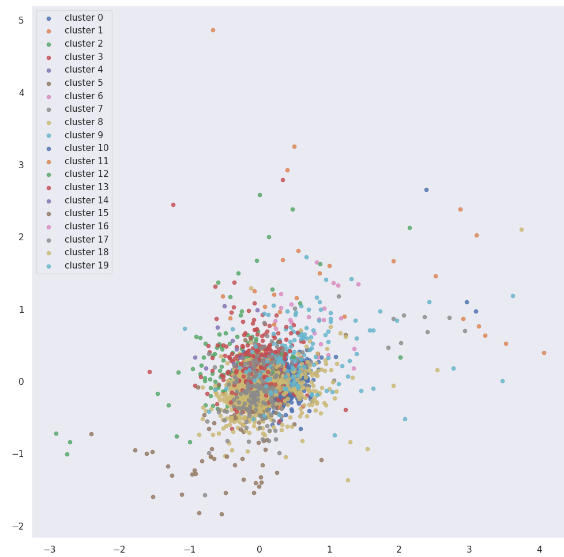

Figure 12: K-mean cluster visualization use Doc2Vec using Vocab-v2.

Accuracy = 0.7817312798725438

	precision	recall	f1-score	support
alt.atheism	0.72	0.51	0.60	319
comp.graphics	0.76	0.76	0.76	389
comp.os.ms-windows.misc	0.70	0.79	0.74	394
comp.sys.ibm.pc.hardware	0.75	0.65	0.70	392
comp.sys.mac.hardware	0.73	0.77	0.75	385
comp.windows.x	0.81	0.73	0.77	395
misc.forsale	0.80	0.90	0.85	390
rec.autos	0.90	0.80	0.84	396
rec.motorcycles	0.92	0.93	0.93	398
rec.sport.baseball	0.89	0.87	0.88	397
rec.sport.hockey	0.84	0.98	0.91	399
sci.crypt	0.81	0.92	0.86	396
sci.electronics	0.73	0.61	0.66	393
sci.med	0.80	0.88	0.84	396
sci.space	0.76	0.93	0.84	394
soc.religion.christian	0.65	0.88	0.75	398
talk.politics.guns	0.66	0.83	0.74	364
talk.politics.mideast	0.85	0.91	0.88	376
talk.politics.misc	0.84	0.48	0.61	310
talk.religion.misc	0.77	0.16	0.26	251
accuracy			0.78	7532
macro avg	0.78	0.76	0.76	7532
weighted avg	0.79	0.78	0.77	7532

Figure 13: Classification accuracy.