```
import pandas as pd

df = pd.read_pickle('consumer_complaint_dataset.data', compression='gzip')
```

```
df
```

	topic	input
0	Debt collection	transworld systems inc. \nis trying to collect...
1	Credit reporting, credit repair services, or o...	I would like to request the suppression of the...
2	Debt collection	Over the past 2 weeks, I have been receiving e...
3	Credit reporting, credit repair services, or o...	I HAD FILED WITH CFPB ON XX/XX/XXXX19 TO HAVE ...
4	Credit reporting, credit repair services, or o...	I have several accounts that the balance is in...
...	...	...
492250	Consumer Loan	I was on automatic payment for my car loan. In...
492251	Debt collection	I recieved a collections call from an unknown ...
492252	Mortgage	On XXXX XXXX, 2015, I contacted XXXX XXXX, who...
492253	Mortgage	I can not get from chase who services my mortg...
492254	Credit card	I made a payment to CITI XXXX Credit Card on X...

492255 rows × 2 columns

```
# Select only 10,000 rows from the dataset by randomly sampling the dataset
df = df.sample(n=10000, random_state=1)
```

```
df
```

	topic	input
351900	Mortgage	I have REPEATEDLY complained that Bank of Amer...
52106	Debt collection	To whom it my concern, the purpose of this com...
244147	Credit reporting, credit repair services, or o...	Last year after the whole XXXX data breach I d...
39437	Checking or savings account	I had unauthorized debits made when my card wa...
4840	Credit reporting, credit repair services, or o...	TryingtoremovedisputeswithExperianandXXXXisbey...
...	...	...
247115	Bank account or service	I opened up a bank account under an offer for ...
398824	Credit reporting, credit repair services, or o...	I HAVE ASK FOR PROF FROM XXXX THAT THESE CHARG...
267384	Bank account or service	I had sufficient funds in the bank to make a t...
49222	Credit reporting, credit repair services, or o...	three inaccurate and fraud accounts have bee r...
149530	Debt collection	I I am receiving multiple calls from the below...

10000 rows × 2 columns

```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
# import the stopwords from nltk
from nltk.corpus import stopwords
import re
stop_words = set(stopwords.words('english'))

# also the data that is "XXXX" the model will not be able to learn anything from it
# so we can remove it from the dataset

sent = [row.split(" ") for row in df['input']]
sent = [[word for word in row if word not in stop_words and not re.match(r'XXXX|XXX|XXXX,', word)] for row in sent]
```

```
sent
```

```
[['I',
  'REPEATEDLY',
  'complained',
  'Bank',
  'America',
  'sent',
  'mortgage',
  'statement',
  'in,',
  'now,',
  '3',
  'years.',
  'They',
  'constantly',
  'confirm',
  'coorect',
  'mailing',
  'address',
  'CA',
  'insist',
  'mailing',
  'statements',
  'there,',
  'I',
  'continue',
  'receive',
  'them.',
  'I',
  'idea',
  'monthly',
  'payment',
  'changed.',
  'They',
  'also',
  'continue',
  'post',
  'late',
  'fees',
  'THEIR',
  'error',
  'taking',
  'mortgage',
  'payment',
  'applying',
  'credit',
  'card',
  'instead',
  'mortgage',
  'threatening',
  'destroying',
  'credit',
  'foreclosure.'],
 ['To',
  'concern,',
  'purpose',
  'complaint',
  'inform',
```

```
# from gensim import models
from gensim.models import Word2Vec
model = Word2Vec(sent, min_count=1,vector_size= 300,workers=5, window=10, sg = 1, epochs=100)
# model = Word2Vec(sent, min_count=1,size= 300,workers=5, window=10, sg = 1, iter=100)
```

```
model.wv['Enclosure']
```

```
array([-0.3021819 , -0.14017399, -0.27313668, -0.23139025, -0.04157692,
       -0.27394965,  0.03283772,  0.5484666 ,  0.2697015 , -0.1195562 ,
        0.24115093, -0.56135565, -0.40819144,  0.29205728, -0.2977418 ,
        0.00873179,  0.19869535,  0.03807858, -0.39437145,  0.87043333,
       -0.08459534, -0.19997433,  1.0388054 , -0.3925643 ,  0.5248623 ,
        0.1826136 , -0.28616858,  0.9274716 ,  0.00732129,  0.39749578,
        0.35168818, -0.4739805 , -0.30911896,  0.37264708,  0.8799671 ,
        0.3131436 , -0.15552446, -0.16495895, -0.45834878,  0.082499  ,
       -0.06684317,  0.22369039, -0.39944938, -0.36624652,  0.8473149 ,
        0.22994816, -0.28498474, -0.29227257, -0.36101672, -0.6365251 ,
       -0.7491646 , -0.23278995, -0.46965614, -0.52318174, -0.0876718 ,
       -0.4109494 , -0.05137783,  0.2679002 ,  0.02189614, -0.23213878,
       -0.38035992, -0.22803228,  0.70444036, -0.19825375,  0.44317308,
        0.08824597, -0.36713278,  0.84456974,  0.09252046, -0.24479967,
       -0.02169257,  0.8225676 ,  0.65739083, -0.25112012, -0.3272639 ,
        0.00733655, -0.08657184,  0.38042295,  0.01059202,  0.55797696,
       -0.15204085, -0.45522842,  0.62778383,  0.21675944,  0.22932689,
       -0.18215898, -0.43433255, -0.7698026 ,  0.39787042, -0.291738  ,
        0.24044757,  0.13127406,  0.23368911, -0.3973821 ,  0.29664356,
        0.2249372 ,  0.8581434 ,  0.569413  ,  0.01826031, -0.2789073 ,
        0.18966694,  0.13569027, -0.73494405, -0.40346664, -0.29437262,
       -0.87629974,  0.20279397,  0.03282205, -0.23650527,  0.12032781,
       -0.0047443 ,  0.20627815,  0.20073643,  0.4814705 , -0.27837706,
        0.34044054,  0.00280263, -0.18695274,  0.13826725, -0.84857774,
 0.5119249 ,  0.14763322, -0.33735165,  0.39431858,  0.46163335,
 0.28811616,  0.22619277, -0.19928263,  0.20121452, -0.68916947,
 0.05533224,  1.2473994 ,  0.44042224,  0.1640861 ,  0.15115128,
 0.34992403, -0.4228823 , -0.30797195, -0.25271237,  0.24867773,
 0.1055171 ,  0.16599607, -0.2460845 ,  0.01575498, -0.01619238,
 1.0559137 ,  0.29218876, -0.12204394,  0.04803645, -0.55359423,
-0.75125676, -0.13887818, -0.5139978 , -0.49061444,  0.4070972 ,
-0.3218644 , -0.3975533 , -0.891875  ,  0.00810808,  0.39773944,
 0.34920427,  0.35193568, -0.09801937,  0.10198489,  0.41960666,
 0.36309698,  0.386097  , -0.39001277,  0.36976078,  0.7711474 ,
 0.4192007 ,  0.22901836, -0.6922588 ,  0.55820775,  0.45167497,
 0.16081597, -0.56134814, -0.12598598,  0.6943582 , -0.00131842,
-0.40175775,  0.0555244 , -0.06158944, -0.35664323, -0.38794303,
-0.6805447 , -0.02855665,  0.02751187,  0.02822603,  0.44729894,
-0.13275047, -0.53574777, -0.30593458, -0.04771127,  0.37242833,
-0.7070671 , -0.26705062, -0.5749381 ,  0.5403602 ,  0.62856126,
 0.6270973 ,  0.31230637,  0.7221946 ,  0.56266916,  0.7612729 ,
-0.34842986,  0.09832946,  0.32511535, -0.7754972 , -0.12901978,
-0.16979729, -0.27213556, -0.13902323, -0.5662872 ,  0.7651976 ,
-0.03925564, -0.26436713, -0.9100574 , -0.17112741, -0.35267106,
-0.22730926,  0.19155364, -0.31167275, -0.00419993, -0.6516782 ,
 0.43162757, -0.5988139 ,  0.2909497 ,  0.07919698,  0.4888539 ,
 0.3133948 , -0.6558874 , -0.7733678 ,  0.07887156,  0.24789687,
-0.47308612, -0.25288478,  0.00863167,  0.0573547 ,  0.29194206,
 0.22288765,  0.11804797,  0.36557806,  0.53140897,  0.4185697 ,
-0.30489123, -0.6026921 ,  0.21250913, -0.2620556 ,  0.72313386,
-0.68135434, -0.26563495, -0.17185506,  0.31005883, -0.54276186,
-0.13164806,  0.20119976,  0.14010249, -0.42676452, -0.38693935,
-0.33924788,  0.01398893,  0.18981454, -0.49461976,  0.27008662,
 0.24647015,  0.30019674,  0.4322413 ,  0.07724059,  0.498837  ,
-0.16621563, -0.12348451,  0.3077014 , -0.9340087 , -0.760981  ,
 0.5048825 ,  1.0141083 ,  0.30202916, -1.0501078 , -0.20811984,
 0.03589093, -0.01423965, -0.28044277,  0.09647406,  0.42425478,
```

```
debt_similar = model.wv.most_similar('debt')[:5]
print("Debt Similar : ")
print(debt_similar)
```

```
Debt Similar :
[('debt.', 0.6274077892303467), ('collector', 0.6012453436851501), ('debt,', 0.5555453300476074), ('collection', 0.5062741637229919
```

```
collection_similar = model.wv.most_similar('collection')[:5]
print("Collection Similar : ")
print(collection_similar)
```

```
Collection Similar :
[('agency', 0.6185796856880188), ('agency.', 0.559133768081665), ('debt', 0.5062741041183472), ('in-turn', 0.47768861055374146), ('t
```

```
risk_similar = model.wv.most_similar('risk')[:5]
print("Risk Similar : ")
print(risk_similar)
```

```
Risk Similar :
[('theft/fraud.', 0.5118493437767029), ('\n\nThanks.', 0.47840505838394165), ('repetition,', 0.44698917865753174), ('www.equifaxsecu
```

```
!pip install scikit-learn matplotlib
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.3.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.7.1)
Requirement already satisfied: numpy<2.0,>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.26.4)
Requirement already satisfied: scipy>=1.5.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.13.1)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.5.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.3.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.53.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.7)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.1.4)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
```

```
from sklearn.manifold import TSNE
import matplotlib.pyplot as plt
import numpy as np

words = []
embeddings = []
```

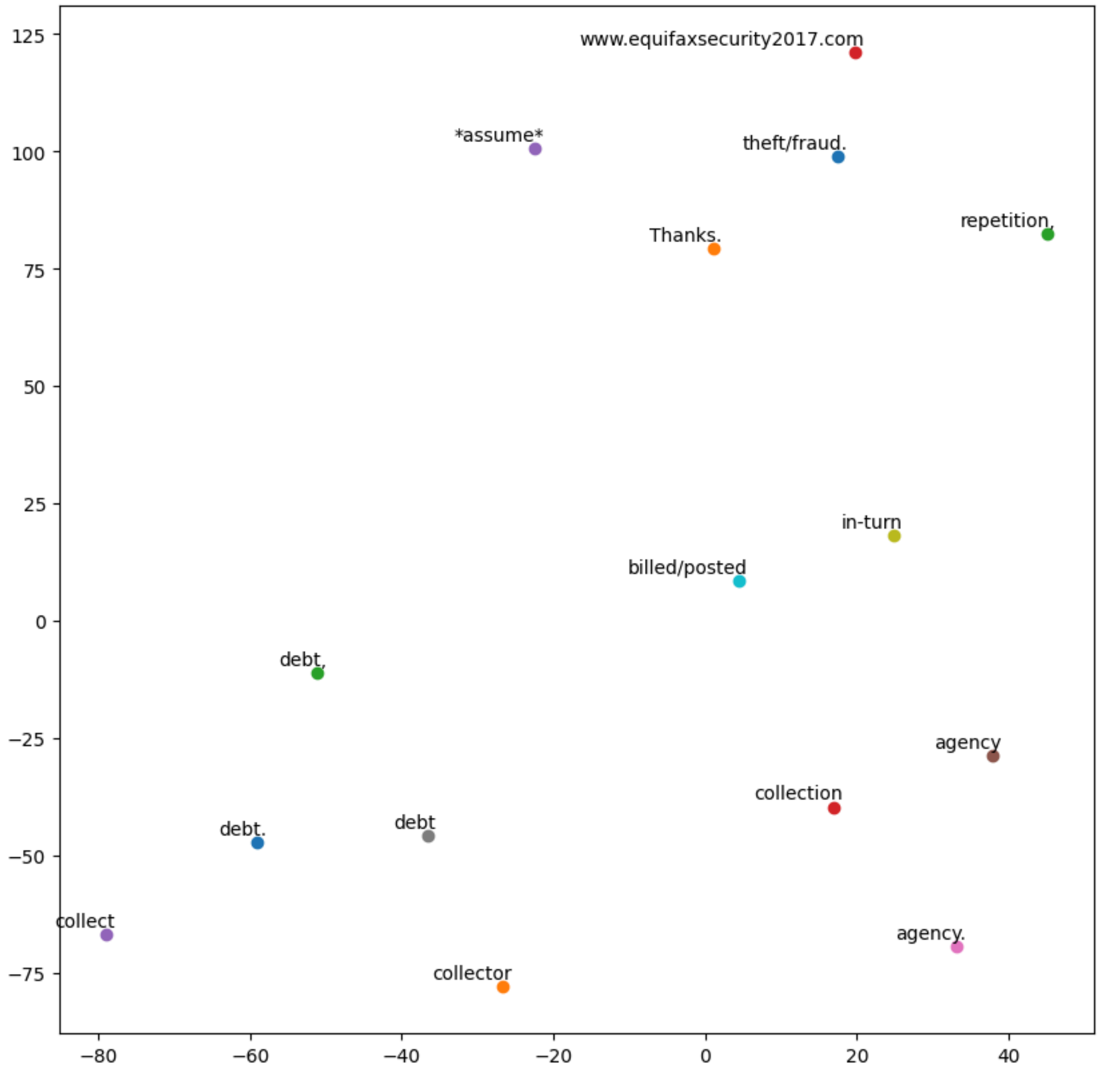
```
for word, similarity in debt_similar:
  words.append(word)
  embeddings.append(model.wv[word])

for word, similarity in collection_similar:
  words.append(word)
  embeddings.append(model.wv[word])

for word, similarity in risk_similar:
  words.append(word)
  embeddings.append(model.wv[word])


# Set perplexity to a value less than the number of samples (15 in this case)
tsne = TSNE(n_components=2, random_state=0, perplexity=5)  # Changed perplexity to 5
np.set_printoptions(suppress=True)
embeddings_array = np.array(embeddings)
embeddings_2d = tsne.fit_transform(embeddings_array)

plt.figure(figsize=(10, 10))
for i, label in enumerate(words):
    x, y = embeddings_2d[i, :]
    plt.scatter(x, y)
    plt.annotate(label, xy=(x, y), xytext=(5, 2), textcoords='offset points',
                 ha='right', va='bottom')
plt.show()
```



นายศวิษฐ์ โกสียอัมพร 65070507238