

Amazon Fine Food Reviews Analysis

Data Source: <https://www.kaggle.com/snap/amazon-fine-food-reviews>

EDA: <https://nycdatascience.com/blog/student-works/amazon-fine-foods-visualization/>

The Amazon Fine Food Reviews dataset consists of reviews of fine foods from Amazon.

Number of reviews: 568,454

Number of users: 256,059

Number of products: 74,258

Timespan: Oct 1999 - Oct 2012

Number of Attributes/Columns in data: 10

Attribute Information:

1. Id
2. ProductId - unique identifier for the product
3. UserId - unique identifier for the user
4. ProfileName
5. HelpfulnessNumerator - number of users who found the review helpful
6. HelpfulnessDenominator - number of users who indicated whether they found the review helpful or not
7. Score - rating between 1 and 5
8. Time - timestamp for the review
9. Summary - brief summary of the review
10. Text - text of the review

Objective:

Given a review, determine whether the review is positive (Rating of 4 or 5) or negative (rating of 1 or 2).

[Q] How to determine if a review is positive or negative?

[Ans] We could use the Score/Rating. A rating of 4 or 5 could be considered a positive review. A review of 1 or 2 could be considered negative. A review of 3 is neutral and ignored. This is an approximate and proxy way of determining the polarity (positivity/negativity) of a review.

Loading the data

The dataset is available in two forms

1. .csv file
2. SQLite Database

In order to load the data, We have used the SQLITE dataset as it easier to query the data and visualise the data efficiently.

Here as we only want to get the global sentiment of the recommendations (positive or negative), we will purposefully ignore all Scores equal to 3. If the score id above 3, then the recommendation will be set to "positive". Otherwise, it will be set to "negative".

```
In [1]: %matplotlib inline
import warnings
warnings.filterwarnings("ignore")

import sqlite3
import pandas as pd
import numpy as np
import nltk
import string
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn.metrics import roc_curve, auc
from nltk.stem.porter import PorterStemmer

import re
# Tutorial about Python regular expressions: https://pymotw.com/2/re/
import string
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer

from gensim.models import Word2Vec
from gensim.models import KeyedVectors
```

```
import pickle

from tqdm import tqdm
import os
```

[1]. Reading Data

```
In [2]: # using the SQLite Table to read data.
con = sqlite3.connect('database.sqlite')
#filtering only positive and negative reviews i.e.
# not taking into consideration those reviews with Score=3
# SELECT * FROM Reviews WHERE Score != 3 LIMIT 500000, will give top 500000 data points
# you can change the number to any other number based on your computing power

# filtered_data = pd.read_sql_query(""" SELECT * FROM Reviews WHERE Score != 3 LIMIT 500000""", con)
# for tsne assignment you can take 5k data points

filtered_data = pd.read_sql_query(""" SELECT * FROM Reviews WHERE Score != 3 LIMIT 10000""", con)

# Give reviews with Score>3 a positive rating, and reviews with a score<3 a negative rating.
def partition(x):
    if x < 3:
        return 'Negative'
    return 'Positive'

#changing reviews with score less than 3 to be positive and vice-versa
actualScore = filtered_data['Score']
positiveNegative = actualScore.map(partition)
filtered_data['Score'] = positiveNegative
print("Number of data points in our data", filtered_data.shape)
filtered_data.head(3)
```

Number of data points in our data (10000, 10)

Out[2]:

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	

1 2 B00813GRG4 A1D87F6ZCVE5NK dll pa 0

2 3 B000LQOCH0 ABXLMWJIXXAIN Natalia
Corres
"Natalia
Corres" 1

```
In [3]: display = pd.read_sql_query("""
SELECT UserId, ProductId, ProfileName, Time, Score, Text, COUNT(*)
FROM Reviews
GROUP BY UserId
HAVING COUNT(*)>1
""", con)
```

```
In [4]: print(display.shape)
display.head()
```

(80668, 7)

Out[4]:

	UserId	ProductId	ProfileName	Time	Score	Text	COUNT(*)
0	#oc-R115TNMSPFT9I7	B007Y59HVM	Breyton	1331510400	2	Overall its just OK when considering the price...	2
1	#oc-R11D9D7SHXIJB9	B005HG9ET0	Louis E. Emory "hoppy"	1342396800	5	My wife has recurring extreme muscle spasms, u...	3
2	#oc-R11DNU2NBKQ23Z	B007Y59HVM	Kim Cieszykowski	1348531200	1	This coffee is horrible and unfortunately not ...	2
3	#oc-R11O5J5ZVQE25C	B005HG9ET0	Penguin Chick	1346889600	5	This will be the bottle that you grab from the...	3
4	#oc-R12KPBODL2B5ZD	B007OSBE1U	Christopher P. Presta	1348617600	1	I didnt like this coffee. Instead of telling y...	2

```
In [5]: display[display['UserId']=='AZY10LLTJ71NX']
```

```
Out[5]:
```

	UserId	ProductId	ProfileName	Time	Score	Text	COUNT(*)
80638	AZY10LLTJ71NX	B006P7E5ZI	undertheshrine "undertheshrine"	1334707200	5	I was recommended to try green tea extract to ...	5

```
In [6]: display['COUNT(*)'].sum()
```

```
Out[6]: 393063
```

Exploratory Data Analysis

[2] Data Cleaning: Deduplication

It is observed (as shown in the table below) that the reviews data had many duplicate entries. Hence it was necessary to remove duplicates in order to get unbiased results for the analysis of the data. Following is an example:

```
In [7]: display= pd.read_sql_query("""
SELECT *
FROM Reviews
WHERE Score != 3 AND UserId="AR5J8UI46CURR"
ORDER BY ProductID
""", con)
display.head()
```

```
Out[7]:
```

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenom
0	78445	B000HDL1RQ	AR5J8UI46CURR	Geetha Krishnan	2	
1	138317	B000HDOPYC	AR5J8UI46CURR	Geetha Krishnan	2	
2	138277	B000HDOPYM	AR5J8UI46CURR	Geetha Krishnan	2	

3	73791	B000HDOPZG	AR5J8UI46CURR	Geetha Krishnan	2
---	-------	------------	---------------	--------------------	---

4	155049	B000PAQ75C	AR5J8UI46CURR	Geetha Krishnan	2
---	--------	------------	---------------	--------------------	---

As can be seen above the same user has multiple reviews of the with the same values for HelpfulnessNumerator, HelpfulnessDenominator, Score, Time, Summary and Text and on doing analysis it was found that

ProductId=B000HDOPZG was Loacker Quadratini Vanilla Wafer Cookies, 8.82-Ounce Packages (Pack of 8)

ProductId=B000HDL1RQ was Loacker Quadratini Lemon Wafer Cookies, 8.82-Ounce Packages (Pack of 8) and so on

It was inferred after analysis that reviews with same parameters other than ProductId belonged to the same product just having different flavour or quantity. Hence in order to reduce redundancy it was decided to eliminate the rows having same parameters.

The method used for the same was that we first sort the data according to ProductId and then just keep the first similar product review and delete the others. for eg. in the above just the review for ProductId=B000HDL1RQ remains. This method ensures that there is only one representative for each product and deduplication without sorting would lead to possibility of different representatives still existing for the same product.

```
In [8]: #Sorting data according to ProductId in ascending order
sorted_data=filtered_data.sort_values('ProductId', axis=0, ascending=True, inplace=False, kind='quicksort', na_position='last')
```

```
In [9]: #Deduplication of entries
final=sorted_data.drop_duplicates(subset={"UserId","ProfileName","Time","Text"}, keep='first', inplace=False)
final.shape
```

```
Out[9]: (9564, 10)
```

```
In [10]: #Checking to see how much % of data still remains
         (final['Id'].size*1.0)/(filtered_data['Id'].size*1.0)*100
```

Out[10]: 95.64

Observation:- It was also seen that in two rows given below the value of HelpfulnessNumerator is greater than HelpfulnessDenominator which is not practically possible hence these two rows too are removed from calculations

```
In [11]: display= pd.read_sql_query("""
         SELECT *
         FROM Reviews
         WHERE Score != 3 AND Id=44737 OR Id=64422
         ORDER BY ProductID
         """, con)

         display.head()
```

Out[11]:

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenom
0	64422	B000MIDROQ	A161DK06JJMCYF	J. E. Stephens "Jeanne"	3	
1	44737	B001EQ55RW	A2V0I904FH7ABY	Ram	3	

```
In [12]: final=final[final.HelpfulnessNumerator<=final.HelpfulnessDenominator]
```

```
In [13]: #Before starting the next phase of preprocessing lets see the number of entries left
         print(final.shape)

         #How many positive and negative reviews are present in our dataset?
         final['Score'].value_counts()

         (9564, 10)
```

Out[13]: Positive 7976

Negative 1588
Name: Score, dtype: int64

[3]. Text Preprocessing.

Now that we have finished deduplication our data requires some preprocessing before we go on further with analysis and making the prediction model.

Hence in the Preprocessing phase we do the following in the order below:-

1. Begin by removing the html tags
2. Remove any punctuations or limited set of special characters like , or . or # etc.
3. Check if the word is made up of english letters and is not alpha-numeric
4. Check to see if the length of the word is greater than 2 (as it was researched that there is no adjective in 2-letters)
5. Convert the word to lowercase
6. Remove Stopwords
7. Finally Snowball Stemming the word (it was observed to be better than Porter Stemming)

After which we collect the words used to describe positive and negative reviews

```
In [14]: # printing some random reviews
sent_0 = final['Text'].values[0]
print(sent_0)
print("="*50)

sent_1000 = final['Text'].values[1000]
print(sent_1000)
print("="*50)

sent_1500 = final['Text'].values[1500]
print(sent_1500)
print("="*50)

sent_4900 = final['Text'].values[4900]
print(sent_4900)
print("="*50)

sent_7500 = final['Text'].values[7500]
print(sent_7500)
print("="*50)
```

We have used the Victor fly bait for 3 seasons. Can't beat it. Great product!

=====

15 month old loves to eat them on the go! They seem great for a healthy, quick, and easy snack!

=====

These chips are truly amazing. They have it all. They're light, crisp, great tasting, nice texture, AND they're all natural... AND low in fat and sodium! Need I say more? I recently bought a bag of them at a regular grocery store, and couldn't believe my taste buds. That's why I excited why I saw them here on Amazon, and decided to buy a case!

=====

These tablets definitely made things sweeter -- like lemons, limes, and grapefruit. But it wasn't to the point of sheer amazement. They also had an interesting effect on cheeses and vinegar, but still did virtually nothing for beer and wine. The tablets are a bit pricey but they do work. If you've got extra money, sure, give them a try, but if you're looking for some amazing way to get your kids to eat broccoli or something along those lines then this is not the answer. Fun experiment, but not life-changing. :)

=====

This one will not disappoint! I got this yesterday and my husband and I tried it this morning with breakfast and we both loved it. I have gotten many different kinds of the k-cups to try but after trying this one, I think I could be happy with just this one and black tiger for those mornings when I want a really strong cup of coffee. The rodeo drive blend is delicious with a nice round feel on the palate and an interesting flavor on the finish with just a hint of smokiness. Perfectly balanced and absolutely wonderful! Plus, I have to admit...I love the label. The different color looks very appealing on my carousel with the other k-cups.

=====

```
In [15]: # remove urls from text python: https://stackoverflow.com/a/40823105/4084039
sent_0 = re.sub(r"http\S+", "", sent_0)
sent_1000 = re.sub(r"http\S+", "", sent_1000)
sent_1500 = re.sub(r"http\S+", "", sent_1500)
sent_4900 = re.sub(r"http\S+", "", sent_4900)

print(sent_0)
print("="*50)
print(sent_1000)
print("="*50)
print(sent_1500)
print("="*50)
print(sent_4900)
print("="*50)
print(sent_7500)
```

We have used the Victor fly bait for 3 seasons. Can't beat it. Great product!

=====

15 month old loves to eat them on the go! They seem great for a healthy, quick, and easy snack!

=====

These chips are truly amazing. They have it all. They're light, crisp, great tasting, nice texture, AND they're all natural... AND low in fat and sodium! Need I say more? I recently bought a bag of them at a regular grocery store, and couldn't believe my taste buds. That's why I excited why I saw them here on Amazon, and decided to buy a case!

=====

These tablets definitely made things sweeter -- like lemons, limes, and grapefruit. But it wasn't to the point of sheer amazement. They also had an interesting effect on cheeses and vinegar, but still did virtually nothing for beer and wine. The tablets are a bit pricey but they do work. If you've got extra money, sure, give them a try, but if you're looking for some amazing way to get your kids to eat broccoli or something along those lines then this is not the answer. Fun experiment, but not life-changing. :)

=====

This one will not disappoint! I got this yesterday and my husband and I tried it this morning with breakfast and we both loved it. I have gotten many different kinds of the k-cups to try but after trying this one, I think I could be happy with just this one and black tiger for those mornings when I want a really strong cup of coffee. The rodeo drive blend is delicious with a nice round feel on the palate and an interesting flavor on the finish with just a hint of smokiness. Perfectly balanced and absolutely wonderful! Plus, I have to admit...I love the label. The different color looks very appealing on my carousel with the other k-cups.

```
In [16]: # https://stackoverflow.com/questions/16206380/python-beautifulsoup-how-to-remove-all-tags-from-an-element
from bs4 import BeautifulSoup

soup = BeautifulSoup(sent_0, 'lxml')
text = soup.get_text()
print(text)
print("="*50)

soup = BeautifulSoup(sent_1000, 'lxml')
text = soup.get_text()
print(text)
print("="*50)

soup = BeautifulSoup(sent_1500, 'lxml')
text = soup.get_text()
print(text)
print("="*50)
```

```
soup = BeautifulSoup(sent_4900, 'lxml')
text = soup.get_text()
print(text)

soup = BeautifulSoup(sent_7500, 'lxml')
text = soup.get_text()
print(text)
```

We have used the Victor fly bait for 3 seasons. Can't beat it. Great product!

=====

15 month old loves to eat them on the go! They seem great for a healthy, quick, and easy snack!

=====

These chips are truly amazing. They have it all. They're light, crisp, great tasting, nice texture, AND they're all natural... AND low in fat and sodium! Need I say more? I recently bought a bag of them at a regular grocery store, and couldn't believe my taste buds. That's why I excited why I saw them here on Amazon, and decided to buy a case!

=====

These tablets definitely made things sweeter -- like lemons, limes, and grapefruit. But it wasn't to the point of sheer amazement. They also had an interesting effect on cheeses and vinegar, but still did virtually nothing for beer and wine. The tablets are a bit pricey but they do work. If you've got extra money, sure, give them a try, but if you're looking for some amazing way to get your kids to eat broccoli or something along those lines then this is not the answer. Fun experiment, but not life-changing. :)

This one will not disappoint! I got this yesterday and my husband and I tried it this morning with breakfast and we both loved it. I have gotten many different kinds of the k-cups to try but after trying this one, I think I could be happy with just this one and black tiger for those mornings when I want a really strong cup of coffee. The rodeo drive blend is delicious with a nice round feel on the palate and an interesting flavor on the finish with just a hint of smokiness. Perfectly balanced and absolutely wonderful! Plus, I have to admit...I love the label. The different color looks very appealing on my carousel with the other k-cups.

```
In [17]: def cleanhtml(sentence): #function to clean the word of any html-tags
          cleanr = re.compile('<.*?>')
          cleantext = re.sub(cleanr, ' ', sentence)
          return cleantext

          def cleanpunc(sentence): #function to clean the word of any punctuation or special characters
              cleaned = re.sub(r'[?!|\\\'|"|#]', r'', sentence)
              cleaned = re.sub(r'[:|-|.|,|)|(|\\|/]', r' ', cleaned)
              return cleaned
```

In [18]: [#https://stackoverflow.com/a/47091490/4084039](https://stackoverflow.com/a/47091490/4084039)

```
import re

def decontracted(phrase):
    # specific
    phrase = re.sub(r"won't", "will not", phrase)
    phrase = re.sub(r"can't", "can not", phrase)

    # general
    phrase = re.sub(r"n't", " not", phrase)
    phrase = re.sub(r"\ 're", " are", phrase)
    phrase = re.sub(r"\ 's", " is", phrase)
    phrase = re.sub(r"\ 'd", " would", phrase)
    phrase = re.sub(r"\ 'll", " will", phrase)
    phrase = re.sub(r"\ 't", " not", phrase)
    phrase = re.sub(r"\ 've", " have", phrase)
    phrase = re.sub(r"\ 'm", " am", phrase)
    return phrase
```

```
In [20]: sent_1500 = decontracted(sent_1500)
print(sent_1500)
print("="*50)

sent_1500 = cleanpunc(sent_1500)
print(sent_1500)
print("="*50)

sent_4900 = decontracted(sent_4900)
print(sent_4900)
print("="*50)

sent_4900 = cleanpunc(sent_4900)
print(sent_4900)
print("="*50)

sent_7500 = cleanpunc(sent_7500)
print(sent_7500)
print("="*50)
```

These chips are truly amazing They have it all They are light crisp great tasting nice texture AND they are all natural AND low in fat and sodium Need I say more I recently bought a bag of them at a regular grocery store and could not believe my taste buds That is why I excited why I saw them here on Amazon and decided to buy a case

=====

These chips are truly amazing They have it all They are light crisp great tasting nice texture AND they are all natural AND low in fat and sodi

um Need I say more I recently bought a bag of them at a regular grocery store and could not believe my taste buds That is why I excited why I saw them here on Amazon and decided to buy a case

=====

These tablets definitely made things sweeter -- like lemons, limes, and grapefruit. But it was not to the point of sheer amazement. They also had an interesting effect on cheeses and vinegar, but still did virtually nothing for beer and wine. The tablets are a bit pricey but they do work. If you have got extra money, sure, give them a try, but if you are looking for some amazing way to get your kids to eat broccoli or something along those lines then this is not the answer. Fun experiment, but not life-changing. :)

=====

These tablets definitely made things sweeter -- like lemons limes and grapefruit But it was not to the point of sheer amazement They also had an interesting effect on cheeses and vinegar but still did virtually nothing for beer and wine The tablets are a bit pricey but they do work If you have got extra money sure give them a try but if you are looking for some amazing way to get your kids to eat broccoli or something along those lines then this is not the answer Fun experiment but not life-changing

=====

This one will not disappoint I got this yesterday and my husband and I tried it this morning with breakfast and we both loved it I have gotten many different kinds of the k-cups to try but after trying this one I think I could be happy with just this one and black tiger for those mornings when I want a really strong cup of coffee The rodeo drive blend is delicious with a nice round feel on the palate and an interesting flavor on the finish with just a hint of smokiness Perfectly balanced and absolutely wonderful Plus I have to admit I love the label The different color looks very appealing on my carousel with the other k-cups

=====

```
In [22]: #remove words with numbers python: https://stackoverflow.com/a/18082370/4084039
sent_0 = re.sub("\S*\d\S*", "", sent_0).strip()
print(sent_0)

sent_0 = cleanpunc(sent_0)
print(sent_0)
print("="*50)
```

We have used the Victor fly bait for seasons. Can't beat it. Great product!

We have used the Victor fly bait for seasons Cant beat it Great product

=====

```
In [23]: #remove spacial character: https://stackoverflow.com/a/5843547/4084039
sent_1500 = re.sub('[^A-Za-z0-9]+', ' ', sent_1500)
```

```
print(sent_1500)
```

These chips are truly amazing They have it all They are light crisp great ta sting nice texture AND they are all natural AND low in fat and sodium Need I say more I recently bought a bag of them at a regular grocery store and coul d not belive my taste buds That is why I excited why I saw them here on Amaz on and decided to buy a case

```
In [24]: # https://gist.github.com/sebleier/554280
# we are removing the words from the stop words list: 'no', 'nor', 'not'
# <br /><br /> ==> after the above steps, we are getting "br br"
# we are including them into stop words list
# instead of <br /> if we have <br/> these tags would have revmoved in the 1st
step

stopwords= set(['br', 'the', 'i', 'me', 'my', 'myself', 'we', 'our', 'ours',
'ourselves', 'you', "you're", "you've", \
    "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he'
, 'him', 'his', 'himself', \
    'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'it
self', 'they', 'them', 'their', \
    'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 't
hat', "that'll", 'these', 'those', \
    'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have',
'has', 'had', 'having', 'do', 'does', \
    'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'becau
se', 'as', 'until', 'while', 'of', \
    'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into',
'through', 'during', 'before', 'after', \
    'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on',
'off', 'over', 'under', 'again', 'further', \
    'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'a
ll', 'any', 'both', 'each', 'few', 'more', \
    'most', 'other', 'some', 'such', 'only', 'own', 'same', 'so', 'tha
n', 'too', 'very', \
    's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "shoul
d've", 'now', 'd', 'll', 'm', 'o', 're', \
    've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn',
"didn't", 'doesn', "doesn't", 'hadn', \
    "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'm
a', 'mightn', "mightn't", 'mustn', \
    "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shoul
dn't", 'wasn', "wasn't", 'weren', "weren't", \
    'won', "won't", 'wouldn', "wouldn't"])
```

```
In [25]: from nltk.stem import SnowballStemmer
snow = SnowballStemmer('english') #initialising the snowballstemmer
```

```

print("Stem/Root words of the some of the words using SnowBall Stemmer:")
print(snow.stem('tasty'))
print(snow.stem('tasteful'))
print(snow.stem('tastiest'))
print(snow.stem('delicious'))
print(snow.stem('amazing'))
print(snow.stem('amaze'))
print(snow.stem('initialize'))
print(snow.stem('fabulous'))
print(snow.stem('unpleasant'))

```

Stem/Root words of the some of the words using SnowBall Stemmer:

```

tasti
tast
tastiest
delici
amaz
amaz
initi
fabul
unpleas

```

Preprocessing Snowball Stemming for one review

```

In [27]: final_string=[]
for i, sent in enumerate(tqdm(final['Text'][2:3].values)):
    filtered_sentence=[]
    print(sent);
    sent=cleanhtml(sent) # remove HTML tags
    for w in sent.split():
        # we have used cleanpunc(w).split(), one more split function here beca
use consider w="abc.def", cleanpunc(w) will return "abc def"
        # if we dont use .split() function then we will be considring "abc de
f" as a single word, but if you use .split() function we will get "abc", "def"
        print(sent.split())
        for cleaned_words in cleanpunc(w).split():
            print("=====>",cleaned_words)
            if((cleaned_words.isalpha()) & (len(cleaned_words)>2)):
                if(cleaned_words.lower() not in stopwords):
                    s=(snow.stem(cleaned_words.lower())).encode('utf8')
                    print("Selected: Stem Word->",s)
                    filtered_sentence.append(s)
            else:
                print("Eliminated as it is a stopwords")

```

```

                                continue
        else:
            print("Eliminated as it is a numerical value or character of length less than 2")
            continue
        print(filtered_sentence)
        str1 = b" ".join(filtered_sentence) #final string of cleaned words
        print("*****")
        final_string.append(str1)
        print(final_string)

```

```

0%|
| 0/1 [00:00<?, ?it/s]

```

I just received my shipment and could hardly wait to try this product. We love "slickers" which is what we call them, instead of stickers because they can be removed so easily. My daughter designed signs to be printed in reverse to use on her car windows. They printed beautifully (we have 'The Print Shop' program). I am going to have a lot of fun with this product because there are windows everywhere and other surfaces like tv screens and computer monitors.

```

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

```

=====> I

Eliminated as it is a numerical value or character of length less than 2

```

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

```

=====> just

Eliminated as it is a stopword

```

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',

```


'which', 'is', 'what', 'we', 'call', 'them', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> received

Selected: Stem Word-> b'receiv'

[b'receiv']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> my

Eliminated as it is a numerical value or character of lenght less than 2

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> shipment

Selected: Stem Word-> b'shipment'

[b'receiv', b'shipment']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> and

Eliminated as it is a stopword

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> could

Selected: Stem Word-> b'could'

[b'receiv', b'shipment', b'could']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> hardly

Selected: Stem Word-> b'hard'

[b'receiv', b'shipment', b'could', b'hard']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> wait

Selected: Stem Word-> b'wait'

[b'receiv', b'shipment', b'could', b'hard', b'wait']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's

creens', 'and', 'computer', 'monitors.']

=====> to

Eliminated as it is a numerical value or character of length less than 2
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> try

Selected: Stem Word-> b'tri'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> this

Eliminated as it is a stopword

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> product

Selected: Stem Word-> b'product'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> We

Eliminated as it is a numerical value or character of length less than 2
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> love

Selected: Stem Word-> b'love'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'l
ove']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> "slickers"

Eliminated as it is a numerical value or character of length less than 2
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> which

Eliminated as it is a stopword

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us

e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> is

Eliminated as it is a numerical value or character of length less than 2
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> what

Eliminated as it is a stopword

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> we

Eliminated as it is a numerical value or character of length less than 2
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> call

Selected: Stem Word-> b'call'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'l
ove', b'call']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',

'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', 'Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> them

Eliminated as it is a stopword

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', 'Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> instead

Selected: Stem Word-> b'instead'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'love', b'call', b'instead']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', 'Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> of

Eliminated as it is a numerical value or character of length less than 2

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', 'Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> stickers

Selected: Stem Word-> b'sticker'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'love', b'call', b'instead', b'of', b'stickers', b'sticker']

ove', b'call', b'instead', b'sticker']
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> because

Eliminated as it is a stopword

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> they

Eliminated as it is a stopword

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> can

Eliminated as it is a stopword

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> be

Eliminated as it is a numerical value or character of length less than 2
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> removed

Selected: Stem Word-> b'remov'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'love', b'call', b'instead', b'sticker', b'remov']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> so

Eliminated as it is a numerical value or character of length less than 2
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> easily

Selected: Stem Word-> b'easili'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'love', b'call', b'instead', b'sticker', b'remov', b'easili']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> My

Eliminated as it is a numerical value or character of length less than 2
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait',
'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter',
'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use',
'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'have',
'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> daughter

Selected: Stem Word-> b'daughter'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'l
ove', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait',
'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter',
'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use',
'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'have',
'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> designed

Selected: Stem Word-> b'design'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'l
ove', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'd
esign']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait',
'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter',
'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use',
'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'have',
'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> signs

Selected: Stem Word-> b'sign'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'l
ove', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'd

esign', b'sign']
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> to

Eliminated as it is a numerical value or character of length less than 2
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> be

Eliminated as it is a numerical value or character of length less than 2
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> printed

Selected: Stem Word-> b'print'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'love', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'design', b'sign', b'print']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',

'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> in

Eliminated as it is a numerical value or character of length less than 2
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> reverse

Selected: Stem Word-> b'revers'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'love', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'design', b'sign', b'print', b'revers']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> to

Eliminated as it is a numerical value or character of length less than 2
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> use

Selected: Stem Word-> b'use'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'love', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'design', b'sign', b'print', b'revers', b'use']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',

'which', 'is', 'what', 'we', 'call', 'them', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> on

Eliminated as it is a numerical value or character of lenght less than 2
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> her

Eliminated as it is a stopword

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> car

Selected: Stem Word-> b'car'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'l
ove', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'd
esign', b'sign', b'print', b'revers', b'use', b'car']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> windows

Selected: Stem Word-> b'window'
[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'l
ove', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'd
esign', b'sign', b'print', b'revers', b'use', b'car', b>window']
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> They

Eliminated as it is a stopword

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> printed

Selected: Stem Word-> b'print'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'l
ove', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'd
esign', b'sign', b'print', b'revers', b'use', b'car', b>window', b'print']
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> beautifully

Selected: Stem Word-> b'beauti'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'l
ove', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'd
esign', b'sign', b'print', b'revers', b'use', b'car', b>window', b'print',
b'beauti']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai

t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> we

Eliminated as it is a numerical value or character of lenght less than 2
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> have

Eliminated as it is a stopword
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> The

Eliminated as it is a stopword
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> Print

Selected: Stem Word-> b'print'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'l

ove', b'call', b'instead', b'call', b'remov', b'easili', b'daughter', b'design', b'sign', b'print', b'revers', b'use', b'car', b>window', b'print', b'beauti', b'print']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> Shop

Selected: Stem Word-> b'shop'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'love', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'design', b'sign', b'print', b'revers', b'use', b'car', b>window', b'print', b'beauti', b'print', b'shop']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> program

Selected: Stem Word-> b'program'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'love', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'design', b'sign', b'print', b'revers', b'use', b'car', b>window', b'print', b'beauti', b'print', b'shop', b'program']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> I

Eliminated as it is a numerical value or character of length less than 2

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> am

Eliminated as it is a numerical value or character of lenght less than 2
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> going

Selected: Stem Word-> b'go'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'l
ove', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'd
esign', b'sign', b'print', b'revers', b'use', b'car', b>window', b'print',
b'beauti', b'print', b'shop', b'program', b'go']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> to

Eliminated as it is a numerical value or character of lenght less than 2
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's

creens', 'and', 'computer', 'monitors.']

=====> have

Eliminated as it is a stopword

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', 'Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> a

Eliminated as it is a numerical value or character of length less than 2

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', 'Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> lot

Selected: Stem Word-> b'lot'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'love', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'design', b'sign', b'print', b'revers', b'use', b'car', b'window', b'print', b'beauti', b'print', b'shop', b'program', b'go', b'lot']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', 'Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> of

Eliminated as it is a numerical value or character of length less than 2

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us

e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> fun

Selected: Stem Word-> b'fun'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'l
ove', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'd
esign', b'sign', b'print', b'revers', b'use', b'car', b>window', b'print',
b'beauti', b'print', b'shop', b'program', b'go', b'lot', b'fun']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> with

Eliminated as it is a stopword

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> this

Eliminated as it is a stopword

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> product

Selected: Stem Word-> b'product'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'l

ove', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'd
esign', b'sign', b'print', b'revers', b'use', b'car', b>window', b'print',
b'beauti', b'print', b'shop', b'program', b'go', b'lot', b'fun', b'produc
t']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> because

Eliminated as it is a stopword

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> there

Eliminated as it is a stopword

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']

=====> are

Eliminated as it is a stopword

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wai
t', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The', 'Print', "Shop'", 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',

'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> windows

Selected: Stem Word-> b>window'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'love', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'design', b'sign', b'print', b'revers', b'use', b'car', b>window', b'print', b'beauti', b'print', b'shop', b'program', b'go', b'lot', b'fun', b'product', b>window']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> everywhere

Selected: Stem Word-> b'everywher'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'love', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'design', b'sign', b'print', b'revers', b'use', b'car', b>window', b'print', b'beauti', b'print', b'shop', b'program', b'go', b'lot', b'fun', b'product', b>window', b'everywher']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> and

Eliminated as it is a stopword

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

```

=====> other
Eliminated as it is a stopword
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']]

```

```

=====> surfaces

```

```

Selected: Stem Word-> b'surfac'
[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'love', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'design', b'sign', b'print', b'revers', b'use', b'car', b>window', b'print', b'beauti', b'print', b'shop', b'program', b'go', b'lot', b'fun', b'product', b>window', b'everywher', b'surfac']
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']]

```

```

=====> like

```

```

Selected: Stem Word-> b'like'
[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'love', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'design', b'sign', b'print', b'revers', b'use', b'car', b>window', b'print', b'beauti', b'print', b'shop', b'program', b'go', b'lot', b'fun', b'product', b>window', b'everywher', b'surfac', b'like']
['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']]

```

```

=====> tv

```

Eliminated as it is a numerical value or character of length less than 2

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> screens

Selected: Stem Word-> b'screen'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'love', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'design', b'sign', b'print', b'revers', b'use', b'car', b>window', b'print', b'beauti', b'print', b'shop', b'program', b'go', b'lot', b'fun', b'product', b>window', b'everywher', b'surfac', b'like', b'screen']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> and

Eliminated as it is a stopword

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"', 'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers', 'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughter', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'use', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we', 'have', '"The', 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'have', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there', 'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 'screens', 'and', 'computer', 'monitors.']

=====> computer

Selected: Stem Word-> b'comput'

[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'love', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'design', b'sign', b'print', b'revers', b'use', b'car', b>window', b'print', b'beauti', b'print', b'shop', b'program', b'go', b'lot', b'fun', b'product', b>window', b'everywher', b'surfac', b'like', b'screen', b'comput']

['I', 'just', 'received', 'my', 'shipment', 'and', 'could', 'hardly', 'wait', 'to', 'try', 'this', 'product.', 'We', 'love', '"slickers"',

```
'which', 'is', 'what', 'we', 'call', 'them,', 'instead', 'of', 'stickers',
'because', 'they', 'can', 'be', 'removed', 'so', 'easily.', 'My', 'daughte
r', 'designed', 'signs', 'to', 'be', 'printed', 'in', 'reverse', 'to', 'us
e', 'on', 'her', 'car', 'windows.', 'They', 'printed', 'beautifully', '(we',
'have', '"The", 'Print', "Shop"', 'program).', 'I', 'am', 'going', 'to', 'ha
ve', 'a', 'lot', 'of', 'fun', 'with', 'this', 'product', 'because', 'there',
'are', 'windows', 'everywhere', 'and', 'other', 'surfaces', 'like', 'tv', 's
creens', 'and', 'computer', 'monitors.']
```

```
=====> monitors
```

```
Selected: Stem Word-> b'monitor'
```

```
[b'receiv', b'shipment', b'could', b'hard', b'wait', b'tri', b'product', b'l
ove', b'call', b'instead', b'sticker', b'remov', b'easili', b'daughter', b'd
esign', b'sign', b'print', b'revers', b'use', b'car', b'window', b'print',
b'beauti', b'print', b'shop', b'program', b'go', b'lot', b'fun', b'produc
t', b'window', b'everywher', b'surfac', b'like', b'screen', b'comput', b'mon
itor']
```

```
*****
```

```
[b'receiv shipment could hard wait tri product love call instead sticker rem
ov easili daughter design sign print revers use car window print beauti prin
t shop program go lot fun product window everywher surfac like screen comput
monitor']
```

```
100%|████████████████████████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 1.15it/s]
```

Preprocessing Snowball Stemming for all reviews

```
In [26]: %%time
str1 = ' '
s=' '
final_string=[]
all_positive_words=[] # store words from +ve reviews here
all_negative_words=[] # store words from -ve reviews here.
i=0
for i, sent in enumerate(tqdm(final['Text'].values)):
    filtered_sentence=[]
    #print(sent);
    sent=cleanhtml(sent) # remove HTML tags
    for w in sent.split():
        # we have used cleanpunc(w).split(), one more split function here beca
use consider w="abc.def", cleanpunc(w) will return "abc def"
        # if we dont use .split() function then we will be considring "abc de
f" as a single word, but if you use .split() function we will get "abc", "def"
        #print(sent.split())
```

```
100% | ██████████  
██████ | 9564/9564 [00:08<00:00, 1180.57it/s]
```

No. of positive words: 289687

No. of negative words: 68290

Most Common positive words [(b'like', 3534), (b'not', 3522), (b'tast', 3239), (b'flavor', 3022), (b'good', 2966), (b'great', 2743), (b'love', 2707), (b'use', 2492), (b'coffe', 2430), (b'one', 2351)]

Most Common negative words [(b'not', 1545), (b'tast', 926), (b'like', 882), (b'product', 774), (b'flavor', 576), (b'one', 555), (b'tri', 536), (b'would', 513), (b'use', 433), (b'coffe', 418)]

```
In [28]: freq_dist_positive=nlTK.FreqDist(all_positive_words)
freq_dist_negative=nlTK.FreqDist(all_negative_words)
print("Most Common Positive Words : ",freq_dist_positive.mOST_common(20))
print("Most Common Negative Words : ",freq_dist_negative.mOST_common(20))
```

Most Common Positive Words : [(b'like', 3534), (b'not', 3522), (b'tast', 3239), (b'flavor', 3022), (b'good', 2966), (b'great', 2743), (b'love', 2707), (b'use', 2492), (b'coffe', 2430), (b'one', 2351), (b'tri', 2296), (b'product', 2194), (b'make', 1904), (b'food', 1727), (b'tea', 1618), (b'get', 1602), (b'would', 1382), (b'realli', 1364), (b'buy', 1360), (b'time', 1352)]

Most Common Negative Words : [(b'not', 1545), (b'tast', 926), (b'like', 882), (b'product', 774), (b'flavor', 576), (b'one', 555), (b'tri', 536), (b'would', 513), (b'use', 433), (b'coffe', 418), (b'good', 417), (b'food', 387), (b'buy', 378), (b'get', 370), (b'order', 350), (b'even', 333), (b'dont', 324), (b'time', 322), (b'bag', 307), (b'make', 305)]

Word Cloud of only Positive Reviews

```
In [29]: #https://www.datacamp.com/community/tutorials/wordcloud-python
#https://medium.com/@cyanamous/amazon-fine-food-dataset-eda-text-featurization-data-visualization-892a40e1c312
```

```
from wordcloud import WordCloud

plt.rcParams['figure.figsize']=(8.0,6.0)
plt.figure(num=None, figsize=(12, 10), dpi=80, facecolor='w', edgecolor='k')
plt.rcParams['font.size']=12
plt.rcParams['savefig.dpi']=100
plt.rcParams['figure.subplot.bottom']=.1

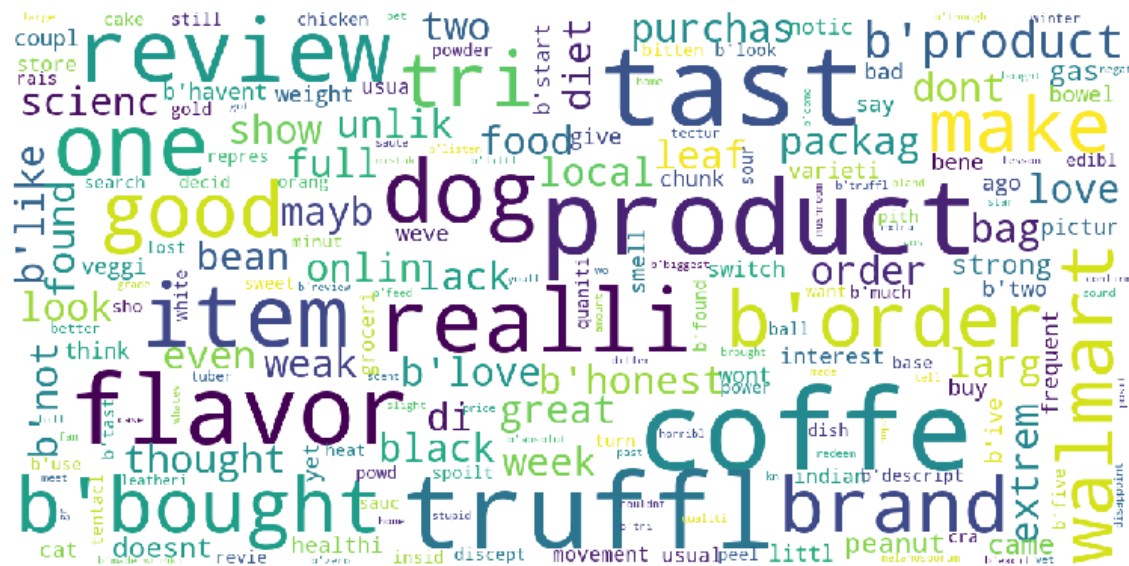
def show_wordcloud(data, title = None):
    wordcloud = WordCloud(background_color='white',stopwords=stopwords,
                           ,max_words=200,max_font_size=40
                           ,scale=3,random_state=1).generate(str(data))
```



```
def show_wordcloud(data, title = None):
    wordcloud = WordCloud(background_color='white', stopwords=stopwords
                           , max_words=200, max_font_size=40
                           , scale=3, random_state=1).generate(str(data))
    fig = plt.figure(1, figsize=(8, 8))
    plt.axis('off')
    if title:
        fig.suptitle(title, fontsize=20)
        fig.subplots_adjust(top=2.3)

    plt.imshow(wordcloud)
    plt.show()

show_wordcloud(final.loc[final['Score'] == 'Negative']['CleanedText'])
```



```
In [31]: # Combining all the above students
from tqdm import tqdm
preprocessed_reviews = []
# tqdm is for printing the status bar
for sentence in tqdm(final['Text'].values):
    sentence = re.sub(r"http\S+", "", sentence)
    sentence = BeautifulSoup(sentence, 'lxml').get_text()
    sentence = decontracted(sentence)
    sentence = re.sub("\S*\d\S*", "", sentence).strip()
    sentence = re.sub('[^A-Za-z]+', ' ', sentence)
    # https://gist.github.com/sebleier/554280
    sentence = ' '.join(e.lower() for e in sentence.split() if e.lower() not i
```

```
n stopwords)
    preprocessed_reviews.append(sentence.strip())
```

```
In [32]: print(preprocessed_reviews[8000])
          print(preprocessed_reviews[9000])
```

[3.2] Preprocess Summary

```
In [33]: ## Similarly you can do preprocessing for review summary also.
# printing some random reviews
sent_500 = final['Summary'].values[500]
print(sent_500)
print("="*50)

sent_1000 = final['Summary'].values[1000]
print(sent_1000)
print("="*50)

sent_2000 = final['Summary'].values[2000]
print(sent_2000)
print("="*50)

sent_3000 = final['Summary'].values[3000]
print(sent_3000)
print("="*50)

sent_6000 = final['Summary'].values[6000]
print(sent_6000)
print("="*50)

sent_8000 = final['Summary'].values[8000]
print(sent_8000)
print("="*50)
```

```
=====
organic dog food
=====
```

```
Wu-yi is
=====
```

```
Don't Waste Your Money
=====
```

```
In [34]: # https://stackoverflow.com/a/47091490/4084039
import re

def decontracted(phrase):
    # specific
    phrase = re.sub(r"won't", "will not", phrase)
    phrase = re.sub(r"can't", "can not", phrase)

    # general
    phrase = re.sub(r"n't", " not", phrase)
    phrase = re.sub(r"\'re", " are", phrase)
    phrase = re.sub(r"\'s", " is", phrase)
    phrase = re.sub(r"\'S", " is", phrase)
    phrase = re.sub(r"\'d", " would", phrase)
    phrase = re.sub(r"\'ll", " will", phrase)
    phrase = re.sub(r"\'t", " not", phrase)
    phrase = re.sub(r"\'ve", " have", phrase)
    phrase = re.sub(r"\'m", " am", phrase)
    return phrase
```

```
In [35]: sent_500 = decontracted(sent_500)
print(sent_500)
print("="*50)
```

```
Excellent, but try to find it elsewhere
=====
```

```
In [36]: #remove spacial character: https://stackoverflow.com/a/5843547/4084039
sent_500 = re.sub('[^A-Za-z0-9]+', ' ', sent_500)
print(sent_500)
print('-'*50)

sent_1000 = re.sub('[^A-Za-z0-9]+', ' ', sent_1000)
print(sent_1000)
print('-'*50)

sent_2000 = re.sub('[^A-Za-z0-9]+', ' ', sent_2000)
print(sent_2000)
```

```
In [38]: print(preprocessed_summary[3000])
```

```
print(preprocessed_summary[4000])
print(preprocessed_summary[5000])
print(preprocessed_summary[6000])
print(preprocessed_summary[7000])
print(preprocessed_summary[8000])
print(preprocessed_summary[9000])
```

```
organic dog food
great coffee fair trade
best dog food price
wu yi
love tea
waste money
taste great
```

[4] Featurization

[4.1] BAG OF WORDS

```
In [39]: #BoW
count_vect = CountVectorizer() #in scikit-learn
count_vect.fit(preprocessed_reviews)
print("some feature names ", count_vect.get_feature_names()[:20])
print('='*50)

final_counts = count_vect.transform(preprocessed_reviews)
print("the type of count vectorizer ", type(final_counts))
print("the shape of out text BOW vectorizer ", final_counts.get_shape())
print("the number of unique words ", final_counts.get_shape()[1])
```

```
some feature names  ['aa', 'aaaa', 'aahhhs', 'ab', 'aback', 'abandon', 'abat
es', 'abberline', 'abbott', 'abby', 'abdominal', 'abiding', 'ability', 'abl
e', 'ablution', 'abnormalities', 'abor', 'abou', 'aboulutely', 'aboutamazo
n']
```

```
=====
the type of count vectorizer  <class 'scipy.sparse.csr.csr_matrix'>
the shape of out text BOW vectorizer  (9564, 18240)
the number of unique words  18240
```

[4.2] Bi-Grams and n-Grams.

```
In [40]: #bi-gram, tri-gram and n-gram
```

```
#removing stop words like "not" should be avoided before building n-grams
# count_vect = CountVectorizer(ngram_range=(1,2))
# please do read the CountVectorizer documentation http://scikit-learn.org/stable/modules/generated/sklearn.feature\_extraction.text.CountVectorizer.html
# you can choose these numebrs min_df=10, max_features=5000, of your choice
count_vect = CountVectorizer(ngram_range=(1,2), min_df=10, max_features=10000)
final_bigram_counts = count_vect.fit_transform(preprocessed_reviews)
print("the type of count vectorizer ",type(final_bigram_counts))
print("the shape of out text BOW vectorizer ",final_bigram_counts.get_shape())
print("the number of unique words including both unigrams and bigrams ", final_bigram_counts.get_shape()[1])
```

```
the type of count vectorizer <class 'scipy.sparse.csr.csr_matrix'>
the shape of out text BOW vectorizer (9564, 5251)
the number of unique words including both unigrams and bigrams 5251
```

[4.3] TF-IDF

```
In [58]: tf_idf_vect = TfidfVectorizer(ngram_range=(1,2), min_df=10)
tf_idf_vect.fit(preprocessed_reviews)
print("some sample features(unique words in the corpus)",tf_idf_vect.get_feature_names()[0:20])
print('='*50)

final_tf_idf = tf_idf_vect.transform(preprocessed_reviews)
print("the type of count vectorizer ",type(final_tf_idf))
print("the shape of out text TFIDF vectorizer ",final_tf_idf.get_shape())
print("the number of unique words including both unigrams and bigrams ", final_tf_idf.get_shape()[1])
```

```
some sample features(unique words in the corpus) ['ability', 'able', 'able b
uy', 'able eat', 'able find', 'able order', 'able use', 'absolute', 'absolut
e best', 'absolute favorite', 'absolutely', 'absolutely best', 'absolutely d
elicious', 'absolutely love', 'absolutely loved', 'absolutely loves', 'absor
bed', 'acacia', 'acai', 'accept']
```

```
=====
the type of count vectorizer <class 'scipy.sparse.csr.csr_matrix'>
the shape of out text TFIDF vectorizer (9564, 5251)
the number of unique words including both unigrams and bigrams 5251
```

[4.4] Word2Vec

```
In [42]: # Train your own Word2Vec model using your own text corpus
```



```

i=0
list_of_sentence=[]
for sentence in preprocessed_reviews:
    list_of_sentence.append(sentence.split())

```

```

In [43]: # Using Google News Word2Vectors
# in this project we are using a pretrained model by google
# its 3.3G file, once you load this into your memory
# it occupies ~9Gb, so please do this step only if you have >12G of ram
# we will provide a pickle file wich contains a dict ,
# and it contains all our courpus words as keys and model[word] as values
# To use this code-snippet, download "GoogleNews-vectors-negative300.bin"
# from https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit
# it's 1.9GB in size.

# http://kavita-ganesan.com/gensim-word2vec-tutorial-starter-code/#.W17SRFAzZP
# Y
# you can comment this whole cell
# or change these variable according to your need

is_your_ram_gt_16g=True
want_to_use_google_w2v = True
want_to_train_w2v = False

if want_to_train_w2v:
    # min_count = 5 considers only words that occurred atleast 5 times
    w2v_model=Word2Vec(list_of_sentence,min_count=5,size=50, workers=4)
    print(w2v_model.wv.most_similar('great'))
    print('='*50)
    print(w2v_model.wv.most_similar('worst'))

elif want_to_use_google_w2v and is_your_ram_gt_16g:
    if os.path.isfile('GoogleNews-vectors-negative300.bin'):
        w2v_model=KeyedVectors.load_word2vec_format('GoogleNews-vectors-negati
ve300.bin', binary=True)
        print(w2v_model.wv.most_similar('great'))
        print(w2v_model.wv.most_similar('worst'))
    else:
        print("you don't have gogole's word2vec file, keep want_to_train_w2v =
True, to train your own w2v ")

```

```

[('terrific', 0.7989331483840942), ('fantastic', 0.7935211658477783), ('trem
endous', 0.7748856544494629), ('wonderful', 0.7647868394851685), ('good', 0.
7291510105133057), ('incredible', 0.7032873630523682), ('marvelous', 0.69711
02952957153), ('phenomenal', 0.6841565370559692), ('amazing', 0.663412809371
9482), ('awesome', 0.6510506868362427)]
[('Worst', 0.6146092414855957), ('weakest', 0.6143776774406433), ('scaries

```

```
t', 0.5957258343696594), ('ugliest', 0.5931180715560913), ('best', 0.5835110545158386), ('bleakest', 0.5718506574630737), ('strongest', 0.567145586013794), ('nastiest', 0.5644308924674988), ('lousiest', 0.5631451606750488), ('toughest', 0.5624395608901978)]
```

```
In [44]: w2v_words = list(w2v_model.wv.vocab)
print("number of words that occurred minimum 5 times ", len(w2v_words))
print("sample words ", w2v_words[0:100])
```

```
number of words that occurred minimum 5 times 3000000
sample words ['</s>', 'in', 'for', 'that', 'is', 'on', '##', 'The', 'with',
'said', 'was', 'the', 'at', 'not', 'as', 'it', 'be', 'from', 'by', 'are',
'I', 'have', 'he', 'will', 'has', '####', 'his', 'an', 'this', 'or', 'thei
r', 'who', 'they', 'but', '$', 'had', 'year', 'were', 'we', 'more', '###',
'up', 'been', 'you', 'its', 'one', 'about', 'would', 'which', 'out', 'can',
'It', 'all', 'also', 'two', 'after', 'first', 'He', 'do', 'time', 'than', 'w
hen', 'We', 'over', 'last', 'new', 'other', 'her', 'people', 'into', 'In',
'our', 'there', 'A', 'she', 'could', 'just', 'years', 'some', 'U.S.', 'thre
e', 'million', 'them', 'what', 'But', 'so', 'no', 'like', 'if', 'only', 'per
cent', 'get', 'did', 'him', 'game', 'back', 'because', 'now', '#.#', 'befor
e']
```

[4.4.1] Converting text into vectors using wAvg W2V, TFIDF-W2V

[4.4.1.1] Avg W2v

```
In [45]: # average Word2Vec
# compute average word2vec for each review.
sent_vectors = []; # the avg-w2v for each sentence/review is stored in this list
for sent in tqdm(list_of_sentence): # for each review/sentence
    sent_vec = np.zeros(300) # as word vectors are of zero length 50, you might need to change this to 300 if you use google's w2v
    cnt_words = 0; # num of words with a valid vector in the sentence/review
    for word in sent: # for each word in a review/sentence
        if word in w2v_words:
            vec = w2v_model.wv[word]
            sent_vec += vec
            cnt_words += 1
    if cnt_words != 0:
        sent_vec /= cnt_words
    sent_vectors.append(sent_vec)
```

```
In [47]: # TF-IDF weighted Word2Vec
tfidf_feat = model.get_feature_names() # tfidf words/col-names
# final_tf_idf is the sparse matrix with row= sentence, col=word and cell_val
# = tfidf

tfidf_sent_vectors = []; # the tfidf-w2v for each sentence/review is stored in
this list
row=0;
for sent in tqdm(list_of_sentence): # for each review/sentence
    sent_vec = np.zeros(300) # as word vectors are of zero length
    weight_sum = 0; # num of words with a valid vector in the sentence/review
    for word in sent: # for each word in a review/sentence
        if word in w2v_words and word in tfidf_feat:
            vec = w2v_model.wv[word]
            # tf_idf = tf_idf_matrix[row, tfidf_feat.index(word)]
            # to reduce the computation we are
            # dictionary[word] = idf value of word in whole corpus
            # sent.count(word) = tf value of word in this review
            tf_idf = dictionary[word]*(sent.count(word)/len(sent))
            sent_vec += (vec * tf_idf)
            weight_sum += tf_idf
    if weight_sum != 0:
        sent_vec /= weight_sum
    tfidf_sent_vectors.append(sent_vec)
    row += 1
```

[5] Applying TSNE

1. you need to plot 4 tsne plots with each of these feature set
 - A. Review text, preprocessed one converted into vectors using (BOW)
 - B. Review text, preprocessed one converted into vectors using (TFIDF)
 - C. Review text, preprocessed one converted into vectors using (AVG W2v)
 - D. Review text, preprocessed one converted into vectors using (TFIDF W2v)
2. [Note 1: The TSNE accepts only dense matrices](#)
3. [Note 2: Consider only 5k to 6k data points](#)

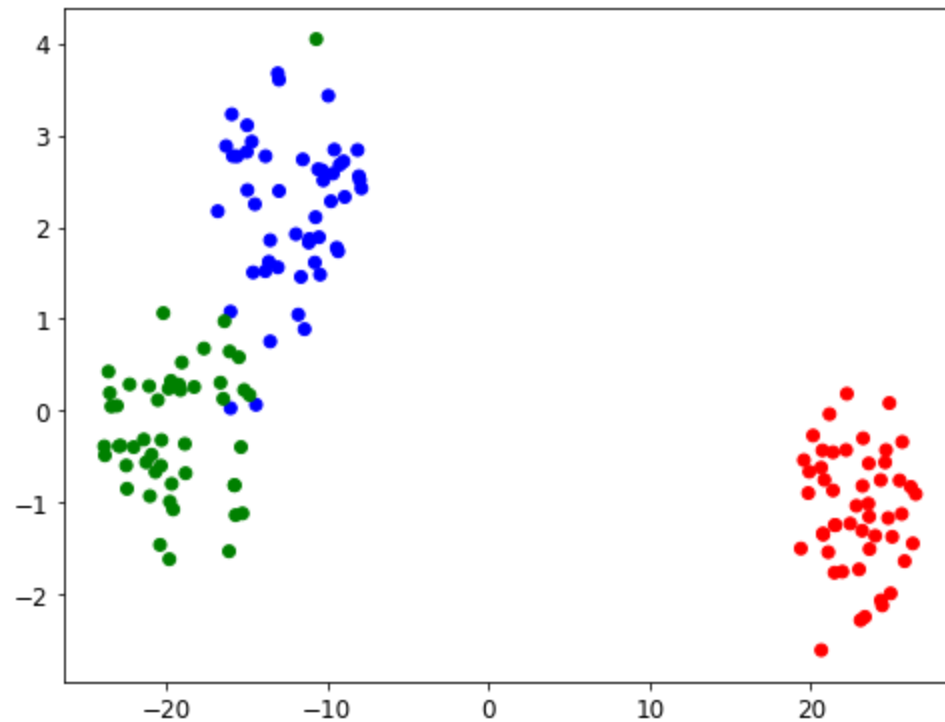
```
In [48]: # https://github.com/pavlin-polcar/fastTSNE you can try this also, this versi
on is little faster than sklearn
import numpy as np
from sklearn.manifold import TSNE
from sklearn import datasets
import pandas as pd
import matplotlib.pyplot as plt

iris = datasets.load_iris()
x = iris['data']
y = iris['target']

tsne = TSNE(n_components=2, perplexity=30, learning_rate=200)

X_embedding = tsne.fit_transform(x)
# if x is a sparse matrix you need to pass it as X_embedding = tsne.fit_transf
orm(x.toarray()) , .toarray() will convert the sparse matrix into dense matrix

for_tsne = np.hstack((X_embedding, y.reshape(-1,1)))
for_tsne_df = pd.DataFrame(data=for_tsne, columns=['Dimension_x', 'Dimension_y'
, 'Score'])
colors = {0:'red', 1:'blue', 2:'green'}
plt.scatter(for_tsne_df['Dimension_x'], for_tsne_df['Dimension_y'], c=for_tsne
_df['Score'].apply(lambda x: colors[x]))
plt.show()
```



[5.1] Applying TNSE on Text BOW vectors

```
In [49]: from openTSNE import TSNE
from sklearn.preprocessing import StandardScaler
final_counts_standardized = StandardScaler(with_mean = False).fit_transform(fi
nal_counts)

print(type(final_bigram_counts))

final_counts_dense = final_counts_standardized.todense()

print(type(final_counts_dense))

tsne = TSNE(
    n_components=2, perplexity=30, learning_rate=200,
    n_jobs=4, initialization="pca", metric="euclidean",
    early_exaggeration_iter=250, early_exaggeration=12, n_iter=750,
    neighbors="exact", negative_gradient_method="bh",
)

tsne_embedding = tsne.fit(final_counts_dense)
```

```

for_tsne = np.vstack((tsne_embedding.T, final['Score'])).T
for_tsne_df = pd.DataFrame(data=for_tsne, columns=['Dimension_x', 'Dimension_y', 'Score'])

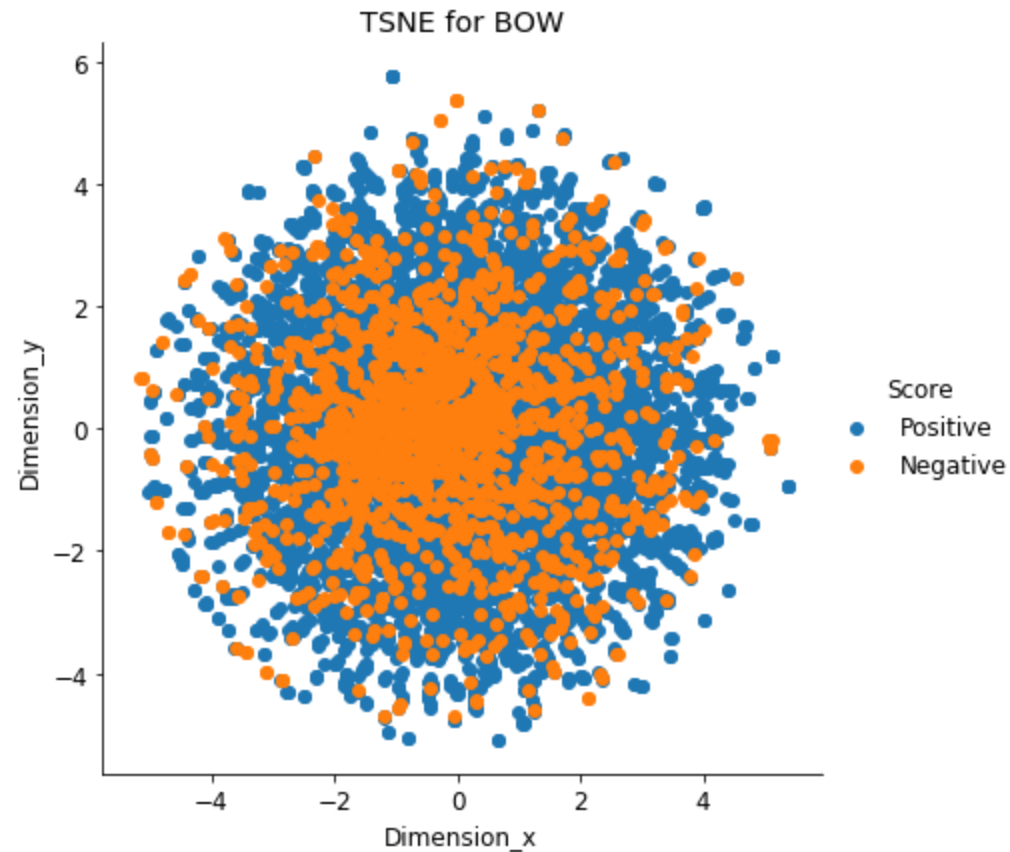
sns.FacetGrid(for_tsne_df, hue = "Score", size = 6).map(plt.scatter, "Dimension_x", "Dimension_y").add_legend()
plt.title("TSNE for BOW")
plt.show()

```

```

<class 'scipy.sparse.csr.csr_matrix'>
<class 'numpy.matrixlib.defmatrix.matrix'>

```



```

In [50]: import numpy as np
from sklearn.manifold import TSNE
from sklearn import datasets
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler
final_bigram_standardized = StandardScaler(with_mean = False).fit_transform(fi
nal_bigram_counts)

```

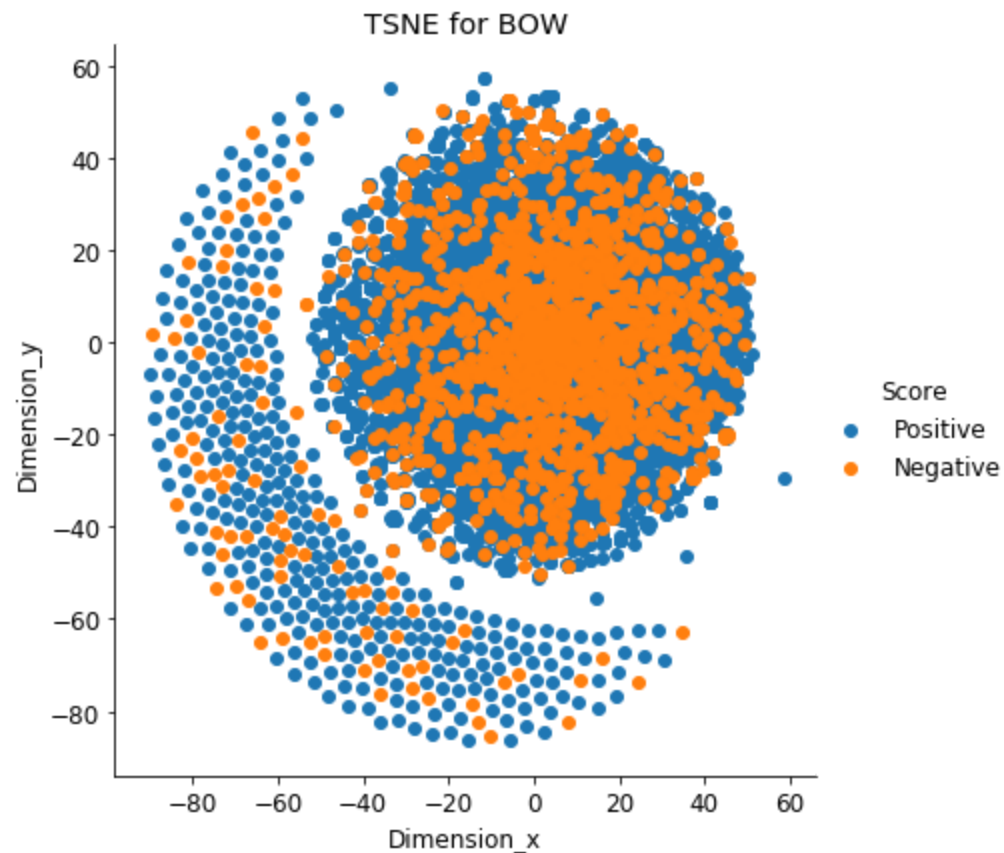
```

final_bigram_dense = final_bigram_standardized.todense()
tsne = TSNE(n_components=2, perplexity=30, learning_rate=200)

final_bigram_embedding = tsne.fit_transform(final_bigram_dense)
# if x is a sparse matrix you need to pass it as X_embedding = tsne.fit_transform(x.toarray()) , .toarray() will convert the sparse matrix into dense matrix
for_bigram_tsne = np.vstack((final_bigram_embedding.T, final['Score'])).T
for_bigram_tsne_df = pd.DataFrame(data=for_bigram_tsne, columns=['Dimension_x', 'Dimension_y', 'Score'])

sns.FacetGrid(for_bigram_tsne_df, hue = "Score", size = 6).map(plt.scatter, "Dimension_x", "Dimension_y").add_legend()
plt.title("TSNE for BOW")
plt.show()

```



[5.1] Applying TNSE on Text TFIDF vectors

In [64]: `import numpy as np`

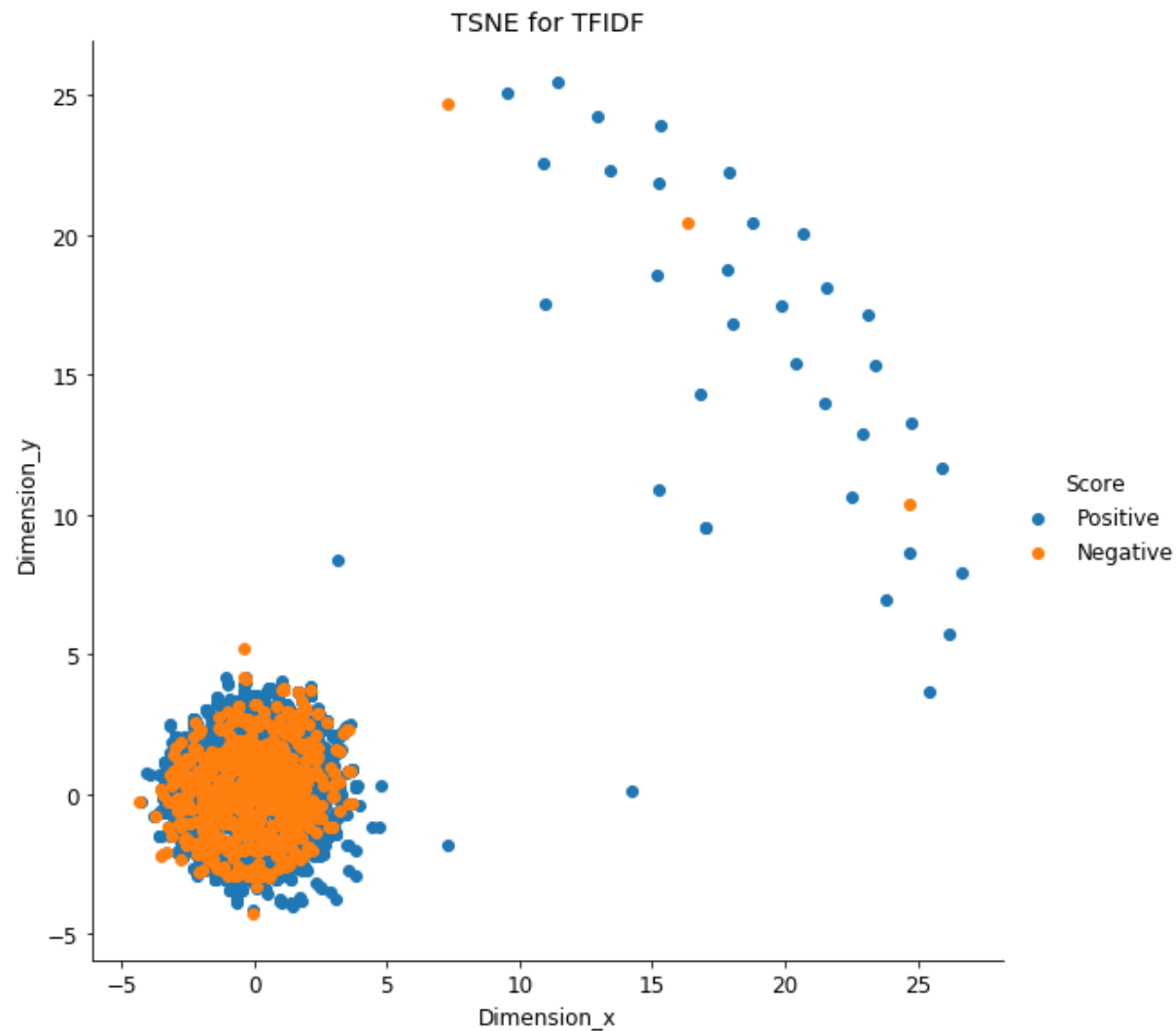
```
from sklearn.manifold import TSNE
from sklearn import datasets
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler
final_tfidf_standardized = StandardScaler(with_mean = False).fit_transform(fin
al_tf_idf)

final_tfidf_dense = final_tfidf_standardized.todense()
tsne = TSNE(n_components=2, perplexity=20, learning_rate=200)

final_tfidf_embedding = tsne.fit_transform(final_tfidf_dense)
# if x is a sparse matrix you need to pass it as X_embedding = tsne.fit_transf
orm(x.toarray()) , .toarray() will convert the sparse matrix into dense matrix
for_tfidf_tsne = np.vstack((final_tfidf_embedding.T,final['Score'])).T
for_tfidf_tsne_df = pd.DataFrame(data=for_tfidf_tsne, columns=['Dimension_x',
'Dimension_y', 'Score'])

sns.FacetGrid(for_tfidf_tsne_df, hue = "Score", size = 8).map(plt.scatter, "Di
mension_x", "Dimension_y").add_legend()
plt.title("TSNE for TFIDF")
plt.show()
```

[5.3] Applying TNSE on Text Avg W2V vectors

```
In [52]: import numpy as np
from sklearn.manifold import TSNE
from sklearn import datasets
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler
final_counts_standardized = StandardScaler(with_mean = False).fit_transform(se
nt_vectors)
```

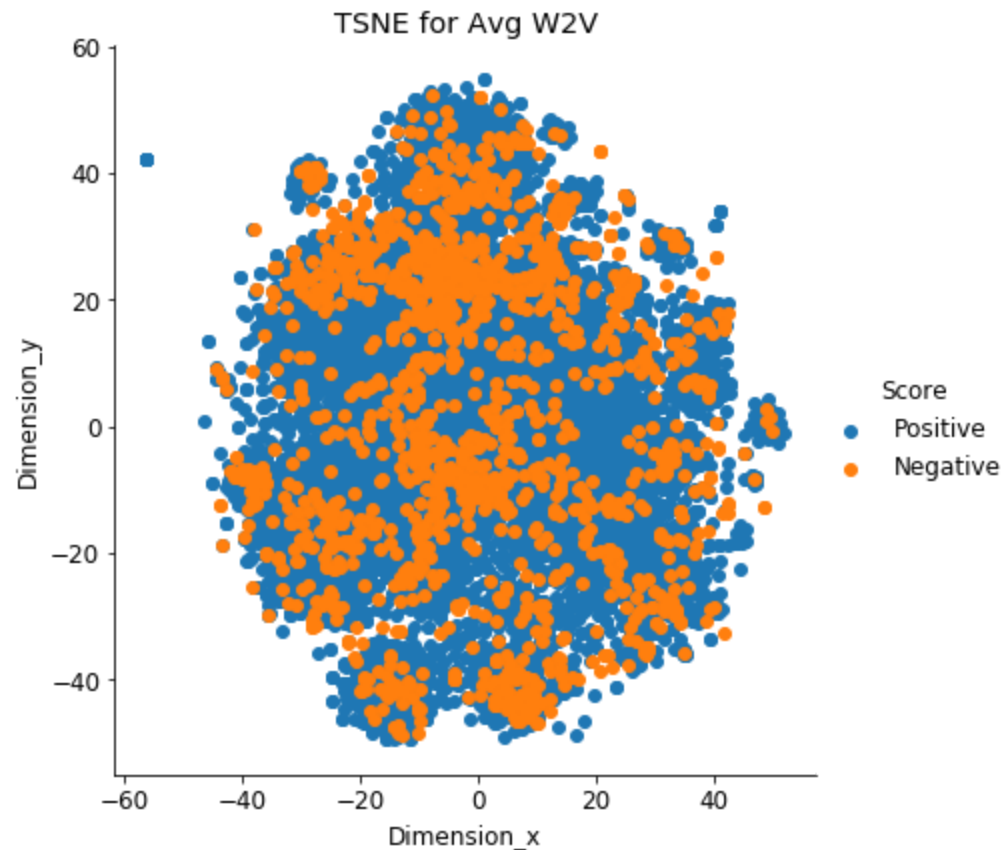
```

type(final_counts_dense)
tsne = TSNE(n_components=2, perplexity=30, learning_rate=200)

final_embedding = tsne.fit_transform(final_counts_standardized)
# if x is a sparse matrix you need to pass it as X_embedding = tsne.fit_transform(x.toarray()) , .toarray() will convert the sparse matrix into dense matrix
for_tsne = np.vstack((final_embedding.T, final['Score'])).T
for_tsne_df = pd.DataFrame(data=for_tsne, columns=['Dimension_x', 'Dimension_y', 'Score'])

sns.FacetGrid(for_tsne_df, hue = "Score", size = 6).map(plt.scatter, "Dimension_x", "Dimension_y").add_legend()
plt.title("TSNE for Avg W2V")
plt.show()

```



[5.4] Applying TNSE on Text TFIDF weighted W2V vectors

```

In [53]: import numpy as np
          from sklearn.manifold import TSNE

```

```

from sklearn import datasets
import pandas as pd
import matplotlib.pyplot as plt

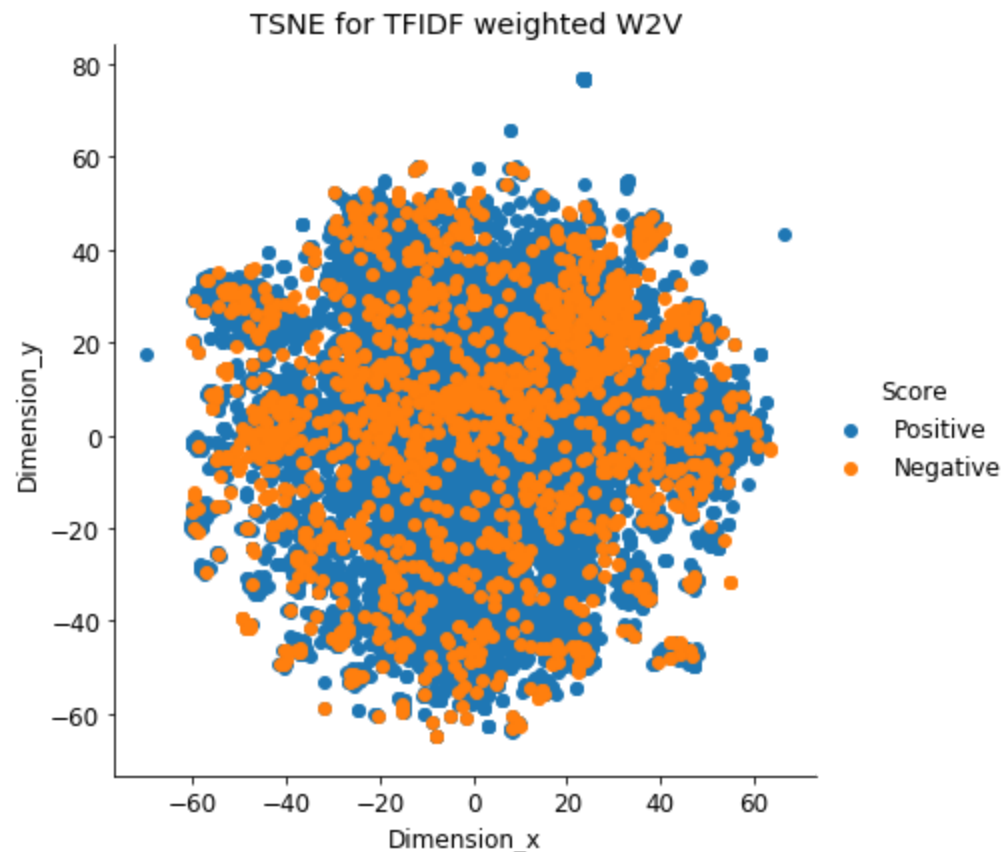
from sklearn.preprocessing import StandardScaler
final_counts_standardized = StandardScaler(with_mean = False).fit_transform(tf
idf_sent_vectors)

tsne = TSNE(n_components=2, perplexity=30, learning_rate=200)

final_embedding = tsne.fit_transform(final_counts_standardized)
# if x is a sparse matrix you need to pass it as X_embedding = tsne.fit_transf
orm(x.toarray()) , .toarray() will convert the sparse matrix into dense matrix
for_tsne = np.vstack((final_embedding.T,final['Score'])).T
for_tsne_df = pd.DataFrame(data=for_tsne, columns=['Dimension_x','Dimension_y'
,'Score'])

sns.FacetGrid(for_tsne_df, hue = "Score", size = 6).map(plt.scatter, "Dimensio
n_x", "Dimension_y").add_legend()
plt.title("TSNE for TFIDF weighted W2V")
plt.show()

```



[6] Conclusions

- From the above TSNE plots it shows that most of the plots are overlapping each other. Since they are overlapped each other quite difficult to separate linearly.

References

- <https://www.datacamp.com/community/tutorials/wordcloud-python>