F215611_CW

March 17, 2023

1 F215611

2 Part 1: Data Pre Processing

- Importing drive data
- Loading data from drive and Cleaning it
- Creating Vocabulary

2.1 Importing libraries, getting drive access of files anf folders

[1]:

```
# Code below was adapted from labs
# Importing few libraries for Loading and cleaning documents
from google.colab import drive
import pandas as pd
import numpy as np
import string
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

# Import few libraries for Creating Vocabulary
from collections import Counter

# Import Few Libraties for Document Word Vector preperation
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import TfidfTransformer

# Import Few Libraries for Tuning n_components value of SVD block
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.decomposition import TruncatedSVD
from matplotlib import pyplot as plt

# Import Few Libraries For Testing System Block
import ipywidgets as widgets
import warnings
warnings.filterwarnings('ignore',category=DeprecationWarning)

#Supress default INFO logging
```

```
import logging
logger = logging.getLogger()
logger.setLevel(logging.CRITICAL)
```

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

[2]:

```
# Mount your Google Drive at the specified mountpoint path to import the␣
 ↪required data in the runtime
drive.mount("/content/drive")
```

```
Mounted at /content/drive
```

[3]:

```
# Accessing folder/files on the drive data path
data_path = "/content/drive/My Drive/Colab Notebooks/COP509cw/Datasets/"
filename = data_path+'JewelleryReviewsLSA.csv'
filename2 = data_path+'JewelleryReviewsQueryRelevantID.csv'
!ls "/content/drive/My Drive/Colab Notebooks/COP509cw/Datasets/"
```

```
JewelleryReviewsLSA.csv              JewelleryReviewsSummarisationTargets.csv
JewelleryReviewsQueryRelevantID.csv
```

2.2 Loading and Cleaning data

Here I read the file JewelleryReviewsLSA.csv and get all the reviews and store it in a list of strings then I clean and scan all the reviews and build the vocabulary incrementally.

Had to convert the string to lower case as some of the stop words were not getting removed just because of first letter in the word being capital.

[4]:

```
# Code below was adapted from labs
# This function load doc into memory
def load_csv(filename):

  # open the file as read only
  reviews = pd.read_csv(filename)
  documents = []
  ratings = []

  # read all text
  for d in reviews.iloc[:,1]:
    documents.append(d.lower())
  return documents

# accept a doc and return all the cleaned list of words
def clean_doc(doc):
        # split into document text by white space
        words = doc.split()
        # remove punctuation from each token
        table = str.maketrans('', '', string.punctuation)
        words = [w.translate(table) for w in words]
        # remove remaining words that are not alphabetic
        words = [word for word in words if word.isalpha()]
        # filter out stop words
        stop_words = set(stopwords.words('english'))
        words = [w for w in words if not w in stop_words]
        # filter out short tokens
        words = [word for word in words if len(word) > 2]
        return words

# Building path to data file
# Then calling both the functions to load and clean documents
filename = data_path+'JewelleryReviewsLSA.csv'
documents = load_csv(filename)

# clean the documents
clean_documents=[]
for d in documents:
  clean_documents.append(clean_doc(d))

# Example of a cleaned document
print("Example of a cleaned document: ",' '.join(clean_documents[0]))
```

```
Example of a cleaned document:  expect like regular size ringbut one look like
ring toy something funy rings ring may cttw round diamond solitaire ring white
gold
```

2.3 Creating vocabulary

Creating the vocabulary and saving it to a text file for future use

[5]:

```
# Code below was adapted from labs
# save file function to save the vocabulary to a file
def save_list(lines, filename):
        data = '\n'.join(lines)
        # open file
        file = open(filename, 'w')
        # write text
        file.write(data)
        # close file
        file.close()



# scan documents and add create a vocabulary by adding all the documents to
# a dictionary
```

```
def add_docs_to_vocab(documents, vocab):
  for d in documents:
    vocab.update(d)
  return vocab

# Initialise Vocabulary
vocab = Counter()

# adding all documents to vocabulary
vocab = add_docs_to_vocab(clean_documents,vocab)

# To remove words with frequencies less than 3
# vocab = {k:v for k, v in vocab.items() if v > 2}

# save tokens to a vocabulary file
save_list(vocab, 'vocab.txt')
```

3 Part 2: Latent Semantic Indexing (LSI)

I have created an app which accepts a query and based on LSI it brings the closest 10 results from the dataset of reviews

LSI model has a few steps:

- Review Vector preperation.
- Tuning n_components value.
- Using SVD(singular value decomposition) to tranform document vectors to a reduced dimention.
- Testing the LSI model on the 8 given queries.

3.0.1 Review Vector preperation

First I clean the reviews to a cleaned review below and then i apply TFIDF Vectorizer to convert into Review vector

[6]:

```
# Code below was adapted from labs
# load doc, clean and return document as a single line of tokens and do it for
# all the documents to return a list of such lines
def docs_to_lines(filename,vocab):
        # load the docs
  docs = load_csv(filename)
  print('Original Review: ',docs[0])
  clean_documents=[]
  lines=[]

  # Doing the below process for all the document
  # 1. Clean document and convert them to clean tokens
  # 2. Convert all the documents to clean lines from these list of clean tokens
  for d in docs:
    doc = clean_doc(d)
    cleaned_doc = [w for w in doc if w in vocab]
    clean_documents.append(' '.join(cleaned_doc))
  return clean_documents

# load the vocabulary
vocab_filename = 'vocab.txt'
vocab = open(vocab_filename,'r')
vocab = vocab.read().split()
vocab = set(vocab)

# Creating document word vectors from vocab
documents = docs_to_lines(filename,vocab)
print('Final Document Vector example: ',documents[0])
```

```
Original Review:  i expect like regular size of ring,but this one look like a
ring for toy or something funy, the mm of our rings is 5mm and this ring may be
is 1mmso ridiculousmartin1/5 ct.tw round diamond solitaire ring in 18k white
gold
Final Document Vector example:  expect like regular size ringbut one look like
ring toy something funy rings ring may cttw round diamond solitaire ring white
gold
```

3.0.2 TFID Vectorization

I tried N-Grams and Count-vectorization technique to see if there is an improvement in the precision and recall of the system but TFIDF performed the best

[7]:

```
# Using TfidfVectorizer to convert word document vectors to corresponding tfidf
# vectors used for LSI
tfidf_vectorizer = TfidfVectorizer(vocabulary=vocab)
transformer = TfidfTransformer(norm='l2')
tfidf_X = transformer.fit_transform(tfidf_vectorizer.fit_transform(documents))
print('TFIDF Document Vector shape: ',tfidf_X.shape)
print('Document Word Vector 0th position: ',documents[0])
print('TFIDF Vector 0th position: \n',tfidf_X[0])
```

```
TFIDF Document Vector shape:  (200, 1095)
Document Word Vector 0th position:  expect like regular size ringbut one look
like ring toy something funy rings ring may cttw round diamond solitaire ring
white gold
TFIDF Vector 0th position:
   (0, 213)     0.29143477776354465
  (0, 252)      0.16409086001173778
  (0, 313)      0.23703515583841775
  (0, 376)      0.33861122217507283
  (0, 395)      0.21897870141251852
  (0, 524)	0.134120106610599
  (0, 536)	0.08475655945742976
  (0, 573)	0.21897870141251852
  (0, 632)	0.0866251635900232
  (0, 765)	0.29143477776354465
  (0, 782)	0.08679387607263346
  (0, 783)	0.33861122217507283
  (0, 787)	0.08124020695984417
  (0, 793)	0.33861122217507283
  (0, 862)	0.12954870421577747
  (0, 885)	0.29143477776354465
  (0, 887)	0.19194240262866433
  (0, 1000)	0.29143477776354465
  (0, 1058)	0.19194240262866433
```

3.0.3 Part 2a: Two Best models for LSI(Evaluation)

- Best Model is made from TFIDF vectors and has n_component=30 and gives me a precision of 75-90 % and a recall of 50-65 %
- second best model has TFIDF vectord and n_component of 15 and gives me a precision of 60-75 % and recall of 35-45 %
- I also Tried N-gram and Count Vectorisation techniques bur TFIDF outperfomed them both
- Graph can be seen at the end that interplots average Precision and average Recall for the 8 queries on this dataset.
- We can also see the performance of TFIDF LSI information retrival system on the 8 queries given in the data set. I tried to bring top 10 reviews relavant to these queries and also plot the average precision/recall interplot graph.

[8]:

```python
# Best performing model
SVD_model = TruncatedSVD(n_components=30)
lsi_x_transformed = SVD_model.fit_transform(tfidf_X)

# 2nd Best performing model
SVD_model2 = TruncatedSVD(n_components=15)
lsi_x_transformed2 = SVD_model2.fit_transform(tfidf_X)

# Takes a Query string and returns a clean string of tokens as a line
def query_to_line(query,vocab):
  # convert string to lower case
  query_lower = query.lower()
  # Cleaning the query to return clean tokens
  clean = clean_doc(query_lower)
  # only keeping tokens that are in vocab
  clean = [w for w in clean if w in vocab]
  return ' '.join(clean)

# Interplot Precision for standard Recall
def InterplotPrecision(p=0.1, Precision=None, Recall=None):
    if p >= 1.0:
        p = 0.9
    # Initialising left and right point of recall
    Mark = np.zeros(2)
    l = 0
    r = 0
    for i in range(len(Recall)):
      if Recall[i] >= p and Mark[0] == 0:
          l = i
          Mark[0] = 1
      if Recall[i] >= p + 0.1 and Mark[1] == 0:
          r = i
          Mark[1] = 1
    y = max(Precision[l:(r+1)],default=0)
    return y

# compute Recall, Precision
def compute_R_P_F1(re_mark=None, QuRe_ID =None):
  Recall = []
  Precision = []
  F1measure = []
  for i in range(len(re_mark)):
    # for all values in the retrieved result calculate recall and precision
    r = sum(re_mark[:(i+1)])
    Re = r/(len(QuRe_ID))
    Pr = r/(i+1)
    # adding values to respective list
    Recall.append(Re)
    Precision.append(Pr)
  return Recall, Precision, F1measure

def compute_RP_yaxis(Precision=None, Recall=None):
  y_axis = [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]
  for i in range(11):
    pInput = 0.1 * i
    #calling function to find y-axis of Interplot between recall and precision
    y_axis[i] = InterplotPrecision(p=pInput, Precision=Precision, Recall=Recall)
  return y_axis

# giving file name and initialising queries list and relavant results list for
# each query
# relavant results list
def initialize_data(queires,rel_id):
  queries_relavant_id = pd.read_csv(filename2)
  for i in range(8):
    id=[]
    for j in queries_relavant_id.iloc[:,i]:
      if not pd.isna(j):
        id.append(int(j))
    rel_id.append(id)

  # queries
  for i in queries_relavant_id.iloc[0,9:17]:
    queries.append(i)
  return queries,rel_id

def get_relavant_reviews(queries,rel_id):
  j=0
  for query in queries:
    # cleaning query string
    query_string = query_to_line(query,vocab)
    vectorizer = TfidfVectorizer(vocabulary=vocab)
    transformer = TfidfTransformer(norm='l2')

    # Query string to TFID query vector
    query_vector = transformer.fit_transform(vectorizer.
 ↪fit_transform([query_string]))

    # reducing Query vector to smaller dimension same as documents
    query_vector = SVD_model.transform(query_vector)
    # Computing cosine similarity
    similarities = cosine_similarity(lsi_x_transformed, query_vector)
    df = pd.read_csv(filename)
    REVIEW = np.array(df[['Reviews']])
    IDS = np.array(df[['ID']])

    # Filtering list of documents according to top 10 similarities
    indexes = np.argsort(similarities.flat)[::-1]
    IDS = IDS[indexes].flat[:]
    re_mark = []
    #Finding out if the result is relavant or not
    for i in range(len(indexes)):
      if (IDS[i]) in rel_id[j]:
        re_mark.append(1)
      else:
        re_mark.append(0)
    # Calculate Recall and Precision scores of all documents corresponding
    # to 6 queries
    Recall, Precision, F1measure = compute_R_P_F1(re_mark=re_mark,␣
 ↪QuRe_ID=rel_id[j])
    Recall = np.array(Recall)
    Precision = np.array(Precision)
    F1measure = np.array(F1measure)
    print('\n' + 'Query%d: '%(j+1) + query)
    print()
    print("Results: ")
    print()
    for i in range(10):
      print("Top " + str(i+1) + ' result: ID: %d '%(IDS[i]), REVIEW[indexes[i]])
    print()
    print("Recall for query "+str(j+1)+" @1~10: ", np.around(Recall[:10],2))
    print("Precision for query "+str(j+1)+" @1~10: ", np.around(Precision[:
 ↪10],2))

    # save all recall and precision values for each query
    AllRecall.append(Recall)
    AllPrecision.append(Precision)
    j=j+1
  return AllRecall,AllPrecision
```

Calling all the functions defined above for LSI Model

[9]:

```
# Calling All the Functions

# Initialize queries list and relavant ids list
rel_id=[]
queries=[]
queries,rel_id = initialize_data(queries,rel_id)

# initialising list to save all recal and precision values of the data set
# for each query
AllRecall = []
AllPrecision = []
AllRecall,AllPrecision = get_relavant_reviews(queries,rel_id)

# calculating average recall and precision for the 6 queries @1-10
AveRecall = (AllRecall[0] + AllRecall[1] + AllRecall[3] + AllRecall[4] +␣
 ↪AllRecall[5] + AllRecall[6] + AllRecall[7])/8
AvePrecision = (AllPrecision[0] + AllPrecision[2] + AllPrecision[3] +␣
 ↪AllPrecision[4] + AllPrecision[5] + AllPrecision[6] + AllPrecision[7])/8
# printing average recall and precision values @1-10
print()
print("\nAverage Recall and average Precision: ")
print("Recall for query 1@1~10: ", np.around(AllRecall[0][:10],2))
print("Precision for query 1@1~10: ", np.around(AllPrecision[0][:10],2))
print()
print()
```

```python
# plot R/P curve for all 8 queries
num_queries=8
x_axis = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
colour = ['black','red','orange','blue','purple','green','pink','brown']
for y in range(num_queries):
  y_axis = compute_RP_yaxis(Precision=AllPrecision[y], Recall=AllRecall[y])
  plt.plot(x_axis, y_axis, '-bo', color=colour[y], label="Query%d"%(y+1))
plt.xlim(0, 1)
plt.ylim(0, 1)
plt.xlabel('Average Recall')
plt.ylabel('Average Precision')
plt.title('Recall/Precision Curve')
plt.legend()
plt.show()
```

```
Query1: The ring is a great gift. My friend loves it

Results:

Top 1 result: ID: 58481  ['my wife loves the ring, it was a great gift.
extremelly cheap and high quality.']
Top 2 result: ID: 17273  ['My mother loved this and was a great birthday gift.
These look even better in person and go great with anything.']
Top 3 result: ID: 56494  ['I bought this ring for my husband and he loved it. I
received it when they said I would and it is a great ring']
Top 4 result: ID: 41876  ["I bought this as a gift for a friends birthday and
she loved it. It's a beautifull ring."]
Top 5 result: ID: 49525  ['this product made for a great gift and great memorize
for my love and me. It something we will always have. a helping gift from the
heart that always shows you care.']
Top 6 result: ID: 26246  ['This was a birthday gift for my 16 YO niece. She
loves the ring and was very happy to have received it.']
Top 7 result: ID: 9726  ['A great gift to your loved one and an ever better
seller. The seller deals with you in the most professional way and the security
measures are superb.']
Top 8 result: ID: 17309  ['I always love Willow Tree. they make great gifts for
great people in your life. I have quite a collection, and I hope to continue to
build it']
Top 9 result: ID: 36164  ['I got the ring as a promise ring for my girlfriend
for Christmas and she loved it. Definitely a great value.']
Top 10 result: ID: 34523  ['This ring has such a good sparkle and it looks like
a ring that should cost 10x the amount. Makes a great gift for someone on a
budget. My girlfriend loves it.']

Recall for query 1 @1~10:  [0.06 0.12 0.19 0.25 0.31 0.38 0.44 0.5  0.56 0.62]
Precision for query 1 @1~10:  [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

```
Query2: horrible bad quality bracelet

Results:

Top 1 result: ID: 55017  ['I bought this bracelet for my girlfriend and I was
not very impressed with the quality. The bracelet itself looks cheap and the
clasp is extremely fragile, in fact the first night she wore the bracelet out
the clasp broke on one side. I definitely would not recommend this bracelet.']
Top 2 result: ID: 7432  ["This bracelet is poor quality of gemstones and poor
craftsmanship. Gemstones are very cloudy and some of the joints in the chain or
fused together and the bracelet therefore can't be straightened out completely.
In other words, if you unfasten the bracelet and lay it out on the counter it
will not form a straight line because some of the joints in the bracelet are
fused together and do not bend so you cannot straighten the bracelet. Don't buy
this, it's junk."]
Top 3 result: ID: 40871  ['I purchased this bracelet as a gift last December.
The quality of the clasp is quite poor and it gradually lost it ability to stay
closed. Recently, the bracelet fell off because of the poor clasp and was
damaged greatly. Avoid this bracelet!']
Top 4 result: ID: 25299  ['the bracelet was not a true 9  the necklace perfect
the bracelet nice quality just not true to length']
Top 5 result: ID: 2114  ['The stones on this bracelet are extremely pale, more
pink than purple. I ended up returning the bracelet because I have amethyst
jewelry and it was extremely poor quality.']
Top 6 result: ID: 57123  ['Very disappointed in the appearance and quality of
the bracelet and its definitely not worth $45.00 - not even close.']
Top 7 result: ID: 13373  ['The item was not as pictured. It is funky and of poor
quality. The seller did not respond when I contacted him about this.']
Top 8 result: ID: 56865  ['Looks exactly like the picture. Very nice quality. A
must for everyone who is a Tiger fan and owns an Italian Charm Bracelet.']
Top 9 result: ID: 54748  ['Item arrived extremely damaged in several places. Not
packaged well had to send it back. Very disappointed with the quality.']
Top 10 result: ID: 17607  ['This product was hollow, which was not clearly
specified in the item description. It was not what I was expecting and seemed to
be poor quality. I returned the item.']

Recall for query 2 @1~10:  [0.17 0.33 0.5  0.67 0.83 1.   1.   1.   1.   1.  ]
Precision for query 2 @1~10:  [1.   1.   1.   1.   1.   1.   0.86 0.75 0.67 0.6
]

Query3: arrived promptly and happy with the seller

Results:

Top 1 result: ID: 33251  ['I am happy with the product, I received it as
advertised and in a timely manner; seller/Amazon kept me updated about
shipment/delivery status. Would recommend item and seller']
```

```
Top 2 result: ID: 22058  ["Item was great quality and came promptly. I'm very
happy with it and recommend it unreservedly."]
Top 3 result: ID: 31657  ['I am happy to say that though I did not receive my
order the first time I ordered this cross, the company and Amazon did stand
behind their product and refund the money spent on the item. Since they did, I
re-ordered the item and it arrived in a timely manner. The cross is really nice.
Thank you for the great customer service!']
Top 4 result: ID: 1816  ['The quality and look were not what I had anticipated.
Very flimsy.I would not recommend this item']
Top 5 result: ID: 4375  ['Very impressed with the quality of my item. Delivery
was fast. Would definately buy from this seller again']
Top 6 result: ID: 10758  ['Very impressed with the quality of my item. Delivery
was fast. Would definately buy from this seller again']
Top 7 result: ID: 29722  ['I received this Italian horn in pristine condition
and I was completely satisfied with the receiving of this product in a timely
manner.']
Top 8 result: ID: 2780  ['arrived before estimated date. many previous orders
from this seller which always are on time and in excellent condition']
Top 9 result: ID: 13373  ['The item was not as pictured. It is funky and of poor
quality. The seller did not respond when I contacted him about this.']
Top 10 result: ID: 41889  ['I was very impressed with the quality and would not
hesitate to purchase other items from the Seller. Their service was also
exceptional.']

Recall for query 3 @1~10:  [0.07 0.07 0.14 0.14 0.14 0.14 0.21 0.29 0.29 0.29]
Precision for query 3 @1~10:  [1.   0.5  0.67 0.5  0.4  0.33 0.43 0.5  0.44 0.4
]

Query4: wear it with casual wear

Results:

Top 1 result: ID: 53660  ['These are nice to wear when you want something casual
to wear. They are very comfortable.']
Top 2 result: ID: 30640  ['This lapel pin is the perfect detail to wear your
colors. I plan to wear it on my lapel when I wear a suit. This pinis nice enough
to wear in formal occasions.Wear it with pride!!']
Top 3 result: ID: 37486  ["This pendant I classify as the best for casual wear.
I wear on the weekends or out & about but isn't not suited for my work or my
going out events"]
Top 4 result: ID: 33746  ['i love this ring my only complaint is that the metal
is so soft that it bent to the point that i couldnt wear it anymore. i still
wear it on a necklace and love it on occasion i can squish it onto my fingure
but it took less than two months for it to be almost impossible for me to
wear.']
Top 5 result: ID: 26535  ["I wear this charm in memory of 3 friends that have
passed away. I wish I didn't have to wear it but it is in their honor that I
do."]
```

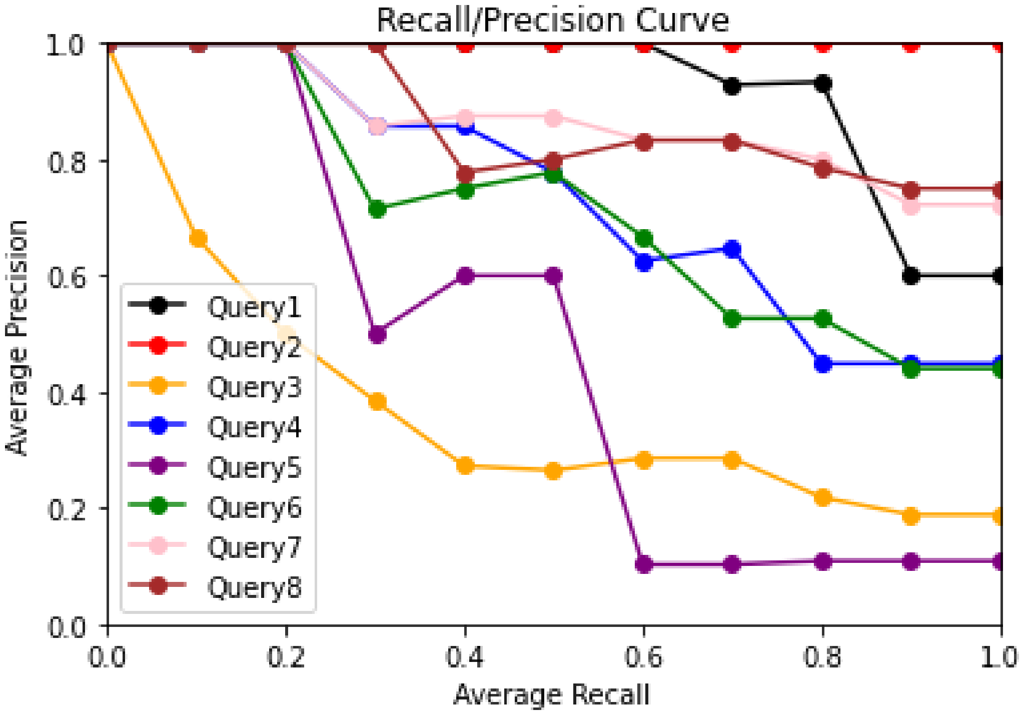
```
Top 6 result: ID: 19852  ['very good for everyday wear or dressing up']
Top 7 result: ID: 44126  ['It is so unique and a pleasure to wear. The stones
catch the light and the style is very comfortable to wear.']
Top 8 result: ID: 52663  ['I have been told that the ring is very comfortable to
wear and he was quite surprised and please to see the Masonic ring in
titanium.']
Top 9 result: ID: 28648  ['The days I do not wear the blue one I wear this one.
I really enjoy wearing something Celtic and pretty.']
Top 10 result: ID: 54548  ["hard and can't wear it, the material is hard and not
easy to wear I just don't like it I should have saved my money and got on from
A&E"]

Recall for query 4 @1~10:  [0.07 0.14 0.21 0.21 0.29 0.36 0.43 0.43 0.5  0.5 ]
Precision for query 4 @1~10:  [1.   1.   1.   0.75 0.8  0.83 0.86 0.75 0.78 0.7
]

Query5: i expected better quality. i will return this item

Results:

Top 1 result: ID: 33571  ['The item was misrepresented. Size and quality were
horrible. I would return this item except family member is in the Coast Guard
and it was sent to him. A total waste of money.']
Top 2 result: ID: 45548  ['This is an attractive and high quality item for a
young teenager. It is too small for an adult.']
Top 3 result: ID: 265  ['The quality of this item was not up to expectations.The
Top was scratched, the hinges did not line up to the pre-drilled holes and the
staining was inconsistant. If I saw this item in a store I would not have
purchased it.']
Top 4 result: ID: 17607  ['This product was hollow, which was not clearly
specified in the item description. It was not what I was expecting and seemed to
be poor quality. I returned the item.']
Top 5 result: ID: 13373  ['The item was not as pictured. It is funky and of poor
quality. The seller did not respond when I contacted him about this.']
Top 6 result: ID: 1816  ['The quality and look were not what I had anticipated.
Very flimsy.I would not recommend this item']
Top 7 result: ID: 17944  ['the product i recieved was nice it came in a timley
matter faster than i expected will order this item again']
Top 8 result: ID: 32674  ['THIS ITEM WAS A WONDERFUL SURPRISE. THE QUALITY IS SO
MUCH MORE THAN I COULD HAVE EVER HOPED FOR.']
Top 9 result: ID: 22058  ["Item was great quality and came promptly. I'm very
happy with it and recommend it unreservedly."]
Top 10 result: ID: 51907  ["I was not completely sure about buying jewlery over
the Internet. For this type of gift I like to see and handle the item to decide.
I was pressed for time and I did like the pictures of the item so I decided to
take a chance. I'm glad I did as I was very impressed by the quality and
appearance when the item arrived. The price was low compared to the quality and
my wife and I were very pleased with this jewlery item."]
```

```
Recall for query 5 @1~10:  [0.2 0.2 0.2 0.4 0.6 0.6 0.6 0.6 0.6 0.6]
Precision for query 5 @1~10:  [1.   0.5  0.33 0.5  0.6  0.5  0.43 0.38 0.33 0.3
]

Query6: looks beautiful. The design is pretty. pefect and color is light

Results:

Top 1 result: ID: 52375  ['The message is very positive and it looks very
pretty. I bought it for my aunt as a present and the color is very nice.']
Top 2 result: ID: 41319  ['The ring is exactly as pictured and looks very pretty
on my hand. The color of the stones is rich and beautiful.']
Top 3 result: ID: 46500  ['The Earrings you sent me are real light in color not
the pretty dark color you show in the picture. They look almost light pink. I
will keep them they are also pretty but not what I expected.']
Top 4 result: ID: 42077  ["This is a solid.beautiful ring. But if you are
expecting the color in rhe picture you will be disappointed. It is barely pink
at all. When I first saw it I thought it was lavendar. It's still pretty but buy
for design not color."]
Top 5 result: ID: 10612  ['You have to see these in person to appreciate how
beautiful they really are. They are so shiny, i feel very pretty in them. They
look very expensive too, much more than the cost.They also go with any color.
They are just beautiful!']
Top 6 result: ID: 45860  ['This is one of the most beautiful rosarys I have
seen. The smoothness and color of the beads is so translucent looking that it
almost looks like glass. The workmanship is excellent and the details are
beautiful. A truly beautiful piece to own.']
Top 7 result: ID: 43945  ["The diamond looks pretty big. For the price, it
shines brilliantly. The color doesn't look very white though. But you don't
expect K color to be very white. Overall, I think it's pretty. and I am very
happy with it."]
Top 8 result: ID: 27474  ["The diamond looks pretty big. For the price, it
shines brilliantly. The color doesn't look very white though. But you don't
expect K color to be very white. Overall, I think it's pretty. and I am very
happy with it."]
Top 9 result: ID: 8341  ['The smaller carnelian beads have a clear beautiful
orange color. It would be great if the elastic was the same color of the beads.
Very beautiful.']
Top 10 result: ID: 9050  ['The ring is pretty enough, but the metal of the ring
is very insubstantial it pushes in very easily.']

Recall for query 6 @1~10:  [0.08 0.17 0.25 0.25 0.25 0.33 0.42 0.5  0.58 0.58]
Precision for query 6 @1~10:  [1.   1.   1.   0.75 0.6  0.67 0.71 0.75 0.78 0.7
]

Query7: This ring looks nothing like the picture. the diamonds are small and not
very noticeable
```

```
Results:

Top 1 result: ID: 209  ['This ring looks nothing like the picture. the diamonds
are small and not very noticeable; I will be sending this back']
Top 2 result: ID: 216  ['This rings looks nothing like the picture at all! The
stones are so small I can barely even tell they are stones, then the ring is so
thin and small. It looks like a ring I would buy in a egg from a machine. I
returned it promptly. This was a very misleading buy.']
Top 3 result: ID: 47345  ["I didn't like this product because the diamonds
looked nothing like the picture. The diamonds are flawed more than a little
bit."]
Top 4 result: ID: 3494  ['It is as nice as it looks on the picture. :) I like
it. :)']
Top 5 result: ID: 37864  ["I fell in love with the picture. The ring showed to
be sligthly brushed looking. When the ring arrived I was quick to learn the
picture looks nothing like the ring. The ring is a bright polish and the yellow
gold is barely visiable. I'm very disappointed with amazon for the lack of
description."]
Top 6 result: ID: 7110  ["The diamonds in the picture are nothing close to the
size they look in the picture but the ring is awesome and I would absolutely
recommend it to anyone for the price. Hands down, no doubt. The ring looks
amazing for the price. Just don't expect the diamonds to look as big as they do
in the picture."]
Top 7 result: ID: 44358  ['I got this ring hoping that it looks like the picture
which was misleading. If you purchase this ring, please be aware that it is 1/10
cttw which means that the diamonds are really small. If you are a man looking to
buy a ring for his significant other, I would recommend you spend it a little
bit more and buy a better looking ring. I ended up returning it.']
Top 8 result: ID: 38637  ['I was a little disappointed when I received my ring
in the mail. In the picture provided above the sides look like they make a heart
shape, or at least it looks like smooth, clean curved lines. The ring I got in
the mail looks like the sides are smushed in and not clean curves. Other then
that I like it. I just wished it looked like the picture.']
Top 9 result: ID: 41872  ['Looked just as well as the picture does. Only thing i
could say is that it is a little more polished than it looks like and the black
stands out which looks very nice.']
Top 10 result: ID: 943  ['It looks like a ring for a man when you look at the
picture online, but in real life its a very feminine looking ring.']

Recall for query 7 @1~10:  [0.07 0.14 0.21 0.21 0.29 0.36 0.43 0.5  0.5  0.57]
Precision for query 7 @1~10:  [1.   1.   1.   0.75 0.8  0.83 0.86 0.88 0.78 0.8
]

Query8: braclet looked just like its picture and is nice quality sterling
silver.

Results:
```

```
Top 1 result: ID: 642  ['This medical alert braclet looked just like its picture
and is nice quality sterling silver.']
Top 2 result: ID: 3494  ['It is as nice as it looks on the picture. :) I like
it. :)']
Top 3 result: ID: 10642  ['From the picture they looked to have some purple in
them but they are clear just like the title says.']
Top 4 result: ID: 53409  ['Although the picture looks like metal beads and
description states sterling silver, these are pearls.']
Top 5 result: ID: 33632  ["I usually love rings from this company but be warned
that this ring doesn't look anything like the picture. The ring I received had a
super long rectangular stone, not a nice square as seen in the picture. Mine was
also tarnished or looked like it had torch marks up the side. I will definitely
be returning this ring."]
Top 6 result: ID: 41872  ['Looked just as well as the picture does. Only thing i
could say is that it is a little more polished than it looks like and the black
stands out which looks very nice.']
Top 7 result: ID: 38637  ['I was a little disappointed when I received my ring
in the mail. In the picture provided above the sides look like they make a heart
shape, or at least it looks like smooth, clean curved lines. The ring I got in
the mail looks like the sides are smushed in and not clean curves. Other then
that I like it. I just wished it looked like the picture.']
Top 8 result: ID: 735  ['This is a perfect size solid charm that looks the same
on either side. Silver is nicely finished and the enamel is a nice highlight.
Really looks like the picture.']
Top 9 result: ID: 10037  ["These are not what I thought they were going to be
they look good in the picture but when I got them the quality just isn't there.
I can't wear them with out completely bending the hooks out of shape. They are
more like silver plastic."]
Top 10 result: ID: 45518  ['It was much smaller than it looked like in the
picture and the silver necklace seemed to be of poorer quality than expected.']

Recall for query 8 @1~10:  [0.08 0.15 0.23 0.31 0.31 0.38 0.38 0.46 0.54 0.62]
Precision for query 8 @1~10:  [1.   1.   1.   1.   0.8  0.83 0.71 0.75 0.78 0.8
]


Average Recall and average Precision:
Recall for query 1@1~10:  [0.06 0.12 0.19 0.25 0.31 0.38 0.44 0.5  0.56 0.62]
Precision for query 1@1~10:  [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

```
<ipython-input-9-fd4779cec0ce>:32: UserWarning: color is redundantly defined by
the 'color' keyword argument and the fmt string "-bo" (-> color='b'). The
keyword argument will take precedence.
  plt.plot(x_axis, y_axis, '-bo', color=colour[y], label="Query%d"%(y+1))
```



3.0.4 Part 2b Tuning n_components value of SVD

I use SVD to compress TFIDF vectors to n_component dimensional data and tune the number to a number for which we preserve the variance and loose the unwanted noise.

For this I tried a way to test similarity by averaging similarity among few similar documents I chose at random, to plot and see which number works and gets maximum variance with minimum loss(here i take similarity as a measure).

I took 6 Queries of similar documents and compared the cosine similarity among each other and then averaged the similarities for each n_component to plot on a graph.I run a loop from 1-50 to plot a graph to see how the similarity changes with n_components(1-50) between 2 queries(each time comparing a document with all others) The 6 queries i selected were

- I bought two of these rings to fit on either side of my own platinum princess wedding ring to help anchor and to add to the ring. Perfect fit! I have had a lot of compliments on my new ring".
- This ring is perfect I say why spend thousands when you don't have to? The ring shines perfectly I love this ring!
- Eve's Addiction was wonderful with sending the ring and the ring is beautiful; my daughter waas thrilled with the ring. Thank you, Dorothy
- my girlfriend especially enjoys this ring because of the thickness of the ring band. In the past she has had similar rings but has always been frustrated by the flimsy ring band, but this ring has a nice solid band.
- I absolutely love this ring! I got this as my engagement ring Feb 09 This ring is beautiful and

durable.

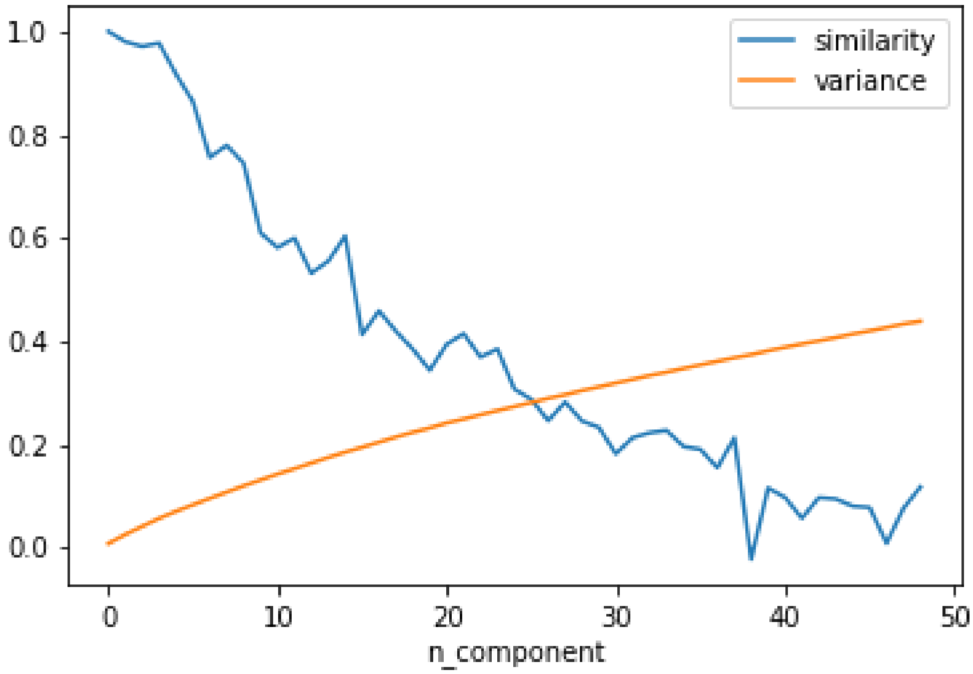
- I love this ring because the flowers go all the way around the ring and it fits my ring finger so if I ever find Mr. Right he doesn't have to buy me an engagement ring, he can just use this as my engagement ring, as I am not into getting into debt before marriage.

After Looking at the plot we can say that n_component can have a value of 20-30 with which it can get a good similarity score without loosing too much of varianec. The number came out to be 30. As below blocks will tell you how the number 30, turns out to be

[10]:

```
# Code below was adapted from labs
def build_graph_data(var_explained,similarity_for_graph):
  # running a loop from 1-50
  for n in range(1,50):
    SVD_model = TruncatedSVD(n_components=n)
    # getting transofrmed vectors
    transformed_x = SVD_model.fit_transform(tfidf_X)

    # comparing consine similarity of document 50 with 47,48,49,51 and 52 and␣
 ↪then
    # averaging the similarity to add the number for a particular n_component to
    # the list similarity_for_graph
    s1 = cosine_similarity([transformed_x[50]], [transformed_x[47]])[0][0]
    s2 = cosine_similarity([transformed_x[50]], [transformed_x[48]])[0][0]
    s3 = cosine_similarity([transformed_x[50]], [transformed_x[49]])[0][0]
    s4 = cosine_similarity([transformed_x[50]], [transformed_x[51]])[0][0]
    s5 = cosine_similarity([transformed_x[50]], [transformed_x[52]])[0][0]

    # Averaging all 5 comparisions to store a value in the list
    similarity_for_graph.append((s1+s2+s3+s4+s5)/5)
    # Calculating Variance explained by this n_component value
    var_explained.append(SVD_model.explained_variance_ratio_.sum())


# Initialise both the lists of similarity and vaeiance
similarity_for_graph=[]
var_explained=[]

# call function to populate the lists
build_graph_data(var_explained,similarity_for_graph)

# Plotting similarity v variance graph
fig = plt.figure()
ax = fig.add_subplot(111)
plt.plot(similarity_for_graph,label='similarity')
plt.plot(var_explained,label='variance')
plt.xlabel("n_component")
plt.legend()
```

```
[10]: <matplotlib.legend.Legend at 0x7f70e5b10b20>
```



3.1 Observations:

I Observed that the system performs really good on the review data set:

Precision:90+% Recall:50-60+%

On all 8 queries which tell that the model learnt the relationships among multiple combination of words used together without training and just based on the TFIDF vector values for each review

3.2 Testing System

To test this system and interact with it you can take a look at the textbox widget provided below where if you enter any query you will get a list of 10 most relavent results

```
[11]: # Display Widget function
def create_app():
  display(text)
  display(button, textbox)
  button.on_click(find_button)

# On Click Fucntion for geting the top 10 results
def find_button(b):
  with textbox:
    if text.value!='':
      # cleaning query string
      query = query_to_line(text.value,vocab)
      # creating vectorizer
      vectorizer = TfidfVectorizer(vocabulary=vocab)
      transformer = TfidfTransformer(norm='l2')
      # Query string to TFID query vector
      query_vector = transformer.fit_transform(vectorizer.
 ↪fit_transform([query]))

      # reducing Query vector to smaller dimension same as documents
      query_vector = SVD_model.transform(query_vector)
      similarities=dict()
      for count, value in enumerate(lsi_x_transformed):
        # Computing cosine similarity
        similarity = cosine_similarity([value], query_vector)
        similarities[count]=similarity

        # Filtering list of documents according to top 10 similarities
        # and saving the results in a DataFrame
        k = {c: sim for c, sim in sorted(similarities.items(), key=lambda item:␣
 ↪item[1])}.keys()
        index = list(k)
        list.reverse(index)
        df = pd.read_csv(filename)
        columns = df.columns
        pd.options.display.max_colwidth = 150
        df = df.iloc[np.array(index),0:3].head(10)
    else:
      query=""
      clear_output(wait=True)
      print('Enter Some Value')
    # Reset the console from old result when the button is clicked
    if query !="":
      clear_output(wait=True)
      print('Transformed query: ',query)
      explained_variance = SVD_model.explained_variance_ratio_.sum()
      print("Sum of explained variance ratio: %d%%" % (int(explained_variance *␣
 ↪100)))
      display(df)


# Best Performing LSI model
SVD_model = TruncatedSVD(n_components=30)
# reducing document vector dimensions
lsi_x_transformed = SVD_model.fit_transform(tfidf_X)

# Widget to enter query and get top 10 results
```

```
textbox = widgets.Output()
button = widgets.Button(description="Get Top 10 results")
text = widgets.Textarea(
value='',
placeholder='Paste enter your query here',
description='Query:',
disabled=False
)
# Crate Widget
create_app()
```

```
Textarea(value='', description='Query:', placeholder='Paste enter your query here')

Button(description='Get Top 10 results', style=ButtonStyle())

Output()
```

4 Part 3 Neural Network Information Retrieval

I used the BERT(pretrained model) to bring in pretrained weights on a large dataset to build a tokenizer for reviews to build tokeized review vectors and compare with the tokenized queries using cosine similarity for retrieving relavant reviews

4.1 Importing Libraries

```
[12]: # Code below was adapted from labs
# Import Few Libraries For BERT Model Block
!pip install transformers
import transformers as ppb

# Import Few Libraries For Feature Extraction Block
import pandas as pd
import torch

# Import Few Libraries For Widget Block
from IPython.display import clear_output
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting transformers
  Downloading transformers-4.27.1-py3-none-any.whl (6.7 MB)
                           6.7/6.7 MB
47.5 MB/s eta 0:00:00
Collecting huggingface-hub<1.0,>=0.11.0
  Downloading huggingface_hub-0.13.2-py3-none-any.whl (199 kB)
                          199.2/199.2 KB
19.8 MB/s eta 0:00:00
```

```
Requirement already satisfied: tqdm>=4.27 in
/usr/local/lib/python3.9/dist-packages (from transformers) (4.65.0)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.9/dist-packages (from transformers) (2022.6.2)
Requirement already satisfied: requests in /usr/local/lib/python3.9/dist-
packages (from transformers) (2.25.1)
Collecting tokenizers!=0.11.3,<0.14,>=0.11.1
  Downloading
tokenizers-0.13.2-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (7.6
MB)
                           7.6/7.6 MB
25.1 MB/s eta 0:00:00
Requirement already satisfied: pyyaml>=5.1 in
/usr/local/lib/python3.9/dist-packages (from transformers) (6.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.9/dist-
packages (from transformers) (23.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.9/dist-
packages (from transformers) (3.9.1)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.9/dist-
packages (from transformers) (1.22.4)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.9/dist-packages (from huggingface-
hub<1.0,>=0.11.0->transformers) (4.5.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.9/dist-packages (from requests->transformers) (2022.12.7)
Requirement already satisfied: chardet<5,>=3.0.2 in
/usr/local/lib/python3.9/dist-packages (from requests->transformers) (4.0.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.9/dist-
packages (from requests->transformers) (2.10)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/usr/local/lib/python3.9/dist-packages (from requests->transformers) (1.26.15)
Installing collected packages: tokenizers, huggingface-hub, transformers
Successfully installed huggingface-hub-0.13.2 tokenizers-0.13.2
transformers-4.27.1
```

4.2 BERT Model

[13]:

```
# BERT Model
model_class, tokenizer_class, pretrained_weights = (ppb.BertModel, ppb.
 ↪BertTokenizer, 'bert-base-uncased')

# Load pretrained model/tokenizer
tokenizer = tokenizer_class.from_pretrained(pretrained_weights)
model = model_class.from_pretrained(pretrained_weights)
```

```
Downloading (…)solve/main/vocab.txt:   0%|          | 0.00/232k [00:00<?, ?B/s]

Downloading (…)okenizer_config.json:   0%|          | 0.00/28.0 [00:00<?, ?B/s]
Downloading (…)lve/main/config.json:   0%|          | 0.00/570 [00:00<?, ?B/s]
```

```
Downloading pytorch_model.bin:   0%|          | 0.00/440M [00:00<?, ?B/s]
```

```
Some weights of the model checkpoint at bert-base-uncased were not used when
initializing BertModel: ['cls.seq_relationship.bias', 'cls.predictions.bias',
'cls.predictions.decoder.weight', 'cls.predictions.transform.LayerNorm.weight',
'cls.seq_relationship.weight', 'cls.predictions.transform.dense.weight',
'cls.predictions.transform.dense.bias',
'cls.predictions.transform.LayerNorm.bias']
- This IS expected if you are initializing BertModel from the checkpoint of a
model trained on another task or with another architecture (e.g. initializing a
BertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing BertModel from the checkpoint of
a model that you expect to be exactly identical (initializing a
BertForSequenceClassification model from a BertForSequenceClassification model).
```

4.3 Feature extraction from Neural Network model

Below is the function that accepts the document

1. Tokenizes it based on BERTokenizer
2. takes only first 512 tokens if length greater than 512 and finds the maximum length among all lists of tokens(reviews)
3. pad all the list of tokens(reviews) having length less than maximum length
4. max these padded lists
5. Create a feature vector USING these padded and masked vectors AND PASSING IT TO THE BERT Model class

```
[14]: def extract_document_features(document):
        with torch.no_grad():
          # Tokenization using BertTokenizer
          tokenized = documents_array.apply((lambda x: tokenizer.encode(x,␣
       ↪add_special_tokens=True)))
          print("Example of First Document: ",documents_array[0])
          print("Example of First tokenized Document: ",tokenized.values[0])
          print("Length of First Document: ",len(documents_array[0].split()))
          print("length of First tokenized Document: ",len(tokenized.values[0]))

          # Computing maximum length and striping all vectors to first 512 values
          max_len = 0
          q = 0
          for i in tokenized.values:

              # BERT only accept maximum 512 values
              if len(i) > 512:
                  temp = tokenized.values[q]
                  tokenized.values[q] = temp[:512]
                  i = tokenized.values[q]
```

```
      print('too much tokenized.values for BERT, only 512 are taken')
    if len(i) > max_len:
      max_len = len(i)
    q += 1


  # padding all doxument vectors to the maximum lenght among the documets
  padded = np.array([i + [0]*(max_len-len(i)) for i in tokenized.values])

  # creating masks for better performance
  attention_mask = np.where(padded != 0, 1, 0)
  attention_mask.shape

  # Converting vectors to tensors
  input_ids = torch.tensor(padded)
  attention_mask = torch.tensor(attention_mask)
  print("Shape of input vector: ",input_ids.shape)

  # creating Document Feature Vector
  last_hidden_states = model(input_ids, attention_mask=attention_mask)
  train_features = last_hidden_states[0][:,0,:].numpy()
 return train_features,max_len
```

[15]:

```
# Converting list of cleaned documents to series of documents for tokenization
documents_array = pd.Series(documents)

train_features,max_len = extract_document_features(documents)
```

```
Example of First Document:  expect like regular size ringbut one look like ring
toy something funy rings ring may cttw round diamond solitaire ring white gold
Example of First tokenized Document:  [101, 5987, 2066, 3180, 2946, 3614, 8569,
2102, 2028, 2298, 2066, 3614, 9121, 2242, 4569, 2100, 7635, 3614, 2089, 14931,
2102, 2860, 2461, 6323, 14017, 29422, 3614, 2317, 2751, 102]
Length of First Document:  22
length of First tokenized Document:  30
Shape of input vector:  torch.Size([200, 79])
```

4.4 Part 3a: NN Model

- Evaluating and testing the model in given queries
- Graph can be seen at the end that interplots average Precision and average Recall for the 8 queries on this dataset just like LSI done earlier

[16]:

```
# Interplot Precision for standard Recall
def InterplotPrecision(p=0.1, Precision=None, Recall=None):
    if p >= 1.0:
        p = 0.9
    # Initialising left and right point of recall
```

```
  Mark = np.zeros(2)
  l = 0
  r = 0
  for i in range(len(Recall)):
   if Recall[i] >= p and Mark[0] == 0:
     l = i
     Mark[0] = 1
   if Recall[i] >= p + 0.1 and Mark[1] == 0:
     r = i
     Mark[1] = 1
  y = max(Precision[l:(r+1)],default=0)
  return y

# compute Recall, Precision
def compute_R_P_F1(re_mark=None, QuRe_ID =None):
 Recall = []
 Precision = []
 F1measure = []
 for i in range(len(re_mark)):
  # for all values in the retrieved result calculate recall and precision
  r = sum(re_mark[:(i+1)])
  Re = r/(len(QuRe_ID))
  Pr = r/(i+1)
  # adding values to respective list
  Recall.append(Re)
  Precision.append(Pr)
 return Recall, Precision, F1measure

def compute_RP_yaxis(Precision=None, Recall=None):
 y_axis = [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]
 for i in range(11):
  pInput = 0.1 * i
  #calling function to find y-axis of Interplot between recall and precision
  y_axis[i] = InterplotPrecision(p=pInput, Precision=Precision, Recall=Recall)
 return y_axis

# Takes a Query string and returns a clean string of tokens as a line
def query_to_line(query,vocab):
 # convert string to lower case
 query_lower = query.lower()
 # Cleaning the query to return clean tokens
 clean = clean_doc(query_lower)
 # only keeping tokens that are in vocab
 clean = [w for w in clean if w in vocab]
 return ' '.join(clean)
```

```
# giving file name and initialising queries list and relavant results list for
# each query
queries_relavant_id = pd.read_csv(filename2)
# relavant results list
rel_id=[]
for i in range(8):
  id=[]
  for j in queries_relavant_id.iloc[:,i]:
    if not pd.isna(j):
      id.append(int(j))
  rel_id.append(id)

# queries
queries=[]
for i in queries_relavant_id.iloc[0,9:17]:
  queries.append(i)

AllRecall = []
AllPrecision = []
result=[]
def␣
 ↪retirieve_results_using_nn_model(train_features,queries,rel_id,AllRecall,AllPrecision,max_l
 ↪
  j=0
  with torch.no_grad():
    for query in queries:
      query_string = query_to_line(query,vocab)
      query_array = pd.Series([query_string])

      # Tokenizing Query Vectors
      query_vector = query_array.apply((lambda x: tokenizer.encode(x,␣
 ↪add_special_tokens=True)))

      # padding all doxument vectors to the maximum lenght among the documets
      padded_query = np.array([i + [0]*(max_len-len(i)) for i in query_vector.
 ↪values])

      # creating masks for better performance
      query_mask = np.where(padded_query != 0, 1, 0)

      # Converting query Vectors to tensors
      query_ids = torch.tensor(padded_query)
      query_mask = torch.tensor(query_mask)
      query_hidden_states = model(query_ids, attention_mask=query_mask)

      # Extracting Features from Query Vector
      query_features = query_hidden_states[0][:,0,:].numpy()
      # Computing cosine similarity
      similarities = cosine_similarity(train_features, query_features)

      # Filtering list of documents according to top 10 similarities
      df = pd.read_csv(filename)
      REVIEW = np.array(df[['Reviews']])
      IDS = np.array(df[['ID']])
      indexes = np.argsort(similarities.flat)[::-1]
      IDS = IDS[indexes].flat[:]

      #Finding out if the result is relavant or not
      re_mark = []
      for i in range(len(indexes)):
        if IDS[i] in rel_id[j]:
          re_mark.append(1)
        else:
          re_mark.append(0)
      # Calculate Recall and Precision scores of all documents corresponding
      # to 6 queries
      Recall, Precision, F1measure = compute_R_P_F1(re_mark=re_mark,␣
  ↪QuRe_ID=rel_id[j])
      Recall = np.array(Recall)
      Precision = np.array(Precision)
      F1measure = np.array(F1measure)


      print('\n' + 'Query%d: '%(j+1) + query)
      print()
      print("Results: ")
      print()
      for i in range(10):
        print("Top " + str(i+1) + ' result: ID: %d '%(IDS[i]),␣
  ↪REVIEW[indexes[i]])
      print()
      print("Recall for query "+str(j+1)+" @1~10: ", np.around(Recall[:10],2))
      print("Precision for query "+str(j+1)+" @1~10: ", np.around(Precision[:
  ↪10],2))

      # save all recall and precision values for each query
      AllRecall.append(Recall)
      AllPrecision.append(Precision)
      j=j+1
  return AllRecall,AllPrecision
```

4.4.1 Calling functions

[17]:

```
AllRecall,AllPrecision =␣
  ↪retirieve_results_using_nn_model(train_features,queries,rel_id,AllRecall,AllPrecision,max_l

# calculating average recall and precision for the 6 queries @1-10
AveRecall = (AllRecall[0] + AllRecall[1] + AllRecall[3] + AllRecall[4] +␣
  ↪AllRecall[5] + AllRecall[6] + AllRecall[7])/8
AvePrecision = (AllPrecision[0] + AllPrecision[2] + AllPrecision[3] +␣
  ↪AllPrecision[4] + AllPrecision[5] + AllPrecision[6] + AllPrecision[7])/8
# printing average recall and precision values @1-10
print()
# printing average recall and precision values @1-10
print("\nAverage Recall and average Precision: ")
print("Average Recall@1~10: ", np.around(AveRecall[:10],2))
print("Average Precision@1~10: ", np.around(AvePrecision[:10],2))
print()

# plot R/P curve for all 8 queries
x_axis = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
colour = ['black','red','orange','blue','purple','green','pink','brown']
for y in range(8):
  y_axis = compute_RP_yaxis(Precision=AllPrecision[y], Recall=AllRecall[y])
  plt.plot(x_axis, y_axis, '-bo', color=colour[y], label="Query%d"%(y+1))
plt.xlim(0, 1)
plt.ylim(0, 1)
plt.xlabel('Average Recall')
plt.ylabel('Average Precision')
plt.title('Recall/Precision Curve')
plt.legend()
plt.show()
```

```
Query1: The ring is a great gift. My friend loves it

Results:

Top 1 result: ID: 26246  ['This was a birthday gift for my 16 YO niece. She
loves the ring and was very happy to have received it.']
Top 2 result: ID: 36164  ['I got the ring as a promise ring for my girlfriend
for Christmas and she loved it. Definitely a great value.']
Top 3 result: ID: 41876  ["I bought this as a gift for a friends birthday and
she loved it. It's a beautifull ring."]
Top 4 result: ID: 45203  ['I got this ring for my birthday and I love it, I
cannot imagine a woman not adoring this ring.']
Top 5 result: ID: 48216  ['I got this ring for my birthday and I love it, I
cannot imagine a woman not adoring this ring.']
Top 6 result: ID: 50650  ['Not only is the ring beautiful, the jeweler was very
accommadating in having the ring reach us in time. We appreciate the care and
quick receiving of the ring. Thank you!']
Top 7 result: ID: 20090  ['Not only is the ring beautiful, the jeweler was very
accommadating in having the ring reach us in time. We appreciate the care and
quick receiving of the ring. Thank you!']
Top 8 result: ID: 58595  ['I absolutely love this ring! I got this as my
engagement ring Feb 09 This ring is beautiful and durable.']
Top 9 result: ID: 28250  ['I wanted to know if this ring is like 2 rings in one,
because this ring is beyond gorgeous, I just love it.']
Top 10 result: ID: 10535  ['I wanted to know if this ring is like 2 rings in
one, because this ring is beyond gorgeous, I just love it.']

Recall for query 1 @1~10:  [0.06 0.12 0.19 0.19 0.19 0.19 0.19 0.19 0.19 0.19]
Precision for query 1 @1~10:  [1.   1.   1.   0.75 0.6  0.5  0.43 0.38 0.33 0.3
]

Query2: horrible bad quality bracelet

Results:

Top 1 result: ID: 3494  ['It is as nice as it looks on the picture. :) I like
it. :)']
Top 2 result: ID: 26246  ['This was a birthday gift for my 16 YO niece. She
loves the ring and was very happy to have received it.']
Top 3 result: ID: 3865  ['What sparkle. It is so pretty and dainty. Just what I
was looking for.']
Top 4 result: ID: 32674  ['THIS ITEM WAS A WONDERFUL SURPRISE. THE QUALITY IS SO
MUCH MORE THAN I COULD HAVE EVER HOPED FOR.']
Top 5 result: ID: 22408  ['The product arrived in a very short period of time
and was perfect. It was described perfectly and was everything I had hoped']
Top 6 result: ID: 56494  ['I bought this ring for my husband and he loved it. I
received it when they said I would and it is a great ring']
Top 7 result: ID: 52867  ['the product arrived in perfect condition but the
shipping is ridiculously slow. i will not order from them again.']
Top 8 result: ID: 41876  ["I bought this as a gift for a friends birthday and
she loved it. It's a beautifull ring."]
Top 9 result: ID: 51396  ['very nice small sized ring I can stack it with other
rings for different looks']
Top 10 result: ID: 44534  ['very nice small sized ring I can stack it with other
rings for different looks']

Recall for query 2 @1~10:  [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Precision for query 2 @1~10:  [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Query3: arrived promptly and happy with the seller

Results:
```

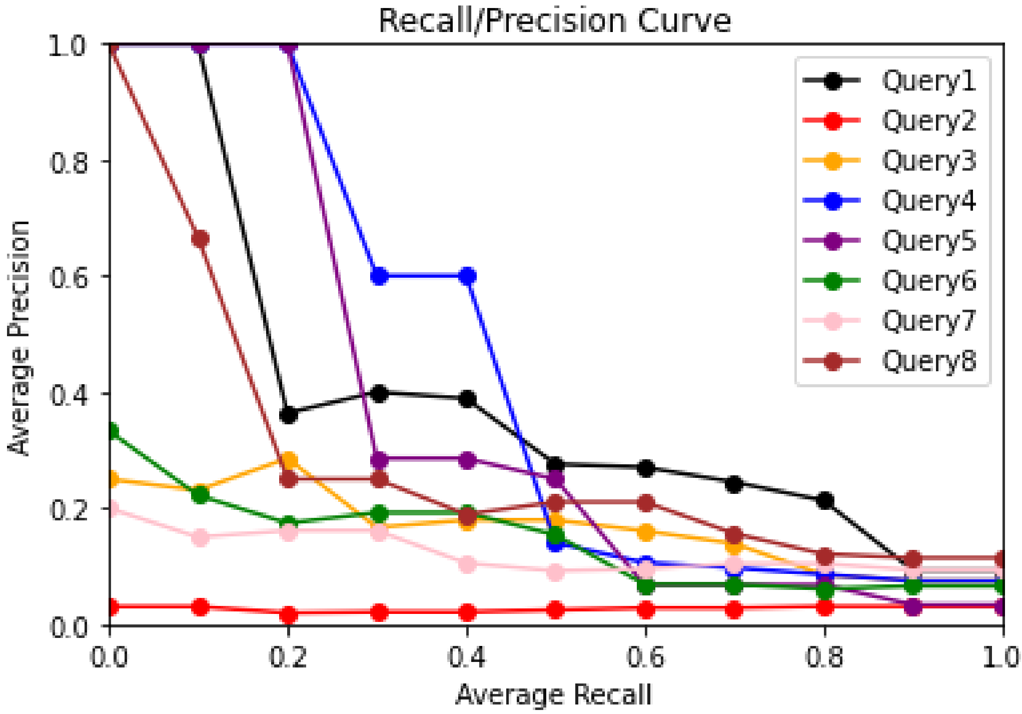
```
Top 1 result: ID: 56494  ['I bought this ring for my husband and he loved it. I
received it when they said I would and it is a great ring']
Top 2 result: ID: 26246  ['This was a birthday gift for my 16 YO niece. She
loves the ring and was very happy to have received it.']
Top 3 result: ID: 41876  ["I bought this as a gift for a friends birthday and
she loved it. It's a beautifull ring."]
Top 4 result: ID: 22408  ['The product arrived in a very short period of time
and was perfect. It was described perfectly and was everything I had hoped']
Top 5 result: ID: 52663  ['I have been told that the ring is very comfortable to
wear and he was quite surprised and please to see the Masonic ring in
titanium.']
Top 6 result: ID: 22058  ["Item was great quality and came promptly. I'm very
happy with it and recommend it unreservedly."]
Top 7 result: ID: 3494  ['It is as nice as it looks on the picture. :) I like
it. :)']
Top 8 result: ID: 36585  ["I am looking forward to wearing them as they sparkle
and catch every eye at my son's wedding on June 30"]
Top 9 result: ID: 52867  ['the product arrived in perfect condition but the
shipping is ridiculously slow. i will not order from them again.']
Top 10 result: ID: 32674  ['THIS ITEM WAS A WONDERFUL SURPRISE. THE QUALITY IS
SO MUCH MORE THAN I COULD HAVE EVER HOPED FOR.']

Recall for query 3 @1~10:  [0.   0.   0.   0.07 0.07 0.07 0.07 0.07 0.14 0.14]
Precision for query 3 @1~10:  [0.   0.   0.   0.25 0.2  0.17 0.14 0.12 0.22 0.2
]

Query4: wear it with casual wear

Results:

Top 1 result: ID: 19852  ['very good for everyday wear or dressing up']
Top 2 result: ID: 53660  ['These are nice to wear when you want something casual
to wear. They are very comfortable.']
Top 3 result: ID: 2134  ['ery suitable for wearing for fashionable occasions.
very dressy']
Top 4 result: ID: 9050  ['The ring is pretty enough, but the metal of the ring
is very insubstantial it pushes in very easily.']
Top 5 result: ID: 52375  ['The message is very positive and it looks very
pretty. I bought it for my aunt as a present and the color is very nice.']
Top 6 result: ID: 3865  ['What sparkle. It is so pretty and dainty. Just what I
was looking for.']
Top 7 result: ID: 30640  ['This lapel pin is the perfect detail to wear your
colors. I plan to wear it on my lapel when I wear a suit. This pinis nice enough
to wear in formal occasions.Wear it with pride!!']
Top 8 result: ID: 52663  ['I have been told that the ring is very comfortable to
wear and he was quite surprised and please to see the Masonic ring in
titanium.']
Top 9 result: ID: 36585  ["I am looking forward to wearing them as they sparkle
```

```
and catch every eye at my son's wedding on June 30"]
Top 10 result: ID: 44126  ['It is so unique and a pleasure to wear. The stones
catch the light and the style is very comfortable to wear.']

Recall for query 4 @1~10:  [0.07 0.14 0.21 0.21 0.21 0.21 0.29 0.29 0.36 0.43]
Precision for query 4 @1~10:  [1.   1.   1.   0.75 0.6  0.5  0.57 0.5  0.56 0.6
]

Query5: i expected better quality. i will return this item

Results:

Top 1 result: ID: 17607  ['This product was hollow, which was not clearly
specified in the item description. It was not what I was expecting and seemed to
be poor quality. I returned the item.']
Top 2 result: ID: 1816  ['The quality and look were not what I had anticipated.
Very flimsy.I would not recommend this item']
Top 3 result: ID: 4375  ['Very impressed with the quality of my item. Delivery
was fast. Would definately buy from this seller again']
Top 4 result: ID: 10758  ['Very impressed with the quality of my item. Delivery
was fast. Would definately buy from this seller again']
Top 5 result: ID: 50019  ['The earrings were just as described the transaction
was smooth and easy and the product was shipped and received in the time frame
that was quoted I am very pleased with this purchase']
Top 6 result: ID: 17944  ['the product i recieved was nice it came in a timley
matter faster than i expected will order this item again']
Top 7 result: ID: 13373  ['The item was not as pictured. It is funky and of poor
quality. The seller did not respond when I contacted him about this.']
Top 8 result: ID: 22058  ["Item was great quality and came promptly. I'm very
happy with it and recommend it unreservedly."]
Top 9 result: ID: 29722  ['I received this Italian horn in pristine condition
and I was completely satisfied with the receiving of this product in a timely
manner.']
Top 10 result: ID: 2780  ['arrived before estimated date. many previous orders
from this seller which always are on time and in excellent condition']

Recall for query 5 @1~10:  [0.2 0.2 0.2 0.2 0.2 0.2 0.4 0.4 0.4 0.4]
Precision for query 5 @1~10:  [1.   0.5  0.33 0.25 0.2  0.17 0.29 0.25 0.22 0.2
]

Query6: looks beautiful. The design is pretty. pefect and color is light

Results:

Top 1 result: ID: 3494  ['It is as nice as it looks on the picture. :) I like
it. :)']
Top 2 result: ID: 44490  ['These are very good quality. They are light weight
and nice small size. Just as described. They look like the picture.']
```

```
Top 3 result: ID: 39932  ['This dainty heart looks absolutely beautiful on. It
picks up the colors of your clothing. It is an amazing price for such a
beautiful pendant.']
Top 4 result: ID: 41876  ["I bought this as a gift for a friends birthday and
she loved it. It's a beautifull ring."]
Top 5 result: ID: 22408  ['The product arrived in a very short period of time
and was perfect. It was described perfectly and was everything I had hoped']
Top 6 result: ID: 44534  ['very nice small sized ring I can stack it with other
rings for different looks']
Top 7 result: ID: 51396  ['very nice small sized ring I can stack it with other
rings for different looks']
Top 8 result: ID: 56865  ['Looks exactly like the picture. Very nice quality. A
must for everyone who is a Tiger fan and owns an Italian Charm Bracelet.']
Top 9 result: ID: 52837  ['I can imagine this sparkling around my girlfriends
tanned toe in the sun. Too bad its winter. But I will be looking forward to. It
should be a pretty sight.']
Top 10 result: ID: 42077  ["This is a solid.beautiful ring. But if you are
expecting the color in rhe picture you will be disappointed. It is barely pink
at all. When I first saw it I thought it was lavendar. It's still pretty but buy
for design not color."]

Recall for query 6 @1~10:  [0.   0.   0.08 0.08 0.08 0.08 0.08 0.08 0.17 0.17]
Precision for query 6 @1~10:  [0.   0.   0.33 0.25 0.2  0.17 0.14 0.12 0.22 0.2
]

Query7: This ring looks nothing like the picture. the diamonds are small and not
very noticeable

Results:

Top 1 result: ID: 51396  ['very nice small sized ring I can stack it with other
rings for different looks']
Top 2 result: ID: 44534  ['very nice small sized ring I can stack it with other
rings for different looks']
Top 3 result: ID: 2185  ['Ring is way too small and looks like a toy when
putting it on. I would not recommend if you want a nice 1/2 carat ring.']
Top 4 result: ID: 39932  ['This dainty heart looks absolutely beautiful on. It
picks up the colors of your clothing. It is an amazing price for such a
beautiful pendant.']
Top 5 result: ID: 36165  ['This ring is alot smaller in person than in pictures,
the pictures make it look like the diamonds are decent size and they are very
small, I was a little disappointed.']
Top 6 result: ID: 22946  ['This ring was a little too small for my ring finger.
It would have been better as a pinky ring.']
Top 7 result: ID: 41319  ['The ring is exactly as pictured and looks very pretty
on my hand. The color of the stones is rich and beautiful.']
Top 8 result: ID: 56865  ['Looks exactly like the picture. Very nice quality. A
must for everyone who is a Tiger fan and owns an Italian Charm Bracelet.']
```

```
Top 9 result: ID: 9050  ['The ring is pretty enough, but the metal of the ring
is very insubstantial it pushes in very easily.']
Top 10 result: ID: 30926  ["Recv'd my ring in a timely manner it looks very
antique would recommend this ring to any garnet lover!"]

Recall for query 7 @1~10:  [0.   0.   0.   0.   0.07 0.07 0.07 0.07 0.07 0.07]
Precision for query 7 @1~10:  [0.   0.   0.   0.   0.2  0.17 0.14 0.12 0.11 0.1
]

Query8: braclet looked just like its picture and is nice quality sterling
silver.

Results:

Top 1 result: ID: 642  ['This medical alert braclet looked just like its picture
and is nice quality sterling silver.']
Top 2 result: ID: 22408  ['The product arrived in a very short period of time
and was perfect. It was described perfectly and was everything I had hoped']
Top 3 result: ID: 3494  ['It is as nice as it looks on the picture. :) I like
it. :)']
Top 4 result: ID: 32674  ['THIS ITEM WAS A WONDERFUL SURPRISE. THE QUALITY IS SO
MUCH MORE THAN I COULD HAVE EVER HOPED FOR.']
Top 5 result: ID: 3865  ['What sparkle. It is so pretty and dainty. Just what I
was looking for.']
Top 6 result: ID: 52867  ['the product arrived in perfect condition but the
shipping is ridiculously slow. i will not order from them again.']
Top 7 result: ID: 39932  ['This dainty heart looks absolutely beautiful on. It
picks up the colors of your clothing. It is an amazing price for such a
beautiful pendant.']
Top 8 result: ID: 265  ['The quality of this item was not up to expectations.The
Top was scratched, the hinges did not line up to the pre-drilled holes and the
staining was inconsistant. If I saw this item in a store I would not have
purchased it.']
Top 9 result: ID: 51396  ['very nice small sized ring I can stack it with other
rings for different looks']
Top 10 result: ID: 44534  ['very nice small sized ring I can stack it with other
rings for different looks']

Recall for query 8 @1~10:  [0.08 0.08 0.15 0.15 0.15 0.15 0.15 0.15 0.15 0.15]
Precision for query 8 @1~10:  [1.   0.5  0.67 0.5  0.4  0.33 0.29 0.25 0.22 0.2
]


Average Recall and average Precision:
Average Recall@1~10:  [0.05 0.07 0.1  0.1  0.11 0.11 0.15 0.15 0.17 0.18]
Average Precision@1~10:  [0.5  0.38 0.42 0.34 0.3  0.25 0.25 0.22 0.24 0.22]
```

```
<ipython-input-17-92211f60f055>:19: UserWarning: color is redundantly defined by
the 'color' keyword argument and the fmt string "-bo" (-> color='b'). The
keyword argument will take precedence.
  plt.plot(x_axis, y_axis, '-bo', color=colour[y], label="Query%d"%(y+1))
```



4.4.2 Observations

My Observation is that LSI Model is working far better than the NN(BERT) model. In terms of precision and recall we can clearly see that the LSI model learnt about the words in the reviews dataset better than the neural network model without training.

Synonymy refers to the fact that different words can have the same or similar meanings, while polysemy refers to the fact that the same word can have multiple meanings. These issues can make it difficult to accurately represent the meaning of a document or to match documents based on their semantic content.

LSI solves this problem by using singular value decomposition (SVD) to transform the original high-dimensional space of review vectors into a lower-dimensional space that captures the underlying topics or themes present in the data. This lower-dimensional representation allows LSI to effectively handle synonymy and polysemy by identifying patterns in the relationships between words and documents, rather than relying solely on individual word meanings.

Where Neural Networks on the other hand rely on word embeddings that were learnt in some other context compared to this small dataset.

This is the reason LSI gets a precision and recall of 90+ and 60+ respectively whereas NN BERT model gets a precision and recall of 20+ and 15+ %

4.5 Part 3b: Widgets for interacting with the models LSI and Neural Network(BERT)

Below is the widget to enter your own query and retieve top 10 relavant results from the given corpus which are reviews for jewellery. From the Drop Down You can choose between LSI and Neural Network Model to Retrive th results. Here, LSI is better than NN by a lot and it can be tested with this widget.

[18]:

```
# Display Widget function
def create_app():
  display(text)
  display(button, textbox)
  button.on_click(find_button)

# On Click Fucntion for geting the top 10 results
def find_button(b):
  with torch.no_grad():
    # cleaning query string
    query_string = query_to_line(text.value,vocab)
    with textbox:
      if text.value!='':
        if dropvals.value==1:
          # Best performing LSI model
          LSI_SVD_model = TruncatedSVD(n_components=30)
          lsi_x_transformed = LSI_SVD_model.fit_transform(tfidf_X)

          # creating vectorizer
          vectorizer = TfidfVectorizer(vocabulary=vocab)
          transformer = TfidfTransformer(norm='l2')
          # Query string to TFID query vector
          query_vector = transformer.fit_transform(vectorizer.
 ↪fit_transform([query_string]))

          # reducing Query vector to smaller dimension same as documents
          query_vector = LSI_SVD_model.transform(query_vector)
          X=lsi_x_transformed
          query_features=query_vector
        else:
          # Tokenizing Query Vectors
          query_vector = query_array.apply((lambda x: tokenizer.encode(x,␣
 ↪add_special_tokens=True)))

          # padding all doxument vectors to the maximum lenght among the␣
 ↪documets
          padded_query = np.array([i + [0]*(max_len-len(i)) for i in␣
 ↪query_vector.values])
```

```
          # creating masks for better performance
          query_mask = np.where(padded_query != 0, 1, 0)

          # Converting query Vectors to tensors
          query_ids = torch.tensor(padded_query)
          query_mask = torch.tensor(query_mask)
          query_hidden_states = model(query_ids, attention_mask=query_mask)

          # Extracting Features from Query Vector
          query_features = query_hidden_states[0][:,0,:].numpy()
          X=train_features
      else:
        clear_output(wait=True)
        print('Enter Some Value')
      # Reset the console from old result when the button is clicked
      if text.value !='':
        similarities=dict()
        print(dropvals.description)
        for count, value in enumerate(X):
          # Computing cosine similarity
          similarity = cosine_similarity([value], query_features)
          similarities[count]=similarity

          # Filtering list of documents according to top 10 similarities
          # and saving the results in a DataFrame
          k = {c: sim for c, sim in sorted(similarities.items(), key=lambda␣
 ↪item: item[1])}.keys()
          index = list(k)
          list.reverse(index)
          df = pd.read_csv(filename)
          columns = df.columns
          pd.options.display.max_colwidth = 150
          df = df.iloc[np.array(index),0:3].head(10)
        clear_output(wait=True)
        print('Model: ',dropvals.label)
        print('Transformed query: ',query_string)
        explained_variance = SVD_model.explained_variance_ratio_.sum()
        print("Sum of explained variance ratio: %d%%" % (int(explained_variance␣
 ↪* 100)))
        display(df)

# Widget to enter query and get top 10 results
textbox = widgets.Output()
button = widgets.Button(description="Get Top 10 results")
text = widgets.Textarea(
value='',
placeholder='Paste enter your query here',
description='Query:',
disabled=False
)

dropvals = widgets.Dropdown(
    options=[('LSI', 1), ('Neural Network', 2)],
    value=1,
    description='Model:',
)
display(dropvals)
# Crate Widget
create_app()
```

```
Dropdown(description='Model:', options=(('LSI', 1), ('Neural Network', 2)),␣
 ↪value=1)
```

```
Textarea(value='', description='Query:', placeholder='Paste enter your query␣
 ↪here')
```

```
Button(description='Get Top 10 results', style=ButtonStyle())
```

```
Output()
```

5 Part 4: Topic Modeling using gensim

In this section First I select a query out of the 8 given queries to get top 50 matching reviews and based on these reviews build topic modeling model using gensim. After basic preprocessing step on the already cleaned reviews I Evaluste the GENSIM model and visualise it using pyLDAvis

This has few steps:

1. Getting top 50 reviews relavant to the defined query
2. Stemming already cleaned reviews
3. Applying Gensim topic model
4. Evaluating the Model
5. Visualising the Model
6. Interacting with the model using widgets

In the Last part I develop a Widget to choose out of the queries then get top 50 relavant documents for the selected query and then evaluate and visualise the model juts as we did in earlier steps.

5.1 importing libraries

[19]:
```
# Importing Libraries for Preprocessing block
import spacy
# Importing Few Libraries For Creating corpus and Dictionary block
import gensim
import gensim.corpora as corpora
```

```python
# Import for Evaluation Block
from gensim.models import CoherenceModel

# Import For Visualisation Block
!pip install pyLDAvis
import pyLDAvis
import pyLDAvis.gensim as gensimvis

# Import for Widget BLock
import functools
from IPython.display import display
from ipywidgets import Button, HBox, VBox,widgets
import ipywidgets
```

```
/usr/local/lib/python3.9/dist-packages/torch/cuda/__init__.py:497: UserWarning:
Can't initialize NVML
  warnings.warn("Can't initialize NVML")
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting pyLDAvis
  Downloading pyLDAvis-3.4.0-py3-none-any.whl (2.6 MB)
                           2.6/2.6 MB
59.6 MB/s eta 0:00:00
Requirement already satisfied: gensim in /usr/local/lib/python3.9/dist-
packages (from pyLDAvis) (3.6.0)
Collecting joblib>=1.2.0
  Downloading joblib-1.2.0-py3-none-any.whl (297 kB)
                          298.0/298.0 KB
26.1 MB/s eta 0:00:00
Requirement already satisfied: numexpr in /usr/local/lib/python3.9/dist-
packages (from pyLDAvis) (2.8.4)
Requirement already satisfied: pandas>=1.3.4 in /usr/local/lib/python3.9/dist-
packages (from pyLDAvis) (1.4.4)
Requirement already satisfied: setuptools in /usr/local/lib/python3.9/dist-
packages (from pyLDAvis) (63.4.3)
Requirement already satisfied: numpy>=1.22.0 in /usr/local/lib/python3.9/dist-
packages (from pyLDAvis) (1.22.4)
Collecting funcy
  Downloading funcy-1.18-py2.py3-none-any.whl (33 kB)
Requirement already satisfied: scikit-learn>=1.0.0 in
/usr/local/lib/python3.9/dist-packages (from pyLDAvis) (1.2.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.9/dist-packages
(from pyLDAvis) (1.10.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.9/dist-packages
(from pyLDAvis) (3.1.2)
Requirement already satisfied: python-dateutil>=2.8.1 in
/usr/local/lib/python3.9/dist-packages (from pandas>=1.3.4->pyLDAvis) (2.8.2)
```

```
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.9/dist-
packages (from pandas>=1.3.4->pyLDAvis) (2022.7.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.9/dist-packages (from scikit-learn>=1.0.0->pyLDAvis)
(3.1.0)
Requirement already satisfied: smart-open>=1.2.1 in
/usr/local/lib/python3.9/dist-packages (from gensim->pyLDAvis) (6.3.0)
Requirement already satisfied: six>=1.5.0 in /usr/local/lib/python3.9/dist-
packages (from gensim->pyLDAvis) (1.15.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.9/dist-
packages (from jinja2->pyLDAvis) (2.1.2)
Installing collected packages: funcy, joblib, pyLDAvis
  Attempting uninstall: joblib
    Found existing installation: joblib 1.1.1
    Uninstalling joblib-1.1.1:
      Successfully uninstalled joblib-1.1.1
ERROR: pip's dependency resolver does not currently take into account all

the packages that are installed. This behaviour is the source of the following

dependency conflicts.

pandas-profiling 3.2.0 requires joblib~=1.1.0, but you have joblib 1.2.0 which

is incompatible.

Successfully installed funcy-1.18 joblib-1.2.0 pyLDAvis-3.4.0
```

5.2 Part 4 a: Getting top 50 relavant documents for the Second query for topic modeling

Here I predefine a query for retrieving the top 50 results.

Below you will find a block where i have created a widget where you select out of 8 options to retrieve the top 50 results

These top 50 results become the corpus for topic modeling.

5.2.1 Defining the function to retrive results based on NN model

[20]:

```
class Interaction:

  def __init__(self,topn,X,model,tokenizer):
    self.data=[]
    self.train_features = X
    self.model=model
    self.tokenizer=tokenizer
    self.topn=topn
    self.index=1

    self.rel_id=rel_id
    self.queries=queries

    # dictionary for queries
    self.d = {"Queries":queries}

    # class variable for showing options of queries
    self.df = pd.DataFrame(data=self.d,index=[1,2,3,4,5,6,7,8])

  # On Click Functions from obc1 to obc 8 for Buttons for each option of query
  def obc1(self,data,button):
      self.data=[]
      data = self.dis(0,data)
      self.data=data
      return data

  def obc2(self,data,button):
      self.data=[]
      data = self.dis(1,data)
      self.data=data
      return data

  def obc3(self,data,button):
      self.data=[]
      data = self.dis(2,data)
      self.data=data
      return data

  def obc4(self,data,button):
      self.data=[]
      data = self.dis(3,data)
      self.data=data
      return data

  def obc5(self,data,button):
      self.data = self.dis(4,data)
      self.data=data
      return data

  def obc6(self,data,button):
      self.data=[]
      data = self.dis(5,data)
      self.data=data
      return data

  def obc7(self,data,button):
      self.data = self.dis(6,data)
      self.data=data
      return data

  def obc8(self,data,button):
      self.data=[]
      data = self.dis(7,data)
      self.data=data
      return data

  def create_widget(self):
    self.button1 = widgets.Button(description='select 1')
    self.button1.on_click(functools.partial(self.obc1, self.data))

    self.button2 = widgets.Button(description='select 2')
    self.button2.on_click(functools.partial(self.obc2, self.data))

    self.button3 = widgets.Button(description='select 3')
    self.button3.on_click(functools.partial(self.obc3, self.data))

    self.button4 = widgets.Button(description='select 4')
    self.button4.on_click(functools.partial(self.obc4, self.data))

    self.button5 = widgets.Button(description='select 5')
    self.button5.on_click(functools.partial(self.obc5, self.data))

    self.button6 = widgets.Button(description='select 6')
    self.button6.on_click(functools.partial(self.obc6, self.data))

    self.button7 = widgets.Button(description='select 7')
    self.button7.on_click(functools.partial(self.obc7, self.data))

    self.button8 = widgets.Button(description='select 8')
    self.button8.on_click(functools.partial(self.obc8, self.data))

  def dis(self,i,data):
    print('hi')
    self.index=i
    clear_output(wait=False)
    self.show_widget()
    return self.get_results(self.df.iloc[i,:][0],data)

  def show_widget(self):
    display(HBox([VBox([widgets.Button(description=''),self.button1,self.
  ↪button2,self.button3,self.button4,self.button5,self.button6,self.
  ↪button7,self.button8]),ipywidgets.
                      HTML(self.df.style.set_table_attributes('class="table"').
  ↪render())]))
```

```
def get_results(self,b,data):
 with torch.no_grad():
  print()
  print("Selected Query: ",b)
  print("Relavant Results: ")

  query_string = query_to_line(b,vocab)
  query_array = pd.Series([query_string])

  # Tokenizing Query Vectors
  query_vector = query_array.apply((lambda x: self.tokenizer.encode(x, add_special_tokens=True)))

  # padding all doxument vectors to the maximum lenght among the documets
  padded_query = np.array([i + [0]*(max_len-len(i)) for i in query_vector.values])

  # creating masks for better performance
  query_mask = np.where(padded_query != 0, 1, 0)

  # Converting query Vectors to tensors
  query_ids = torch.tensor(padded_query)
  query_mask = torch.tensor(query_mask)
  query_hidden_states = self.model(query_ids, attention_mask=query_mask)

  # Extracting Features from Query Vector
  query_features = query_hidden_states[0][:,0,:].numpy()

  similarities=dict()
  for count, value in enumerate(self.train_features):

   # Computing similarity between query vector and document vectors
   similarity = cosine_similarity([value], query_features)
   similarities[count]=similarity

   # Getting indexes of similarities to fetch most relavant documents
   k = {c: sim for c, sim in sorted(similarities.items(), key=lambda item: item[1])}.keys()
   index = list(k)
   list.reverse(index)
  csv_data = pd.read_csv(filename)
  REVIEW = np.array(csv_data[['Reviews']])
  IDS = np.array(csv_data[['ID']])
  indexes = IDS = IDS[index].flat[:]
  csv_data = csv_data.iloc[np.array(index),0:2].head(self.topn)
  pd.options.display.max_colwidth = 150
  csv_data.reset_index(drop=True,inplace=True)
```

```
      display(csv_data)
      data.clear()
      for i in range(self.topn):
        data.append(REVIEW[index[i]][0])
      return data

  def query_to_line(self,query,vocab):
    # load the docs
    tokens = query.lower()
    clean = clean_doc(tokens)
    clean = [w for w in clean if w in vocab]
    return ' '.join(clean)
```

5.2.2 Calling the functions

```
[21]: # BERT Model
model_class, tokenizer_class, pretrained_weights = (ppb.BertModel, ppb.
 ↪BertTokenizer, 'bert-base-uncased')

# Load pretrained model/tokenizer
tokenizer = tokenizer_class.from_pretrained(pretrained_weights)
model = model_class.from_pretrained(pretrained_weights)


data=[]
# Interaction Class object
interaction = Interaction(50,train_features,model,tokenizer)
data = interaction.get_results(interaction.queries[1],data)
```

```
Some weights of the model checkpoint at bert-base-uncased were not used when
initializing BertModel: ['cls.seq_relationship.bias', 'cls.predictions.bias',
'cls.predictions.decoder.weight', 'cls.predictions.transform.LayerNorm.weight',
'cls.seq_relationship.weight', 'cls.predictions.transform.dense.weight',
'cls.predictions.transform.dense.bias',
'cls.predictions.transform.LayerNorm.bias']
- This IS expected if you are initializing BertModel from the checkpoint of a
model trained on another task or with another architecture (e.g. initializing a
BertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing BertModel from the checkpoint of
a model that you expect to be exactly identical (initializing a
BertForSequenceClassification model from a BertForSequenceClassification model).


Selected Query:  horrible bad quality bracelet
Relavant Results:

       ID  \
0    3494
```

1	26246
2	3865
3	32674
4	22408
5	56494
6	52867
7	41876
8	51396
9	44534
10	37896
11	52663
12	36585
13	216
14	265
15	642
16	53409
17	22058
18	52837
19	11087
20	9050
21	44490
22	56830
23	56830
24	56830
25	17442
26	36164
27	39932
28	2780
29	25378
30	58481
31	6158
32	57123
33	2185
34	19944
35	44489
36	45203
37	48216
38	54748
39	19852
40	11247
41	56865
42	52375
43	30926
44	15959
45	17607
46	17944
47	30773
48	42077

```
49  10209

                                                                                ␣
↪                                                                 Reviews
0                                                                               ␣
↪             It is as nice as it looks on the picture. :) I like it. :)
1                                                 This was a birthday gift for␣
↪my 16 YO niece. She loves the ring and was very happy to have received it.
2                                                                               ␣
↪  What sparkle. It is so pretty and dainty. Just what I was looking for.
3                                                       THIS ITEM WAS A␣
↪WONDERFUL SURPRISE. THE QUALITY IS SO MUCH MORE THAN I COULD HAVE EVER HOPED␣
↪FOR.
4                           The product arrived in a very short period of time␣
↪and was perfect. It was described perfectly and was everything I had hoped
5                                          I bought this ring for my husband and␣
↪he loved it. I received it when they said I would and it is a great ring
6                                       the product arrived in perfect condition␣
↪but the shipping is ridiculously slow. i will not order from them again.
7                                                                I bought this␣
↪as a gift for a friends birthday and she loved it. It's a beautifull ring.
8                                                                          very␣
↪nice small sized ring I can stack it with other rings for different looks
9                                                                          very␣
↪nice small sized ring I can stack it with other rings for different looks
10                  my only wish on this ring is- I wish the cut potrion went␣
↪all the way around the ring. Other than that a great very comfortable ring.
11                  I have been told that the ring is very comfortable to wear␣
↪and he was quite surprised and please to see the Masonic ring in titanium.
12                                                I am looking forward to␣
↪wearing them as they sparkle and catch every eye at my son's wedding on June 30
13  This rings looks nothing like the picture at all! The stones are so small I␣
↪can barely even tell they are stones, then the ring is so thin and sma…
14  The quality of this item was not up to expectations.The Top was scratched,␣
↪the hinges did not line up to the pre-drilled holes and the staining wa…
15                                                           This medical alert␣
↪braclet looked just like its picture and is nice quality sterling silver.
16                                                  Although the picture looks␣
↪like metal beads and description states sterling silver, these are pearls.
17                                                        Item was great quality␣
↪and came promptly. I'm very happy with it and recommend it unreservedly.
18  I can imagine this sparkling around my girlfriends tanned toe in the sun.␣
↪Too bad its winter. But I will be looking forward to. It should be a pre…
19                      its what i wanted :) but its not my favorite piercing of␣
↪mine but i have to wear the bioplast cuz i break out with certain metals
```

```
20                                                   The ring is pretty enough,␣
 ↪but the metal of the ring is very insubstantial it pushes in very easily.
21                                 These are very good quality. They are light␣
 ↪weight and nice small size. Just as described. They look like the picture.
22                 I really liked these earrings. However, i agree with one of␣
 ↪the earlier reviews that I thought they would have been a little bigger.\n
23                 I really liked these earrings. However, i agree with one of␣
 ↪the earlier reviews that I thought they would have been a little bigger.\n
24                 I really liked these earrings. However, i agree with one of␣
 ↪the earlier reviews that I thought they would have been a little bigger.\n
25             Eve's Addiction was wonderful with sending the ring and the ring␣
 ↪is beautiful; my daughter waas thrilled with the ring. Thank you, Dorothy
26                                           I got the ring as a promise ring␣
 ↪for my girlfriend for Christmas and she loved it. Definitely a great value.
27         This dainty heart looks absolutely beautiful on. It picks up the␣
 ↪colors of your clothing. It is an amazing price for such a beautiful pendant.
28                               arrived before estimated date. many previous␣
 ↪orders from this seller which always are on time and in excellent condition
29  My fiance and I looked at many different rings and I fell in love with this␣
 ↪ring, it was everything I wanted in an engagement ring! Now I have bee…
30                                                                        my␣
 ↪wife loves the ring, it was a great gift. extremelly cheap and high quality.
31                   Wonderful shopping experience I purchased the item for␣
 ↪holiday presents and the whole order came quickly and in wonderful condition.
32                                  Very disappointed in the appearance and␣
 ↪quality of the bracelet and its definitely not worth $45.00 - not even close.
33                                Ring is way too small and looks like a toy␣
 ↪when putting it on. I would not recommend if you want a nice 1/2 carat ring.
34                                                                               ␣
 ↪        am very pleased with my purchase, speedy shipping will use again
35                                             I just got these yesterday as a␣
 ↪Christmas gift- so far they look just like the picture and seem very nice.
36                                                         I got this ring for␣
 ↪my birthday and I love it, I cannot imagine a woman not adoring this ring.
37                                                         I got this ring for␣
 ↪my birthday and I love it, I cannot imagine a woman not adoring this ring.
38                           Item arrived extremely damaged in several places.␣
 ↪Not packaged well had to send it back. Very disappointed with the quality.
39                                                                               ␣
 ↪                              very good for everyday wear or dressing up
40                                                                               ␣
 ↪    I have always wanted a claddaugh ring.This price was great.I love it
41                          Looks exactly like the picture. Very nice quality. A␣
 ↪must for everyone who is a Tiger fan and owns an Italian Charm Bracelet.
42                                The message is very positive and it looks very␣
 ↪pretty. I bought it for my aunt as a present and the color is very nice.
```

```
43                                                  Recv'd my ring in a timely␣
 ↪manner it looks very antique would recommend this ring to any garnet lover!
44  I ordered a ring that stated Toe Ring" in the description. The ring came␣
 ↪very quickly but was not a toe ring. I don't know anyone with a size 8 to…
45  This product was hollow, which was not clearly specified in the item␣
 ↪description. It was not what I was expecting and seemed to be poor quality. I…
46                                            the product i recieved was nice it␣
 ↪came in a timley matter faster than i expected will order this item again
47  Nice ring for the an inexpensive ring, but one stone was missing. It was not␣
 ↪worth returning it as you would pay more for postage than what the ri…
48  This is a solid.beautiful ring. But if you are expecting the color in rhe␣
 ↪picture you will be disappointed. It is barely pink at all. When I first…
49                                                                          ␣
 ↪This works perfectly for my rings and my wider rings - both narrow and wide.
```

5.2.3 Preprocessing Data

The corpus created above is raw and needs some preprocessing like cleaning, lemmatization, stop word removal,punctuation removal etc.

[22]:

```python
# accept a doc and return all the cleaned list of words
def clean_doc(doc):
        # split into document text by white space
        words = doc.split()
        # remove punctuation from each token
        table = str.maketrans('', '', string.punctuation)
        words = [w.translate(table) for w in words]
        # remove remaining words that are not alphabetic
        words = [word for word in words if word.isalpha()]
        # filter out stop words
        stop_words = set(stopwords.words('english'))
        words = [w for w in words if not w in stop_words]
        # filter out short tokens
        words = [word for word in words if len(word) > 2]
        return words

# clean_doc('i received my ring and was a little disappointed that the ring is␣
 ↪not completely blue (like the picture shows). It looks like I got a blue␣
 ↪flower with green leaves. So it makes the ring look blue and green. Very␣
 ↪small ring. Not worth $6.99 but more like $3.')

# sentence of sentences to list of list of words
data_words = []
for s in data:
  s = clean_doc(s)
  cleaned_doc = [w for w in s if w in vocab]
  data_words.append(cleaned_doc)
```

```
# [['I', 'got', 'the', 'ring', 'as', 'a', 'promise', 'ring', 'for', 'my',␣
↪'girlfriend', 'for', 'Christmas', 'and', 'she', 'loved', 'it.',␣
↪'Definitely', 'a', 'great', 'value.']
```

[23]:

```
# Lemmatization function

def lemmatization(texts, allowed_postags=['NOUN', 'ADJ', 'VERB', 'ADV']):
    """https://spacy.io/api/annotation"""
    texts_out = []
    for sent in texts:
        doc = nlp(" ".join(sent))
        texts_out.append([token.lemma_ for token in doc if token.pos_ in␣
↪allowed_postags])
    return texts_out
```

[24]:

```
#Call functions for cleaning lemmatization of text

# Initialize spacy 'en' model, keeping only tagger component (for efficiency)
# python3 -m spacy download en
nlp = spacy.load('en_core_web_sm', disable=['parser', 'ner'])

# Do lemmatization keeping only noun, adj, vb, adv
data_lemmatized = lemmatization(data_words, allowed_postags=['NOUN', 'ADJ',␣
↪'VERB', 'ADV'])

print("Cleaned and lemmarized tokens of first docuemnt: ",data_lemmatized[:
↪1][0])
```

```
Cleaned and lemmarized tokens of first docuemnt:  ['nice', 'look', 'picture']
```

5.2.4 Creating corpus and Dictionary

[25]:

```
# below code was adapted from labs
# Create Dictionary
id2word = corpora.Dictionary(data_lemmatized)

# Term Document Frequency
corpus = [id2word.doc2bow(text) for text in data_lemmatized]

c=[]
for (i,j) in corpus[:1][0]:
  c.append((id2word.get(i),j))
print("Cleaned and lemmarized tokens of first docuemnt: ",data_lemmatized[:
↪1][0])
print("Bag-of-words (BoW) format of first document: ",c)
```

```
Cleaned and lemmarized tokens of first docuemnt:  ['nice', 'look', 'picture']
```

```
Bag-of-words (BoW) format of first document:  [('look', 1), ('nice', 1),
('picture', 1)]
```

5.2.5 Applying Gensim Model

5.2.6 Gensim Model

For This i used the gensim library to get the model. Here i have to mention few attributes I like num_topics, update_every, chunksize, passes etc for all these i used multiple values and tried and tested them to come up with some values which i like.

I chose 3 topics to represent sad, staisfied, happy type of reviews

```
[26]: # below code was adapted from https://www.machinelearningplus.com/nlp/
       ↪topic-modeling-gensim-python/
      # Gensim LDA model
      lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                                  id2word=id2word,
                                                  num_topics=3,
                                                  random_state=100,
                                                  update_every=3,
                                                  chunksize=4,
                                                  passes=2,
                                                  alpha='auto',
                                                  per_word_topics=True)
```

```
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
```

5.2.7 Evaluating Model

I used Perplexity and Coherence Score to evaluate the model where a good model has high Coherence Score and

[27]:

```
# code was adapted from https://www.youtube.com/watch?
 ↪v=nNvPvvuPnGs&t=1s&ab_channel=RitheshSreenivasan
# Compute Perplexity
print('\nPerplexity: ', lda_model.log_perplexity(corpus))  # a measure of how␣
 ↪good the model is. lower the better.

# Compute Coherence Score
coherence_model_lda = CoherenceModel(model=lda_model, texts=data_lemmatized,␣
 ↪dictionary=id2word, coherence='c_v')
coherence_lda = coherence_model_lda.get_coherence()
print('\nCoherence Score: ', coherence_lda)
```

```
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
```

```
Perplexity:  -5.716496944688342

Coherence Score:  0.5608393950892389
```

5.2.8 Visualising Model

[28]:

```
# code was adapted from https://www.youtube.com/watch?
 ↪v=nNvPvvuPnGs&t=1s&ab_channel=RitheshSreenivasan
# Visualize the topics

vis_data = gensimvis.prepare(lda_model, corpus, id2word, sort_topics=False)
pyLDAvis.display(vis_data)
```

```
/usr/local/lib/python3.9/dist-packages/pyLDAvis/_prepare.py:243: FutureWarning:
In a future version of pandas all arguments of DataFrame.drop except for the
```

```
argument 'labels' will be keyword-only.
  default_term_info = default_term_info.sort_values(
```

```
[28]: <IPython.core.display.HTML object>
```

5.2.9 Observations

As Above I have selected default query which query number 2 out of the 8 options to bring top 50 reviews, then based on those reviews a corpus is built and then using Gensim model Topic Modeling will be performed which will accept the number of topics you want to model out of the corpus. Here i have choosen 3 topic as reviews can be represented as bad, satisfied, happy categories and we can see that here in the visualisation for the default run.

The topics for every query are segregated clearly in the new space of principal components defined internally by PCA based on the vectors of words defined internally.

Here if you see individual topics

Topic 3: Represents 'Happy' customer where you can see words like **love**, **great**, **ring**, **perfect**, **adore** etc.

Topic 2: represents 'Sad' customer where you can see words like **expect**, **barely**, **disappointed**, **return**, **worth** etc

Topic 1: represents 'satisfied' customer where you can see words like **think**, **really**, **quality**, **price**, **think** etc

Therefore, My Model Did an OK job in segregating the corpus into topics based on the words being used in them.

5.3 Part 4b: Interactive tool to get top 50 relavant reviews of the selected query

Below is the tool to select among queries for retrieving top 50 results to build a corpus out of those 50 documents and based on BERT model create a Topic Model in the subsequent cell

- Execute below cell to run a widget to select out of the query options
- Run the next cell to build an LDA model, evaluate it and then visualise it.
- If no query is selected by default it will select the second query and perform topic modeling on the top 50 results from that query

```
[29]: # BERT Model
model_class, tokenizer_class, pretrained_weights = (ppb.BertModel, ppb.
 ↪BertTokenizer, 'bert-base-uncased')

# Load pretrained model/tokenizer
tokenizer = tokenizer_class.from_pretrained(pretrained_weights)
model = model_class.from_pretrained(pretrained_weights)

# Interaction Class object
interaction = Interaction(50,train_features,model,tokenizer)
# Create Widget on the console
```

```
interaction.create_widget()
interaction.show_widget()
if not len(interaction.data)>0:
  interaction.data=interaction.dis(1,[])
```

```
HBox(children=(VBox(children=(Button(style=ButtonStyle()),␣
 ↪Button(description='select 1', style=ButtonStyle())…
```

```
Selected Query:  horrible bad quality bracelet
Relavant Results:

       ID  \
0    3494
1   26246
2    3865
3   32674
4   22408
5   56494
6   52867
7   41876
8   51396
9   44534
10  37896
11  52663
12  36585
13    216
14    265
15    642
16  53409
17  22058
18  52837
19  11087
20   9050
21  44490
22  56830
23  56830
24  56830
25  17442
26  36164
27  39932
28   2780
29  25378
30  58481
31   6158
32  57123
33   2185
34  19944
```

```
35  44489
36  45203
37  48216
38  54748
39  19852
40  11247
41  56865
42  52375
43  30926
44  15959
45  17607
46  17944
47  30773
48  42077
49  10209
```

```
                                                                                  ␣
 ↪                                                                     Reviews
0                                                                                 ␣
 ↪             It is as nice as it looks on the picture. :) I like it. :)
1                                               This was a birthday gift for␣
 ↪my 16 YO niece. She loves the ring and was very happy to have received it.
2                                                                                 ␣
 ↪  What sparkle. It is so pretty and dainty. Just what I was looking for.
3                                                      THIS ITEM WAS A␣
 ↪WONDERFUL SURPRISE. THE QUALITY IS SO MUCH MORE THAN I COULD HAVE EVER HOPED␣
 ↪FOR.
4                           The product arrived in a very short period of time␣
 ↪and was perfect. It was described perfectly and was everything I had hoped
5                                          I bought this ring for my husband and␣
 ↪he loved it. I received it when they said I would and it is a great ring
6                                       the product arrived in perfect condition␣
 ↪but the shipping is ridiculously slow. i will not order from them again.
7                                                                I bought this␣
 ↪as a gift for a friends birthday and she loved it. It's a beautifull ring.
8                                                                          very␣
 ↪nice small sized ring I can stack it with other rings for different looks
9                                                                          very␣
 ↪nice small sized ring I can stack it with other rings for different looks
10                      my only wish on this ring is- I wish the cut potrion went␣
 ↪all the way around the ring. Other than that a great very comfortable ring.
11                   I have been told that the ring is very comfortable to wear␣
 ↪and he was quite surprised and please to see the Masonic ring in titanium.
12                                              I am looking forward to␣
 ↪wearing them as they sparkle and catch every eye at my son's wedding on June 30
13  This rings looks nothing like the picture at all! The stones are so small I␣
 ↪can barely even tell they are stones, then the ring is so thin and sma…
```

```
14  The quality of this item was not up to expectations.The Top was scratched,␣
 ↪the hinges did not line up to the pre-drilled holes and the staining wa…
15                                                           This medical alert␣
 ↪braclet looked just like its picture and is nice quality sterling silver.
16                                                  Although the picture looks␣
 ↪like metal beads and description states sterling silver, these are pearls.
17                                                        Item was great quality␣
 ↪and came promptly. I'm very happy with it and recommend it unreservedly.
18  I can imagine this sparkling around my girlfriends tanned toe in the sun.␣
 ↪Too bad its winter. But I will be looking forward to. It should be a pre…
19                      its what i wanted :) but its not my favorite piercing of␣
 ↪mine but i have to wear the bioplast cuz i break out with certain metals
20                                                   The ring is pretty enough,␣
 ↪but the metal of the ring is very insubstantial it pushes in very easily.
21                                 These are very good quality. They are light␣
 ↪weight and nice small size. Just as described. They look like the picture.
22                 I really liked these earrings. However, i agree with one of␣
 ↪the earlier reviews that I thought they would have been a little bigger.\n
23                 I really liked these earrings. However, i agree with one of␣
 ↪the earlier reviews that I thought they would have been a little bigger.\n
24                 I really liked these earrings. However, i agree with one of␣
 ↪the earlier reviews that I thought they would have been a little bigger.\n
25             Eve's Addiction was wonderful with sending the ring and the ring␣
 ↪is beautiful; my daughter waas thrilled with the ring. Thank you, Dorothy
26                                           I got the ring as a promise ring␣
 ↪for my girlfriend for Christmas and she loved it. Definitely a great value.
27         This dainty heart looks absolutely beautiful on. It picks up the␣
 ↪colors of your clothing. It is an amazing price for such a beautiful pendant.
28                               arrived before estimated date. many previous␣
 ↪orders from this seller which always are on time and in excellent condition
29  My fiance and I looked at many different rings and I fell in love with this␣
 ↪ring, it was everything I wanted in an engagement ring! Now I have bee…
30                                                                        my␣
 ↪wife loves the ring, it was a great gift. extremelly cheap and high quality.
31                   Wonderful shopping experience I purchased the item for␣
 ↪holiday presents and the whole order came quickly and in wonderful condition.
32                                  Very disappointed in the appearance and␣
 ↪quality of the bracelet and its definitely not worth $45.00 - not even close.
33                                Ring is way too small and looks like a toy␣
 ↪when putting it on. I would not recommend if you want a nice 1/2 carat ring.
34                                                                              ␣
 ↪        am very pleased with my purchase, speedy shipping will use again
35                                             I just got these yesterday as a␣
 ↪Christmas gift- so far they look just like the picture and seem very nice.
36                                                         I got this ring for␣
 ↪my birthday and I love it, I cannot imagine a woman not adoring this ring.
```

```
37                                                         I got this ring for␣
 ↪my birthday and I love it, I cannot imagine a woman not adoring this ring.
38                           Item arrived extremely damaged in several places.␣
 ↪Not packaged well had to send it back. Very disappointed with the quality.
39                                                                              ␣
 ↪                               very good for everyday wear or dressing up
40                                                                              ␣
 ↪    I have always wanted a claddaugh ring.This price was great.I love it
41                          Looks exactly like the picture. Very nice quality. A␣
 ↪must for everyone who is a Tiger fan and owns an Italian Charm Bracelet.
42                                The message is very positive and it looks very␣
 ↪pretty. I bought it for my aunt as a present and the color is very nice.
43                                                 Recv'd my ring in a timely␣
 ↪manner it looks very antique would recommend this ring to any garnet lover!
44  I ordered a ring that stated Toe Ring" in the description. The ring came␣
 ↪very quickly but was not a toe ring. I don't know anyone with a size 8 to…
45  This product was hollow, which was not clearly specified in the item␣
 ↪description. It was not what I was expecting and seemed to be poor quality. I…
46                                           the product i recieved was nice it␣
 ↪came in a timley matter faster than i expected will order this item again
47  Nice ring for the an inexpensive ring, but one stone was missing. It was not␣
 ↪worth returning it as you would pay more for postage than what the ri…
48  This is a solid.beautiful ring. But if you are expecting the color in rhe␣
 ↪picture you will be disappointed. It is barely pink at all. When I first…
49                                                                          ␣
 ↪This works perfectly for my rings and my wider rings - both narrow and wide.
```

5.3.1 Now you have the top 50 results for selected query Lets build a topic model

[30]:

```python
# accept a doc and return all the cleaned list of words
def clean_doc(doc):
        # split into document text by white space
        words = doc.split()
        # remove punctuation from each token
        table = str.maketrans('', '', string.punctuation)
        words = [w.translate(table) for w in words]
        # remove remaining words that are not alphabetic
        words = [word for word in words if word.isalpha()]
        # filter out stop words
        stop_words = set(stopwords.words('english'))
        words = [w for w in words if not w in stop_words]
        # filter out short tokens
        words = [word for word in words if len(word) > 2]
        return words
```

```
# clean_doc('i received my ring and was a little disappointed that the ring is␣
 ↪not completely blue (like the picture shows). It looks like I got a blue␣
 ↪flower with green leaves. So it makes the ring look blue and green. Very␣
 ↪small ring. Not worth $6.99 but more like $3.')

# sentence of sentences to list of list of words
data_words = []
for s in interaction.data:
  s = clean_doc(s)
  cleaned_doc = [w for w in s if w in vocab]
  data_words.append(cleaned_doc)

#Call functions for cleaning lemmatization of text
# Initialize spacy 'en' model, keeping only tagger component (for efficiency)
# python3 -m spacy download en
nlp = spacy.load('en_core_web_sm', disable=['parser', 'ner'])

# Do lemmatization keeping only noun, adj, vb, adv
data_lemmatized = lemmatization(data_words, allowed_postags=['NOUN', 'ADJ',␣
 ↪'VERB', 'ADV'])

print("Cleaned and lemmarized tokens of first docuemnt: ",data_lemmatized[:
 ↪1][0])

# Create Dictionary
id2word = corpora.Dictionary(data_lemmatized)

# Term Document Frequency
corpus = [id2word.doc2bow(text) for text in data_lemmatized]

c=[]
for (i,j) in corpus[:1][0]:
  c.append((id2word.get(i),j))
print("Bag-of-words (BoW) format of first document: ",c)

# Gensim LDA Model
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                           id2word=id2word,
                                           num_topics=3,
                                           random_state=100,
                                           update_every=3,
                                           chunksize=4,
                                           passes=2,
                                           alpha='auto',
                                           per_word_topics=True)


# Compute Perplexity
```

```
print('\nPerplexity: ', lda_model.log_perplexity(corpus))  # a measure of how␣
 ↪good the model is. lower the better.

# Compute Coherence Score
coherence_model_lda = CoherenceModel(model=lda_model, texts=data_lemmatized,␣
 ↪dictionary=id2word, coherence='c_v')
coherence_lda = coherence_model_lda.get_coherence()
print('\nCoherence Score: ', coherence_lda)

print()
print()

# Visualize the topics
vis_data = gensimvis.prepare(lda_model, corpus, id2word, sort_topics=False)
pyLDAvis.display(vis_data)
```

```
Cleaned and lemmarized tokens of first docuemnt:  ['nice', 'look', 'picture']
Bag-of-words (BoW) format of first document:  [('look', 1), ('nice', 1),
('picture', 1)]
```

```
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
```

```
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
```

```
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
```

```
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
```

```
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
```

```
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
```

```
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
```

```
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
```

```
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)
/usr/local/lib/python3.9/dist-packages/gensim/models/ldamodel.py:1077:
DeprecationWarning: Calling np.sum(generator) is deprecated, and in the future
will give a different result. Use np.sum(np.fromiter(generator)) or the python
sum builtin instead.
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt
in doc)


Perplexity:  -5.716496944688342

Coherence Score:  0.5608393950892389



/usr/local/lib/python3.9/dist-packages/pyLDAvis/_prepare.py:243: FutureWarning:
In a future version of pandas all arguments of DataFrame.drop except for the
argument 'labels' will be keyword-only.
  default_term_info = default_term_info.sort_values(
```

[30]:

```
<IPython.core.display.HTML object>
```

6 Part 5 Text Summarisation

In This Section I will Use Summertime Library to Summarize the top 10 results that are retrived using the Neural Network Model in Task 3

6.1 Part 5 a & Part 5b: Downloading Summertime

[31]:

```
# Download SummerTime
# Swith to the Summertime directory

!git clone https://github.com/Yale-LILY/SummerTime.git
```

```
Cloning into 'SummerTime'…
remote: Enumerating objects: 4385, done.
remote: Counting objects: 100% (690/690), done.
remote: Compressing objects: 100% (192/192), done.
remote: Total 4385 (delta 598), reused 498 (delta 498), pack-reused 3695
Receiving objects: 100% (4385/4385), 9.84 MiB | 9.83 MiB/s, done.
Resolving deltas: 100% (2407/2407), done.
```

[32]:

```
import fileinput
import sys
s1 = 'transformers~=4.5.1'
s1_new = 'transformers'
s2 = 'tensorboard==2.4.1'
```

```
s2_new='tensorboard'

def replaceAll(file,searchExp,replaceExp):
    for line in fileinput.input(file, inplace=1):
        if searchExp in line:
            line = line.replace(searchExp,replaceExp)
        sys.stdout.write(line)

replaceAll('/content/SummerTime/requirements.txt',s1,s1_new)
replaceAll('/content/SummerTime/requirements.txt',s2,s2_new)

%cd SummerTime/

# Pip install Summertime locally

!pip install -e .
```

```
/content/SummerTime
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Obtaining file:///content/SummerTime
  Installing build dependencies … done
  Checking if build backend supports build_editable … done
  Getting requirements to build wheel … done
  Preparing metadata (pyproject.toml) … done
Collecting easynmt~=2.0.1
  Downloading EasyNMT-2.0.2.tar.gz (23 kB)
  Preparing metadata (setup.py) … done
Collecting transformers~=4.5.1
  Downloading transformers-4.5.1-py3-none-any.whl (2.1 MB)
                           2.1/2.1 MB
26.0 MB/s eta 0:00:00
Collecting nltk==3.6.2
  Downloading nltk-3.6.2-py3-none-any.whl (1.5 MB)
                           1.5/1.5 MB
32.7 MB/s eta 0:00:00
Collecting gensim~=3.8.3
  Downloading gensim-3.8.3.tar.gz (23.4 MB)
                           23.4/23.4 MB
20.4 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Collecting black~=21.12b0
  Downloading black-21.12b0-py3-none-any.whl (156 kB)
                          156.7/156.7 KB
12.6 MB/s eta 0:00:00
Collecting tensorboard~=2.4.1
  Downloading tensorboard-2.4.1-py3-none-any.whl (10.6 MB)
```

```
                           10.6/10.6 MB
47.4 MB/s eta 0:00:00
Collecting sentencepiece~=0.1.95
  Downloading
sentencepiece-0.1.97-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(1.3 MB)
                           1.3/1.3 MB
51.1 MB/s eta 0:00:00
Collecting click==7.1.2
  Downloading click-7.1.2-py2.py3-none-any.whl (82 kB)
                           82.8/82.8 KB
8.2 MB/s eta 0:00:00
Collecting readability-lxml
  Downloading readability_lxml-0.8.1-py3-none-any.whl (20 kB)
Requirement already satisfied: numpy in /usr/local/lib/python3.9/dist-packages
(from summertime==1.2.1) (1.22.4)
Requirement already satisfied: prettytable in /usr/local/lib/python3.9/dist-
packages (from summertime==1.2.1) (3.6.0)
Collecting summ-eval==0.70
  Downloading summ_eval-0.70-py3-none-any.whl (62.5 MB)
                           62.5/62.5 MB
11.6 MB/s eta 0:00:00
Collecting lexrank~=0.1.0
  Downloading lexrank-0.1.0-py3-none-any.whl (69 kB)
                           69.8/69.8 KB
6.8 MB/s eta 0:00:00
Collecting sklearn
  Downloading sklearn-0.0.post1.tar.gz (3.6 kB)
  Preparing metadata (setup.py) … done
Collecting tqdm~=4.49.0
  Downloading tqdm-4.49.0-py2.py3-none-any.whl (69 kB)
                           69.8/69.8 KB
3.9 MB/s eta 0:00:00
Collecting orjson
  Downloading orjson-3.8.7-cp39-cp39-manylinux_2_28_x86_64.whl (140 kB)
                          140.9/140.9 KB
15.7 MB/s eta 0:00:00
Collecting py7zr~=0.16.1
  Downloading py7zr-0.16.4-py3-none-any.whl (67 kB)
                           67.7/67.7 KB
7.8 MB/s eta 0:00:00
Collecting flake8
  Downloading flake8-6.0.0-py2.py3-none-any.whl (57 kB)
                           57.8/57.8 KB
6.9 MB/s eta 0:00:00
Collecting gdown~=4.2.0
  Downloading gdown-4.2.2.tar.gz (13 kB)
  Installing build dependencies … done
  Getting requirements to build wheel … done
  Preparing metadata (pyproject.toml) … done
Collecting progressbar
  Downloading progressbar-2.5.tar.gz (10 kB)
  Preparing metadata (setup.py) … done
Collecting spacy==3.0.6
  Downloading spacy-3.0.6-cp39-cp39-manylinux2014_x86_64.whl (12.6 MB)
                           12.6/12.6 MB
74.2 MB/s eta 0:00:00
Collecting datasets~=1.6.2
  Downloading datasets-1.6.2-py3-none-any.whl (221 kB)
                          221.8/221.8 KB
19.7 MB/s eta 0:00:00
Requirement already satisfied: cython in /usr/local/lib/python3.9/dist-
packages (from summertime==1.2.1) (0.29.33)
Collecting jupyter
  Downloading jupyter-1.0.0-py2.py3-none-any.whl (2.7 kB)
Collecting pytextrank
  Downloading pytextrank-3.2.4-py3-none-any.whl (30 kB)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.9/dist-
packages (from summertime==1.2.1) (4.9.3)
Requirement already satisfied: torch~=1.8 in /usr/local/lib/python3.9/dist-
packages (from summertime==1.2.1) (1.13.1+cu116)
Collecting fasttext~=0.9.2
  Downloading fasttext-0.9.2.tar.gz (68 kB)
                           68.8/68.8 KB
7.1 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Requirement already satisfied: regex in /usr/local/lib/python3.9/dist-packages
(from nltk==3.6.2->summertime==1.2.1) (2022.6.2)
Requirement already satisfied: joblib in /usr/local/lib/python3.9/dist-packages
(from nltk==3.6.2->summertime==1.2.1) (1.2.0)
Requirement already satisfied: catalogue<2.1.0,>=2.0.3 in
/usr/local/lib/python3.9/dist-packages (from spacy==3.0.6->summertime==1.2.1)
(2.0.8)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.9/dist-
packages (from spacy==3.0.6->summertime==1.2.1) (23.0)
Requirement already satisfied: wasabi<1.1.0,>=0.8.1 in
/usr/local/lib/python3.9/dist-packages (from spacy==3.0.6->summertime==1.2.1)
(0.10.1)
Requirement already satisfied: pathy>=0.3.5 in /usr/local/lib/python3.9/dist-
packages (from spacy==3.0.6->summertime==1.2.1) (0.10.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.9/dist-packages
(from spacy==3.0.6->summertime==1.2.1) (3.1.2)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in
/usr/local/lib/python3.9/dist-packages (from spacy==3.0.6->summertime==1.2.1)
(3.0.8)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in
```

```
/usr/local/lib/python3.9/dist-packages (from spacy==3.0.6->summertime==1.2.1)
(2.25.1)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in
/usr/local/lib/python3.9/dist-packages (from spacy==3.0.6->summertime==1.2.1)
(2.0.7)
Collecting pydantic<1.8.0,>=1.7.1
  Downloading pydantic-1.7.4-cp39-cp39-manylinux2014_x86_64.whl (10.3 MB)
                           10.3/10.3 MB
50.5 MB/s eta 0:00:00
Collecting thinc<8.1.0,>=8.0.3
  Downloading
thinc-8.0.17-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (668 kB)
                          668.8/668.8 KB
39.9 MB/s eta 0:00:00
Requirement already satisfied: setuptools in
/usr/local/lib/python3.9/dist-packages (from spacy==3.0.6->summertime==1.2.1)
(63.4.3)
Collecting typer<0.4.0,>=0.3.0
  Downloading typer-0.3.2-py3-none-any.whl (21 kB)
Requirement already satisfied: srsly<3.0.0,>=2.4.1 in
/usr/local/lib/python3.9/dist-packages (from spacy==3.0.6->summertime==1.2.1)
(2.4.6)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in
/usr/local/lib/python3.9/dist-packages (from spacy==3.0.6->summertime==1.2.1)
(1.0.9)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.4 in
/usr/local/lib/python3.9/dist-packages (from spacy==3.0.6->summertime==1.2.1)
(3.0.12)
Requirement already satisfied: blis<0.8.0,>=0.4.0 in
/usr/local/lib/python3.9/dist-packages (from spacy==3.0.6->summertime==1.2.1)
(0.7.9)
Collecting pytorch-pretrained-bert
  Downloading pytorch_pretrained_bert-0.6.2-py3-none-any.whl (123 kB)
                          123.8/123.8 KB
12.8 MB/s eta 0:00:00
Collecting bert-score
  Downloading bert_score-0.3.13-py3-none-any.whl (61 kB)
                           61.1/61.1 KB
6.9 MB/s eta 0:00:00
Collecting sacremoses
  Downloading sacremoses-0.0.53.tar.gz (880 kB)
                          880.6/880.6 KB
36.9 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Requirement already satisfied: networkx in /usr/local/lib/python3.9/dist-
packages (from summ-eval==0.70->summertime==1.2.1) (3.0)
Requirement already satisfied: psutil in /usr/local/lib/python3.9/dist-packages
(from summ-eval==0.70->summertime==1.2.1) (5.4.8)
```

```
Collecting wmd
  Downloading wmd-1.3.2.tar.gz (104 kB)
                          104.6/104.6 KB
10.5 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Collecting stanza
  Downloading stanza-1.5.0-py3-none-any.whl (802 kB)
                          802.5/802.5 KB
48.1 MB/s eta 0:00:00
Collecting moverscore
  Downloading moverscore-1.0.3.tar.gz (7.7 kB)
  Preparing metadata (setup.py) … done
Requirement already satisfied: scipy in /usr/local/lib/python3.9/dist-packages
(from summ-eval==0.70->summertime==1.2.1) (1.10.1)
Collecting pyemd==0.5.1
  Downloading pyemd-0.5.1.tar.gz (91 kB)
                           91.5/91.5 KB
10.5 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Collecting blanc
  Downloading blanc-0.3.0-py3-none-any.whl (29 kB)
Requirement already satisfied: gin-config in /usr/local/lib/python3.9/dist-
packages (from summ-eval==0.70->summertime==1.2.1) (0.5.0)
Requirement already satisfied: six in /usr/local/lib/python3.9/dist-packages
(from summ-eval==0.70->summertime==1.2.1) (1.15.0)
Collecting sacrebleu
  Downloading sacrebleu-2.3.1-py3-none-any.whl (118 kB)
                          118.9/118.9 KB
10.8 MB/s eta 0:00:00
Requirement already satisfied: typing-extensions>=3.10.0.0 in
/usr/local/lib/python3.9/dist-packages (from black~=21.12b0->summertime==1.2.1)
(4.5.0)
Requirement already satisfied: platformdirs>=2 in /usr/local/lib/python3.9/dist-
packages (from black~=21.12b0->summertime==1.2.1) (3.1.1)
Collecting tomli<2.0.0,>=0.2.6
  Downloading tomli-1.2.3-py3-none-any.whl (12 kB)
Collecting mypy-extensions>=0.4.3
  Downloading mypy_extensions-1.0.0-py3-none-any.whl (4.7 kB)
Collecting pathspec<1,>=0.9.0
  Downloading pathspec-0.11.1-py3-none-any.whl (29 kB)
Collecting dill
  Downloading dill-0.3.6-py3-none-any.whl (110 kB)
                          110.5/110.5 KB
13.4 MB/s eta 0:00:00
Collecting xxhash
  Downloading
xxhash-3.2.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (212 kB)
                          212.2/212.2 KB
```

```
21.1 MB/s eta 0:00:00
Collecting huggingface-hub<0.1.0
  Downloading huggingface_hub-0.0.19-py3-none-any.whl (56 kB)
                           56.9/56.9 KB
5.9 MB/s eta 0:00:00
Collecting multiprocess
  Downloading multiprocess-0.70.14-py39-none-any.whl (132 kB)
                           132.9/132.9

KB 1.4 MB/s eta 0:00:00
Requirement already satisfied: fsspec in /usr/local/lib/python3.9/dist-
packages (from datasets~=1.6.2->summertime==1.2.1) (2023.3.0)
Requirement already satisfied: pyarrow>=1.0.0<4.0.0 in
/usr/local/lib/python3.9/dist-packages (from datasets~=1.6.2->summertime==1.2.1)
(9.0.0)
Requirement already satisfied: pandas in /usr/local/lib/python3.9/dist-packages
(from datasets~=1.6.2->summertime==1.2.1) (1.4.4)
Requirement already satisfied: protobuf in /usr/local/lib/python3.9/dist-
packages (from easynmt~=2.0.1->summertime==1.2.1) (3.19.6)
Collecting pybind11>=2.2
  Using cached pybind11-2.10.4-py3-none-any.whl (222 kB)
Requirement already satisfied: filelock in /usr/local/lib/python3.9/dist-
packages (from gdown~=4.2.0->summertime==1.2.1) (3.9.1)
Requirement already satisfied: smart_open>=1.8.1 in
/usr/local/lib/python3.9/dist-packages (from gensim~=3.8.3->summertime==1.2.1)
(6.3.0)
Requirement already satisfied: pyrsistent>=0.14.0 in
/usr/local/lib/python3.9/dist-packages (from lexrank~=0.1.0->summertime==1.2.1)
(0.19.3)
Collecting urlextract>=0.7
  Downloading urlextract-1.8.0-py3-none-any.whl (21 kB)
Collecting path.py>=10.5
  Downloading path.py-12.5.0-py3-none-any.whl (2.3 kB)
Collecting pybcj>=0.5.0
  Downloading
pybcj-1.0.1-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (49 kB)
                           49.6/49.6 KB
5.2 MB/s eta 0:00:00
Collecting pycryptodomex>=3.6.6
  Downloading
pycryptodomex-3.17-cp35-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.1
MB)
                           2.1/2.1 MB
64.3 MB/s eta 0:00:00
Collecting multivolumefile>=0.2.3
  Downloading multivolumefile-0.2.3-py3-none-any.whl (17 kB)
Collecting pyzstd>=0.14.4
  Downloading
```

```
pyzstd-0.15.4-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (384 kB)
                          384.0/384.0 KB
32.4 MB/s eta 0:00:00
Collecting texttable
  Downloading texttable-1.6.7-py2.py3-none-any.whl (10 kB)
Collecting pyppmd>=0.17.0
  Downloading
pyppmd-1.0.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (138 kB)
                          138.7/138.7 KB
15.6 MB/s eta 0:00:00
Collecting brotli>=1.0.9
  Downloading Brotli-1.0.9-cp39-cp39-manylinux1_x86_64.whl (357 kB)
                          357.2/357.2 KB
33.6 MB/s eta 0:00:00
Requirement already satisfied: markdown>=2.6.8 in
/usr/local/lib/python3.9/dist-packages (from
tensorboard~=2.4.1->summertime==1.2.1) (3.4.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
/usr/local/lib/python3.9/dist-packages (from
tensorboard~=2.4.1->summertime==1.2.1) (1.8.1)
Requirement already satisfied: werkzeug>=0.11.15 in
/usr/local/lib/python3.9/dist-packages (from
tensorboard~=2.4.1->summertime==1.2.1) (2.2.3)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
/usr/local/lib/python3.9/dist-packages (from
tensorboard~=2.4.1->summertime==1.2.1) (0.4.6)
Requirement already satisfied: grpcio>=1.24.3 in /usr/local/lib/python3.9/dist-
packages (from tensorboard~=2.4.1->summertime==1.2.1) (1.51.3)
Collecting google-auth<2,>=1.6.3
  Downloading google_auth-1.35.0-py2.py3-none-any.whl (152 kB)
                          152.9/152.9 KB
16.4 MB/s eta 0:00:00
Requirement already satisfied: absl-py>=0.4 in
/usr/local/lib/python3.9/dist-packages (from
tensorboard~=2.4.1->summertime==1.2.1) (1.4.0)
Requirement already satisfied: wheel>=0.26 in /usr/local/lib/python3.9/dist-
packages (from tensorboard~=2.4.1->summertime==1.2.1) (0.40.0)
Collecting tokenizers<0.11,>=0.10.1
  Downloading tokenizers-0.10.3-cp39-cp39-manylinux_2_5_x86_64.manylinux1_x86_64
.manylinux_2_12_x86_64.manylinux2010_x86_64.whl (3.3 MB)
                           3.3/3.3 MB
51.3 MB/s eta 0:00:00
Requirement already satisfied: soupsieve>1.2 in
/usr/local/lib/python3.9/dist-packages (from beautifulsoup4->summertime==1.2.1)
(2.4)
Collecting pycodestyle<2.11.0,>=2.10.0
  Downloading pycodestyle-2.10.0-py2.py3-none-any.whl (41 kB)
                           41.3/41.3 KB
```

```
3.8 MB/s eta 0:00:00
Collecting mccabe<0.8.0,>=0.7.0
  Downloading mccabe-0.7.0-py2.py3-none-any.whl (7.3 kB)
Collecting pyflakes<3.1.0,>=3.0.0
  Downloading pyflakes-3.0.1-py2.py3-none-any.whl (62 kB)
                           62.8/62.8 KB
6.6 MB/s eta 0:00:00
Requirement already satisfied: ipywidgets in
/usr/local/lib/python3.9/dist-packages (from jupyter->summertime==1.2.1) (7.7.1)
Requirement already satisfied: jupyter-console in /usr/local/lib/python3.9/dist-
packages (from jupyter->summertime==1.2.1) (6.1.0)
Requirement already satisfied: notebook in /usr/local/lib/python3.9/dist-
packages (from jupyter->summertime==1.2.1) (6.3.0)
Collecting qtconsole
  Downloading qtconsole-5.4.1-py3-none-any.whl (120 kB)
                          120.9/120.9 KB
11.5 MB/s eta 0:00:00
Requirement already satisfied: nbconvert in /usr/local/lib/python3.9/dist-
packages (from jupyter->summertime==1.2.1) (6.5.4)
Requirement already satisfied: ipykernel in /usr/local/lib/python3.9/dist-
packages (from jupyter->summertime==1.2.1) (5.3.4)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.9/dist-packages
(from prettytable->summertime==1.2.1) (0.2.6)
Collecting pygments>=2.7.4
  Downloading Pygments-2.14.0-py3-none-any.whl (1.1 MB)
                           1.1/1.1 MB
49.9 MB/s eta 0:00:00
Collecting graphviz>=0.13
  Downloading graphviz-0.20.1-py3-none-any.whl (47 kB)
                           47.0/47.0 KB
4.6 MB/s eta 0:00:00
Collecting icecream>=2.1
  Downloading icecream-2.1.3-py2.py3-none-any.whl (8.4 kB)
Requirement already satisfied: chardet in /usr/local/lib/python3.9/dist-packages
(from readability-lxml->summertime==1.2.1) (4.0.0)
Requirement already satisfied: lxml in /usr/local/lib/python3.9/dist-packages
(from readability-lxml->summertime==1.2.1) (4.9.2)
Collecting cssselect
  Downloading cssselect-1.2.0-py2.py3-none-any.whl (18 kB)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.9/dist-
packages (from google-auth<2,>=1.6.3->tensorboard~=2.4.1->summertime==1.2.1)
(4.9)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/usr/local/lib/python3.9/dist-packages (from google-
auth<2,>=1.6.3->tensorboard~=2.4.1->summertime==1.2.1) (0.2.8)
Collecting cachetools<5.0,>=2.0.0
  Downloading cachetools-4.2.4-py3-none-any.whl (10 kB)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
```

```
/usr/local/lib/python3.9/dist-packages (from google-auth-
oauthlib<0.5,>=0.4.1->tensorboard~=2.4.1->summertime==1.2.1) (1.3.1)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.9/dist-packages
(from huggingface-hub<0.1.0->datasets~=1.6.2->summertime==1.2.1) (6.0)
Collecting asttokens>=2.0.1
  Downloading asttokens-2.2.1-py2.py3-none-any.whl (26 kB)
Collecting colorama>=0.3.9
  Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Collecting executing>=0.3.1
  Downloading executing-1.2.0-py2.py3-none-any.whl (24 kB)
Requirement already satisfied: importlib-metadata>=4.4 in
/usr/local/lib/python3.9/dist-packages (from
markdown>=2.6.8->tensorboard~=2.4.1->summertime==1.2.1) (6.0.0)
Requirement already satisfied: matplotlib>=3.4 in /usr/local/lib/python3.9/dist-
packages (from networkx->summ-eval==0.70->summertime==1.2.1) (3.7.1)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.9/dist-
packages (from pandas->datasets~=1.6.2->summertime==1.2.1) (2022.7.1)
Requirement already satisfied: python-dateutil>=2.8.1 in
/usr/local/lib/python3.9/dist-packages (from
pandas->datasets~=1.6.2->summertime==1.2.1) (2.8.2)
Collecting path
  Downloading path-16.6.0-py3-none-any.whl (26 kB)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/usr/local/lib/python3.9/dist-packages (from
requests<3.0.0,>=2.13.0->spacy==3.0.6->summertime==1.2.1) (1.26.15)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.9/dist-
packages (from requests<3.0.0,>=2.13.0->spacy==3.0.6->summertime==1.2.1) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.9/dist-packages (from
requests<3.0.0,>=2.13.0->spacy==3.0.6->summertime==1.2.1) (2022.12.7)
Collecting uritools
  Downloading uritools-4.0.1-py3-none-any.whl (10 kB)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/usr/local/lib/python3.9/dist-packages (from
werkzeug>=0.11.15->tensorboard~=2.4.1->summertime==1.2.1) (2.1.2)
Requirement already satisfied: ipython>=5.0.0 in /usr/local/lib/python3.9/dist-
packages (from ipykernel->jupyter->summertime==1.2.1) (7.9.0)
Requirement already satisfied: jupyter-client in /usr/local/lib/python3.9/dist-
packages (from ipykernel->jupyter->summertime==1.2.1) (6.1.12)
Requirement already satisfied: traitlets>=4.1.0 in
/usr/local/lib/python3.9/dist-packages (from
ipykernel->jupyter->summertime==1.2.1) (5.7.1)
Requirement already satisfied: tornado>=4.2 in /usr/local/lib/python3.9/dist-
packages (from ipykernel->jupyter->summertime==1.2.1) (6.2)
Requirement already satisfied: jupyterlab-widgets>=1.0.0 in
/usr/local/lib/python3.9/dist-packages (from
ipywidgets->jupyter->summertime==1.2.1) (3.0.5)
Requirement already satisfied: ipython-genutils~=0.2.0 in
```

```
/usr/local/lib/python3.9/dist-packages (from
ipywidgets->jupyter->summertime==1.2.1) (0.2.0)
Requirement already satisfied: widgetsnbextension~=3.6.0 in
/usr/local/lib/python3.9/dist-packages (from
ipywidgets->jupyter->summertime==1.2.1) (3.6.2)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in
/usr/local/lib/python3.9/dist-packages (from jupyter-
console->jupyter->summertime==1.2.1) (2.0.10)
Collecting typing
  Downloading typing-3.7.4.3.tar.gz (78 kB)
                           78.6/78.6 KB
7.9 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Collecting portalocker
  Downloading portalocker-2.7.0-py2.py3-none-any.whl (15 kB)
Requirement already satisfied: mistune<2,>=0.8.1 in
/usr/local/lib/python3.9/dist-packages (from
nbconvert->jupyter->summertime==1.2.1) (0.8.4)
Requirement already satisfied: pandocfilters>=1.4.1 in
/usr/local/lib/python3.9/dist-packages (from
nbconvert->jupyter->summertime==1.2.1) (1.5.0)
Requirement already satisfied: nbclient>=0.5.0 in /usr/local/lib/python3.9/dist-
packages (from nbconvert->jupyter->summertime==1.2.1) (0.7.2)
Requirement already satisfied: entrypoints>=0.2.2 in
/usr/local/lib/python3.9/dist-packages (from
nbconvert->jupyter->summertime==1.2.1) (0.4)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.9/dist-
packages (from nbconvert->jupyter->summertime==1.2.1) (0.7.1)
Requirement already satisfied: jupyter-core>=4.7 in
/usr/local/lib/python3.9/dist-packages (from
nbconvert->jupyter->summertime==1.2.1) (5.2.0)
Requirement already satisfied: nbformat>=5.1 in /usr/local/lib/python3.9/dist-
packages (from nbconvert->jupyter->summertime==1.2.1) (5.7.3)
Requirement already satisfied: jupyterlab-pygments in
/usr/local/lib/python3.9/dist-packages (from
nbconvert->jupyter->summertime==1.2.1) (0.2.2)
Requirement already satisfied: bleach in /usr/local/lib/python3.9/dist-packages
(from nbconvert->jupyter->summertime==1.2.1) (6.0.0)
Requirement already satisfied: tinycss2 in /usr/local/lib/python3.9/dist-
packages (from nbconvert->jupyter->summertime==1.2.1) (1.2.1)
Requirement already satisfied: argon2-cffi in /usr/local/lib/python3.9/dist-
packages (from notebook->jupyter->summertime==1.2.1) (21.3.0)
Requirement already satisfied: prometheus-client in
/usr/local/lib/python3.9/dist-packages (from
notebook->jupyter->summertime==1.2.1) (0.16.0)
Requirement already satisfied: pyzmq>=17 in /usr/local/lib/python3.9/dist-
packages (from notebook->jupyter->summertime==1.2.1) (23.2.1)
Requirement already satisfied: Send2Trash>=1.5.0 in
```

```
/usr/local/lib/python3.9/dist-packages (from
notebook->jupyter->summertime==1.2.1) (1.8.0)
Requirement already satisfied: terminado>=0.8.3 in
/usr/local/lib/python3.9/dist-packages (from
notebook->jupyter->summertime==1.2.1) (0.17.1)
Collecting boto3
  Downloading boto3-1.26.93-py3-none-any.whl (135 kB)
                          135.1/135.1 KB
13.5 MB/s eta 0:00:00
Collecting qtpy>=2.0.1
  Downloading QtPy-2.3.0-py3-none-any.whl (83 kB)
                           83.6/83.6 KB
10.6 MB/s eta 0:00:00
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in
/usr/local/lib/python3.9/dist-packages (from
requests<3.0.0,>=2.13.0->spacy==3.0.6->summertime==1.2.1) (1.7.1)
Requirement already satisfied: tabulate>=0.8.9 in /usr/local/lib/python3.9/dist-
packages (from sacrebleu->summ-eval==0.70->summertime==1.2.1) (0.8.10)
Collecting emoji
  Downloading emoji-2.2.0.tar.gz (240 kB)
                          240.9/240.9 KB
20.9 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.9/dist-
packages (from importlib-
metadata>=4.4->markdown>=2.6.8->tensorboard~=2.4.1->summertime==1.2.1) (3.15.0)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.9/dist-
packages (from ipython>=5.0.0->ipykernel->jupyter->summertime==1.2.1) (0.7.5)
Requirement already satisfied: pexpect in /usr/local/lib/python3.9/dist-packages
(from ipython>=5.0.0->ipykernel->jupyter->summertime==1.2.1) (4.8.0)
Requirement already satisfied: decorator in /usr/local/lib/python3.9/dist-
packages (from ipython>=5.0.0->ipykernel->jupyter->summertime==1.2.1) (4.4.2)
Requirement already satisfied: backcall in /usr/local/lib/python3.9/dist-
packages (from ipython>=5.0.0->ipykernel->jupyter->summertime==1.2.1) (0.2.0)
Collecting jedi>=0.10
  Downloading jedi-0.18.2-py2.py3-none-any.whl (1.6 MB)
                           1.6/1.6 MB
55.9 MB/s eta 0:00:00
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.9/dist-packages (from matplotlib>=3.4->networkx->summ-
eval==0.70->summertime==1.2.1) (1.0.7)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.9/dist-packages (from matplotlib>=3.4->networkx->summ-
eval==0.70->summertime==1.2.1) (1.4.4)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.9/dist-
packages (from matplotlib>=3.4->networkx->summ-eval==0.70->summertime==1.2.1)
(0.11.0)
Requirement already satisfied: importlib-resources>=3.2.0 in
```

```
/usr/local/lib/python3.9/dist-packages (from matplotlib>=3.4->networkx->summ-
eval==0.70->summertime==1.2.1) (5.12.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.9/dist-packages (from matplotlib>=3.4->networkx->summ-
eval==0.70->summertime==1.2.1) (3.0.9)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.9/dist-
packages (from matplotlib>=3.4->networkx->summ-eval==0.70->summertime==1.2.1)
(8.4.0)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.9/dist-packages (from matplotlib>=3.4->networkx->summ-
eval==0.70->summertime==1.2.1) (4.39.0)
Requirement already satisfied: fastjsonschema in /usr/local/lib/python3.9/dist-
packages (from nbformat>=5.1->nbconvert->jupyter->summertime==1.2.1) (2.16.3)
Requirement already satisfied: jsonschema>=2.6 in /usr/local/lib/python3.9/dist-
packages (from nbformat>=5.1->nbconvert->jupyter->summertime==1.2.1) (4.3.3)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in
/usr/local/lib/python3.9/dist-packages (from pyasn1-modules>=0.2.1->google-
auth<2,>=1.6.3->tensorboard~=2.4.1->summertime==1.2.1) (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.9/dist-
packages (from requests-oauthlib>=0.7.0->google-auth-
oauthlib<0.5,>=0.4.1->tensorboard~=2.4.1->summertime==1.2.1) (3.2.2)
Requirement already satisfied: ptyprocess in /usr/local/lib/python3.9/dist-
packages (from terminado>=0.8.3->notebook->jupyter->summertime==1.2.1) (0.7.0)
Requirement already satisfied: argon2-cffi-bindings in
/usr/local/lib/python3.9/dist-packages (from
argon2-cffi->notebook->jupyter->summertime==1.2.1) (21.2.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.9/dist-
packages (from bleach->nbconvert->jupyter->summertime==1.2.1) (0.5.1)
Collecting jmespath<2.0.0,>=0.7.1
  Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Collecting s3transfer<0.7.0,>=0.6.0
  Downloading s3transfer-0.6.0-py3-none-any.whl (79 kB)
                           79.6/79.6 KB
9.1 MB/s eta 0:00:00
Collecting botocore<1.30.0,>=1.29.93
  Downloading botocore-1.29.93-py3-none-any.whl (10.5 MB)
                           10.5/10.5 MB
88.9 MB/s eta 0:00:00
Requirement already satisfied: parso<0.9.0,>=0.8.0 in
/usr/local/lib/python3.9/dist-packages (from
jedi>=0.10->ipython>=5.0.0->ipykernel->jupyter->summertime==1.2.1) (0.8.3)
Requirement already satisfied: attrs>=17.4.0 in /usr/local/lib/python3.9/dist-
packages (from
jsonschema>=2.6->nbformat>=5.1->nbconvert->jupyter->summertime==1.2.1) (22.2.0)
Requirement already satisfied: cffi>=1.0.1 in /usr/local/lib/python3.9/dist-
packages (from argon2-cffi-
bindings->argon2-cffi->notebook->jupyter->summertime==1.2.1) (1.15.1)
Requirement already satisfied: pycparser in /usr/local/lib/python3.9/dist-
```

```
packages (from cffi>=1.0.1->argon2-cffi-
bindings->argon2-cffi->notebook->jupyter->summertime==1.2.1) (2.21)
Building wheels for collected packages: pyemd, easynmt, fasttext, gdown, gensim,
progressbar, sklearn, moverscore, sacremoses, wmd, emoji, typing
  Building wheel for pyemd (setup.py) … done
  Created wheel for pyemd: filename=pyemd-0.5.1-cp39-cp39-linux_x86_64.whl
size=540981
sha256=4bb945ba8507f171a263b0a038009895e0908c9af83a1e26a45dfbb4cbbe04c0
  Stored in directory: /root/.cache/pip/wheels/64/bf/3e/0859be9a0108fc932a29b943
792dcafb3b979555cf1bb5add6
  Building wheel for easynmt (setup.py) … done
  Created wheel for easynmt: filename=EasyNMT-2.0.2-py3-none-any.whl size=19920
sha256=89d20316bf4c27236a5835b4afb354134f957e2f0e60e18418232d6f29effdf5
  Stored in directory: /root/.cache/pip/wheels/26/53/00/5761f3b9bf6af87bdbc44029
2a4eb98a6afb25823dd76fca26
  Building wheel for fasttext (setup.py) … done
  Created wheel for fasttext: filename=fasttext-0.9.2-cp39-cp39-linux_x86_64.whl
size=4395627
sha256=da971b390bab968de56224dbe041d3cf5c22d93461d0dd419ac979446b739ae0
  Stored in directory: /root/.cache/pip/wheels/64/57/bc/1741406019061d5664914b07
0bd3e71f6244648732bc96109e
  Building wheel for gdown (pyproject.toml) … done
  Created wheel for gdown: filename=gdown-4.2.2-py3-none-any.whl size=14495
sha256=1bb75f418ec0cd854db46ac6722153a8104b09db86d82fa1b997565ecaa4911c
  Stored in directory: /root/.cache/pip/wheels/d3/d1/f3/112c8482aa998cd2fbf9d0c8
fd3a15b06a5581ca43152878c9
  Building wheel for gensim (setup.py) … done
  Created wheel for gensim: filename=gensim-3.8.3-cp39-cp39-linux_x86_64.whl
size=26527999
sha256=c87b521f33a4a54e37bc8922261ddbfdac78974aa462ff575e8dbc1fe91ef483
  Stored in directory: /root/.cache/pip/wheels/ca/5d/af/618594ec2f28608c1d6ee7d2
b7e95a3e9b06551e3b80a491d6
  Building wheel for progressbar (setup.py) … done
  Created wheel for progressbar: filename=progressbar-2.5-py3-none-any.whl
size=12080
sha256=bcabbe9e5e47734a9c6c5b51f2d4d952fca6930b9fdfe404d9406756d20b696f
  Stored in directory: /root/.cache/pip/wheels/d7/d9/89/a3f31c76ff6d51dc3b157562
8f59afe59e4ceae3f2748cd7ad
  Building wheel for sklearn (setup.py) … done
  Created wheel for sklearn: filename=sklearn-0.0.post1-py3-none-any.whl
size=2955
sha256=f9361c61800cd29bd25d9cf2fef0c392e47d03f991974c4cc9f7f983acaf9c17
  Stored in directory: /root/.cache/pip/wheels/f8/e0/3d/9d0c2020c44a519b9f02ab4f
a6d2a4a996c98d79ab2f569fa1
  Building wheel for moverscore (setup.py) … done
  Created wheel for moverscore: filename=moverscore-1.0.3-py3-none-any.whl
size=7963
sha256=347f2a1578188811461180d792b98450562e52b5f453aafbc60206bec2c2f46b
```

```
  Stored in directory: /root/.cache/pip/wheels/ec/c2/18/826e61ab6e3989b946b3dea3
45711552870ce9096209c9378c
  Building wheel for sacremoses (setup.py) … done
  Created wheel for sacremoses: filename=sacremoses-0.0.53-py3-none-any.whl
size=895259
sha256=71e08964663219768229aec3716b4e12d50b0c20baf046ad38cd2000aaff25a1
  Stored in directory: /root/.cache/pip/wheels/12/1c/3d/46cf06718d63a32ff798a895
94b61e7f345ab6b36d909ce033
  Building wheel for wmd (setup.py) … done
  Created wheel for wmd: filename=wmd-1.3.2-cp39-cp39-linux_x86_64.whl
size=1236782
sha256=708555de7f7789c9bdfe0a71f7d63d325a2e89fad22642087526416092ea3623
  Stored in directory: /root/.cache/pip/wheels/f2/bb/7b/46bc1b99fbd5018b8cfeb75e
6ffaa9d64c0bcecc026a5514b6
  Building wheel for emoji (setup.py) … done
  Created wheel for emoji: filename=emoji-2.2.0-py3-none-any.whl size=234926
sha256=9570a388240310227b2e6127b80731c10d9a63b9eaa8471c833b7468509b4873
  Stored in directory: /root/.cache/pip/wheels/9a/b8/0f/f580817231cbf59f6ade9fd1
32ff60ada1de9f7dc85521f857
  Building wheel for typing (setup.py) … done
  Created wheel for typing: filename=typing-3.7.4.3-py3-none-any.whl size=26321
sha256=a0daf8a7d6ea913e3ac96d28284bf4fa56aa14e0b0d86323ed9ee72fe24ad9db
  Stored in directory: /root/.cache/pip/wheels/fa/17/1f/332799f975d1b2d7f9b3f33b
bccf65031e794717d24432caee
Successfully built pyemd easynmt fasttext gdown gensim progressbar sklearn
moverscore sacremoses wmd emoji typing
Installing collected packages: tokenizers, texttable, sklearn, sentencepiece,
progressbar, executing, brotli, xxhash, wmd, uritools, typing, tqdm, tomli,
qtpy, pyzstd, pyppmd, pygments, pyflakes, pyemd, pydantic, pycryptodomex,
pycodestyle, pybind11, pybcj, portalocker, pathspec, path, orjson, mypy-
extensions, multivolumefile, mccabe, jmespath, jedi, graphviz, emoji, dill,
cssselect, colorama, click, cachetools, asttokens, urlextract, typer, thinc,
stanza, sacremoses, sacrebleu, readability-lxml, py7zr, path.py, nltk,
multiprocess, moverscore, icecream, huggingface-hub, google-auth, gensim,
flake8, fasttext, botocore, black, transformers, s3transfer, lexrank, gdown,
datasets, tensorboard, spacy, qtconsole, easynmt, boto3, blanc, bert-score,
pytorch-pretrained-bert, pytextrank, summ-eval, jupyter, summertime
  Attempting uninstall: tokenizers
    Found existing installation: tokenizers 0.13.2
    Uninstalling tokenizers-0.13.2:
      Successfully uninstalled tokenizers-0.13.2
  Attempting uninstall: tqdm
    Found existing installation: tqdm 4.65.0
    Uninstalling tqdm-4.65.0:
      Successfully uninstalled tqdm-4.65.0
  Attempting uninstall: tomli
    Found existing installation: tomli 2.0.1
    Uninstalling tomli-2.0.1:
```

```
    Successfully uninstalled tomli-2.0.1
Attempting uninstall: pygments
  Found existing installation: Pygments 2.6.1
  Uninstalling Pygments-2.6.1:
    Successfully uninstalled Pygments-2.6.1
Attempting uninstall: pydantic
  Found existing installation: pydantic 1.10.6
  Uninstalling pydantic-1.10.6:
    Successfully uninstalled pydantic-1.10.6
Attempting uninstall: graphviz
  Found existing installation: graphviz 0.10.1
  Uninstalling graphviz-0.10.1:
    Successfully uninstalled graphviz-0.10.1
Attempting uninstall: click
  Found existing installation: click 8.1.3
  Uninstalling click-8.1.3:
    Successfully uninstalled click-8.1.3
Attempting uninstall: cachetools
  Found existing installation: cachetools 5.3.0
  Uninstalling cachetools-5.3.0:
    Successfully uninstalled cachetools-5.3.0
Attempting uninstall: typer
  Found existing installation: typer 0.7.0
  Uninstalling typer-0.7.0:
    Successfully uninstalled typer-0.7.0
Attempting uninstall: thinc
  Found existing installation: thinc 8.1.9
  Uninstalling thinc-8.1.9:
    Successfully uninstalled thinc-8.1.9
Attempting uninstall: nltk
  Found existing installation: nltk 3.7
  Uninstalling nltk-3.7:
    Successfully uninstalled nltk-3.7
Attempting uninstall: huggingface-hub
  Found existing installation: huggingface-hub 0.13.2
  Uninstalling huggingface-hub-0.13.2:
    Successfully uninstalled huggingface-hub-0.13.2
Attempting uninstall: google-auth
  Found existing installation: google-auth 2.16.2
  Uninstalling google-auth-2.16.2:
    Successfully uninstalled google-auth-2.16.2
Attempting uninstall: gensim
  Found existing installation: gensim 3.6.0
  Uninstalling gensim-3.6.0:
    Successfully uninstalled gensim-3.6.0
Attempting uninstall: transformers
  Found existing installation: transformers 4.27.1
  Uninstalling transformers-4.27.1:
      Successfully uninstalled transformers-4.27.1
  Attempting uninstall: gdown
    Found existing installation: gdown 4.4.0
    Uninstalling gdown-4.4.0:
      Successfully uninstalled gdown-4.4.0
  Attempting uninstall: tensorboard
    Found existing installation: tensorboard 2.11.2
    Uninstalling tensorboard-2.11.2:
      Successfully uninstalled tensorboard-2.11.2
  Attempting uninstall: spacy
    Found existing installation: spacy 3.4.4
    Uninstalling spacy-3.4.4:
      Successfully uninstalled spacy-3.4.4
  Running setup.py develop for summertime
ERROR: pip's dependency resolver does not currently take into account all
the packages that are installed. This behaviour is the source of the following
dependency conflicts.
tensorflow 2.11.0 requires tensorboard<2.12,>=2.11, but you have tensorboard
2.4.1 which is incompatible.
pandas-profiling 3.2.0 requires joblib~=1.1.0, but you have joblib 1.2.0 which
is incompatible.
pandas-profiling 3.2.0 requires pydantic>=1.8.1, but you have pydantic 1.7.4
which is incompatible.
google-api-core 2.11.0 requires google-auth<3.0dev,>=2.14.1, but you have
google-auth 1.35.0 which is incompatible.
flask 2.2.3 requires click>=8.0, but you have click 7.1.2 which is incompatible.
en-core-web-sm 3.4.1 requires spacy<3.5.0,>=3.4.0, but you have spacy 3.0.6
which is incompatible.
Successfully installed asttokens-2.2.1 bert-score-0.3.13 black-21.12b0
blanc-0.3.0 boto3-1.26.93 botocore-1.29.93 brotli-1.0.9 cachetools-4.2.4
click-7.1.2 colorama-0.4.6 cssselect-1.2.0 datasets-1.6.2 dill-0.3.6
easynmt-2.0.2 emoji-2.2.0 executing-1.2.0 fasttext-0.9.2 flake8-6.0.0
gdown-4.2.2 gensim-3.8.3 google-auth-1.35.0 graphviz-0.20.1 huggingface-
hub-0.0.19 icecream-2.1.3 jedi-0.18.2 jmespath-1.0.1 jupyter-1.0.0 lexrank-0.1.0
mccabe-0.7.0 moverscore-1.0.3 multiprocess-0.70.14 multivolumefile-0.2.3 mypy-
extensions-1.0.0 nltk-3.6.2 orjson-3.8.7 path-16.6.0 path.py-12.5.0
pathspec-0.11.1 portalocker-2.7.0 progressbar-2.5 py7zr-0.16.4 pybcj-1.0.1
pybind11-2.10.4 pycodestyle-2.10.0 pycryptodomex-3.17 pydantic-1.7.4 pyemd-0.5.1
pyflakes-3.0.1 pygments-2.14.0 pyppmd-1.0.0 pytextrank-3.2.4 pytorch-pretrained-
bert-0.6.2 pyzstd-0.15.4 qtconsole-5.4.1 qtpy-2.3.0 readability-lxml-0.8.1
```

```
s3transfer-0.6.0 sacrebleu-2.3.1 sacremoses-0.0.53 sentencepiece-0.1.97
sklearn-0.0.post1 spacy-3.0.6 stanza-1.5.0 summ-eval-0.70 summertime-1.2.1
tensorboard-2.4.1 texttable-1.6.7 thinc-8.0.17 tokenizers-0.10.3 tomli-1.2.3
tqdm-4.49.0 transformers-4.5.1 typer-0.3.2 typing-3.7.4.3 uritools-4.0.1
urlextract-1.8.0 wmd-1.3.2 xxhash-3.2.0
```

6.1.1 Ignore Above PIP Errors

[33]:

```
## Finish setup
# Setup ROUGE (needed to use ROUGE evaluation metric)
!export ROUGE_HOME=/usr/local/bin/python/dist-packages/summ_eval/ROUGE-1.5.5/
!pip install -U  git+https://github.com/bheinzerling/pyrouge.git

# import modules for this notebook

from pprint import pprint
import nltk
nltk.download('stopwords')

# !pip install en_core_web_sm==3.0.0
!python -m spacy download en_core_web_sm

!pip install --upgrade transformers
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting git+https://github.com/bheinzerling/pyrouge.git
  Cloning https://github.com/bheinzerling/pyrouge.git to /tmp/pip-req-build-
rg77s6fc
  Running command git clone --filter=blob:none --quiet
https://github.com/bheinzerling/pyrouge.git /tmp/pip-req-build-rg77s6fc
  Resolved https://github.com/bheinzerling/pyrouge.git to commit
08e9cc35d713f718a05b02bf3bb2e29947d436ce
  Preparing metadata (setup.py) … done
Building wheels for collected packages: pyrouge
  Building wheel for pyrouge (setup.py) … done
  Created wheel for pyrouge: filename=pyrouge-0.1.3-py3-none-any.whl size=191923
sha256=b2a4db4d2732fa6ecd4035adb4a643075f5e2fbac0188b3124cc7ce83d40ff4d
  Stored in directory: /tmp/pip-ephem-wheel-cache-4ca3mng1/wheels/bd/07/80/f2410
50743bda1488efce41793a0b5502c97888adf191110d3
Successfully built pyrouge
Installing collected packages: pyrouge
Successfully installed pyrouge-0.1.3
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
```

```
/usr/local/lib/python3.9/dist-packages/torch/cuda/__init__.py:497: UserWarning:
Can't initialize NVML
```

```
  warnings.warn("Can't initialize NVML")
2023-03-17 10:25:37.124040: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc:64] Could
not load dynamic library 'libnvinfer.so.7'; dlerror: libnvinfer.so.7: cannot
open shared object file: No such file or directory; LD_LIBRARY_PATH:
/usr/local/nvidia/lib:/usr/local/nvidia/lib64
2023-03-17 10:25:37.126426: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc:64] Could
not load dynamic library 'libnvinfer_plugin.so.7'; dlerror:
libnvinfer_plugin.so.7: cannot open shared object file: No such file or
directory; LD_LIBRARY_PATH: /usr/local/nvidia/lib:/usr/local/nvidia/lib64
2023-03-17 10:25:37.132788: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Cannot
dlopen some TensorRT libraries. If you would like to use Nvidia GPU with
TensorRT, please make sure the missing libraries mentioned above are installed
properly.
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting en-core-web-sm==3.0.0
  Downloading https://github.com/explosion/spacy-
models/releases/download/en_core_web_sm-3.0.0/en_core_web_sm-3.0.0-py3-none-
any.whl (13.7 MB)
                           13.7/13.7 MB
46.2 MB/s eta 0:00:00
Requirement already satisfied: spacy<3.1.0,>=3.0.0 in
/usr/local/lib/python3.9/dist-packages (from en-core-web-sm==3.0.0) (3.0.6)
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.9/dist-
packages (from spacy<3.1.0,>=3.0.0->en-core-web-sm==3.0.0) (1.22.4)
Requirement already satisfied: blis<0.8.0,>=0.4.0 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.1.0,>=3.0.0->en-core-web-
sm==3.0.0) (0.7.9)
Requirement already satisfied: catalogue<2.1.0,>=2.0.3 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.1.0,>=3.0.0->en-core-web-
sm==3.0.0) (2.0.8)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.1.0,>=3.0.0->en-core-web-
sm==3.0.0) (1.0.9)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.1.0,>=3.0.0->en-core-web-
sm==3.0.0) (2.25.1)
Requirement already satisfied: wasabi<1.1.0,>=0.8.1 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.1.0,>=3.0.0->en-core-web-
sm==3.0.0) (0.10.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.9/dist-packages
(from spacy<3.1.0,>=3.0.0->en-core-web-sm==3.0.0) (3.1.2)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.1.0,>=3.0.0->en-core-web-
sm==3.0.0) (3.0.8)
```

```
Requirement already satisfied: srsly<3.0.0,>=2.4.1 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.1.0,>=3.0.0->en-core-web-
sm==3.0.0) (2.4.6)
Requirement already satisfied: pathy>=0.3.5 in /usr/local/lib/python3.9/dist-
packages (from spacy<3.1.0,>=3.0.0->en-core-web-sm==3.0.0) (0.10.1)
Requirement already satisfied: pydantic<1.8.0,>=1.7.1 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.1.0,>=3.0.0->en-core-web-
sm==3.0.0) (1.7.4)
Requirement already satisfied: thinc<8.1.0,>=8.0.3 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.1.0,>=3.0.0->en-core-web-
sm==3.0.0) (8.0.17)
Requirement already satisfied: typer<0.4.0,>=0.3.0 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.1.0,>=3.0.0->en-core-web-
sm==3.0.0) (0.3.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.9/dist-
packages (from spacy<3.1.0,>=3.0.0->en-core-web-sm==3.0.0) (23.0)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.1.0,>=3.0.0->en-core-web-
sm==3.0.0) (4.49.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.9/dist-
packages (from spacy<3.1.0,>=3.0.0->en-core-web-sm==3.0.0) (63.4.3)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.1.0,>=3.0.0->en-core-web-
sm==3.0.0) (2.0.7)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.4 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.1.0,>=3.0.0->en-core-web-
sm==3.0.0) (3.0.12)
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in
/usr/local/lib/python3.9/dist-packages (from
pathy>=0.3.5->spacy<3.1.0,>=3.0.0->en-core-web-sm==3.0.0) (6.3.0)
Requirement already satisfied: chardet<5,>=3.0.2 in
/usr/local/lib/python3.9/dist-packages (from
requests<3.0.0,>=2.13.0->spacy<3.1.0,>=3.0.0->en-core-web-sm==3.0.0) (4.0.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.9/dist-
packages (from requests<3.0.0,>=2.13.0->spacy<3.1.0,>=3.0.0->en-core-web-
sm==3.0.0) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.9/dist-packages (from
requests<3.0.0,>=2.13.0->spacy<3.1.0,>=3.0.0->en-core-web-sm==3.0.0) (2022.12.7)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/usr/local/lib/python3.9/dist-packages (from
requests<3.0.0,>=2.13.0->spacy<3.1.0,>=3.0.0->en-core-web-sm==3.0.0) (1.26.15)
Requirement already satisfied: click<7.2.0,>=7.1.1 in
/usr/local/lib/python3.9/dist-packages (from
typer<0.4.0,>=0.3.0->spacy<3.1.0,>=3.0.0->en-core-web-sm==3.0.0) (7.1.2)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.9/dist-
packages (from jinja2->spacy<3.1.0,>=3.0.0->en-core-web-sm==3.0.0) (2.1.2)
Installing collected packages: en-core-web-sm
```

```
  Attempting uninstall: en-core-web-sm
    Found existing installation: en-core-web-sm 3.4.1
    Uninstalling en-core-web-sm-3.4.1:
      Successfully uninstalled en-core-web-sm-3.4.1
Successfully installed en-core-web-sm-3.0.0
 Download and installation successful
You can now load the package via spacy.load('en_core_web_sm')
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Requirement already satisfied: transformers in /usr/local/lib/python3.9/dist-
packages (4.5.1)
Collecting transformers
  Using cached transformers-4.27.1-py3-none-any.whl (6.7 MB)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.9/dist-
packages (from transformers) (23.0)
Collecting tokenizers!=0.11.3,<0.14,>=0.11.1
  Using cached
tokenizers-0.13.2-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (7.6
MB)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.9/dist-
packages (from transformers) (1.22.4)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.9/dist-
packages (from transformers) (6.0)
Requirement already satisfied: requests in /usr/local/lib/python3.9/dist-
packages (from transformers) (2.25.1)
Collecting huggingface-hub<1.0,>=0.11.0
  Using cached huggingface_hub-0.13.2-py3-none-any.whl (199 kB)
Requirement already satisfied: filelock in /usr/local/lib/python3.9/dist-
packages (from transformers) (3.9.1)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.9/dist-packages (from transformers) (2022.6.2)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.9/dist-
packages (from transformers) (4.49.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.9/dist-packages (from huggingface-
hub<1.0,>=0.11.0->transformers) (4.5.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/usr/local/lib/python3.9/dist-packages (from requests->transformers) (1.26.15)
Requirement already satisfied: chardet<5,>=3.0.2 in
/usr/local/lib/python3.9/dist-packages (from requests->transformers) (4.0.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.9/dist-packages (from requests->transformers) (2022.12.7)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.9/dist-
packages (from requests->transformers) (2.10)
Installing collected packages: tokenizers, huggingface-hub, transformers
  Attempting uninstall: tokenizers
    Found existing installation: tokenizers 0.10.3
    Uninstalling tokenizers-0.10.3:
      Successfully uninstalled tokenizers-0.10.3
  Attempting uninstall: huggingface-hub
    Found existing installation: huggingface-hub 0.0.19
    Uninstalling huggingface-hub-0.0.19:
      Successfully uninstalled huggingface-hub-0.0.19
  Attempting uninstall: transformers
    Found existing installation: transformers 4.5.1
    Uninstalling transformers-4.5.1:
      Successfully uninstalled transformers-4.5.1
ERROR: pip's dependency resolver does not currently take into account all

the packages that are installed. This behaviour is the source of the following

dependency conflicts.

datasets 1.6.2 requires huggingface-hub<0.1.0, but you have huggingface-hub

0.13.2 which is incompatible.

summertime 1.2.1 requires transformers~=4.5.1, but you have transformers 4.27.1

which is incompatible.

Successfully installed huggingface-hub-0.13.2 tokenizers-0.13.2
transformers-4.27.1
```

6.1.2 Ignore Above PIP Errors

6.1.3 Interactive Widget

I use the widget to choose among the 8 given queries and then bring Top 10 reviews based the the Neural Network Retrieval System designed above. After that i build the corpus out of those 10 reviews and summarise them to give 20 summaries of those reviews

[34]:
```
# BERT Model
model_class, tokenizer_class, pretrained_weights = (ppb.BertModel, ppb.
 ↪BertTokenizer, 'bert-base-uncased')

# Load pretrained model/tokenizer
tokenizer = tokenizer_class.from_pretrained(pretrained_weights)
model = model_class.from_pretrained(pretrained_weights)

# Interaction Class object
interaction = Interaction(10,train_features,model,tokenizer)
# Create Widget on the console
interaction.create_widget()
interaction.show_widget()
if not len(interaction.data)>0:
  interaction.data=interaction.dis(1,[])
```

```
HBox(children=(VBox(children=(Button(style=ButtonStyle()),␣
 ↪Button(description='select 1', style=ButtonStyle())…
Selected Query:  horrible bad quality bracelet
Relavant Results:

      ID  \
0   3494
1  26246
2   3865
3  32674
4  22408
5  56494
6  52867
7  41876
8  51396
9  44534


                                                                                ␣
 ↪                                        Reviews
0                                                                     It is as␣
 ↪nice as it looks on the picture. :) I like it. :)
1                        This was a birthday gift for my 16 YO niece. She loves␣
 ↪the ring and was very happy to have received it.
2                                                         What sparkle. It is so␣
 ↪pretty and dainty. Just what I was looking for.
3                              THIS ITEM WAS A WONDERFUL SURPRISE. THE QUALITY␣
 ↪IS SO MUCH MORE THAN I COULD HAVE EVER HOPED FOR.
4  The product arrived in a very short period of time and was perfect. It was␣
 ↪described perfectly and was everything I had hoped
5                 I bought this ring for my husband and he loved it. I received␣
 ↪it when they said I would and it is a great ring
6              the product arrived in perfect condition but the shipping is␣
 ↪ridiculously slow. i will not order from them again.
7                                         I bought this as a gift for a friends␣
 ↪birthday and she loved it. It's a beautifull ring.
8                                                   very nice small sized ring I␣
 ↪can stack it with other rings for different looks
9                                                   very nice small sized ring I␣
 ↪can stack it with other rings for different looks
```

6.1.4 Creating Corpus and Assigning the returned reviews to it

[35]:
```
# Concatenating the top 10 documents to get a text block to summarize
corpus = interaction.data
corpus
```

[35]:
```
['It is as nice as it looks on the picture. :) I like it. :)',
 'This was a birthday gift for my 16 YO niece. She loves the ring and was very
```

```
happy to have received it.',
 'What sparkle. It is so pretty and dainty. Just what I was looking for.',
 'THIS ITEM WAS A WONDERFUL SURPRISE. THE QUALITY IS SO MUCH MORE THAN I COULD
HAVE EVER HOPED FOR.',
 'The product arrived in a very short period of time and was perfect. It was
described perfectly and was everything I had hoped',
 'I bought this ring for my husband and he loved it. I received it when they
said I would and it is a great ring',
 'the product arrived in perfect condition but the shipping is ridiculously
slow. i will not order from them again.',
 "I bought this as a gift for a friends birthday and she loved it. It's a
beautifull ring.",
 'very nice small sized ring I can stack it with other rings for different
looks',
 'very nice small sized ring I can stack it with other rings for different
looks']
```

6.1.5 Summarization Model

Using TextRankModel to summarize

[36]:
```
# Below code was adapted from https://huggingface.co/philschmid/
 ↪distilbart-cnn-12-6-samsum
from summertime import model
# TextRank model
textrank = model.TextRankModel()

# print(corpus[322:323])
textrank_summary = textrank.summarize(corpus)
for i in range(len(textrank_summary)):
  print("Summary Review %d: "%(i+1), textrank_summary[i])
```

```
/usr/local/lib/python3.9/dist-packages/spacy/util.py:877: UserWarning: [W095]
Model 'en_core_web_sm' (3.0.0) was trained with spaCy v3.0 and may not be 100%
compatible with the current version (3.4.4). If you see errors or degraded
performance, download a newer compatible model or retrain your custom model with
the current spaCy version. For more details and available updates, run: python
-m spacy validate
  try:
```

```
Summary Review 1:  It is as nice as it looks on the picture.
Summary Review 2:  This was a birthday gift for my 16 YO niece.
Summary Review 3:  What sparkle.
Summary Review 4:  THE QUALITY IS SO MUCH MORE THAN I COULD HAVE EVER HOPED FOR.
Summary Review 5:  The product arrived in a very short period of time and was
perfect.
Summary Review 6:  I received it when they said I would and it is a great ring
Summary Review 7:  the product arrived in perfect condition but the shipping is
```

```
ridiculously slow.
Summary Review 8:  I bought this as a gift for a friends birthday and she loved
it.
Summary Review 9:  very nice small sized ring I can stack it with other rings
for different looks
Summary Review 10:  very nice small sized ring I can stack it with other rings
for different looks
```

6.1.6 Evaluation of the summarization model

6.1.7 For getting the targets I used ChatGPT to summarise top 10 relavant reviews of the second query

I manually tried getting summaries of the reviews and added them to a targets list for evaluation of the summary model

[37]:

```
# Building target

targets=[['My 16-year-old niece was delighted to receive the ring as a birthday present.',
'Bought ring as Christmas promise gift for girlfriend; she loved it, great value.',
'Friend loved this beautiful ring I gifted for her birthday.',
'Received this ring on my birthday, love it! Any woman would adore this ring.',
'Received this ring on my birthday, love it! Any woman would adore this ring.',
'The jeweler\'s accommodation ensured timely receipt of this beautiful ring. Grateful and appreciative. Thank you!',
'Jeweler\'s accommodation ensured timely receipt of beautiful ring. Grateful and appreciative. Thank you!',
'This beautiful and durable ring is my beloved engagement ring since Feb \'09. Love it!',
'Is this ring two-in-one? It\'s beyond gorgeous and I simply love it.',
'Is this ring two-in-one? It\'s beyond gorgeous and I simply love it.'],
['Looks great in picture and I like it.',
'16 YO niece loves the ring as birthday gift.,'
,'Perfect sparkle!',
'What a surprise! Quality surpasses expectations by far.'
,'Arrived quickly, perfect and as described. Everything I hoped for.'
,'Husband loved ring, received on time, great quality as described.'
,'Product perfect, but shipping too slow. Won\'t order again from them.'
,'Beautiful ring, friend loved it as birthday gift.'
,'Nice small ring, can stack with others for varied looks.'
,'Small and nice ring, stackable for diverse looks with other rings.'],
['The ring I purchased for my husband was well-received by him. It arrived on schedule and is an excellent piece of jewelry.',
'I gave this ring as a birthday present to my sixteen-year-old niece, who was thrilled to receive it and loves it.',
'As a birthday gift for my friend, I purchased this beautiful ring which she␣
 ↪adored.',
'The product arrived quickly and was in flawless condition. It was accurately␣
 ↪described, and fulfilled all my expectations.',
'The recipient was pleasantly surprised to receive a Masonic ring made of␣
 ↪titanium and found it to be very comfortable to wear, as I had been informed.
 ↪',
'The item was of excellent quality and arrived promptly. I am thoroughly␣
 ↪pleased with it and wholeheartedly recommend it.',
'The item looks just as attractive as it does in the picture. :) I am pleased␣
 ↪with it. :)',
'I am eagerly anticipating wearing them at my son\'s wedding on June 30th as␣
 ↪they sparkle and are sure to catch everyone\'s attention.',
'Although the product arrived in perfect condition, the shipping was␣
 ↪unreasonably slow, and as a result, I will not be ordering from them again.',
'I was pleasantly surprised by the item, as it exceeded my expectations in␣
 ↪terms of quality. It was a delightful surprise.'],
['very good for everyday wear or dressing up',
'These are nice to wear when you want something casual to wear. They are very␣
 ↪comfortable.',
'ery suitable for wearing for fashionable occasions. very dressy',
'The ring is pretty enough, but the metal of the ring is very insubstantial it␣
 ↪pushes in very easily.',
'The message is very positive and it looks very pretty. I bought it for my aunt␣
 ↪as a present and the color is very nice.',
'What sparkle. It is so pretty and dainty. Just what I was looking for.',
'This lapel pin is the perfect detail to wear your colors. I plan to wear it on␣
 ↪my lapel when I wear a suit. This pinis nice enough to wear in formal␣
 ↪occasions.Wear it with pride!!',
'I have been told that the ring is very comfortable to wear and he was quite␣
 ↪surprised and please to see the Masonic ring in titanium.',
'I am looking forward to wearing them as they sparkle and catch every eye at my␣
 ↪son\'s wedding on June 30',
'It is so unique and a pleasure to wear. The stones catch the light and the␣
 ↪style is very comfortable to wear.'],
['The product lacked clarity in description, was hollow, low-quality, and␣
 ↪returned.',
'Item didn\'t meet my expectations, flimsy quality, not recommended.',
'Impressed by item quality, fast delivery. Will buy again from this seller.',
'Impressed by item quality, fast delivery. Will buy again from this seller.',
'Earrings matched description, smooth transaction, timely shipping, pleased␣
 ↪with purchase.',
'Received nice product in timely manner, faster than expected. Will reorder.',
'Item not as pictured, poor quality, funky. Seller unresponsive to contact.',
'Item was high-quality, arrived promptly. Very satisfied and highly recommend.',
'Received Italian horn in pristine condition, completely satisfied with timely␣
 ↪delivery.',
'Arrived earlier than expected. Seller always delivers on time and in excellent␣
 ↪condition.'],
['Love it! Looks as good in person as in the picture. :)',
'Lightweight, high-quality, small size. Accurately described and true to the␣
 ↪picture.',
'Gorgeous, color-matching heart pendant. Unbeatable price for such stunning␣
 ↪beauty when worn.',
'Gifted this beautiful ring to a friend on her birthday, and she adored it.',
'Arrived quickly, described perfectly, and met all my expectations. A flawless␣
 ↪experience.',
'Lovely, stackable ring in a small size for versatile and customizable looks.',
'Lovely, stackable ring in a small size for versatile and customizable looks.',
'High-quality and true to the picture. A must-have for Tiger fans with Italian␣
 ↪charm bracelets.',
'Can envision this sparkling on my girlfriend\'s sun-kissed toe, though it\'s␣
 ↪winter. A pretty sight to anticipate.',
'Solid, stunning ring, but color may disappoint - not very pink, could appear␣
 ↪lavender. Buy for design, not color.'],
['A lovely, petite ring that can be layered with others for varying styles.',
'This dainty ring can be layered with others to create diverse styles.',
'Not recommended for those seeking a 1/2 carat ring; too small and toy-like␣
 ↪when worn.',
'A stunning, delicate heart pendant that complements clothing colors and is␣
 ↪affordable.',
'Disappointing smaller size of diamonds on the ring compared to the picture␣
 ↪portrayal.',
'The ring was small for the wearer\'s finger and more suitable for a pinky␣
 ↪finger.',
'Ring appears as pictured, has a lovely color that looks beautiful when worn.',
'Excellent quality and resemblance to the picture. Recommended for Italian␣
 ↪Charm Bracelet owners and Tiger fans.',
'The ring\'s appearance is satisfactory, but the flimsy metal bends and dents␣
 ↪easily.',
'Ring arrived promptly, has an antique appearance. Recommended for fans of␣
 ↪garnet.'],
['The medical alert bracelet resembles its picture and is made of quality␣
 ↪sterling silver.',
'Product arrived quickly, matched description, and exceeded expectations.␣
 ↪Perfect in every way.',
'Looks nice, just like the picture. I like it. :)',
'This item exceeded expectations, quality surpassed hopes - a wonderful␣
 ↪surprise.',
'So sparkly, pretty, and dainty - exactly what I wanted.',
'Product arrived flawless, but shipping was excessively slow. Won\'t order from␣
 ↪them again.',
'Dainty heart pendant looks beautiful on and complements clothing colors.␣
 ↪Amazing price for beauty.',
'Item did not meet expectations; scratches on top, inconsistent staining, and␣
 ↪misaligned hinges. Unlikely to purchase in-store.',
'Lovely small ring, great for stacking with other rings to create different␣
 ↪looks.',
'Lovely small ring, great for stacking with other rings to create different␣
 ↪looks.']]
```

[38]:

```
from summertime.evaluation import BertScore

# Calculate BertScore
bert_metric = BertScore()
score = bert_metric.evaluate(textrank_summary, targets[interaction.index])
print("Bert Score: ",score.get('bert_score_f1'))
```

```
Some weights of the model checkpoint at bert-base-uncased were not used when
initializing BertModel: ['cls.seq_relationship.bias', 'cls.predictions.bias',
'cls.predictions.decoder.weight', 'cls.predictions.transform.LayerNorm.weight',
'cls.seq_relationship.weight', 'cls.predictions.transform.dense.weight',
'cls.predictions.transform.dense.bias',
'cls.predictions.transform.LayerNorm.bias']
- This IS expected if you are initializing BertModel from the checkpoint of a
model trained on another task or with another architecture (e.g. initializing a
BertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing BertModel from the checkpoint of
a model that you expect to be exactly identical (initializing a
BertForSequenceClassification model from a BertForSequenceClassification model).
```

```
hash_code: bert-base-uncased_L8_no-idf_version=0.3.12(hug_trans=4.27.1)
Bert Score:  0.6007832884788513
```

6.2 Observations

In This I used Summertime TextRankModel to summarize each review against targets produced on ChatGPT manually. I then evaluated the model using BertScore which gave a score of 0.60 which is good.

In this model I observed it basically summaries using Extractive Methods where it organised and shortens the original text which may or may not contain multiple sentences to a single setence from the original text that best represents the original text. So instead of writing the whole text the original text can be represented by a shorter version(Summary).

7 References

- https://stackoverflow.com/questions/43374920/how-to-automatically-annotate-maximum-value-in-pyplot
- https://howtothink.readthedocs.io/en/latest/PvL_H.html
- https://www.machinelearningplus.com/nlp/topic-modeling-gensim-python/
- https://www.youtube.com/watch?v=nNvPvvuPnGs&t=1s&ab_channel=RitheshSreenivasan
- https://www.machinelearningplus.com/nlp/topic-modeling-gensim-python/
- https://huggingface.co/philschmid/distilbart-cnn-12-6-samsum