NLPCoursework

March 20, 2023

#Mount Drive

[1]:
```
from google.colab import drive
drive.mount("/content/drive")
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[2]:
```
data_path = "/content/drive/My Drive/Colab Notebooks/COP509cw/Datasets/"
!ls "/content/drive/My Drive/Colab Notebooks/COP509cw/Datasets/"

dataset = 'JewelleryReviewsLSA.csv'
query = 'JewelleryReviewsQueryRelevantID.csv'
summary = 'JewelleryReviewsSummarisationTargets.csv'
```

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

[3]:
```
import pandas as pd
import nltk;
from nltk.corpus import stopwords
import string
from collections import Counter
from nltk.tokenize import word_tokenize
nltk.download('popular')
```

```
[nltk_data] Downloading collection 'popular'
[nltk_data]    |
[nltk_data]    | Downloading package cmudict to /root/nltk_data…
[nltk_data]    |   Package cmudict is already up-to-date!
[nltk_data]    | Downloading package gazetteers to /root/nltk_data…
[nltk_data]    |   Package gazetteers is already up-to-date!
[nltk_data]    | Downloading package genesis to /root/nltk_data…
[nltk_data]    |   Package genesis is already up-to-date!
[nltk_data]    | Downloading package gutenberg to /root/nltk_data…
[nltk_data]    |   Package gutenberg is already up-to-date!
[nltk_data]    | Downloading package inaugural to /root/nltk_data…
```

```
[nltk_data]    |   Package inaugural is already up-to-date!
[nltk_data]    | Downloading package movie_reviews to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package movie_reviews is already up-to-date!
[nltk_data]    | Downloading package names to /root/nltk_data…
[nltk_data]    |   Package names is already up-to-date!
[nltk_data]    | Downloading package shakespeare to /root/nltk_data…
[nltk_data]    |   Package shakespeare is already up-to-date!
[nltk_data]    | Downloading package stopwords to /root/nltk_data…
[nltk_data]    |   Package stopwords is already up-to-date!
[nltk_data]    | Downloading package treebank to /root/nltk_data…
[nltk_data]    |   Package treebank is already up-to-date!
[nltk_data]    | Downloading package twitter_samples to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package twitter_samples is already up-to-date!
[nltk_data]    | Downloading package omw to /root/nltk_data…
[nltk_data]    |   Package omw is already up-to-date!
[nltk_data]    | Downloading package omw-1.4 to /root/nltk_data…
[nltk_data]    |   Package omw-1.4 is already up-to-date!
[nltk_data]    | Downloading package wordnet to /root/nltk_data…
[nltk_data]    |   Package wordnet is already up-to-date!
[nltk_data]    | Downloading package wordnet2021 to /root/nltk_data…
[nltk_data]    |   Package wordnet2021 is already up-to-date!
[nltk_data]    | Downloading package wordnet31 to /root/nltk_data…
[nltk_data]    |   Package wordnet31 is already up-to-date!
[nltk_data]    | Downloading package wordnet_ic to /root/nltk_data…
[nltk_data]    |   Package wordnet_ic is already up-to-date!
[nltk_data]    | Downloading package words to /root/nltk_data…
[nltk_data]    |   Package words is already up-to-date!
[nltk_data]    | Downloading package maxent_ne_chunker to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package maxent_ne_chunker is already up-to-date!
[nltk_data]    | Downloading package punkt to /root/nltk_data…
[nltk_data]    |   Package punkt is already up-to-date!
[nltk_data]    | Downloading package snowball_data to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package snowball_data is already up-to-date!
[nltk_data]    | Downloading package averaged_perceptron_tagger to
[nltk_data]    |     /root/nltk_data…
[nltk_data]    |   Package averaged_perceptron_tagger is already up-
[nltk_data]    |       to-date!
[nltk_data]    |
[nltk_data]  Done downloading collection popular
```

[3]: True

1 Question 1

Pre-process the Dataset Load file into memory and Perform Tokenization (Word & Sentence), Stop word removal, Stemming, removal of short tokens

[4]:
```
# load doc into memory
def load_doc(filename):
        # open the csv file as read into memory
        df = pd.read_csv(filename, delimiter=',', header=0)

        return df

# turn a doc into clean tokens
# This code was copied from - (link)
# Source - Lab solutions for 22COP509 NLP course
def clean_doc_vocab(doc):
        tokens = word_tokenize(doc)
        # convert to lower case
        tokens = [w.lower() for w in tokens]
        #remove duplicate words
        tokens = set(tokens)
        # remove punctuation from each token
        table = str.maketrans('', '', string.punctuation)
        tokens = [w.translate(table) for w in tokens]
        # remove remaining tokens that are not alphabetic
        tokens = [word for word in tokens if word.isalpha()]
        # filter out stop words
        stop_words = set(stopwords.words('english'))
        tokens = [w for w in tokens if not w in stop_words]
        # filter out short tokens
        tokens = [word for word in tokens if len(word) > 1]

        return tokens

# load doc and add to vocab
# This code was adapted from - (link)
def add_doc_to_vocab(doc, vocab):
        # clean doc
        tokens = clean_doc_vocab(doc)
        # update counts
        vocab.update(tokens)

 # save list to file
 # # This code was copied from - [https://colab.research.google.com/drive/
↪1dTFUKVnqCJVuQckf0kbfaFNb8kK7bX5y?usp=sharing#scrollTo=NKc7lfr6D-ts]
```

```
def save_list(lines, filename):
        # convert lines to a single blob of text
        data = '\n'.join(lines)
        # open file
        file = open(filename, 'w')
        # write text
        file.write(data)
        # close file
        file.close()

def build_vocab(reviews, vocab):
        # iterate through each row of the dataframe and build vocab from token
        for index, row in reviews.iterrows():
                add_doc_to_vocab(row['Reviews'], vocab)

        # keep tokens with a min occurrence - This code was copied from - (link)
        min_occurane = 2
        tokens = [k for k,c in vocab.items() if c >= min_occurane]

        # Save vocab list in a text file
        save_list(tokens, 'vocab.txt')

#Load data
data = load_doc(data_path + dataset)

# View data summary to check for possible null records
print(data.shape)

vocab = Counter()
lines = build_vocab(data, vocab)
# print the size of the vocab
print(len(vocab))
# print the top words in the vocab
print(vocab.most_common(50))

# keep tokens with a min occurrence
min_occurane = 2
tokens = [k for k,c in vocab.items() if c >= min_occurane]
print(len(tokens))

# # This code block was copied from - [https://colab.research.google.com/drive/
 ↪1dTFUKVnqCJVvQckfOkbfaFNb8kK7bX5y?usp=sharing#scrollTo=NKc7lfr6D-ts]
# load documents, clean and return line of tokens
def doc_to_line(doc, vocab):
        # clean doc
        tokens = clean_doc_vocab(doc)
        # filter by vocab
```

```
        tokens = [w for w in tokens if w in vocab]
        return ' '.join(tokens)

 # load all docs in a directory
def process_docs(doc, vocab):
        lines = list()
        docs = list()

        # walk through all files
        for index, row in doc.iterrows():
                # load and clean the doc
                line = doc_to_line(row['Reviews'], vocab)
                # add to list
                lines.append(line)
                docs.append(row['Reviews'])

        return lines, docs

def read_file(doc):
  # load data
  file = open(doc,'rt')
  text = file.read()
  file.close()

  return text
```

```
(200, 3)
1090
[('ring', 105), ('like', 44), ('quality', 38), ('rings', 34), ('looks', 34),
('would', 33), ('look', 32), ('one', 32), ('wear', 32), ('picture', 32), ('nt',
31), ('beautiful', 31), ('love', 30), ('item', 26), ('great', 24), ('nice', 24),
('time', 19), ('bought', 19), ('silver', 17), ('small', 17), ('size', 16),
('got', 16), ('pretty', 16), ('price', 15), ('received', 15), ('perfect', 15),
('first', 14), ('recommend', 14), ('gift', 14), ('little', 14), ('looking', 14),
('color', 14), ('even', 13), ('really', 12), ('product', 12), ('buy', 12),
('looked', 12), ('definitely', 11), ('also', 11), ('diamond', 10), ('wedding',
10), ('seller', 10), ('diamonds', 10), ('right', 10), ('purchased', 10),
('wanted', 10), ('wearing', 10), ('much', 10), ('order', 10), ('stones', 10)]
450
```

Build Bag of Words

[5]:
```
# load the vocabulary
vocab_filename = 'vocab.txt'
vocab = read_file(vocab_filename)
vocab = vocab.split()
vocab = set(vocab)
```

```
# load all training reviews
reviews, docs = process_docs(data, vocab)
```

2 Question 2 - Latent Semantic Indexing (LSI)

Latent Semantic Indexing is a natural language processing technique that analyzes relationships between a set of documents and the terms they contain. Singular Value Decomposition (SVD) is used by LSI to transform the original term-document matrix into a lower-dimensional space where the relationships between terms and documents are represented as latent (hidden) concepts. This enables LSI to capture the underlying semantic meaning of words and identify related documents even when they lack many common terms. LSI has found widespread application in information retrieval, text classification, and topic modeling.

Src - Databricks Academy

2.1 2a - Retrieve Top 10 most similar reviews

Retrieval comprises of performing three primary steps—generate a representation of the query that specifies the information need, generate a representation of the document that captures the distribution over the information contained, and match the query and the document representations to estimate their mutual relevance.

In doing so, Document ranking is employed. This ranking typically involves a query and document representation steps, followed by a matching stage. Neural models can be useful either for generating good representations or in estimating relevance, or both.

Mitra, B. and Craswell, N. (2018) An Introduction to Neural Information Retrieval, Microsoft.com. doi: 10.1561/1500000061.

Perform Encoding

[6]:
```
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import TruncatedSVD
from scipy.sparse import rand
from sklearn.metrics.pairwise import cosine_similarity
# prepare words encoding of docs - TF-IDF Approach
# # This code block was copied from - https://colab.research.google.com/drive/
↪1BXr4DuL-uKdQeTHUI_jVfhyuAykrair8?usp=sharing#scrollTo=xZk_CppdfOSk
def prepare_data(train_docs, mode, vocab):
        # encode training data set
        vectorizer = CountVectorizer(vocabulary=vocab)
        transformer = TfidfTransformer(norm='l2')
        Xtrain = transformer.fit_transform(vectorizer.fit_transform(train_docs))
        return Xtrain

# # This code block was copied from - https://colab.research.google.com/drive/
↪1gonQXIxPTDk7WUsbDQ2G7neURoWP6efH#scrollTo=HFeOO9Ca-BX-&line=12&uniqifier=1
# preprocess query
```

```python
def preprocess_query(query, mode, vocab):
  line = doc_to_line(query, vocab)
  vectorizer = CountVectorizer(vocabulary=vocab)
  transformer = TfidfTransformer(norm='l2')
  encoded = transformer.fit_transform(vectorizer.fit_transform([line]))
  return encoded

Xtrain = prepare_data(reviews, 'tfidf', vocab)
trunc_SVD_model = TruncatedSVD(n_components=5)
approx_Xtrain = trunc_SVD_model.fit_transform(Xtrain)
print("Approximated Xtrain shape: " + str(approx_Xtrain.shape))
```

```
Approximated Xtrain shape: (200, 5)
```

[7]:

```python
import numpy as np

querys = ['The ring is a great gift. My friend loves it',
          'horrible bad quality bracelet',
          'arrived promptly and happy with the seller',
          'wear it with casual wear',
          'i expected better quality. i will return this item',
          'looks beautiful. The design is pretty. pefect and color is light',
          'This ring looks nothing like the picture. the diamonds are small and␣
 ↪not very noticeable',
          'braclet looked just like its picture and is nice quality sterling␣
 ↪silver.'
]


doc_ids = list()
for index, query in enumerate(querys):
  Top_n_reviews=10
  # retrieval
  encoded_query = preprocess_query(query, 'tfidf', vocab)
  # print(encoded_query.shape)

  transformed_query = trunc_SVD_model.transform(encoded_query)
  similarities = cosine_similarity(approx_Xtrain, transformed_query)
  # print("Similarities shape: " + str(similarities.shape))
  indexes = np.argsort(similarities.flat)[-Top_n_reviews:]
  doc_id = [data.iloc[indexes[i]]['ID'] for i in range(len(indexes))]
  doc_ids.append(doc_id)
  # indexes = np.argsort(similarities.flat)[::-1]

  print('_'*100)
  print(f'Query {index + 1}: {query}')
  print('='*100)
```

```python
print(f"Top {Top_n_reviews} documents retrieved:  {str(doc_id)}")
# print(f"Top {Top_n_reviews} documents retrieved:  {data.
↪iloc[indexes]["ID"]}")
similarity_score = ', '.join([str(round(score, 3)) for score in similarities.
                              flat[indexes]])
print(f"\nSimilarities scores: {similarity_score}")
print('='*100)
for i in range(Top_n_reviews, 0, -1):
  print(f"{i}th Ranked result:")
  print("Doc ID: " + str(indexes[-i]))
  # print("ID"+ str(data.iloc[indexes[i]]))
  # print(reviews[indexes[-i]])
  print(docs[indexes[-i]])
  print("Similarities: " + str(similarities.flat[indexes[-i]]))
  print('\n')
```

```
--------------------------------------------------------------------------------
--------------------
Query 1: The ring is a great gift. My friend loves it
================================================================================
====================
Top 10 documents retrieved:  [9726, 2033, 26246, 41876, 17309, 35694, 17273,
44591, 36164, 49525]

Similarities scores: 0.946, 0.956, 0.96, 0.974, 0.974, 0.974, 0.979, 0.987,
0.992, 0.997
================================================================================
====================
10th Ranked result:
Doc ID: 109
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.
Similarities: 0.9464378518512112



9th Ranked result:
Doc ID: 106
I love my birthstone and I wanted a piece of jewelry that symbolized the simple
purity of the Blue Topaz. This ring did that for me. As a gift to myself for my
birthday this year, it was definitely a great gift and a welcomed addition to my
collection.
Similarities: 0.9562317029850116



8th Ranked result:
Doc ID: 105
This was a birthday gift for my 16 YO niece. She loves the ring and was very
```

happy to have received it.
Similarities: 0.9601823211536001

7th Ranked result:
Doc ID: 114
I bought this as a gift for a friends birthday and she loved it. It's a beautifull ring.
Similarities: 0.9737436814242606

6th Ranked result:
Doc ID: 115
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
Similarities: 0.9740717255875417

5th Ranked result:
Doc ID: 113
My neice loves her birth stone so I got it for her for a Christmas Gift.I also love it also. great
Similarities: 0.9744105617823462

4th Ranked result:
Doc ID: 117
My mother loved this and was a great birthday gift. These look even better in person and go great with anything.
Similarities: 0.978740473935319

3th Ranked result:
Doc ID: 26
my husband loves it only thing is you cant have this ring resized due to the way the ring is made
Similarities: 0.9870709101596368

2th Ranked result:
Doc ID: 103
I got the ring as a promise ring for my girlfriend for Christmas and she loved it. Definitely a great value.
Similarities: 0.9917776235081937

1th Ranked result:
Doc ID: 111

```
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.
Similarities: 0.9974515774727934


--------------------------------------------------------------------------------
--------------------
Query 2: horrible bad quality bracelet
================================================================================
====================
Top 10 documents retrieved:  [4375, 10758, 1816, 265, 13373, 33571, 2114, 45548,
54748, 38305]

Similarities scores: 0.969, 0.969, 0.97, 0.97, 0.975, 0.977, 0.978, 0.985,
0.986, 0.989
================================================================================
====================
10th Ranked result:
Doc ID: 95
Very impressed with the quality of my item. Delivery was fast. Would definately
buy from this seller again
Similarities: 0.968782743270094


9th Ranked result:
Doc ID: 96
Very impressed with the quality of my item. Delivery was fast. Would definately
buy from this seller again
Similarities: 0.968782743270094


8th Ranked result:
Doc ID: 5
The quality and look were not what I had anticipated. Very flimsy.I would not
recommend this item
Similarities: 0.9695092387756503


7th Ranked result:
Doc ID: 6
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.
Similarities: 0.969746395060482


6th Ranked result:
```

Doc ID: 153
The item was not as pictured. It is funky and of poor quality. The seller did not respond when I contacted him about this.
Similarities: 0.9746027268592024

5th Ranked result:
Doc ID: 157
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.
Similarities: 0.9770901004181627

4th Ranked result:
Doc ID: 123
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.
Similarities: 0.9778037962243905

3th Ranked result:
Doc ID: 41
This is an attractive and high quality item for a young teenager. It is too small for an adult.
Similarities: 0.9854028356895472

2th Ranked result:
Doc ID: 156
Item arrived extremely damaged in several places. Not packaged well had to send it back. Very disappointed with the quality.
Similarities: 0.986475933460864

1th Ranked result:
Doc ID: 99
The flute charm is so detailed and is of very high quality. You can see all the keys, any flute fan would adore having this item.
Similarities: 0.9886809098399278

Query 3: arrived promptly and happy with the seller
==

```
Top 10 documents retrieved:  [8110, 27679, 10758, 4375, 29722, 41889, 19944,
49216, 33251, 22058]

Similarities scores: 0.963, 0.964, 0.965, 0.965, 0.97, 0.971, 0.973, 0.975,
0.992, 0.995
================================================================================
====================
10th Ranked result:
Doc ID: 93
I was very pleased with the quality of this item. Will definately reccommend
Eve's Addiction to all my friends and family.
Similarities: 0.9629886314217946


9th Ranked result:
Doc ID: 128
Item was shipped and received within the time limit given. Good quality product
t t t t t t t
Similarities: 0.9642036333457844


8th Ranked result:
Doc ID: 96
Very impressed with the quality of my item. Delivery was fast. Would definately
buy from this seller again
Similarities: 0.9653794089864282


7th Ranked result:
Doc ID: 95
Very impressed with the quality of my item. Delivery was fast. Would definately
buy from this seller again
Similarities: 0.9653794089864282


6th Ranked result:
Doc ID: 131
I received this Italian horn in pristine condition and I was completely
satisfied with the receiving of this product in a timely manner.
Similarities: 0.969525302112588


5th Ranked result:
Doc ID: 97
I was very impressed with the quality and would not hesitate to purchase other
items from the Seller. Their service was also exceptional.
Similarities: 0.9705996254909582
```

```
4th Ranked result:
Doc ID: 135
am very pleased with my purchase, speedy shipping will use again
Similarities: 0.972971925549139


3th Ranked result:
Doc ID: 138
My item came quickly and in plenty of time for Christmas. They were a huge hit
with the person who received them
Similarities: 0.9745706640774308


2th Ranked result:
Doc ID: 125
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
Similarities: 0.9922734132101019


1th Ranked result:
Doc ID: 100
Item was great quality and came promptly. I'm very happy with it and recommend
it unreservedly.
Similarities: 0.9952969379133926


--------------------------------------------------------------------------------
--------------------
Query 4: wear it with casual wear
================================================================================
====================
Top 10 documents retrieved:  [12483, 2131, 44126, 2134, 11087, 28648, 37486,
36585, 535, 19852]

Similarities scores: 0.952, 0.953, 0.975, 0.976, 0.978, 0.979, 0.985, 0.985,
0.99, 0.991
================================================================================
====================
10th Ranked result:
Doc ID: 28
They definitely help lessen your appitite, however my ears were sore after
wearing for about 3 hours and the next few days I tried to wear them off and on
and to increase the wearing time. If you have a good pain tolerance you may not
notice any discomfort, as for me my ears lobes were swollen and I had to stop
wearing them for 4 days.
```

Similarities: 0.952412407045051

9th Ranked result:
Doc ID: 143
I wanted a classy piece to wear on my right hand for work when I'm wearing Gold.
I found that I will end up wearing this outside of work. Very classy looking
Similarities: 0.9525631226554471

8th Ranked result:
Doc ID: 152
It is so unique and a pleasure to wear. The stones catch the light and the style
is very comfortable to wear.
Similarities: 0.9745112039331063

7th Ranked result:
Doc ID: 145
ery suitable for wearing for fashionable occasions. very dressy
Similarities: 0.9755194862953138

6th Ranked result:
Doc ID: 13
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
Similarities: 0.9779021063838653

5th Ranked result:
Doc ID: 140
The days I do not wear the blue one I wear this one. I really enjoy wearing
something Celtic and pretty.
Similarities: 0.9788447604217865

4th Ranked result:
Doc ID: 141
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
Similarities: 0.984803415218609

3th Ranked result:
Doc ID: 146
I am looking forward to wearing them as they sparkle and catch every eye at my
son's wedding on June 30

Similarities: 0.9853059829648068

2th Ranked result:
Doc ID: 14
It serves the purpose, but it seemed to me that the image was a lot prettier and sparklier than it turned out to be. I wear it UNDER my shirt since it does not compliment anything I wear.
Similarities: 0.9899723875818819

1th Ranked result:
Doc ID: 144
very good for everyday wear or dressing up
Similarities: 0.9910997079537142

Query 5: i expected better quality. i will return this item
==
Top 10 documents retrieved: [1816, 33571, 41889, 8110, 45548, 4375, 10758, 13373, 54748, 38305]

Similarities scores: 0.986, 0.986, 0.986, 0.991, 0.991, 0.991, 0.991, 0.994, 0.995, 0.997
==
10th Ranked result:
Doc ID: 5
The quality and look were not what I had anticipated. Very flimsy.I would not recommend this item
Similarities: 0.9856629268895757

9th Ranked result:
Doc ID: 157
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.
Similarities: 0.9861474694462455

8th Ranked result:
Doc ID: 97
I was very impressed with the quality and would not hesitate to purchase other items from the Seller. Their service was also exceptional.

Similarities: 0.9863031564674286

7th Ranked result:
Doc ID: 93
I was very pleased with the quality of this item. Will definately reccommend Eve's Addiction to all my friends and family.
Similarities: 0.9906267947437524

6th Ranked result:
Doc ID: 41
This is an attractive and high quality item for a young teenager. It is too small for an adult.
Similarities: 0.9912410103121873

5th Ranked result:
Doc ID: 95
Very impressed with the quality of my item. Delivery was fast. Would definately buy from this seller again
Similarities: 0.9913698416482508

4th Ranked result:
Doc ID: 96
Very impressed with the quality of my item. Delivery was fast. Would definately buy from this seller again
Similarities: 0.9913698416482508

3th Ranked result:
Doc ID: 153
The item was not as pictured. It is funky and of poor quality. The seller did not respond when I contacted him about this.
Similarities: 0.9936487737696234

2th Ranked result:
Doc ID: 156
Item arrived extremely damaged in several places. Not packaged well had to send it back. Very disappointed with the quality.
Similarities: 0.9952076037825203

1th Ranked result:
Doc ID: 99
The flute charm is so detailed and is of very high quality. You can see all the

```
keys, any flute fan would adore having this item.
Similarities: 0.9972884321243866


------------------------------------------------------------------------------------------------------
Query 6: looks beautiful. The design is pretty. pefect and color is light
======================================================================================================
Top 10 documents retrieved:  [43945, 27474, 12358, 32767, 28543, 42077, 46500,
41319, 39932, 45860]

Similarities scores: 0.924, 0.924, 0.927, 0.93, 0.93, 0.957, 0.962, 0.969,
0.971, 0.976
======================================================================================================
10th Ranked result:
Doc ID: 161
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.
Similarities: 0.9239421100333045


9th Ranked result:
Doc ID: 160
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.
Similarities: 0.9239421100333045


8th Ranked result:
Doc ID: 163
These look quite like their photograph. They are very colorful and you know they
are turtles. I've seen them elsewhere for quite a high price and these are
beautiful.
Similarities: 0.9271019179717989


7th Ranked result:
Doc ID: 0
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
Similarities: 0.9301791849721579
```

6th Ranked result:
Doc ID: 192
The diamond had a crack in one Garnet and another one had a large chip.
Similarities: 0.9304106264859551

5th Ranked result:
Doc ID: 45
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.
Similarities: 0.9567885798390113

4th Ranked result:
Doc ID: 159
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.
Similarities: 0.9619529234838238

3th Ranked result:
Doc ID: 164
The ring is exactly as pictured and looks very pretty on my hand. The color of the stones is rich and beautiful.
Similarities: 0.969096685207524

2th Ranked result:
Doc ID: 165
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.
Similarities: 0.9706493608870369

1th Ranked result:
Doc ID: 158
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.
Similarities: 0.9756106120431935

Query 7: This ring looks nothing like the picture. the diamonds are small and

```
not very noticeable
================================================================================
====================
Top 10 documents retrieved:  [3494, 11356, 28542, 37864, 6649, 47345, 943,
41872, 38637, 209]

Similarities scores: 0.954, 0.958, 0.967, 0.973, 0.977, 0.978, 0.987, 0.987,
0.989, 0.992
================================================================================
====================
10th Ranked result:
Doc ID: 188
It is as nice as it looks on the picture. :) I like it. :)
Similarities: 0.954007201382691


9th Ranked result:
Doc ID: 174
Although the picture shows a cute looking ring this ring isn't pretty. The
fringed look, only looks like the ring has been left on the floor and someone
ran it over with a vaccum cleaner.
Similarities: 0.9581234351658509


8th Ranked result:
Doc ID: 171
The ring was nice and looked like picture but had a crack in one Garnet and
another one had a large chip.
Similarities: 0.9670399912352006


7th Ranked result:
Doc ID: 183
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.
Similarities: 0.9727204399650787


6th Ranked result:
Doc ID: 178
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.
Similarities: 0.9765844015019487
```

```
5th Ranked result:
Doc ID: 173
I didn't like this product because the diamonds looked nothing like the picture.
The diamonds are flawed more than a little bit.
Similarities: 0.9780401549362725


4th Ranked result:
Doc ID: 182
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.
Similarities: 0.9868370605925979


3th Ranked result:
Doc ID: 191
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.
Similarities: 0.9873913912702579


2th Ranked result:
Doc ID: 176
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.
Similarities: 0.9886068639154282


1th Ranked result:
Doc ID: 170
This ring looks nothing like the picture. the diamonds are small and not very
noticeable; I will be sending this back
Similarities: 0.9923302665020217


--------------------------------------------------------------------------------
--------------------
Query 8: braclet looked just like its picture and is nice quality sterling
silver.
================================================================================
====================
Top 10 documents retrieved:  [47345, 41872, 735, 10642, 3494, 53409, 44490,
45518, 56865, 642]
```

```
Similarities scores: 0.926, 0.932, 0.94, 0.941, 0.948, 0.968, 0.98, 0.988,
0.988, 0.999
================================================================================
====================
10th Ranked result:
Doc ID: 173
I didn't like this product because the diamonds looked nothing like the picture.
The diamonds are flawed more than a little bit.
Similarities: 0.9256913540974214


9th Ranked result:
Doc ID: 191
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.
Similarities: 0.9319192802062732


8th Ranked result:
Doc ID: 189
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.
Similarities: 0.9401247764332208


7th Ranked result:
Doc ID: 185
From the picture they looked to have some purple in them but they are clear just
like the title says.
Similarities: 0.9413515984537485


6th Ranked result:
Doc ID: 188
It is as nice as it looks on the picture. :) I like it. :)
Similarities: 0.9477428350411371


5th Ranked result:
Doc ID: 193
Although the picture looks like metal beads and description states sterling
silver, these are pearls.
Similarities: 0.967850835704277
```

```
4th Ranked result:
Doc ID: 196
These are very good quality. They are light weight and nice small size. Just as
described. They look like the picture.
Similarities: 0.9801763182326474


3th Ranked result:
Doc ID: 187
It was much smaller than it looked like in the picture and the silver necklace
seemed to be of poorer quality than expected.
Similarities: 0.9880931102532827


2th Ranked result:
Doc ID: 194
Looks exactly like the picture. Very nice quality. A must for everyone who is a
Tiger fan and owns an Italian Charm Bracelet.
Similarities: 0.988340741702246


1th Ranked result:
Doc ID: 184
This medical alert braclet looked just like its picture and is nice quality
sterling silver.
Similarities: 0.9994153382721591
```

Define functions for Emperical tuning of Weighting schemes

[8]:
```python
from sklearn.feature_extraction.text import TfidfVectorizer
import matplotlib.pyplot as plt

def prepare_tf_data(train_docs, mode, vocab):
  # encode training data set
        vectorizer = CountVectorizer(vocabulary=vocab)
        transformer = TfidfTransformer(norm=None, use_idf=False,␣
 ↪sublinear_tf=True)
        Xtrain = transformer.fit_transform(vectorizer.fit_transform(train_docs))
        return Xtrain

def preprocess_tf_query(query, mode, vocab):
  line = doc_to_line(query, vocab)
  vectorizer = CountVectorizer(vocabulary=vocab)
  transformer = TfidfTransformer(norm=None, use_idf=False, sublinear_tf=True)
  encoded = transformer.fit_transform(vectorizer.fit_transform([line]))
  return encoded
```

```
# prepare words encoding of docs - Training emperically
# # This code block was copied from - https://colab.research.google.com/drive/
↪1BXr4DuL-uKdQeTHUI_jVfhyuAykrair8?usp=sharing#scrollTo=xZk_CppdfOSk
def _prepare_data(train_docs, mode, vocab):
  # Tune the LSI model
  if mode == 'tfidf':
    encoded = prepare_data(train_docs, mode, vocab)
  if mode == 'binary':
    transformer = CountVectorizer(vocabulary=vocab, binary=True)
  if mode == 'count':
    transformer = CountVectorizer(vocabulary=vocab)
  if mode == 'tf':
    encoded = prepare_tf_data(train_docs, mode, vocab)

  return encoded

# # This code block was copied from - https://colab.research.google.com/drive/
↪1gonQXIxPTDk7WUsbDQ2G7neURoWP6efH#scrollTo=HFeOO9Ca-BX-&line=12&uniqifier=1
# preprocess query
def _preprocess_query(query, mode, vocab):
  line = doc_to_line(query, vocab)
  # Tune the LSI model
  #   for scheme in weighting_schemes:
  if mode == 'tfidf':
    transformed = preprocess_query(query, mode, vocab)
  if mode == 'binary':
    transformed = CountVectorizer(vocabulary=vocab, binary=True)
  if mode == 'count':
    transformed = CountVectorizer(vocabulary=vocab)
  if mode == 'tf':
    transformed = preprocess_tf_query(query, mode, vocab)

  return transformed

Xtrain = _prepare_data(reviews, 'tfidf', vocab)
trunc_SVD_model = TruncatedSVD(n_components=5)
approx_Xtrain = trunc_SVD_model.fit_transform(Xtrain)
# print("Approximated Xtrain shape: " + str(approx_Xtrain.shape))
```

2.2 2b - Emperically Tune the LSI model

Define Evaluation Metrics

[9]:
```
# Interplot Precision for standard Recall
def InterplotPrecision(p=0.1, Precision=None, Recall=None):

    if p >= 1.0:
```

```
        p = 0.9

    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:
        # if Recall[i] >= 1.0 and Mark[1] == 0:
            r = i
            Mark[1] = 1
    y = max(Precision[l:(r+1)])
    return y

# obtain y axis for R/P curve
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
    y_axis[i] = InterplotPrecision(p=pInput, Precision=Precision, Recall=Recall)
  return y_axis

# compute Recall, Precision, F1-measure
def compute_R_P_F1(re_mark=None, QuRe_ID =None):
  Recall = []
  Precision = []
  F1measure = []
  for i in range(len(re_mark)):
    r = sum(re_mark[:(i+1)])
    Re = r/(len(QuRe_ID))
    Pr = r/(i+1)
    # avoid divisor to be 0
    FD = Re + Pr
    if FD == 0:
      FD=1
    F1 = 2*Re*Pr/FD

    Recall.append(Re)
    Precision.append(Pr)
    F1measure.append(F1)
  return Recall, Precision, F1measure
```

2.2.1 Evaluate LSI models

Deprecated *LSI*

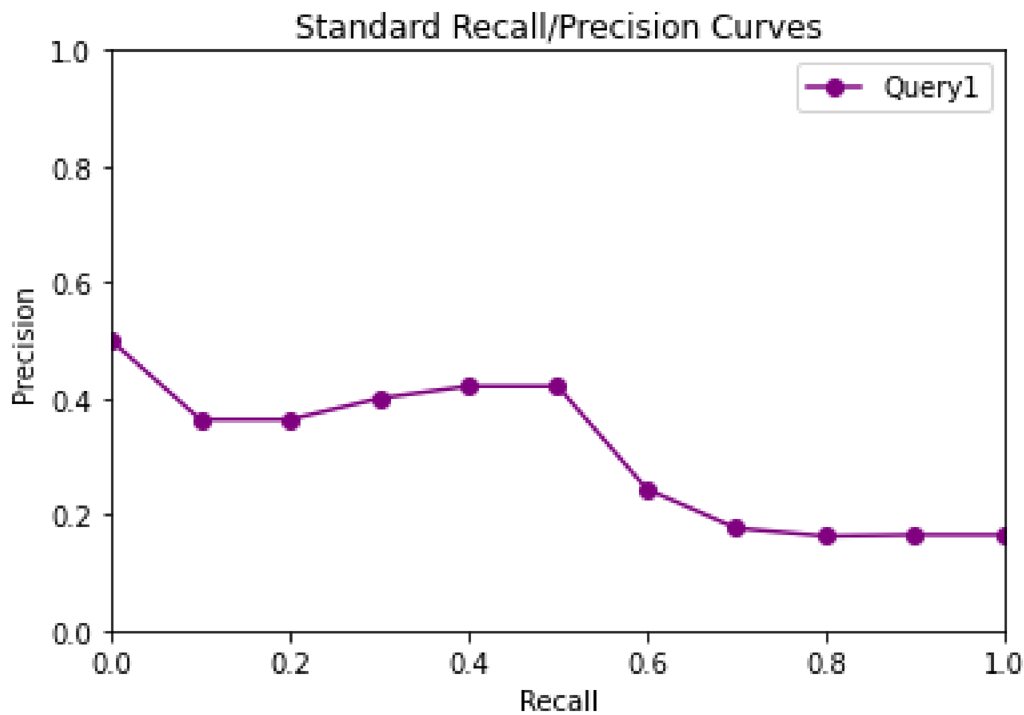
```
[10]: Xtrain = _prepare_data(reviews, 'tf', vocab)
trunc_SVD_model = TruncatedSVD(n_components=5)
approx_Xtrain = trunc_SVD_model.fit_transform(Xtrain)

re_ID = [[36164,58481,26246,2033,48779,34523,9726,56494,49525,45278,35694,41876,17309,11135,17273,1
        [57123,25299,55017,7432,2114,40871],
        [33251,17304,50019,27679,6158,22408,29722,36677,2780,17944,19944,31657,52867,49216],
        [40373,28648,37486,30640,2131,19852,2134,36585,26535,51474,21070,56330,53660,44126],
        [13373,17607,41459,54748,33571],
        [45860,46500,27474,43945,52837,12358,41319,39932,45146,50197,8341,52375],
        [209,28542,216,47345,11356,33632,38637,7110,6649,51356,44358,36165,943,37864],
        [642,10642,37794,45518,3494,735,10037,41872,28542,53409,56865,44489,44490]]


AllRecall = list()
AllPrecision = list()
AllF1measure = list()
_y_axis_lsi_tf = list()
_y_axis_lsi_tfidf = list()
# loop queries
j = 0
for query in querys:
  # retrieval
  encoded_query = _preprocess_query(query, 'tf', vocab)
  transformed_query = trunc_SVD_model.transform(encoded_query)
  similarities = cosine_similarity(approx_Xtrain, transformed_query)

  # rank the index
  indexes = np.argsort(similarities.flat)[::-1]
  doc_id = [data.iloc[indexes[i]]['ID'] for i in range(len(indexes))]

  # Mark the relevant index
  re_mark = []
  for i in range(len(indexes)):
    if (doc_id[i]) in re_ID[j]:
      re_mark.append(1)
    else:
      re_mark.append(0)
  # print(re_mark)

  # compute Recall, Precision, F1-measure
```

```
    Recall, Precision, F1measure = compute_R_P_F1(re_mark=re_mark,␣
 ↪QuRe_ID=re_ID[j])

    print('\n' + 'Query%d: '%(j+1) + query)
    # for i in range(10):
    #     print("Top " + str(i+1) + ' result: ID%d '%(indexes[i]+1),␣
 ↪ArRe_train_lines[indexes[i]])
    Recall = np.array(Recall)
    Precision = np.array(Precision)
    F1measure = np.array(F1measure)
    # print(re_mark)
    print("Recall@1~10: ", np.around(Recall[:10],2))
    print("Precision@1~10: ", np.around(Precision[:10],2))
    print("F1measure@1~10: ", np.around(F1measure[:10],2))

    # save
    AllRecall.append(Recall)
    AllPrecision.append(Precision)
    AllF1measure.append(F1measure)

    # plot R/P curve
    x_axis = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
    y_axis = compute_RP_yaxis(Precision=Precision, Recall=Recall)
    _y_axis_lsi_tfidf.append(y_axis)
    plt.plot(x_axis, y_axis, '-bo', color="purple", label="Query%d"%(j+1))
    plt.xlim(0, 1)
    plt.ylim(0, 1)
    plt.xlabel('Recall')
    plt.ylabel('Precision')
    plt.title('Standard Recall/Precision Curves')
    plt.legend()
    plt.show()

    j += 1

# compute average Recall, average Precision, average F1-measure
AllRecall = np.array(AllRecall)
AllPrecision = np.array(AllPrecision)
AllF1measure = np.array(AllF1measure)
AveRecall = (AllRecall[0] + AllRecall[1] + AllRecall[2] + AllRecall[3] +␣
 ↪AllRecall[4] + AllRecall[5] + AllRecall[6] + AllRecall[7])/8
AvePrecision = (AllPrecision[0] + AllPrecision[1]+AllPrecision[2] +␣
 ↪AllPrecision[3]+AllPrecision[4] + AllPrecision[5] + AllPrecision[6] +␣
 ↪AllPrecision[7])/8
AveF1measure = (AllF1measure[0] + AllF1measure[1]+AllF1measure[2] +␣
 ↪AllF1measure[3]+AllF1measure[4] + AllF1measure[5] + AllF1measure[6] +␣
 ↪AllF1measure[7])/8
```

```python
print("\nAverage Recall, average Precision, average F1-measure: ")
print("average Recall@1~10: ", np.around(AveRecall[:10],2))
print("average Precision@1~10: ", np.around(AvePrecision[:10],2))
print("average F1measure@1~10: ", np.around(AveF1measure[:10],2))

# plot average R/P curve
x_axis = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
y_axis = compute_RP_yaxis(Precision=AvePrecision, Recall=AveRecall)
plt.plot(x_axis, y_axis, '-bo', color="blue", label="Average")
plt.xlim(0, 1)
plt.ylim(0, 1)
plt.xlabel('average Recall')
plt.ylabel('average Precision')
plt.title('Standard Average Recall/Precision Curves')
plt.legend()
plt.show()

LSI_y_axis_avg = y_axis
```

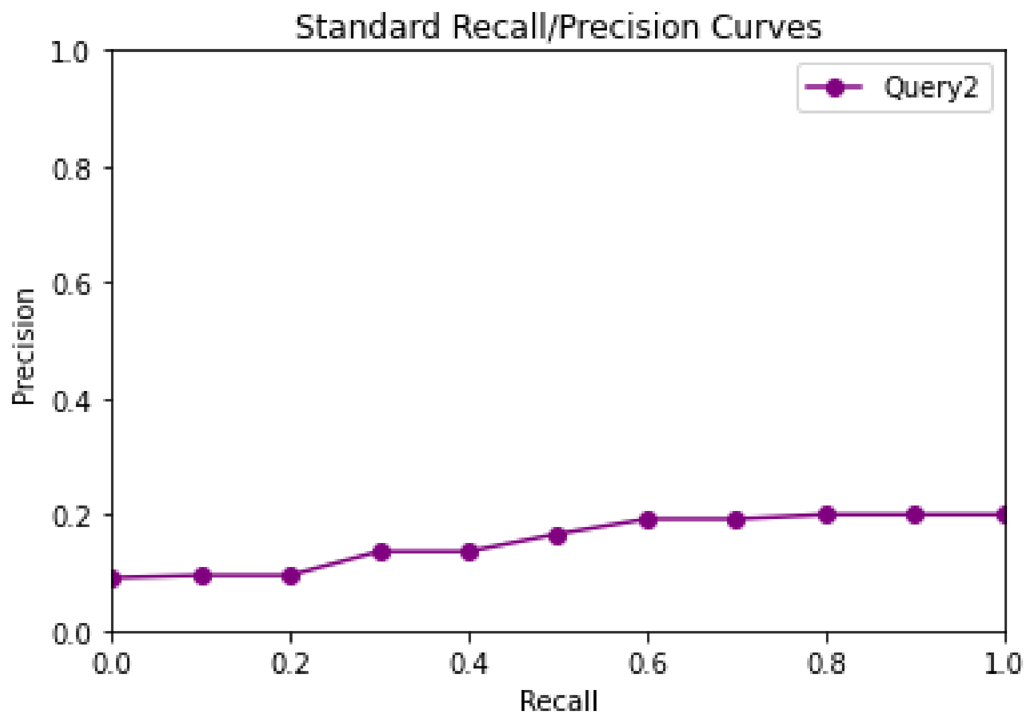
```
Query1: The ring is a great gift. My friend loves it
Recall@1~10:  [0.   0.06 0.06 0.06 0.06 0.06 0.12 0.12 0.19 0.19]
Precision@1~10:  [0.   0.5  0.33 0.25 0.2  0.17 0.29 0.25 0.33 0.3 ]
F1measure@1~10:  [0.   0.11 0.11 0.1  0.1  0.09 0.17 0.17 0.24 0.23]
```

```
<ipython-input-10-1e3085a49a4c>:64: 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="purple", label="Query%d"%(j+1))
```



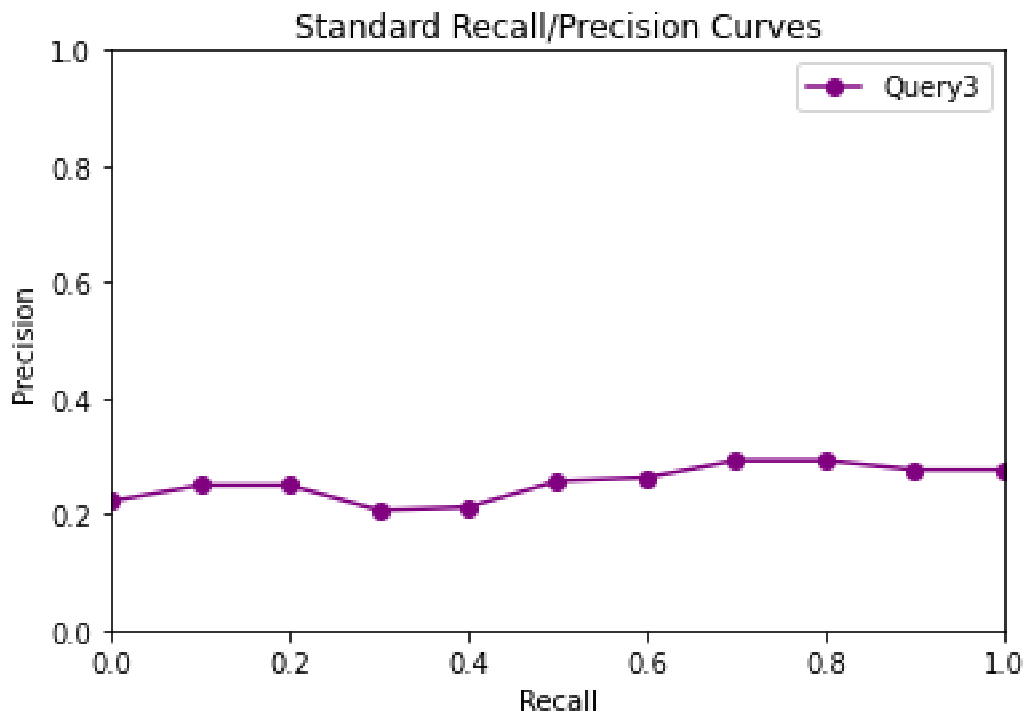

```
Query2: horrible bad quality bracelet
Recall@1~10:  [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Precision@1~10:  [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
F1measure@1~10:  [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

```
<ipython-input-10-1e3085a49a4c>:64: 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="purple", label="Query%d"%(j+1))
```



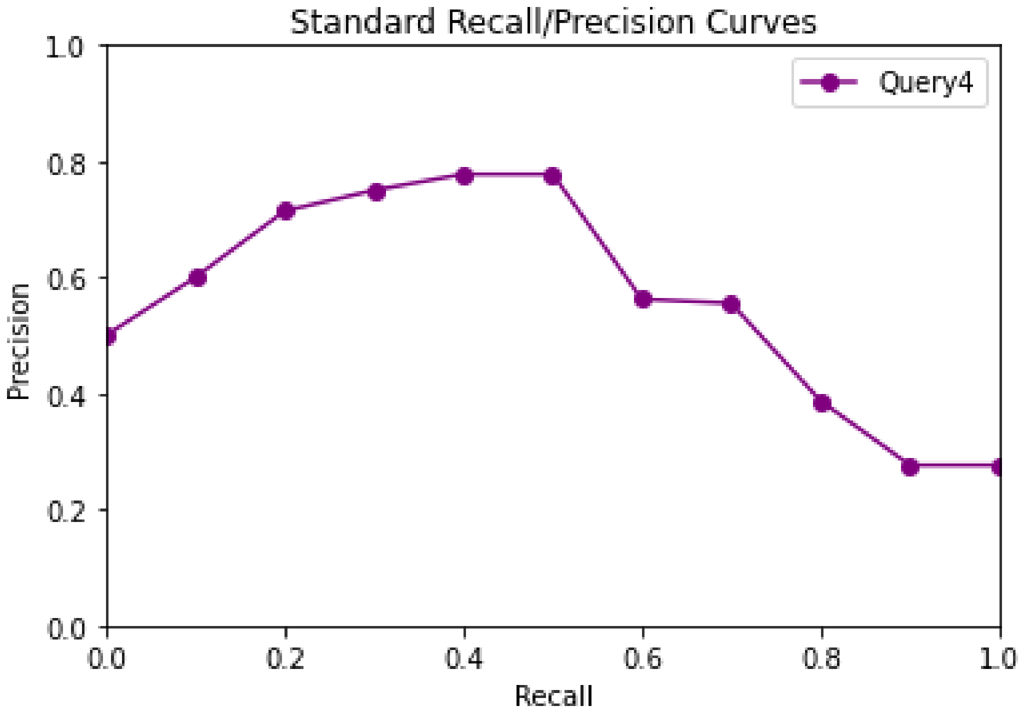

```
Query3: arrived promptly and happy with the seller
Recall@1~10:  [0.   0.   0.   0.   0.   0.07 0.07 0.07 0.14 0.14]
Precision@1~10:  [0.   0.   0.   0.   0.   0.17 0.14 0.12 0.22 0.2 ]
F1measure@1~10:  [0.   0.   0.   0.   0.   0.1  0.1  0.09 0.17 0.17]
```

```
<ipython-input-10-1e3085a49a4c>:64: 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="purple", label="Query%d"%(j+1))
```



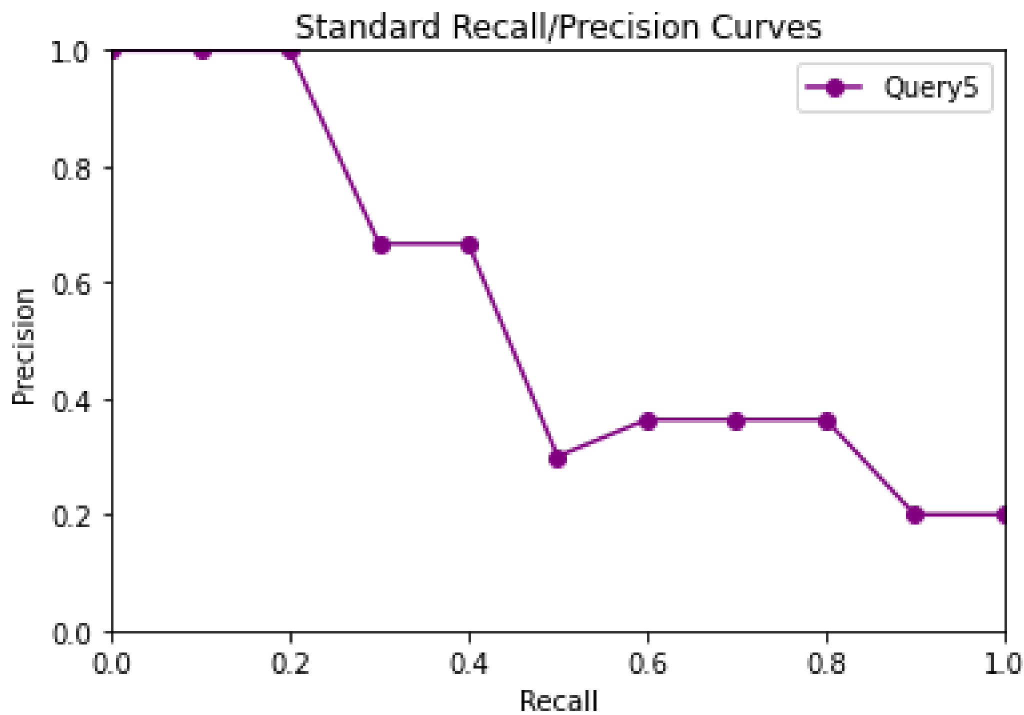

```
Query4: wear it with casual wear
Recall@1~10:  [0.   0.07 0.07 0.14 0.21 0.29 0.36 0.43 0.5  0.5 ]
Precision@1~10:  [0.   0.5  0.33 0.5  0.6  0.67 0.71 0.75 0.78 0.7 ]
F1measure@1~10:  [0.   0.12 0.12 0.22 0.32 0.4  0.48 0.55 0.61 0.58]
```

```
<ipython-input-10-1e3085a49a4c>:64: 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="purple", label="Query%d"%(j+1))
```



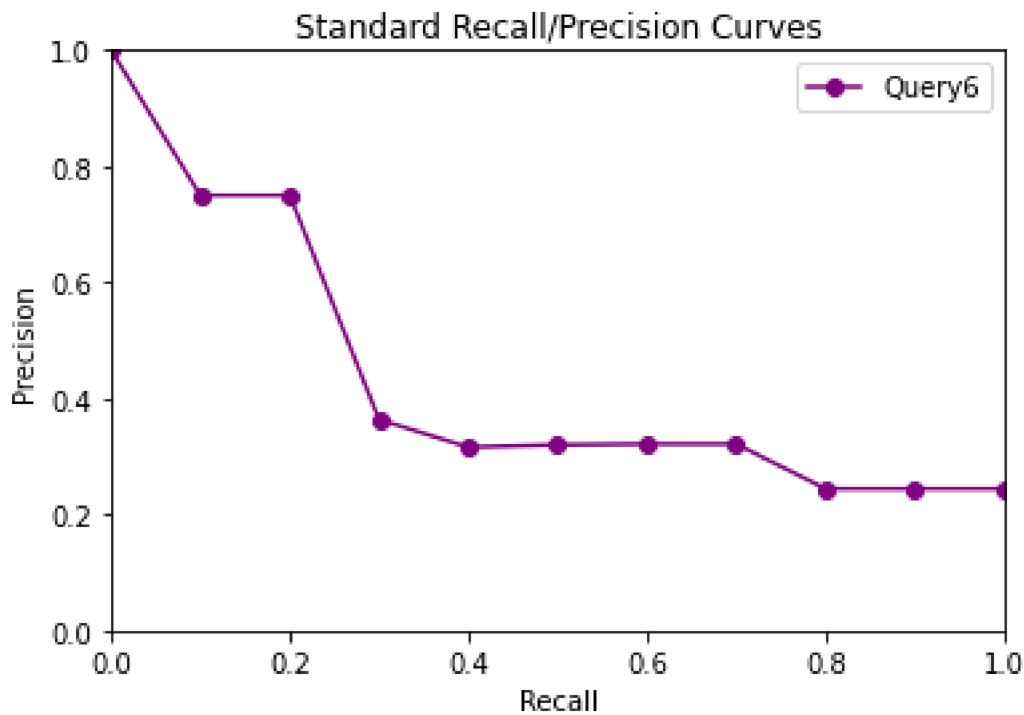

```
Query5: i expected better quality. i will return this item
Recall@1~10:  [0.2 0.2 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.6]
Precision@1~10:  [1.   0.5  0.67 0.5  0.4  0.33 0.29 0.25 0.22 0.3 ]
F1measure@1~10:  [0.33 0.29 0.5  0.44 0.4  0.36 0.33 0.31 0.29 0.4 ]
```

```
<ipython-input-10-1e3085a49a4c>:64: 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="purple", label="Query%d"%(j+1))
```



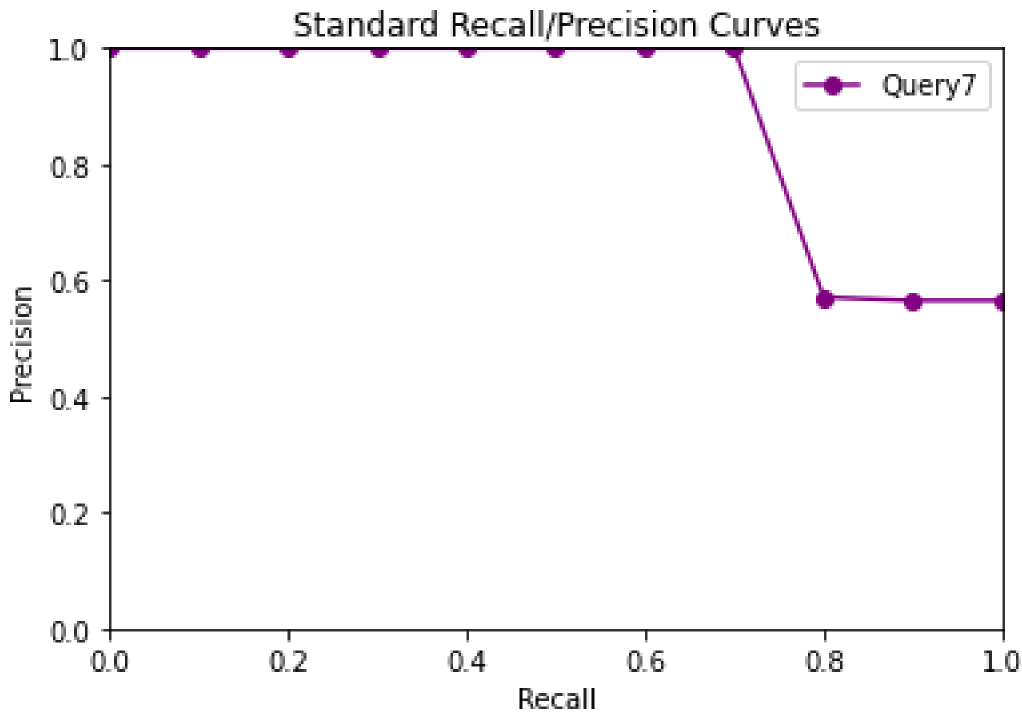

```
Query6: looks beautiful. The design is pretty. pefect and color is light
Recall@1~10:  [0.08 0.08 0.17 0.25 0.25 0.25 0.25 0.25 0.25 0.25]
Precision@1~10:  [1.   0.5  0.67 0.75 0.6  0.5  0.43 0.38 0.33 0.3 ]
F1measure@1~10:  [0.15 0.14 0.27 0.38 0.35 0.33 0.32 0.3  0.29 0.27]
```

```
<ipython-input-10-1e3085a49a4c>:64: 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="purple", label="Query%d"%(j+1))
```

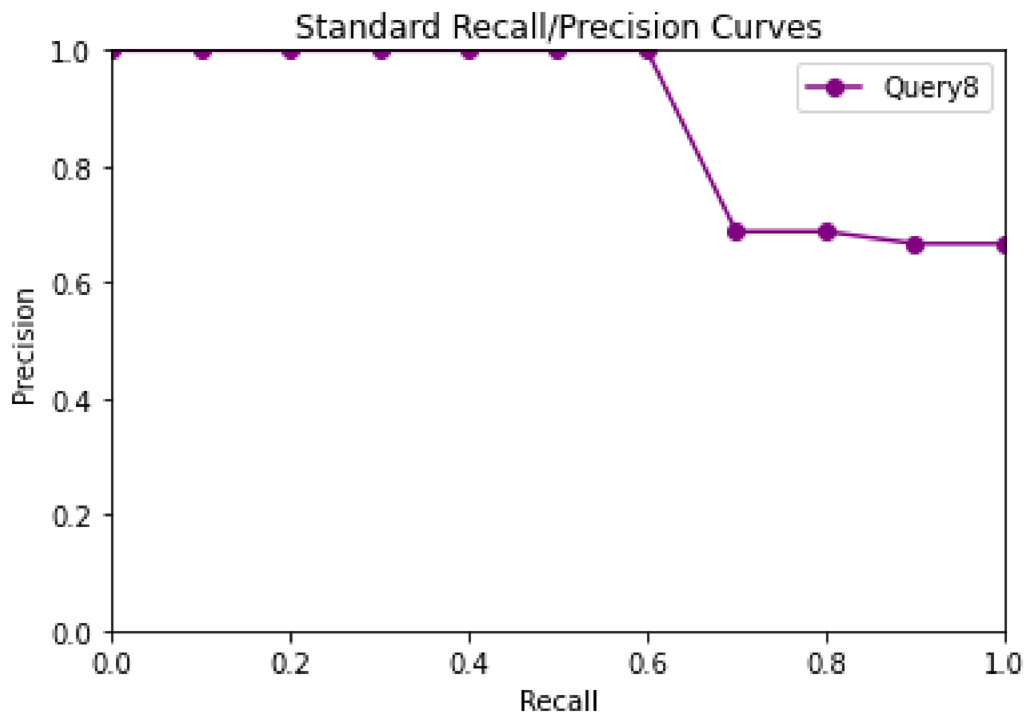



```
Query7: This ring looks nothing like the picture. the diamonds are small and not
very noticeable
Recall@1~10:  [0.07 0.14 0.21 0.29 0.36 0.43 0.5  0.57 0.64 0.71]
Precision@1~10:  [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
F1measure@1~10:  [0.13 0.25 0.35 0.44 0.53 0.6  0.67 0.73 0.78 0.83]
```

```
<ipython-input-10-1e3085a49a4c>:64: 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="purple", label="Query%d"%(j+1))
```

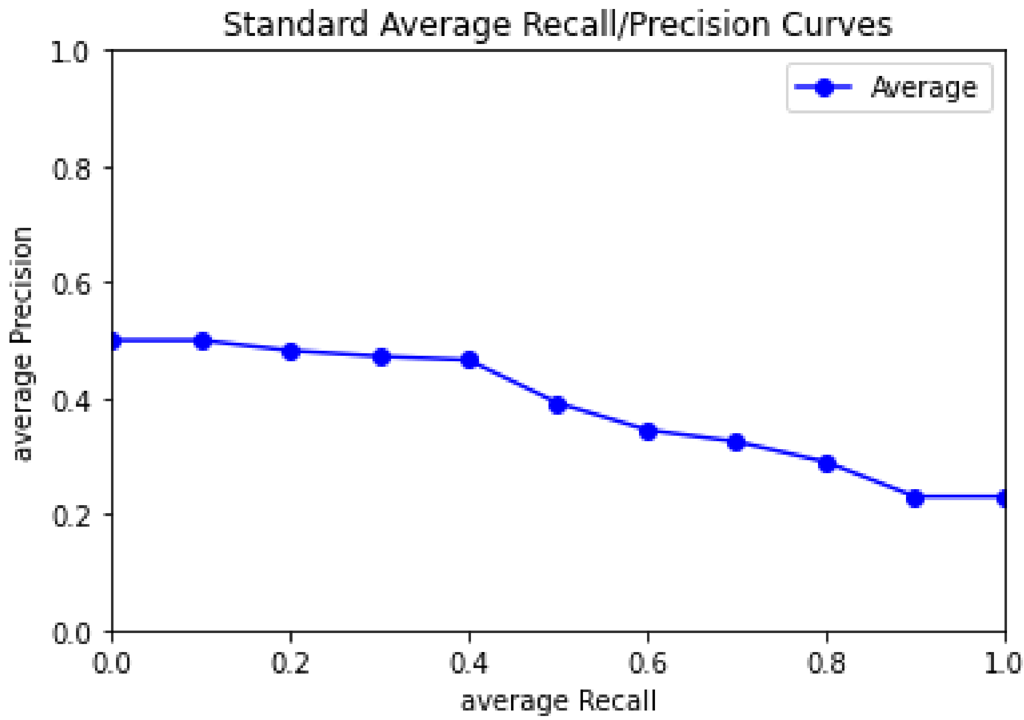



```
<ipython-input-10-1e3085a49a4c>:64: UserWarning: color is redundantly defined by
the 'color' keyword argument and the fmt string "-bo" (-> color='b'). The
keyword argument will take precedence.
  plt.plot(x_axis, y_axis, '-bo', color="purple", label="Query%d"%(j+1))


Query8: braclet looked just like its picture and is nice quality sterling
silver.
Recall@1~10:  [0.08 0.15 0.23 0.31 0.38 0.46 0.54 0.62 0.62 0.62]
Precision@1~10:  [1.   1.   1.   1.   1.   1.   1.   1.   0.89 0.8 ]
F1measure@1~10:  [0.14 0.27 0.38 0.47 0.56 0.63 0.7  0.76 0.73 0.7 ]
```




```
Average Recall, average Precision, average F1-measure:
average Recall@1~10:  [0.05 0.09 0.14 0.18 0.21 0.24 0.28 0.31 0.34 0.38]
average Precision@1~10:  [0.5  0.5  0.5  0.5  0.48 0.48 0.48 0.47 0.47 0.45]
average F1measure@1~10:  [0.1  0.15 0.21 0.26 0.28 0.31 0.35 0.36 0.39 0.4 ]
```

```
<ipython-input-10-1e3085a49a4c>:91: 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="blue", label="Average")
```



Adapted LSI implementation

Tune with TFIDF and SVD dimension of 5

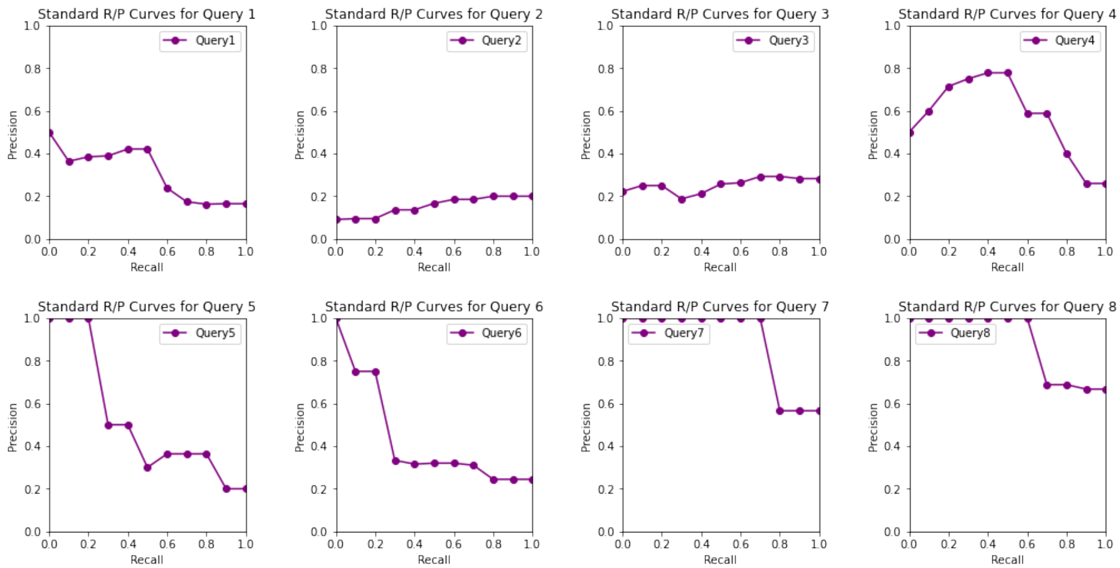
[11]:
```
import warnings
warnings.filterwarnings('ignore')
# !pip install prettytable
from prettytable import PrettyTable

# Plot table
def plot_df_table(result):
  pt = PrettyTable()
  # Add columns to the PrettyTable
  pt.field_names = ["Query", "Recall@1~10", "Precision@1~10", "F1measure@1~10"]

  # Add rows from results to the table
  for index, row in result.iterrows():
    pt.add_row([row['Query'], row['Recall@1~10'], row['Precision@1~10'],␣
↪row['F1measure@1~10']])
  print(pt)

  # Plot table
def plot_avg_table(result):
  pt = PrettyTable()
```

```
  # Add columns to the PrettyTable
  pt.field_names = ['Query', 'Average Recall@1~10', 'Average Precision@1~10',␣
↪'Average F1measure@1~10']

  # Add rows from results to the table
  for index, row in result.iterrows():
    pt.add_row([index+1, row['Average Recall@1~10'], row['Average␣
↪Precision@1~10'], row['Average F1measure@1~10']])
  print(pt)

# Define tuning parameters for weighthing schemes and SVD parameters
Xtrain = _prepare_data(reviews, 'tf', vocab)
trunc_SVD_model = TruncatedSVD(n_components=5)
approx_Xtrain = trunc_SVD_model.fit_transform(Xtrain)

re_ID =␣
↪[[36164,58481,26246,2033,48779,34523,9726,56494,49525,45278,35694,41876,17309,11135,17273,1
         [57123,25299,55017,7432,2114,40871],
        ␣
↪[33251,17304,50019,27679,6158,22408,29722,36677,2780,17944,19944,31657,52867,49216],
        ␣
↪[40373,28648,37486,30640,2131,19852,2134,36585,26535,51474,21070,56330,53660,44126],
         [13373,17607,41459,54748,33571],
        ␣
↪[45860,46500,27474,43945,52837,12358,41319,39932,45146,50197,8341,52375],
        ␣
↪[209,28542,216,47345,11356,33632,38637,7110,6649,51356,44358,36165,943,37864],
        ␣
↪[642,10642,37794,45518,3494,735,10037,41872,28542,53409,56865,44489,44490]]


AllRecall = list()
AllPrecision = list()
AllF1measure = list()
_y_axis_lsi_tf = list()
_y_axis_lsi_tfidf = list()

# Create a figure with subplots
fig, axes = plt.subplots(2, 4, figsize=(15, 8))
fig.tight_layout(pad=5.0)
axes = axes.ravel()

# Create an empty DataFrame to store results
results_df = pd.DataFrame(columns=['Query', 'Recall@1~10', 'Precision@1~10',␣
↪'F1measure@1~10'])
```

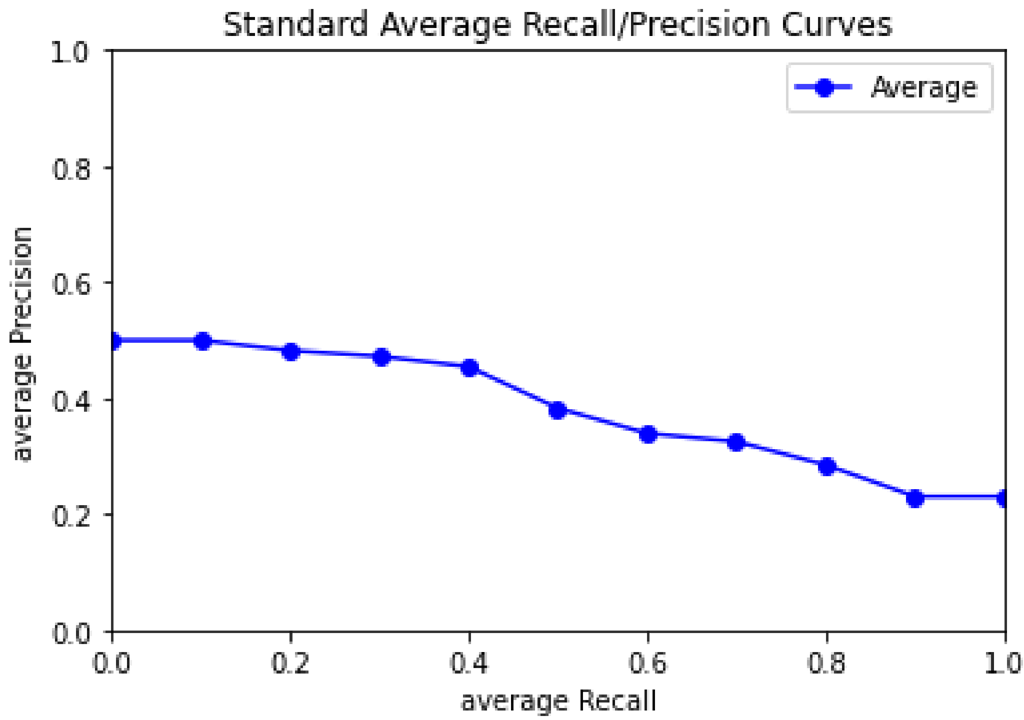
```python
# loop queries
j = 0
for query in querys:
  # retrieval
  encoded_query = _preprocess_query(query, 'tfidf', vocab)
  transformed_query = trunc_SVD_model.transform(encoded_query)
  similarities = cosine_similarity(approx_Xtrain, transformed_query)

  # rank the index
  indexes = np.argsort(similarities.flat)[::-1]
  doc_id = [data.iloc[indexes[i]]['ID'] for i in range(len(indexes))]

  # Mark the relevant index
  re_mark = []
  for i in range(len(indexes)):
    if (doc_id[i]) in re_ID[j]:
      re_mark.append(1)
    else:
      re_mark.append(0)
  # print(re_mark)

  # compute Recall, Precision, F1-measure
  Recall, Precision, F1measure = compute_R_P_F1(re_mark=re_mark,␣
↪QuRe_ID=re_ID[j])

  # Save the results in the DataFrame
  results_df.loc[j] = [f"Query{j+1}", np.around(Recall[:10], 2), np.
↪around(Precision[:10], 2), np.around(F1measure[:10], 2)]
  # save
  AllRecall.append(Recall)
  AllPrecision.append(Precision)
  AllF1measure.append(F1measure)

  # Plot R/P curve in the subplot
  x_axis = np.linspace(0, 1, 11)
  y_axis = compute_RP_yaxis(Precision=Precision, Recall=Recall)
  _y_axis_lsi_tfidf.append(y_axis)
  axes[j].plot(x_axis, y_axis, '-bo', color="purple", label="Query%d" % (j + 1))
  axes[j].set_xlim(0, 1)
  axes[j].set_ylim(0, 1)
  axes[j].set_xlabel('Recall')
  axes[j].set_ylabel('Precision')
  axes[j].set_title(f'Standard R/P Curves for Query {j + 1}')
  axes[j].legend()

  j += 1
```

```
# Print the results in a formatted table
print("\nResults for top n docs returned for each query:")
plot_df_table(results_df)

# Show the subplots
plt.show()

# print(results_df)

# compute average Recall, average Precision, average F1-measure
AllRecall = np.array(AllRecall)
AllPrecision = np.array(AllPrecision)
AllF1measure = np.array(AllF1measure)
AveRecall = (AllRecall[0] + AllRecall[1] + AllRecall[2] + AllRecall[3] +␣
 ↪AllRecall[4] + AllRecall[5] + AllRecall[6] + AllRecall[7])/8
AvePrecision = (AllPrecision[0] + AllPrecision[1]+AllPrecision[2] +␣
 ↪AllPrecision[3]+AllPrecision[4] + AllPrecision[5] + AllPrecision[6] +␣
 ↪AllPrecision[7])/8
AveF1measure = (AllF1measure[0] + AllF1measure[1]+AllF1measure[2] +␣
 ↪AllF1measure[3]+AllF1measure[4] + AllF1measure[5] + AllF1measure[6] +␣
 ↪AllF1measure[7])/8

# Create a DataFrame for average results
avg_results_df = pd.DataFrame({'Average Recall@1~10': np.around(AveRecall[10:],␣
 ↪2),
                                'Average Precision@1~10': np.
 ↪around(AvePrecision[10:], 2),
                                'Average F1measure@1~10': np.
 ↪around(AveF1measure[10:], 2)})

plot_avg_table(avg_results_df.iloc[:10])

# Plot average R/P curve
x_axis = np.linspace(0, 1, 11)
y_axis = compute_RP_yaxis(Precision=AvePrecision, Recall=AveRecall)
plt.plot(x_axis, y_axis, '-bo', color="blue", label="Average")
plt.xlim(0, 1)
plt.ylim(0, 1)
plt.xlabel('average Recall')
plt.ylabel('average Precision')
plt.title('Standard Average Recall/Precision Curves')
plt.legend()
plt.show()

LSI_y_axis_avg = y_axis
```

```
Results for top n docs returned for each query:
+--------+-------------------------------------------------------+----------------
---------------------------------------+-------------------------------------------
-----------+
| Query  |                      Recall@1~10                      |
Precision@1~10                    |                    F1measure@1~10
|
+--------+-------------------------------------------------------+----------------
---------------------------------------+-------------------------------------------
-----------+
| Query1 | [0.   0.06 0.06 0.06 0.06 0.06 0.12 0.12 0.19 0.19] | [0.   0.5  0.33
0.25 0.2  0.17 0.29 0.25 0.33 0.3 ] | [0.   0.11 0.11 0.1  0.1  0.09 0.17 0.17
0.24 0.23] |
| Query2 |            [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]            |            [0.
0. 0. 0. 0. 0. 0. 0. 0. 0.]            |            [0. 0. 0. 0. 0. 0. 0. 0. 0.
0.]            |
| Query3 | [0.   0.   0.   0.   0.   0.07 0.07 0.07 0.14 0.14] | [0.   0.   0.
0.   0.   0.17 0.14 0.12 0.22 0.2 ] | [0.   0.   0.   0.   0.   0.1  0.1  0.09
0.17 0.17] |
| Query4 | [0.   0.   0.07 0.14 0.21 0.29 0.36 0.43 0.5  0.5 ] | [0.   0.   0.33
0.5  0.6  0.67 0.71 0.75 0.78 0.7 ] | [0.   0.   0.12 0.22 0.32 0.4  0.48 0.55
0.61 0.58] |
| Query5 |     [0.2 0.2 0.2 0.4 0.4 0.4 0.4 0.4 0.4 0.6]      | [1.   0.5  0.33
0.5  0.4  0.33 0.29 0.25 0.22 0.3 ] | [0.33 0.29 0.25 0.44 0.4  0.36 0.33 0.31
0.29 0.4 ] |
| Query6 | [0.08 0.08 0.17 0.25 0.25 0.25 0.25 0.25 0.25 0.25] | [1.   0.5  0.67
0.75 0.6  0.5  0.43 0.38 0.33 0.3 ] | [0.15 0.14 0.27 0.38 0.35 0.33 0.32 0.3
0.29 0.27] |
| Query7 | [0.07 0.14 0.21 0.29 0.36 0.43 0.5  0.57 0.64 0.71] |            [1.
1. 1. 1. 1. 1. 1. 1. 1. 1.]            | [0.13 0.25 0.35 0.44 0.53 0.6  0.67 0.73
0.78 0.83] |
| Query8 | [0.08 0.15 0.23 0.31 0.38 0.46 0.54 0.62 0.62 0.62] | [1.   1.   1.
1.   1.   1.   1.   1.   0.89 0.8 ] | [0.14 0.27 0.38 0.47 0.56 0.63 0.7  0.76
0.73 0.7 ] |
+--------+-------------------------------------------------------+----------------
---------------------------------------+-------------------------------------------
-----------+
```



Query	Average Recall@1~10	Average Precision@1~10	Average F1measure@1~10
1	0.44	0.45	0.43
2	0.46	0.44	0.43
3	0.48	0.43	0.44
4	0.48	0.4	0.43
5	0.49	0.38	0.42
6	0.52	0.38	0.43
7	0.53	0.37	0.42
8	0.56	0.37	0.43
9	0.58	0.36	0.43
10	0.58	0.35	0.43



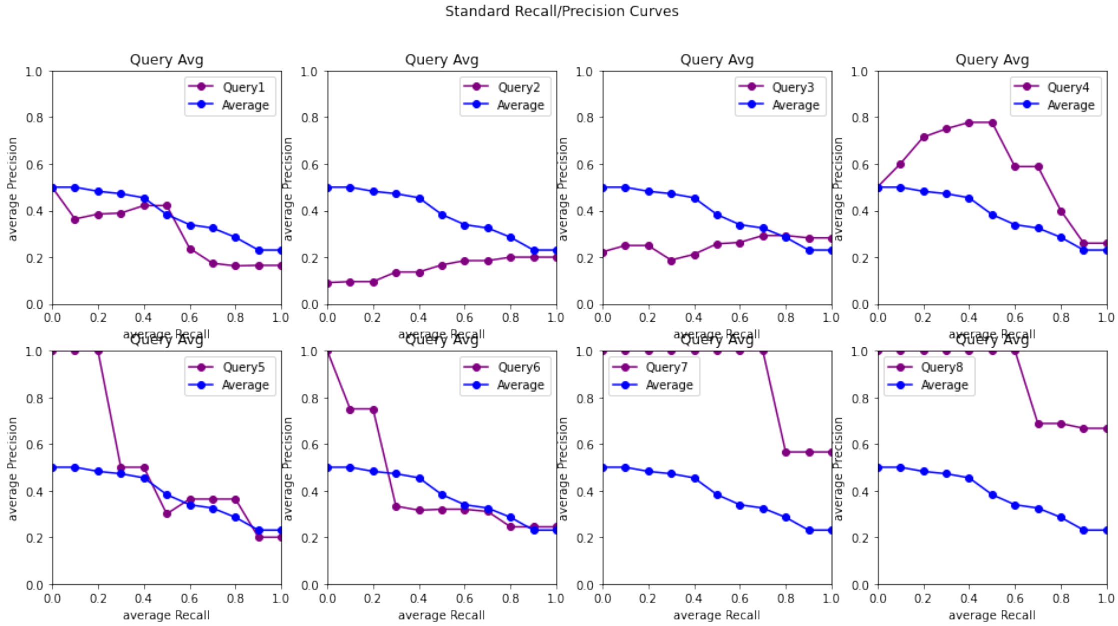
[12]:
```
import warnings
warnings.filterwarnings('ignore')

# Initialize the figure
fig, axs = plt.subplots(2, 4, figsize=(16, 8))
fig.suptitle('Standard Recall/Precision Curves')

re_ID =␣
↪[[36164,58481,26246,2033,48779,34523,9726,56494,49525,45278,35694,41876,17309,11135,17273,1
        [57123,25299,55017,7432,2114,40871],
        ␣
↪[33251,17304,50019,27679,6158,22408,29722,36677,2780,17944,19944,31657,52867,49216],
        ␣
↪[40373,28648,37486,30640,2131,19852,2134,36585,26535,51474,21070,56330,53660,44126],
        [13373,17607,41459,54748,33571],
        ␣
↪[45860,46500,27474,43945,52837,12358,41319,39932,45146,50197,8341,52375],
        ␣
↪[209,28542,216,47345,11356,33632,38637,7110,6649,51356,44358,36165,943,37864],
        ␣
↪[642,10642,37794,45518,3494,735,10037,41872,28542,53409,56865,44489,44490]]

AllRecall = list()
```

```python
AllPrecision = list()
AllF1measure = list()
_y_axis_lsi_tf = list()
_y_axis_lsi_tfidf = list()

# Loop queries
for j, query in enumerate(querys):
  # Retrieval
  encoded_query = _preprocess_query(query, 'tfidf', vocab)
  transformed_query = trunc_SVD_model.transform(encoded_query)
  similarities = cosine_similarity(approx_Xtrain, transformed_query)

  # Rank the index
  indexes = np.argsort(similarities.flat)[::-1]
  doc_id = [data.iloc[indexes[i]]['ID'] for i in range(len(indexes))]

  # Mark the relevant index
  re_mark = []
  for i in range(len(indexes)):
    if doc_id[i] in re_ID[j]:
      re_mark.append(1)
    else:
      re_mark.append(0)

  # Compute Recall, Precision, F1-measure
  Recall, Precision, F1measure = compute_R_P_F1(re_mark=re_mark,␣
↪QuRe_ID=re_ID[j])

  print('\n' + 'Query%d: '%(j+1) + query)
  print("Recall@1~10: ", np.around(Recall[:10],2))
  print("Precision@1~10: ", np.around(Precision[:10],2))
  print("F1measure@1~10: ", np.around(F1measure[:10],2))

  # Save
  AllRecall.append(Recall)
  AllPrecision.append(Precision)
  AllF1measure.append(F1measure)

   # Plot R/P curve
  x_axis = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]

  # Compute y-axis
  Recall = np.array(Recall)
  Precision = np.array(Precision)
  y_axis = compute_RP_yaxis(Precision=Precision, Recall=Recall)

  # Save y-axis for averaging later
```

```python
    _y_axis_lsi_tfidf.append(y_axis)

    # Plot R/P curve in subplot
    row = j // 4
    col = j % 4
    axs[row, col].plot(x_axis, y_axis, '-bo', color="purple",␣
↪label="Query%d"%(j+1))
    axs[row, col].set_xlim(0, 1)
    axs[row, col].set_ylim(0, 1)
    axs[row, col].set_xlabel('Recall')
    axs[row, col].set_ylabel('Precision')
    axs[row, col].set_title('Query %d'%(j+1))
    axs[row, col].legend()

# Compute average Recall, average Precision, average F1-measure
AllRecall = np.array(AllRecall)
AllPrecision = np.array(AllPrecision)
AllF1measure = np.array(AllF1measure)
AveRecall = np.mean(AllRecall, axis=0)
AvePrecision = np.mean(AllPrecision, axis=0)
AveF1measure = np.mean(AllF1measure, axis=0)

# Plot average R/P curve
y_axis = compute_RP_yaxis(Precision=AvePrecision, Recall=AveRecall)
for i in range(2):
  for j in range(4):
    axs[i, j].plot(x_axis, y_axis, '-bo', color="blue", label="Average")
    axs[i, j].set_xlim(0, 1)
    axs[i, j].set_ylim(0, 1)
    axs[i, j].set_xlabel('average Recall')
    axs[i, j].set_ylabel('average Precision')
    axs[i, j].set_title('Query Avg')
    axs[i, j].legend()

plt.show()
```

```
Query1: The ring is a great gift. My friend loves it
Recall@1~10:  [0.   0.06 0.06 0.06 0.06 0.06 0.12 0.12 0.19 0.19]
Precision@1~10:  [0.   0.5  0.33 0.25 0.2  0.17 0.29 0.25 0.33 0.3 ]
F1measure@1~10:  [0.   0.11 0.11 0.1  0.1  0.09 0.17 0.17 0.24 0.23]

Query2: horrible bad quality bracelet
Recall@1~10:  [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Precision@1~10:  [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
F1measure@1~10:  [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

```
Query3: arrived promptly and happy with the seller
Recall@1~10:  [0.   0.   0.   0.   0.   0.07 0.07 0.07 0.14 0.14]
Precision@1~10:  [0.   0.   0.   0.   0.   0.17 0.14 0.12 0.22 0.2 ]
F1measure@1~10:  [0.   0.   0.   0.   0.   0.1  0.1  0.09 0.17 0.17]

Query4: wear it with casual wear
Recall@1~10:  [0.   0.   0.07 0.14 0.21 0.29 0.36 0.43 0.5  0.5 ]
Precision@1~10:  [0.   0.   0.33 0.5  0.6  0.67 0.71 0.75 0.78 0.7 ]
F1measure@1~10:  [0.   0.   0.12 0.22 0.32 0.4  0.48 0.55 0.61 0.58]

Query5: i expected better quality. i will return this item
Recall@1~10:  [0.2 0.2 0.2 0.4 0.4 0.4 0.4 0.4 0.4 0.6]
Precision@1~10:  [1.   0.5  0.33 0.5  0.4  0.33 0.29 0.25 0.22 0.3 ]
F1measure@1~10:  [0.33 0.29 0.25 0.44 0.4  0.36 0.33 0.31 0.29 0.4 ]

Query6: looks beautiful. The design is pretty. pefect and color is light
Recall@1~10:  [0.08 0.08 0.17 0.25 0.25 0.25 0.25 0.25 0.25 0.25]
Precision@1~10:  [1.   0.5  0.67 0.75 0.6  0.5  0.43 0.38 0.33 0.3 ]
F1measure@1~10:  [0.15 0.14 0.27 0.38 0.35 0.33 0.32 0.3  0.29 0.27]

Query7: This ring looks nothing like the picture. the diamonds are small and not
very noticeable
Recall@1~10:  [0.07 0.14 0.21 0.29 0.36 0.43 0.5  0.57 0.64 0.71]
Precision@1~10:  [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
F1measure@1~10:  [0.13 0.25 0.35 0.44 0.53 0.6  0.67 0.73 0.78 0.83]

Query8: braclet looked just like its picture and is nice quality sterling
silver.
Recall@1~10:  [0.08 0.15 0.23 0.31 0.38 0.46 0.54 0.62 0.62 0.62]
Precision@1~10:  [1.   1.   1.   1.   1.   1.   1.   1.   0.89 0.8 ]
F1measure@1~10:  [0.14 0.27 0.38 0.47 0.56 0.63 0.7  0.76 0.73 0.7 ]
```



Tune Tf (Sublinear) and SVD dimension of 3

---

3 3 - Neural Information retrieval

Approach taken for Neural Information Retreival (NIR)

Overall, the cosine similarity approach to NIR is based on the ability of word embedding techniques to capture semantic relationships between words and the cosine similarity measure to compare query and document embeddings. This method has yielded promising results in improving the accuracy of traditional IR models.

Approach employed involved encoding the query and document text into vector representations, computing their cosine similarity, ranking the documents based on their scores, and evaluating the model's accuracy using metrics such as precision, recall, and F1 score using BERT.

[13]:
```
# load all docs in a directory
def __process_query(queries, vocab):
        lines = list()
        # walk through all files in the folder
        for doc in queries:
                # print(len(doc))
```

```
            line = doc_to_line(doc, vocab)
            # add to list
            lines.append(line)
    return lines

train_docs = reviews
query_docs = __process_query(querys, vocab)

train_docs = pd.Series(train_docs)
query_docs = pd.Series(query_docs)

# print(train_docs.shape)
# print(test_docs.shape)
```

[14]:
```
!pip install transformers
```

```
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 (4.27.1)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.9/dist-
packages (from transformers) (23.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.11.0 in
/usr/local/lib/python3.9/dist-packages (from transformers) (0.13.2)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.9/dist-packages (from transformers) (2022.10.31)
Requirement already satisfied: tokenizers!=0.11.3,<0.14,>=0.11.1 in
/usr/local/lib/python3.9/dist-packages (from transformers) (0.13.2)
Requirement already satisfied: requests in /usr/local/lib/python3.9/dist-
packages (from transformers) (2.27.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.9/dist-
packages (from transformers) (6.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.9/dist-
packages (from transformers) (3.10.0)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.9/dist-
packages (from transformers) (4.65.0)
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: urllib3<1.27,>=1.21.1 in
/usr/local/lib/python3.9/dist-packages (from requests->transformers) (1.26.15)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.9/dist-
packages (from requests->transformers) (3.4)
```

```
Requirement already satisfied: charset-normalizer~=2.0.0 in
/usr/local/lib/python3.9/dist-packages (from requests->transformers) (2.0.12)
```

[15]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import cross_val_score
import torch
import transformers as ppb
import warnings
warnings.filterwarnings('ignore')
```

[16]:
```
# Loading the Pre-trained BERT model
# For DistilBERT:
model_class, tokenizer_class, pretrained_weights = (ppb.DistilBertModel, ppb.
↪DistilBertTokenizer, 'distilbert-base-uncased')

## Want BERT instead of distilBERT? Uncomment the following line:
#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)
```

```
Some weights of the model checkpoint at distilbert-base-uncased were not used
when initializing DistilBertModel: ['vocab_transform.bias',
'vocab_transform.weight', 'vocab_projector.bias', 'vocab_layer_norm.bias',
'vocab_layer_norm.weight', 'vocab_projector.weight']
- This IS expected if you are initializing DistilBertModel 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 DistilBertModel from the
checkpoint of a model that you expect to be exactly identical (initializing a
BertForSequenceClassification model from a BertForSequenceClassification model).
```

[17]:
```
N = len(train_docs)

with torch.no_grad():
    # Tokenization
    tokenized = train_docs[0:N].apply((lambda x: tokenizer.encode(x,␣
↪add_special_tokens=True)))

    # padding
```

```
    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')

        # print(len(i))
        if len(i) > max_len:
            max_len = len(i)
        q += 1

    padded = np.array([i + [0]*(max_len-len(i)) for i in tokenized.values])
    np.array(padded).shape

    # masking
    attention_mask = np.where(padded != 0, 1, 0)
    attention_mask.shape

    # run the model
    input_ids = torch.tensor(padded)
    attention_mask = torch.tensor(attention_mask)

    print(input_ids.shape)

    last_hidden_states = model(input_ids, attention_mask=attention_mask)

    train_features = last_hidden_states[0][:,0,:].numpy()

print(len(train_features))
```

```
torch.Size([200, 49])
200
```

[18]:

```
N2 = len(query_docs)

with torch.no_grad():
    # Tokenization
    tokenized = query_docs[0:N2].apply((lambda x: tokenizer.encode(x,␣
 ↪add_special_tokens=True)))

    # padding
    max_len = 0
```

```python
    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')

        # print(len(i))
        if len(i) > max_len:
            max_len = len(i)
        q += 1

    padded = np.array([i + [0]*(max_len-len(i)) for i in tokenized.values])
    np.array(padded).shape
    # masking
    attention_mask = np.where(padded != 0, 1, 0)
    attention_mask.shape
    # run the model
    input_ids = torch.tensor(padded)
    attention_mask = torch.tensor(attention_mask)

    print(input_ids.shape)

    last_hidden_states = model(input_ids, attention_mask=attention_mask)

    query_features = last_hidden_states[0][:,0,:].numpy()

print(len(query_features))
```

```
torch.Size([8, 9])
8
```

3.1 3a - Evaluate approach 1

[19]:
```python
# Top_n_rankings = 3
# # Compute relevance scores and rank documents
# bert_similarity_scores = cosine_similarity(query_features, train_features)
# ranking = bert_similarity_scores.argsort()[:, ::-1]
# # indexes = np.argsort(bert_similarity_scores.flat)[-Top_n_rankings:]
# # print(ranking)
# # print("&&&&&")
# # print(indexes)
# # print("&&&&&")
# # print(ranking[0, :1][0])
```

```
# doc_list = list()
# AllRecall = []
# AllPrecision = []
# AllF1measure = []
# # loop queries
# j = 0
# for key, value in enumerate(ranking):
#   query_ids = ranking[key, :Top_n_rankings]
#   indexes = np.argsort(query_ids)

#   d_id = [i for i in query_ids]
#   ndoc_id = [data.iloc[k]['ID'] for k in query_ids]
#   ndoc_text = [data.iloc[k]['Reviews'] for k in query_ids]
#   # print(query_ids)
#   print(d_id)
#   print(indexes)
#   print('**')
#   print(ndoc_id)
#   # print(ndoc_text)
#   doc_list.append(ndoc_id)
#   print('_'*100)
#   print(f'Query ids {key + 1}: {querys[key]}')
#   # print(f'Query {key + 1}: {querys[key]}')
#   print('='*100)
#   print(f"Retrieved documents:  {str(ndoc_id)}")
#   # BERT_similarity_scores = ', '.join([str(round(score, 3)) for score in␣
 ↪query_ids])
#   BERT_similarity_score = ', '.join([str(round(score, 3)) for score in␣
 ↪bert_similarity_scores.
#                                   flat[d_id]])
#   print(f"\nSimilarities scores: {BERT_similarity_score}")
#   print('='*100)
#   for i in range(Top_n_reviews, 0, -1):
#     print(f"{i}th Ranked result:")
#     print("Doc ID: " + str(indexes[-i]))
#     # print("ID"+ str(data.iloc[indexes[i]]))
#     # print(reviews[indexes[-i]])
#     print(docs[indexes[-i]])
#     print("Similarities: " + str(similarities.flat[indexes[-i]]))
#     print('\n')

#   for q in query_ranks:
#     print(q)
#     print(data.iloc[q]['ID'])
#   print(data.iloc[ranking[i, :Top_n_rankings][i]])
#   # Mark the relevant index
```

```
# bert_re_mark = []
# for i in range(len(query_ranks)):
#   print(ndoc_id[i])
#   print(re_ID[i])
#   if (ndoc_id[i]) in re_ID[i]:
#     bert_re_mark.append(1)
#   else:
#     bert_re_mark.append(0)
#   # print(re_mark)

# # compute Recall, Precision, F1-measure
# Recall, Precision, F1measure = compute_R_P_F1(re_mark=bert_re_mark,␣
↪QuRe_ID=re_ID[j])

# print('\n' + 'Query%d: '%(j+1) + querys[i])
# # for i in range(10):
# #   print("Top " + str(i+1) + ' result: ID%d '%(indexes[i]+1),␣
↪ArRe_train_lines[indexes[i]])
# Recall = np.array(Recall)
# Precision = np.array(Precision)
# F1measure = np.array(F1measure)
# # print(re_mark)
# print("Recall@1~10: ", np.around(Recall[:10],2))
# print("Precision@1~10: ", np.around(Precision[:10],2))
# print("F1measure@1~10: ", np.around(F1measure[:10],2))

# # save
# AllRecall.append(Recall)
# AllPrecision.append(Precision)
# AllF1measure.append(F1measure)

# # plot R/P curve
# x_axis = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
# y_axis = compute_RP_yaxis(Precision=Precision, Recall=Recall)
# plt.plot(x_axis, y_axis, '-bo', color="purple", label="Query%d"%(j+1))
# plt.xlim(0, 1)
# plt.ylim(0, 1)
# plt.xlabel('Recall')
# plt.ylabel('Precision')
# plt.title('Standard Recall/Precision Curves')
# plt.legend()
# plt.show()

# j += 1

# # compute average Recall, average Precision, average F1-measure
# AllRecall = np.array(AllRecall)
```

```
# AllPrecision = np.array(AllPrecision)
# AllF1measure = np.array(AllF1measure)
# AveRecall = (AllRecall[:,0] + AllRecall[:,1] + AllRecall[:,2] + AllRecall[:
↪,3] + AllRecall[:,4] + AllRecall[:,5] + AllRecall[:,6] + AllRecall[:,7])/8
# AvePrecision = (AllPrecision[0] + AllPrecision[1]+AllPrecision[2] +␣
↪AllPrecision[3]+AllPrecision[4] + AllPrecision[5] + AllPrecision[6] +␣
↪AllPrecision[7])/8
# AveF1measure = (AllF1measure[0] + AllF1measure[1]+AllF1measure[2] +␣
↪AllF1measure[3]+AllF1measure[4] + AllF1measure[5] + AllF1measure[6] +␣
↪AllF1measure[7])/8

# print("\nAverage Recall, average Precision, average F1-measure: ")
# print("average Recall@1~10: ", np.around(AveRecall[:10],2))
# print("average Precision@1~10: ", np.around(AvePrecision[:10],2))
# print("average F1measure@1~10: ", np.around(AveF1measure[:10],2))

# # plot average R/P curve
# x_axis = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
# y_axis = compute_RP_yaxis(Precision=AvePrecision, Recall=AveRecall)
# plt.plot(x_axis, y_axis, '-bo', color="blue", label="Average")
# plt.xlim(0, 1)
# plt.ylim(0, 1)
# plt.xlabel('average Recall')
# plt.ylabel('average Precision')
# plt.title('Standard Average Recall/Precision Curves')
# plt.legend()
# plt.show()
```

3.2 Aproach 3

[20]:
```
# Top_n_rankings = 3

# # Compute relevance scores and rank documents
# bert_similarity_scores = cosine_similarity(query_features, train_features)
# ranking = bert_similarity_scores.argsort()[:, ::-1]

# doc_list = []
# AllRecall = []
# AllPrecision = []
# AllF1measure = []

# # Loop through queries
# for key, value in enumerate(ranking):
#     query_ids = ranking[key, :Top_n_rankings]
#     indexes = np.argsort(query_ids)[::-1]
```

```
#     d_id = [i for i in query_ids]
#     ndoc_id = [data.iloc[k]['ID'] for k in query_ids]
#     ndoc_text = [data.iloc[k]['Reviews'] for k in query_ids]

#     doc_list.append(ndoc_id)
#     print('_'*100)
#     print(f'Query ids {key + 1}: {querys[key]}')
#     print('='*100)
#     print(f"Retrieved documents:  {str(ndoc_id)}")

#     # Compute cosine similarity for each retrieved document and print results
#     for i in range(Top_n_rankings):
#         print(f"{i+1}th ranked result:")
#         doc_index = indexes[i]
#         doc_id = ndoc_id[doc_index]
#         doc_text = ndoc_text[doc_index]
#         similarity_score = bert_similarity_scores[key, doc_index]
#         print(f"Doc ID: {doc_id}")
#         print(f"Doc Text: {doc_text}")
#         print(f"Similarity Score: {similarity_score}\n")

#         # Mark the relevant index
#     re_mark = []
#     for i in range(len(indexes)):
#       if (doc_list[i]) in re_ID[j]:
#         re_mark.append(1)
#       else:
#         re_mark.append(0)
#     # print(re_mark)

#     # compute Recall, Precision, F1-measure
#     Recall, Precision, F1measure = compute_R_P_F1(re_mark=re_mark,␣
↪QuRe_ID=re_ID[j])

#     print('\n' + 'Query%d: '%(j+1) + query)
#     # for i in range(10):
#     #   print("Top " + str(i+1) + ' result: ID%d '%(indexes[i]+1),␣
↪ArRe_train_lines[indexes[i]])
#     Recall = np.array(Recall)
#     Precision = np.array(Precision)
#     F1measure = np.array(F1measure)
#     # print(re_mark)
#     print("Recall@1~10: ", np.around(Recall[:10],2))
#     print("Precision@1~10: ", np.around(Precision[:10],2))
#     print("F1measure@1~10: ", np.around(F1measure[:10],2))

#     # save
```

```
#     AllRecall.append(Recall)
#     AllPrecision.append(Precision)
#     AllF1measure.append(F1measure)

#     # plot R/P curve
#     x_axis = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
#     y_axis = compute_RP_yaxis(Precision=Precision, Recall=Recall)
#     plt.plot(x_axis, y_axis, '-bo', color="purple", label="Query%d"%(j+1))
#     plt.xlim(0, 1)
#     plt.ylim(0, 1)
#     plt.xlabel('Recall')
#     plt.ylabel('Precision')
#     plt.title('Standard Recall/Precision Curves')
#     plt.legend()
#     plt.show()

#     j += 1

#     # compute average Recall, average Precision, average F1-measure
#     AllRecall = np.array(AllRecall)
#     AllPrecision = np.array(AllPrecision)
#     AllF1measure = np.array(AllF1measure)
#     AveRecall = (AllRecall[0] + AllRecall[1] + AllRecall[2] + AllRecall[3] +
 ↪AllRecall[4] + AllRecall[5] + AllRecall[6] + AllRecall[7])/8
#     AvePrecision = (AllPrecision[0] + AllPrecision[1]+AllPrecision[2] +
 ↪AllPrecision[3]+AllPrecision[4] + AllPrecision[5] + AllPrecision[6] +
 ↪AllPrecision[7])/8
#     AveF1measure = (AllF1measure[0] + AllF1measure[1]+AllF1measure[2] +
 ↪AllF1measure[3]+AllF1measure[4] + AllF1measure[5] + AllF1measure[6] +
 ↪AllF1measure[7])/8

#     print("\nAverage Recall, average Precision, average F1-measure: ")
#     print("average Recall@1~10: ", np.around(AveRecall[:10],2))
#     print("average Precision@1~10: ", np.around(AvePrecision[:10],2))
#     print("average F1measure@1~10: ", np.around(AveF1measure[:10],2))

#     # plot average R/P curve
#     x_axis = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
#     y_axis = compute_RP_yaxis(Precision=AvePrecision, Recall=AveRecall)
#     plt.plot(x_axis, y_axis, '-bo', color="blue", label="Average")
#     plt.xlim(0, 1)
#     plt.ylim(0, 1)
#     plt.xlabel('average Recall')
#     plt.ylabel('average Precision')
#     plt.title('Standard Average Recall/Precision Curves')
#     plt.legend()
#     plt.show()
```

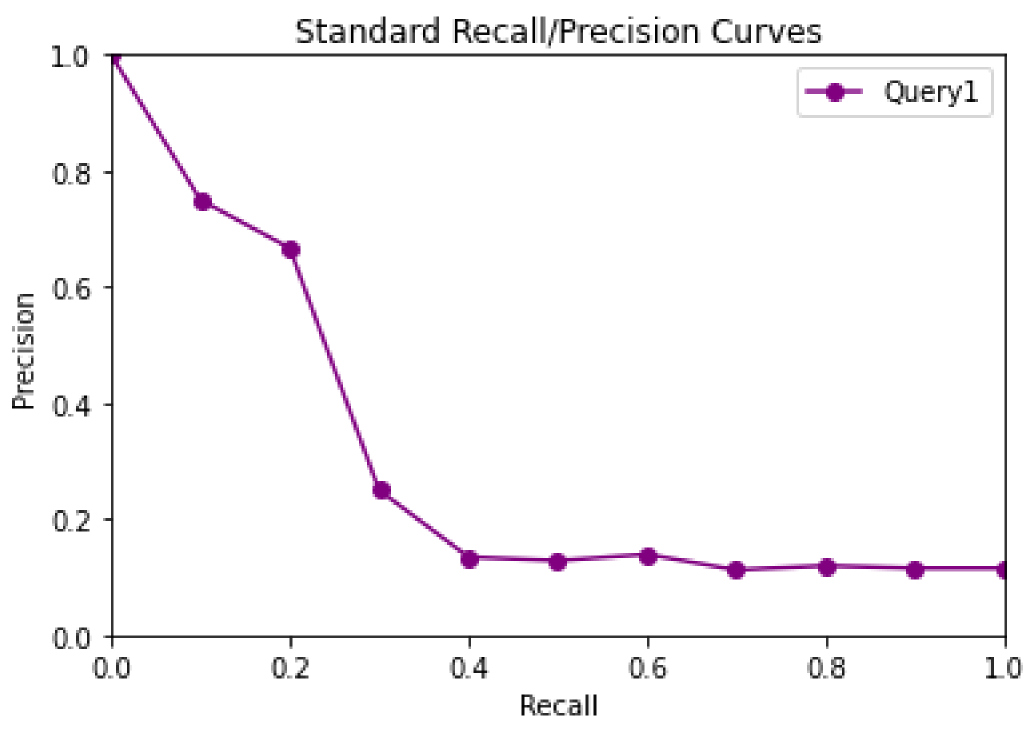
3.3 3a - Approach 4

[21]:
```
Top_n_rankings = 10
BERT_AllRecall = list()
BERT_AllPrecision = list()
BERT_AllF1measure = list()
_y_axis = list()
_avg_y_axis = list()
j = 0

for key,value in enumerate(querys):
  Q_features = query_features[key,:]
  Q_features = Q_features.reshape(1,-1)
  BERT_similarity_scores = cosine_similarity(train_features,Q_features)
  # ###
  BERT_idx = np.argsort(BERT_similarity_scores.flat)[::-1]
  doc_id = [data.iloc[BERT_idx[i]]["ID"] for i in range(len(BERT_idx))]
  # #####
  BERT_doc_idx = []
  for i in range(len(BERT_idx)):
    BERT_doc_idx.append(doc_id[i])
  # ###
  BERT_re_mark = []
  for i in range(len(BERT_idx)):
    if (BERT_doc_idx[i]) in re_ID[j]:
      BERT_re_mark.append(1)
    else:
      BERT_re_mark.append(0)

  # compute Recall, Precision, F1-measure
  BERT_Recall, BERT_Precision, BERT_F1measure =␣
↪compute_R_P_F1(re_mark=BERT_re_mark, QuRe_ID=re_ID[j])

  print('\n' + 'Query%d: '%(j+1) + query)
  for x in range(Top_n_rankings):
    print("Top " + str(x+1) + ' result: ID%d '%(BERT_doc_idx[x]),␣
↪reviews[BERT_idx[x]])
  BERT_Recall = np.array(BERT_Recall)
  BERT_Precision = np.array(BERT_Precision)
  BERT_F1measure = np.array(BERT_F1measure)
  # print(re_mark)
  print("Recall@1~10: ", np.around(BERT_Recall[:10],2))
  print("Precision@1~10: ", np.around(BERT_Precision[:10],2))
  print("F1measure@1~10: ", np.around(BERT_F1measure[:10],2))

  # save
  BERT_AllRecall.append(BERT_Recall)
```

```python
    BERT_AllPrecision.append(BERT_Precision)
    BERT_AllF1measure.append(BERT_F1measure)

    # plot R/P curve
    x_axis = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
    y_axis = compute_RP_yaxis(Precision=BERT_Precision, Recall=BERT_Recall)
    _y_axis.append(y_axis)
    plt.plot(x_axis, y_axis, '-bo', color="purple", label="Query%d"%(j+1))
    plt.xlim(0, 1)
    plt.ylim(0, 1)
    plt.xlabel('Recall')
    plt.ylabel('Precision')
    plt.title('Standard Recall/Precision Curves')
    plt.legend()
    plt.show()

    j += 1

# compute average Recall, average Precision, average F1-measure
BERT_AllRecall = np.array(BERT_AllRecall)
BERT_AllPrecision = np.array(BERT_AllPrecision)
BERT_AllF1measure = np.array(BERT_AllF1measure)
#calculate the average metrices for 8 queries
BERT_AveRecall = (BERT_AllRecall[0] + BERT_AllRecall[1] + BERT_AllRecall[2] +␣
 ↪BERT_AllRecall[3] + BERT_AllRecall[4] + BERT_AllRecall[5] +␣
 ↪BERT_AllRecall[6] + BERT_AllRecall[7])/8
BERT_AvePrecision = (BERT_AllPrecision[0] + BERT_AllPrecision[1] +␣
 ↪BERT_AllPrecision[2] + BERT_AllPrecision[3] + BERT_AllPrecision[4] +␣
 ↪BERT_AllPrecision[5] + BERT_AllPrecision[6] + BERT_AllPrecision[7])/8
BERT_AveF1measure = (BERT_AllF1measure[0] + BERT_AllF1measure[1] +␣
 ↪BERT_AllF1measure[2] + BERT_AllF1measure[3] + BERT_AllF1measure[4] +␣
 ↪BERT_AllF1measure[5] + BERT_AllF1measure[6] + BERT_AllF1measure[7])/8

print("\nAverage Recall, average Precision, average F1-measure: ")
print("average Recall@1~10: ", np.around(BERT_AveRecall[:10],2))
print("average Precision@1~10: ", np.around(BERT_AvePrecision[:10],2))
print("average F1measure@1~10: ", np.around(BERT_AveF1measure[:10],2))

# plot average R/P curve
x_axis = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
y_axis = compute_RP_yaxis(Precision=BERT_AvePrecision, Recall=BERT_AveRecall)
plt.plot(x_axis, y_axis, '-bo', color="blue", label="Average")
plt.xlim(0, 1)
plt.ylim(0, 1)
plt.xlabel('average Recall')
plt.ylabel('average Precision')
plt.title('Standard Average Recall/Precision Curves')
```

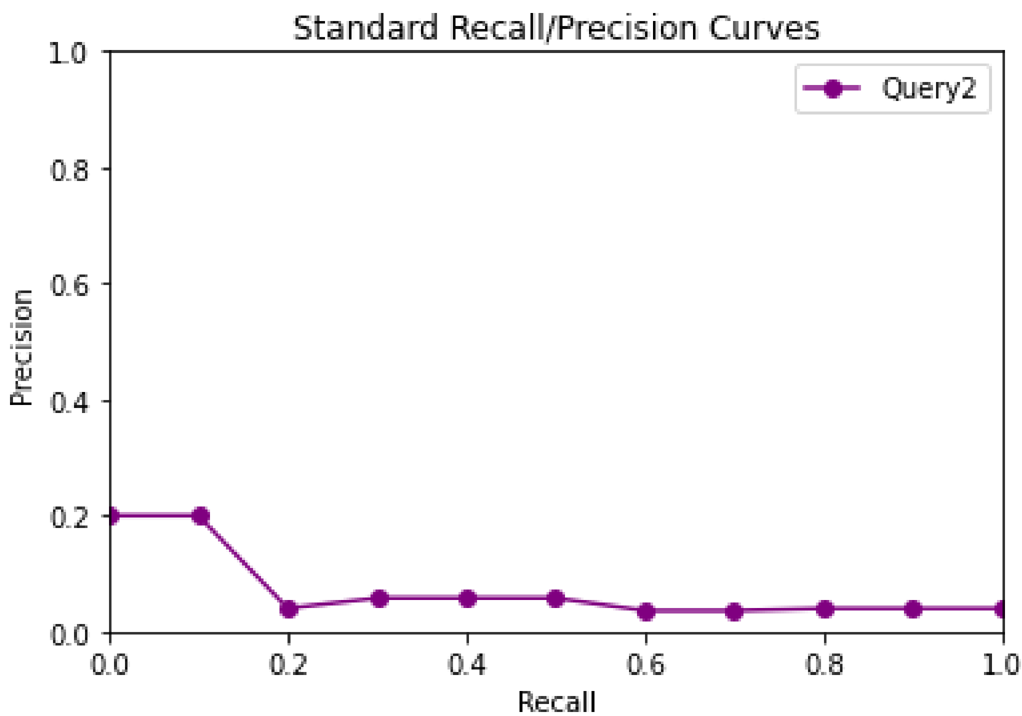
```
plt.legend()
plt.show()

BERT_y_axis_avg = y_axis
```

```
Query1: braclet looked just like its picture and is nice quality sterling
silver.
Top 1 result: ID41876  bought friends birthday loved gift ring
Top 2 result: ID58595  absolutely got beautiful engagement love durable ring
Top 3 result: ID48779  got rings thing ring love gift
Top 4 result: ID58481  cheap great loves high gift quality wife ring
Top 5 result: ID3865  dainty pretty looking sparkle
Top 6 result: ID26246  happy ring received loves gift birthday
Top 7 result: ID6522  love suggest jewelry collection every ring
Top 8 result: ID48216  got birthday imagine love adoring ring woman
Top 9 result: ID45203  got birthday imagine love adoring ring woman
Top 10 result: ID25299  bracelet true perfect nice quality necklace
Recall@1~10:  [0.06 0.06 0.12 0.19 0.19 0.25 0.25 0.25 0.25 0.25]
Precision@1~10:  [1.   0.5  0.67 0.75 0.6  0.67 0.57 0.5  0.44 0.4 ]
F1measure@1~10:  [0.12 0.11 0.21 0.3  0.29 0.36 0.35 0.33 0.32 0.31]
```



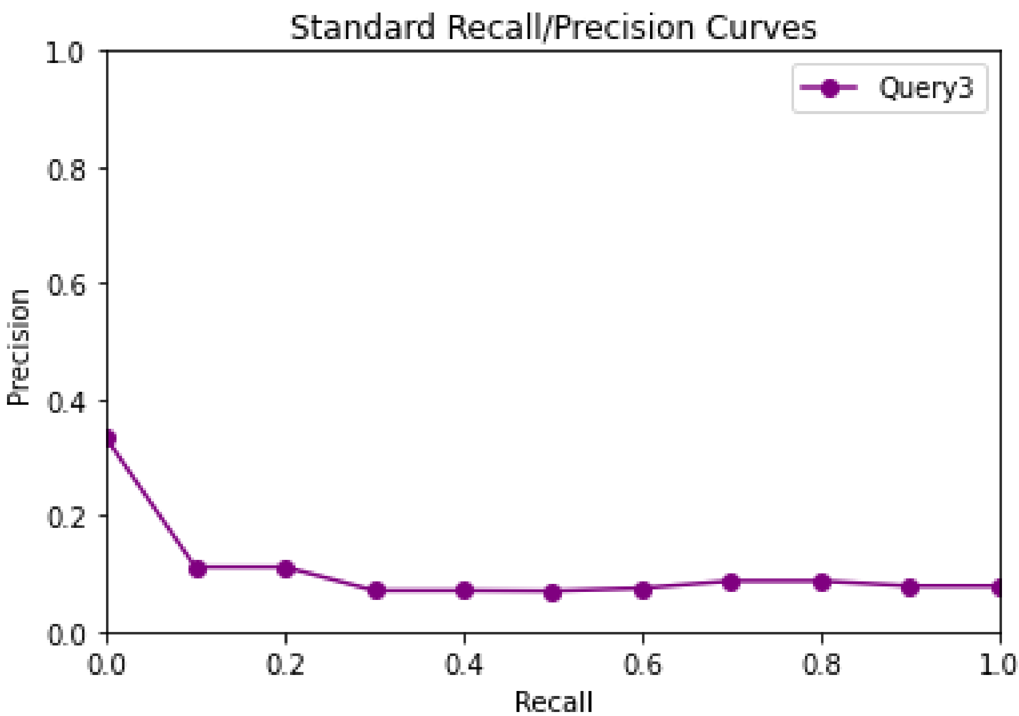

```
Query2: braclet looked just like its picture and is nice quality sterling
```

```
silver.
Top 1 result: ID9050  enough metal pretty easily ring
Top 2 result: ID41876  bought friends birthday loved gift ring
Top 3 result: ID19944  pleased use purchase shipping
Top 4 result: ID2134  wearing
Top 5 result: ID25299  bracelet true perfect nice quality necklace
Top 6 result: ID3865  dainty pretty looking sparkle
Top 7 result: ID58595  absolutely got beautiful engagement love durable ring
Top 8 result: ID22058  promptly item happy great came quality recommend
Top 9 result: ID58481  cheap great loves high gift quality wife ring
Top 10 result: ID48779  got rings thing ring love gift
Recall@1~10:  [0.   0.   0.   0.   0.17 0.17 0.17 0.17 0.17 0.17]
Precision@1~10:  [0.   0.   0.   0.   0.2  0.17 0.14 0.12 0.11 0.1 ]
F1measure@1~10:  [0.   0.   0.   0.   0.18 0.17 0.15 0.14 0.13 0.12]
```



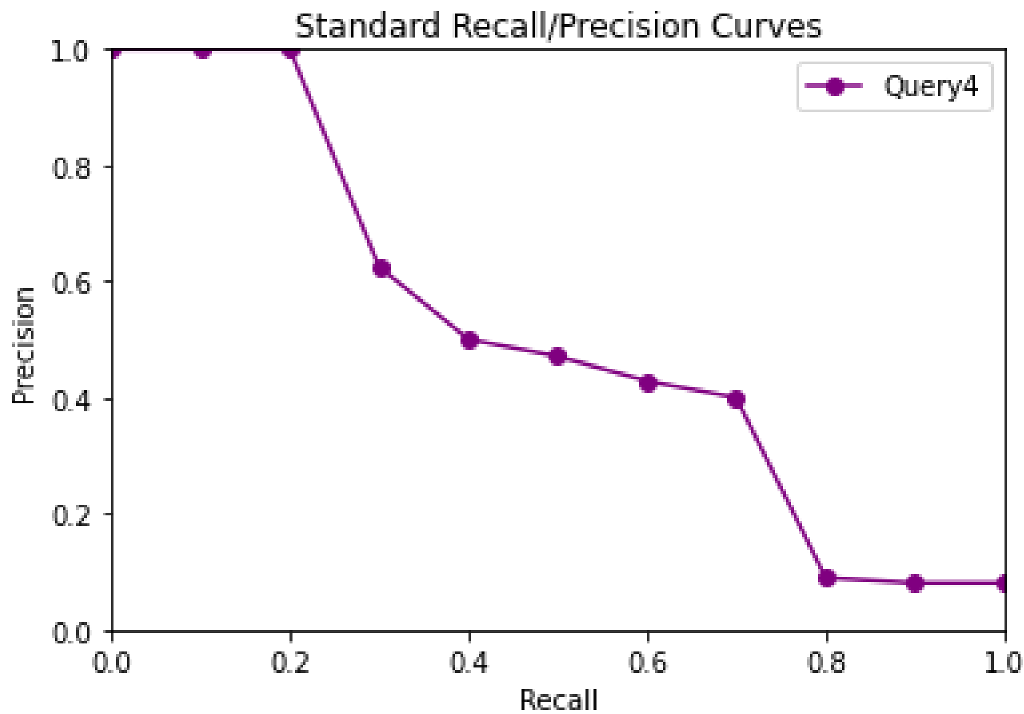

```
Query3: braclet looked just like its picture and is nice quality sterling
silver.
Top 1 result: ID22058  promptly item happy great came quality recommend
Top 2 result: ID37896  around ring comfortable wish great way
Top 3 result: ID19944  pleased use purchase shipping
Top 4 result: ID3865  dainty pretty looking sparkle
Top 5 result: ID42026  like small rings clearly fast ring product amazon guess
came
```

```
Top 6 result: ID17607  expecting seemed returned clearly product quality item
description poor
Top 7 result: ID30926  looks manner recommend timely ring would garnet
Top 8 result: ID11247  wanted love price always
Top 9 result: ID209  nothing like looks small diamonds picture ring sending back
Top 10 result: ID47345  nothing like looked diamonds picture bit product little
nt
Recall@1~10:  [0.   0.   0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.07]
Precision@1~10:  [0.   0.   0.33 0.25 0.2  0.17 0.14 0.12 0.11 0.1 ]
F1measure@1~10:  [0.   0.   0.12 0.11 0.11 0.1  0.1  0.09 0.09 0.08]
```

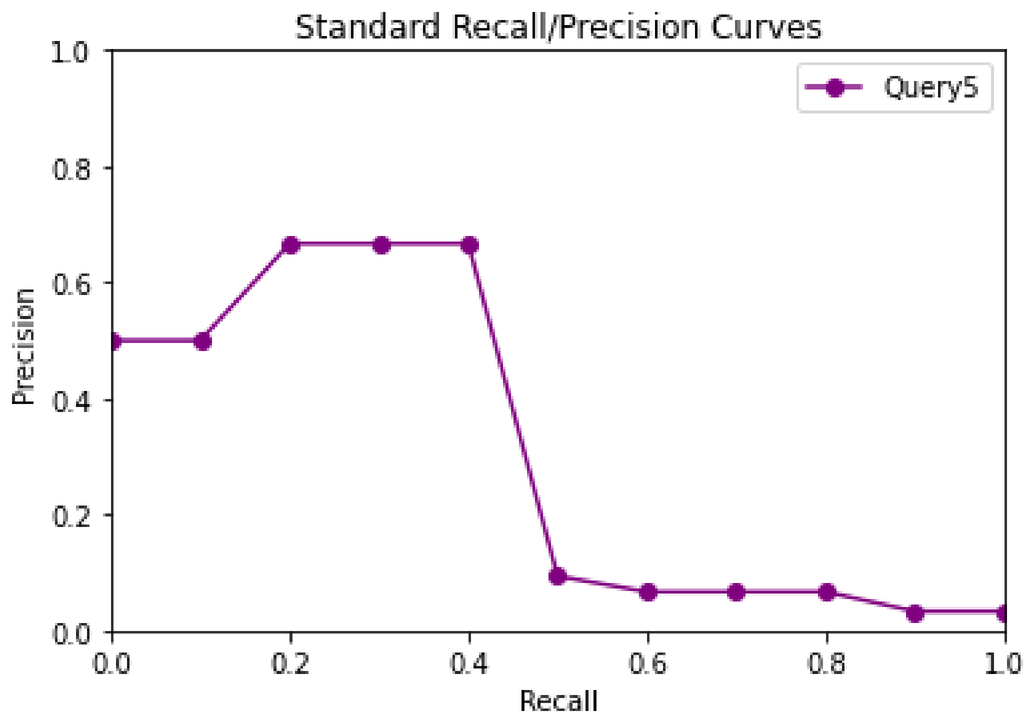



```
Query4: braclet looked just like its picture and is nice quality sterling
silver.
Top 1 result: ID53660  something nice casual want comfortable wear
Top 2 result: ID19852  good wear everyday
Top 3 result: ID30640  enough detail nice wear colors suit perfect
Top 4 result: ID2134  wearing
Top 5 result: ID44135  index great fits comfortable awesome finger ring wear
Top 6 result: ID50640  index great fits comfortable awesome finger ring wear
Top 7 result: ID11087  favorite wear mine wanted
Top 8 result: ID37486  work wear pendant going casual nt
Top 9 result: ID3865  dainty pretty looking sparkle
Top 10 result: ID52663  please ring wear comfortable quite see
```

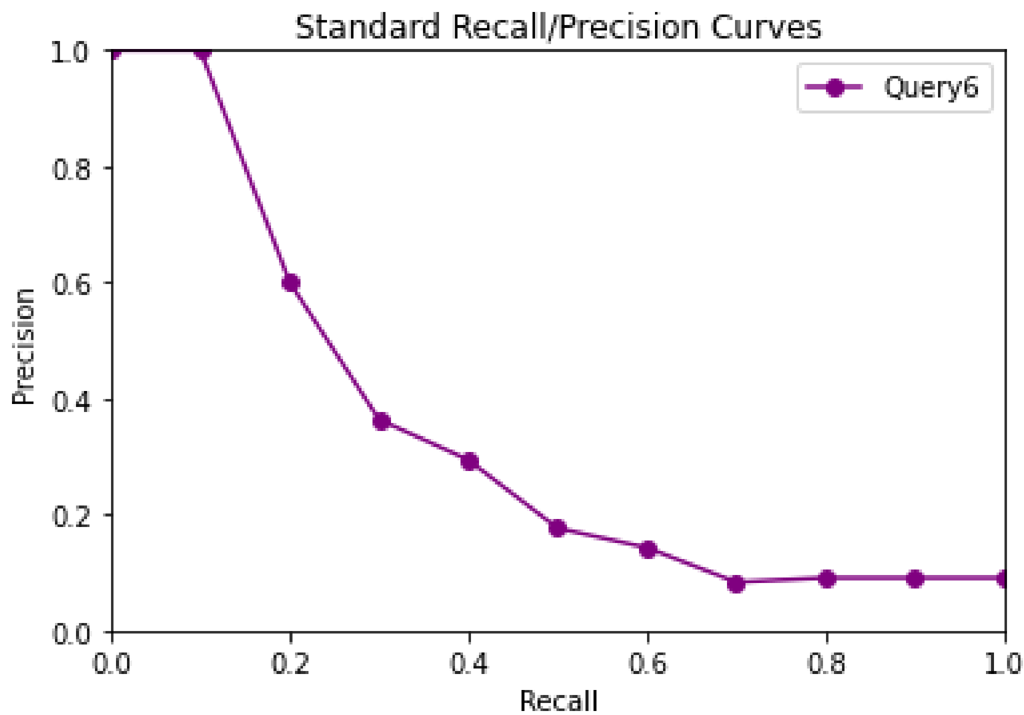
```
Recall@1~10:  [0.07 0.14 0.21 0.29 0.29 0.29 0.29 0.36 0.36 0.36]
Precision@1~10:  [1.   1.   1.   1.   0.8  0.67 0.57 0.62 0.56 0.5 ]
F1measure@1~10:  [0.13 0.25 0.35 0.44 0.42 0.4  0.38 0.45 0.43 0.42]
```



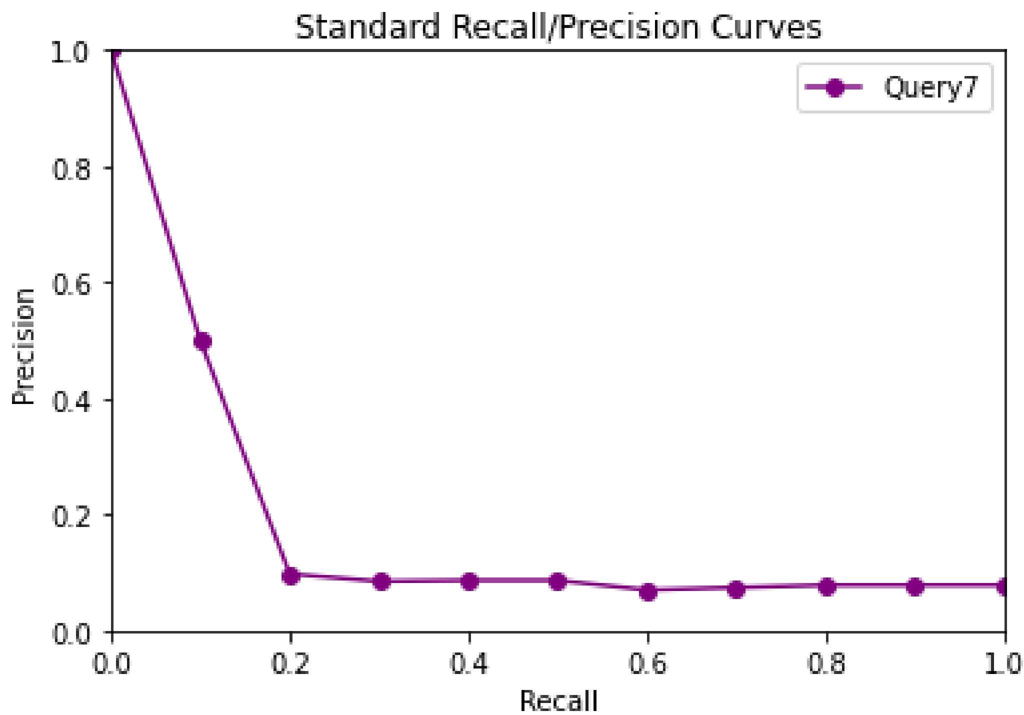

```
Query5: braclet looked just like its picture and is nice quality sterling
silver.
Top 1 result: ID1816  would look item quality recommend
Top 2 result: ID13373  pictured quality seller item poor
Top 3 result: ID17607  expecting seemed returned clearly product quality item
description poor
Top 4 result: ID45548  item attractive high quality small
Top 5 result: ID17944  expected nice product order item came
Top 6 result: ID22058  promptly item happy great came quality recommend
Top 7 result: ID19852  good wear everyday
Top 8 result: ID19944  pleased use purchase shipping
Top 9 result: ID22946  would better pinky little finger small ring
Top 10 result: ID42026  like small rings clearly fast ring product amazon guess
came
Recall@1~10:  [0.  0.2 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4]
Precision@1~10:  [0.   0.5  0.67 0.5  0.4  0.33 0.29 0.25 0.22 0.2 ]
F1measure@1~10:  [0.   0.29 0.5  0.44 0.4  0.36 0.33 0.31 0.29 0.27]
```



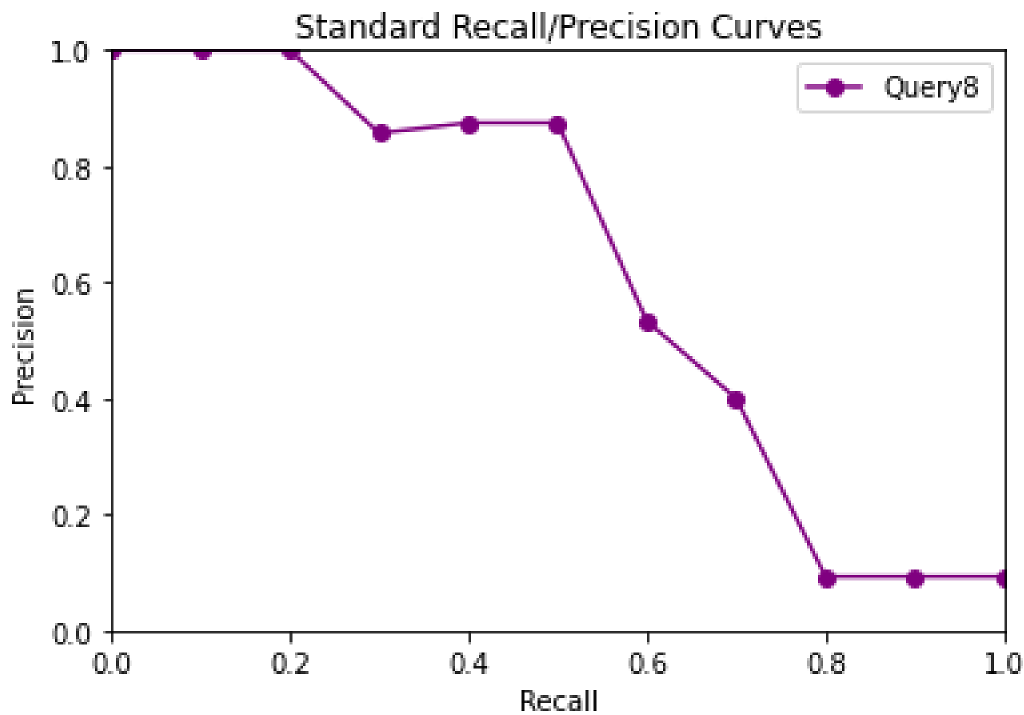

```
Query6: braclet looked just like its picture and is nice quality sterling
silver.
Top 1 result: ID52375  bought looks present nice pretty color
Top 2 result: ID39932  absolutely looks heart price amazing pendant dainty
colors beautiful
Top 3 result: ID44490  like look described small good light nice picture quality
size
Top 4 result: ID37794  like look price earrings polished good light picture
quality comfortable silver stones
Top 5 result: ID50197  look smooth beautiful
Top 6 result: ID735  like looks really nice picture charm enamel nicely size
perfect either side solid silver
Top 7 result: ID36165  like look person small diamonds smaller alot ring
pictures size little make disappointed
Top 8 result: ID42077  buy pink expecting saw picture pretty first thought ring
still color barely design disappointed
Top 9 result: ID642  like silver picture nice quality sterling looked
Top 10 result: ID943  like look looks online real picture looking ring life man
Recall@1~10:  [0.08 0.17 0.17 0.17 0.25 0.25 0.25 0.25 0.25 0.25]
Precision@1~10:  [1.   1.   0.67 0.5  0.6  0.5  0.43 0.38 0.33 0.3 ]
F1measure@1~10:  [0.15 0.29 0.27 0.25 0.35 0.33 0.32 0.3  0.29 0.27]
```



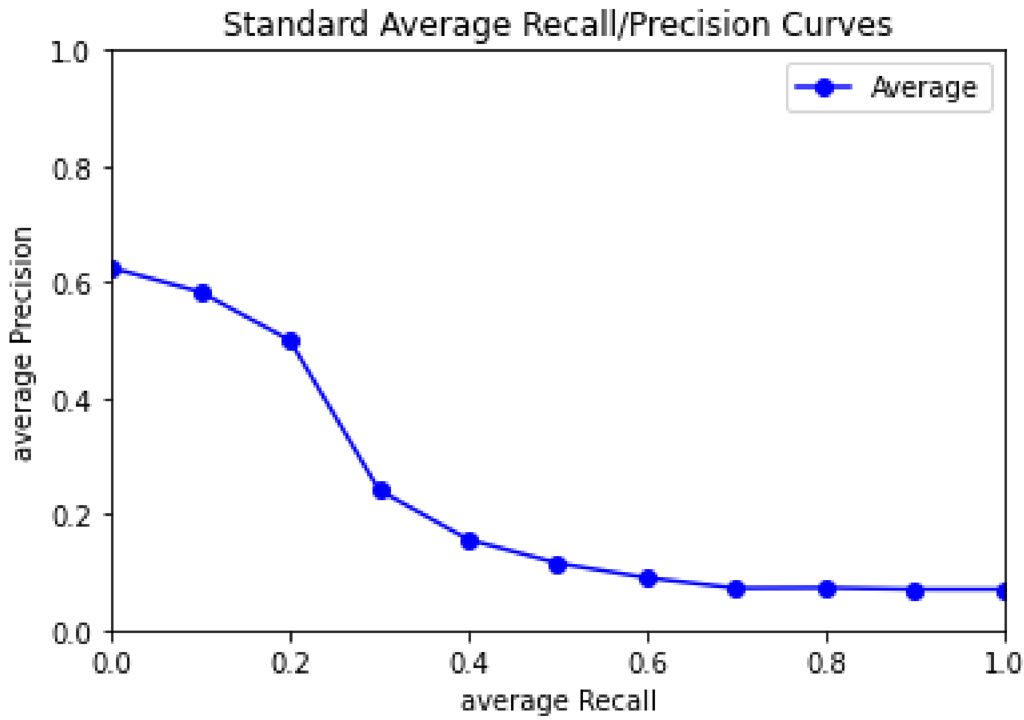

```
Query7: braclet looked just like its picture and is nice quality sterling
silver.
Top 1 result: ID209  nothing like looks small diamonds picture ring sending back
Top 2 result: ID3865  dainty pretty looking sparkle
Top 3 result: ID3494  like looks nice picture
Top 4 result: ID47345  nothing like looked diamonds picture bit product little
nt
Top 5 result: ID10642  clear like looked purple picture
Top 6 result: ID9050  enough metal pretty easily ring
Top 7 result: ID41876  bought friends birthday loved gift ring
Top 8 result: ID48779  got rings thing ring love gift
Top 9 result: ID58595  absolutely got beautiful engagement love durable ring
Top 10 result: ID50197  look smooth beautiful
Recall@1~10:  [0.07 0.07 0.07 0.14 0.14 0.14 0.14 0.14 0.14 0.14]
Precision@1~10:  [1.   0.5  0.33 0.5  0.4  0.33 0.29 0.25 0.22 0.2 ]
F1measure@1~10:  [0.13 0.12 0.12 0.22 0.21 0.2  0.19 0.18 0.17 0.17]
```




```
Query8: braclet looked just like its picture and is nice quality sterling
silver.
Top 1 result: ID642  like silver picture nice quality sterling looked
Top 2 result: ID45518  like expected seemed looked necklace much smaller picture
quality silver
Top 3 result: ID37794  like look price earrings polished good light picture
quality comfortable silver stones
Top 4 result: ID57123  worth bracelet appearance close definitely quality even
disappointed
Top 5 result: ID44490  like look described small good light nice picture quality
size
Top 6 result: ID735  like looks really nice picture charm enamel nicely size
perfect either side solid silver
Top 7 result: ID44489  like look got nice picture far christmas gift
Top 8 result: ID10642  clear like looked purple picture
Top 9 result: ID12358  like look seen price high know quite beautiful
Top 10 result: ID52375  bought looks present nice pretty color
Recall@1~10:  [0.08 0.15 0.23 0.23 0.31 0.38 0.46 0.54 0.54 0.54]
Precision@1~10:  [1.   1.   1.   0.75 0.8  0.83 0.86 0.88 0.78 0.7 ]
F1measure@1~10:  [0.14 0.27 0.38 0.35 0.44 0.53 0.6  0.67 0.64 0.61]
```




```
Average Recall, average Precision, average F1-measure:
average Recall@1~10:  [0.05 0.1  0.16 0.19 0.23 0.24 0.25 0.27 0.27 0.27]
average Precision@1~10:  [0.62 0.56 0.58 0.53 0.5  0.46 0.41 0.39 0.35 0.31]
average F1measure@1~10:  [0.09 0.17 0.24 0.27 0.3  0.31 0.3  0.31 0.29 0.28]
```



3a Comparisim

[22]:
```
# for key, value in enumerate(querys):
  # plot R/P curves for models being compared (LSI and Neural Information
 ↪Retrival (NIR))
  # print('\n' + 'Query%d: '%(index+1) + query)
  # x_axis = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
  # plot_LSI = _y_axis_lsi_tfidf[index]
  # plot_NIR = _y_axis[index]
  # plt.plot(x_axis, plot_LSI, '-bo', color="green", label="Query%d with
 ↪LSI"%(index+1))
  # plt.plot(x_axis, plot_NIR, '-bo', color="red", label="Query%d with
 ↪NRI"%(index+1))
  # plt.xlim(0, 1)
  # plt.ylim(0, 1)
  # plt.xlabel('Recall')
  # plt.ylabel('Precision')
  # plt.title('Standard Recall/Precision - Compare LSI and Neural Approach')
  # plt.legend()
  # plt.show()

# plot R/P average curves for models being compared (LSI and Neural Information
 ↪Retrival (NIR))
```

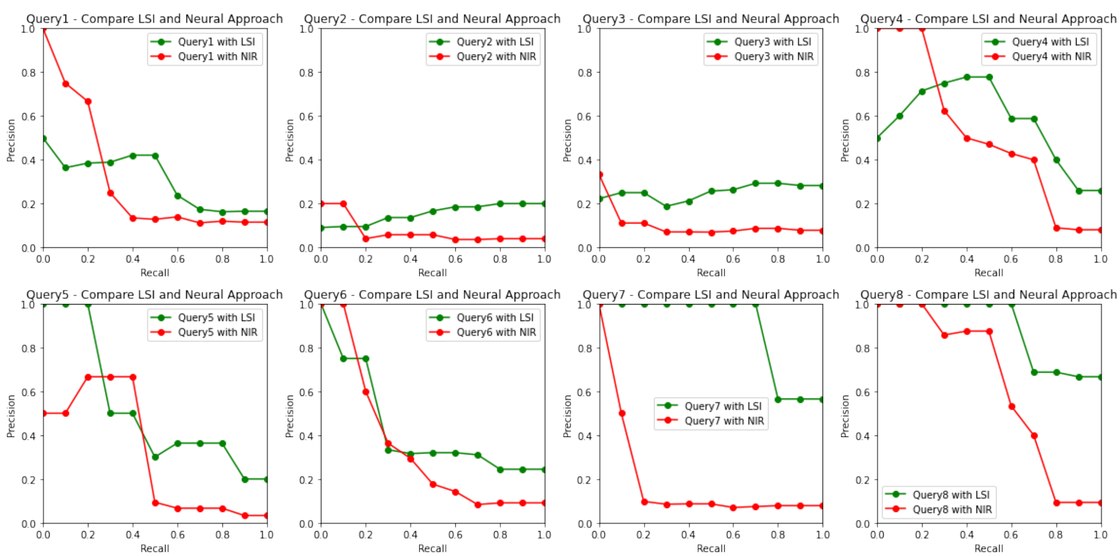
```
# x_axis = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
# plot_LSI_avg = LSI_y_axis_avg
# plot_NIR_avg = BERT_y_axis_avg
# plt.plot(x_axis, plot_LSI_avg, '-bo', color="orange", label="Average with␣
↪LSI")
# plt.plot(x_axis, plot_NIR_avg, '-bo', color="blue", label="Average with NRI")
# plt.xlim(0, 1.2)
# plt.ylim(0, 1.2)
# plt.xlabel('average Recall')
# plt.ylabel('average Precision')
# plt.title('Standard Average Recall/Precision Curves for both LSI and NIR')
# plt.legend()
# plt.show()
```

[23]:

```
from prettytable import PrettyTable

fig, axs = plt.subplots(nrows=2, ncols=4, figsize=(16, 8))

table = PrettyTable()
table.field_names = ["Query", "LSI Precision", "LSI Recall", "NIR Precision",␣
↪"NIR Recall"]

for key, (query, index) in enumerate(zip(querys, range(len(querys)))):
    # plot R/P curves for both methods(NIR and LSI)
    print('\n' + 'Query%d: '%(index+1) + query)
    x_axis = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
    plot_LSI = _y_axis_lsi_tfidf[index]
    plot_NIR = _y_axis[index]
    row = key // 4
    col = key % 4
    ax = axs[row, col]
    ax.plot(x_axis, plot_LSI, '-bo', color="green", label="Query%d with␣
↪LSI"%(index+1))
    ax.plot(x_axis, plot_NIR, '-bo', color="red", label="Query%d with␣
↪NIR"%(index+1))
    ax.set_xlim(0, 1)
    ax.set_ylim(0, 1)
    ax.set_xlabel('Recall')
    ax.set_ylabel('Precision')
    ax.set_title('Query%d - Compare LSI and Neural Approach'%(index+1))
    ax.legend()

    # calculate R/P values
    LSI_precision = _y_axis_lsi_tfidf[index][5]
    LSI_recall = x_axis[5]
    NIR_precision = _y_axis[index][5]
    NIR_recall = x_axis[5]
```

```
    # add R/P values to the table
    table.add_row([query, round(LSI_precision, 2), round(LSI_recall, 2),␣
↪round(NIR_precision, 2), round(NIR_recall, 2)])

plt.tight_layout()
plt.show()

print(table)
```

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

Query2: horrible bad quality bracelet

Query3: arrived promptly and happy with the seller

Query4: wear it with casual wear

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

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

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

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



```
+---------------------------------------------------------------------------------
-----------+---------------+------------+--------------+------------+
|                                          Query
| LSI Precision | LSI Recall | NIR Precision | NIR Recall |
+---------------------------------------------------------------------------------
-----------+---------------+------------+--------------+------------+
|                       The ring is a great gift. My friend loves it
|       0.42     |     0.5      |       0.13      |     0.5      |
|                              horrible bad quality bracelet
|       0.17     |     0.5      |       0.06      |     0.5      |
|                        arrived promptly and happy with the seller
|       0.26     |     0.5      |       0.07      |     0.5      |
|                                 wear it with casual wear
|       0.78     |     0.5      |       0.47      |     0.5      |
|                    i expected better quality. i will return this item
|       0.3      |     0.5      |       0.09      |     0.5      |
|             looks beautiful. The design is pretty. pefect and color is light
|       0.32     |     0.5      |       0.18      |     0.5      |
| This ring looks nothing like the picture. the diamonds are small and not very
noticeable |       1.0       |     0.5      |       0.09      |     0.5      |
|        braclet looked just like its picture and is nice quality sterling
silver.          |       1.0       |     0.5      |       0.88      |     0.5      |
+---------------------------------------------------------------------------------
-----------+---------------+------------+--------------+------------+
```

3.4 3b - An Interactive interface for users to type in their own query

[24]:
```
def build_query_embedding(query, n):
  query_doc = pd.Series(query)
  with torch.no_grad():
      # Tokenization
      tokenized = query_doc[0:n].apply((lambda x: tokenizer.encode(x,␣
 ↪add_special_tokens=True)))

      # padding
      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')

          # print(len(i))
```

```
        if len(i) > max_len:
            max_len = len(i)
        q += 1

    padded = np.array([i + [0]*(max_len-len(i)) for i in tokenized.values])
    np.array(padded).shape
    # masking
    attention_mask = np.where(padded != 0, 1, 0)
    attention_mask.shape
    # run the model
    input_ids = torch.tensor(padded)
    attention_mask = torch.tensor(attention_mask)

    last_hidden_states = model(input_ids, attention_mask=attention_mask)

    query_features = last_hidden_states[0][:,0,:].numpy()
    return query_features
```

[25]:

```
from prettytable import PrettyTable, ALL

def search(query, n_results=5):
  # Get the query embedding using the BERT-based model
  embedding = build_query_embedding(query, n_results)

  # Calculate the cosine similarity between the query embedding and the
 ↪document embeddings
  similarity = cosine_similarity(train_features, embedding)

  # Get the indexes of the top n_results most similar documents sorted by
 ↪similarity score
  indexes = np.argsort(similarity, axis=None)[::-1][:n_results]

  # Get the document IDs, texts, and similarity scores of the top n_results
 ↪most similar documents
  d_id = [i for i in indexes]
  ndoc_id = [data.iloc[k]['ID'] for k in indexes]
  ndoc_text = [data.iloc[k]['Reviews'] for k in indexes]
  similarity_scores = [np.around(similarity[k], 4) for k in indexes]

  # Return the query IDs, document IDs, document texts, and similarity scores
  return d_id, ndoc_id, ndoc_text, similarity_scores

def print_search_results(d_id, ndoc_id, ndoc_text, similarity_scores):
  # Create a PrettyTable object to format the search results
  results = PrettyTable()

  # Set the field names and formatting options for the table
```

```
  results.field_names = ["Rank", "Doc ID", "Score", "Text"]
  results.hrules = ALL
  results.vrules = ALL
  results.align["Rank"] = "c"
  results.align["Doc ID"] = "l"
  results.align["Similarity Score"] = "c"
  results.align["Text"] = "l"
  results.float_format = ".4"

  # Add the query as the first row of the table
  results.add_row(["Query", "", "", query])

  # Add the top n_results most similar documents to the table
  for i in range(n_results):
    results.add_row([i+1, ndoc_id[i], similarity_scores[i],  ndoc_text[i]])

  # Print the formatted table
  print(results)
```

[26]:
```
n_results = 10 #@param {type:"slider", min:1, max:10, step:1}
# Get the query and rank from the user using input forms
```

[27]:
```
query = input("Enter your query:")

# n_results = int(input("Enter the number of results you want to retrieve:"))

_id, ndoc_id, doc_text, similarity = search(query, n_results)
print_search_results(_id, doc_id, doc_text, similarity)
```

```
Enter your query:Pretty necklace. Perfet gift
+-------+--------+----------+-----------------------------------------------
--------------------------------------------------------------------------------
-------------+
|  Rank | Doc ID |  Score   | Text
|
+-------+--------+----------+-----------------------------------------------
--------------------------------------------------------------------------------
-------------+
| Query |        |          | Pretty necklace. Perfet gift
|
+-------+--------+----------+-----------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   1   | 642    | [0.9757] | the bracelet was not a true 9  the necklace
perfect the bracelet nice quality just not true to length
|
+-------+--------+----------+-----------------------------------------------
```

```
--------------------------------------------------------------------------------
-------------+
|   2   | 45518  | [0.9727] | i got this ring as a gift from my boyfriend and i
love it. the only thing is that if the rings are not position correctly it
pinches the skin. |
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   3   | 37794  | [0.9719] | I absolutely love this ring! I got this as my
engagement ring Feb 09 This ring is beautiful and durable.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   4   | 57123  | [0.9709] | I bought this as a gift for a friends birthday and
she loved it. It's a beautifull ring.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   5   | 44490  | [0.9709] | I got this ring for my birthday and I love it, I
cannot imagine a woman not adoring this ring.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   6   | 735    | [0.9709] | I got this ring for my birthday and I love it, I
cannot imagine a woman not adoring this ring.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   7   | 44489  | [0.9707] | my wife loves the ring, it was a great gift.
extremelly cheap and high quality.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   8   | 10642  | [0.9706] | I love the ring and suggest every girl should have
this ring in their jewelry collection.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   9   | 12358  | [0.9696] | This was a birthday gift for my 16 YO niece. She
loves the ring and was very happy to have received it.
|
+-------+--------+----------+---------------------------------------------------
```

```
--------------------------------------------------------------------------------
-------------+
|   10   | 52375  | [0.9683] | This ring looks nothing like the picture. the
diamonds are small and not very noticeable; I will be sending this back
|
+--------+--------+----------+------------------------------------------------
--------------------------------------------------------------------------------
-------------+
```

Fetch top 10 for each query in the query list

[28]:
```python
queries = ['The ring is a great gift. My friend loves it',
           'horrible bad quality bracelet',
           'arrived promptly and happy with the seller',
           'wear it with casual wear',
           'i expected better quality. i will return this item',
           'looks beautiful. The design is pretty. pefect and color is light',
           'This ring looks nothing like the picture. the diamonds are small and␣
 ↪not very noticeable',
           'braclet looked just like its picture and is nice quality sterling␣
 ↪silver.'
]

corpus_list = list()

for index, query in enumerate(queries):
  query_id, query_doc_id, query_doc_text, query_similarity = search(query,␣
 ↪n_results)
  print('\n' + f"QUERY {index+1} - {query}")
  print_search_results(_id, doc_id, doc_text, similarity)
  # Build Corpus list for use in Text sumarization. This was done here because␣
 ↪the installation of Summertime messes up my environment
  corpus_list.append(query_doc_text)
```

```
QUERY 1 - The ring is a great gift. My friend loves it
+--------+--------+----------+------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|  Rank  | Doc ID |  Score   | Text
|
+--------+--------+----------+------------------------------------------------
--------------------------------------------------------------------------------
-------------+
| Query  |        |          | The ring is a great gift. My friend loves it
|
+--------+--------+----------+------------------------------------------------
--------------------------------------------------------------------------------
```

```
-------------+
|   1   | 642    | [0.9757] | the bracelet was not a true 9  the necklace
perfect the bracelet nice quality just not true to length
|
+-------+--------+----------+----------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   2   | 45518  | [0.9727] | i got this ring as a gift from my boyfriend and i
love it. the only thing is that if the rings are not position correctly it
pinches the skin. |
+-------+--------+----------+----------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   3   | 37794  | [0.9719] | I absolutely love this ring! I got this as my
engagement ring Feb 09 This ring is beautiful and durable.
|
+-------+--------+----------+----------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   4   | 57123  | [0.9709] | I bought this as a gift for a friends birthday and
she loved it. It's a beautifull ring.
|
+-------+--------+----------+----------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   5   | 44490  | [0.9709] | I got this ring for my birthday and I love it, I
cannot imagine a woman not adoring this ring.
|
+-------+--------+----------+----------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   6   | 735    | [0.9709] | I got this ring for my birthday and I love it, I
cannot imagine a woman not adoring this ring.
|
+-------+--------+----------+----------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   7   | 44489  | [0.9707] | my wife loves the ring, it was a great gift.
extremelly cheap and high quality.
|
+-------+--------+----------+----------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   8   | 10642  | [0.9706] | I love the ring and suggest every girl should have
this ring in their jewelry collection.
|
+-------+--------+----------+----------------------------------------------------
--------------------------------------------------------------------------------
```

```
-------------+
|   9   | 12358  | [0.9696] | This was a birthday gift for my 16 YO niece. She
loves the ring and was very happy to have received it.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   10  | 52375  | [0.9683] | This ring looks nothing like the picture. the
diamonds are small and not very noticeable; I will be sending this back
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+

QUERY 2 - horrible bad quality bracelet
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|  Rank | Doc ID |  Score   | Text
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
| Query |        |          | horrible bad quality bracelet
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   1   | 642    | [0.9757] | the bracelet was not a true 9  the necklace
perfect the bracelet nice quality just not true to length
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   2   | 45518  | [0.9727] | i got this ring as a gift from my boyfriend and i
love it. the only thing is that if the rings are not position correctly it
pinches the skin. |
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   3   | 37794  | [0.9719] | I absolutely love this ring! I got this as my
engagement ring Feb 09 This ring is beautiful and durable.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   4   | 57123  | [0.9709] | I bought this as a gift for a friends birthday and
she loved it. It's a beautifull ring.
```

```
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   5   | 44490  | [0.9709] | I got this ring for my birthday and I love it, I
cannot imagine a woman not adoring this ring.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   6   | 735    | [0.9709] | I got this ring for my birthday and I love it, I
cannot imagine a woman not adoring this ring.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   7   | 44489  | [0.9707] | my wife loves the ring, it was a great gift.
extremelly cheap and high quality.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   8   | 10642  | [0.9706] | I love the ring and suggest every girl should have
this ring in their jewelry collection.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   9   | 12358  | [0.9696] | This was a birthday gift for my 16 YO niece. She
loves the ring and was very happy to have received it.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   10  | 52375  | [0.9683] | This ring looks nothing like the picture. the
diamonds are small and not very noticeable; I will be sending this back
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+

QUERY 3 - arrived promptly and happy with the seller
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|  Rank | Doc ID |  Score   | Text
|
+-------+--------+----------+---------------------------------------------------
```

```
--------------------------------------------------------------------------------
-------------+
| Query |        |          | arrived promptly and happy with the seller
|
+--------+--------+----------+--------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   1    | 642    | [0.9757] | the bracelet was not a true 9  the necklace
perfect the bracelet nice quality just not true to length
|
+--------+--------+----------+--------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   2    | 45518  | [0.9727] | i got this ring as a gift from my boyfriend and i
love it. the only thing is that if the rings are not position correctly it
pinches the skin. |
+--------+--------+----------+--------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   3    | 37794  | [0.9719] | I absolutely love this ring! I got this as my
engagement ring Feb 09 This ring is beautiful and durable.
|
+--------+--------+----------+--------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   4    | 57123  | [0.9709] | I bought this as a gift for a friends birthday and
she loved it. It's a beautifull ring.
|
+--------+--------+----------+--------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   5    | 44490  | [0.9709] | I got this ring for my birthday and I love it, I
cannot imagine a woman not adoring this ring.
|
+--------+--------+----------+--------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   6    | 735    | [0.9709] | I got this ring for my birthday and I love it, I
cannot imagine a woman not adoring this ring.
|
+--------+--------+----------+--------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   7    | 44489  | [0.9707] | my wife loves the ring, it was a great gift.
extremelly cheap and high quality.
|
+--------+--------+----------+--------------------------------------------------
--------------------------------------------------------------------------------
```

```
-------------+
|   8   | 10642  | [0.9706] | I love the ring and suggest every girl should have
this ring in their jewelry collection.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   9   | 12358  | [0.9696] | This was a birthday gift for my 16 YO niece. She
loves the ring and was very happy to have received it.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   10  | 52375  | [0.9683] | This ring looks nothing like the picture. the
diamonds are small and not very noticeable; I will be sending this back
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+

QUERY 4 - wear it with casual wear
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|  Rank | Doc ID |  Score   | Text
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
| Query |        |          | wear it with casual wear
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   1   | 642    | [0.9757] | the bracelet was not a true 9  the necklace
perfect the bracelet nice quality just not true to length
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   2   | 45518  | [0.9727] | i got this ring as a gift from my boyfriend and i
love it. the only thing is that if the rings are not position correctly it
pinches the skin. |
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   3   | 37794  | [0.9719] | I absolutely love this ring! I got this as my
engagement ring Feb 09 This ring is beautiful and durable.
```

```
|
+--------+--------+----------+--------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   4    | 57123  | [0.9709] | I bought this as a gift for a friends birthday and
she loved it. It's a beautifull ring.
|
+--------+--------+----------+--------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   5    | 44490  | [0.9709] | I got this ring for my birthday and I love it, I
cannot imagine a woman not adoring this ring.
|
+--------+--------+----------+--------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   6    | 735    | [0.9709] | I got this ring for my birthday and I love it, I
cannot imagine a woman not adoring this ring.
|
+--------+--------+----------+--------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   7    | 44489  | [0.9707] | my wife loves the ring, it was a great gift.
extremelly cheap and high quality.
|
+--------+--------+----------+--------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   8    | 10642  | [0.9706] | I love the ring and suggest every girl should have
this ring in their jewelry collection.
|
+--------+--------+----------+--------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   9    | 12358  | [0.9696] | This was a birthday gift for my 16 YO niece. She
loves the ring and was very happy to have received it.
|
+--------+--------+----------+--------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   10   | 52375  | [0.9683] | This ring looks nothing like the picture. the
diamonds are small and not very noticeable; I will be sending this back
|
+--------+--------+----------+--------------------------------------------------
--------------------------------------------------------------------------------
-------------+
```

QUERY 5 - i expected better quality. i will return this item

Rank	Doc ID	Score	Text
Query			i expected better quality. i will return this item
1	642	[0.9757]	the bracelet was not a true 9 the necklace perfect the bracelet nice quality just not true to length
2	45518	[0.9727]	i got this ring as a gift from my boyfriend and i love it. the only thing is that if the rings are not position correctly it pinches the skin.
3	37794	[0.9719]	I absolutely love this ring! I got this as my engagement ring Feb 09 This ring is beautiful and durable.
4	57123	[0.9709]	I bought this as a gift for a friends birthday and she loved it. It's a beautifull ring.
5	44490	[0.9709]	I got this ring for my birthday and I love it, I cannot imagine a woman not adoring this ring.
6	735	[0.9709]	I got this ring for my birthday and I love it, I cannot imagine a woman not adoring this ring.

```
-------------+
|   7   | 44489  | [0.9707] | my wife loves the ring, it was a great gift.
extremelly cheap and high quality.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   8   | 10642  | [0.9706] | I love the ring and suggest every girl should have
this ring in their jewelry collection.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   9   | 12358  | [0.9696] | This was a birthday gift for my 16 YO niece. She
loves the ring and was very happy to have received it.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   10  | 52375  | [0.9683] | This ring looks nothing like the picture. the
diamonds are small and not very noticeable; I will be sending this back
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+

QUERY 6 - looks beautiful. The design is pretty. pefect and color is light
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|  Rank | Doc ID |  Score   | Text
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
| Query |        |          | looks beautiful. The design is pretty. pefect and
color is light
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   1   | 642    | [0.9757] | the bracelet was not a true 9  the necklace
perfect the bracelet nice quality just not true to length
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   2   | 45518  | [0.9727] | i got this ring as a gift from my boyfriend and i
```

```
love it. the only thing is that if the rings are not position correctly it
pinches the skin. |
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   3   | 37794  | [0.9719] | I absolutely love this ring! I got this as my
engagement ring Feb 09 This ring is beautiful and durable.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   4   | 57123  | [0.9709] | I bought this as a gift for a friends birthday and
she loved it. It's a beautifull ring.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   5   | 44490  | [0.9709] | I got this ring for my birthday and I love it, I
cannot imagine a woman not adoring this ring.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   6   | 735    | [0.9709] | I got this ring for my birthday and I love it, I
cannot imagine a woman not adoring this ring.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   7   | 44489  | [0.9707] | my wife loves the ring, it was a great gift.
extremelly cheap and high quality.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   8   | 10642  | [0.9706] | I love the ring and suggest every girl should have
this ring in their jewelry collection.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   9   | 12358  | [0.9696] | This was a birthday gift for my 16 YO niece. She
loves the ring and was very happy to have received it.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   10  | 52375  | [0.9683] | This ring looks nothing like the picture. the
```

```
diamonds are small and not very noticeable; I will be sending this back
|
+-------+--------+----------+-----------------------------------------------------
--------------------------------------------------------------------------------
-------------+

QUERY 7 - This ring looks nothing like the picture. the diamonds are small and
not very noticeable
+-------+--------+----------+-----------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|  Rank | Doc ID |  Score   | Text
|
+-------+--------+----------+-----------------------------------------------------
--------------------------------------------------------------------------------
-------------+
| Query |        |          | This ring looks nothing like the picture. the
diamonds are small and not very noticeable
|
+-------+--------+----------+-----------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   1   | 642    | [0.9757] | the bracelet was not a true 9  the necklace
perfect the bracelet nice quality just not true to length
|
+-------+--------+----------+-----------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   2   | 45518  | [0.9727] | i got this ring as a gift from my boyfriend and i
love it. the only thing is that if the rings are not position correctly it
pinches the skin. |
+-------+--------+----------+-----------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   3   | 37794  | [0.9719] | I absolutely love this ring! I got this as my
engagement ring Feb 09 This ring is beautiful and durable.
|
+-------+--------+----------+-----------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   4   | 57123  | [0.9709] | I bought this as a gift for a friends birthday and
she loved it. It's a beautifull ring.
|
+-------+--------+----------+-----------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   5   | 44490  | [0.9709] | I got this ring for my birthday and I love it, I
cannot imagine a woman not adoring this ring.
```

```
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   6   | 735    | [0.9709] | I got this ring for my birthday and I love it, I
cannot imagine a woman not adoring this ring.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   7   | 44489  | [0.9707] | my wife loves the ring, it was a great gift.
extremelly cheap and high quality.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   8   | 10642  | [0.9706] | I love the ring and suggest every girl should have
this ring in their jewelry collection.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   9   | 12358  | [0.9696] | This was a birthday gift for my 16 YO niece. She
loves the ring and was very happy to have received it.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   10  | 52375  | [0.9683] | This ring looks nothing like the picture. the
diamonds are small and not very noticeable; I will be sending this back
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+

QUERY 8 - braclet looked just like its picture and is nice quality sterling
silver.
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|  Rank | Doc ID |  Score   | Text
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
| Query |        |          | braclet looked just like its picture and is nice
quality sterling silver.
|
```

```
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   1   | 642    | [0.9757] | the bracelet was not a true 9  the necklace
perfect the bracelet nice quality just not true to length
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   2   | 45518  | [0.9727] | i got this ring as a gift from my boyfriend and i
love it. the only thing is that if the rings are not position correctly it
pinches the skin. |
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   3   | 37794  | [0.9719] | I absolutely love this ring! I got this as my
engagement ring Feb 09 This ring is beautiful and durable.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   4   | 57123  | [0.9709] | I bought this as a gift for a friends birthday and
she loved it. It's a beautifull ring.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   5   | 44490  | [0.9709] | I got this ring for my birthday and I love it, I
cannot imagine a woman not adoring this ring.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   6   | 735    | [0.9709] | I got this ring for my birthday and I love it, I
cannot imagine a woman not adoring this ring.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   7   | 44489  | [0.9707] | my wife loves the ring, it was a great gift.
extremelly cheap and high quality.
|
+-------+--------+----------+---------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   8   | 10642  | [0.9706] | I love the ring and suggest every girl should have
this ring in their jewelry collection.
|
```

```
+-------+--------+----------+-------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   9   | 12358  | [0.9696] | This was a birthday gift for my 16 YO niece. She
loves the ring and was very happy to have received it.
|
+-------+--------+----------+-------------------------------------------------
--------------------------------------------------------------------------------
-------------+
|   10  | 52375  | [0.9683] | This ring looks nothing like the picture. the
diamonds are small and not very noticeable; I will be sending this back
|
+-------+--------+----------+-------------------------------------------------
--------------------------------------------------------------------------------
-------------+
```

4 4 - Topic Modelling

[29]:
```
# %%capture
# !pip install bertopic
```

4.1 a - Top n search results clustered them into topics

[30]:
```
# Process data into a list
def __process_data(queries, vocab):
        lines = list()
        # walk through all files in the folder
        for doc in queries:
                # print(len(doc))
                line = doc_to_line(doc, vocab)
                # add to list
                lines.append(line)
        return lines
```

[31]:
```
from bertopic import BERTopic



# # Fetch the desired number of results
n_results = int(input("Enter the number of results you want to retrieve:"))

# n_results = 65 #@param {type:"slider", min:50, max:200, step:5}
# Get the query and rank from the user using input forms

# Perform Search and retrieve the top search results
_id, doc_id, docs, similarity = search(query, n_results)

```

```python
# Collect the texts from the search results
cleaned_docs = __process_data(docs, vocab)

# Built Topic Cluster
topic_model = BERTopic(language="english", calculate_probabilities=True,␣
 ↪verbose=True)
topics, probs = topic_model.fit_transform(cleaned_docs)

print('\n' + '='*100)
print('Topics')
print('='*100)
topic_model.get_topic(0)  # Select the most frequent topic
```

```
Enter the number of results you want to retrieve:100

Batches:   0%|          | 0/4 [00:00<?, ?it/s]

2023-03-20 04:11:03,778 - BERTopic - Transformed documents to Embeddings
2023-03-20 04:11:12,969 - BERTopic - Reduced dimensionality
2023-03-20 04:11:12,984 - BERTopic - Clustered reduced embeddings


================================================================================
====================
Topics
================================================================================
====================
```

[31]:
```
[('ring', 0.1580517209975038),
 ('rings', 0.09133390976747399),
 ('love', 0.07828620837212055),
 ('like', 0.06904416404931639),
 ('picture', 0.06826165653614105),
 ('looks', 0.06781808971865956),
 ('small', 0.05864491647817589),
 ('great', 0.05573153857006345),
 ('gift', 0.046673331192923356),
 ('one', 0.045666954883736996)]
```

[32]:
```python
# from bertopic import BERTopic
# # # Fetch the desired number of results
# n_results = int(input("Enter the number of results you want to retrieve:"))

# # n_results = 65 #@param {type:"slider", min:50, max:200, step:5}
# # Get the query and rank from the user using input forms

# # Perform Search and retrieve the top search results
# _id, doc_id, docs, similarity = search(query, n_results)
```

```
# # Collect the texts from the search results
# cleaned_docs = __process_data(docs, vocab)

# # Built Topic Cluster
# topic_model = BERTopic(language="english", calculate_probabilities=True,␣
↪verbose=True)
# topics, probs = topic_model.fit_transform(cleaned_docs)

# print('\n' + '='*100)
# print('Topics')
# print('='*100)
# topic_model.get_topic(0)  # Select the most frequent topic
# topic_model.visualize_topics()
# topic_model.visualize_distribution(probs[200], min_probability=0.015)
# topic_model.visualize_barchart(top_n_topics=5)
```

[52]:
```
from bertopic import BERTopic
from umap import UMAP

topic_object = list()
def build_topic(docs, corpus, n_results=50):

  # Collect the texts from the search results and perform cleaning
  cleaned_docs = __process_data(docs, vocab)
  topic_model = BERTopic(language="english", calculate_probabilities=True,␣
↪verbose=True)
  topics, probs = topic_model.fit_transform(cleaned_docs)
  topic_object.append(topic_model)
  # a = topic_model.get_topic(0)
  # b = topic_model.visualize_topics()
  # c = topic_model.visualize_distribution(probs[n_results], min_probability=0.
↪0001)
  # d_topics = topic_model.visualize_hierarchy(top_n_topics=n_results)
  # e = topic_model.visualize_barchart(top_n_topics=n_results)

  # _topic = topic_model.get_topic(0) # Return the most frequent topics
  # _visualise = topic_model.visualize_topics()
  # _distribution = topic_model.visualize_distribution(probs[n_results],␣
↪min_probability=0.0001)
  # _hierachy = topic_model.visualize_hierarchy(top_n_topics=n_results)
  # _terms = topic_model.visualize_barchart(top_n_topics=n_results)

  # return _topic,_visualise, _distribution, _hierachy, _terms
  return topic_object

def build_topic2(docs, corpus, n_results=50):
```

```python
  umap_model = UMAP(n_neighbors=10,
                    n_components=7,
                    min_dist=0.0,
                    metric='cosine',
                    random_state=42)


  # Collect the texts from the search results and perform cleaning
  cleaned_docs = __process_data(docs, vocab)
  # topic_model = BERTopic(language="english", calculate_probabilities=True,␣
↪verbose=True)
  topic_model = BERTopic(umap_model=umap_model, language="english")
  topics, probs = topic_model.fit_transform(cleaned_docs)
  topic_object.append(topic_model)
  # a = topic_model.get_topic(0)
  # b = topic_model.visualize_topics()
  # c = topic_model.visualize_distribution(probs[n_results], min_probability=0.
↪0001)
  # d_topics = topic_model.visualize_hierarchy(top_n_topics=n_results)
  # e = topic_model.visualize_barchart(top_n_topics=n_results)

  # _topic = topic_model.get_topic(0) # Return the most frequent topics
  # _visualise = topic_model.visualize_topics()
  # _distribution = topic_model.visualize_distribution(probs[n_results],␣
↪min_probability=0.0001)
  # _hierachy = topic_model.visualize_hierarchy(top_n_topics=n_results)
  # _terms = topic_model.visualize_barchart(top_n_topics=n_results)

  # return _topic,_visualise, _distribution, _hierachy, _terms
  return topic_object

def build_viz(topic_model):
  a = topic_model.visualize_barchart(top_n_topics=5)
  b = topic_model.visualize_topics()
  # c = topic_model.visualize_distribution(probs[100], min_probability=0.0001)
  d = topic_model.visualize_hierarchy(top_n_topics=50)
  return a, b, d
```

[53]:
```python
# Get the query and rank from the user using input forms
# Fetch the desired number of results
# n_results = 110 #@param {type:"slider", min:50, max:200, step:10}

n_results = int(input("Enter the number of results you want to retrieve:"))

# Perform Search and retrieve the top search results
for index, query in enumerate(queries):
  _id, doc_id, docs, similarity = search(query, n_results)
```

```python
# Build topic for the n returned result
# a, b, c, d, e = build_topic(docs, vocab, n_results)
topic_mod = build_topic(docs, vocab, n_results)
print('\n' + '='*100)
print(f'Topics for Query{index+1}: {query}')
print('='*100)
for topic in topic_mod:
  a = topic.get_topic(0)
for i in a:
  print(i)
  # topic.visualize_topics()

# d_topics.get_topic(0)
```

```
Enter the number of results you want to retrieve:100

Batches:   0%|          | 0/4 [00:00<?, ?it/s]

2023-03-20 04:28:27,701 - BERTopic - Transformed documents to Embeddings
2023-03-20 04:28:30,023 - BERTopic - Reduced dimensionality
2023-03-20 04:28:30,037 - BERTopic - Clustered reduced embeddings


================================================================================
====================
Topics for Query1: The ring is a great gift. My friend loves it
================================================================================
====================
('like', 0.14433566362190403)
('quality', 0.1431988196528711)
('picture', 0.11991498508103643)
('look', 0.10939811453681614)
('item', 0.09572335021971412)
('nice', 0.09101567230288485)
('would', 0.07801343340247273)
('looked', 0.06759283814096244)
('small', 0.05987337023963492)
('looks', 0.056241834372557316)

Batches:   0%|          | 0/4 [00:00<?, ?it/s]

2023-03-20 04:28:30,562 - BERTopic - Transformed documents to Embeddings
2023-03-20 04:28:33,136 - BERTopic - Reduced dimensionality
2023-03-20 04:28:33,882 - BERTopic - Clustered reduced embeddings


================================================================================
====================
Topics for Query2: horrible bad quality bracelet
================================================================================
```

```
====================
('ring', 0.12183276766721064)
('like', 0.07357355190349747)
('love', 0.07094177580877746)
('quality', 0.06422563110494164)
('would', 0.06392440406931604)
('picture', 0.06098297988339629)
('looks', 0.051566995565807634)
('item', 0.05024508090027026)
('nice', 0.049033269567508635)
('rings', 0.04469891647111494)

Batches:   0%|          | 0/4 [00:00<?, ?it/s]

2023-03-20 04:28:34,944 - BERTopic - Transformed documents to Embeddings
2023-03-20 04:28:37,239 - BERTopic - Reduced dimensionality
2023-03-20 04:28:37,606 - BERTopic - Clustered reduced embeddings


================================================================================
====================
Topics for Query3: arrived promptly and happy with the seller
================================================================================
====================
('ring', 0.18266129198040318)
('love', 0.11802913329731222)
('rings', 0.10439393232350536)
('birthday', 0.07987559243111725)
('bought', 0.07987559243111725)
('engagement', 0.07987559243111725)
('gift', 0.07586680103852315)
('got', 0.07249746109079427)
('loves', 0.06840980965378012)
('perfect', 0.054373095818095706)

Batches:   0%|          | 0/4 [00:00<?, ?it/s]

2023-03-20 04:28:38,632 - BERTopic - Transformed documents to Embeddings
2023-03-20 04:28:40,929 - BERTopic - Reduced dimensionality
2023-03-20 04:28:41,610 - BERTopic - Clustered reduced embeddings


================================================================================
====================
Topics for Query4: wear it with casual wear
================================================================================
====================
('ring', 0.16741287085463472)
('love', 0.1357272272144948)
('got', 0.08986440093913625)
('birthday', 0.08587515932959594)
```

```
('rings', 0.08369605746348885)
('loves', 0.07291979020160717)
('engagement', 0.06870012746367675)
('gift', 0.06527246845183929)
('beautiful', 0.06239116796425616)
('gorgeous', 0.05879397303469153)

Batches:   0%|          | 0/4 [00:00<?, ?it/s]

2023-03-20 04:28:42,633 - BERTopic - Transformed documents to Embeddings
2023-03-20 04:28:44,866 - BERTopic - Reduced dimensionality
2023-03-20 04:28:45,649 - BERTopic - Clustered reduced embeddings


================================================================================
====================
Topics for Query5: i expected better quality. i will return this item
================================================================================
====================
('ring', 0.1767252109690951)
('rings', 0.11706519938460207)
('love', 0.10631575187166697)
('like', 0.09568417668450027)
('engagement', 0.06253981636196944)
('gift', 0.0597580126473667)
('got', 0.05741446621189006)
('loves', 0.052783524245708646)
('gorgeous', 0.052783524245708646)
('silver', 0.049467585350981884)

Batches:   0%|          | 0/4 [00:00<?, ?it/s]

2023-03-20 04:28:46,566 - BERTopic - Transformed documents to Embeddings
2023-03-20 04:28:49,582 - BERTopic - Reduced dimensionality
2023-03-20 04:28:50,785 - BERTopic - Clustered reduced embeddings


================================================================================
====================
Topics for Query6: looks beautiful. The design is pretty. pefect and color is
light
================================================================================
====================
('ring', 0.17447167112703932)
('love', 0.09487787034879627)
('rings', 0.08694488667774279)
('like', 0.07499733824066926)
('looks', 0.07416238048485875)
('picture', 0.05984590052564846)
('small', 0.05498404732650638)
('one', 0.05313879680946455)
```

```
('would', 0.04953829995403266)
('great', 0.044574857797625024)

Batches:   0%|          | 0/4 [00:00<?, ?it/s]

2023-03-20 04:28:51,683 - BERTopic - Transformed documents to Embeddings
2023-03-20 04:28:53,962 - BERTopic - Reduced dimensionality
2023-03-20 04:28:54,609 - BERTopic - Clustered reduced embeddings


================================================================================
====================
Topics for Query7: This ring looks nothing like the picture. the diamonds are
small and not very noticeable
================================================================================
====================
('like', 0.14613847925805665)
('picture', 0.11696714546231618)
('look', 0.09509311400165651)
('nice', 0.09190204711274937)
('looks', 0.09097746030502701)
('small', 0.0840289789573354)
('quality', 0.0827067820954791)
('ring', 0.08074483702857593)
('gold', 0.05594923597765204)
('nothing', 0.05594923597765204)

Batches:   0%|          | 0/4 [00:00<?, ?it/s]

2023-03-20 04:28:55,635 - BERTopic - Transformed documents to Embeddings
2023-03-20 04:28:57,923 - BERTopic - Reduced dimensionality
2023-03-20 04:28:58,615 - BERTopic - Clustered reduced embeddings


================================================================================
====================
Topics for Query8: braclet looked just like its picture and is nice quality
sterling silver.
================================================================================
====================
('ring', 0.16333370068318911)
('rings', 0.0919023780855288)
('love', 0.07220901135291548)
('like', 0.06947389951020426)
('picture', 0.0686865216183162)
('looks', 0.0682401940115143)
('small', 0.0590099263317745)
('one', 0.05251564462030217)
('great', 0.04906861344537485)
('gift', 0.04696382910491251)
```

[54]:
```
# viz = list()
for topic in topic_mod:
  try:
    a,b,d = build_viz(topic)
    a.show()
    b.show()
    # c.show()
    d.show()
  except ValueError:  #raised if `y` is empty.
    pass

# for i in d_topics:
#   i.visualize_topics()
# topic_model.visualize_distribution(probs[200], min_probability=0.015)
```

4.1b - Cluster topics for each Query

[ ]:
```
# # Feed number of n results

# topic_obj = list()
# all_topics = set()

# #Iterate over querys
# for key, query in enumerate(querys):
#   # Perform Search and retrieve the top search results
#   _id, doc_id, docs, similarity = search(query, n_results)

#   # Build topic for the n returned result for each query
#   topics = build_topic(docs, vocab)
#   topic_obj.append(topics)

#   print('\n' + '='*100)
#   print(f"Query {key+1} - {query} ")
#   print('='*100)
#   topics = topics.get_topic(0)
#   for i in range(len(topics)):
#     print(i)
#     all_topics.add(i[0])
```

4.1c - Visualize the topics

[ ]:
```
# for topic in topic_obj:
#   topic.visualize_topics()
```

Visualize Keywords

[ ]:
```
# from wordcloud import WordCloud
# import matplotlib.pyplot as plt
```

```
# # Convert the set to a space-separated string
# word_string = ' '.join(docs)

# # Create the WordCloud object
# wordcloud = WordCloud(width = 800, height = 800, background_color ='white',
#                       min_font_size = 10).generate(word_string)

# # Plot the WordCloud
# plt.figure(figsize = (8, 8), facecolor = None)
# plt.imshow(wordcloud)
# plt.axis("off")
# plt.tight_layout(pad = 0)

# # Show the plot
# plt.show()
```

5 Question 5 - Topic Summarization

Install SummerTime

[1]:
```
# 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% (655/655), done.
remote: Compressing objects: 100% (185/185), done.
remote: Total 4385 (delta 568), reused 470 (delta 470), pack-reused 3730
Receiving objects: 100% (4385/4385), 9.84 MiB | 21.21 MiB/s, done.
Resolving deltas: 100% (2406/2406), done.
```

[2]:
```
%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 gensim~=3.8.3
  Downloading gensim-3.8.3.tar.gz (23.4 MB)
                           23.4/23.4 MB
35.2 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Requirement already satisfied: sentencepiece~=0.1.95 in
/usr/local/lib/python3.9/dist-packages (from summertime==1.2.1) (0.1.97)
Collecting lexrank~=0.1.0
  Downloading lexrank-0.1.0-py3-none-any.whl (69 kB)
                           69.8/69.8 KB
11.9 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
77.6 MB/s eta 0:00:00
Requirement already satisfied: prettytable in
/usr/local/lib/python3.9/dist-packages (from summertime==1.2.1) (3.6.0)
Collecting flake8
  Downloading flake8-6.0.0-py2.py3-none-any.whl (57 kB)
                           57.8/57.8 KB
10.1 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
22.9 MB/s eta 0:00:00
Requirement already satisfied: beautifulsoup4 in
/usr/local/lib/python3.9/dist-packages (from summertime==1.2.1) (4.9.3)
Collecting pytextrank
  Downloading pytextrank-3.2.4-py3-none-any.whl (30 kB)
Requirement already satisfied: cython in /usr/local/lib/python3.9/dist-packages
(from summertime==1.2.1) (0.29.33)
Collecting datasets~=1.6.2
  Downloading datasets-1.6.2-py3-none-any.whl (221 kB)
                          221.8/221.8 KB
33.3 MB/s eta 0:00:00
Collecting black~=21.12b0
  Downloading black-21.12b0-py3-none-any.whl (156 kB)
                          156.7/156.7 KB
24.4 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
11.5 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 easynmt~=2.0.1
  Downloading EasyNMT-2.0.2.tar.gz (23 kB)
  Preparing metadata (setup.py) … done
Collecting readability-lxml
  Downloading readability_lxml-0.8.1-py3-none-any.whl (20 kB)
Collecting jupyter
  Downloading jupyter-1.0.0-py2.py3-none-any.whl (2.7 kB)
Collecting progressbar
  Downloading progressbar-2.5.tar.gz (10 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
92.4 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
11.5 MB/s eta 0:00:00
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
12.3 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Collecting click==7.1.2
  Downloading click-7.1.2-py2.py3-none-any.whl (82 kB)
                           82.8/82.8 KB
13.1 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
81.6 MB/s eta 0:00:00
Collecting spacy==3.0.6
  Downloading spacy-3.0.6-cp39-cp39-manylinux2014_x86_64.whl (12.6 MB)
                           12.6/12.6 MB
84.3 MB/s eta 0:00:00
Requirement already satisfied: numpy in /usr/local/lib/python3.9/dist-
packages (from summertime==1.2.1) (1.22.4)
Collecting summ-eval==0.70
  Downloading summ_eval-0.70-py3-none-any.whl (62.5 MB)
                           62.5/62.5 MB
7.3 MB/s eta 0:00:00
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) (1.1.1)
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: 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)
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: jinja2 in /usr/local/lib/python3.9/dist-packages
(from spacy==3.0.6->summertime==1.2.1) (3.1.2)
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: 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)
Collecting typer<0.4.0,>=0.3.0
  Downloading typer-0.3.2-py3-none-any.whl (21 kB)
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: setuptools in /usr/local/lib/python3.9/dist-
packages (from spacy==3.0.6->summertime==1.2.1) (63.4.3)
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
117.5 MB/s eta 0:00:00
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)
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
62.6 MB/s eta 0:00:00
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)
Collecting pyemd==0.5.1
  Downloading pyemd-0.5.1.tar.gz (91 kB)
                           91.5/91.5 KB
14.4 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
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)
Collecting bert-score
  Downloading bert_score-0.3.13-py3-none-any.whl (61 kB)
                           61.1/61.1 KB
8.9 MB/s eta 0:00:00
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 pytorch-pretrained-bert
  Downloading pytorch_pretrained_bert-0.6.2-py3-none-any.whl (123 kB)
                          123.8/123.8 KB
20.6 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)
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 sacremoses
  Downloading sacremoses-0.0.53.tar.gz (880 kB)
                          880.6/880.6 KB
70.8 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Collecting wmd
  Downloading wmd-1.3.2.tar.gz (104 kB)
                          104.6/104.6 KB
18.0 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: networkx in /usr/local/lib/python3.9/dist-
packages (from summ-eval==0.70->summertime==1.2.1) (3.0)
Collecting sacrebleu
  Downloading sacrebleu-2.3.1-py3-none-any.whl (118 kB)
                          118.9/118.9 KB
20.5 MB/s eta 0:00:00
Collecting stanza
  Downloading stanza-1.5.0-py3-none-any.whl (802 kB)
                          802.5/802.5 KB
66.8 MB/s eta 0:00:00
Requirement already satisfied: platformdirs>=2 in
```

```
/usr/local/lib/python3.9/dist-packages (from black~=21.12b0->summertime==1.2.1)
(3.1.1)
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)
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 tomli<2.0.0,>=0.2.6
  Downloading tomli-1.2.3-py3-none-any.whl (12 kB)
Collecting multiprocess
  Downloading multiprocess-0.70.14-py39-none-any.whl (132 kB)
                           132.9/132.9 KB
22.9 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 1.0 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
8.8 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)
Collecting dill
  Downloading dill-0.3.6-py3-none-any.whl (110 kB)
                           110.5/110.5 KB
17.6 MB/s eta 0:00:00
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: 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: 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)
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)
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 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
23.1 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
46.2 MB/s eta 0:00:00
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
7.7 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
45.8 MB/s eta 0:00:00
Collecting texttable
  Downloading texttable-1.6.7-py2.py3-none-any.whl (10 kB)
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
94.4 MB/s eta 0:00:00
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: 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: absl-py>=0.4 in /usr/local/lib/python3.9/dist-
packages (from tensorboard~=2.4.1->summertime==1.2.1) (1.4.0)
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
23.3 MB/s eta 0:00:00
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: wheel>=0.26 in /usr/local/lib/python3.9/dist-
```

```
packages (from tensorboard~=2.4.1->summertime==1.2.1) (0.40.0)
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: 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 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
103.8 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
7.1 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
11.8 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)
Collecting qtconsole
  Downloading qtconsole-5.4.1-py3-none-any.whl (120 kB)
                          120.9/120.9 KB
20.4 MB/s eta 0:00:00
Requirement already satisfied: ipykernel in /usr/local/lib/python3.9/dist-
packages (from jupyter->summertime==1.2.1) (5.3.4)
Requirement already satisfied: nbconvert in /usr/local/lib/python3.9/dist-
packages (from jupyter->summertime==1.2.1) (6.5.4)
Requirement already satisfied: notebook in /usr/local/lib/python3.9/dist-
packages (from jupyter->summertime==1.2.1) (6.3.0)
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
85.4 MB/s eta 0:00:00
Collecting icecream>=2.1
  Downloading icecream-2.1.3-py2.py3-none-any.whl (8.4 kB)
Collecting graphviz>=0.13
  Downloading graphviz-0.20.1-py3-none-any.whl (47 kB)
```

```
                            47.0/47.0 KB
8.3 MB/s eta 0:00:00
Requirement already satisfied: chardet in /usr/local/lib/python3.9/dist-
packages (from readability-lxml->summertime==1.2.1) (4.0.0)
Collecting cssselect
  Downloading cssselect-1.2.0-py2.py3-none-any.whl (18 kB)
Requirement already satisfied: lxml in /usr/local/lib/python3.9/dist-packages
(from readability-lxml->summertime==1.2.1) (4.9.2)
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)
Collecting cachetools<5.0,>=2.0.0
  Downloading cachetools-4.2.4-py3-none-any.whl (10 kB)
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)
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 executing>=0.3.1
  Downloading executing-1.2.0-py2.py3-none-any.whl (24 kB)
Collecting colorama>=0.3.9
  Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Collecting asttokens>=2.0.1
  Downloading asttokens-2.2.1-py2.py3-none-any.whl (26 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: 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)
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)
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: 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: 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: 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
11.5 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: entrypoints>=0.2.2 in
/usr/local/lib/python3.9/dist-packages (from
nbconvert->jupyter->summertime==1.2.1) (0.4)
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: bleach in /usr/local/lib/python3.9/dist-packages
(from nbconvert->jupyter->summertime==1.2.1) (6.0.0)
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: 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: 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: jupyterlab-pygments in
```

```
/usr/local/lib/python3.9/dist-packages (from
nbconvert->jupyter->summertime==1.2.1) (0.2.2)
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: 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: tinycss2 in /usr/local/lib/python3.9/dist-
packages (from nbconvert->jupyter->summertime==1.2.1) (1.2.1)
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: argon2-cffi in /usr/local/lib/python3.9/dist-
packages (from notebook->jupyter->summertime==1.2.1) (21.3.0)
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)
Requirement already satisfied: prometheus-client in
/usr/local/lib/python3.9/dist-packages (from
notebook->jupyter->summertime==1.2.1) (0.16.0)
Collecting boto3
  Downloading boto3-1.26.93-py3-none-any.whl (135 kB)
                          135.1/135.1 KB
20.6 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
14.0 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
27.0 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: 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: pickleshare in /usr/local/lib/python3.9/dist-
packages (from ipython>=5.0.0->ipykernel->jupyter->summertime==1.2.1) (0.7.5)
Collecting jedi>=0.10
```

```
  Downloading jedi-0.18.2-py2.py3-none-any.whl (1.6 MB)
                           1.6/1.6 MB
85.9 MB/s eta 0:00:00
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)
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: 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: 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: 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: 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: 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: 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: 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: 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 botocore<1.30.0,>=1.29.93
```

```
  Downloading botocore-1.29.93-py3-none-any.whl (10.5 MB)
                           10.5/10.5 MB
120.3 MB/s eta 0:00:00
Collecting s3transfer<0.7.0,>=0.6.0
  Downloading s3transfer-0.6.0-py3-none-any.whl (79 kB)
                           79.6/79.6 KB
9.8 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=541000
sha256=41d9009d609648622c879cc676f79aeecf92052e802499021d28d5f070c913e0
  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=3bbd29f2805318d2eca63e3620cb6b2b9adf918eb221fdcc07083f53fbc46799
  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=4395649
sha256=d976f43c4bd9a6d3445652fe322dbb1bf6b58e01b6d753b4024a059248721a42
  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=06f71219467600b6e42eddc87ebcca57a2ffda15d666bba6889ebf28d9dbd6ab
  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=26528072
sha256=19f6c19b178741e613fa000a7b7392d2796763e67641d6c11fd76fc441f9e0d9
  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=cf97f9f57a2b01fe01c59647f6c87da006c3b60329a9eada35b678a55e233799
  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=795f8056572345f67a7edff1688dbcdd1c5ccf9f48edea0a340adea692a1527a
  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=26224f12c0f102ecc9a33025170ba8c912487cedc8e178df2eab943e9821f917
  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=419729c9474ec987729cc6ee3dc22a3d00a07b1810ea23f3cd2324726641afcf
  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=1230902
sha256=238f77b5f2ba75ef60c7f70ae00f3a63b21e7a0d192fea027888bfa2ec51aa5b
  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=c3c875f89a9e124d9fd95bd4c14970f0111a6ffdd7e964fb90dad0aa42948a06
  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=c922349a7022cbbb03c2ce8469d072fe7a088233213935176ea6a42d9ea6469f
  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, 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.
sentence-transformers 2.2.2 requires huggingface-hub>=0.4.0, but you have
huggingface-hub 0.0.19 which is incompatible.
sentence-transformers 2.2.2 requires transformers<5.0.0,>=4.6.0, but you have
transformers 4.5.1 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 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
```

[3]:
```
## 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
```

```
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-f0knsiz6
  Running command git clone --filter=blob:none --quiet
https://github.com/bheinzerling/pyrouge.git /tmp/pip-req-build-f0knsiz6
  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=a189fa525d0b988fc288a012fc237bd4329780c4b845c1d5025f285b006fc9a8
  Stored in directory: /tmp/pip-ephem-wheel-cache-
na8zjyo2/wheels/bd/07/80/f241050743bda1488efce41793a0b5502c97888adf191110d3
Successfully built pyrouge
Installing collected packages: pyrouge
Successfully installed pyrouge-0.1.3
```

[4]:

```
# If you've been prompted to restart the kernel in either of the two cells␣
↪above,
# Please do so
# Then run this cell to go back to the relevant directory

%cd /content/SummerTime/

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

```
/content/SummerTime
2023-03-17 10:31:37.530741: I tensorflow/core/platform/cpu_feature_guard.cc:193]
This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical
operations:  AVX2 AVX512F AVX512_VNNI FMA
To enable them in other operations, rebuild TensorFlow with the appropriate
compiler flags.
2023-03-17 10:31:37.681080: I tensorflow/core/util/port.cc:104] oneDNN custom
operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them
off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2023-03-17 10:31:38.443168: 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/lib64-nvidia
2023-03-17 10:31:38.443269: 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/lib64-nvidia
2023-03-17 10:31:38.443288: 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
42.5 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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')
```

[5]:
```
!pip install --upgrade transformers
```

```
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: tqdm>=4.27 in /usr/local/lib/python3.9/dist-
packages (from transformers) (4.49.0)
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: 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: requests in /usr/local/lib/python3.9/dist-
packages (from transformers) (2.25.1)
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: regex!=2019.12.17 in
/usr/local/lib/python3.9/dist-packages (from transformers) (2022.6.2)
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: 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)
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
  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
```

Implement Summarization

[6]:
```python
def build_query_embedding(query, n):
  query_doc = pd.Series(query)
  with torch.no_grad():
      # Tokenization
      tokenized = query_doc[0:n].apply((lambda x: tokenizer.encode(x,␣
↪add_special_tokens=True)))

      # padding
      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')

          # print(len(i))
          if len(i) > max_len:
              max_len = len(i)
          q += 1

      padded = np.array([i + [0]*(max_len-len(i)) for i in tokenized.values])
      np.array(padded).shape
      # masking
```

```python
        attention_mask = np.where(padded != 0, 1, 0)
        attention_mask.shape
        # run the model
        input_ids = torch.tensor(padded)
        attention_mask = torch.tensor(attention_mask)

        last_hidden_states = model(input_ids, attention_mask=attention_mask)

        query_features = last_hidden_states[0][:,0,:].numpy()
        return query_features

from prettytable import PrettyTable, ALL

def search(query, n_results=5):
    # Get the query embedding using the BERT-based model
    embedding = build_query_embedding(query, n_results)

    # Calculate the cosine similarity between the query embedding and the␣
↪document embeddings
    similarity = cosine_similarity(train_features, embedding)

    # Get the indexes of the top n_results most similar documents sorted by␣
↪similarity score
    indexes = np.argsort(similarity, axis=None)[::-1][:n_results]

    # Get the document IDs, texts, and similarity scores of the top n_results␣
↪most similar documents
    d_id = [i for i in indexes]
    ndoc_id = [data.iloc[k]['ID'] for k in indexes]
    ndoc_text = [data.iloc[k]['Reviews'] for k in indexes]
    similarity_scores = [np.around(similarity[k], 4) for k in indexes]

    # Return the query IDs, document IDs, document texts, and similarity scores
    return d_id, ndoc_id, ndoc_text, similarity_scores

def print_search_results(d_id, ndoc_id, ndoc_text, similarity_scores):
    # Create a PrettyTable object to format the search results
    results = PrettyTable()

    # Set the field names and formatting options for the table
    results.field_names = ["Rank", "Doc ID", "Score", "Text"]
    results.hrules = ALL
    results.vrules = ALL
    results.align["Rank"] = "c"
    results.align["Doc ID"] = "l"
    results.align["Similarity Score"] = "c"
    results.align["Text"] = "l"
```

```
  results.float_format = ".4"

  # Add the query as the first row of the table
  results.add_row(["Query", "", "", query])

  # Add the top n_results most similar documents to the table
  for i in range(n_results):
    results.add_row([i+1, ndoc_id[i], similarity_scores[i],  ndoc_text[i]])

  # Print the formatted table
  print(results)
```

[43]:
```
from summertime import model

# corpus = corpus_list
summary_list = list()
for key, corpus in enumerate(corpus_list):
  print('\n' + f"QUERY {key+1} SUMMARIES")
  lexrank = model.LexRankModel(corpus)
  # # Inference
  summary = lexrank.summarize(corpus)
  summary_list.append(summary)
  for i in range(len(summary)):
    print("Summary Review %d: "%(i+1), summary[i])


# Get the query and rank from the user using input forms
# Fetch the desired number of results


# Add the top n_results most similar documents to the table
# for i in range(n_results):
#   results.add_row([i+1, ndoc_id[i], similarity_scores[i],  ndoc_text[i]])

# Longformer2Roberta
# longformer = model.LongformerModel()

# longformer_summary = longformer.summarize(corpus)
# for i in range(len(longformer_summary)):
#   print("\nSummary Review %d: "%(i+1), longformer_summary[i])
```

```
QUERY 1 SUMMARIES
Summary Review 1:  extremelly cheap and high quality. my wife loves the ring, it
was a great gift.
Summary Review 2:  It's a beautifull ring. I bought this as a gift for a friends
birthday and she loved it.
```

Summary Review 3: Thank you, Dorothy Eve's Addiction was wonderful with sending the ring and the ring is beautiful; my daughter waas thrilled with the ring.
Summary Review 4: It closes firmly with a clic and has a classic look. The look is very beautiful with a smooth finish.
Summary Review 5: I love the ring and suggest every girl should have this ring in their jewelry collection.
Summary Review 6: I would recommend this amethyst ring to anyone who is in the market for a reasonably priced amethyst ring. This ring is just absolutely stunning and beautiful!
Summary Review 7: I wanted to know if this ring is like 2 rings in one, because this ring is beyond gorgeous, I just love it.
Summary Review 8: I wanted to know if this ring is like 2 rings in one, because this ring is beyond gorgeous, I just love it.
Summary Review 9: She loves the ring and was very happy to have received it. This was a birthday gift for my 16 YO niece.
Summary Review 10: The ring shines perfectly I love this ring! This ring is perfect I say why spend thousands when you don't have to?

QUERY 2 SUMMARIES
Summary Review 1: :) :) I like it.
Summary Review 2: great way to support the local pro sports team without wearing an oversized jersey or a hat to mess up the hair
Summary Review 3: Very disappointed in the appearance and quality of the bracelet and its definitely not worth $45.00 - not even close.
Summary Review 4: Other than that a great very comfortable ring. my only wish on this ring is- I wish the cut potrion went all the way around the ring.
Summary Review 5: 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
Summary Review 6: the product came very fast and was just like how amazon explained itthe ring is very clearly written NO WAR and thick which i likebut i guess if you like small rings this isnt your ring
Summary Review 7: Just what I was looking for. It is so pretty and dainty.
Summary Review 8: Will definately reccommend Eve's Addiction to all my friends and family. I was very pleased with the quality of this item.
Summary Review 9: The ring is pretty enough, but the metal of the ring is very insubstantial it pushes in very easily.
Summary Review 10: I just got these yesterday as a Christmas gift- so far they look just like the picture and seem very nice.

QUERY 3 SUMMARIES
Summary Review 1: I'm very happy with it and recommend it unreservedly. Item was great quality and came promptly.
Summary Review 2: the product came very fast and was just like how amazon explained itthe ring is very clearly written NO WAR and thick which i likebut i guess if you like small rings this isnt your ring
Summary Review 3: Very disappointed in the appearance and quality of the bracelet and its definitely not worth $45.00 - not even close.
Summary Review 4: great way to support the local pro sports team without

wearing an oversized jersey or a hat to mess up the hair
Summary Review 5: Recv'd my ring in a timely manner it looks very antique would recommend this ring to any garnet lover!
Summary Review 6: It was described perfectly and was everything I had hoped The product arrived in a very short period of time and was perfect.
Summary Review 7: the product i recieved was nice it came in a timley matter faster than i expected will order this item again
Summary Review 8: Other than that a great very comfortable ring. my only wish on this ring is- I wish the cut potrion went all the way around the ring.
Summary Review 9: Just what I was looking for. It is so pretty and dainty.
Summary Review 10: It closes firmly with a clic and has a classic look. The look is very beautiful with a smooth finish.

QUERY 4 SUMMARIES
Summary Review 1: very good for everyday wear or dressing up
Summary Review 2: very dressy ery suitable for wearing for fashionable occasions.
Summary Review 3: They are very comfortable. These are nice to wear when you want something casual to wear.
Summary Review 4: I wear on the weekends or out & about but isn't not suited for my work or my going out events This pendant I classify as the best for casual wear.
Summary Review 5: The stones catch the light and the style is very comfortable to wear. It is so unique and a pleasure to wear.
Summary Review 6: 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
Summary Review 7: 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.
Summary Review 8: :) :) I like it.
Summary Review 9: great way to support the local pro sports team without wearing an oversized jersey or a hat to mess up the hair
Summary Review 10: ! This pinis nice enough to wear in formal occasions.Wear it with pride!

QUERY 5 SUMMARIES
Summary Review 1: You can see all the keys, any flute fan would adore having this item. The flute charm is so detailed and is of very high quality.
Summary Review 2: It is too small for an adult. This is an attractive and high quality item for a young teenager.
Summary Review 3: 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.
Summary Review 4: The diamond had a crack in one Garnet and another one had a large chip.
Summary Review 5: I love the ring and suggest every girl should have this ring in their jewelry collection.
Summary Review 6: great way to support the local pro sports team without wearing an oversized jersey or a hat to mess up the hair
Summary Review 7: Very flimsy.I would not recommend this item The quality and

look were not what I had anticipated.
Summary Review 8: Recv'd my ring in a timely manner it looks very antique would recommend this ring to any garnet lover!
Summary Review 9: the product i recieved was nice it came in a timley matter faster than i expected will order this item again
Summary Review 10: the product came very fast and was just like how amazon explained itthe ring is very clearly written NO WAR and thick which i likebut i guess if you like small rings this isnt your ring

QUERY 6 SUMMARIES
Summary Review 1: I bought it for my aunt as a present and the color is very nice. The message is very positive and it looks very pretty.
Summary Review 2: The silver does not look as in the picture but just like polished silver. the earrings are as in the picture, stones look good and are light and comfortable, reasonable quality for the price.
Summary Review 3: This medical alert braclet looked just like its picture and is nice quality sterling silver.
Summary Review 4: Although the picture looks like metal beads and description states sterling silver, these are pearls.
Summary Review 5: CAN BE LAYED FOR A BEAUTIFUL LOOK. YET IT IS VERY PRETTY.
Summary Review 6: A truly beautiful piece to own. The workmanship is excellent and the details are beautiful.
Summary Review 7: From the picture they looked to have some purple in them but they are clear just like the title says.
Summary Review 8: The color of the stones is rich and beautiful. The ring is exactly as pictured and looks very pretty on my hand.
Summary Review 9: I've seen them elsewhere for quite a high price and these are beautiful. They are very colorful and you know they are turtles.
Summary Review 10: 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.

QUERY 7 SUMMARIES
Summary Review 1: The diamond had a crack in one Garnet and another one had a large chip.
Summary Review 2: The ring was nice and looked like picture but had a crack in one Garnet and another one had a large chip.
Summary Review 3: the diamonds are small and not very noticeable; I will be sending this back This ring looks nothing like the picture.
Summary Review 4: Although the picture looks like metal beads and description states sterling silver, these are pearls.
Summary Review 5: 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.
Summary Review 6: I ended up returning the bracelet because I have amethyst jewelry and it was extremely poor quality. The stones on this bracelet are extremely pale, more pink than purple.
Summary Review 7: From the picture they looked to have some purple in them but

```
they are clear just like the title says.
Summary Review 8:  This medical alert braclet looked just like its picture and
is nice quality sterling silver.
Summary Review 9:  The diamonds are flawed more than a little bit. I didn't like
this product because the diamonds looked nothing like the picture.
Summary Review 10:  the product came very fast and was just like how amazon
explained itthe ring is very clearly written NO WAR and thick which i likebut i
guess if you like small rings this isnt your ring

QUERY 8 SUMMARIES
Summary Review 1:  This medical alert braclet looked just like its picture and
is nice quality sterling silver.
Summary Review 2:  The silver does not look as in the picture but just like
polished silver. the earrings are as in the picture, stones look good and are
light and comfortable, reasonable quality for the price.
Summary Review 3:  Although the picture looks like metal beads and description
states sterling silver, these are pearls.
Summary Review 4:  I ended up returning the bracelet because I have amethyst
jewelry and it was extremely poor quality. The stones on this bracelet are
extremely pale, more pink than purple.
Summary Review 5:  A must for everyone who is a Tiger fan and owns an Italian
Charm Bracelet. Very nice quality.
Summary Review 6:  Really looks like the picture. Silver is nicely finished and
the enamel is a nice highlight.
Summary Review 7:  It is an amazing price for such a beautiful pendant. It picks
up the colors of your clothing.
Summary Review 8:  The silver and black enamel cross ring speaks for its
self.Utterly undescibable. Many have ask where to purchase the ring so they to
can share their inner feelings in an outward way without having to say a work,
but allow the ring to speak for them.
Summary Review 9:  The ring was nice and looked like picture but had a crack in
one Garnet and another one had a large chip.
Summary Review 10:  I bought it for my aunt as a present and the color is very
nice. The message is very positive and it looks very pretty.
```

5a - Evaluation

[44]:
```python
_values = list()

for i, value in enumerate(corpus_list):
  _values.append(value[0])
  print(f"{i+1} - {value[0]}")
```

```
1 - my wife loves the ring, it was a great gift. extremelly cheap and high
quality.
2 - It is as nice as it looks on the picture. :) I like it. :)
3 - Item was great quality and came promptly. I'm very happy with it and
recommend it unreservedly.
```

```
4 - very good for everyday wear or dressing up
5 - The flute charm is so detailed and is of very high quality. You can see all
the keys, any flute fan would adore having this item.
6 - 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.
7 - The diamond had a crack in one Garnet and another one had a large chip.
8 - This medical alert braclet looked just like its picture and is nice quality
sterling silver.
```

[45]:
```
!pip install tabulate

from tabulate import tabulate
from summertime.evaluation import SUPPORTED_EVALUATION_METRICS
from summertime.evaluation import BertScore, Meteor, Bleu
import summertime.evaluation as st_eval

print(SUPPORTED_EVALUATION_METRICS)

targets = [
    'Gifted a ring to 16-year-old niece who loved and was happy to receive it.',
    'Ring is pretty, but the metal is insubstantial and can easily be pushed in.
 ↪',
    'High-quality item arrived quickly. Extremely satisfied and wholeheartedly␣
 ↪recommend.',
    'Suitable and fashionable for dressy occasions.',
    'Received nice product earlier than expected. Will reorder.',
    'Colorful turtle items resembling the photograph. Comparable to expensive␣
 ↪ones, and beautiful.',
    'Ring appears smaller than pictured with very small diamonds, causing␣
 ↪disappointment.',
    'Medical alert bracelet matches picture and is made of quality sterling␣
 ↪silver.',
]

summaries = _values

# Calculate BertScore
bert_metric = BertScore()
bert_results = bert_metric.evaluate(summaries, targets)

# Calculate Meteor
meteor_metric = Meteor()
meteor_results = meteor_metric.evaluate(summaries, targets)

# Calculate BLEU
bleu_metric = Bleu()
bleu_results = bleu_metric.evaluate(summaries, targets)
```

```python
# Print evaluation results
print(f"BertScore: {bert_results}")
print(f"Meteor: {meteor_results}")
print(f"BLEU: {bleu_results}")

# Print evaluation results in a table
table_data = [["BertScore", bert_results['bert_score_f1']],
              ["Meteor", meteor_results['meteor']],
              ["BLEU", bleu_results['bleu']]]

print(tabulate(table_data, headers=["Metric", "Score"], tablefmt="grid"))
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Requirement already satisfied: tabulate in /usr/local/lib/python3.9/dist-
packages (0.8.10)
[<class 'summertime.evaluation.bertscore_metric.BertScore'>, <class
'summertime.evaluation.bleu_metric.Bleu'>, <class
'summertime.evaluation.rouge_metric.Rouge'>, <class
'summertime.evaluation.rougewe_metric.RougeWe'>, <class
'summertime.evaluation.meteor_metric.Meteor'>]
```

```
HBox(children=(FloatProgress(value=0.0, description='Downloading␣
↪(…)okenizer_config.json', max=28.0, style=Pro…


HBox(children=(FloatProgress(value=0.0, description='Downloading (…)lve/main/
↪config.json', max=570.0, style=Pr…


HBox(children=(FloatProgress(value=0.0, description='Downloading (…)solve/main/
↪vocab.txt', max=231508.0, style…


HBox(children=(FloatProgress(value=0.0, description='Downloading pytorch_model.
↪bin', max=440473133.0, style=Pr…
```

```
Some weights of the model checkpoint at bert-base-uncased were not used when
initializing BertModel: ['cls.predictions.transform.dense.weight',
'cls.predictions.decoder.weight', 'cls.predictions.transform.dense.bias',
'cls.predictions.bias', 'cls.predictions.transform.LayerNorm.weight',
'cls.seq_relationship.weight', 'cls.seq_relationship.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).
[nltk_data] Downloading package wordnet to /root/nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
```

```
hash_code: bert-base-uncased_L8_no-idf_version=0.3.12(hug_trans=4.27.1)
BertScore: {'bert_score_f1': 0.5179015398025513}
Meteor: {'meteor': 0.11549545745316996}
BLEU: {'bleu': 0.03292378035323646}
+-----------+-----------+
| Metric    |     Score |
+===========+===========+
| BertScore | 0.517902  |
+-----------+-----------+
| Meteor    | 0.115495  |
+-----------+-----------+
| BLEU      | 0.0329238 |
+-----------+-----------+
```

Summarize evaluation anwer - TODO

5b - Interactive summarization

[56]:
```
no_sentences = input("Enter number of sentences you desire to be summarized")
# list_val = list()
user_summaries = list()
summary_list = list()
for i in range(0,int(no_sentences)):
  list_val = input(f"Enter sentence {i+1} \n")
  user_summaries.append(list_val)
  lexrank = model.LexRankModel(user_summaries)
  # # Inference
  summary = lexrank.summarize(user_summaries)
  summary_list.append(summary)

for i, val in enumerate(user_summaries):
  print(f"Summary for Sentence {i+1}")
  print(f"Summary {i+1} - {summary_list[i]}")
  print('\n')


# for key, corpus in enumerate(corpus_list):
#   print('\n' + f"QUERY {key+1} SUMMARIES")
#   lexrank = model.LexRankModel(corpus)
#   # # Inference
#   summary = lexrank.summarize(corpus)
#   summary_list.append(summary)
#   for i in range(len(summary)):
```

```
#     print("Summary Review %d: "%(i+1), summary[i])
```

```
Enter number of sentences you desire to be summarized2
Enter sentence 1
Arsenal was founded in 1886 as Dial Square, by workers at the Royal Arsenal in
Woolwich. The club's first game was against Eastern Wanderers and ended in a 6-0
victory for Dial Square.
Enter sentence 2
Arsenal is one of the most successful clubs in English football history, having
won 13 league titles, 14 FA Cups, and two European trophies. The club's most
successful period came under the management of Arsene Wenger, who led the team
to three Premier League titles and four FA Cups during his tenure from 1996 to
2018.
Summary for Sentence 1
Summary 1 - ["The club's first game was against Eastern Wanderers and ended in a
6-0 victory for Dial Square. Arsenal was founded in 1886 as Dial Square, by
workers at the Royal Arsenal in Woolwich."]


Summary for Sentence 2
Summary 2 - ["The club's first game was against Eastern Wanderers and ended in a
6-0 victory for Dial Square. Arsenal was founded in 1886 as Dial Square, by
workers at the Royal Arsenal in Woolwich.", "The club's most successful period
came under the management of Arsene Wenger, who led the team to three Premier
League titles and four FA Cups during his tenure from 1996 to 2018. Arsenal is
one of the most successful clubs in English football history, having won 13
league titles, 14 FA Cups, and two European trophies."]
```

[ ]:
```
!sudo apt-get install texlive-xetex texlive-fonts-recommended␣
↪texlive-plain-generic
```

```
Reading package lists… Done
Building dependency tree
Reading state information… Done
The following additional packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono
  fonts-texgyre fonts-urw-base35 javascript-common libapache-pom-java
  libcommons-logging-java libcommons-parent-java libfontbox-java libgs9
  libgs9-common libidn11 libijs-0.35 libjbig2dec0 libjs-jquery libkpathsea6
  libpdfbox-java libptexenc1 libruby2.7 libsynctex2 libteckit0 libtexlua53
  libtexluajit2 libzzip-0-13 lmodern poppler-data preview-latex-style rake
  ruby ruby-minitest ruby-net-telnet ruby-power-assert ruby-test-unit
  ruby-xmlrpc ruby2.7 rubygems-integration t1utils teckit tex-common tex-gyre
  texlive-base texlive-binaries texlive-latex-base texlive-latex-extra
  texlive-latex-recommended texlive-pictures tipa xfonts-encodings
```

```
  xfonts-utils
Suggested packages:
  fonts-noto fonts-freefont-otf | fonts-freefont-ttf apache2 | lighttpd
  | httpd libavalon-framework-java libcommons-logging-java-doc
  libexcalibur-logkit-java liblog4j1.2-java poppler-utils ghostscript
  fonts-japanese-mincho | fonts-ipafont-mincho fonts-japanese-gothic
  | fonts-ipafont-gothic fonts-arphic-ukai fonts-arphic-uming fonts-nanum ri
  ruby-dev bundler debhelper gv | postscript-viewer perl-tk xpdf | pdf-viewer
  xzdec texlive-fonts-recommended-doc texlive-latex-base-doc python3-pygments
  icc-profiles libfile-which-perl libspreadsheet-parseexcel-perl
  texlive-latex-extra-doc texlive-latex-recommended-doc texlive-luatex
  texlive-pstricks dot2tex prerex ruby-tcltk | libtcltk-ruby
  texlive-pictures-doc vprerex
The following NEW packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono
  fonts-texgyre fonts-urw-base35 javascript-common libapache-pom-java
  libcommons-logging-java libcommons-parent-java libfontbox-java libgs9
  libgs9-common libidn11 libijs-0.35 libjbig2dec0 libjs-jquery libkpathsea6
  libpdfbox-java libptexenc1 libruby2.7 libsynctex2 libteckit0 libtexlua53
  libtexluajit2 libzzip-0-13 lmodern poppler-data preview-latex-style rake
  ruby ruby-minitest ruby-net-telnet ruby-power-assert ruby-test-unit
  ruby-xmlrpc ruby2.7 rubygems-integration t1utils teckit tex-common tex-gyre
  texlive-base texlive-binaries texlive-fonts-recommended texlive-latex-base
  texlive-latex-extra texlive-latex-recommended texlive-pictures
  texlive-plain-generic texlive-xetex tipa xfonts-encodings xfonts-utils
0 upgraded, 55 newly installed, 0 to remove and 23 not upgraded.
Need to get 169 MB of archives.
After this operation, 536 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 fonts-droid-fallback all
1:6.0.1r16-1.1 [1,805 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal/main amd64 fonts-lato all 2.0-2
[2,698 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal/main amd64 poppler-data all 0.4.9-2
[1,475 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal/universe amd64 tex-common all 6.13
[32.7 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal/main amd64 fonts-urw-base35 all
20170801.1-3 [6,333 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libgs9-common
all 9.50~dfsg-5ubuntu4.6 [681 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal/main amd64 libidn11 amd64
1.33-2.2ubuntu2 [46.2 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal/main amd64 libijs-0.35 amd64
0.35-15 [15.7 kB]
Get:9 http://archive.ubuntu.com/ubuntu focal/main amd64 libjbig2dec0 amd64
0.18-1ubuntu1 [60.0 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libgs9 amd64
9.50~dfsg-5ubuntu4.6 [2,173 kB]
```

```
Get:11 http://archive.ubuntu.com/ubuntu focal/main amd64 libkpathsea6 amd64
2019.20190605.51237-3build2 [57.0 kB]
Get:12 http://archive.ubuntu.com/ubuntu focal/universe amd64 dvisvgm amd64
2.8.1-1build1 [1,048 kB]
Get:13 http://archive.ubuntu.com/ubuntu focal/universe amd64 fonts-lmodern all
2.004.5-6 [4,532 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 fonts-noto-mono
all 20200323-1build1~ubuntu20.04.1 [80.6 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal/universe amd64 fonts-texgyre all
20180621-3 [10.2 MB]
Get:16 http://archive.ubuntu.com/ubuntu focal/main amd64 javascript-common all
11 [6,066 B]
Get:17 http://archive.ubuntu.com/ubuntu focal/universe amd64 libapache-pom-java
all 18-1 [4,720 B]
Get:18 http://archive.ubuntu.com/ubuntu focal/universe amd64 libcommons-parent-
java all 43-1 [10.8 kB]
Get:19 http://archive.ubuntu.com/ubuntu focal/universe amd64 libcommons-logging-
java all 1.2-2 [60.3 kB]
Get:20 http://archive.ubuntu.com/ubuntu focal/main amd64 libjs-jquery all
3.3.1~dfsg-3 [329 kB]
Get:21 http://archive.ubuntu.com/ubuntu focal/main amd64 libptexenc1 amd64
2019.20190605.51237-3build2 [35.5 kB]
Get:22 http://archive.ubuntu.com/ubuntu focal/main amd64 rubygems-integration
all 1.16 [5,092 B]
Get:23 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 ruby2.7 amd64
2.7.0-5ubuntu1.7 [95.6 kB]
Get:24 http://archive.ubuntu.com/ubuntu focal/main amd64 ruby amd64 1:2.7+1
[5,412 B]
Get:25 http://archive.ubuntu.com/ubuntu focal/main amd64 rake all 13.0.1-4 [61.6
kB]
Get:26 http://archive.ubuntu.com/ubuntu focal/main amd64 ruby-minitest all
5.13.0-1 [40.9 kB]
Get:27 http://archive.ubuntu.com/ubuntu focal/main amd64 ruby-net-telnet all
0.1.1-2 [12.6 kB]
Get:28 http://archive.ubuntu.com/ubuntu focal/main amd64 ruby-power-assert all
1.1.7-1 [11.4 kB]
Get:29 http://archive.ubuntu.com/ubuntu focal/main amd64 ruby-test-unit all
3.3.5-1 [73.2 kB]
Get:30 http://archive.ubuntu.com/ubuntu focal/main amd64 ruby-xmlrpc all 0.3.0-2
[23.8 kB]
Get:31 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libruby2.7
amd64 2.7.0-5ubuntu1.7 [3,533 kB]
Get:32 http://archive.ubuntu.com/ubuntu focal/main amd64 libsynctex2 amd64
2019.20190605.51237-3build2 [55.0 kB]
Get:33 http://archive.ubuntu.com/ubuntu focal/universe amd64 libteckit0 amd64
2.5.8+ds2-5ubuntu2 [320 kB]
Get:34 http://archive.ubuntu.com/ubuntu focal/main amd64 libtexlua53 amd64
2019.20190605.51237-3build2 [105 kB]
```

```
Get:35 http://archive.ubuntu.com/ubuntu focal/main amd64 libtexluajit2 amd64
2019.20190605.51237-3build2 [235 kB]
Get:36 http://archive.ubuntu.com/ubuntu focal/universe amd64 libzzip-0-13 amd64
0.13.62-3.2ubuntu1 [26.2 kB]
Get:37 http://archive.ubuntu.com/ubuntu focal/main amd64 xfonts-encodings all
1:1.0.5-0ubuntu1 [573 kB]
Get:38 http://archive.ubuntu.com/ubuntu focal/main amd64 xfonts-utils amd64
1:7.7+6 [91.5 kB]
Get:39 http://archive.ubuntu.com/ubuntu focal/universe amd64 lmodern all
2.004.5-6 [9,474 kB]
Get:40 http://archive.ubuntu.com/ubuntu focal/universe amd64 preview-latex-style
all 11.91-2ubuntu2 [184 kB]
Get:41 http://archive.ubuntu.com/ubuntu focal/main amd64 t1utils amd64 1.41-3
[56.1 kB]
Get:42 http://archive.ubuntu.com/ubuntu focal/universe amd64 teckit amd64
2.5.8+ds2-5ubuntu2 [687 kB]
Get:43 http://archive.ubuntu.com/ubuntu focal/universe amd64 tex-gyre all
20180621-3 [6,209 kB]
Get:44 http://archive.ubuntu.com/ubuntu focal/universe amd64 texlive-binaries
amd64 2019.20190605.51237-3build2 [8,041 kB]
Get:45 http://archive.ubuntu.com/ubuntu focal/universe amd64 texlive-base all
2019.20200218-1 [20.8 MB]
Get:46 http://archive.ubuntu.com/ubuntu focal/universe amd64 texlive-fonts-
recommended all 2019.20200218-1 [4,972 kB]
Get:47 http://archive.ubuntu.com/ubuntu focal/universe amd64 texlive-latex-base
all 2019.20200218-1 [990 kB]
Get:48 http://archive.ubuntu.com/ubuntu focal/universe amd64 libfontbox-java all
1:1.8.16-2 [207 kB]
Get:49 http://archive.ubuntu.com/ubuntu focal/universe amd64 libpdfbox-java all
1:1.8.16-2 [5,199 kB]
Get:50 http://archive.ubuntu.com/ubuntu focal/universe amd64 texlive-latex-
recommended all 2019.20200218-1 [15.7 MB]
Get:51 http://archive.ubuntu.com/ubuntu focal/universe amd64 texlive-pictures
all 2019.20200218-1 [4,492 kB]
Get:52 http://archive.ubuntu.com/ubuntu focal/universe amd64 texlive-latex-extra
all 2019.202000218-1 [12.5 MB]
Get:53 http://archive.ubuntu.com/ubuntu focal/universe amd64 texlive-plain-
generic all 2019.202000218-1 [24.6 MB]
82% [53 texlive-plain-generic 6,852 kB/24.6 MB 28%]              4,961 kB/s 7s
```

[1]:
```
# %%shell
!jupyter nbconvert --to pdf '/content/drive/My Drive/Colab Notebooks/COP509cw/
↪NLPCoursework.ipynb'
```

```
[NbConvertApp] Converting notebook /content/drive/My Drive/Colab
Notebooks/COP509cw/NLPCoursework.ipynb to pdf
/usr/local/lib/python3.9/dist-packages/nbconvert/filters/datatypefilter.py:41:
```

```
UserWarning: Your element with mimetype(s) dict_keys(['text/html']) is not able
to be represented.
  warn(
[NbConvertApp] Support files will be in NLPCoursework_files/
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Making directory ./NLPCoursework_files
[NbConvertApp] Writing 380052 bytes to notebook.tex
[NbConvertApp] Building PDF
Traceback (most recent call last):
  File "/usr/local/bin/jupyter-nbconvert", line 8, in <module>
    sys.exit(main())
  File "/usr/local/lib/python3.9/dist-packages/jupyter_core/application.py",
line 277, in launch_instance
    return super().launch_instance(argv=argv, **kwargs)
  File "/usr/local/lib/python3.9/dist-packages/traitlets/config/application.py",
line 992, in launch_instance
    app.start()
  File "/usr/local/lib/python3.9/dist-packages/nbconvert/nbconvertapp.py", line
423, in start
    self.convert_notebooks()
  File "/usr/local/lib/python3.9/dist-packages/nbconvert/nbconvertapp.py", line
597, in convert_notebooks
    self.convert_single_notebook(notebook_filename)
  File "/usr/local/lib/python3.9/dist-packages/nbconvert/nbconvertapp.py", line
560, in convert_single_notebook
    output, resources = self.export_single_notebook(
  File "/usr/local/lib/python3.9/dist-packages/nbconvert/nbconvertapp.py", line
488, in export_single_notebook
```

```
    output, resources = self.exporter.from_filename(
  File "/usr/local/lib/python3.9/dist-packages/nbconvert/exporters/exporter.py",
line 189, in from_filename
    return self.from_file(f, resources=resources, **kw)
  File "/usr/local/lib/python3.9/dist-packages/nbconvert/exporters/exporter.py",
line 206, in from_file
    return self.from_notebook_node(
  File "/usr/local/lib/python3.9/dist-packages/nbconvert/exporters/pdf.py", line
194, in from_notebook_node
    self.run_latex(tex_file)
  File "/usr/local/lib/python3.9/dist-packages/nbconvert/exporters/pdf.py", line
164, in run_latex
    return self.run_command(
  File "/usr/local/lib/python3.9/dist-packages/nbconvert/exporters/pdf.py", line
111, in run_command
    raise OSError(
OSError: xelatex not found on PATH, if you have not installed xelatex you may
need to do so. Find further instructions at
https://nbconvert.readthedocs.io/en/latest/install.html#installing-tex.
```

[ ]: