HW3

March 4, 2023

1 BA 820 Homework 3 (100 Points)

Group Member Names:

Reminder: you should not be sharing code across groups

Please submit 1) PDF answers and 2) python notebook. Grading will be based on the homework answer write up PDF. Python notebook is for reference and back up only. So please make sure that your all the outputs and answers are clearly visible in the pdf.

1.1 1 Latent Dirichlet Allocation [60pts]

In this problem, we will use Latent Dirichlet Allocation to perform topic modeling on Amazon Review datasets. In particular, we will take an in-depth look at different aspects of LDA model.

1.2 1.1 Installation

To perform LDA and visualize, please use Python 3.X. You will also need to install Numpy, Scipy, gensim, nltk, pyLDAvis library. Refer to requirements.txt for more details. Use the following code to install the libraries.

[ ]:
```

```

[14]:
```
# %pip install gensim
# # install gensim for LDA
# %pip install nltk
# # install nltk to preprocess sentences
# %pip install pyldavis
# # to visualize LDA topics
# %pip install matplotlib
# # for plotting
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

The cell below tests if the packages we need have been installed correctly, and that we are in the correct environment.

[14]:
```
import gensim
import nltk
nltk.download("stopwords")
import pyLDAvis
import matplotlib
%matplotlib inline
import gzip # to unzip the data
import re # to replace punctuations
from nltk.corpus import stopwords # list of stopwords
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/ryanli/nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
```

1.3 1.2 Datasets

You can download the Amazon reviews dataset of Cellphones & Accessory 5-Core Data here. Place the downloaded dataset in the same folder as this notebook. You can use the following code to read a datat from GZIp file

[16]:
```
# A function to read the zipped data at a specfic path
#
# How to use:
# PATH = "/path/to/file"
# for line in parse(PATH):
#   do something with line
#
def parse(path):
    g = gzip.open(path, 'r')
    for l in g:
        yield eval(l)

parse('./reviews_Cell_Phones_and_Accessories_5.json.gz')
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[16]:
```
<generator object parse at 0x7fe3a9943270>
```

1.4 1.3 Data Cleaning

Now we will preprocess the data using the following steps: 1. Remove stopwords 2. Lower-case all words 3. Remove words with less than 2 characters 4. Remove punctuation 5. Split each sentence into a list of words

[17]:
```python
# A function to clean a single line of text
def clean_line(line):
    """ Clean stopwords and punction for each line

    Args:
        line (string): one line in file

    Returns:
        list(str): a list of all words in the sentence
    """
    punctuationRegex = r'\W+|\d+'
    stopWords = set(stopwords.words('english'))
    line = line.split(" ")
    filtered_content = []
    for word in line:
        if word not in stopWords:
            if len(word) >= 2:
                word = re.sub(punctuationRegex, '', word)
                filtered_content.append(word.lower())

        #########################
        # YOUR CLEANING CODE HERE
        #########################
    return filtered_content
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

Finally, we put parse() and clean_line() function together and then extract the first 10,000 reviews into a new text file as your experiment dataset

[18]:
```python
def read_dataset(fname):
    """ Read the 10000 lines in given dataset into list and clean stop words.

    Args:
        fname (string): filename of Amazon Review Dataset

    Returns:
        list of list of words: we view each document as a list, including a␣
↪list of all words
    """
    count = 0
    exp_dataset = []
    for review in parse(fname):
        line = review["reviewText"]
        new_line = clean_line(line)
        exp_dataset.append(new_line)
        count += 1
        if count > 10000:
            break
    return exp_dataset
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[19]:
```
%%time
r = read_dataset("reviews_Cell_Phones_and_Accessories_5.json.gz")
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

```
CPU times: user 1.79 s, sys: 126 ms, total: 1.92 s
Wall time: 1.92 s
```

1.5 1.4 Topic Analysis

[**5pts**] **Q1.4.1.1** Use topic numbers 3, 6, 9, 12, 15 respectively and print out all topics with 5 words.

For this We will use gensim to train an LDA model. gensim requires the following steps:

Construct a gensim.corpora.dictionary from the dataset Construct a gensim "corpus" using this dictionary, by mapping each word to an index in the dictionary Run LDA on this corpus

[20]:
```
import gensim.corpora as corpora
dictionary = gensim.corpora.Dictionary(r)# create a gensim dictionary, store it␣
↪in variable "dictionary"
corpus = [dictionary.doc2bow(text) for text in r] # create the gensim corpus,␣
↪store it in variable "corpus"
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

The function below prints the top num words in each topic for a given model.

```
[21]: def print_topic_words(model,num):
          """ print top words in model topics.

          Args:
              model: LDA model

          Returns:
              none
          """
          #########################
          # YOUR CODE HERE
          num_topics = model.num_topics
          for topic_id in range(model.num_topics):

              words = model.show_topic(topic_id, num)
              print('Topic #'+str(topic_id))
              print(words)
          print('*'*100)
          return
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

The following function builds multiple LDA models with number of topics specified in the list num_topics.

```
[22]: def build_num_topic_model(dictionary, corpus, num_topics):
          """ Build lda model with given parameters, use print_topic_words to print␣
       ↪words

          Args:
              dictionary: dictionary built from dataset
              corpus: corpus built from dataset
              num_topics: list of numbers

          Returns:
```

```
        none
    """
    for num_topic in num_topics:
        #########################
        # YOUR CODE HERE
        #    - Build model
        #    - Print the top 5 words
        model = gensim.models.ldamodel.LdaModel(corpus=corpus,␣
↪id2word=dictionary, num_topics=num_topic)
        print_topic_words(model,5)
        print()
        #########################
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[23]:
```
build_num_topic_model(dictionary, corpus, [3, 6, 9, 12, 15])
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

```
Topic #0
[('iphone', 0.049988937), ('case', 0.044588145), ('mophie', 0.029491577),
('pack', 0.016167073), ('battery', 0.015551079)]
Topic #1
[('battery', 0.053863686), ('juice', 0.032217674), ('pack', 0.025239624),
('use', 0.021139733), ('iphone', 0.014763042)]
Topic #2
[('it', 0.011770983), ('', 0.010990056), ('the', 0.010160064), ('one',
0.009855113), ('headset', 0.00896502)]
********************************************************************************
********************

Topic #0
[('iphone', 0.060730338), ('case', 0.048411805), ('mophie', 0.036066864),
('pack', 0.030226164), ('juice', 0.017514575)]
Topic #1
[('phone', 0.025212038), ('', 0.009756702), ('it', 0.006486306), ('the',
0.0062855217), ('nokia', 0.005830606)]
Topic #2
[('chore', 0.021735761), ('', 0.016555201), ('phone', 0.016255949), ('the',
0.013253648), ('battery', 0.008088013)]
Topic #3
[('battery', 0.07535601), ('juice', 0.033460412), ('use', 0.022550948),
('packs', 0.022294905), ('addon', 0.017953185)]
Topic #4
[('it', 0.018092047), ('the', 0.01357703), ('works', 0.009031401), ('one',
0.0084875645), ('charger', 0.008487379)]
Topic #5
[('battery', 0.033448108), ('depth', 0.024660764), ('shed', 0.023491481),
('doubles', 0.019727886), ('juice', 0.019350667)]
********************************************************************************
********************

Topic #0
[('', 0.013399965), ('headset', 0.011195139), ('the', 0.01026294), ('sound',
0.0100378115), ('it', 0.009417792)]
Topic #1
[('jeans', 0.04032441), ('afford', 0.03214429), ('pocket', 0.032126807),
('case', 0.022572855), ('packs', 0.02207533)]
Topic #2
[('ear', 0.010716085), ('headset', 0.010705293), ('it', 0.008945575), ('the',
0.008465755), ('', 0.007923029)]
Topic #3
[('battery', 0.06794924), ('juice', 0.05357414), ('iphone', 0.026931776),
('use', 0.025504222), ('pack', 0.024402974)]
Topic #4
[('phone', 0.005569753), ('otterbox', 0.0030983167), ('case', 0.003000497),
('iphone', 0.002905604), ('belt', 0.0027501856)]
Topic #5
[('depth', 0.047129855), ('chore', 0.045366537), ('indicators', 0.040657707),
('round', 0.03206294), ('use', 0.020910447)]
Topic #6
[('mophie', 0.044429235), ('battery', 0.031780843), ('pack', 0.029753966),
('iphone', 0.026220683), ('the', 0.020310745)]
Topic #7
[('case', 0.14249006), ('iphone', 0.10763665), ('bottom', 0.030545874),
('matte', 0.030047478), ('bulk', 0.024476316)]
Topic #8
[('ear', 0.023304794), ('one', 0.011273804), ('the', 0.010706767), ('',
0.0076914104), ('good', 0.0064517776)]
********************************************************************************
********************

Topic #0
[('phone', 0.033437587), ('n', 0.02029127), ('camera', 0.01701345), ('',
0.009947426), ('the', 0.00878431)]
Topic #1
[('cable', 0.033513956), ('works', 0.01802505), ('work', 0.015509081), ('it',
0.012660433), ('great', 0.012120939)]
Topic #2
[('iphone', 0.05383972), ('case', 0.046995748), ('battery', 0.04511285),
('pack', 0.028889544), ('doubles', 0.023986598)]
Topic #3
[('loops', 0.007488123), ('phone', 0.00474112), ('', 0.0039490783), ('it',
0.00320709), ('grade', 0.0028761753)]
Topic #4
[('usb', 0.021007678), ('charger', 0.019100927), ('charge', 0.01733626), ('',
0.014650443), ('cable', 0.014454197)]
Topic #5
[('case', 0.015256979), ('', 0.011679472), ('good', 0.01155959), ('it',
0.011071538), ('phone', 0.010443613)]
Topic #6
[('', 0.032375928), ('the', 0.013734878), ('phone', 0.010321793), ('headset',
0.010008032), ('use', 0.00745206)]
Topic #7
[('juice', 0.0647375), ('iphone', 0.051919654), ('mophie', 0.043157183),
('case', 0.034868628), ('pack', 0.030433208)]
Topic #8
[('battery', 0.029137425), ('heft', 0.019551078), ('shed', 0.019262668),
('depth', 0.019096432), ('portability', 0.01889739)]
Topic #9
[('sound', 0.02045997), ('quality', 0.019394355), ('headset', 0.013209888),
('good', 0.012940927), ('the', 0.012055912)]
Topic #10
[('mophie', 0.13081421), ('chore', 0.1028071), ('case', 0.015604132),
('wearing', 0.01488472), ('the', 0.012020744)]
Topic #11
[('phone', 0.012894277), ('the', 0.0103597045), ('', 0.007468583), ('great',
0.007218316), ('it', 0.0071018576)]
********************************************************************************
********************

Topic #0
[('iphone', 0.035331883), ('gs', 0.026429344), ('it', 0.010819086), ('great',
0.008700517), ('product', 0.008048945)]
Topic #1
[('juice', 0.07784112), ('battery', 0.06418749), ('iphone', 0.029490737),
('trade', 0.02726192), ('increased', 0.027169129)]
Topic #2
[('the', 0.015835652), ('case', 0.015587554), ('use', 0.011977891), ('one',
0.011945349), ('phone', 0.011728513)]
Topic #3
[('headset', 0.023420885), ('bluetooth', 0.015204793), ('', 0.013330614),
('the', 0.01117781), ('headphones', 0.008446767)]
Topic #4
[('battery', 0.058722995), ('use', 0.021935757), ('extra', 0.021885075),
('need', 0.021299466), ('packs', 0.020999534)]
Topic #5
[('shed', 0.05552653), ('addon', 0.026653752), ('phone', 0.022245193), ('use',
0.013120688), ('the', 0.0129701765)]
Topic #6
[('iphone', 0.11490369), ('case', 0.11443563), ('mophie', 0.092860885),
('bottom', 0.036691163), ('pocket', 0.029011954)]
Topic #7
[('it', 0.0124197), ('phone', 0.0056719584), ('works', 0.005231663), ('this',
0.005160905), ('great', 0.0051017627)]
Topic #8
[('indicators', 0.018574141), ('chore', 0.017971773), ('round', 0.01753495),
('the', 0.014975294), ('use', 0.0145416455)]
Topic #9
[('one', 0.02378402), ('it', 0.0135599235), ('phone', 0.013382202), ('good',
0.0123992115), ('well', 0.011036244)]
Topic #10
[('ipad', 0.008079812), ('word', 0.0047998475), ('', 0.004608364), ('product',
0.0039943974), ('good', 0.0039774044)]
Topic #11
[('pack', 0.1117998), ('travelling', 0.05195673), ('juice', 0.048136707),
('battery', 0.04060678), ('afford', 0.038041648)]
Topic #12
[('cable', 0.041248135), ('price', 0.01824455), ('works', 0.015823437), ('it',
0.01282719), ('good', 0.012024066)]
Topic #13
[('it', 0.00528682), ('use', 0.0051802928), ('this', 0.0049495595), ('would',
0.0045988085), ('timely', 0.0045871367)]
Topic #14
[('phone', 0.020641403), ('screen', 0.011001832), ('the', 0.01016877), ('email',
0.009359231), ('device', 0.007116174)]
********************************************************************************
********************
```

[3pts] Q1.4.1.2 Explain what could be interpreted for each topics, and describe the similarity and difference between different topic numbers.

From all the topics, we can see that they are related to the product. For example, topic 0 is related to the product's quality, topic 1 is related to the product's price, topic 2 is related to the product's shipping, and topic 3 is related to the product's accessories. The topics are different from each other because the number of topics is different. The more topics, the more detailed the topics are.

[2pts] Q1.4.1.3 Which topic number would you choose? Explain.

I would choose topic 3 because it is the most detailed topic. It has the most words in the topic, and the words in the topic are more related to the product.

1.6 1.5 Model Evaluation

[12 pts] Q1.5.1 Now we investigate two methods to evaluate our model and choose the topic number

1.Perplexity is a measurement of how well a probability distribution or probability model predicts a sample. A low perplexity indicates the probability distribution is good at predicting the sample. We can use model.log_perplexity(document) to evaluate the perplexity of our LDA model.

2.Topic coherence is a one type of interpretability measurement for a topic. It measures if a set of top keywords describe a coherent and singular concept. A good topic will have high topic coherence score. We can use CoherenceModel(model=ldamodel).get_coherence() to calculate it.

Plot Perplexity and topic coherence scores of our LDA model for topic number 3,6,9,12,15,20,50.

The code below trains topic models with different numbers of topics and measures their coherence and perplexity.

[62]:
```
# perplexity
# run different number of topics to get perplexity and coherence value for this␣
↪model
from gensim.models.coherencemodel import CoherenceModel
def get_measurement_for_model(dictionary, corpus, topic_nums):
    """ Build lda model with given parameters

    Args:
        dictionary: dictionary built from dataset
        corpus: corpus built from dataset
        topic_nums: a list contains all possible topic number

    Returns:
        2 lists: one of perplexities, and one of coherence value
    """
    perplexity = []
    coherence_value=[]
    for num_topic in topic_nums:
        #########################
        # YOUR CODE HERE
        #   - Build model
        #   - Compute and store coherence
        #   - Compute and store perplexity
        lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,␣
↪id2word=dictionary, num_topics=num_topic)
        coherence_model_lda = CoherenceModel(model=lda_model, texts=r,␣
↪dictionary=dictionary, coherence='c_v')
        coherence_lda = coherence_model_lda.get_coherence()
        coherence_value.append(coherence_lda)
        perplexity.append(lda_model.log_perplexity(corpus))
        #########################
    return perplexity,coherence_value
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[63]:
```
perplexity, coherence = get_measurement_for_model(dictionary, corpus, [3, 6, 9,␣
↪12, 15, 20, 50])
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[64]:
```
print(perplexity)
print(coherence)
```

```
[-7.989182071149332, -8.390229442833055, -8.346125040279292, -8.633199534861175,
-8.83779426863714, -9.140899563844522, -10.91491858770547]
[0.2903301449275196, 0.2765158146490121, 0.28660017179148223,
0.26913343896078196, 0.27619421568994695, 0.2801812799751062,
0.3010712490269261]
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

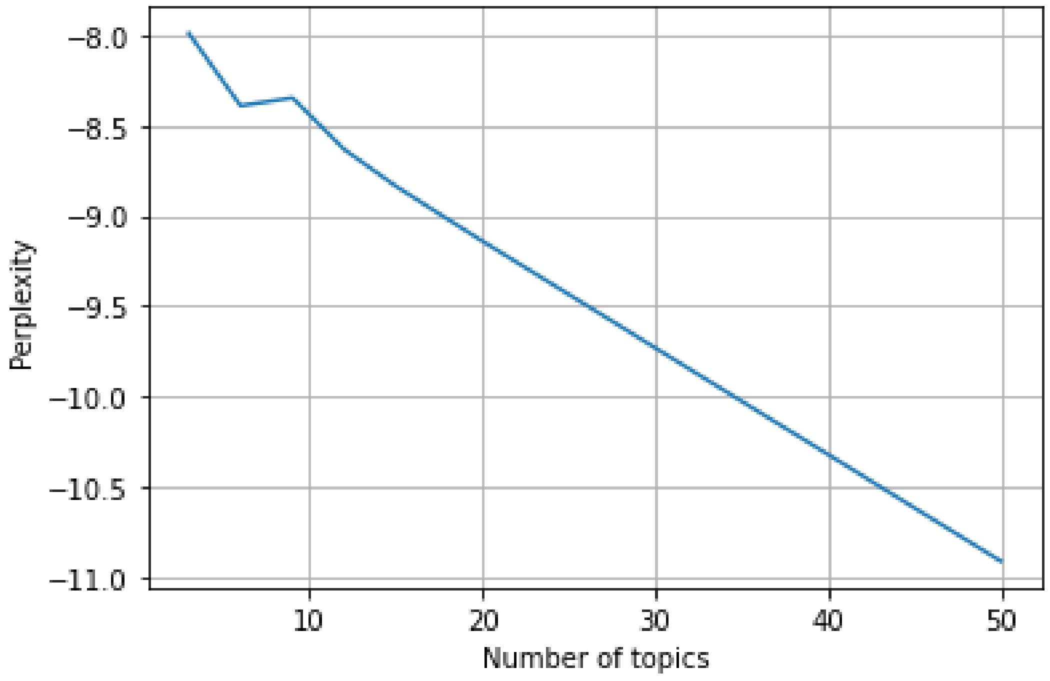
We can now plot the coherence and perplexity of each model.

[27]:
```
import matplotlib.pyplot as plt
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[65]:
```
plt.plot([3, 6, 9, 12, 15, 20, 50], perplexity)
plt.grid()
plt.xlabel("Number of topics")
plt.ylabel("Perplexity")
plt.show()
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```



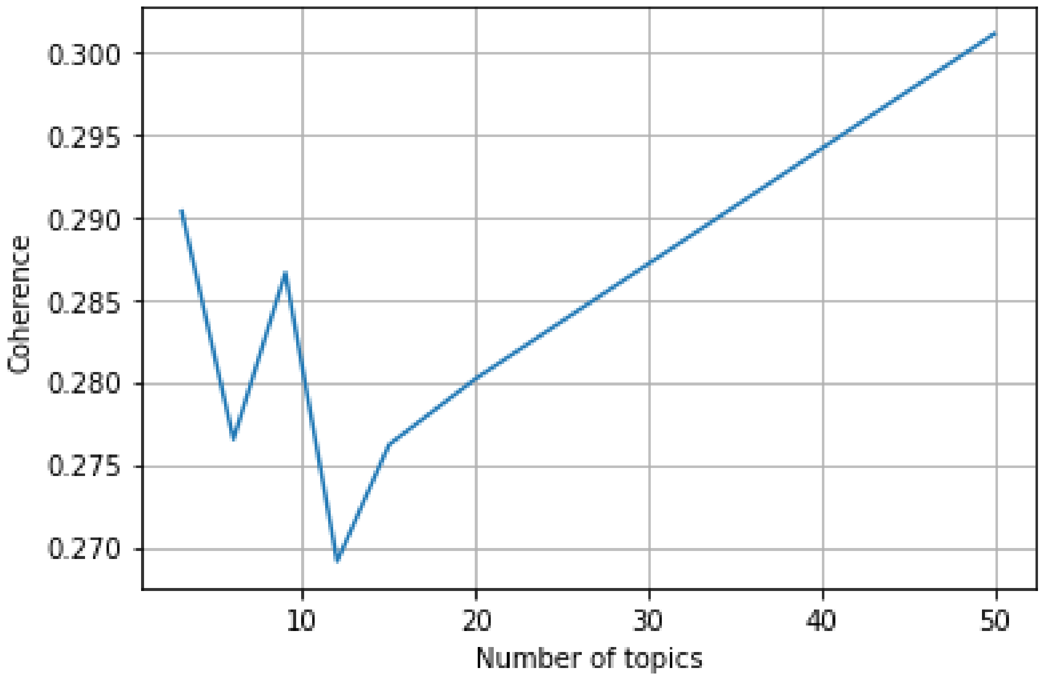
[2pts] Q1.5.2 From the above graph what topic number would you choose and why? Is it a good idea to choose the topic number based on perplexity? why or why not?

I would choose 50 because it has the lowest perplexity. It is a good idea to choose the topic number based on perplexity because it is a measurement of how well a probability distribution or probability model predicts a sample. A low perplexity indicates the probability distribution is good at predicting the sample. However, coherence is also important because it measures if a set of top keywords describe a coherent and singular concept. A good topic will have high topic coherence score.

[66]:
```python
plt.plot([3, 6, 9, 12, 15, 20, 50], coherence)
plt.grid()
plt.xlabel("Number of topics")
plt.ylabel("Coherence")
plt.show()
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
```

```
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```



[2pts] **Q1.5.3** From the above graph what topic number would you choose and why?

I will choose 50 because it has the highest coherence. It is a good idea to choose the topic number based on coherence because it measures if a set of top keywords describe a coherent and singular concept. A good topic will have high topic coherence score.

[4pts]Q1.5.4 Compare two methods you implemented in the previous quesions, which one do you think is better and why? In answering, please discuss the actual topics generated.

I would choose 50 because it has the highest coherence even though it is not the lowest perplexity. The both features are important. The coherence is more important than the perplexity because the coherence is a measurement of interpretability.

1.7 1.6 Alpha and Beta in LDA

[7pts]Q1.6.1 In this problem, we will check the two most important parameters in LDA model: alpha and beta. Alpha represents document-topic density - with a higher alpha, documents are made up of more topics, and with lower alpha, documents contain fewer topics. Beta represents topic-word density - with a high beta, topics are made up of most of the words in the corpus, and with a low beta they consist of few words.

[67]:
```
best_topic_num = 50 # CHANGE THIS
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[68]:
```
#model 1
#########################
# YOUR CODE HERE
#   - Build model for alpha = 1/num_topic = eta
#   - Print top words
model1 = gensim.models.ldamodel.LdaModel(corpus=corpus, id2word=dictionary,␣
 ↪num_topics=best_topic_num, alpha=1/best_topic_num, eta=1/best_topic_num)
print_topic_words(model1,5)
#########################
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)

Topic #0
[('n', 0.08150391), ('original', 0.031707175), ('good', 0.022617655), ('works',
0.021976823), ('price', 0.011978005)]
Topic #1
[('lacks', 0.015647171), ('thicker', 0.011835372), ('tips', 0.011807772),
('igo', 0.011707211), ('sync', 0.010087838)]
Topic #2
[('phone', 0.015390878), ('the', 0.011762522), ('', 0.009820316), ('led',
0.0087164985), ('it', 0.007119962)]
Topic #3
[('battery', 0.023713734), ('pod', 0.014484513), ('the', 0.009952275), ('old',
0.009927774), ('it', 0.00966524)]
Topic #4
[('pack', 0.033400457), ('mirror', 0.027951114), ('screens', 0.020829944),
('the', 0.013018089), ('wonderful', 0.008920175)]
Topic #5
[('phone', 0.026403463), ('', 0.0183843), ('the', 0.017029699), ('it',
0.009529888), ('get', 0.008001397)]
Topic #6
[('phone', 0.018074982), ('one', 0.017590076), ('velcro', 0.01044999), ('n',
0.009737546), ('duty', 0.0082156975)]
Topic #7
[('juice', 0.49867427), ('air', 0.052546278), ('extra', 0.025661541), ('for',
0.022732276), ('use', 0.022527141)]
```

```
Topic #8
[('flashlight', 0.053401634), ('affordable', 0.014687794), ('rate',
0.010503267), ('rapid', 0.008528651), ('lined', 0.008504638)]
Topic #9
[('phone', 0.021966329), ('case', 0.020538151), ('purse', 0.0138481995),
('pouch', 0.01170281), ('would', 0.010195203)]
Topic #10
[('seller', 0.0846574), ('shipped', 0.022499163), ('refund', 0.021171927),
('itouch', 0.0132743055), ('fantastic', 0.0118661765)]
Topic #11
[('', 0.02772597), ('blueant', 0.0157241), ('headset', 0.009983467), ('phone',
0.009720894), ('the', 0.008902677)]
Topic #12
[('bubble', 0.030781435), ('cam', 0.0036592647), ('rendered', 0.0032812227),
('restored', 0.0022876416), ('bam', 0.0018173596)]
Topic #13
[('jeans', 0.18138456), ('afford', 0.18015544), ('purse', 0.052629236),
('matte', 0.045941856), ('convenient', 0.036689088)]
Topic #14
[('phone', 0.012801644), ('', 0.01270811), ('sweat', 0.012332326), ('mount',
0.008863422), ('the', 0.008350288)]
Topic #15
[('loops', 0.03233592), ('ps', 0.02900893), ('spending', 0.011040307), ('gifts',
0.008854071), ('intended', 0.008686149)]
Topic #16
[('', 0.024221692), ('phone', 0.012333863), ('penny', 0.0061500347), ('good',
0.0058604595), ('screen', 0.0058020963)]
Topic #17
[('headphones', 0.025398223), ('sound', 0.023503825), ('use', 0.013769354),
('quality', 0.013741889), ('bluetooth', 0.013539093)]
Topic #18
[('it', 0.02601932), ('product', 0.012701698), ('iphone', 0.011101681), ('this',
0.010265709), ('works', 0.00831027)]
Topic #19
[('pack', 0.05052211), ('battery', 0.041913625), ('use', 0.02755823),
('doubles', 0.026550064), ('iphone', 0.016488058)]
Topic #20
[('otterbox', 0.021030359), ('iphone', 0.012082325), ('', 0.010314211), ('the',
0.01029776), ('it', 0.008022535)]
Topic #21
[('ok', 0.031032855), ('daughter', 0.01972418), ('gift', 0.017652556), ('it',
0.012442405), ('product', 0.011264591)]
Topic #22
[('headset', 0.034030516), ('ear', 0.03398249), ('the', 0.021178687), ('it',
0.013661281), ('bluetooth', 0.01256133)]
Topic #23
[('cable', 0.06760532), ('great', 0.043141626), ('works', 0.0346477),
('product', 0.028907737), ('one', 0.022312162)]
```

```
Topic #24
[('case', 0.12973256), ('iphone', 0.07302019), ('increased', 0.036834635),
('accessible', 0.033440925), ('pocket', 0.02759473)]
Topic #25
[('heft', 0.06916502), ('chore', 0.065072514), ('depth', 0.06482637),
('wearing', 0.016560556), ('life', 0.015156227)]
Topic #26
[('travelling', 0.4512853), ('thanks', 0.041919213), ('convenient',
0.034737106), ('great', 0.026524166), ('taking', 0.015485113)]
Topic #27
[('', 0.024081847), ('usb', 0.019472936), ('computer', 0.015122433), ('plug',
0.008938273), ('port', 0.007975126)]
Topic #28
[('protector', 0.056678344), ('screen', 0.050192293), ('it', 0.036768593),
('great', 0.026718738), ('works', 0.024452584)]
Topic #29
[('phone', 0.019074183), ('', 0.017149758), ('battery', 0.013709087), ('it',
0.009462544), ('this', 0.008579075)]
Topic #30
[('samsung', 0.05586437), ('s', 0.030907726), ('galaxy', 0.025308654), ('it',
0.019901648), ('one', 0.014990237)]
Topic #31
[('phone', 0.035851322), ('it', 0.011481166), ('the', 0.008856471), ('',
0.008557167), ('use', 0.007538767)]
Topic #32
[('didnt', 0.019977925), ('phone', 0.014637157), ('time', 0.008399577),
('icons', 0.006709329), ('hoping', 0.0058139837)]
Topic #33
[('screen', 0.024627907), ('phone', 0.01978811), ('case', 0.01734226), ('one',
0.017322892), ('like', 0.015295189)]
Topic #34
[('iphone', 0.479854), ('gs', 0.1291745), ('shed', 0.12559545), ('protective',
0.043148004), ('pack', 0.0256681)]
Topic #35
[('mophie', 0.27174), ('case', 0.08968944), ('bulk', 0.060948893), ('finish',
0.051669303), ('matte', 0.05054209)]
Topic #36
[('phone', 0.04337674), ('it', 0.01377613), ('', 0.013171715), ('use',
0.01289538), ('phones', 0.012127003)]
Topic #37
[('', 0.031733435), ('tip', 0.022382153), ('connectors', 0.016716909), ('phone',
0.011189049), ('the', 0.007864059)]
Topic #38
[('cheap', 0.011034539), ('mics', 0.010673437), ('scratched', 0.010496814),
('ip', 0.008909802), ('alternate', 0.008422939)]
Topic #39
[('phone', 0.033891294), ('microusb', 0.02478277), ('lg', 0.020448413),
('cable', 0.020321863), ('e', 0.019260745)]
```

```
Topic #40
[('work', 0.02390374), ('dont', 0.019188099), ('waste', 0.012386325), ('buy',
0.012143031), ('spend', 0.011434159)]
Topic #41
[('described', 0.017363744), ('the', 0.014161457), ('it', 0.012422122), ('evo',
0.011035296), ('nexus', 0.010511671)]
Topic #42
[('jacket', 0.36310437), ('air', 0.05991357), ('pocket', 0.039723184),
('bottom', 0.028704744), ('case', 0.019913428)]
Topic #43
[('battery', 0.15366985), ('juice', 0.10062175), ('portability', 0.062236436),
('wife', 0.022639873), ('iphones', 0.022588626)]
Topic #44
[('film', 0.016757764), ('the', 0.014528215), ('earphones', 0.0109294355),
('booster', 0.01090841), ('earbuds', 0.009144088)]
Topic #45
[('', 0.014568037), ('quality', 0.014457493), ('one', 0.01344864), ('working',
0.01212067), ('tooth', 0.011387442)]
Topic #46
[('phone', 0.03566572), ('it', 0.022548955), ('case', 0.021441573), ('one',
0.014330054), ('screen', 0.013493303)]
Topic #47
[('apple', 0.07717839), ('de', 0.019406296), ('brands', 0.00969916), ('product',
0.008546699), ('', 0.008039034)]
Topic #48
[('phone', 0.021200474), ('dock', 0.014655319), ('unit', 0.013212308), ('the',
0.010967679), ('well', 0.007238499)]
Topic #49
[('he', 0.023444582), ('husband', 0.020740457), ('lol', 0.0102017615), ('likes',
0.009979018), ('constructed', 0.009382218)]
********************************************************************************
********************
```

[69]:
```
#model 2
model2 = gensim.models.ldamodel.LdaModel(corpus=corpus, id2word=dictionary,␣
↪num_topics=best_topic_num, alpha=1/2, eta=1/5)
print_topic_words(model2,5)
#########################
# YOUR CODE HERE
#   - Build model for alpha = 1/2, eta = 1/5
#   - Print top words
#########################
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
```

```
  and should_run_async(code)
Topic #0
[('phone', 0.0020630762), ('encumbered', 0.0017770107), ('beefs', 0.0017770107),
('attachedif', 0.0017770107), ('effectly', 0.0017770107)]
Topic #1
[('the', 0.0033137626), ('it', 0.0032906923), ('', 0.0030425761), ('phone',
0.0027707843), ('one', 0.002485853)]
Topic #2
[('works', 0.015176524), ('it', 0.013756396), ('phone', 0.011935157), ('one',
0.011423237), ('cable', 0.009912332)]
Topic #3
[('one', 0.0027666225), ('great', 0.0022899495), ('works', 0.0020769553),
('price', 0.0018719853), ('phone', 0.0017766245)]
Topic #4
[('it', 0.00316266), ('', 0.00276303), ('encumbered', 0.0017132206),
('attachedif', 0.0017132206), ('effectly', 0.0017132206)]
Topic #5
[('beefs', 0.0018398087), ('effectly', 0.0018398087), ('encumbered',
0.0018398087), ('attachedif', 0.0018398087), ('phone', 0.0016889035)]
Topic #6
[('it', 0.002437534), ('great', 0.0019352613), ('phone', 0.0019272502),
('effectly', 0.0018195558), ('attachedif', 0.0018195558)]
Topic #7
[('it', 0.003038537), ('phone', 0.002063638), ('great', 0.0019847583), ('one',
0.0019798211), ('good', 0.001849653)]
Topic #8
[('pre', 0.0037843247), ('it', 0.0019429433), ('beefs', 0.001796659),
('attachedif', 0.001796659), ('encumbered', 0.001796659)]
Topic #9
[('gs', 0.0037991167), ('one', 0.002411876), ('use', 0.0023385787), ('this',
0.0022846768), ('it', 0.0021240688)]
Topic #10
[('it', 0.0031591544), ('one', 0.002208035), ('great', 0.001935799), ('phone',
0.0018227089), ('beefs', 0.0016651202)]
Topic #11
[('phone', 0.002071717), ('effectly', 0.0017961921), ('beefs', 0.0017961921),
('attachedif', 0.0017961921), ('encumbered', 0.0017961921)]
Topic #12
[('it', 0.0023174724), ('great', 0.0020272308), ('beefs', 0.0018124571),
('attachedif', 0.0018124571), ('effectly', 0.0018124571)]
Topic #13
[('it', 0.0035628728), ('docking', 0.0030746905), ('phone', 0.0027159948),
('great', 0.002080564), ('', 0.0020278087)]
Topic #14
[('phone', 0.0030826097), ('it', 0.0027022955), ('beefs', 0.0017761315),
('encumbered', 0.0017761315), ('attachedif', 0.0017761315)]
Topic #15
```

```
[('great', 0.0025329073), ('it', 0.0022246875), ('one', 0.002024246), ('works',
0.0018577151), ('the', 0.0017139147)]
Topic #16
[('case', 0.097491294), ('matte', 0.06414093), ('finish', 0.04093917),
('bottom', 0.036798656), ('bulk', 0.034011416)]
Topic #17
[('phone', 0.023629239), ('nokia', 0.009087533), ('trade', 0.008147552), ('',
0.007988902), ('the', 0.007176746)]
Topic #18
[('it', 0.0023117096), ('use', 0.0021047902), ('phone', 0.0019691496),
('encumbered', 0.001687233), ('attachedif', 0.001687233)]
Topic #19
[('phone', 0.002539598), ('it', 0.002006972), ('beefs', 0.0018106925),
('attachedif', 0.0018106925), ('effectly', 0.0018106925)]
Topic #20
[('great', 0.003230138), ('it', 0.00293434), ('works', 0.002192954),
('effectly', 0.0017271835), ('encumbered', 0.0017271835)]
Topic #21
[('attachedif', 0.0017804027), ('encumbered', 0.0017804027), ('beefs',
0.0017804027), ('effectly', 0.0017804027), ('iphone', 0.0017643525)]
Topic #22
[('effectly', 0.0018141348), ('encumbered', 0.0018141348), ('beefs',
0.0018141348), ('attachedif', 0.0018141348), ('it', 0.0017193679)]
Topic #23
[('it', 0.0031893454), ('the', 0.0028190014), ('phone', 0.002493897), ('',
0.0021971595), ('product', 0.0021460203)]
Topic #24
[('case', 0.16700569), ('iphone', 0.11167659), ('protective', 0.036640797),
('tight', 0.03457078), ('mophie', 0.030043568)]
Topic #25
[('', 0.015172238), ('phone', 0.012388529), ('n', 0.011116254), ('signal',
0.00890976), ('cell', 0.0068544853)]
Topic #26
[('phone', 0.0024971743), ('great', 0.0024144321), ('', 0.0019752267), ('use',
0.0019598792), ('the', 0.0018906761)]
Topic #27
[('effectly', 0.0018275362), ('encumbered', 0.0018275362), ('attachedif',
0.0018275362), ('beefs', 0.0018275362), ('the', 0.001713511)]
Topic #28
[('it', 0.0034913733), ('phone', 0.0024893908), ('use', 0.0017142455), ('beefs',
0.0016884308), ('encumbered', 0.0016884308)]
Topic #29
[('vent', 0.002612636), ('it', 0.0025395148), ('the', 0.002288905), ('great',
0.0019763075), ('encumbered', 0.0015931278)]
Topic #30
[('great', 0.0023892939), ('it', 0.0023376571), ('phone', 0.0018390426), ('one',
0.0018043182), ('the', 0.0017931784)]
Topic #31
```

```
[('phone', 0.002264492), ('it', 0.0022069665), ('the', 0.0021205433),
('encumbered', 0.0017184872), ('attachedif', 0.0017184872)]
Topic #32
[('great', 0.0026642669), ('it', 0.0019375858), ('effectly', 0.0018548579),
('attachedif', 0.0018548579), ('encumbered', 0.0018548579)]
Topic #33
[('effectly', 0.001861425), ('encumbered', 0.001861425), ('beefs', 0.001861425),
('attachedif', 0.001861425), ('great', 0.0018430701)]
Topic #34
[('the', 0.021598749), ('heft', 0.018342437), ('use', 0.018209541), ('jacket',
0.015543001), ('doubles', 0.015327671)]
Topic #35
[('it', 0.00260338), ('phone', 0.0024178256), ('the', 0.002093597),
('attachedif', 0.0016619483), ('effectly', 0.0016619483)]
Topic #36
[('phone', 0.0022077782), ('it', 0.0019335755), ('beefs', 0.0017927049),
('encumbered', 0.0017927049), ('effectly', 0.0017927049)]
Topic #37
[('effectly', 0.0018390476), ('beefs', 0.0018390476), ('attachedif',
0.0018390476), ('encumbered', 0.0018390476), ('phone', 0.0017386469)]
Topic #38
[('de', 0.0038594746), ('bateria', 0.0017417548), ('effectly', 0.0017029488),
('attachedif', 0.0017029488), ('encumbered', 0.0017029488)]
Topic #39
[('beefs', 0.001841758), ('effectly', 0.001841758), ('encumbered', 0.001841758),
('attachedif', 0.001841758), ('it', 0.0017980055)]
Topic #40
[('it', 0.0033519578), ('great', 0.0027466628), ('price', 0.001888066),
('works', 0.0017780063), ('one', 0.0017719045)]
Topic #41
[('phone', 0.0023792514), ('it', 0.0019182453), ('attachedif', 0.0017400242),
('encumbered', 0.0017400242), ('beefs', 0.0017400242)]
Topic #42
[('iphone', 0.027835263), ('doubles', 0.020757252), ('chore', 0.019331604),
('dimensions', 0.01581044), ('depth', 0.015141637)]
Topic #43
[('it', 0.007260741), ('phone', 0.0062719393), ('use', 0.0061774747), ('',
0.003845984), ('the', 0.0032171062)]
Topic #44
[('the', 0.0018616312), ('beefs', 0.0018383483), ('effectly', 0.0018383483),
('encumbered', 0.0018383483), ('attachedif', 0.0018383483)]
Topic #45
[('headset', 0.024804452), ('the', 0.014625299), ('sound', 0.012284124),
('great', 0.011144993), ('ear', 0.0105548)]
Topic #46
[('one', 0.002594007), ('it', 0.0017569589), ('beefs', 0.0017410796),
('encumbered', 0.0017410796), ('effectly', 0.0017410796)]
Topic #47
```

```
[('phone', 0.0022102634), ('it', 0.001986995), ('beefs', 0.001685129),
('encumbered', 0.001685129), ('attachedif', 0.001685129)]
Topic #48
[('battery', 0.06990224), ('juice', 0.04597884), ('pack', 0.045224667),
('mophie', 0.040010232), ('iphone', 0.038663134)]
Topic #49
[('good', 0.0018333701), ('beefs', 0.0018285607), ('effectly', 0.0018285607),
('attachedif', 0.0018285607), ('encumbered', 0.0018285607)]
********************************************************************************
********************
```

```
[70]: #model 3
model3 = gensim.models.ldamodel.LdaModel(corpus=corpus, id2word=dictionary,␣
 ↪num_topics=best_topic_num, alpha='auto', eta='auto')
print_topic_words(model3,5)
#########################
# YOUR CODE HERE
#   - Build model for alpha = 'auto' = eta
#   - Print top words
#########################
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

```
Topic #0
[('sound', 0.013278967), ('volume', 0.012559377), ('the', 0.012189787),
('phone', 0.012028249), ('use', 0.011045517)]
Topic #1
[('money', 0.058521584), ('waste', 0.04960528), ('products', 0.023518553),
('dont', 0.010306245), ('substantial', 0.009119033)]
Topic #2
[('headset', 0.031820837), ('ear', 0.023353124), ('bluetooth', 0.014992564),
('it', 0.014128464), ('the', 0.011986177)]
Topic #3
[('film', 0.029222833), ('working', 0.020304462), ('stopped', 0.015819212),
('son', 0.014162653), ('de', 0.01260207)]
Topic #4
[('shed', 0.22354925), ('chore', 0.22019887), ('use', 0.02715837), ('wearing',
0.024033561), ('level', 0.016640743)]
Topic #5
[('phone', 0.04257497), ('cell', 0.018748699), ('would', 0.013342376), ('it',
0.012241454), ('get', 0.011436115)]
Topic #6
[('storm', 0.054942194), ('it', 0.009334485), ('showing', 0.008398844),
('juice', 0.007539291), ('use', 0.007262647)]
Topic #7
[('phone', 0.027792059), ('it', 0.015685914), ('juice', 0.012450253), ('work',
0.008462263), ('one', 0.008399888)]
Topic #8
[('phone', 0.046036765), ('the', 0.013772912), ('camera', 0.0114518525), ('n',
0.01102952), ('', 0.010888911)]
Topic #9
[('belt', 0.028384523), ('clip', 0.02836283), ('the', 0.014365981), ('cord',
0.010936562), ('good', 0.010723004)]
Topic #10
[('lol', 0.02474713), ('genuine', 0.015910745), ('recomend', 0.014509135),
('waited', 0.011355993), ('meter', 0.009810692)]
Topic #11
[('phone', 0.025366161), ('it', 0.011390211), ('use', 0.009699622), ('buy',
0.0092627425), ('card', 0.009210127)]
Topic #12
[('usb', 0.027510127), ('charger', 0.021463517), ('', 0.01515339), ('the',
0.012084402), ('one', 0.011128273)]
Topic #13
[('great', 0.04597189), ('it', 0.030973194), ('good', 0.029945137), ('works',
0.029216949), ('price', 0.02575967)]
Topic #14
[('signal', 0.03516533), ('bars', 0.018625665), ('antenna', 0.014240752),
('earphones', 0.012637621), ('tip', 0.0125449635)]
Topic #15
[('iphones', 0.5862664), ('cables', 0.01829015), ('use', 0.012026037), ('home',
0.010121985), ('broke', 0.0072549586)]
Topic #16
[('', 0.038138792), ('phone', 0.014174078), ('the', 0.013171066), ('one',
0.00842601), ('signal', 0.008189027)]
Topic #17
[('phone', 0.02446869), ('voice', 0.013674058), ('the', 0.012260094), ('use',
0.00979408), ('call', 0.008633997)]
Topic #18
[('cable', 0.03573601), ('phone', 0.026633916), ('it', 0.017331356), ('s',
0.016736388), ('works', 0.01256348)]
Topic #19
[('product', 0.017939027), ('it', 0.016470872), ('one', 0.016411649),
('charger', 0.014637033), ('buy', 0.014131535)]
Topic #20
[('one', 0.014197151), ('break', 0.014108867), ('it', 0.012270119), ('phone',
0.009986212), ('like', 0.008751538)]
Topic #21
[('nokia', 0.030793747), ('data', 0.022572763), ('phone', 0.020818785),
('cable', 0.01903229), ('', 0.01603498)]
Topic #22
[('ultimate', 0.011698772), ('apple', 0.010243023), ('good', 0.0073301857),
('thicker', 0.0073012733), ('battery', 0.006801464)]
Topic #23
[('juice', 0.3583936), ('pack', 0.3438956), ('round', 0.07686898), ('battery',
0.034595083), ('life', 0.015241517)]
Topic #24
[('it', 0.017351778), ('ear', 0.014490085), ('', 0.012208868), ('blackberry',
0.010834078), ('like', 0.009777589)]
Topic #25
[('phone', 0.028079236), ('use', 0.012895064), ('one', 0.01114888), ('device',
0.0106099015), ('great', 0.010230961)]
Topic #26
[('great', 0.029160792), ('product', 0.025315078), ('would', 0.01932183),
('droid', 0.016061643), ('one', 0.0155606745)]
Topic #27
[('regularly', 0.313216), ('iphone', 0.08882063), ('uses', 0.038078107),
('little', 0.023657275), ('cable', 0.018598493)]
Topic #28
[('doubles', 0.052774683), ('pocket', 0.030179802), ('trade', 0.029920762),
('jacket', 0.029908534), ('portability', 0.028051589)]
Topic #29
[('bb', 0.068248324), ('condition', 0.019671543), ('planet', 0.010609675),
('cleaner', 0.009421378), ('voltage', 0.00938864)]
Topic #30
[('heft', 0.6287725), ('jawbone', 0.025385208), ('port', 0.009132168),
('otherwise', 0.008133944), ('starting', 0.0049809506)]
Topic #31
[('otterbox', 0.0253335), ('one', 0.017273126), ('pouch', 0.012286149), ('it',
0.009920439), ('flashlight', 0.008985023)]
Topic #32
[('travelling', 0.17159913), ('normally', 0.064318806), ('uses', 0.05348415),
('everyday', 0.049038142), ('use', 0.029745445)]
Topic #33
[('screen', 0.14446905), ('protector', 0.05060084), ('protectors', 0.038577147),
('stylus', 0.019962719), ('apply', 0.013816104)]
Topic #34
[('', 0.023314359), ('phone', 0.01155663), ('browsing', 0.011328658), ('google',
0.010437282), ('scratched', 0.010219595)]
Topic #35
[('ex', 0.016669037), ('rock', 0.012345247), ('phone', 0.01171119), ('',
0.009823835), ('elsewhere', 0.008045965)]
Topic #36
[('increased', 0.38493663), ('use', 0.014162668), ('the', 0.013866705),
('little', 0.013824341), ('juice', 0.012674928)]
Topic #37
[('iphone', 0.3268576), ('gs', 0.06362314), ('g', 0.05966615), ('rechargeable',
0.03795736), ('use', 0.027230961)]
Topic #38
[('packs', 0.37697828), ('increased', 0.045236986), ('battery', 0.024882467),
('life', 0.017135572), ('use', 0.014412687)]
Topic #39
[('wifi', 0.013735434), ('work', 0.013513186), ('use', 0.011051287), ('gps',
0.010357365), ('boost', 0.009784556)]
Topic #40
[('indicators', 0.080097534), ('obviously', 0.0676703), ('thanks', 0.054032244),
('owners', 0.041647565), ('battery', 0.0350018)]
Topic #41
[('mounting', 0.013297623), ('contacted', 0.012145527), ('lined', 0.011871784),
('wondering', 0.010866231), ('adding', 0.009043039)]
Topic #42
[('bold', 0.015098361), ('', 0.014673899), ('prime', 0.012960954), ('the',
0.012659755), ('battery', 0.010088292)]
Topic #43
[('addon', 0.2423279), ('jeans', 0.24072747), ('increase', 0.109010935),
('iphones', 0.055063732), ('use', 0.020541158)]
Topic #44
[('love', 0.011047282), ('', 0.010851091), ('it', 0.01064881), ('the',
0.010443533), ('attractive', 0.009758497)]
Topic #45
[('depth', 0.14077197), ('indicators', 0.029785287), ('use', 0.026999364),
('alternative', 0.021555623), ('the', 0.016127907)]
Topic #46
[('power', 0.017449094), ('one', 0.015545161), ('cable', 0.014821774), ('plug',
0.0135484), ('cord', 0.010684343)]
Topic #47
[('case', 0.088920385), ('slimmer', 0.04477553), ('tight', 0.040458877), ('air',
0.039977394), ('accessible', 0.03904491)]
Topic #48
[('battery', 0.24768506), ('afford', 0.061992716), ('glad', 0.05548328),
('life', 0.035715297), ('mophie', 0.03396572)]
Topic #49
[('mophie', 0.11125704), ('case', 0.10159461), ('iphone', 0.045071322),
('bottom', 0.0353163), ('protective', 0.028947975)]
********************************************************************************
********************
```

[3pts]1.6.2 Explain how the different alpha and beta values theoretically influence the LDA model. Then describe what you find in the empirical result (e.g difference in topic words and topics)

The higher alpha and beta, the more topics and words in the topic. The lower alpha and beta, the less topics and words in the topic. In the first model when both are low, it has less topics but model 2 has more compare to model 1. Same come to model 3, it has more than model1.

1.8 1.7 LDA on a short text dataset

[10pts]1.7.1 In this part, we will read a dataset from twitter and build a LDA model. On Windows, download and unzip the dataset from this link. Place the downloaded dataset in the same folder as this notebook. Use the first 10,000 lines in the "training.1600000.processed.noemoticon.csv" file.

[34]:
```
!wget http://cs.stanford.edu/people/alecmgo/trainingandtestdata.zip # Linux and␣
↪OSX only
!unzip trainingandtestdata.zip # Linux and OSX only
```

```
zsh:1: command not found: wget
unzip:  cannot find or open trainingandtestdata.zip, trainingandtestdata.zip.zip
or trainingandtestdata.zip.ZIP.
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[35]:
```
!head -n 10000 training.1600000.processed.noemoticon.csv > twitter.csv # Linux␣
↪and OSX only
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[36]:
```
def read_twitter(fname):
    """ Read the given dataset into list and clean stop words.

    Args:
        fname (string): filename of Twitter Dataset

    Returns:
        list of list of words: we view each document as a list, including a␣
↪list of all words
    """

    twitter = []
    with open(fname,encoding="utf-8") as f:
        for line in f:
            tweet = f.readline().split(",")[5]

            stopWords = set(stopwords.words('english'))
            tweet = tweet.split(" ")
            clean_tweet = []
            punctuationRegex = r'\W+|\d+'
            for w in tweet:
                w = w.lower()
                if w not in stopWords:
```

```
                w = re.sub(punctuationRegex, ' ', w)
                if len(w) >= 2:
                    clean_tweet.append(w)

        twitter.append(clean_tweet)




        ########################
        # YOUR CLEANING CODE HERE
        ########################
    return twitter
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[37]:
```
%%time
twitter = read_twitter('twitter.csv')
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

```
CPU times: user 408 ms, sys: 53.1 ms, total: 461 ms
Wall time: 460 ms
```

[38]:
```
twitter_dictionary = gensim.corpora.Dictionary(twitter) # TODO: build dictionary
twitter_corpus = [twitter_dictionary.doc2bow(text) for text in twitter] # TODO:␣
↪build corpus for model
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[39]:
```
%%time
########################
# YOUR CODE HERE
#    - Build model
#    - Print top words
t_model = gensim.models.ldamodel.LdaModel(corpus=twitter_corpus,␣
↪id2word=twitter_dictionary, num_topics=5)
print_topic_words(t_model,5)
########################
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

```
Topic #0
[(' is', 0.009836194), ('i m', 0.007289609), ('like', 0.006140389), ('work',
0.0052561504), ('still', 0.004966049)]
Topic #1
[(' i', 0.010554), ('i m', 0.010124678), ('think', 0.005957), ('like',
0.0052914284), ('want', 0.0049571455)]
Topic #2
[('get', 0.007398168), ('night', 0.0073218886), ('home', 0.0059108683), ('today
', 0.005856536), ('last', 0.0058111106)]
Topic #3
[(' i', 0.012295593), ('work', 0.011885202), ('go', 0.009249411), ('got',
0.0088013075), ('miss', 0.0075263306)]
Topic #4
[(' i', 0.009341358), ('get', 0.009163124), ('go', 0.008256394), ('need',
0.007435016), ('work', 0.00628568)]
********************************************************************************
********************
CPU times: user 1.29 s, sys: 4.52 ms, total: 1.29 s
Wall time: 1.29 s
```

1.9 1.8 LDA visualization

[**10pts**]**1.8.1** We will now visualize the LDA output using pyLDAvis. PyLDAVis shows the following:

1. The distances between topics, as a map in 2-D space.
2. The variance in the topic-word distribution, as the size of a circle in this map.
3. The most “salient” terms in each topic.

```
[40]: sentences = read_dataset("reviews_Cell_Phones_and_Accessories_5.json.gz")[:
      ↪1000] # CHANGE TO YOUR DATASET
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
```

```
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[41]:
```
%%time
import pyLDAvis.gensim
import pyLDAvis.gensim_models
```

```
CPU times: user 4.04 ms, sys: 2.88 ms, total: 6.92 ms
Wall time: 8.23 ms
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[42]:
```
%%time
data = None
########################
# YOUR CODE HERE
#   - Initalize pyLDAvis with your model
#   - Make sure to use a subset of the sentences in the dataset
#     if your pyLDAvis call in the cell below is taking too long
########################
sentences_dictionary = gensim.corpora.Dictionary(sentences) # TODO: build␣
 ↪dictionary
sentences_corpus = [sentences_dictionary.doc2bow(text) for text in sentences] #␣
 ↪TODO: build corpus for model
sentences_model = gensim.models.ldamodel.LdaModel(corpus=sentences_corpus,␣
 ↪id2word=sentences_dictionary, num_topics=5) # TODO: build model
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

```
CPU times: user 1.05 s, sys: 11.6 ms, total: 1.06 s
Wall time: 1.06 s
```

[43]:
```
pyLDAvis.enable_notebook()
data = pyLDAvis.gensim_models.prepare(sentences_model, sentences_corpus,␣
 ↪sentences_dictionary)
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
```

```
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/pyLDAvis/_prepare.py:243: FutureWarning: In a future version of pandas
all arguments of DataFrame.drop except for the argument 'labels' will be
keyword-only.
  default_term_info = default_term_info.sort_values(
```

[44]:
```
pyLDAvis.enable_notebook()
pyLDAvis.display(data)
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[44]:
```
<IPython.core.display.HTML object>
```

2 2.word2Vec [40pts]

In this problem, we use Amazon Review Dataset to perform Word2Vec and Doc2Vec to extract insights relevant for e-commerce business. For this question, download and use the dataset here.

2.1 2.1 Data Cleaning

The following code reads the data from a GZIP file.

[22]:
```python
# A function to read the zipped data at a specfic path
#
# How to use:
# PATH = "/path/to/file"
# for line in parse(PATH):
#   do something with line
#
def parse(path):
    g = gzip.open(path, 'r')
    for l in g:
        yield eval(l)
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

We will now read the data and preprocess it using the following steps:

1. Remove stopwords
2. Lower-case all words
3. Remove words with less than 2 characters
4. Remove punctuation
5. Split each sentence into a list of words

And finally extract 10000 reviews.

[23]:
```
# A function to clean a single line of text
def clean_line(line):
    """ Clean stopwords and punction for each line

    Args:
        line (string): one line in file

    Returns:
        list(str): a list of all words in the sentence
    """
    punctuationRegex = r'\W+|\d+'
    stopWords = set(stopwords.words('english'))
    line = line.split(" ")
    filtered_content = []

    for word in line:
        if word not in stopWords:
            if len(word) > 2:
                filtered_content.append(word.lower())
    filtered_content = [re.sub(punctuationRegex, '',word) for word in␣
 ↪filtered_content]
    return filtered_content

def read_dataset(fname):
    """ Read the 100000 lines in given dataset into list and clean stop words.

    Args:
        fname (string): filename of Amazon Review Dataset

    Returns:
        list of list of words: we view each document as a list, including a␣
 ↪list of all words
    """
    count = 0
    exp_dataset = []
    for review in parse(fname):
        line = review["reviewText"]
        new_line = clean_line(line)
        exp_dataset.append(new_line)
        count += 1
        if count > 100000:
            break
    return exp_dataset
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[24]:
```
%%time
r = read_dataset("reviews_Electronics_5.json.gz")
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

```
CPU times: user 23.5 s, sys: 1.58 s, total: 25.1 s
Wall time: 25.2 s
```

2.2 2.2 Build a doc2vec model

[**3pts**]**2.2.1** In this question, first we will build a Word2Vec model using ginsim using size=300, min_count=40, win- dow=10, negative=10, max_vocab_size=10000. Train the model for 30 epochs.

[25]:
```
from gensim.models import Word2Vec
# YOUR CODE HERE
w2v = Word2Vec(r, vector_size=300, min_count=40, window=10, negative=10,␣
 ↪max_vocab_size=10000, workers=4, epochs=30)
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[**2pts**]**2.2.2** Use model.wv.doesnt_match to find a word in [“Canon”,“Nikon”,“junk”] that does not belong.

[49]:
```
# YOUR CODE HERE
words = ["Canon", "Nikon", "junk"]
dmwords = w2v.wv.doesnt_match(words)
print(f"The word that does not belong is: {dmwords}")
```

```
The word that does not belong is: junk

/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[**3pts**]**2.2.3** Come up with 3 other word lists and apply the above function. Explain your observation.

[50]:
```
words = ["camera", "IPhone", "bottle"]
dmwords = w2v.wv.doesnt_match(words)
print(f"The word that does not belong is: {dmwords}")
```

```
The word that does not belong is: camera

/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

Bottle is the only non electronic item in the list.

[**2pts**]**2.2.4** What are some tasks in e-commerce that can be solved with this simple function?

In e-commerce, the above w2v model and doesnt_match function can be used for item categorization and similiar product recommendations to increase sales.

2.3 2.3 Build a doc2vec model

[**15 pts**] **2.3.1** Each review is marked by other customers as "helpful" or not. The "helpful: [a, b]" item in each review is (a) the number of people who marked the review as helpful, and (b) the total number of people who have marked the review as helpful or unhelpful. The "helpfulness" score of a review can be calculated as a/b. Define a "helpful" review as one with helpfulness score >= 0.8. Given a review that is only slightly helpful, could we find textually similar reviews but have higher helpfulness? Build Doc2Vec model with gensim on review data. Use product ID "B00006I5WJ" and ReviewerID with "A14453U0KFWF31" as an example, find top 5 helpful reviews of the same product with similarity score above 0.8.

[5]:
```
import os
def read_reviewers_data(fname, min_count=0):
    '''
    Save all reviews into their own product asin files.
    Make sure you have 'product' folder when you run this answer.
    In each file, you can choose your own log structure. In this answer, log␣
 ↪strucutre is like
        "reviewText"\t"reviewerID"\t"helpful"
    Args:
        fname: dataset file path
        min_count: minimum number of reviews of a product
    Returns:
        none
    '''
    if not os.path.isdir('product'):
        os.makedirs('product')
    asin_list = []
    tmp_list = []
    last_asin = ""
    j = 0
    for i in parse(fname):
        if last_asin != i['asin']:
            if len(tmp_list) > min_count:
                f = open("product/" + last_asin+".txt", 'w')
                for one in tmp_list:
                    f.write(one)
                f.close()
            tmp_list = []
            last_asin = i['asin']
        tmp_list.append(i["reviewText"] + '\t' + i["reviewerID"] +
                        '\t' + handle_helpful(i["helpful"]) + "\n")
        j += 1
        if j > 100000:
            break

def handle_helpful(helpful):
    '''
    Helper function for helpful_score calculate
    Args:
        helpful: list. The first element is the number of people think this is␣
 ↪helpful. The second element
            is the total number of people evaluate this comment
    Returns:
        String: number represent helpfulness
    '''
    if helpful[1] != 0:
        helpfulness = 1.0 * helpful[0] / helpful[1]
        return str(helpfulness)
    else:
        return str(0)
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
```

```
  and should_run_async(code)
```

[52]:
```python
read_reviewers_data("reviews_Electronics_5.json.gz")
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[6]:
```python
class TaggedReviewDocument(object):
    '''
    This class could save all products and review information in its dictionary␣
 ↪and generate iter for TaggedDocument
        which could used for Doc2Vec model
    '''
    def __init__(self, dirname):
        self.dirname = dirname
        self.helpfulness = {}  # key:reviewerID value:helpfulness
        self.product = {}      # key:asin value:reviewerID
        self.asin = []

    def __iter__(self):
        for filename in os.listdir(self.dirname):
            asin_code = filename[:-4] #delete ".txt"
            self.product[asin_code] = []
            self.asin.append(asin_code)
            for line in enumerate(open(self.dirname + "/" + filename)):
                line_content = line[1].split("\t")
                self.product[asin_code].append(line_content[1])
                self.helpfulness[line_content[1]] = float(line_content[2])
                yield TaggedDocument(clean_line(line_content[0]),␣
 ↪[line_content[1], line_content[2]])
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[54]:
```python
documents = TaggedReviewDocument("product")
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
```

```
  and should_run_async(code)
```

[58]:
```
%%time
from gensim.models.doc2vec import TaggedDocument, Doc2Vec
# YOUR CODE HERE
d2v = Doc2Vec(documents, window = 3, workers = 2,vector_size =3, min_count=5 )
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

```
CPU times: user 5min 28s, sys: 28.9 s, total: 5min 57s
Wall time: 5min 2s
```

2.4 Find similar reviews

[59]:
```
def find_similar_reviews(asin,reviewer_id):
    '''
    If one review is similar to the specefic review and it is much helpful,␣
↪save it to a list
    Args:
        asin: product asin
        reviewer_id: the specific review
    Returns:
        list of reviewer id
    '''
    result = []
    ########################
    # YOUR CODE HERE
    ########################

    vector = d2v.infer_vector([asin, reviewer_id])
    product_vecs = d2v.docvecs
    similarity = d2v.docvecs.most_similar([vector], topn=len(product_vecs))

    r_id_list = []

    for i in documents.product[asin]:
        if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
            r_id_list.append(i)

    h_list = []
    for i in r_id_list:
        h_list.append((i,documents.helpfulness[i]))
    result = sorted(h_list, key=lambda x: x[1], reverse=True)[:5]

    return result
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[60]:
```
print(find_similar_reviews("B00006I5WJ", "A14453U0KFWF31"))
```

```
[('A2IXUSSQ4KJEAX', 1.0), ('A1ZJSYEQQDIDAA', 1.0), ('AGITH5SMFTBOB', 1.0),
('A5C7KD02LS69I', 1.0), ('A1MSY5U64K1Y3D', 1.0)]
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
<ipython-input-59-3a4a219117bf>:16: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  product_vecs = d2v.docvecs
<ipython-input-59-3a4a219117bf>:17: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  similarity = d2v.docvecs.most_similar([vector], topn=len(product_vecs))
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
```

```
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
```

```
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
```

```
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
```

```
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
```

```
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
```

```
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
<ipython-input-59-3a4a219117bf>:22: DeprecationWarning: Call to deprecated
`docvecs` (The `docvecs` property has been renamed `dv`.).
  if d2v.docvecs.similarity(reviewer_id,i) > 0.8 and i != reviewer_id:
```

2.5 2.4 Build a doc2vec model using product descriptions

[10pts]2.4.1 Use product descriptions (located in meta data here) to build a Doc2Vec model. When building the doc2vec model, use vector_size=100, window=15, min_count=5, max_vocab_size=1000, and train it for 1 epoch.

[7]:
```python
def read_product_description(fname):
    '''
    Load all product descriptions
    Args:
        fname: dataset file path
    Returns:
        dict: key is asin, value is description content
    '''
    result = {}
    for i in parse(fname):
        try:
            if "Camera & Photo" in i["categories"][0]:
                result[i["asin"]]=i["description"]
        except:
            continue
    return result
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[8]:
```
class TaggedDescriptionDocument(object):
    '''
    This class could save all products and review information in its dictionary␣
 ↪and generate iter for TaggedDocument
        which could used for Doc2Vec model
    '''
    def __init__(self, descriptondict):
        self.descriptondict = descriptondict


    def __iter__(self):
        for asin in self.descriptondict:
            for content in self.descriptondict[asin]:
                yield TaggedDocument(clean_line(content), [asin])
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[10]:
```
description_dict = read_product_description("meta_Electronics.json.gz")
des_documents = TaggedDescriptionDocument(description_dict)
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[12]:
```
# Build a doc2vec model
from gensim.models.doc2vec import TaggedDocument, Doc2Vec

model = Doc2Vec(vector_size=100, window=15, min_count=5, max_vocab_size=1000,␣
 ↪epochs=1)
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
```

```
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[13]:
```
model.build_vocab(des_documents)
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[ ]:
```
model.train(des_documents, total_examples=model.corpus_count, epochs=model.
 ↪epochs)
```

[ ]:
```
pip install
```

[**5pts**]**2.4.2** Find the most similar product for Canon EOS 5D (asin:B0007Y791C) not made by Canon.

[21]:
```
most_similar_products = model.dv.most_similar(positive=[model.
 ↪dv['B0007Y791C']],negative = 'canon', topn=5)

for asin in most_similar_products:
    print(f"ASIN: {asin} ,Description: {des_documents.descriptondict[asin[0]]}")
```

```
/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-21-10faca351423> in <module>
----> 1 most_similar_products = model.dv.most_similar(positive=[model.
 ↪dv['B0007Y791C']],negative = 'canon', topn=5)
      2 
      3 for asin in most_similar_products:
      4     print(f"ASIN: {asin} ,Description: {des_documents.
 ↪descriptondict[asin[0]]}")

~/opt/miniconda3/lib/python3.9/site-packages/gensim/models/keyedvectors.py in␣
 ↪most_similar(self, positive, negative, topn, clip_start, clip_end,␣
 ↪restrict_vocab, indexer)
    839 
    840         # compute the weighted average of all keys
```

```
--> 841         mean = self.get_mean_vector(keys, weight, pre_normalize=True,␣
 ↪post_normalize=True, ignore_missing=False)
    842         all_keys = [
    843             self.get_index(key) for key in keys if isinstance(key,␣
 ↪_KEY_TYPES) and self.has_index_for(key)

~/opt/miniconda3/lib/python3.9/site-packages/gensim/models/keyedvectors.py in␣
 ↪get_mean_vector(self, keys, weights, pre_normalize, post_normalize,␣
 ↪ignore_missing)
    516                 total_weight += abs(weights[idx])
    517             elif not ignore_missing:
--> 518                 raise KeyError(f"Key '{key}' not present in vocabulary"
    519
    520         if total_weight > 0:

KeyError: "Key 'canon' not present in vocabulary"
```

[19]:
```
most_similar_products = model.dv.most_similar(positive=[model.
 ↪dv['B0007Y791C']], topn=5)

for asin in most_similar_products:
    print(f"ASIN: {asin} ,Description: {des_documents.descriptondict[asin[0]]}")
```

```
ASIN: ('B0007Y791C', 1.0) ,Description: World's smallest and lightest full-frame
digital SLR as of August 2005 the sensor operates without a conversion
factor.New larger 2.5-inch LCD screen can be viewed even at extreme angles of up
to 170 degrees.New larger 2.5-inch LCD screen can be viewed even at extreme
angles of up to 170 degrees.Consecutive shooting allows the capture of 3.0
frames per second for up to 60 consecutive JPEG or 17 RAW frames in a burst.
ASIN: ('B001BP1YEE', 0.4226471781730652) ,Description: Why Choose? Why Settle?
Introducing the New Kata 3N1!Why choose between a backpack and a sling when you
simply can pick up one of the new Kata 3N1 bags and get the best features of
each type of bag? These unique bags provide ultimate freedom designed with three
carrying options in one bag: right-handed sling, left-handed sling and backpack.
Quick access to a camera has never been faster with a sling that offers a quick
release and one-handed bag opening for camera access. Left-handed photographers
can celebrate as they discover the first photo bag that can be configured as a
left-handed sling. When not in a situation that requires quick camera access,
convert the 3N1 into an ergonomic and comfortable backpack. All these product
features come in a package that includes the unique combination of protection,
light weight and style that has become Kata&#x92;s calling card.Kata 3NI
HighlightsFlexible Configurations to Ensure Maximum UseUse both as a sling and
backpackAs a photographer, you choose a sling for the ability to gain quick
access to your camera in shooting situations. A backpack is key for carrying
your equipment long distances without straining your back. Finally, there is a
bag that offers the best of both worlds in one sleek design. By including two
padded straps that can be easily hidden, you decide which product configuration
```

works best for you. Large sturdy strap buckles make it easy to quickly convert your 3N1 from a right-handed sling to a left-handed sling or a backpack. A waist strap is included to provide maximum comfort to your back when using the 3N1 in backpack mode. You no longer have to decide which camera bag to bring with you into the field. Load up your 3N1 and you are set for all photgraphic situations.Uses Thermal Shield Technology to ensure maximum protectionSuperior Protection for Your DSLRSuperior protection is key to the Kata design philosophy. The creation of TST (Thermal Shield Technology) is a physical expression of this philosophy. TST is modeled to create protective zones where your camera will most likely be impacted. Impact force is distributed to ensure that your camera remains safe and always ready for that important scene or shot. The 3N1 is also designed with reinforced corners to ensure that when in sling mode, the bag holds its shape to ensure your camera does not fall out.Fast, One Handed Camera Access with Quick Release BuckleWhen in sling mode, quick access to your camera is provided via a quick release clip with single pull opening of the camera compartment door. Pinch the clip and pull and the zippers easily slide to offer one-handed access to your camera. Easy access means you are always ready when the action happens and need to capture that perfect shot on the fly.Rain Cover Provides All-Weather ProtectionIncluded with each 3N1 model is a sturdy all-weather rain cover. You never know when Mother Nature will throw you a curve and you need to be prepared. This rain cover wraps around your entire sling / backpack ensuring that your valuable gear is protected. The bright color also reflects sunlight to keep your rucksack and sensitive digital equipment cooler in hot locations.Optional InserTrolley (KT VG-DTS) for Easy TravelingBoth the 3N1-20 and 3N1-30 have a sleeve for use with the Kata InserTrolley (Model KT VG-DTS) or most luggage trolleys. This allows for easy transport of your camera. While connected to the trolley, your camera remains fully accessible. The Kata Insertrolley can be folded for easy storage when not in use. Kata's patented extendable wheel system provides better stabilization when wheeling your 3N1 bag.Ensure you find the right fit for your camera
ASIN: ('B007C5LRGC', 0.41753923892974854) ,Description: MOD classic straps are designer/boutique straps, for both SLR cameras, both digital and film. Their unique designs stand out from the crowd of plain, black, synthetic looking products that dominate the strap category. Classic straps are cut from the best fabrics available, and are backed with a soft "minky" backing. Strong connector ends on the Classic straps make these straps a must have for the fashionable photographer who needs strength and quality. All MOD products are manufactured in the USA, deep in the heart of Texas.
ASIN: ('B000MSZQQI', 0.41364675760269165) ,Description: Specifically designed for Panasonic cameras, the Rock-DVX combines the finest degree of zoom control with the type of wide-sweep rocker favored by broadcast operators. The result is an easy-to-use, effective means of achieving smooth, sustainable zooms. Details include a compact, intuitive design; a pressure-sensitive, side-by-side rocker switch; and a low-profile record/pause button. Offering a greater range of motion than most zoom controls, the Rock-DVX is compatible with the Panasonic DVX100, DVX100A, DVX100B, HVX200, DVC30, DVC60, and DVC80 cameras.FeaturesMaterial: ABS plasticCompatibility: DVX100, DVX100A, DVX100B, HVX200, DVC30, DVC60, and DVC80Dimensions: 2 by 2.75 inchesMax clamp diameter:

```
1.25 inchesCord length: 40 inchesWeight: 0.25 poundsWarranty: 2 years parts and
labor
ASIN: ('B00C6PHV9A', 0.3896889388561249) ,Description: The SumacLife Camera
Sleeve lets you carry & protect your digital camera in style. The case features
a stylish micro-suede exterior for classy good looks & a soft shock absorbent
neoprene bubble lined interior to cushion & protect your digital camera against
impact, dents & scratches. This sleeve is compact but yet roomy enough to
accommodate a wide range of small accessories such memory cards, batteries & USB
cables. The dual zipper action opening provides you quick & easy access to your
camera, so you won't miss out on another picture perfect moment. (External
Dimensions 4.75 x 3.5 x 1 // Internal Dimensions 4.2 x 3 x 1)

/Users/ryanli/opt/miniconda3/lib/python3.9/site-
packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
not call `transform_cell` automatically in the future. Please pass the result to
`transformed_cell` argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[ ]:
```
# Based on the above result, the most similar product is B001BP1YEE
```