INFS 770 – Advanced Data Mining Applications

Assignment 3: *Text data preprocessing and topic modeling*

**Due 4/5/2021 + one-week grace period**

Dataset: In this assignment, we will use a dataset I created by copying/pasting a number of news articles posted on google news. Please go to D2L -> Content -> Assignments -> Assignment 3 to download the data file “tp\_dataset.zip”. Unzip the file and put the folder “tp\_dataset” in your iPython notebook word directory.

**Things to submit:** Please develop and submit an iPython notebook titled 770\_21\_a3\_*yourlastname*. In the notebook, you need to finish the tasks below, where (C) indicates that you need to write code for the task, (O) indicates that you need to show output, and (A) that you need to type your answers using Markdown text. Please use “run all” to run your code before you submit so that your iPython notebook will show the outputs of your code. You will lose 1 point if you do not “run all”. You probably want to copy and modify the code from the iPython notebook “topic modeling” posted on D2L.

In your iPython notebook, at the beginning of each cell, you need to indicate which task the cell is about. For example, in the cell related to task 1, you should first type “# Task 1: Import data”. If you do not clearly label the cells, you will lose 1-2 points (out of 18 points). Whenever you see “print” in the questions, you need write print statement to print the intended outputs.

**Tasks:**

T1. Import data. You need to write code to read filenames from the tp\_dataset folder, and given a filename, you need to use f.read() method to read each file as a string. You add these strings to a list and call this list “docs”. Please print the list. The pseudo code for doing this task is below. You are required to write python code to implement the pseudo code. (C)(O)

docs = []

import os

for each filename in the folder “tp\_dateset”: # you want to use the method os.listdir()

given a filename, open the file as f: # to open a file, you use the path (folder + “/”+ filename) to the file.

use f.read() to read the file as a string

append the string to the list docs

T2: Write code to tokenize the docs using the TF-IDF vectorizer (You need to copy and modify the corresponding code in the topic modeling.ipynb file I used in lecture). In this task you need to 1) lowercase each word, 2) remove punctuations, 3) remove numbers, 4) do a lemmatization, 5) remove stop words, 6) **remove words that appear in only one or two documents**, and 7) **remove words that appears in over 50% of documents**. When you do TFIDF, you often need to ignore words that too common or too rare. The words that appear too often in a corpus is called “corpus-specific stop-words”. When you do topic modeling, it is extremely important to set an appropriate threshold for removing the corpus-specific stop words. Here we just set it to be 50%. In real practice, you need to tune this parameter. In this task, you set norm to be “l2” when you do TF-IDF. Please print the features (i.e., vectorizer.vocabulary\_) extracted based on TF-IDF. (C)(O)

T3. Create a textbox and use your own language to discuss what TF-IDF means. (A)

T4. Write code to convert the vectorized data to a gensim corpus object. Print id2word. (C)(O)

T5. Write code to compute coherence scores for different numbers of topics including 2, 3, 4, and 5. Since our dataset is quite small, when you run LDA, please set number of passes to 50 –You need to run a sufficient number of iterations to make sure LDA converges, but if your dataset is very large, you don’t want to set this parameter to be too large; otherwise, it will take too much time to run your code. Given each number, you need to run LDA 30 times and take the average of the coherence scores you obtained in each run. Print the average coherence score for each number of topics including 2,3,4, and 5. (C)(O)

T6. You usually cannot rely on the coherence scores to determine the optional number of topics. Next, you want to further compare the topic 3 numbers of topics with the highest coherence scores. For each of them, you need to write to code to print the topics and also print the document/topic matrix. For instance, if 3 is among the top 3, you need to run LDA (with num\_topics=3, random\_state=10, passes=50). You need to print the topics as shown in Fig 1 and the document/topic matrix as shown in Fig 2 (Please note these are just examples to show the format of the outputs. You are not required to have the same ones). Please print the topics and the document/topic matrix for each of the top 3 numbers with the highest average coherence scores. (C)(O)

[(0, ‘0.008\*”biden” + 0.008\*”infrastructure” + 0.007\*”tax” + 0.007\*”house” + 0.007\*”spend” + 0.006\*”trillion” + 0.006\*”white” + 0.006\*”proposal” + 0.006\*”next” + 0.006\*”republicans”’), (1, ‘0.011\*”apple” + 0.011\*”homepod” + 0.007\*”device” + 0.007\*”mini” + 0.007\*”might” + 0.007\*”code” + 0.007\*”could” + 0.007\*”speakers” + 0.006\*”speaker” + 0.006\*”launch”’), (2, ‘0.010\*”vaccine” + 0.007\*”astrazeneca” + 0.007\*”data” + 0.007\*”efficacy” + 0.007\*”dose” + 0.006\*”countries” + 0.006\*”age” + 0.006\*”participants” + 0.006\*”find” + 0.006\*”result”’)]

Fig 1. Topics as tuples

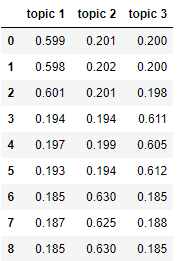


Fig. 2 Document/topic matrix (all the float number are rounded to 3 decimal points)

T7. Create a textbox and tell me among the top 3 numbers with the highest average coherence scores, which one gives you the best topic modeling results. You need to provide your justifications. Please note there’s no correct or wrong answer. You will get the points for this task as long as your justifications are reasonable. (A)

T8: Run truncated SVD(set n\_components to be the optimal number of topics you found in T7) and print topics and the topic/document table. (C)(O)