Ling 413 Term Project: Topic Models for Healthcare

Jacob Fuehne
jfuehne2
Issued: 11/16/2020
Due:

Task#1: 12/04/2020 (by midnight): hard deadline!!!

Tasks #2a and #2b: 12/18/2020 (by midnight): hard deadline!!!

Total points: 100

The goal of this assignment is to give you hands on experience with topic models for a real application. Thus, you will implement and experiment with some topic models, and then write a short report about your experiences and findings (in a file called analysis.txt). As you know, topic models capture the most important topics in a text corpus.

Note: This project with this healthcare dataset is NOT publishable (due to data compliance issues, as explained in class)!!!

Scenario:

You are a newly employed text analyst at a data analytics company. Congratulations!
Your first project is for a new client, a (new) healthcare company. This company is in the process of collecting a very large dataset of patient comments from all their clinical locations throughout the country and is interested in getting insights on patient experience from this collection (i.e., insights into patient comments about doctors, nurses, clinics, healthcare services, etc.). They will deliver the data in 6 months.

However, since this is the first project on healthcare for your company, your boss would like you to start working on it ASAP and is asking you to perform some explorative analysis of a similar dataset (of your choice) and prepare a preliminary report on what kinds of insights can be generated from such data.

For this assignment, you will sit down and decide on the design decisions you need to make to solve the task, then test your models on a relevant healthcare dataset, and finally write a report on the results. Specifically, to help you in this process, you are asked to work on a series of tasks which describe the exploratory analytics process for this application.

1) **<u>Task#1: Corpus collection and Corpus Descriptive analysis</u>** [20 points]

First, you have to find and collect a dataset that is similar to the one the healthcare client will provide later. The task is rather challenging since such secondary data are difficult to find due to compliance issues. However, after considerable research, you manage to find a freely-available patient review dataset from RateMD (http://ratemds.com), one of the most popular platforms for physician reviews in the United States (Note: we obtained IRB approval for using this dataset for this purpose).

RateMD Data Description:
Founded in 2004, RateMD has the largest number of user-submitted reviews with narratives by a large margin. In RateMD, every doctor is given an ID which uniquely specifies a doctor's profile information: name, gender, location, specialization. The website also provides the overall average rating for a doctor (a real number between 1 and 5, with 1 being the worst rating) and the review text. For this project you decided to download and collect a corpus of 20,421 patient reviews together with some meta-data (as shown below).

For each doctor, the entry consists of a line of 4 tab-separated fields: [Dr's Name; Gender; Location; Specialization] followed by [Overall rating; Review text], one per line, depending how many people rated this doctor.

For example, Dr. Thomas here has been reviewed by two people, so there are two lines [Overall rating; Review text]:

Dr. Shirley A. Thomas   Female   Fishers, IN   Gynecologist (OBGYN)
Overall rating: 4.75   Best doctor in the world. She not only is beyond knowledgeable from her 40 years of practice but she cares about us.. A lot. She doesn't need the money, she does this because it gives her joy delivering babies. I would fly from Cali to see her, that's how much I trust her.
Overall rating: 3.75   Dr Thomas stays up to date with new information, which makes me feel confident each time I see her. I feel like she is very through and great about making sure that all of my questions are answered, and I trust her advice, which adds …

You will work on the following problems:

<u>Problem#1:</u>
Do a descriptive analysis of your corpus and provide (using the table below): the distribution of reviews per gender and sentiment (show both count and percent coverage). Here the sentiment can be only positive or negative -- determined by mapping the overall ratings of at most 3 into negative (i.e., [1,3]) and the rest into positive (i.e., (3,5]). E.g., the overall ratings of the example above maps into positive sentiments.

Gender	Sentiment (count and %)		Total (count and %)
	Positive (count, % of gender, % of total)	Negative (count, % of gender, % of total)	
Female	2953 61.44% 14.46%	1853 38.55% 9.07%	4806 23.53%
Male	10616 67.98% 51.98%	4999 32.01% 24.47%	15615 76.45%

Also provide and comment on the size of the reviews in the corpus: i.e., the length of the smallest review and of the largest review, as well as the average length of the reviews in the corpus. Here the length of a review is defined as the number of tokens (i.e., any sequence of characters separated by space and/or beginning/end of review).

ANSWER:
(assuming we exclude the reviews that were blank)
The longest review is 899 tokens. The shortest review is 1 token. And the average length of reviews was 65 tokens long.

Problem#2:
Analyze and explain why this dataset from RateMD is a valid, relevant corpus for your project.
For this, you are referred to the corpus design principles discussed in class (i.e., language variety of interest, sampling, representativeness, etc.). In particular, consider the following helping questions and fill in the entries in the table below.

Note: Your reference corpus is the corpus to be provided by the healthcare company.

No.	Questions	RateMD corpus	Healthcare company's corpus (i.e., reference corpus)
1	What is the language variety of the corpus (i.e., genre)?	Reviews written by patients that have been seen by doctors on RateMD	Reviews written by patients of the company's clinics
2	What is the size of the corpus?	20,421 reviews	500,000 reviews
3	What meta-data is provided with the reviews?	Doctor's name, gender, office location, type of doctor, review sentiment	Doctor's name, gender, clinic location; review sentiment
4	What socio-demographic information is provided about the patients who wrote the reviews?	N/A	Gender, age, economic and educational status
5	Is the corpus balanced along the meta-data dimensions considered? (look only at sentiment and gender)	No (the dimensions are not uniformly distributed. There are	No (the dimensions are not uniformly

		more than 3x the number of reviews for male doctors. There is also a slight bias between the % number of positive and negative reviews for male and female. It's a natural distribution)	distributed; they exhibit a natural distribution)

Compare the answers to the questions in the table above (3^{rd} and 4^{th} columns) and use this comparison to identify and comment on one important disadvantage of using RateMD as a good, relevant corpus for this project (i.e., 'good, relevant' here means how similar it is to the corpus the healthcare company will provide in the future).
Hint: Think of who is writing the reviews for RateMD. How does this compare with the healthcare company's data (i.e., who wrote of the reviews).

ANSWER:
Since the goal of the project in the scenario is to get insights into the patient experience, the fact that the RateMD corpus has no demographic information about patients is a big disadvantage. With these anonymous reviews, there is no way to get insight into how people of different backgrounds are experiencing their visits to different doctors. If there was a certain demographic of patients that was disproportionately having negative experiences, it would be impossible to find out with this dataset. And if that were true, it would be something that the hypothetical company in the scenario would be very interested to know about.

2) **Task#2a: Exploratory Analysis of Corpus with LDA** [40 points]

You have to write a python program that takes as input the corpus (i.e., your RateMD corpus) along with a given number of topics k, and generates these topics. For this task you will experiment with LDA (Latent Dirichlet Allocation).

Specifically, as explained in class, in this procedure you have to consider a number of steps:

Step 1: Clean the corpus
Your text corpus has to be cleaned before you give it as input to the topic model.
Thus, you have to decide on what kind of cleaning steps you need to consider. Start by considering the cleaning procedures suggested for the Wikipedia case study done in class.

- Lowercase and Punctuation Removal
- Stop word removal
- Stemming vs. Lemmatization

- Other data cleaning steps
    - removal of File attachment,
    - removal of Image attachments, URLs, Infobox, XML labels, etc.
    - Spelling correction
    - <u>Your own stop list</u>
    - <u>Filter extremes</u>
        - remove any words that were in at most i documents (i.e., i = 5) and any word that appeared in more than x% of documents (i.e., x = 60)
    - <u>Other word removal</u>

Which of these steps make sense here? Any other steps necessary but not listed there?
Moreover, you will experiment with lemmatization as well, so you have to run the LDA model *with* and *without* lemmatization (as shown below). And you have to calculate the runtime for the lemmatization. How long did it take?

ANSWER:
It makes sense to convert tokens to lowercase, to remove punctuation, remove stop words, remove words with length less than 3, and to filter out the extremes. Specifically, I choose to filter out the words that have occurred in less than 4 articles and filter out the words that have occurred in more than 40% of the articles.

<u>Step 2:</u> Create the dictionary
After you have cleaned your corpus, you will create the term dictionary. How large is your dictionary?
ANSWER:
`8162 unique tokens`
<u>Step 3:</u> Convert the list of documents in your corpus into Document-Term Matrix using the dictionary prepared at Step 2 (again, a term is a word).

<u>Step 4:</u> Run the LDA model on the document-term matrix

Here you have to run LDA with two sets of parameters:

1) Set 1: number of topics (k = 10), number of passes (pass = 20), and number of iterations (iterations = 2000).
2) Set 2: number of topics (k = 20), number of passes (pass = 20), and number of iterations (iterations = 2000).

For each LDA run you have to calculate the runtime. How long did it take to run with Set 1 and how long with Set2?

ANSWER:

In [127]:
```
#Task 2a - Run the LDA model on the document term matrix
#Set1:
# LDAset1 = Lda.load("../assignment/LDAset1model-NoLemma/model")
# if model is lost, create a new file with this code
start = time.process_time()
LDAset1 = Lda(mycorpus, num_topics=10,id2word=dictionary,passes=20,iterations=2000)
set1TimeElapsed = time.process_time()-start
print("Elapsed time for set1 in seconds:", set1TimeElapsed)

#save that model with this code:
LDAset1modelNoLemma = "../assignment/LDAset1model-NoLemma/model"
LDAset1.save(LDAset1modelNoLemma)
```

```
Elapsed time for set1 in seconds: 125.734375
```

In [128]:
```
#Task 2a - Run the LDA model on the document term matrix
#Set2:
# LDAset2 = Lda.load("../assignment/LDAset2model-NoLemma/model")
# if model is lost, create a new file with this code
start = time.process_time()
LDAset2 = Lda(mycorpus, num_topics=20,passes=20,id2word=dictionary, iterations=2000)
set2TimeElapsed = time.process_time()-start
print("Elapsed time for set2:", set2TimeElapsed)

#save that model with this code:
LDAset2modelNoLemma = "../assignment/LDAset2model-NoLemma/model"
LDAset2.save(LDAset2modelNoLemma)
```

```
Elapsed time for set2: 271.796875
```

It took 125.73 seconds to run with Set 1 and it took 271.80 seconds to run with Set 2. The preprocessing step took about 2.05 seconds.

Step 5: For each of the k topics, print the top 10 words

After following the LDA procedure outlined in steps 1-5 above, work on the following problems:

Problem#1:
Here you run the LDA models (with Set1 and Set2, respectively) without lemmatization.
Place the topics in two tables (showing the top 10 words per topic as done in class: Table 2.1 shows the Topics 1-5, and Table 2.2. shows the Topics 6-10). Then analyze the goodness of your topics – meaning, manually label each topic with a topic word or phrase identifying its theme. Could you find a label for each of your topics? Which ones were easy to label and which were noisier (and thus, not easy to label)?
SET 1 - NO LEMMA

Topic1 – waiting time	Topic2 – bedside manners	Topic3 – Obstetrics	Topic4 – Las Vegas surgeons	Topic5 – Fuzzy, semantic category not clear
office staff time wait appointment doctor get see never room	doctor recommend manner great bedside excellent knowledgeable would caring highly	daughter son child baby pregnancy delivered first old birth hospital	pain surgery back years vegas two months without severe las	doctor patients patient like care medical doctors time dont know

Topic6 – Chronic issues	Topic7 – Highly praised reviews	Topic8 – Time dedication	Topic9 – Critical procedures	Topic10 – Fuzzy, semantic category not clear
years knee issues tooth health doc top care helped skin	doctor best ever years one ive life doctors hes seen	time staff always great feel questions doctor takes friendly office	surgery surgeon would recommend staff procedure went great cancer experience	told would said doctor went back never didnt could see

SET 2 – NO LEMMA

Topic1 – Fuzzy, semantic category not clear	Topic2 – Fuzzy, semantic category not clear	Topic3 – waiting time	Topic4 – comfort	Topic5 – Fuzzy, semantic category not clear
doctor like time dont get know good doesnt see want	surgery pain surgeon back procedure went performed breast still done	wait time appointment room waiting minutes hour long exam waited	feel staff great made like comfortable really make makes office	son reviews rather could year one old school read may
Topic6 – Obstetrics	Topic7 – Life Saving	Topic8 – Second opinion	Topic9 – Orthopedics/plastic surgery	Topic10 – Fuzzy, semantic category not clear
child baby pregnancy children daughter delivered pregnant first	life husband cancer years saved heart daughter mother	second opinion eye straight correct clinic eyes arrogant	patel hes tooth many able years nose helped	care patient health primary years issues specialist lack

love kids	god thank	get fix	teeth top	quality uncaring
Topic11 – Fuzzy, semantic category not clear	Topic12 – bedside manners	Topic13 – Time dedication	Topic14 – Phone calls	Topic15 – Highly praised reviews
told doctor went said would problem didnt back never got	manner bedside good cold weight manners root pleasant efficient loss	questions time answer staff always answered answers concerns professional treatment	office staff rude doctor front never phone service calls ever	best doctor ever patients one cares years doctors ive hes
Topic16 – Highly praised reviews 2?	Topic17 – Phone calls 2?	Topic18 – Insurance coverage	Topic19 – Very negative reviews	Topic20 – Highly praised reviews 3?
recommend would highly anyone doctor great excellent family staff wonderful	called office call would told get even said back see	insurance medical test tests results patients also treatment medication health	doctor rude ever worst even side away experience horrible stay	doctor staff time always great caring knowledgeable helpful takes excellent

ANSWER:
For Set1, I was able to come up with a label for 8 out of 10 of the topics. I found waiting time, bedside manners, obstetrics, Time dedication, and Highly praised reviews easy to label, and I found chronic issues, critical procedures, and Las Vegas surgeons to be more difficult to label. As mentioned, I was unable to find a clear label for Topic5 and Topic10.

For Set2, I was able to come up with a label for 15 out of 20 of the topics. I also found myself with several topics (16,17,20) that I would consider to be semantic duplicates of other topics. With labelling, I found waiting time, comfort, Obstetrics, Life Saving, Second opinion, bedside manners, Time dedication, Phone calls, very negative reviews, and Insurance coverage to be easy to label. I think that the various highly praised reviews topics (15, 16, 20) had clear cohesion, but it was not clear why they formed seperate topics. The same would apply to the two phone calls labels (14,17). As mentioned, I was unable to find a clear label for Topics 1, 2, 5, 10, and 11. I found Topic9 Orthopedics/plastic surgery to be difficult to label because I thought it originally might have been something related to dentistry, but after looking at instances of Patel and thinking about this topic carefully, I came to the conclusion that topic9 is likely relating to reconstructive/appearance based procedures that orthopedic surgeons and plastic surgeons would perform. I believe Patel was included here because there is an orthopedic surgeon named Patel with many reviews where they are mentioned by name.

<u>Problem#2:</u>

Follow the instructions for Problem#1 above, but with lemmatization this time. Consider both noun and verb lemmatization (with WordNetLemmatizer from NLTK). Don't forget to calculate the runtime for the lemmatization step.

ANSWER:

In [145]:
```
#Task 2a - Run the LDA model on the document term matrix
#Set1 Lemma:
# LDAset1Lemma = Lda.load("../assignment/Lemma/LDAset1model-Lemma/model")
# if model is lost, create a new file with this code
start = time.process_time()
LDAset1Lemma = Lda(mycorpusLemma, num_topics=10,id2word=dictionaryLemma,passes=20,iterations=2000)
set1TimeElapsed = time.process_time()-start
print("Elapsed time for set1 lemma in seconds:", set1TimeElapsed)

#save that model with this code:
LDAset1modelLemma = "../assignment/Lemma/LDAset1model-Lemma/model"
LDAset1Lemma.save(LDAset1modelLemma)
```

```
Elapsed time for set1 lemma in seconds: 153.21875
```

In [146]:
```
#Task 2a - Run the LDA model on the document term matrix
#Set2 Lemma:
# LDAset2Lemma = Lda.load("../assignment/Lemma/LDAset2model-Lemma/model")
# if model is lost, create a new file with this code
start = time.process_time()
LDAset2Lemma = Lda(mycorpusLemma, num_topics=20,id2word=dictionaryLemma,passes=20,iterations=2000)
set2TimeElapsed = time.process_time()-start
print("Elapsed time for set2 lemma in seconds:", set2TimeElapsed)

#save that model with this code:
LDAset2modelLemma = "../assignment/Lemma/LDAset2model-Lemma/model"
LDAset2Lemma.save(LDAset2modelLemma)
```

```
Elapsed time for set2 lemma in seconds: 254.0625
```

Repeating my answers for problem1, but with using lemma, I kept all of the same filters for my preprocessing and simply added lemmatizing for noun and verbs. I found that the new dictionary was 7041 unique tokens and that the preprocessing step with lemmatization is 67.59 seconds (compared to 2.05 seconds without lemmatization). Training the LDA model for set1 with the lemmatized dictionary took 153.22 seconds (compared to 125.73 seconds without) and it took 254.06 seconds for set2 (compared to 271.80 seconds without). This leads me to conclude that the preprocessing time considerably increases with lemmatization, but the time to train the model may increase or decrease with lemmatization, depending on the number of topics.

SET 1 - LEMMA

Topic1 – Highly positive reviews	Topic2 – Highly positive reviews 2?	Topic3 – Time dedication	Topic4 – Fuzzy, semantic category not clear	Topic5 – Critical procedures
doctor care patient year best time see always take ever	great life staff patel save best thank doctor doc know	recommend staff doctor would great highly time question answer helpful	doctor like say know would get want make never dont	surgery procedure surgeon recommend would result perform cancer breast first
Topic6 – Phone calls	Topic7 – Waiting time	Topic8 – Chronic issues	Topic9 – Insurance coverage	Topic10 – Prescriptions and diagnosing
staff office call rude doctor patient phone front service nurse	wait time appointment see hour get room minute doctor office	pain surgery back year glyman dentist knee work life problem	tell test call insurance say would doctor get see take	problem medication treatment prescribe diagnose condition side help give medical

SET 2 - LEMMA

Topic1 – Waiting time	Topic2 – Highly positive reviews	Topic3 – bedside manners	Topic4 – Office staff	Topic5 – Responsiveness
wait time office doctor see get appointment staff hour patient	care recommend patient doctor year would excellent physician highly family	manner make feel bedside child baby great comfortable deliver pregnancy	staff service office medical wife poor competent provide lack receive	call day tell get back take would see week give
Topic6 – Insurance coverag e	Topic7 – Fuzzy, semantic category not clear	Topic8 – Very negative reviews	Topic9 – Highly positive reviews 2?	Topic10 – Chronic issues
insurance test doctor pay medical result bill refuse send visit	say tell would didnt want know never get back think	son practice rude stay staff away else terrible wouldnt worst	staff great doctor helpful friendly knowledgeable office recommend always time	tooth arrogant hip allergy disorder root anxiety perfect open headache
Topic11 – Fuzzy, semantic category not clear	Topic12 – Prescriptions and treatment	Topic13 – Long term joint pain	Topic14 – Time dedication	Topic15 – Fuzzy, semantic category not clear
daughter hospital clinic accept brain old completely dad tumor incompetent	medication find patient try help get listen work treatment prescribe	pain year back knee life surgery ago help injury month	time question answer take concern make feel ask listen like	surgery surgeon would breast result recommend great recovery fix hand
Topic16 – Highly positive re views 3?	Topic17 – Serious condition diagnosis	Topic18 – Fuzzy, semantic category not clear	Topic19 – Stressful exams/procedures	Topic20 – Highly positive revie ws 4?

doctor life patient care like know treat people help save	treatment condition diagnose opinion diagnosis treat cancer option disease second	husband tell year take one see look find month still	procedure perform exam check use prompt nervous felt prior examination	doctor best ever see year one ive know love doc

ANSWER CONT.:
For set1 lemma, I was able to come up with a label for 9 out of 10 of the topics. I found time dedication, critical procedures, phone calls, waiting time, chronic issues, insurance coverage, and prescriptions and diagnosing to be easy to label. As had happened previously in problem1, I found that there were multiple topics that seemed to have a theme of highly positive reviews. And just as before, these topics had clear cohesion, but they may not have needed to have been seperate topics. I was unable to come up with a label for Topic4.

For set2 Lemma, I was able to come up with a label for 17 out of 20 of the topics. I also found myself with several topics (9,16,20) that I would consider to be semantic duplicates of other topics. I found that waiting time, bedside manners, office staff, responsiveness, insurance coverage, chronic issues, prescriptions and treatments, long term joint pain, time dedication, serious condition diagnosis, and stressful exams/procedures to be easy to label. As with the No Lemma set two, I think that the various highly praised reviews topics (2,9,16,20) had clear cohesion, but it was probably not semantically necessary for them to be in different topics. I was unable to come up with a label for topics 7, 11, and 15.

Problem#3:
Compare your program's outputs with and without lemmatization for k=10 and also for k=20 (Sets 1 and 2). Which of these settings generates better topics? Is lemmatization worth doing? For this, compare the goodness of the topics with and without lemmatization and across parameter sets. Analyze and explain.

ANSWER:
Comparing the results of set 1 and 2, with and without lemmatization, I think it is clear to see that lemmatization will provide better cohesion among topics, and it will produce less ambigious topic groupings. The best overall approach would appear to be Set 1 with 10 topics and applying lemmatization, as this results in the smallest percentage of "fuzzy" labels. Considering that in Set 1 without lemmatization, I came up with a label of "las vegas surgeons" (chosen after skimming through the dataset because there was a seemingly disproportionate high number of reviews for surgeons in the Las Vegas area, and it seemed that the LDA model had grouped these together), I think it is also reasonable to say that the topics generated with lemmatization seemed to be more useful/generalizable.

3) **Task#2b: Exploratory Analysis of Corpus with ccLDA** [40 points]

Problem#1:

Split the RateMD corpus into two collections of reviews along the gender dimension: collection C1 will contain reviews about female doctors, and C2 reviews about male doctors. Further, split each collection in two sub-collections on the sentiment dimension: e.g., C1.1 (positive reviews about female doctors) and C1.2 (negative reviews about female doctors), etc.

Replicate the data preparation step in Task#2a above (i.e., data cleaning) and run the model with ccLDA instead of LDA. You have to make sure that the data you give as input to ccLDA is in the format required by ccLDA. For this, you have to read the readme file and run the topic model with two sets of parameters:

1) Set1: 10 topics and 2000 iterations
2) Set2: 20 topics and 2000 iterations

Calculate the runtime of ccLDA in each setting.

What do you notice? Is the ccLDA runtime faster than the LDA running time in Python (across similar sets of parameters)?
Show the 10 topics (top 10 words per topic) and the 20 topics, respectively. Can you label them? How many do you think are noisier?

ANSWER:
I found that the ccLDA model takes 333.66 seconds for Set1 and 508 seconds for Set2, thus making ccLDA considerably slower than LDA with lemmatization (153.22 and 254.06 seconds respectively). However, for topic cohesion, ccLDA is clearly superior for the 10 topic list, as I found that I was able to label 9 out of 10 of the topics for set1. I think this was less true of the 20 topic list, as I found myself confused when labelling groups. I only achieved 16 out of 20 for set2. While not listed because there are too many top ten lists, being able to cross-reference the top ten lists of the different collections was very beneficial to deciphering uncertain topic labels, or to make others more specific (ie, highly positive reviews can clearly be seen as sentiment based reviews by cross referencing the negative collections top ten lists). I do think, however, that the 1 fuzzy topic label left for set1 was slightly less semantically coherent than it was for LDA. I think this was also true for the 4 fuzzy labels of set2. It would seem to me that ccLDA has the effect of making clear labels more clear, while making some fuzzy labels even worse.

SET 1 – ccLDA

Topic1 – Wait times	Topic2 – Diagnosing	Topic3 – Chronic issues	Topic4 – Time dedication	Topic5 – Family doctor/pediatricians?
wait call appointment get time see office	patient medical treatment physician condition diagnosis issue	back pain problem month year week would	time best question like feel answer never	year best see take husband life son

hour minute day	practice health problem	get find could	make talk listen	hospital could never
Topic6 – Fuzzy, semantic catego ry not clear	Topic7 – Office staff	Topic8 – Bedside manners	Topic9 – Appearance related procedures	Topic10 – Sentiment focused reviews
get say tell want dont try didnt think know ask	staff best office doctor patient service work visit good need	best would doctor recommend manner bedside anyone child baby first	best would look procedure experience make one result work even	doctor best patient care ever see know one help find

SET 2 - ccLDA

Topic1 – Fuzzy, semantic category not clear	Topic2 – Office staff	Topic3 – Fuzzy, semantic category not clear	Topic4 – Family doctor	Topic5 – Chronic issues
one find review say give enough believe read different may	staff best office nurse great extremely also nice helpful always	get see first could day one visit come make find	patient medical care year physician practice health family treat many	pain back year help give walk better problem physical month
Topic6 – Time dedication	Topic7 – Fuzzy, semantic category not clear	Topic8 – Sentiment based reviews	Topic9 – Sentiment based reviews 2?	Topic10 – Obstetrics
time question answer take concern ask explain	know dont like want people think doesnt	doctor ever best one life ive see	would best recommend anyone doctor need experience	child daughter first baby son old husband

best give seem	help someone see	meet year find	refer extremely friend	pregnancy best since
Topic11 – Fuzzy, semantic category not clear	Topic12 – Bedside manners	Topic13 – Comfort/ability to feel at ease	Topic14 – Wait times	Topic15 – Insurance
tell say didnt even ask take get come see need	best patient manner care bedside doctor listen knowledge seem interested	make like feel time felt patient really never listen much	wait time appointment hour see room minute long get late	office insurance visit service get pay need work try make
Topic16 – Critical procedures	Topic17 – Phone service	Topic18 – Diagnosing	Topic19 – Sentiment based reviews 3?	Topic20 – Appearance related procedures
surgery hospital would day perform cancer surgeon procedure put two	call get office back phone never return day appointment even	treatment problem condition diagnose medication test diagnosis symptom give also	best doctor good care really take doc job nice happy	best look procedure eye result face year work breast want

Here is where you can download ccLDA: http://michaeljpaul.com/downloads/mftm.php
Here is a link to the ccLDA paper that might help you in this process. You do not have to fully understand the model details, but you should know how to run it, what kind of input it accepts and what kind of output it generates:
http://www.aclweb.org/anthology/D09-1146

<u>Extra-credit problem</u>: [15 points]
[Due date: 12/18/2020 (by midnight): hard deadline!!!]

a) Repeat **Task#2a** above, but this time, instead of giving LDA a bag of words as input, train it on a tf-idf representation (i.e., the new corpus representation: tf-idf real-valued weights). (Note: use **ntc** as the SMART scheme).

Show the 10 topics and the 20 topics, respectively (for each, show the top 10 words per topic). Can you label them? How many do you think are noisier? Are the output topics better than the ones you obtained at Task#2a? Why/why not? Explain.

b) In your opinion, is tf-idf useful for topic modeling? Explain.
If yes, in which steps of the process (i.e., for what purpose)? Elaborate.

<u>Project Deliverables:</u>

- write a README file including a detailed description of the functionality of your code, and complete instructions on how to run them;
- make sure you include your name (code and README file);
- make sure all your programs run correctly (Jupyter notebook file(s) or a python-program.py);
- include the answers to all the questions at each task (i.e., your report) in in a file analysis.txt (i.e.: submit the file analysis.txt with answers to Task#1 no later than Dec. 4th; and the same file, with the answers to all tasks by Dec. 18th).