COMPSCI 4NL3 Homework 2: Corpus Analysis

Winter 2026
Due: February 2nd

1 Overview

The goal of this assignment is to gain experience analyzing a corpus of your choosing using methods covered in the class.

2 Requirements

You should perform the following steps:

1. Collect a dataset
2. Convert to bag-of-words format
3. Compute Naïve Bayes probabilities for terms in each category
4. Run topic modeling
5. Experiment with the effects of text normalization on the above

See below for details.

2.1 Dataset

You should find a dataset that you are interested in learning more about. Your dataset must be organized into two or more distinct categories. The available metadata will give you ideas. Examples include:

1. Books from Project Gutenberg from different authors, genres, or time periods
2. Subtitles from Open Subtitles or The Internet Movie Script Database with different genre, TV series, time period
3. Posts from multiple different subreddits on Reddit.com
4. Congressional hearings from different years, presidential terms, committees, or political parties
5. A Text Classification Dataset from Hugging Face datasets hub (quality may vary - check thoroughly before using)
6. Any other dataset you are interested in. Choose something you will enjoy learning about!

You should aim to have at least 100 documents per category (though you may have many more). Remember that *document* is a concept that can refer to texts of various types and sizes. A book could be broken up into chapters, treating each chapter as a document. A TV serious could be split into episodes.

The dataset should not already be processed. This means you should not use a dataset that already is tokenized into a bag-of-words.

The dataset can be in any language, but you should be able to describe/analyze what you find in English for the report, and you may need to research and apply specialized preprocessing steps on your own if you work with a non-English language (as in the previous assignment).

If there is a dataset you really want to study that does not meet the criteria above, please post on teams and we will work with you to see if there is a way to make the dataset fit the requirements.

2.2 Bag-of-words

After you have your corpus, you should process it to convert it to a BoW format. You do not need to provide any specific output to prove that you did this, but it will be required for the subsequent steps. You may use your code from the previous assignment to preprocess the data. Otherwise, you may write new preprocessing code or use a third party library if you understand what the library is doing. If you want to store your data as a document-term matrix and are having trouble with the amount of memory this requires, you may find the SciPy Sparse Matrix Representations useful. These methods do not require you to store all of the 0s in your matrix and can lead to a much smaller memory footprint.

2.3 Naïve Bayes

Next, you will use a Naïve Bayes model of probability to compute the probabilities of words belonging to the categories you defined earlier. The goal is to produce lists of words that represent what is unique about each category compared to the other(s). You don't need to split the training and testing sets and experiment with different approaches to maximize the F1 score for this assignment. In this assignment, we won't actually use the classifier part of the Naïve Bayes model. We will use this type of model instead as a way to estimate probabilities and associate words with classes.

You should use the data that you have collected to compute the probabilities of each word belonging to a given category, $P(w|c)$ as you would in a Naïve Bayes model. To find out which words are most associated with a category (c), we would want to know how much higher this probability is compared to $P(w|c_o)$ for the other categories $c_o \in C_o$ where C_o is the set of other categories you are considering ($c \notin C_o$). To determine this we can compute the **log likelihood ratio**:

$$llr(w, c) = log(\frac{P(w|c)}{P(w|C_o)}) = log(P(w|c)) - log(P(w|C_o)) \tag{1}$$

This gives us a measure of how much more likely it is that we observe the word given the document belongs to class c than it is to observe the same word given a document that is not in class c. If you only have 2 classes, $P(w|C_o)$ is equal to $P(w|c_o)$ where c_o is the single class other than c and $llr(w, c_o) = -llr(w, c)$. Otherwise, more generally it is given by:

$$P(w|C_o) = \frac{\sum_{c_o \in C_o} count(w, c_o)}{\sum_{c_o \in C_o} \sum_{w' \in V} count(w'|c_o)}, \tag{2}$$

where C_o is the set of classes that are not c and V is the vocabulary for the entire corpus. You should also use add-one smoothing for all of the probabilities.

Given this, you should produce a list of the top 10 words sorted by their log-likelihood ratios for each class $c \in C$, the set of all classes (categories from your selected dataset). When computing probabilities, you may use count data, binary, or TF-IDF transformed data.

2.4 Topic Modeling

In this step, you will run Latent Dirichlet Allocation using all of the documents in your corpus. You may choose the number of topics that you feel is most appropriate and gives the results that either look most reasonable to you or optimize a metric like coherence. You do not need to implement LDA yourself, and should use a 3rd party library like gensim to do topic modeling and may use libraries like pyLDAvis to help present your results. You may also use jsLDA for both topic modeling and generating interesting visualizations (but you may not use the example corpus they present).

You should present your topics (or a selection of the topics you think are most interesting/useful) in a table where the first column contains your own manually assigned label for the topic (e.g. *school*), and the subsequent columns contain the top 25 terms for that topic, sorted by their probability of belonging to that topic, along with their probabilities (e.g. *homework* (0.02), *class* (0.01), etc.).

Finally, for each category, find the average distribution of all topics for documents in that category and report the top 3-5 topics for each category. You can determine this by taking the topic distribution for each document in a given category and averaging the probabilities for each topic across all documents in the category.

2.5 Experimentation

After your code is written and you are able to complete all the steps, try at least one additional variation of text normalization (e.g., using the options available from the previous homework) and one additional variation of the bag-of-words representation (counts, binary, TF-IDF). Take note of how the results from all steps changed and try to decide which of the configurations gives you the most insightful results.

Your overall goal with this assignment should be to produce meaningful insight and analysis for your dataset, comparing and contrasting the documents from each of your categories. This does not mean your results need to be surprising. Many results may make perfect sense to you, but you should be able to see that simply quantifying these results across a large set of documents is meaningful.

3 Deliverables

You should submit the code you used as well as a PDF report documenting your approach and findings.

3.1 Code

Your code should be written in Python and should include enough documentation/instructions for someone else to be able to run. You must submit your code via Avenue. Your code should include a simple README file that explains the files/directories, and how to set up and run the code. Your should run using Python 3.14 and make sure that your file read/write operations use UTF-8 (this makes it much easier for us to run your code).

You are welcome to use code snippets from examples in class, things you find online, or from AI code generation tools. Just make sure to give proper attribution to code you did not write. Follow the syllabus instructions for how to report the use of AI tools. However, you may not copy code that does the entire assignment (e.g. someone who did this assignment in a previous semester).

You do not have to automate the entire process from dataset collection to generation of the figures and tables for the report. You may, for instance, generate the figure using another tool like Libre Office Calc. Your code should showcase how you did things like preprocessing, computing probabilities, and topic modeling.

3.2 Report

Your report should have the following structure. To get full credit for the assignment, please make sure you include everything below.

1. **Dataset.** (3 points) Describe the dataset you chose and why you think it is interesting or useful to analyze. How did you collect the data, choose the categories, and split it into *documents*? Include tables and figures to show the size of your dataset, e.g. a table with the number of documents and the average number of tokens per document, broken down by category.

2. **Methodology.** (3 points) Describe the steps that you performed and what informed your decisions. For example, if you decided not to lowercase the text because it gave better results at a later stage, include that decision and your reasoning for doing that. This should include details of your preprocessing steps, e.g. lowercasing, stemming, not just saying "each document was preprocessed". Describe the kind of analysis you performed. For any steps that you did not implement yourself (e.g. topic modeling), you should mention which package/library was used.

3. **Results and Analysis.** (4 points) Present your results as formatted tables and figures. You should have at least one table or figure for each of 2.3, 2.4, and 2.5. This must include at least the results of the required steps, but may also include any interesting findings you came across (e.g. results of topic modeling with and without a given preprocessing step that made a difference in the quality of the results). For each table and figure, include a description of your main takeaways.

4. **Discussion.** (5 points) Include two subsections in the discussion. The first should talk about what you learned about your dataset. Imagine that you are describing what your results showed, at a high level, to a friend who does not have any NLP experience but is interested in the corpus that you chose. The second subsection should cover what lessons you personally learned during the completion of the assignment. You might write about how you found and processed the data, preprocessing effects on downstream analysis, topic modeling results, limitations of your approaches, or other interesting aspects that were new to you.

There is no page limit for the report. In a single-column format, similar to the one that this document is written in, around 3-5 pages is the expected length (including figures). You should include the PDF titled homework2_report.pdf and the files needed to run your code.

The report should be well-organized and professionally presented. Please avoid things like blurry, low-resolution or poorly-cropped screenshots, submitting one long paragraph with no subsections or formatting, or copy pasting long strings from your program output with no formatting applied.

This assignment is out of 25 points. The report is worth 15 points. Your code is worth 10 points. Your code should implement each of the steps outlined above and should be reasonably well documented, such that the instructors for this course can quickly read and understand the code.

4 Double Check Your Files

Immediately after submitting your assignment, go back to the submission page and download your work. Make sure that you can open it, that the files open properly, and that you have submitted the proper files. You must make sure that your PDF can be opened and is not corrupted, encrypted, or otherwise damaged. When it comes time for grading, if you submitted the wrong files, there is no way to prove that you completed the work on time and you will receive a zero for whatever aspect of the assignment is missing.

5 Help

Please use the Teams channel to ask questions about the assignment when you need guidance or pointers on this homework. You are free to discuss your approach and ideas with classmates, but should not share code or reuse data. You may use generative AI tools if you find them helpful, but please clearly document how they were used in the report and follow the guidelines in the syllabus for what you **must** include when using generative AI. If you use generative AI and do not report it, you may receive a 0 for the assignment. You take full responsibility for the deliverables you submit.