Assessment: Getting Started with Textual Data

Instructions

To earn a micro-badge for this workshop, write code for/answer the prompts on the next page. You should do your work in a notebook environment (like Jupyter); include all code, code outputs, and short answers directly in your notebook. Submit this notebook to GradPathways by exporting it to an `html` file (in Jupyter Notebooks, you can do this by going to `File > Save Page As` in your browser).

Links

- GradPathways Badge
- Event page
- Workshop reader

Environment and File Setup

For this assessment, you will be cleaning, mining, and modeling a corpus of texts. We intend this to reinforce the materials the workshops sessions cover and to serve as an occasion with which you can practice text mining for your own research. To wit: you are invited to use your *own* corpus of text files to complete the assessment; by the time you've finished, you'll have built a foundation for text mining in a research project.

If you don't want to use your own corpus, or if you don't have one ready to hand, we've also provided one. Under `data/novels/` you'll find 150 plaintext files of English-language novels. You can find a corresponding file manifest at `data/manifest.csv`, which you may use as a reference when doing your own work. If you *do* want to use your own corpus, we suggest using at least 100 documents. But it also depends: 100 tweets won't get you very far, whereas 100 research papers likely will. Try to aim for a corpus that has **at least 100,000 total words**.

In addition to the input directory and file manifest, we have created an empty directory, `data/output/`, which you should use to store any data that you generate while working. You are free to make other directories as you see fit.

While we ask that you do this work in a notebook environment, you may use a local environment on your own computer or Google Colab. Regardless of which environment you use, you will need to make sure that you have all required packages installed. The requirements file for these packages is under `data/requirements.txt`.

The directory structure for this assessment is:

```
requirements.txt                A list of required packages
text_mining_assessment.pdf      These instructions
data/                           The data directory
|-- manifest.csv                A file manifest
|-- novels/                     Plaintext files of 150 novels
|-- output/                     An empty directory for code outputs
`-- voyant_stoplist.txt         A list of stopwords
```

Rubric

Readers at GradPathways will be looking for a few things in this assessment.

1. Working code: were you able to successfully implement code for each prompt?
2. Understanding the code: can you explain what your code does and why you implemented it?
3. Supporting examples and materials: have you used graphs and other results to produce evidence for your findings?
4. Critical reflection: do your short answers provide context (conceptual, domain-specific, etc.) for your findings and observations? Can you use your results to reason about your corpus, or even provide preliminary hypotheses?

Prompts

1. Cleaning
    a. Select a text from your corpus and load it into your notebook. Regardless of which corpus you've chosen, the texts are going to need to be cleaned so you can mine them for information. So, using this single text, define a series of functions to clean your corpus (it may take a few tries to get it cleaned to your liking).
    b. In a few sentences, explain how you developed these functions. What did you want to accomplish with each one? What considerations did you need to take into account? What challenges did you encounter?
    c. Once you have developed your cleaning workflow to your liking, clean all the texts in the corpus. If you'd like, save the cleaned texts to `data/output/`.
2. Document Info
    a. Now that your corpus is clean, it's time to format it into a usable data structure for text mining. Convert your texts into a document-term matrix of **raw** counts.
    b. With that done, you can begin surveying your corpus:
        - What is the longest text in the corpus? The shortest?
        - What is the mean type-token ratio? Would you say this corpus is lexically diverse? Why or why not?
    c. Construct graphs as appropriate to help you in your exploratory work.
3. Document Similarity
    a. Now, convert your texts into a document-term matrix of **tf-idf** scores and then produce a cosine similarity matrix from these scores. Remember that these scores will help us identify similarities across the texts your corpus.
    b. To see this in action, select three texts. Identify the most- and least-similar texts for each one.
    c. In a few sentences, briefly reflect on these (dis)similarities. If you're familiar with the texts you've chosen, are you surprised by what you've found? Sample a few terms novels and compare their respective tf-idf scores. Do you notice any patterns?
    d. One thing we haven't had you do is lemmatize the texts. If you were to lemmatize the texts, how would you expect that process to affect the similarity scores?
4. Topic Modeling
    a. It's time to topic model! Produce a preliminary topic model of your cleaned corpus and explore the results. After doing so, explain whether you think the number of topics you've set is appropriate.
    b. Remember that topic modeling is often an iterative process, so return to the model initialization and find what you think is an optimal number of topics for your corpus. Adjust hyperparameters as needed. If you do adjust them, explain your reasoning for doing so (this is likely going to involve you drawing on your own domain knowledge—embrace that context!).
    c. Visualize your topic model using `LDAvis`. Using it, try to assign labels to at least three of the topics in your model (that is, what you think these topics are 'about'). This is often an important step for making your results interpretable and shareable with other researchers. If you were to share your model, how would explain your reasoning for these labels? Use examples from the corpus as needed.