COMPSCI 4NL3 Homework 1: Counting Tokens

Winter 2026
Due: January 16th

1 Overview

This is a first assignment to get you started dealing with text data. The goal is to experiment with text normalization and processing text files. The code you write for this assignment might be useful for future assignments.

2 Requirements

You should perform the following steps:

1. Download a text file (or corpus) containing natural language
2. Write code to normalize the text and count the tokens to produce a sorted list
3. Visualize the counts
4. Describe your findings

See below for details.

2.1 Text File

There are many sources of text data available online. You may want to look, for instance, at Project Gutenberg and look for "Plain Text" versions. There are more examples of text corpora here, but you can use other text you find online. It must be plain text and in a format that is meant for humans to read (not already processed).

You may try to run your code on different files, but at least one file selected should contain natural language and at least 50,000 total tokens. When you process the file, you will get the total tokens, so you will know the length. You can concatenate multiple files. You may choose a file in a non-English language, but in this case you will need to look into how to properly normalize text for that language, as some normalization challenges vary by language.

2.2 Normalize and Count

Next, write a program that takes a plain text file as input, containing natural language, and produces a list of each normalized type (unique token) and the number of times it appears in the file, sorted from largest to smallest.

The user should be able to type:

```
$ python normalize_text.py myfile.txt (<your-options>)
```

Where your-options are command line options for different preprocessing steps. This should produce output like this:

```
the 5000
dog 2930
ran 492
...
```

Options that the user can control must include the following. Use the **exact names in parentheses** for the flag name (so it's easier for us to run).

1. lowercasing (lowercase)
2. stemming (stem)
3. lemmatization (lemmatize)
4. removal of stopwords (stopwords)
5. One option you come up with (myopt)

For the last option, look through your data and the results from other preprocessing steps. What looks like it will be useful in generating a more meaningful list? Your program should be able to accept multiple options at the same time, for example:

```
$ python normalize_text.py myfile.txt -lemmatize -stopwords -myopt
```

2.3 Visualization

Lastly, you will take your sorted list of normalized token counts and produce a bar plot where the x-axis is the rank of the word in your list and the y-axis is the frequency. You notice that the results are easier to view when using a log-scale for the x and/or y axis. You may visualize more than one input file if you like, and compare the visualizations. Make sure your figure includes a title, labeled axes, and everything is readable.

2.4 Findings

Check your results. What do you notice about the words at the top of your list? Other than being "common words", do you notice any other kinds of properties that they share? Now check the other end of your list. What do you notice about these words?

After creating your figure(s), check out this Wikipedia page, especially the introduction and subsection titled "Word frequencies in natural languages". How do your results compare to the ones you see there? Why do you think that they are similar or different?

What does this imply about the impact of removing stopwords like "the" on the total number of tokens in a corpus, compared to removing regular content words (e.g., nouns and verbs)? Answer these questions in your report.

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 A2L as a zip. Your code should include a simple README file that explains the files/directories, and how to set up and run the code. Your code 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). However, you MAY NOT use NLTK's tokenize functions (or tokenize functions from other libraries).

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. The code should, at minimum, showcase how you produced your list of counts.

3.2 Report

Your report should have the following structure:

1. **Data.** (3 points) Describe the file you selected and what the source was, as well as anything you found interesting about it.
2. **Methodology.** (3 points) Describe, at a high level, the approach you took to writing your software and generating your figures. What are the options that the user can select when performing the text normalization steps. What is the additional option that you added beyond the defined requirements? Why do you think this is a useful option?
3. **Sample Output.** (4 points) You should show the first and last 25 words and their counts from your program's output, as well as the figures that you generated. The sample output should correspond to an input file that meets the requirements.
4. **Discussion.** (5 points) Your discussion should contain two parts. First, include 1-2 paragraphs to address the questions from §2.4. Next, write 1-2 paragraphs about what you learned during the completion of the assignment. You might write about selecting your data, unexpected issues that you ran into and how you resolved them, any new skills or software libraries you learned in order to complete the assignment, or anything else that you learned. Please be specific (i.e., do not write vague statements "The assignment was great, I learned so much about NLP").

There is no page limit for the report. In a single-column format, similar to the one that this document is written in, around 1-2 pages is the expected length (including figures). You should title the PDF homework1_report_MACID.pdf, with your given MACID username.

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.