Corpus Analysis Exercise
Methods in Computational Linguistics

Franziska Weeber

November 18, 2025

This exercise is based on material from Dr. Eva Maria Vecchi.

Introduction

The main goal of this lab session is for you to get some experience analyzing and comparing corpora. The exercises described later will require you to process several provided corpora: Read each corpus, process it as instructed, compute the described statistics, and then compare these different statistics across the provided corpora.

The participation in this lab session is voluntary: you do not have to submit any result. If you want to get feedback, please ask questions during the session. Feel free to work in small groups (max 5 people, preferably of different academic backgrounds). Within these groups, you can request help with programming, discuss any issues that come up and the outcome of the different steps.

Provided data

You are provided with several corpus fragments, each from a corpus that has specific characteristics, because the purpose of this session is for you to compare different corpora. You do not have to process all of these corpora. But once your program is written, you should be able to process all these corpora by just passing each of them as an argument (after decompressing the compressed files: `tar -xzvf <filename.tar.gz>`).

1. ACL_partial_corpus.tar.gz – a (partial) collection of ACL abstracts
2. BNCSplitWordsCorpus.tar.gz – a dialogue corpus, part of the BNC corpus
3. english-brown.tar.gz – a fragment of the Brown corpus (news category)
4. MovieCorpus.tar.gz – a corpus of movie scripts
5. TwitterLowerAsciiCorpus.tar.gz – a fragment of Twitter conversations

You are also allowed – actually encouraged! – to analyze your own corpus if you have some such data that you want to work with.

Recommended python libraries

I recommend the following python libraries. You are allowed to use any library you like. However, the exercises are described from the point of view of the libraries I recommend.

NLTK a toolkit with numerous algorithms for (multilingual!) language processing `https://www.nltk.org/`

`matplotlib` a library for a wide variety of data visualizations `https://matplotlib.org/`

Corpus analysis

For each corpus, we aim to compute the following statistics:

overall statistics :

- number of sentences
- number of tokens
- number of types (i.e., number of lemmas)

distributions :

- distribution of sentence lengths (how many sentences of length n_i are there in the corpus, where $\{n_i|i\}$ is the set of sentence lengths that appear in the corpus
- distribution of types in the corpus
- distribution of types in the corpus without stop words
- distribution of POS tags in the corpus
- distribution of open-class POS tags in the corpus

ratios, averages, mins and maxs :

- types/tokens ratio
- average, min and max sentence lengths in each corpus

To obtain these statistics do the following processing steps:

1. Read in the corpus line by line from the given file. For help with I/O operations in python, check out this website:
   - `https://docs.python.org/3/tutorial/inputoutput.html`
   - **NOTE:** the corpora fragments provided are similarly formatted, so once you develop your program, you should be able to provide any of the input corpora (decompress them first!) as an argument, and obtain the statistics.

2. Assume each line read from the file is a document, and it should be split into sentences, and each sentence tokenized. For this we use NLTK. For help on how to split a string into sentence and tokenize each sentence, check out these websites:

   - `http://www.nltk.org/book/ch03.html`
   - `https://www.nltk.org/api/nltk.tokenize.html`
   - You can also tokenize without splitting into sentences, but we will count sentences as we will keep track of sentence lengths as well, so it is best to split into sentences from the beginning.
   - **Keep track of sentence lengths, and the total number of tokens**

3. POS tag the sentences and lemmatize the words using NLTK. Here is help on how to do that:

   - `http://www.nltk.org/book/ch05.html`
   - **keep track of lemmas (i.e., types) and POS tags – overall counts, and also distributions**

4. Compute the required statistics

5. Use `matplotlib` to plot and compare distributions of sentence lengths across different corpora, and the lemma distributions. To plot the lemma distributions across different corpora, the x-axis should represent the words, and the y-axis the relative frequency of that particular word in the different corpora. For help on plotting, check out the following website:

   - `https://matplotlib.org/3.3.3/tutorials/introductory/sample_plots.html#sphx-glr-tutorials-introductory-sample-plots-py`