Methods in Computational Linguistics
Corpus Metadata & Analysis: Exercise

Franziska Weeber

Master of Science *Computational Linguistics*
Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart

November 18, 2025

Today’s Exercise and Slides

Today’s exercise and slides are based on the material of Dr. Eva Maria Vecchi.

Today's Exercise

- Read in a set of various corpora
- process them (as instructed)
- compute basic statistics
- compare these different statistics across the provided corpora

Resources

- corpora
    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
- CorpusAnalysis_exercise.py – a file containing an example function structure to complete the exercise

Folder Setup and VS Code

You can decide on where you want to store your corpora and code (e. g., whether you want to introduce more structure with subfolders), the following is just a recommendation so you do not have to adapt the code snippets we provide.

- create a folder for the methods exercises which you open in VS Code
- store all the code and resources in this folder

Requirements

- use your VS Code terminal from the Methods Exercise folder mentioned in the previous slide.
- Recommended: Install everything in a virtual environment to avoid version conflicts etc.
- Required libraries:
  - Python3
  - `NLTK`
  - `matplotlib`
  - `chardet`

Virtual Environments

Make sure copying this code does not introduce any unintended whitespace!

Listing: Requirements

```
# create virtual environment in your current folder in a new folder named venv
python -m venv ./venv

# EITHER activate the virtual environment (Linux, Mac)
source venv/bin/activate
# OR activate the virtual environment (Windows)
venv/Scripts/activate

# ... install your packages (next slide)

# Deactivate again
deactivate
```

Installing Requirements

Make sure copying this code does not introduce any unintended whitespace and your virtual environment is activated!

Listing: Requirements

```
# Make sure you have pip for dep installations
python -m pip install --upgrade pip

# Install dependencies
pip install nltk matplotlib chardet

# Unzip corpora files
tar -xzvf *.tar.gz
```

Run a Python file (`.py`)

First, find your python executable path

- `which python` (in terminal with activated environment)
- if your virtual environment is in the folder `venv`:
  - Windows: `venv/bin/python`
  - Linux/Mac: `venv/Scripts/python`

Then, go to...

- `Python: Select Interpreter`
- `Enter Interpreter Path`
- Enter the path you just identified

Now your virtual environment will be used when executing the file.

Run a Jupyter Notebook file (.ipynb)

Make sure copying this code does not introduce any unintended whitespace and your virtual environment is activated!

Listing: Notebook Kernel

```
# Install dependencies
pip install ipykernel

# add the named kernel
python -m ipykernel install --user --name=
    methodsenv
```

Now, you can select this kernel named `methodsenv` on the top right of the notebook.

To run the code, run the cells where you define the functions first and then copy the code below `if __name__ == '__main__':` to your notebook and remove the indentation.

Quick Guide

- The following quick guide is a suggestion on how to structure your code and which libraries to use
- Feel free to use something else that works
- All functions here are already prepared for you in the uploaded python file, use that as a starting point!

Quick Guide

Listing: libraries

```
# Don't forget all libraries you'll need, eg:
import os  # for reading in files
import nltk  # corpus processing
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.corpus import stopwords
from nltk import pos_tag, WordNetLemmatizer
import matplotlib.pyplot as plt  # plotting
from collections import Counter
import chardet  # encoding
import string  # working with strings
import re  # regular expressions
```

Quick Guide

Listing: nltk

```
nltk.download(’punkt’)
nltk.download(’stopwords’)
nltk.download(’averaged_perceptron_tagger’)
nltk.download(’wordnet’)
nltk.download(’punkt_tab’)
nltk.download(’averaged_perceptron_tagger_eng’)
```

Quick Guide

Listing: encoding check

```python
def detect_file_encoding(file_path):
    """Detect the encoding of a file."""
    with open(file_path, "rb") as f:
        # Read the first 10,000 bytes
        raw_data = f.read(10000)
        result = chardet.detect(raw_data)
        return result["encoding"]
```

Listing: load corpus, one by one

```python
def load_corpus(file_path):
    """Load the content of the file with fallback
         encodings."""
```

Quick Guide

Listing: tokenize all sentences in the corpus

```
def tokenize_sentences(lines):
    """Tokenize the lines into sentences and words."""
```

Listing: process corpus

```
def process_tokens(tokens):
    """Process tokens to compute lemmas and POS tags,
       removing punctuation (& stopwords)"""
```

Listing: compute defined statistics

```
def compute_statistics(sentences, tokens, lemmas,
   pos_tags):
    """Compute various statistics for the corpus."""
```

Quick Guide

Listing: plot statistics

```python
def visualize_statistics(stats, corpus_name):
    """Visualize distributions and statistics."""
    # Sentence length distribution

    # Top 20 lemmas

    # POS tag distribution
```