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

Data

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

Requirements

- Recommended: Install everything in a virtual environment to avoid version conflicts etc.
- Required libraries:
  - Python3
  - `NLTK`
  - `matplotlib`
  - `chardet`

Requirements

Listing: Requirements

```
# create virtual environment (Linux, Mac)
python -m venv './venv'

# activate the virtual environment (Linux, Mac)
source venv/bin/activate

# 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
```

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

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

```
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

```
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

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

    # Top 20 lemmas

    # POS tag distribution
```