CORPORA, WORDS, TOKENIZATION

Mehmet Can Yavuz, PhD.

Adapted from Julia Hockenmaier, NLP S2023 - course material
https://courses.grainger.illinois.edu/cs447/sp2023/

KEY CONCEPTS

- Corpora
- Text
- Tokenization

Polonius: What do you read, my lord?
Hamlet: Words, words, words.
Act 2, scene 2, Hamlet

Text Data Hierarchy

Corpora

Collected Plays

Corpus

Hamlet

Document

Act 2, Scene 2

Token

Words

WHAT IS A CORPUS?

- **A corpus** (plural: *corpora*) is a collection of natural language data (i.e., a data set)
- **Raw corpora** have only minimal (or no) processing:
  - Sentence boundaries may or may not be identified. There may or may not be metadata
  - Typos (written text) or disfluencies (spoken language) may or may not be corrected.
- **Annotated corpora** contain some labels (e.g., POS tags, sentiment labels), or linguistic structures (e.g., syntax trees, semantic interpretations), etc.

CORPUS INFORMATION

- **Basic statistics:** corpus size, word counts, etc.
- **Intellectual Property** (copyright, licenses)
- What **language** (variety) is represented?
  - English, British English, African American English, French,
- What **genre** is represented in this corpus?
  - Newswire, social media, fiction, poetry, essays, conversations,
- When and in what **situation** was the text produced?
  - Web crawls vs. conversations recorded in a lab, vs. crowdsourced tasks (e.g image descriptions), etc.
- Speaker/Author **demographics**?
- **Motivation/purpose** for data collection
- **Annotation Process**
  - Annotation guidelines
  - Annotator demographics and training
  - Quality assessment (inter-annotator agreement, adjudication)

WORDS AND TOKENIZATION

- **Tokenization** is the **process of identifying word boundaries**.
- Text is just **a sequence of characters**:
  - **Text:** Of course he wants to take the advanced course too. He already took two beginners' courses.
- How do we split this text into **words and sentences**?
  - [[Of, course, he, wants, to, take, the, advanced, course, too, .],
    [He, already, took, two, beginners', courses, .]]

Current immunosuppression protocols to prevent lung transplant rejection reduce pro-inflammatory and T-helper type 1 (Th1) cytokines. However, Th1 T-cell pro-inflammatory cytokine production is important in host defense against bacterial infection in the lungs. Excessive immunosuppression of Th1 T-cell pro-inflammatory cytokines leaves patients susceptible to infection.

Current immunosuppression protocols to prevent lung transplant rejection reduce pro-inflammatory and T-helper type 1 (Th1) cytokines.

However, Th1 T-cell pro-inflammatory cytokine production is important in host defense against bacterial infection in the lungs.

Excessive immunosuppression of Th1 T-cell pro-inflammatory cytokines leaves patients susceptible to infection.

sentence boundary

= period + space(s) + capital letter

Regular expression in Perl:

```
s/\. +([A-Z])/\.\n\1/g;
```

BUT THERE WILL PROBABLY BE ERRORS…

- How can we identify that this is two sentences?
  - Mr. Smith went to D.C. Ms. Xu went to Chicago instead.

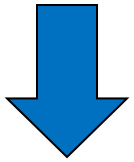
  Challenge: punctuation marks in abbreviations (Mr., D.C, Ms,…)

  [It's easy to handle a small number of known exceptions, but much harder to identify these cases in general]
- See also this headline from the NYT (08/26/20):
  - Anthony Martignetti ('Anthony!'), Who Raced Home for Spaghetti, Dies at 63
- How many sentences are in this text?
  - "The San Francisco-based restaurant," they said, "doesn't charge $10".

  Answer: just one, even though "they said" appears in the middle of another sentence.
- Similarly, we typically treat this also just as one sentence:
  - They said: "The San Francisco-based restaurant doesn't charge $10".

TOKENIZATION

The protein is activated by IL2.



The protein is activated by IL2 .

- Convert a sentence into a sequence of *tokens*

- Why do we tokenize?
- Because we do not want to treat a sentence as a sequence of *characters*!

SPELLING VARIANTS, TYPOS, ETC.

- The same word can be written in different ways:
  - with different **capitalizations**: lowercase "cat" (in standard running text)
    - capitalized "Cat" (as first word in a sentence, or in titles/headlines), all-caps "CAT" (e.g. in headlines)
  - with different **abbreviation** or **hyphenation** styles:
    - US-based, US based, U.S.-based, U.S. based US-EU relations, U.S./E.U. relations, …
  - with **spelling variants** (e.g. regional variants of English):
    - labor vs labour, materialize vs materialise,
  - with **typos** (teh)

- Good practice: Be aware of (and/or document) any normalization (lowercasing, spell-checking, …) your system uses!

COUNTING WORDS: TOKENS VS TYPES

- When counting words in text, we distinguish between word **types** and word **tokens**:

— The **vocabulary** of a language is the set of (unique) word types:
  - V = {a, aardvark, ..., zyzzva}

— The **tokens** in a document include all occurrences of the word types in that document or corpus
  - (this is what a standard word count tells you)

— The **frequency** of a word (type) in a document
  - the number of occurrences (tokens) of that type

HOW MANY DIFFERENT WORDS ARE THERE IN ENGLISH?

- How large is the **vocabulary** of English (or any other language)?
  - **Vocabulary size** = the number of distinct word types Google N-gram corpus: 1 trillion tokens,
  - 13-million-word types that appear 40+ times

- If you count words in text, you will find that…
  - …a few words (mostly closed-class) are very frequent (the, be, to, of, and, a, in, that,…)
  - … most words (all open class) are very rare.
  - … even if you've read a lot of text,
  - you will keep finding words you haven't seen before.

- **Word frequency**: the number of occurrences of a word type in a text (or in a collection of texts)

VOCABULARY SIZE AND CORPUS SIZE

- The number of distinct word types (vocabulary size) increases with the size of the corpus
- **Herdan's Law/Heap's Law:**
  - A corpus of N tokens has a vocabulary of size

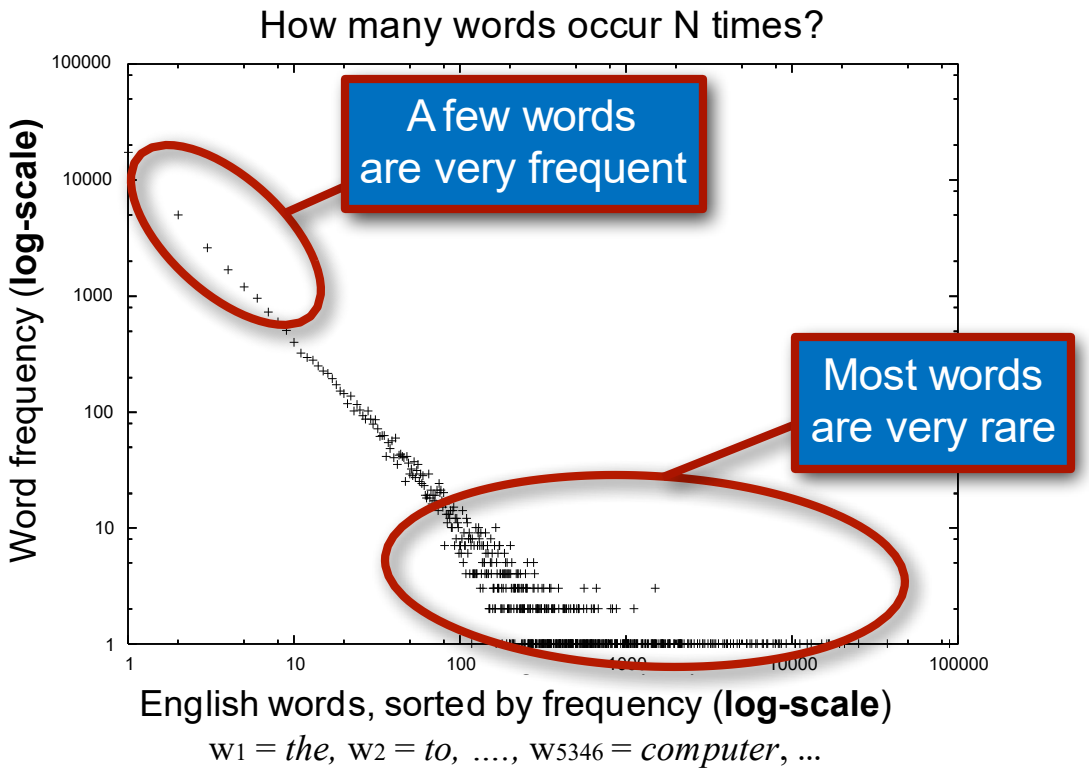
$$V = kN^{\beta}$$

  - for positive constants k and $0 < \beta < 1$

ZIPF'S LAW: THE LONG TAIL

the *r*-th most common word w_r has $P(w_r) \propto 1/r$

- In natural language:
  - A small number of events (e.g. words) occur with high frequency.
  - A large number of events occur with very low frequency



IMPLICATIONS OF ZIPF'S LAW FOR NLP

The good: Any text will contain words but words that are very common. We have seen these words often enough that we know (almost) everything about them. These words will help us get at the structure (and possibly meaning) of this text.

The bad: Any text will contain several words that are rare. We know something about these words but haven't seen them often enough to know everything about them. They may occur with a meaning or a part of speech we haven't seen before.

The ugly: Any text will contain several words that are unknown to us. We have never seen them before, but we still need to get at the structure (and meaning) of these texts.

DEALING WITH THE BAD AND THE UGLY

- Our systems need to be able to **generalize** from what they have seen to unseen events.

- There are two (complementary) approaches to generalization:

— **Linguistics** provides us with insights about the rules and structures in language that we can exploit in the (symbolic) representations we use

E.g.: a finite set of grammar rules is enough to describe an infinite language

— **Machine Learning/Statistics** allows us to learn models (and/or representations) from real data that often work well empirically on unseen data

E.g. most statistical or neural NLP

HOW DO WE REPRESENT WORDS?

- Option 1: Words are **atomic symbols**

— Each (surface) word form is its own symbol

— Add some generalization by mapping different forms of a word to the same symbol

  — **Normalization:** map all variants of the same word (form) to the same canonical variant (e.g. lowercase everything, normalize spellings, perhaps spell-check)

  — **Lemmatization**: map each word to its lemma

  (esp. in English, the lemma is still a word in the language, but lemmatized text is no longer grammatical)

  — **Stemming**: remove endings that differ among word forms (no guarantee that the resulting symbol is an actual word)

HOW DO WE REPRESENT WORDS?

- Option 2: Represent **the structure** of each word

  "books" => "book+N+pl" (or "book+V+3rd +sg")

— This requires a morphological analyzer (more later today)

— The output is often a lemma ("book") plus morphological information ("N pl" i.e. plural noun)

  — This is particularly useful for highly inflected languages, e.g. Czech, Finnish, Turkish, etc. (less so for English or Chinese):

  — In Czech, you might need to know that nejnezajímavějším is a regular, feminine, plural, dative adjective in the superlative.

HOW DO WE REPRESENT UNKNOWN WORDS?

- Many NLP systems assume a fixed vocabulary but still must handle **out-of-vocabulary (OOV)** words.

Option 1: the **UNK** token

Replace all rare words (with a frequency at or below a given threshold, e.g. 2, 3, or 5) in your training data with an UNK token (UNK = “Unknown word”).

Replace *all* unknown words that you come across after training (including rare training words) with the same UNK token

Option 2: **substring-based** representations

[often used in neural models]

Represent (rare and unknown) words [“Champaign”] as sequences of characters [‘C’, ‘h’, ‘a’,…,’g’, ’n'] or substrings [“Ch”, “amp”, “ai”, “gn”]

Byte Pair Encoding (BPE): learn which character sequences are common in the vocabulary of your language, and treat those common sequences as atomic units of your vocabulary

REVIEW OF THE KEY CONCEPTS

- Corpora
- Text
- Tokenization

Polonius: What do you read, my lord?
Hamlet: Words, words, words.
Act 2, scene 2, Hamlet