Lecture 9: Corpus, Word distributions, Word vectors

LING-351 Language Technology and LLMs

Instructor: Hakyung Sung

September 23, 2025

Table of contents

Review

Last class

- Text as data: Two different approaches

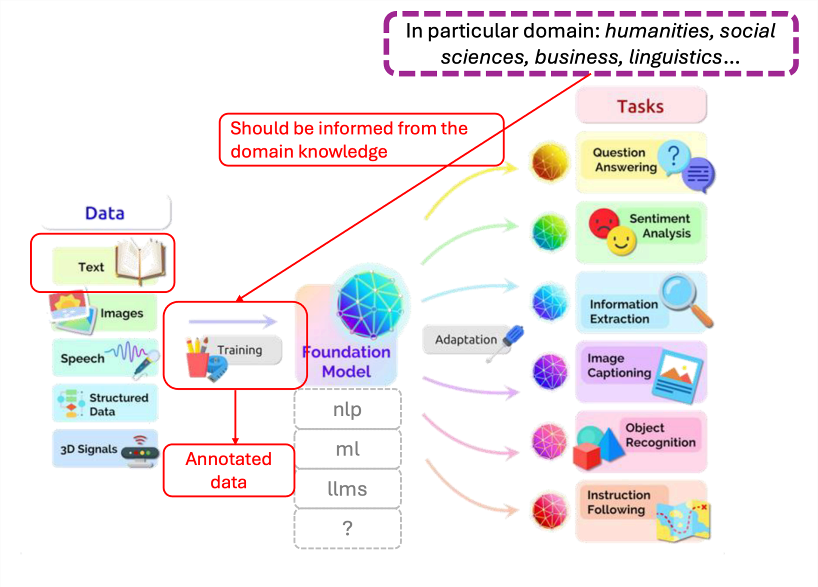
Last class

- Text as data: Two different approaches
- Questions with answers in text

Last class

- Text as data: Two different approaches
- Questions with answers in text
- Good data for the data-driven approach

Logistics of the data-driven approach: Annotation



Lesson plan

Lesson plan

- ~~Review~~

Lesson plan

- ~~Review~~
- Exploring corpora

Lesson plan

- ~~Review~~
- Exploring corpora
- Word distributions

Lesson plan

- ~~Review~~
- Exploring corpora
- Word distributions
- Word vectors

Exploring English corpora

What is a corpus?

- Corpus: Curated collection of texts (and sometimes audio/video) (for linguistic analysis).

What is a corpus?

- **Corpus**: Curated collection of texts (and sometimes audio/video) (for linguistic analysis).
- Balanced corpus: Samples across genres/registers to represent a broad snapshot

What is a corpus?

- **Corpus**: Curated collection of texts (and sometimes audio/video) (for linguistic analysis).
- Balanced corpus: Samples across genres/registers to represent a broad snapshot
- Monitor corpus: Continuously updated to track change over time

What is a corpus?

- **Corpus**: Curated collection of texts (and sometimes audio/video) (for linguistic analysis).
- Balanced corpus: Samples across genres/registers to represent a broad snapshot
- Monitor corpus: Continuously updated to track change over time
- Annotations/metadata: POS tags, lemmas, syntax, dates, genre, speaker info, etc.

1. Brown Corpus (Francis & Kucera, 1979)

- Earliest million-word, machine-readable corpus of American English.

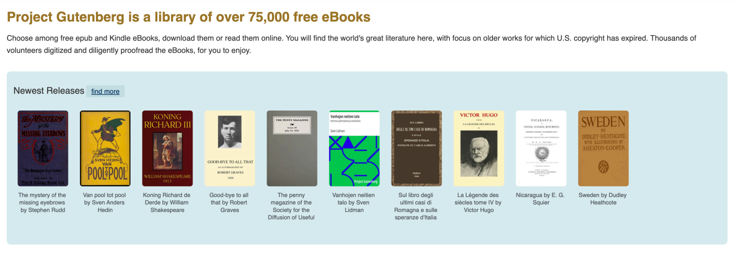
1. Brown Corpus (Francis & Kucera, 1979)

- Earliest million-word, machine-readable corpus of American English.
- **Balanced** across genres: news, editorials, religion, fiction, magazines, academic, etc.

1. Brown Corpus (Francis & Kucera, 1979)

- Earliest million-word, machine-readable corpus of American English.
- **Balanced** across genres: news, editorials, religion, fiction, magazines, academic, etc.
- We've been used it for the Python tutorial!

Digitized public-domain books; classic literature and beyond.



Sourced from *https://www.gutenberg.org/*

3. British National Corpus (BNC; Burnard & Aston, 1998)

100M words of late-20thC British English

The British National Corpus: The platform gives access to five million words from the BNC representing informal conversations between British English speakers from the 1990s.

The British National Corpus 2014: The platform gives access to five million words from the BNC 2014 representing informal conversation between British English speakers from 2000s.

Sourced from *`https://wp.lancs.ac.uk/corpusforschools/bnclab/`*

Suite of corpora for child–caregiver interaction across multiple languages.

Sourced from *`https://talkbank.org/childes/`*

5. English-Corpora.org (Mark Davies et al.)

“These are the most widely used online corpora, and they serve many different purposes for teachers and researchers at universities throughout the world.”

Sourced from *`https://www.english-corpora.org/`*

6. Social media datasets

- Reddit, Yelp, Stack Exchange, and similar sources often have exportable datasets (Check here? *https://socialmediaie.github.io/MetaCorpus/#metacorpus*)

7. More places to explore

- Lancaster University CQPweb hubs:
  https://cqpweb.lancs.ac.uk/

7. More places to explore

- Lancaster University CQPweb hubs: *https://cqpweb.lancs.ac.uk/*
- Georgetown CQP: *https://gucorpling.org/cqp/*

7. More places to explore

- Lancaster University CQPweb hubs: *`https://cqpweb.lancs.ac.uk/`*
- Georgetown CQP: *`https://gucorpling.org/cqp/`*
- Many others via university libraries, national archives, and domain-specific repositories.

In-class activity

Step 1 (15 mins)
Examine **one corpus** and explore its key features. *Please make sure to take notes, as you will be asked to submit your output in the upcoming tutorial section!

Step 2 (10 mins)
Introduce the corpus you explored to your peers in small groups. (*Shared deck*)

Step 3 (15 mins)
Share key points from each group with the whole class.

Building your own corpus: practicalities

- **Collection:** Sources, sampling, permissions, web scrapping

Building your own corpus: practicalities

- **Collection:** Sources, sampling, permissions, web scrapping
- **Ethics:** Consent, privacy, potential harms, data sharing vs. open science.

Building your own corpus: practicalities

- Collection: Sources, sampling, permissions, web scrapping
- Ethics: Consent, privacy, potential harms, data sharing vs. open science.
- Access: Will others be able to *replicate* your study from your release?

Bias, coverage, and equity

- Who is represented? How were texts selected?

Bias, coverage, and equity

- Who is represented? How were texts selected?
- Standard varieties often dominate; minoritized varieties/languages are under-resourced.

Bias, coverage, and equity

- Who is represented? How were texts selected?
- Standard varieties often dominate; minoritized varieties/languages are under-resourced.
- Corpus choices can reproduce social inequalities—make limitations **explicit** in write-ups.

Word distributions

Word distributions

Examining word distributions is the first and most important step in corpus/text analysis.

Word distributions

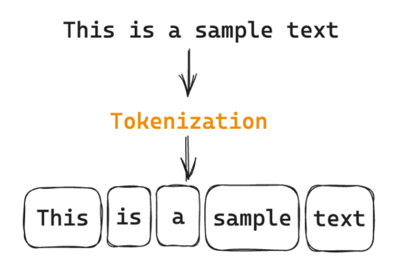
Examining word distributions is the first and most important step in corpus/text analysis.

As we've discussed in previous lectures/Python tutorials, we have already implemented several techniques such as...

Word distributions

Examining word distributions is the first and most important step in corpus/text analysis.

As we've discussed in previous lectures/Python tutorials, we have already implemented several techniques such as…



Zipf's power law (1932)

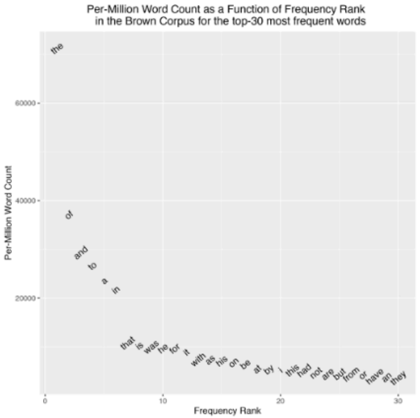


Figure 4.3: Per-million-word frequency of words in the Brown Corpus as a function of their frequency rank (ordered from left to right as the first most frequent word, the second most frequent, and so on).

Zipf's power law (1932)

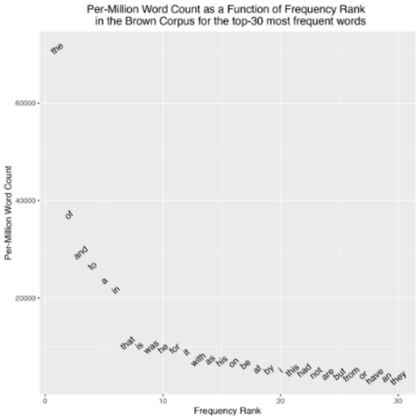


Figure 4.3: Per-million-word frequency of words in the Brown Corpus as a function of their frequency rank (ordered from left to right as the first most frequent word, the second most frequent, and so on).

Implication: Few words are very frequent; many are rare ⇒ long tail.

Zipf's power law (1932)

- type vs. token

Zipf's power law (1932)

- type vs. token
- frequency $\propto$ 1/rank.

Zipf's power law (1932)

- type vs. token
- frequency $\propto$ 1/rank.
- e.g., Brown corpus: *the* $\approx$ 6% tokens; *of* $\approx$ 3%; *and* $\approx$ 2.6%.

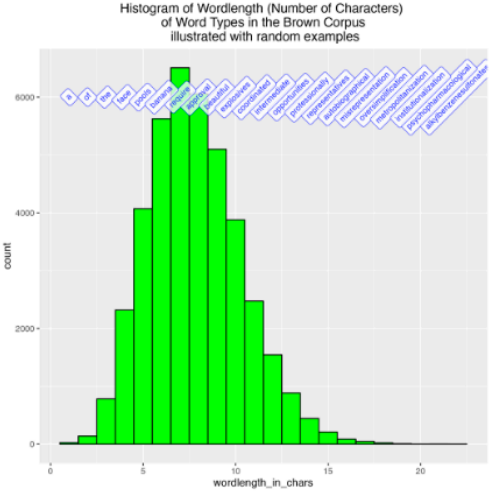


Figure 4.4: Histogram of the length (number of characters) of all word types in the Brown Corpus.

Zipf's brevity law

- More frequent words tend to be **shorter** (characters/syllables).

Zipf's brevity law

- More frequent words tend to be **shorter** (characters/syllables).
- Efficiency pressure: frequent items economize articulatory/processing effort.

Zipf's brevity law

- More frequent words tend to be **shorter** (characters/syllables).
- Efficiency pressure: frequent items economize articulatory/processing effort.
- Most frequent Brown words: monosyllabic, $\leq$3 letters (*the, of, and, a, in, to, is, was, I, for*).

Heaps' law

As you read more tokens in a corpus, you keep seeing new word **types**, but the **rate** of new words **slows down**.

Heaps' law

As you read more tokens in a corpus, you keep seeing new word **types**, but the **rate** of new words **slows down**.

Example

- After 1,000 tokens: $\sim$700 unique words

Why it matters?

Heaps' law

As you read more tokens in a corpus, you keep seeing new word **types**, but the **rate** of new words **slows down**.

Example

- After 1,000 tokens: $\sim$700 unique words
- After 10,000 tokens: not 7,000, but maybe $\sim$2,500–3,500

Why it matters?

Heaps' law

As you read more tokens in a corpus, you keep seeing new word **types**, but the **rate** of new words **slows down**.

Example

- After 1,000 tokens: $\sim$700 unique words
- After 10,000 tokens: not 7,000, but maybe $\sim$2,500–3,500

Why it matters?

- Estimate how much data you need before vocabulary "stabilizes"

Heaps’ law

As you read more tokens in a corpus, you keep seeing new word **types**, but the **rate** of new words **slows down**.

Example

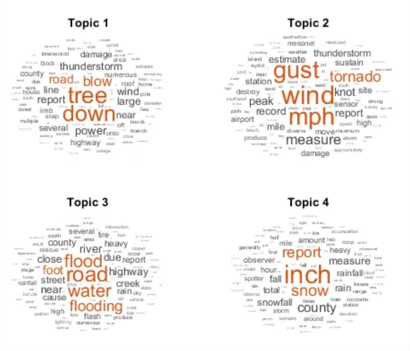
- After 1,000 tokens: $\sim$700 unique words
- After 10,000 tokens: not 7,000, but maybe $\sim$2,500–3,500

Why it matters?

- Estimate how much data you need before vocabulary “stabilizes”
- Reminds us that growth is **sublinear**

Topic modeling

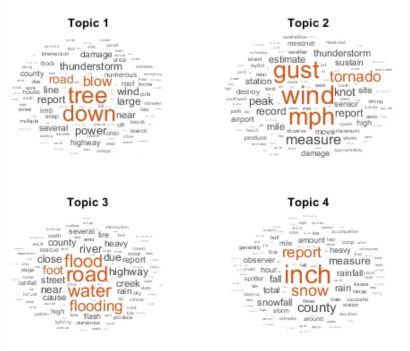
- A topic modeling is a type of **statistical modeling** for discovering the **abstract** topic that occur in a collection of documents.



Sourced from *https://medium.com/analytics-vidhya/*

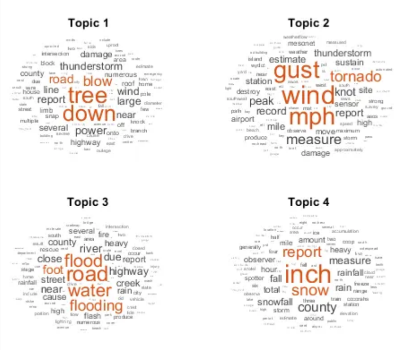
Topic modeling

- A topic modeling is a type of **statistical modeling** for discovering the **abstract** topic that occur in a collection of documents.
- **Exploratory**: No annotated labels; discover latent structure using word frequencies/distributions



Sourced from *https://medium.com/analytics-vidhya/*

- A topic modeling is a type of **statistical modeling** for discovering the **abstract** topic that occur in a collection of documents.
- **Exploratory**: No annotated labels; discover latent structure using word frequencies/distributions
- Classic model: LDA (Blei et al., 2003).



Sourced from *https://medium.com/analytics-vidhya/*

LDA: minimal workflow

Key idea: (1) Each document is a *mixture of topics*. (2) Each topic is a *distribution over words*. (3) Given only the words, LDA uses Bayesian inference to approximate the hidden topic structure.

1. Collect documents; tokenize, lemmatize; remove stop words.

LDA: minimal workflow

Key idea: (1) Each document is a *mixture of topics*. (2) Each topic is a *distribution over words*. (3) Given only the words, LDA uses Bayesian inference to approximate the hidden topic structure.

1. Collect documents; tokenize, lemmatize; remove stop words.
2. Choose K topics; start with random assignments.

LDA: minimal workflow

Key idea: (1) Each document is a *mixture of topics*. (2) Each topic is a *distribution over words*. (3) Given only the words, LDA uses Bayesian inference to approximate the hidden topic structure.

1. Collect documents; tokenize, lemmatize; remove stop words.
2. Choose K topics; start with random assignments.
3. Iterate: update topic assignments using document–topic and topic–word counts.

LDA: minimal workflow

Key idea: (1) Each document is a *mixture of topics*. (2) Each topic is a *distribution over words*. (3) Given only the words, LDA uses Bayesian inference to approximate the hidden topic structure.

1. Collect documents; tokenize, lemmatize; remove stop words.
2. Choose K topics; start with random assignments.
3. Iterate: update topic assignments using document–topic and topic–word counts.
4. Inspect top words per topic; give each topic a human-readable label.

LDA: minimal workflow

Key idea: (1) Each document is a *mixture of topics*. (2) Each topic is a *distribution over words*. (3) Given only the words, LDA uses Bayesian inference to approximate the hidden topic structure.

1. Collect documents; tokenize, lemmatize; remove stop words.
2. Choose *K* topics; start with random assignments.
3. Iterate: update topic assignments using document–topic and topic–word counts.
4. Inspect top words per topic; give each topic a human-readable label.
5. We'll do some hands-on practice with topic modeling on Thursday!

Word vectors

Encoding

- Words themselves cannot be given as inputs to computers

Encoding

- Words themselves cannot be given as inputs to computers
- BUT, numbers can be given as inputs to computers

Encoding

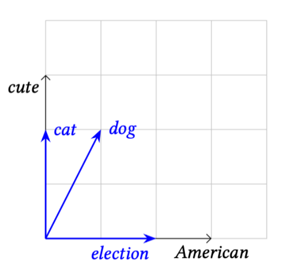
- Words themselves cannot be given as inputs to computers
- BUT, numbers can be given as inputs to computers
- Encoding = converting words to vectors

Encoding

- Words themselves cannot be given as inputs to computers
- BUT, numbers can be given as inputs to computers
- Encoding = converting words to vectors
    - *vector*: an ordered list of numbers (e.g., [0.1, 0.3, -0.5])

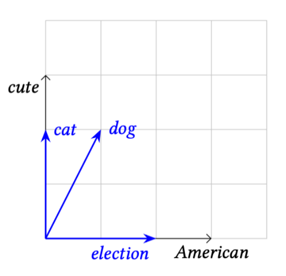
Word vectors

- If two words often occur near the same neighbors (e.g., *dog* and *cat* near *cute*), their vectors will be similar.



Word vectors

- If two words often occur near the same neighbors (e.g., *dog* and *cat* near *cute*), their vectors will be similar.
- *How*? Algorithms can automatically learn these vectors from corpus data



Word vectors

Core idea:

- Start with a large corpus

Word vectors

Core idea:

- Start with a large corpus
- Every word in a fixed vocabulary is represented by a vector

Core idea:

- Start with a large corpus
- Every word in a fixed vocabulary is represented by a vector
- Go through each position t in the text, which has a center word and a context word

Word vectors

Core idea:

- Start with a large corpus
- Every word in a fixed vocabulary is represented by a vector
- Go through each position t in the text, which has a center word and a context word
- Calculate the probability of a center word given a context word (or vice versa)

Word vectors

Core idea:

- Start with a large corpus
- Every word in a fixed vocabulary is represented by a vector
- Go through each position t in the text, which has a center word and a context word
- Calculate the probability of a center word given a context word (or vice versa)
- Keep adjusting the word vectors to **maximize** the probability

Word vectors

Core idea:

- Start with a large corpus
- Every word in a fixed vocabulary is represented by a vector
- Go through each position t in the text, which has a center word and a context word
- Calculate the probability of a center word given a context word (or vice versa)
- Keep adjusting the word vectors to **maximize** the probability
- (*more on this in the NLP class*!)

Vector arithmetic

- Once the vectors are learned, word vectors can be used for **mathematical operations**.

Vector arithmetic

- Once the vectors are learned, word vectors can be used for **mathematical operations**.
- *Word2Vec* (Mikolov et al. 2013):

$$\text{king} - \text{man} + \text{woman} \approx \text{queen}$$

Vector arithmetic

- Once the vectors are learned, word vectors can be used for **mathematical operations**.
- *Word2Vec* (Mikolov et al. 2013):

$$\text{king} - \text{man} + \text{woman} \approx \text{queen}$$

- We'll also explore *Word2Vec* on Thursday.

Wrap-up

Wrap-up

- Exploring corpora

Wrap-up

- Exploring corpora
    - Confident explaining about at least one corpus:)

Wrap-up

- Exploring corpora
    - Confident explaining about at least one corpus:)
- Word distributions

Wrap-up

- Exploring corpora
    - Confident explaining about at least one corpus:)
- Word distributions
    - Zipf's power law

Wrap-up

- Exploring corpora
  - Confident explaining about at least one corpus:)
- Word distributions
  - Zipf's power law
  - Topic modeling

Wrap-up

- Exploring corpora
    - Confident explaining about at least one corpus:)
- Word distributions
    - Zipf's power law
    - Topic modeling
- Word vectors

Reminder!

By October 2nd...

1. Review the sample papers on the course website (*https://hksung.github.io/Fall25_LING351/materials/*)
2. Add your names to the shared sheet (*https://docs.google.com/spreadsheets/d/1on8icHoXUsj74m1UNEHk8CycHEAmVH1nRsUatpn9xYc/edit?usp=sharing*) - *First come first served*
3. You may also choose articles beyond this list (e.g., CALL), but please check with me first
4. **Choose one paper you like best**