Language & Technology

Lecture 4: More Word-Based Models

Alëna Aksënova & Aniello De Santo

Stony Brook University
alena.aksenova@stonybrook.edu
aniello.desanto@stonybrook.edu

Previously on "Language & Technology"...

- Counting words!

string = "The sun shone, having no alternative, on the nothing new."

tokens ["the", "sun", "shone", "having", "no", "alternative", "on", "the", "nothing", "new"]

- More technically: for each word type we count its word tokens.

word type a word of a given language

word token instance of the word in a text

Previously on "Language & Technology"...

Steps of Word Counting Model

1. collect corpus
2. tokenize strings
3. count tokens for each type

Counting words is surprisingly useful:

- Culturomics
- Stylistic analysis

But there's more!

Previously on "Language & Technology"...

Steps of Word Counting Model

1. collect corpus
2. tokenize strings
3. count tokens for each type

Counting words is surprisingly useful:

- Culturomics
- Stylistic analysis

But there's more!

Application 3: Authorship Attribution

- Computational stylistic analysis isn't just good at predicting success of novels.
- Word counts pick up on **statistical tendencies in writing** of individual authors
- This means that potential authors can be inferred from word frequencies!

Example: Auditing the Bard's Oeuvre

- Several works of Shakespeare have supposedly been (partially) authored by somebody else:
  - *Edward III*
    co-authored with Thomas Kyd
  - *The Comedy of Errors*,
    Love's Labour Lost,
    The Tempest
    co-authored with Francis Bacon
- These are **fringe ideas** in literary circles.
- But there are word counting models that **support these claims**.

Another Example: Harry Potter's Mum

- New book: *The Cuckoo's Calling* by Robert Galbraith
- *Sunday Times* suspected that to be a pseudonym for J. K. Rowling.
- Evidence from word-based model:
  - 100 most common words
  - average word length
  - word pairs
    (discussed in next lecture)
  - character 4-grams
    (discussed in next lecture)

A Friendly Pointer

Article on LanguageLog:
`http://languagelog.ldc.upenn.edu/nll/?p=5315`

Evaluation

- As with culturomics, it is unclear what to make of the findings.
  - J.K. Rowling: success!
  - Shakespeare: success?
- One can get vastly different results depending on:
  - absolute size of test corpus
  - relative size (e.g. ratio of Shakespeare to Bacon)
  - whether certain words are ignored
  - whether one pays attention to rare words or frequent words
- Many of these studies feel backwards:
  - They start with a specific hypothesis and then present one specific model that supports this hypothesis.
  - But why is this model better than all of the alternatives?

Moral of the Story

- Authorship attribution models still ill-understood
- In the future, these models might answer long-standing questions or pose new ones.
- Maybe you can try it in an English class.

Application 4: Word Meanings

- For some tasks, the word types are not that important, what matters is their **meaning**:
  - web search
  - Google *Ad Sense* (automatic ad placement)
- But what is a word meaning?

Application 4: Word Meanings

- For some tasks, the word types are not that important, what matters is their **meaning**:
  - web search
  - Google *Ad Sense* (automatic ad placement)
- But what is a word meaning?

Example: Dictionary Approach

dog

- is a **mammal**,
- descended from **wolf**,
- is commonly a **pet**,
- subtypes are **poodle**, **bulldog**, ...
- has **fur**,
- ...

Why the Dictionary Approach is Problematic

- Such dictionaries have been tried for computers.
  e.g. WordNet
- They must be created by hand, which is a big problem:
  - expensive
  - only available for some languages
  - many new words missing
- We need dictionaries that can be generated automatically.

Meaning as Word Use

- The philosopher **Ludwig Wittgenstein** said that a word's meaning is its use.

Computational Counterpart

A word's meaning is given by how often it occurs together with other words.

Counting Tokens for Word Meaning

Step 1: Record in how many sentences words **occur together**

Example

The dog barked at the cat. The cat ran away. The dog ran after the cat. The dog kept barking. He also kept running.

	dog	cat	bark	run
dog	-			
cat		-		
bark			-	
run				-

Counting Tokens for Word Meaning

Step 1: Record in how many sentences words **occur together**

Example

The dog barked at the cat. The cat ran away. The dog ran after the cat. The dog kept barking. He also kept running.

	dog	cat	bark	run
dog	-	2		
cat		-		
bark			-	
run				-

Counting Tokens for Word Meaning

Step 1: Record in how many sentences words **occur together**

Example

The dog barked at the cat. The cat ran away. The dog ran after the cat. The dog kept barking. He also kept running.

	dog	cat	bark	run
dog	-	2	2	
cat		-		
bark			-	
run				-

Counting Tokens for Word Meaning

Step 1: Record in how many sentences words **occur together**

Example

The dog barked at the cat. The cat ran away. The dog ran after the cat. The dog kept barking. He also kept running.

	dog	cat	bark	run
dog	-	2	2	1
cat		-		
bark			-	
run				-

Counting Tokens for Word Meaning

Step 1: Record in how many sentences words **occur together**

Example

The dog barked at the cat. The cat ran away. The dog ran after the cat. The dog kept barking. He also kept running.

	dog	cat	bark	run
dog	-	2	2	1
cat	2	-		
bark			-	
run				-

Counting Tokens for Word Meaning

Step 1: Record in how many sentences words **occur together**

Example

The dog barked at the cat. The cat ran away. The dog ran after the cat. The dog kept barking. He also kept running.

	dog	cat	bark	run
dog	-	2	2	1
cat	2	-	1	
bark			-	
run				-

Counting Tokens for Word Meaning

Step 1: Record in how many sentences words **occur together**

Example

The dog barked at the cat. The cat ran away. The dog ran after the cat. The dog kept barking. He also kept running.

	dog	cat	bark	run
dog	-	2	2	1
cat	2	-	1	2
bark			-	
run				-

Counting Tokens for Word Meaning

Step 1: Record in how many sentences words **occur together**

Example

The dog barked at the cat. The cat ran away. The dog ran after the cat. The dog kept barking. He also kept running.

	dog	cat	bark	run
dog	-	2	2	1
cat	2	-	1	2
bark	2		-	
run				-

Counting Tokens for Word Meaning

Step 1: Record in how many sentences words **occur together**

Example

The dog barked at the cat. The cat ran away. The dog ran after the cat. The dog kept barking. He also kept running.

	dog	cat	bark	run
dog	-	2	2	1
cat	2	-	1	2
bark	2	1	-	
run				-

Counting Tokens for Word Meaning

Step 1: Record in how many sentences words **occur together**

Example

The dog barked at the cat. The cat ran away. The dog ran after the cat. The dog kept barking. He also kept running.

	dog	cat	bark	run
dog	-	2	2	1
cat	2	-	1	2
bark	2	1	-	0
run				-

Counting Tokens for Word Meaning

Step 1: Record in how many sentences words **occur together**

Example

The dog barked at the cat. The cat ran away. The dog ran after the cat. The dog kept barking. He also kept running.

	dog	cat	bark	run
dog	-	2	2	1
cat	2	-	1	2
bark	2	1	-	0
run	1			-

Counting Tokens for Word Meaning

Step 1: Record in how many sentences words **occur together**

Example

The dog barked at the cat. The cat ran away. The dog ran after the cat. The dog kept barking. He also kept running.

	dog	cat	bark	run
dog	-	2	2	1
cat	2	-	1	2
bark	2	1	-	0
run	1	2		-

Counting Tokens for Word Meaning

Step 1: Record in how many sentences words **occur together**

Example

The dog barked at the cat. The cat ran away. The dog ran after the cat. The dog kept barking. He also kept running.

	dog	cat	bark	run
dog	-	2	2	1
cat	2	-	1	2
bark	2	1	-	0
run	1	2	0	-

From Vector to Vector Spaces

Step 2: Construct an n-dimensional vector space.

n is given by the number of word types in the text

2-Dimensional Vector Space with *dog* and *cat*

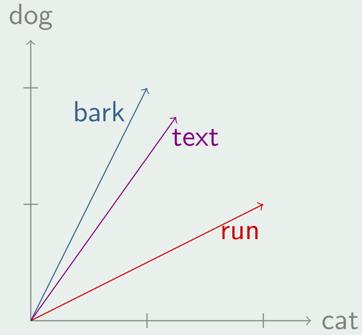
	dog	cat	bark	run
dog	-	2	2	1
cat	2	-	1	2
bark	2	1	-	0
run	1	2	0	-

- *bark* more closely related to *dog*
- *run* more closely related to *cat*

From Word Meaning to Text Meaning

Step 3: construct vector for whole text from word vectors

Meaning for *He either barks or runs and barks*



Is that Realistic?

- **Possible Concerns**
  - Is word meaning really just a bunch of numbers?
  - In a real-word model, the vector space will have thousands of dimensions.
- But this might actually be what you have stored in your brain!

Psycholinguistic Experiment

- Word association task
- The more similar the meaning vectors of two words, the faster test subjects recognize them as related.
- Also: Masked priming effects

Usage for Web Search and Ad Sense

- **Web Search**
  - construct meaning vector for every website
  - convert search string to vector
  - rank websites by vector similarity
- **Ad Sense**
  - construct meaning vector for every website
  - associate every ad with a vector
  - pick ad that most closely matches website vector

Bells and Whistles

- weigh words in search string by linear order
  aliens conspiracy $\neq$ conspiracy aliens
- prioritize certain words in text
- compress vector space for efficiency

Evaluation

- For word meaning, the approach seems to work.
- But it is bad for sentence/text meaning.

Example

The following two sentences receive the same vector:

(1) a. Dog bites man!
    b. Man bites dog!

A Friendly Pointer

How meaning actually works: **LIN 346 Language and Meaning**

An Observation on Frequencies: Zipf's Law

- Word models care about word frequency.
- But there is a problem...

Zipf's Law

The frequency of a type is inversely proportional to its rank.

In Plain English

The most frequent word is

- 2 times as common as the 2nd most frequent word,
- 3 times as common as the 3rd most frequent word,
- and so on.

An Observation on Frequencies: Zipf's Law

- Word models care about word frequency.
- But there is a problem...

Zipf's Law

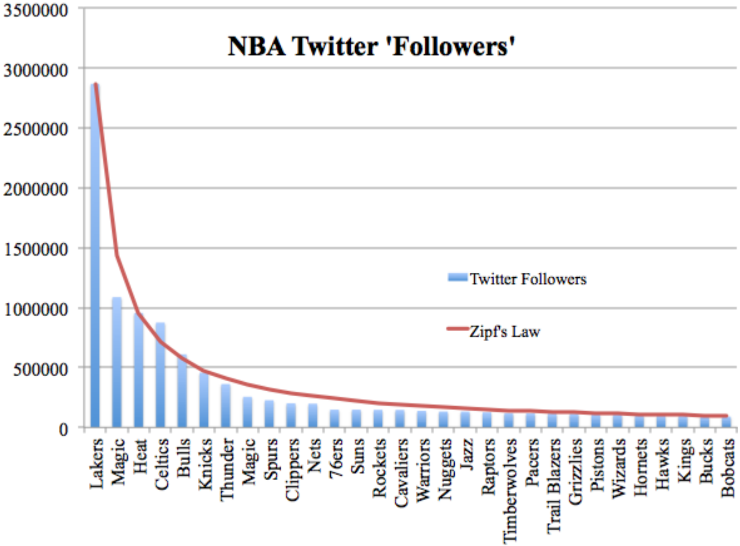
The frequency of a type is inversely proportional to its rank.

In Plain English

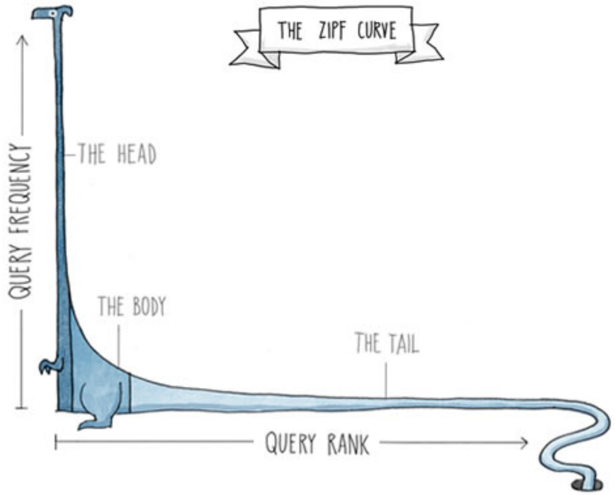
The most frequent word is

- 2 times as common as the 2nd most frequent word,
- 3 times as common as the 3rd most frequent word,
- and so on.

An Example from. . . the NBA?



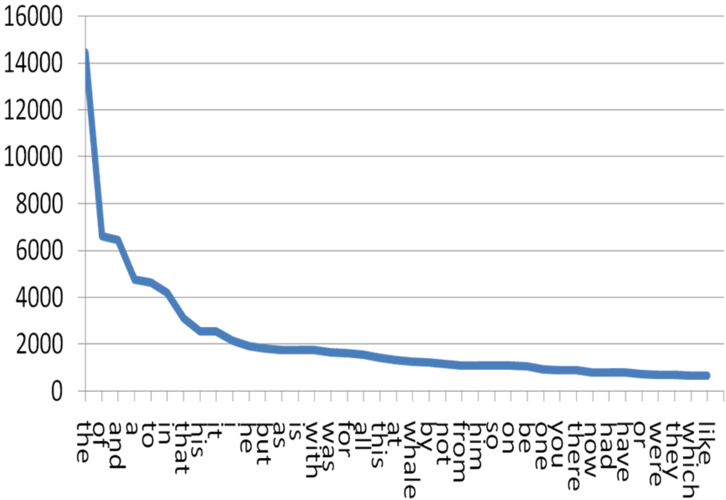
Visualizing Zipf Distributions



Zipf's Law is Everywhere...

- A distribution is probably Zipfian if
  - there is a **long neck**:
    a few types make up the majority of tokens,
  - there is a **long tail**:
    most types only have 1 token (**hapax legomenon**)
- Surprisingly, Zipf's Law shows up in tons of places:
  - size of large cities in a country
  - citations for academic papers
  - frequencies of last names
  - frequencies of weekdays in text

. . . Even in Language!



Stop Words

- About 150 words make up 50% of all English texts:
  the, *of*, *and*, *a*, . . .
- These are called **stop words**.
- Stop words are not very informative for many applications.
- So they are usually discarded after the tokenization step.
- Failure to do so can greatly reduce the model's performance.

Steps of Word Counting Model (Revised)

1. collect corpus
2. remove stop words
3. tokenize strings
4. count tokens for each type

Stop Words

- About 150 words make up 50% of all English texts:
  the, *of*, *and*, *a*, . . .
- These are called **stop words**.
- Stop words are not very informative for many applications.
- So they are usually discarded after the tokenization step.
- Failure to do so can greatly reduce the model's performance.

Steps of Word Counting Model (Revised)

1. collect corpus
2. remove stop words
3. tokenize strings
4. count tokens for each type

Example: A Text Without (Non)-Stop Words

- Stop words are much less informative than non-stop words.
- Just check the example below.

Stop Words only

The having no on the

Example: A Text Without (Non)-Stop Words

- Stop words are much less informative than non-stop words.
- Just check the example below.

Stop Words and Non-Stop Words

The sun shone having no alternative on the nothing new

Example: A Text Without (Non)-Stop Words

- Stop words are much less informative than non-stop words.
- Just check the example below.

Non-Stop Words only

sun shone alternative nothing new

An Important Consequence of Zipf's Law

- Texts mostly consist of stop words.
- Hence it can be difficult to get representative counts for non-stop words.

Sparse Data Problem

- Most of the data is not informative.
- You need tons of data to have enough useful data.

An Important Consequence of Zipf's Law

- Texts mostly consist of stop words.
- Hence it can be difficult to get representative counts for non-stop words.

Sparse Data Problem

- Most of the data is not informative.
- You need tons of data to have enough useful data.

Example

- Most models require corpora with at least a few million sentences.
- Really good models (e.g. Google translate) use billions of data points.

Summary

- Unigram (= word-based) models are surprisingly useful.
  - culturomics
  - stylistic analysis
  - authorship attribution
  - word meaning
  - web search
  - ad sense
  - and much more
- They just count words and do some number crunching with the frequencies.
- These models are everywhere, but they are also very simplistic.
- Quality depends largely on how much data is available.