Introduction to English Linguistics

12: Google Books Ngrams Viewer

N-gram

- Apparently coined 1963
- *Oxford English Dictionary* definition (entry written Sep 2003):

  A sequence of n letters or characters (where n is a variable: see N n. *6a, 6b), esp. one occurring within a longer sequence such as a passage of text.*

Why Use the Term in Computational Linguistics/Corpus Linguistics?

Why Use the Term in Computational Linguistics/Corpus Linguistics?

- Because the alternative is to lemmatize your corpus,
- And lemmatization is hard.

(NB Google Books have lemmatized their corpus, but they haven't let their corpus search function depend on this feature.)

What Is Google Books?

What Is Google Books?

- Began in 2002
- Went live in 2004
- Aims to digitize large numbers of books
- Upwards of 25 million books scanned
- Met with a great deal of litigation (notably Author's Guild and the American Association of Publishers)
- The project has slowed down since c. 2012
- Official (but dated) history page reads "we're not done—not until all of the books in the world can be found by everyone, everywhere, at any time they need them."

What Is the Value of Equipping Google Books with an Ngram Reader?

What Is the Value of Equipping Google Books with an Ngram Reader?

- The largest searchable corpus of print works and ebooks in the history of the world
- Historical value: quantify the historical use of concepts
- Linguistic value: quantify the historical use of words, phrases, spellings
  - Greatly facilitates *OED* attestation research!
- Not feasible to lemmatize so large a corpus *reliably*; bracketing out linguistic entities is the next best approach

Demonstration

books.google.com/ngrams

Terminology

Gram

A sequence of characters

Unigram

A sequence of characters not interrupted by a space (“word”)

Bigram

A sequence of characters interrupted by a single space (“compound”)

Algorithm

Any unigram is scored against the full corpus of unigrams for the chosen language corpus;

Any bigram is scored against the full corpus of bigrams for the chosen language corpus.

Algorithm

Any unigram is scored against the full corpus of unigrams for the chosen language corpus;

Any bigram is scored against the full corpus of bigrams for the chosen language corpus.

Thus a graph plotting a unigram and a bigram is not, strictly speaking, a comparison.

Usage (1/2)

- Enter comma-separated queries to see them plotted against each other

Usage (1/2)

- Enter comma-separated queries to see them plotted against each other
- A wildcard (*) returns the top ten matches
  e.g. `the weather is *`

Usage (1/2)

- Enter comma-separated queries to see them plotted against each other
- A wildcard (`*`) returns the top ten matches
  e.g. `the weather is *`
- `gram_INF` returns inflected forms of a lexical form `gram`
  e.g. `seek_INF` returns *sought*, *seek*, *seeking*, *seeks*

Usage (1/2)

- Enter comma-separated queries to see them plotted against each other
- A wildcard (`*`) returns the top ten matches
  e.g. `the weather is *`
- `gram_INF` returns inflected forms of a lexical form `gram`
  e.g. `seek_INF` returns *sought*, *seek*, *seeking*, *seeks*
- `gram_NOUN`, `gram_VERB`, etc. tries to return only the matching part of speech
  e.g. `feast_VERB` should not find a hit in the sequence “a feast”

Usage (1/2)

- Enter comma-separated queries to see them plotted against each other
- A wildcard (`*`) returns the top ten matches
  e.g. `the weather is *`
- `gram_INF` returns inflected forms of a lexical form `gram`
  e.g. `seek_INF` returns *sought*, *seek*, *seeking*, *seeks*
- `gram_NOUN`, `gram_VERB`, etc. tries to return only the matching part of speech
  e.g. `feast_VERB` should not find a hit in the sequence “a feast”
- `gram_*` plots all parts of speech for that form against each other
  e.g. `feast_*` returns the noun *feast*, the verb *feast*, the adjective *feast*, and some noise

Usage (1/2)

- Enter comma-separated queries to see them plotted against each other
- A wildcard (`*`) returns the top ten matches
  e.g. `the weather is *`
- `gram_INF` returns inflected forms of a lexical form `gram`
  e.g. `seek_INF` returns *sought*, *seek*, *seeking*, *seeks*
- `gram_NOUN`, `gram_VERB`, etc. tries to return only the matching part of speech
  e.g. `feast_VERB` should not find a hit in the sequence "a feast"
- `gram_*` plots all parts of speech for that form against each other
  e.g. `feast_*` returns the noun *feast*, the verb *feast*, the adjective *feast*, and some noise
- Parts of speech on their own return any match
  e.g. `kiss _PRON_ mother` should return "kiss your mother," "kiss my mother," etc., but plotted as a single line;

Usage (1/2)

- Enter comma-separated queries to see them plotted against each other
- A wildcard (`*`) returns the top ten matches
  e.g. `the weather is *`
- `gram_INF` returns inflected forms of a lexical form `gram`
  e.g. `seek_INF` returns *sought*, *seek*, *seeking*, *seeks*
- `gram_NOUN`, `gram_VERB`, etc. tries to return only the matching part of speech
  e.g. `feast_VERB` should not find a hit in the sequence "a feast"
- `gram_*` plots all parts of speech for that form against each other
  e.g. `feast_*` returns the noun *feast*, the verb *feast*, the adjective *feast*, and some noise
- Parts of speech on their own return any match
  e.g. `kiss _PRON_ mother` should return "kiss your mother," "kiss my mother," etc., but plotted as a single line;
- Parts of speech preceded by a wildcard are separated out into different matches
  e.g. `kiss *_PRON mother` should return separate statistics on each of "kiss your mother," "kiss my mother," etc.

Usage (2/2)

- Sentence boundaries: `_START_` / `_END_`
- Dependency relations: `weather=>stormy`
- Combined plots: `+`, e.g. `(ale + lager + beer)`
- Subtracted plots: `-`, e.g. `(ale + lager + beer) - (sparkly + sparkly wine + champagne)`
- Divided plots: `/`, e.g. `beer / wine`
- Multiplied plots: `*`, e.g. `fish,(wallaby * 100)`
- Plots from multiple corpora: `:`, e.g. `wizard:eng_2012,wizard:eng_fiction_2012`
- Syntactic "root": `_ROOT_`, e.g. `_ROOT_=>eat` to return clauses with *eat* as the finite verb

What Are the Limitations of the Google Books Ngram Reader for Linguistic Purposes?

How to Do Well on Assignment 4

- Read closely
- Address **every part** of each question
- Address **every relevant aspect** of each issue
- Use reliable sources where not sure:
  - Our textbook (Barber, Beal, and Shaw)
  - Other textbooks in the library system
  - Scholarly articles on the Google Books Ngram Viewer

(But I've covered everything my lectures!)