Words

Outline

- The word
- Word frequency distributions
- Comparing word frequencies

The word

What's a word (Packard 2000; 1/)

- The *orthographic* word: sequences of characters separated by conventionalized delimiters (like whitespace)
- The *sociological* (or *naïve*) word:

Chao (1968:136):

> **...that type of unit intermediate in size between a phoneme and a sentence, which the general non-linguistic public is conscious of, talks about, has an everyday term for, and is practically concerned with in various ways.**

(In English this is roughly the whitespace-delimited word; in Chinese, the character.)

What’s a word (Packard 2000; 2/)

- The *phonological* (or *prosodic*) word: the minimal abstract sequence of sounds (*phones* and *phonemes*) “standing alone” as opposed to leaning on other words
- The *syntactic* (or *grammatical*) word: the minimal phrasal unit; a head, an X^0
- The *lexicographic* (or *lexical*) word: whatever units are used as “headwords” or citation forms in dictionaries

These notions often overlap but are not identical.

Tokenization (1/)

Consider the following sentence (from the Wall St. Journal):

> *Rolls-Royce Motor Cars Inc. said it expects its U.S. sales to remain steady at about 1,200 cars in 1990.*

Is *Rolls-Royce* one word or two? How about *1,200*? How about *1990.*?

This is also language-dependent. It is close to deterministic in English, so we use a series of *regular expressions* to split tokens.

Tokenization (2/)

One convention for English is the Penn Treebank tokenizer rules, which produce:

```
['Rolls-Royce', 'Motor', 'Cars', 'Inc.', 'said', 'it',
'expects', 'its', 'U.S.', 'sales', 'to', 'remain', 'steady',
'at', 'about', '1,200', 'cars', 'in', '1990', '.']
```

- Hyphenated compounds like *Rolls-Royce* are treated as a single token, and
- sentential punctuation like commas and periods are treated as single tokens,
- but, punctuation inside other tokens are **not** treated as separate tokens.

Tokens

Tools like the Penn Treebank tokenizer tend to privilege the syntactic word at the cost of other notions of wordhood.

> E.g., clitics are treated as separate words.
>
> [The] [queen]**['s]** [favorite] [corgi] [barked] [at] [the] [Prime] [Minister][.]

Clitics

Clitics are syntactically word-like, but phonologically dependent: they must "lean" on another word.

An important diagnostic for clitichood (vs. affixhood) is *promiscuous attachment* (Zwicky & Pullum 1983): the "host" does not need to be of a particular category, it merely needs to a phonological word belonging to the appropriate phrase.

English possessive *'s*

The English possessive leans on the right edge of possessor noun phrase:

[*Peter*]*'s mother* (cf. German *Peters Mutter*)
[*The Queen of England*]*'s corgis* (cf. **die Königin von Englands corgis*)
[*The zombie movie we all hated*]*'s director*
[*The woman I saw yesterday in the park next to the Pleurotus ostreatus-infested oak tree*]*'s new hat*

Because virtually any word can be the host of an *'s* proclitic, it makes sense to treat it as a separate token. The tokenizer sacrifices the naïve (or *sociological*) notion of wordhood in favor of *syntactic wordhood* (an X^0, a head).

Challenging languages

Word segmentation is non-trivial in nearly all languages, but it is far more challenging in scripts—e.g., Chinese, Japanese, and Thai—which do not reliably mark word boundaries with space, or in Vietnamese, where whitespace is an unreliable cue to word boundaries.

For these languages, machine learning is usually necessary. One good option is UDPipe and their collection of models for 60 languages.

(Most of the scripts of Europe and the Near East were written like this until the late medieval/early modern era, too.)

Chinese Treebank 9.0

以**组建**在国内外具有实力影响的大公司、大集团为目标进行的资本结构优化，正在给青岛经济带来新的活力。

近两年来，青岛这个中国沿海十四个开放城市之一的城市，在资金及政策上重点支持了五十个名牌产品和五十个重点企业，集中力量发展了电子、机械、石化、橡胶、家电和饮料六大支柱产业。

目前，青岛的资本运营形式已呈兼并、联合、参股、控股、收购等多元化态势。

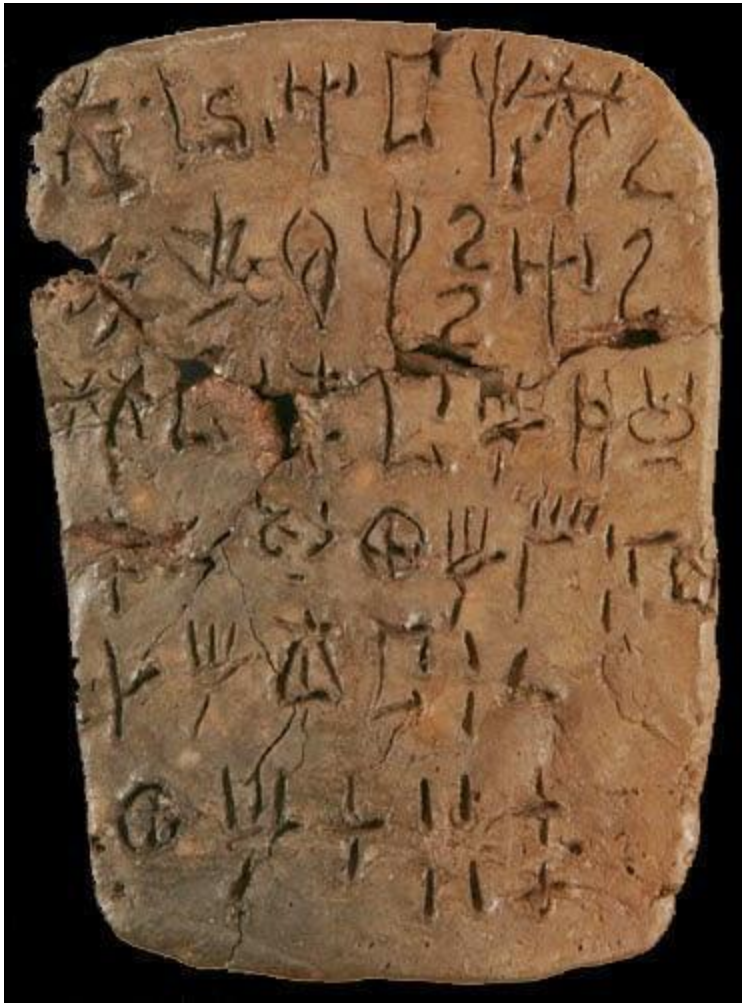
在市场的引导下，一批实力强劲的企业迅速组建大集团。

Thai Wikipedia

ธงไชย แมคอินไตย์ ชื่อเล่น เบิร์ด (เกิด 8 ธันวาคม พ.ศ. 2501) เป็นนักร้อง นักแสดงชาวไทย ได้รับขนานนามว่าเป็น "ซูเปอร์สตาร์เมืองไทย" แรกเข้าวงการบันเทิงเป็นนักแสดงสมทบ ต่อมาได้รับบทพระเอก โดยภาพยนตร์ที่สร้างชื่อเสียงให้กับเขาที่สุดเรื่อง ด้วยรักคือรัก ส่วนละครที่สร้างชื่อเสียงที่สุดของเขาคือบท "โกโบริ" ในละครคู่กรรม ด้านวงการเพลงซึ่งเป็นอาชีพหลักเขาเริ่มต้นจากการประกวดร้องเพลงของสยามกลการ ต่อมาเป็นนักร้องในสังกัดบริษัท จีเอ็มเอ็ม แกรมมี่ จำกัด (มหาชน) ซึ่งประสบความสำเร็จสูงสุดของประเทศไทย มียอดจำหน่ายอยู่ในระดับแนวหน้าของทวีปเอเชียยอดรวมกว่า 25 ล้านชุด

Vietnamese Wikipedia

Thần thoại **Hy Lạp** là tập hợp những huyền thoại và truyền thuyết của người Hy Lạp cổ đại liên quan đến các vị thần, các anh hùng, bản chất của thế giới, và nguồn gốc cũng như ý nghĩa của các tín ngưỡng, nghi lễ tôn giáo của họ. Chúng là một phần của tôn giáo Hy Lạp cổ đại và nay là một phần của một tôn giáo hiện đại lưu hành ở Hy Lạp và trên thế giới gọi là Hellenismos. Các học giả hiện đại tham khảo và nghiên cứu các truyện thần thoại này để rọi sáng vào các thể chế tôn giáo, chính trị Hy Lạp cổ đại, nền văn minh của nó cũng như để tìm hiểu về bản thân sự hình thành huyền thoại.



[Linear A tablet, Zakros, Crete, 1450 BCE.
Image credit: The Antiquated Antiquarian.]

[Latin bronze diploma, Naples, 113/4 C.E.
Image credit: the Metropolitan Museum of Art]

Word frequency distributions

Large numbers of rare events

Word frequency distributions are very sparse. They are not governed by the law of large numbers and therefore cannot be understood in terms of normal statistics.

Instead they are governed by a separate set of statistical laws, those for large numbers of rare events (LNRE).

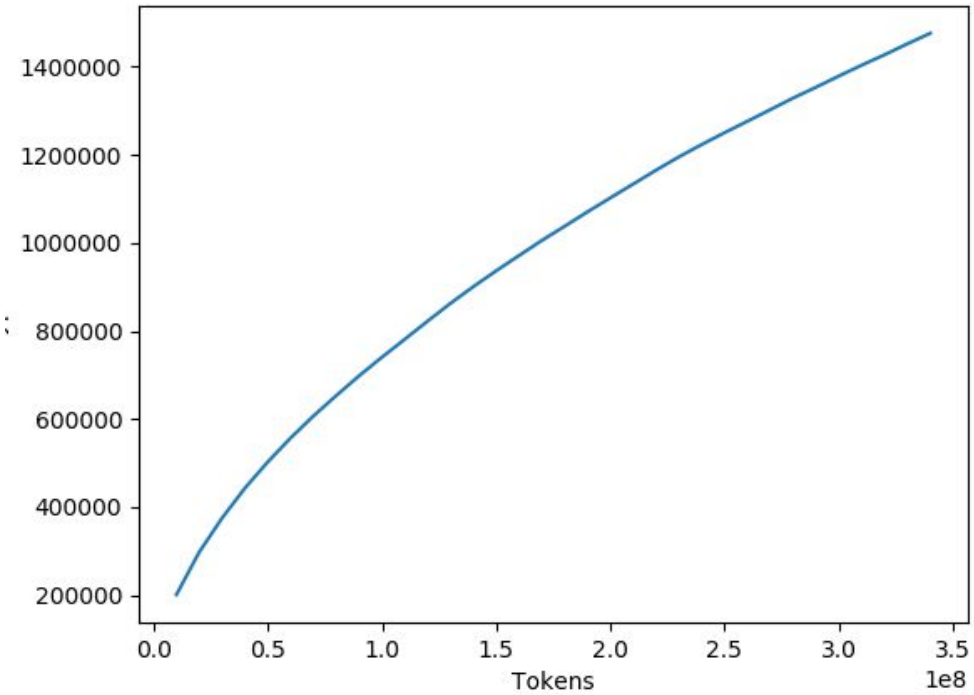
Today’s corpus

2009 English newscrawl data from statmt.org, case-folded and tokenized using the Penn Treebank tokenizer.

- 14.7m sentences
- 347m tokens
- 1.49m types

Most frequent tokens: *the* *,* *.* *to* *of* *and* *a* *in* *that*

Vocabulary growth rate



Hapax legomena

As with all LNRE random variables, word frequency exhibits “few giants” and “many dwarves”.

Among them are the 786k *hapax legomena* (sg. *hapax legomenon*), tokens that only occur once.

In other words, for every 450 or so tokens we see a new token.

It is hard to distinguish between *structural* and *accidental* zeros.

Zipf’s Law: definition

Word frequency r is proportional to the inverse of word rank R. Or:

$$r(C, \alpha) = C / R^{\alpha}$$

where C is a constant sensitive to sample size and α is ≈ -1.

These can be estimated from the linear regression formula:

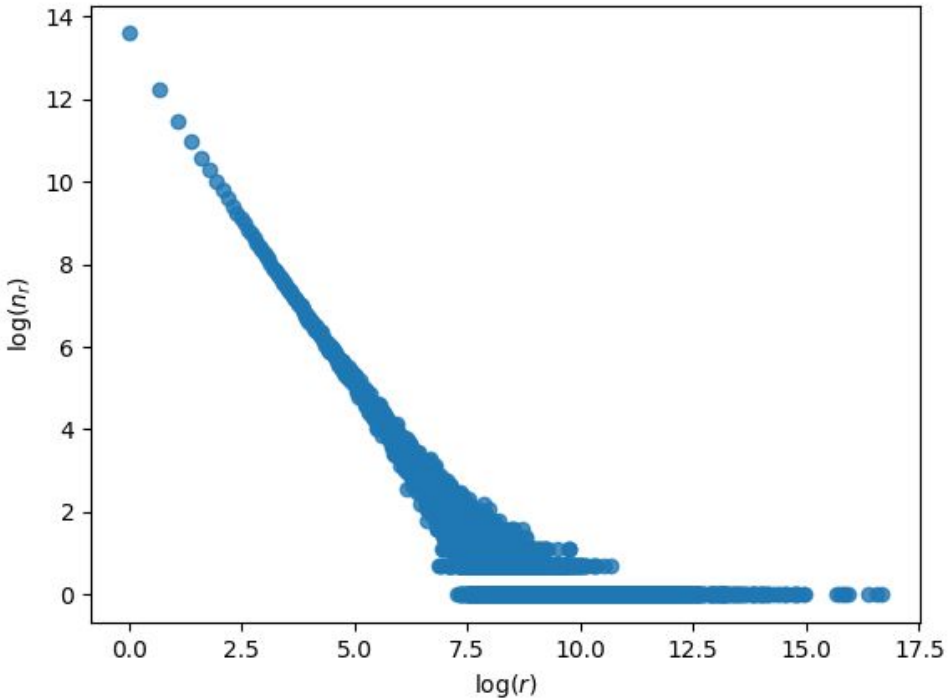
$$\log r \sim \log C + \alpha \log R + \varepsilon$$

where ε is the error term.

Frequency-of-frequency representation (Good 1953)

For many purposes it is easier to think in terms of frequencies of frequencies, so that n_r is the number of types of frequency r.

Zipf's Law (n_r)



Z_r transform (Gale & Sampson 1995)

Noting that the “tail” of the preceding graph is dominated by high-frequency items with small n_r, propose to smooth it out by averaging using neighbors:

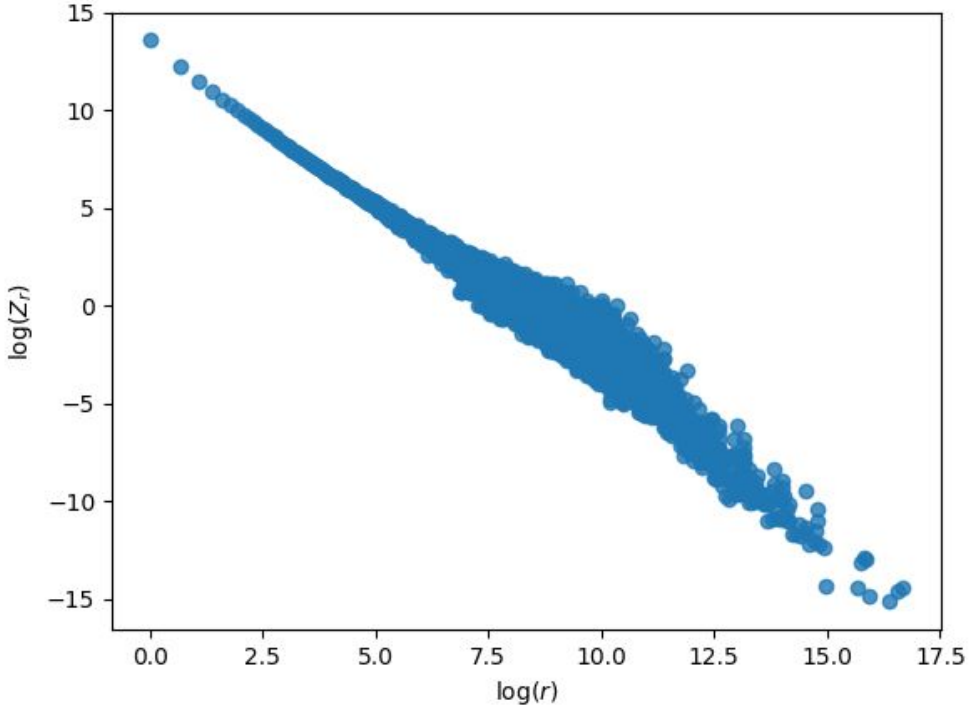
$$Z_i = 2\, n_i \,/\, (r_{i+1} - r_{i-1})$$

and the edge cases:

$$Z_1 = n_1 \,/\, (r_2 - r_1)$$
$$Z_N = n_N \,/\, (r_N - r_{N-1})$$

For small r, the denominator will equal 2 on the left; for high r, it will be large and help to smooth.

Zipf's Law (Z_r)



$\alpha = -1.589$, $R^2 = .9381$

LNRE calculator

Given a TSV file in which the first column are token and the second column are integral counts of those tokens, my LNRE calculator computes some basic statistics and graphs the "Zipf curve".

Some reference frequency distributions are provided here.

README.md

This directory contains various linguistic frequency distributions, represented by two-column TSV files where the first column is the linguistic representation (usually, though not always, a token or word) and the second its frequency.

The LNRE calculator can ingest these files and produce useful descriptive statistics.

Table of contents

Conjugation [edit]

Conjugation of *amar* (See Appendix:Spanish verbs) [hide ▲]

infinitive				**amar**			
gerund				amando			
past participle			**masculine**		**feminine**		
		singular	amado		amada		
		plural	amados		amadas		
		singular			**plural**		
		1st person	**2nd person**	**3rd person**	**1st person**	**2nd person**	**3rd person**
indicative		**yo**	**tú** **vos**	**él/ella/ello** **usted**	**nosotros** **nosotras**	**vosotros** **vosotras**	**ellos/ellas** **ustedes**
	present	amo	amas[tú] amás[vos]	ama	amamos	amáis	aman
	imperfect	amaba	amabas	amaba	amábamos	amabais	amaban
	preterite	amé	amaste	amó	amamos	amasteis	amaron
	future	amaré	amarás	amará	amaremos	amaréis	amarán
	conditional	amaría	amarías	amaría	amaríamos	amaríais	amarían
subjunctive		**yo**	**tú** **vos**	**él/ella/ello** **usted**	**nosotros** **nosotras**	**vosotros** **vosotras**	**ellos/ellas** **ustedes**
	present	ame	ames[tú] amés[vos][2]	ame	amemos	améis	amen
	imperfect (ra)	amara	amaras	amara	amáramos	amarais	amaran
	imperfect (se)	amase	amases	amase	amásemos	amaseis	amasen
	future[1]	amare	amares	amare	amáremos	amareis	amaren
imperative		**—**	**tú** **vos**	**usted**	**nosotros** **nosotras**	**vosotros** **vosotras**	**ustedes**
	affirmative		ama[tú] amá[vos]	ame	amemos	amad	amen
	negative		no ames	no ame	no amemos	no améis	no amen

[1] Mostly obsolete form, now mainly used in legal jargon.
[2] Argentine and Uruguayan *voseo* prefers the *tú* form for the present subjunctive.

Selected combined forms of *amar* [show ▼]

Zipf’s Law: Spanish verb paradigms (Chan 2008)

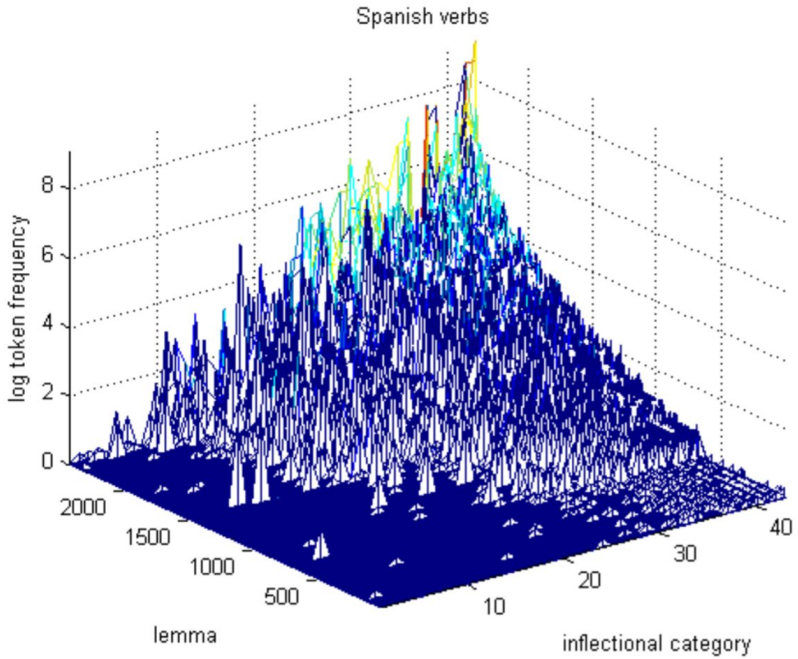


Figure 4.5. Log token frequency by lemma and inflectional category, Spanish verbs.

Comparing word frequencies

Sample corpora

Baseline corpus (to be defined): 2009 English newscrawl data from statmt.org (346m tokens)

Corpus 1: Yahoo! Horoscopes (1.8m tokens)

Corpus 2: King James Version (1.1m tokens)

Example

Words present in horoscopes but not news: *snugglebug*, *wingperson*, *lovewise*, *sugarpie*, *nutsy*, *heartspace*, *patootie*, *you-time*, *overexplain*, ...

Words present in news but not horoscopes: *Obama*, *billion*, *minister*, *London*, *administration*, ...

Words present in the KJV but not news: *calleth*, *transgressings*, *deliveredst*, *everlastingness*, *soothsayings*, *whorish*, *foresaken*, …

From frequencies to probabilities

We can convert a word frequency $c(w)$ to a probability using maximum likelihood estimation:

$$p(w) = c(w) / N$$

where N is the total number of tokens in the corpus.

Probability differences

We *could* use raw probability deltas (e.g., $p_1 - p_2$) for comparison. This has range [−1, 1]. However:

- Maybe *p*'s are (accidental?) zeros.
- High-frequency words will have the most extreme values, e.g.:
  - Words most associated with horoscopes: *you*, *your*, *to*, *a*, *it*, ...
  - Words most associated with the KJV: *shall, he, of, and, the*, ...

From probabilities to odds

The odds of a probability p is simply:

$$O = p \,/\, (1 - p)$$

This has the range $[0, +\infty]$.

For instance, for $p = .9$, $O = 9$, and for $p = .1$, $O = 0.1111$.

From odds to log-odds

Because of the strange range…

e.g., $p < .5$ implies $0 < O < 1$, whereas $p > .5$ implies $1 < O < \infty$,

it is often preferable to work in log-space, where the range is $[-\infty, +\infty]$.

$$\begin{aligned} \log O &= \log p - \log(1 - p) \\ &= \log c - \log(N - c) \end{aligned}$$

For instance, for $p = .9$, $\log(O) = 2.197$, and for $p = .1$, $\log(O) = -2.197$.

From log-odds to log-odds ratios (1/)

To compare two probabilities, we can compute their log-odds ratio, defined as the difference between two log odds. This preserves the [−∞, +∞] range.

For example, for the probabilities .9 (log O = 2.197) and .4 (log O = −.405) the log-odds ratio is 2.602.

From log-odds to log-odds ratios (2/)

Let $c_1(w)$, $c_2(w)$ be the frequencies of some word w in corpus 1 and corpus 2, respectively.

Let N_1, N_2 be the total number of tokens in corpus 1 and corpus 2, respectively. Then:

$$\log O_i(w) = \log[\, c_i(w) \,/\, (N_i - c_i(w)) \,]$$

$$= \log[\, c_i(w) \,] - \log[\, N_i - c_i(w) \,]$$

$$\delta_{i,j}(w) = \log O_i(w) - \log O_j(w)$$

From log-odds to log-odds ratios (3/)

Problems:

- Log-odds (and therefore their ratios) are undefined when counts are zero.
- High-frequency words still have the most extreme values, e.g.:
  - Words most associated with horoscopes: *around*, *comes*, *everyone*, *attention*, *usual*, ...
  - Words most associated with the KJV: *thee*, *Father*, *ye*, *unto*, *Lord*, ...

Advance Access publication February 16, 2009

Political Analysis (2008) 16:372–403
doi:10.1093/pan/mpn018

Fightin' Words: Lexical Feature Selection and Evaluation for Identifying the Content of Political Conflict

Burt L. Monroe
Department of Political Science, Quantitative Social Science Initiative, The Pennsylvania State University, e-mail: burtmonroe@psu.edu (corresponding author)

Michael P. Colaresi
Department of Political Science, Michigan State University, e-mail: colaresi@msu.edu

Kevin M. Quinn
Department of Government and Institute for Quantitative Social Science, Harvard University, e-mail: kevin_quinn@harvard.edu

Entries in the burgeoning ''text-as-data'' movement are often accompanied by lists or visualizations of how word (or other lexical feature) usage differs across some pair or set of documents. These are intended either to establish some target semantic concept (like the content of partisan frames) to estimate word-specific measures that feed forward into another analysis (like locating parties in ideological space) or both. We discuss a variety of techniques for selecting words that capture partisan, or other, differences in political speech and for evaluating the relative importance of those words. We introduce and emphasize several new approaches based on Bayesian shrinkage and regularization. We illustrate the relative utility of these approaches with analyses of partisan, gender, and distributive speech in the U.S. Senate.

From log-odds ratios to informative Dirichlet priors (1/)

We can use a large "background" corpus as a *prior*, an estimate of expected word frequencies.

We do this by adding the background corpus counts to both the numerator and denominator, "shrinking" the probabilities/odds towards the prior probabilities/odds.

From log-odds ratios to informative Dirichlet priors (2/)

Let $c_1(w)$, $c_2(w)$, $c_3(w)$ be the frequencies of some word w in corpora 1-3, respectively.

Let N_1, N_2, N_3, be the total number of tokens in corpora 1-3, respectively. Then:

$$\log O_{i,k}(w) = \log[\ (c_i(w) + c_k(w)) \ / \ (N_i + N_k - c_i(w) - c_k(w))\ ]$$

$$= \log[\ c_i(w) + c_k(w)] - \log[\ N_i + N_k - c_i(w) - c_k(w)\ ]$$

$$\delta_{i,j,k}(w) = \log O_{i,k}(w) - \log O_{j,k}(w)$$

NB: this is defined even if $c_1(w)$ or $c_2(w) = 0$, so long as $c_3(w) > 0$.

From log-odds ratios to informative Dirichlet priors (3/)

- Words associated with horoscopes: *you*, *new, feel, love, time*...
- Words associated with the KJV: *behold, king, God, Lord, children*...

From log-odds ratios to informative Dirichlet priors (4/)

Finally, we can scale the log-odds ratios to take their variance into account.

The sample variance for word *w* is given by:

$$\sigma^2_{i,j,k}(w) = 1 / [\, c_i(w) + N_k \,] + 1 / [\, c_j(w) + N_k \,]$$

Then a *z*-scored version of the log-odds ratio is given by:

$$z_{i,j,k}(w) = \delta_{i,j,k} / \mathrm{sqrt}[\, \sigma^2_{i,j,k}(w) \,]$$

From log-odds ratios to informative Dirichlet priors (5/)

- Words associated with horoscopes: *you*, *her*, *about, people, year*, ...
- Words associated with the KJV: *unmerciful*, *greediness*, *defile*, *glutton*, *gnash*, ...

Implementations

It's relatively easy to implement the Fightin' Words method yourself:

- sentence-split, tokenize, and (optionally) case-fold your data, then
- use `collections.Counter` objects to collect the raw counts,
- then use `math.log` and basic arithmetic take care of the rest.

I have a simple Cython-based implementation available here.

Cornell's ConvoKit also has an implementation (not yet tested).

Questions?

For next week

- Read Jurafsky & Martin (draft 3rd edition) section 20.5.1, which describes the Fightin' Words method.
  - Feel free to browse the whole chapter: it's interesting.
- Make sure your Conda installation is in good working order:
  - If your Conda is ancient or malfunctioning, just delete it and start over.
  - Make sure `python` is an alias to Python 3.8. (Try `python --version` to confirm.)