Roadmap

Lexical statistics: the basics

Zipf's law
- Typical frequency patterns
- Zipf's law
- Consequences

Applications
- Productivity in morphology
- Productivity beyond morphology
- Lexical richness
- Conclusion and outlook

Counting Words:
Introduction

Marco Baroni & Stefan Evert

Málaga, 7 August 2006

Roadmap

- Introduction and motivation

Roadmap

- Introduction and motivation
- LNRE modeling: soft

Roadmap

- Introduction and motivation
- LNRE modeling: soft
- LNRE modeling: hard

Roadmap

- Introduction and motivation
- LNRE modeling: soft
- LNRE modeling: hard
- Playtime!

Roadmap

- Introduction and motivation
- LNRE modeling: soft
- LNRE modeling: hard
- Playtime!
- The bad news and outlook

Outline

Roadmap

Lexical statistics: the basics

Zipf's law
Typical frequency patterns
Zipf's law
Consequences

Applications
Productivity in morphology
Productivity beyond morphology
Lexical richness
Conclusion and outlook

Roadmap

Lexical statistics: the basics

Zipf's law

Applications

Lexical statistics

Zipf 1949/1961, Baayen 2001, Evert 2005

Roadmap

Lexical statistics: the basics

Zipf's law

Typical frequency patterns

Zipf's law

Consequences

Applications

Productivity in morphology

Productivity beyond morphology

Lexical richness

Conclusion and outlook

- Statistical study of distribution of **types** (words and other units) in texts
- Different from other categorical data because of extreme richness of types

Basic terminology

- N: sample/corpus size, number of **tokens** in the sample

Basic terminology

- N: sample/corpus size, number of **tokens** in the sample
- V: vocabulary size, number of distinct **types** in the sample

Basic terminology

- N: sample/corpus size, number of **tokens** in the sample
- V: vocabulary size, number of distinct **types** in the sample
- V_m: type count of **spectrum element** m, number of types in the sample with token frequency m

Basic terminology

- N: sample/corpus size, number of **tokens** in the sample
- V: vocabulary size, number of distinct **types** in the sample
- V_m: type count of **spectrum element** m, number of types in the sample with token frequency m
- V_1: **hapax legomena** count, number of types that occur only once in the sample (for hapaxes, Count(**types**) = Count(**tokens**))

Basic terminology

- N: sample/corpus size, number of **tokens** in the sample
- V: vocabulary size, number of distinct **types** in the sample
- V_m: type count of **spectrum element** m, number of types in the sample with token frequency m
- V_1: **hapax legomena** count, number of types that occur only once in the sample (for hapaxes, Count(**types**) = Count(**tokens**))
- A sample: `a b b c a a b a`

Basic terminology

- N: sample/corpus size, number of **tokens** in the sample
- V: vocabulary size, number of distinct **types** in the sample
- V_m: type count of **spectrum element** m, number of types in the sample with token frequency m
- V_1: **hapax legomena** count, number of types that occur only once in the sample (for hapaxes, Count(**types**) = Count(**tokens**))
- A sample: `a b b c a a b a`
- N: 8; V: 3; V_1: 1

Rank/frequency profile

- The sample: `a b b c a a b a d`

Rank/frequency profile

- The sample: `a b b c a a b a d`
- Frequency list ordered by decreasing frequency

t	f
a	4
b	3
c	1
d	1

Rank/frequency profile

- The sample: `a b b c a a b a d`
- Frequency list ordered by decreasing frequency

t	f
a	4
b	3
c	1
d	1

- Replace type labels with ranks to obtain rank/frequency profile:

r	f
1	4
2	3
3	1
4	1

Rank/frequency profile

- The sample: `a b b c a a b a d`
- Frequency list ordered by decreasing frequency

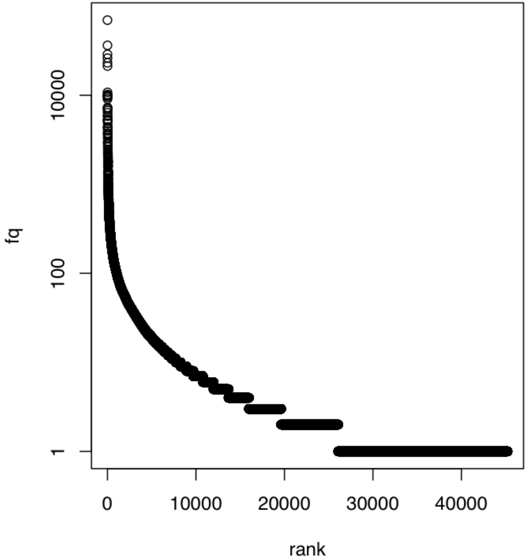
t	f
a	4
b	3
c	1
d	1

- Replace type labels with ranks to obtain rank/frequency profile:

r	f
1	4
2	3
3	1
4	1

- Allows expression of frequency in function of rank of type

Rank/frequency profile of Brown corpus



Frequency spectrum

- The sample: `a b b c a a b a d`

Frequency spectrum

- The sample: `a b b c a a b a d`
- Frequency classes: 1 (`c`, `d`), 3 (`b`), 4 (`a`)

Frequency spectrum

- The sample: `a b b c a a b a d`
- Frequency classes: 1 (`c`, `d`), 3 (`b`), 4 (`a`)
- Frequency spectrum:

m	V_m
1	2
3	1
4	1

Rank/frequency profiles and frequency spectra

Introduction

Baroni & Evert

Roadmap

Lexical statistics: the basics

Zipf's law

Typical frequency patterns

Zipf's law

Consequences

Applications

Productivity in morphology

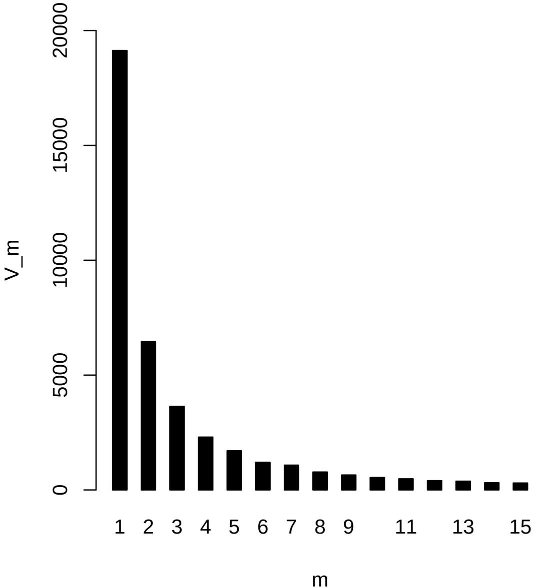
Productivity beyond morphology

Lexical richness

Conclusion and outlook

- From rank/frequency profile to spectrum: count occurrences of each f in profile to obtain V_f values of corresponding spectrum elements
- From spectrum to rank/frequency profile: given highest f (i.e., m) in a spectrum, the ranks 1 to V_f in the corresponding rank/frequency profile will have frequency f, the ranks $V_f + 1$ to $V_f + V_g$ (where g is the second highest frequency in the spectrum) will have frequency g, etc.

Frequency spectrum of Brown corpus



Vocabulary growth curve

- The sample: `a b b c a a b a`

Vocabulary growth curve

- The sample: a b b c a a b a
- N: 1, V: 1, V_1: 1

Vocabulary growth curve

- The sample: a b b c a a b a
- N: 1, V: 1, V_1: 1
- N: 3, V: 2, V_1: 1

Vocabulary growth curve

- The sample: a b b c a a b a
- N: 1, V: 1, V_1: 1
- N: 3, V: 2, V_1: 1
- N: 5, V: 3, V_1: 1

Vocabulary growth curve

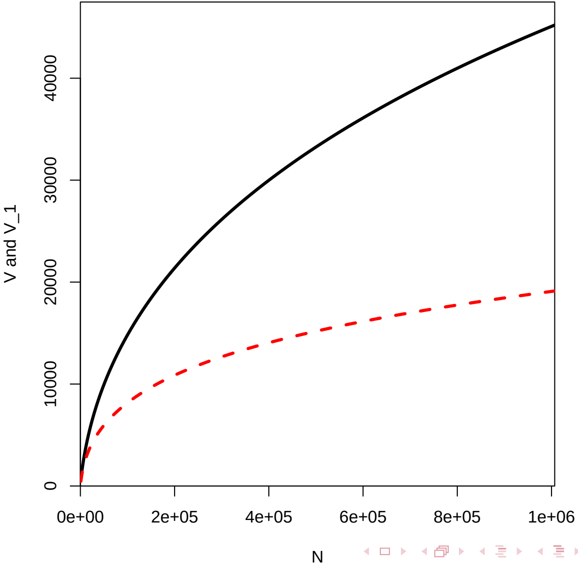
- The sample: `a b b c a a b a`
- N: 1, V: 1, V_1: 1
- N: 3, V: 2, V_1: 1
- N: 5, V: 3, V_1: 1
- N: 8, V: 3, V_1: 1

Vocabulary growth curve

- The sample: `a b b c a a b a`
- N: 1, V: 1, V_1: 1
- N: 3, V: 2, V_1: 1
- N: 5, V: 3, V_1: 1
- N: 8, V: 3, V_1: 1
- (Most VGCs on our slides smoothed with **binomial interpolation**)

Vocabulary growth curve of Brown corpus

With V_1 growth in red



Outline

Roadmap

Lexical statistics: the basics

Zipf's law
- Typical frequency patterns
- Zipf's law
- Consequences

Applications
- Productivity in morphology
- Productivity beyond morphology
- Lexical richness
- Conclusion and outlook

Roadmap

Lexical statistics: the basics

Zipf's law

Applications

Typical frequency patterns

Top and bottom ranks in the Brown corpus

top frequencies			bottom frequencies		
rank	**fq**	**word**	**rank range**	**fq**	**randomly selected examples**
1	62642	the	7967-8522	10	recordings undergone privileges
2	35971	of	8523-9236	9	Leonard indulge creativity
3	27831	and	9237-10042	8	unnatural Lolotte authenticity
4	25608	to	10043-11185	7	diffraction Augusta postpone
5	21883	a	11186-12510	6	uniformly throttle agglutinin
6	19474	in	12511-14369	5	Bud Councilman immoral
7	10292	that	14370-16938	4	verification gleamed groin
8	10026	is	16939-21076	3	Princes nonspecifically Arger
9	9887	was	21077-28701	2	blitz pertinence arson
10	8811	for	28702-53076	1	Salaries Evensen parentheses

Typical frequency patterns
BNC

Introduction

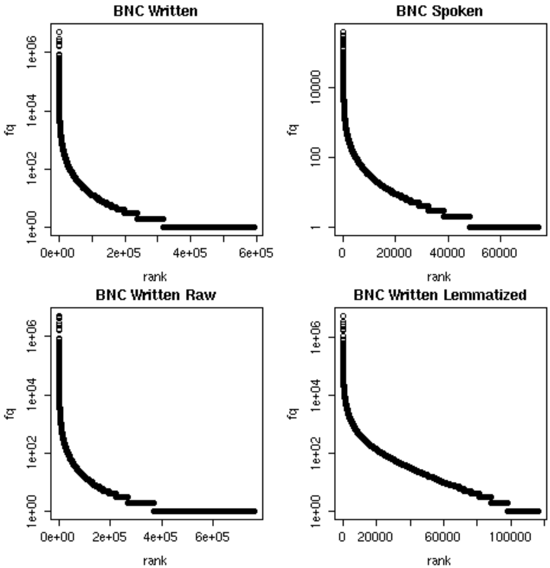
Baroni & Evert

Roadmap

Lexical statistics: the basics

Zipf's law
- Typical frequency patterns
- Zipf's law
- Consequences

Applications
- Productivity in morphology
- Productivity beyond morphology
- Lexical richness
- Conclusion and outlook



Typical frequency patterns
Other corpora

Introduction

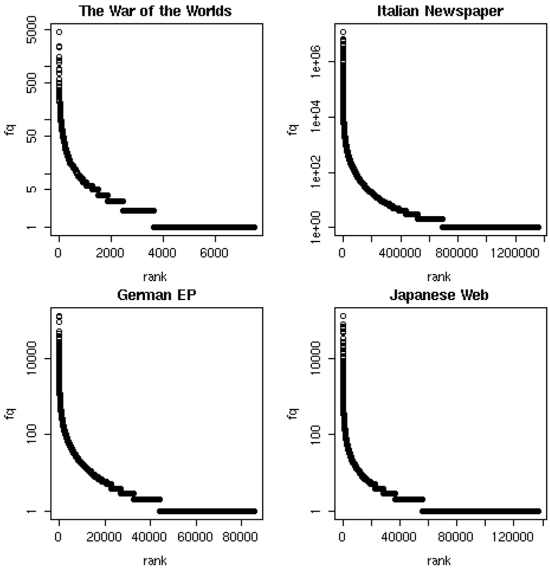
Baroni & Evert

Roadmap

Lexical statistics: the basics

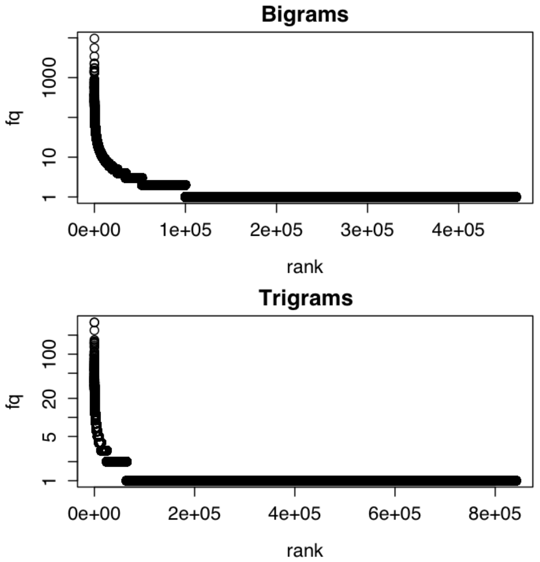
Zipf's law
- Typical frequency patterns
- Zipf's law
- Consequences

Applications
- Productivity in morphology
- Productivity beyond morphology
- Lexical richness
- Conclusion and outlook

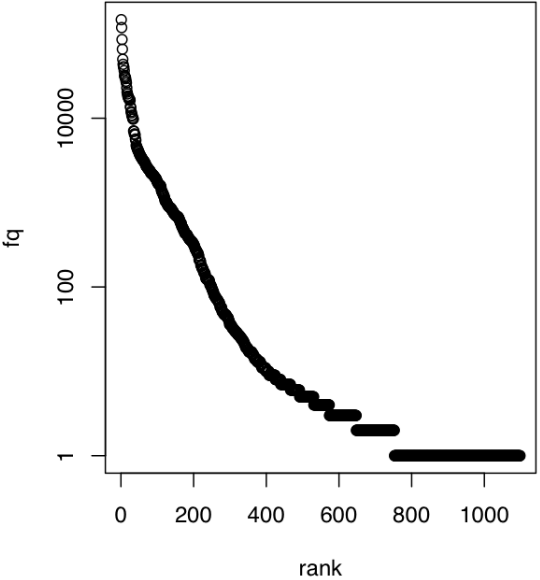


Typical frequency patterns

Brown bigrams and trigrams



Typical frequency patterns
The Italian prefix *ri-* in the *la Repubblica* corpus



Zipf's law

- Language after language, corpus after corpus, linguistic type after linguistic type. . .

Roadmap

Lexical statistics: the basics

Zipf's law

Typical frequency patterns

Zipf's law

Consequences

Applications

Productivity in morphology

Productivity beyond morphology

Lexical richness

Conclusion and outlook

Zipf's law

- Language after language, corpus after corpus, linguistic type after linguistic type…
- same "few giants, many dwarves" pattern is encountered

Zipf's law

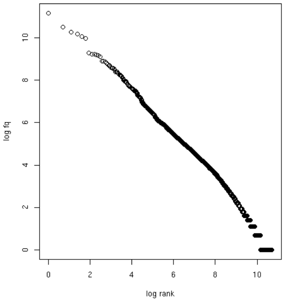
- Language after language, corpus after corpus, linguistic type after linguistic type...
- same "few giants, many dwarves" pattern is encountered
- Similarity of plots suggests that relation between rank and frequency could be captured by a law

Zipf's law

- Language after language, corpus after corpus, linguistic type after linguistic type...
- same "few giants, many dwarves" pattern is encountered
- Similarity of plots suggests that relation between rank and frequency could be captured by a law
- Nature of relation becomes clearer if we plot $\log f$ in function of $\log r$

Zipf's law

- Language after language, corpus after corpus, linguistic type after linguistic type...
- same "few giants, many dwarves" pattern is encountered
- Similarity of plots suggests that relation between rank and frequency could be captured by a law
- Nature of relation becomes clearer if we plot $\log f$ in function of $\log r$



Zipf's law

- Straight line in double-logarithmic space corresponds to **power law** for original variables

Zipf's law

- Straight line in double-logarithmic space corresponds to **power law** for original variables
- This leads to Zipf's (1949, 1965) famous law:

$$f(w) = \frac{C}{r(w)^a}$$

Zipf's law

- Straight line in double-logarithmic space corresponds to **power law** for original variables
- This leads to Zipf's (1949, 1965) famous law:

$$f(w) = \frac{C}{r(w)^a}$$

- With $a = 1$ and $C = 60,000$, Zipf's law predicts that most frequent word has frequency 60,000; second most frequent word has frequency 30,000; third word has frequency 20,000...

Zipf's law

- Straight line in double-logarithmic space corresponds to **power law** for original variables
- This leads to Zipf's (1949, 1965) famous law:

$$f(w) = \frac{C}{r(w)^a}$$

- With $a = 1$ and $C = 60,000$, Zipf's law predicts that most frequent word has frequency 60,000; second most frequent word has frequency 30,000; third word has frequency 20,000...
- and long tail of 80,000 words with frequency between 1.5 and 0.5

Zipf's law
Logarithmic version

- **Zipf's power law**:

$$f(w) = \frac{C}{r(w)^a}$$

Zipf's law
Logarithmic version

- **Zipf's power law**:

$$f(w) = \frac{C}{r(w)^a}$$

- If we take logarithm of both sides, we obtain:

$$\log f(w) = \log C - a \log r(w)$$

Zipf's law
Logarithmic version

- **Zipf's power law**:

$$f(w) = \frac{C}{r(w)^a}$$

- If we take logarithm of both sides, we obtain:

$$\log f(w) = \log C - a \log r(w)$$

- I.e., Zipf's law predicts that rank/frequency profiles are straight lines in double logarithmic space, which, we saw, is a reasonable approximation

Zipf's law
Logarithmic version

- **Zipf's power law**:

$$f(w) = \frac{C}{r(w)^a}$$

- If we take logarithm of both sides, we obtain:

$$\log f(w) = \log C - a \log r(w)$$

- I.e., Zipf's law predicts that rank/frequency profiles are straight lines in double logarithmic space, which, we saw, is a reasonable approximation
- Best fit a and C can be found with least squares method

Zipf's law
Logarithmic version

- **Zipf's power law**:

$$f(w) = \frac{C}{r(w)^a}$$

- If we take logarithm of both sides, we obtain:

$$\log f(w) = \log C - a \log r(w)$$

- I.e., Zipf's law predicts that rank/frequency profiles are straight lines in double logarithmic space, which, we saw, is a reasonable approximation
- Best fit a and C can be found with least squares method
- Provides intuitive interpretation of a and C:
  - a is **slope** determining how fast log frequency decreases with log rank
  - $\log C$ is **intercept**, i.e., predicted log frequency of word with rank 1 (log rank 0), i.e., most frequent word

Introduction

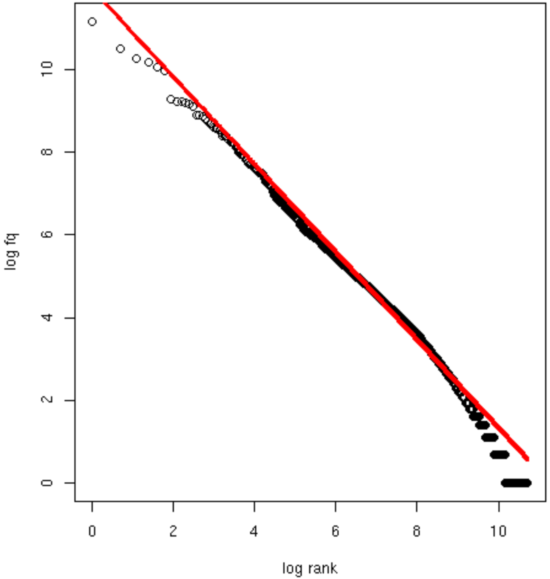
Baroni & Evert

Roadmap

Lexical statistics: the basics

Zipf's law
- Typical frequency patterns
- Zipf's law
- Consequences

Applications
- Productivity in morphology
- Productivity beyond morphology
- Lexical richness
- Conclusion and outlook



Fit of Zipf's law

- At right edge (low frequencies):

Fit of Zipf's law

- At right edge (low frequencies):
  - "Bell-bottom" pattern expected as we are fitting continuous model to discrete frequencies

Fit of Zipf's law

Introduction

Baroni & Evert

Roadmap

Lexical statistics: the basics

Zipf's law

Typical frequency patterns

Zipf's law

Consequences

Applications

Productivity in morphology

Productivity beyond morphology

Lexical richness

Conclusion and outlook

- At right edge (low frequencies):
  - "Bell-bottom" pattern expected as we are fitting continuous model to discrete frequencies
  - More worryingly, in large corpora frequency drops more rapidly than predicted by Zipf's law

Fit of Zipf's law

- At right edge (low frequencies):
  - "Bell-bottom" pattern expected as we are fitting continuous model to discrete frequencies
  - More worryingly, in large corpora frequency drops more rapidly than predicted by Zipf's law
- At left edge (high frequencies):

Fit of Zipf's law

- At right edge (low frequencies):
  - "Bell-bottom" pattern expected as we are fitting continuous model to discrete frequencies
  - More worryingly, in large corpora frequency drops more rapidly than predicted by Zipf's law
- At left edge (high frequencies):
  - Highest frequencies lower than predicted

Fit of Zipf's law

- At right edge (low frequencies):
  - "Bell-bottom" pattern expected as we are fitting continuous model to discrete frequencies
  - More worryingly, in large corpora frequency drops more rapidly than predicted by Zipf's law
- At left edge (high frequencies):
  - Highest frequencies lower than predicted → Mandelbrot's correction

Zipf-Mandelbrot's law
Mandelbrot 1953

- Mandelbrot's extra parameter:

$$f(w) = \frac{C}{(r(w) + b)^a}$$

Roadmap

Lexical statistics: the basics

Zipf's law
Typical frequency patterns
Zipf's law
Consequences

Applications
Productivity in morphology
Productivity beyond morphology
Lexical richness
Conclusion and outlook

Zipf-Mandelbrot's law

Mandelbrot 1953

- Mandelbrot's extra parameter:

$$f(w) = \frac{C}{(r(w) + b)^a}$$

- Zipf's law is special case with $b = 0$

Zipf-Mandelbrot's law
Mandelbrot 1953

- Mandelbrot's extra parameter:

$$f(w) = \frac{C}{(r(w) + b)^a}$$

- Zipf's law is special case with $b = 0$
- Assuming $a = 1$, $C = 60,000$, $b = 1$:
  - For word with rank 1, Zipf's law predicts frequency of 60,000; Mandelbrot's variation predicts frequency of 30,000
  - For word with rank 1,000, Zipf's law predicts frequency of 60; Mandelbrot's variation predicts frequency of 59.94

Zipf-Mandelbrot's law
Mandelbrot 1953

- Mandelbrot's extra parameter:

$$f(w) = \frac{C}{(r(w) + b)^a}$$

- Zipf's law is special case with $b = 0$
- Assuming $a = 1$, $C = 60,000$, $b = 1$:
  - For word with rank 1, Zipf's law predicts frequency of 60,000; Mandelbrot's variation predicts frequency of 30,000
  - For word with rank 1,000, Zipf's law predicts frequency of 60; Mandelbrot's variation predicts frequency of 59.94
- No longer a straight line in double logarithmic space; finding best fit harder than least squares

Zipf-Mandelbrot's law

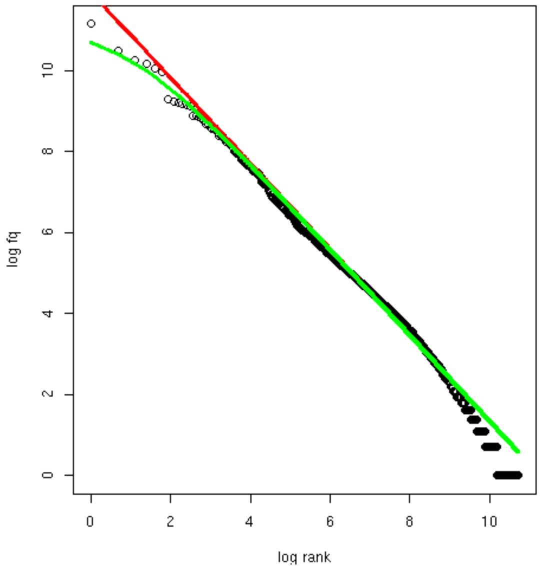
Mandelbrot 1953

- Mandelbrot's extra parameter:

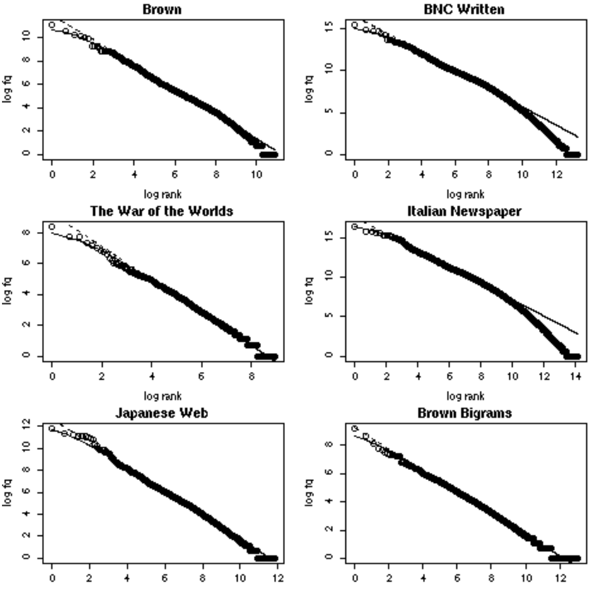
$$f(w) = \frac{C}{(r(w) + b)^a}$$

- Zipf's law is special case with $b = 0$
- Assuming $a = 1$, $C = 60,000$, $b = 1$:
  - For word with rank 1, Zipf's law predicts frequency of 60,000; Mandelbrot's variation predicts frequency of 30,000
  - For word with rank 1,000, Zipf's law predicts frequency of 60; Mandelbrot's variation predicts frequency of 59.94
- No longer a straight line in double logarithmic space; finding best fit harder than least squares
- Zipf-Mandelbrot's law is basis of LNRE statistical models we will introduce

Mandelbrot's adjustment
Fitting the Brown rank/frequency profile



More fits


Brown
BNC Written
The War of the Worlds
Italian Newspaper
Japanese Web
Brown Bigrams
log fq
log rank

A few mildly interesting things about Zipf(-Mandelbrot)'s law

- a is often close to 1 for word frequency distributions (hence simplified version: $f = C/r$, and -1 slope in log-log space)

A few mildly interesting things about Zipf(-Mandelbrot)'s law

- a is often close to 1 for word frequency distributions (hence simplified version: $f = C/r$, and -1 slope in log-log space)
- Zipf's law also provides good fit to frequency spectra

A few mildly interesting things about Zipf(-Mandelbrot)'s law

- a is often close to 1 for word frequency distributions (hence simplified version: $f = C/r$, and -1 slope in log-log space)
- Zipf's law also provides good fit to frequency spectra
- Monkey languages display Zipf's law (intuition: few short words have very high chances to be generated; long tail of highly unlikely long words)

Roadmap
Lexical statistics: the basics
Zipf's law
Typical frequency patterns
Zipf's law
Consequences
Applications
Productivity in morphology
Productivity beyond morphology
Lexical richness
Conclusion and outlook

A few mildly interesting things about Zipf(-Mandelbrot)'s law

- a is often close to 1 for word frequency distributions (hence simplified version: $f = C/r$, and -1 slope in log-log space)
- Zipf's law also provides good fit to frequency spectra
- Monkey languages display Zipf's law (intuition: few short words have very high chances to be generated; long tail of highly unlikely long words)
- Zipf's law is everywhere (Li 2002)

Consequences

- Data sparseness

Introduction

Baroni & Evert

Roadmap

Lexical statistics: the basics

Zipf's law

Typical frequency patterns

Zipf's law

Consequences

Applications

Productivity in morphology

Productivity beyond morphology

Lexical richness

Conclusion and outlook

Consequences

- Data sparseness
- Standard statistics, normal approximation not appropriate for lexical type distributions

Consequences

- Data sparseness
- Standard statistics, normal approximation not appropriate for lexical type distributions
- V is not stable, will grow with sample size, we need special methods to estimate V and related quantities at arbitrary sizes (including V of whole type population)

Consequences

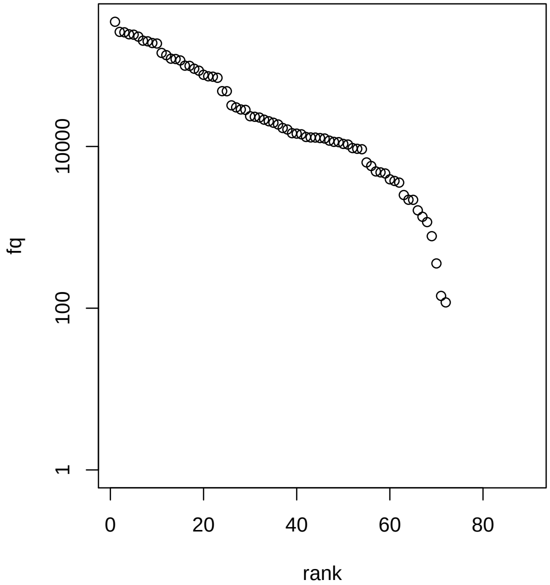
- Data sparseness
- Standard statistics, normal approximation not appropriate for lexical type distributions
- V is not stable, will grow with sample size, we need special methods to estimate V and related quantities at arbitrary sizes (including V of whole type population)

V, sample size and the Zipfian distribution

- Significant tail of hapax legomena indicates that chances of encountering new type if we keep sampling are high
- Zipfian distribution implies vocabulary curve that is still growing at largest sample size

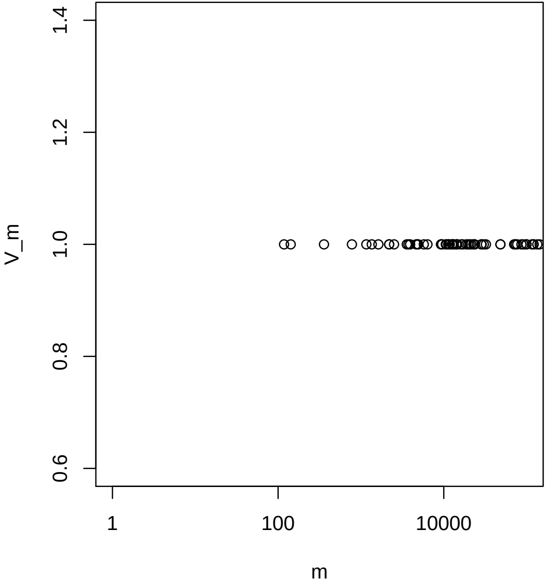
Pronouns in Italian (*la Repubblica*)

Rank/frequency profile



Pronouns in Italian

Frequency spectrum



Introduction

Baroni & Evert

Roadmap

Lexical statistics: the basics

Zipf's law

Typical frequency patterns

Zipf's law

Consequences

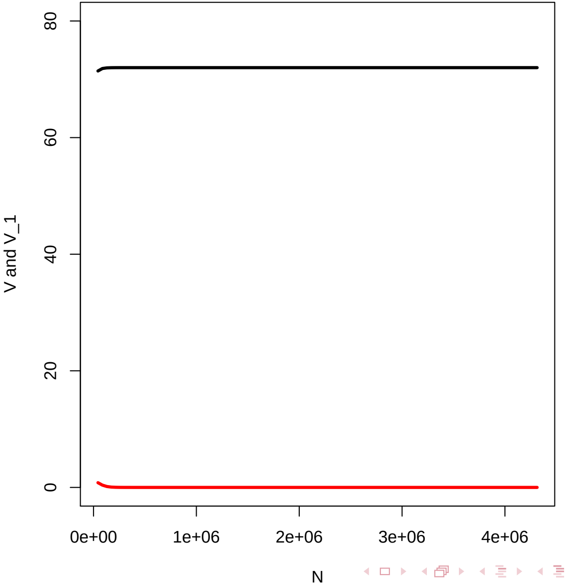
Applications

Productivity in morphology

Productivity beyond morphology

Lexical richness

Conclusion and outlook



Introduction

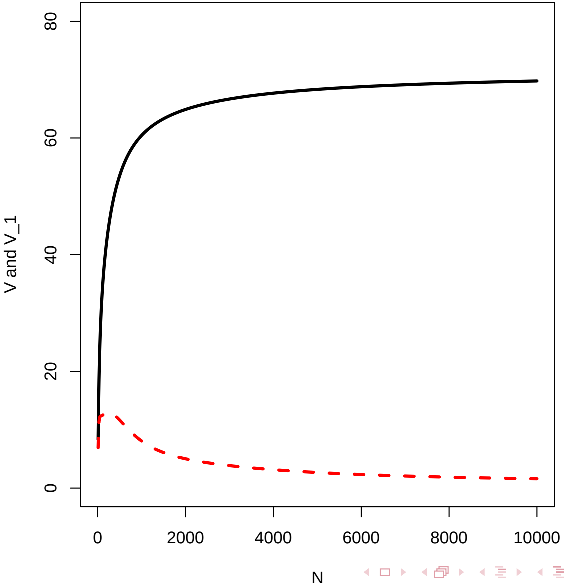
Baroni & Evert

Roadmap

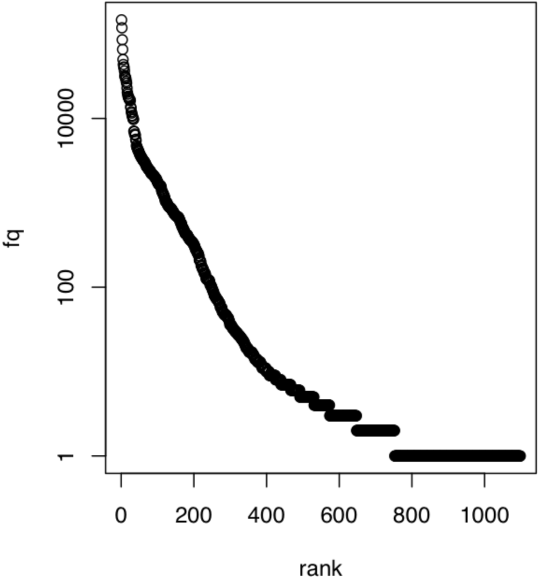
Lexical statistics: the basics

Zipf's law
- Typical frequency patterns
- Zipf's law
- Consequences

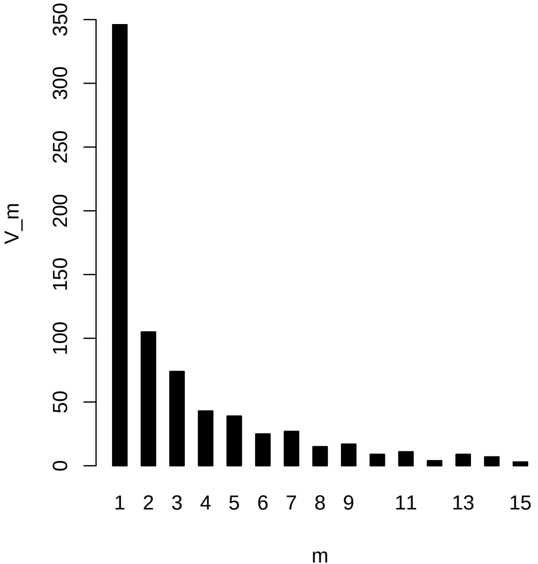
Applications
- Productivity in morphology
- Productivity beyond morphology
- Lexical richness
- Conclusion and outlook



ri- in Italian (*la Repubblica*)
Rank/frequency profile

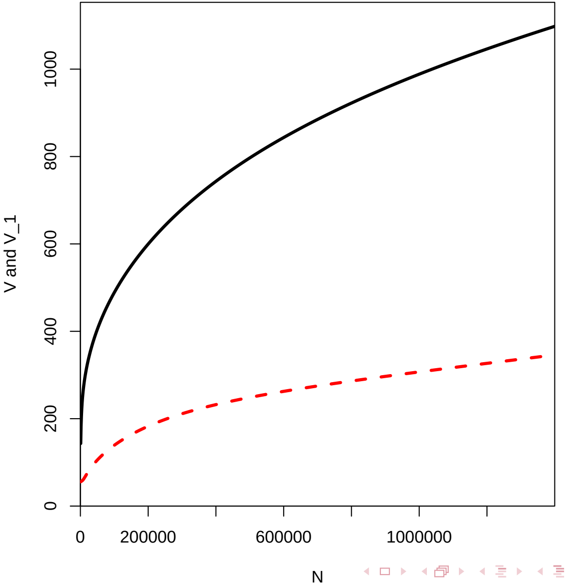


ri- in Italian
Frequency spectrum



ri- in Italian

Vocabulary growth curve



Roadmap

Lexical statistics: the basics

Zipf's law
- Typical frequency patterns
- Zipf's law
- Consequences

Applications
- Productivity in morphology
- Productivity beyond morphology
- Lexical richness
- Conclusion and outlook

Outline

Roadmap

Lexical statistics: the basics

Zipf's law

Applications

Applications

- Productivity (in morphology and elsewhere)
- Lexical richness (in stylometry, language acquisition/pathology and elsewhere)
- Extrapolation of type counts and type frequency distribution for practical NLP purposes (e.g., estimating proportion of OOV words, typos, etc.)
- ... (e.g., Good-Turing smoothing, prior distribution for Bayesian language modeling)

Productivity

- In many linguistic problems, rate of growth of VGC is interesting issue in itself
- Baayen (1989 and later) makes link between linguistic notion of productivity and vocabulary growth rate

Productivity in morphology: the classic definition

Schultink (1961), translated by Booij

Productivity as morphological phenomenon is the possibility which language users have to form an in principle uncountable number of new words unintentionally, by means of a morphological process which is the basis of the form-meaning correspondence of some words they know.

V as a measure of productivity

- Comparable for same N only!

V as a measure of productivity

- Comparable for same N only!
- Good first approximation, but it is measuring attestedness, not potential:

V as a measure of productivity

- Comparable for same N only!
- Good first approximation, but it is measuring attestedness, not potential:
  - (According to rough BNC counts) *de-* verbs have V of 141, *un-* verbs have V of 119, contra our intuition

V as a measure of productivity

- Comparable for same N only!
- Good first approximation, but it is measuring attestedness, not potential:
  - (According to rough BNC counts) *de-* verbs have V of 141, *un-* verbs have V of 119, contra our intuition
  - We want productivity index of pronouns to be 0, not 72!

Baayen's $\mathscr{P}$

- Operationalize **productivity** of a process as probability that the next token created by the process that we sample is a new word
- This is same as probability that next token in sample is hapax legomenon
- Thus, we can estimate probability of sampling a new word as relative frequency of hapax legomena in our sample:

$$\mathscr{P} = \frac{V_1}{N}$$

Baayen's $\mathscr{P}$

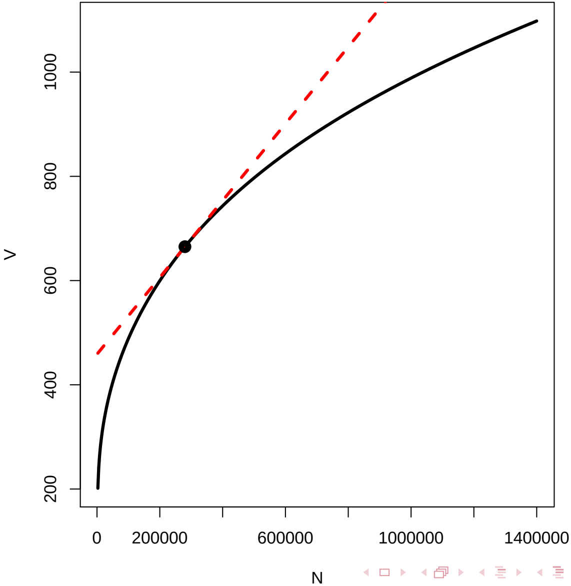
$$\mathscr{P} = \frac{V_1}{N}$$

- Probability to sample token representing type we will never encounter again (token labeled "hapax") at first stage of sampling (when we are at the beginning of N-token-sample) is given by the proportion of hapaxes in the whole N-token-sample divided by the total number of tokens in the sample
- Thus, this must also be probability that *last* token sampled represents new type
- $\mathscr{P}$ as productivity measure matches intuition that productivity should measure *potential* of process to generate new forms

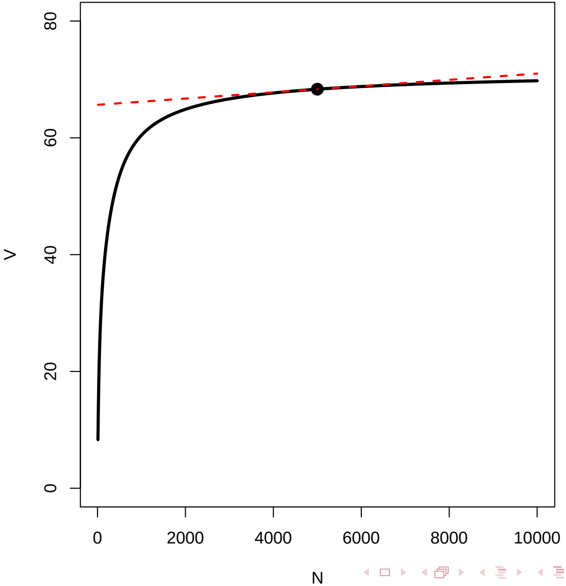
$\mathscr{P}$ as vocabulary growth rate

- $\mathscr{P}$ measures the potentiality of growth of V in a very literal way, i.e., it is the growth rate of V, the rate at which vocabulary size increases
- $\mathscr{P}$ is (approximation to) the *derivative* of V at N, i.e., the slope of the tangent to the vocabulary growth curve at N (Baayen 2001, pp. 49-50)
- Again, "rate of growth" of vocabulary generated by word formation process seems good match for intuition about productivity of word formation process

ri- in Italian *la Repubblica* corpus



Pronouns in Italian *la Repubblica* corpus



Baayen's $\mathscr{P}$ and intuition

class	V	V_1	N	$\mathscr{P}$
it. ri-	1098	346	1,399,898	0.00025
it. pronouns	72	0	4,313,123	0
en. un-	119	25	7,618	.00328
en. de-	141	16	86,130	.000185

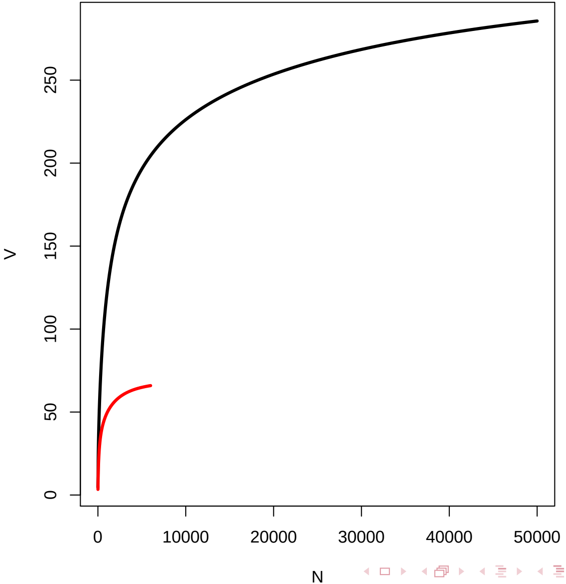
$\mathscr{P}$ and sample size

- We saw that as N increases, V also increases (for at-least-mildly-productive processes)

$\mathscr{P}$ and sample size

- We saw that as N increases, V also increases (for at-least-mildly-productive processes)
- Thus, V cannot be compared at different Ns

V and *N*
English *re-* and *mis-*



$\mathscr{P}$ and sample size

Introduction

Baroni & Evert

Roadmap

Lexical statistics: the basics

Zipf's law

Typical frequency patterns

Zipf's law

Consequences

Applications

Productivity in morphology

Productivity beyond morphology

Lexical richness

Conclusion and outlook

- We saw that as N increases, V also increases (for at-least-mildly-productive processes)
- Thus, V cannot be compared at different Ns

$\mathscr{P}$ and sample size

- We saw that as N increases, V also increases (for at-least-mildly-productive processes)
- Thus, V cannot be compared at different Ns
- However, growth rate is also systematically decreasing as N becomes larger

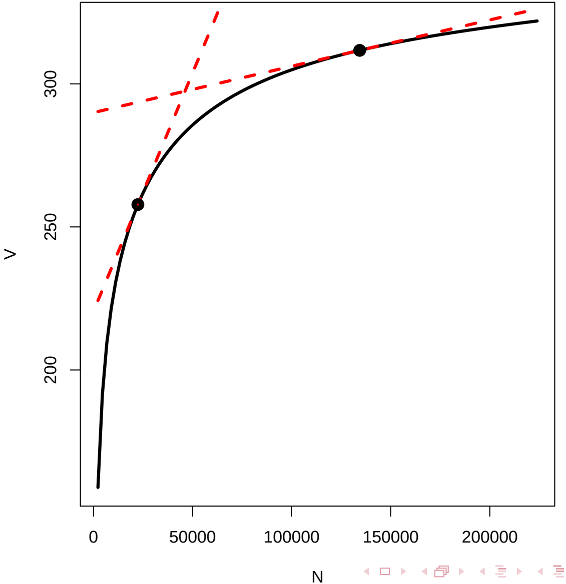
$\mathscr{P}$ and sample size

- We saw that as N increases, V also increases (for at-least-mildly-productive processes)
- Thus, V cannot be compared at different Ns
- However, growth rate is also systematically decreasing as N becomes larger
- At the beginning, any word will be a hapax legomenon; as sample increases, hapaxes will be increasingly lower proportion of sample

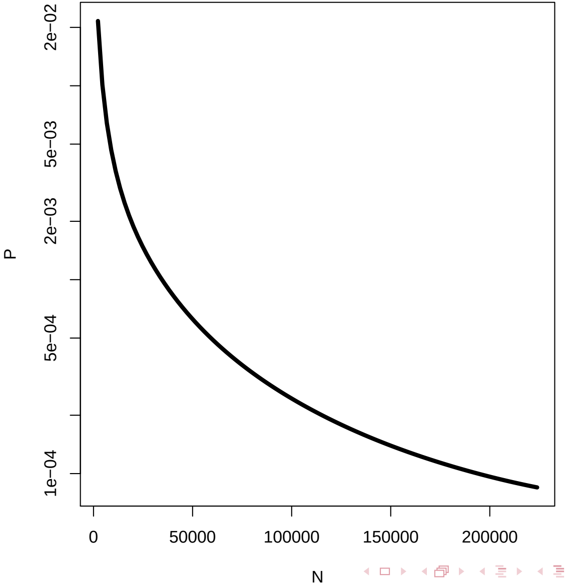
$\mathscr{P}$ and sample size

- We saw that as N increases, V also increases (for at-least-mildly-productive processes)
- Thus, V cannot be compared at different Ns
- However, growth rate is also systematically decreasing as N becomes larger
- At the beginning, any word will be a hapax legomenon; as sample increases, hapaxes will be increasingly lower proportion of sample
- A specific instance of the more general problem of "variable constants" (Tweedie and Baayen 1998) in lexical statistics (cf. type/token ratio)

Growth rate of *re-* at different sample sizes



$\mathscr{P}$ as a function of N (*re-*)



V and $\mathscr{P}$ at arbitrary Ns

- In order to compare V and $\mathscr{P}$ of processes (and predict how process will develop in larger samples)...

V and $\mathscr{P}$ at arbitrary Ns

- In order to compare V and $\mathscr{P}$ of processes (and predict how process will develop in larger samples)...
- we need to be able to estimate V and V_1 at arbitrary Ns

V and $\mathscr{P}$ at arbitrary Ns

- In order to compare V and $\mathscr{P}$ of processes (and predict how process will develop in larger samples)...
- we need to be able to estimate V and V_1 at arbitrary Ns
- Once we compare $\mathscr{P}$ at same N, we might as well compare V_1 directly (since $\mathscr{P} = V_1/N$ and N will be constant across compared processes)

V and $\mathscr{P}$ at arbitrary Ns

- In order to compare V and $\mathscr{P}$ of processes (and predict how process will develop in larger samples)...
- we need to be able to estimate V and V_1 at arbitrary Ns
- Once we compare $\mathscr{P}$ at same N, we might as well compare V_1 directly (since $\mathscr{P} = V_1/N$ and N will be constant across compared processes)
- Most intuitive: VGC plot comparison

Roadmap

Lexical statistics: the basics

Zipf's law
- Typical frequency patterns
- Zipf's law
- Consequences

Applications
- Productivity in morphology
- Productivity beyond morphology
- Lexical richness
- Conclusion and outlook

Productivity beyond morphology

- Measuring generative potential of process/category not limited to morphology

Productivity beyond morphology

- Measuring generative potential of process/category not limited to morphology
- Applications in lexicology, collocation and idiom studies, morphosyntax, syntax, language technology

Productivity beyond morphology

- Measuring generative potential of process/category not limited to morphology
- Applications in lexicology, collocation and idiom studies, morphosyntax, syntax, language technology
- E.g., measure growth of nouns, adjectives, loanwords, relative productivity of two constructions, growth of UNKNOWN lemmas as dataset increases. . .

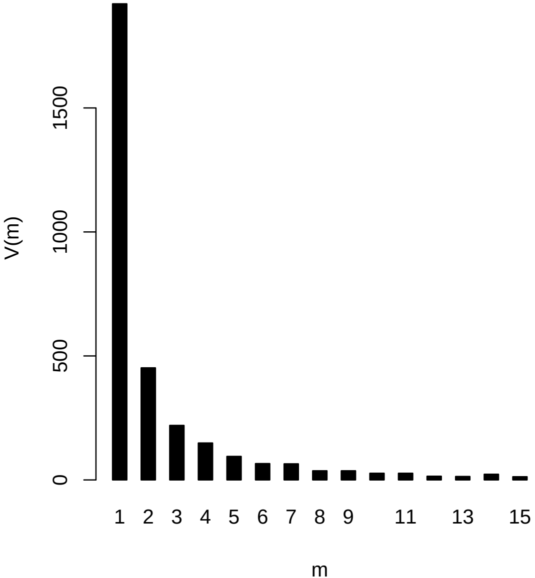
Productivity beyond morphology

- Measuring generative potential of process/category not limited to morphology
- Applications in lexicology, collocation and idiom studies, morphosyntax, syntax, language technology
- E.g., measure growth of nouns, adjectives, loanwords, relative productivity of two constructions, growth of UNKNOWN lemmas as dataset increases…
- An example: measuring productivity of NP and PP expansions in German TIGER treebank

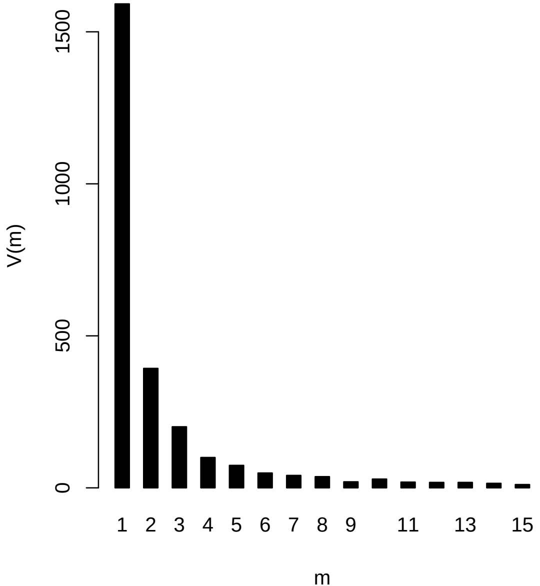
TIGER expansions

- Types are non-terminal rewrite rules for NP and PP, e.g:
  - NP $\longrightarrow$ ART ADJA NN
  - PP $\longrightarrow$ APPR ART NN
- Frequency of occurrence of expansions collected from about 900,000 tokens (50,000 sentences) of German newspaper text from Frankfurter Rundschau
- http://www.ims.uni-stuttgart.de/projekte/TIGER

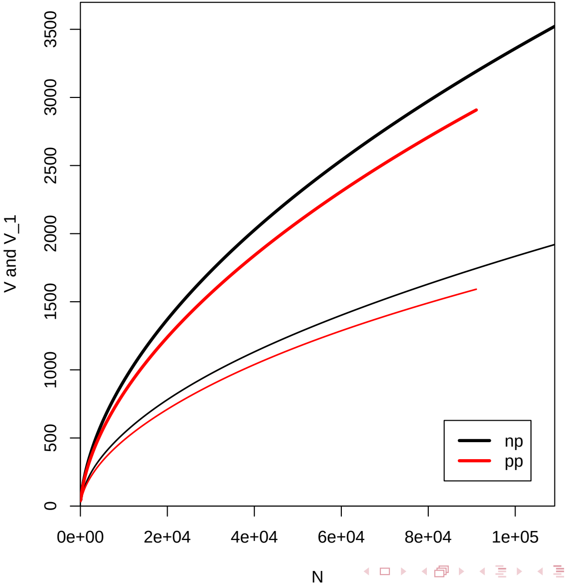
NP spectrum



PP spectrum



Growth curves of NP and PP



Lexical richness

- How many words did Shakespeare know? Are the later Harry Potters more lexically diverse than the early ones?

Lexical richness

- How many words did Shakespeare know? Are the later Harry Potters more lexically diverse than the early ones?
- Are advanced learners distinguishable from native speakers in terms of vocabulary richness? How many words do 5-year-old children know?

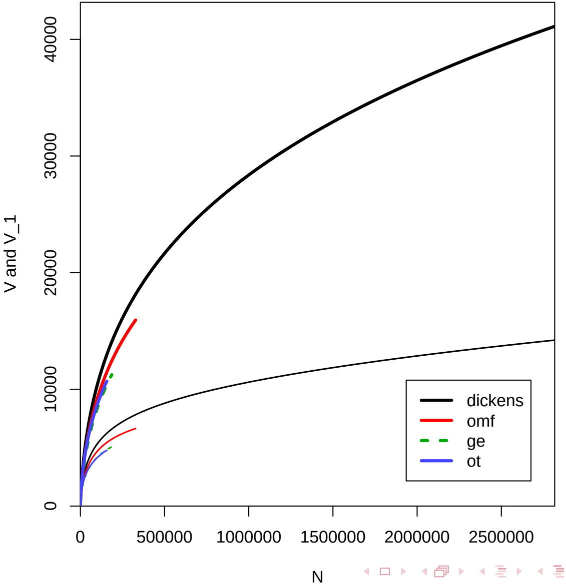
Lexical richness

- How many words did Shakespeare know? Are the later Harry Potters more lexically diverse than the early ones?
- Are advanced learners distinguishable from native speakers in terms of vocabulary richness? How many words do 5-year-old children know?
- Can changes in V detect the onset of Alzheimer's disease? (Garrard et al. 2005)

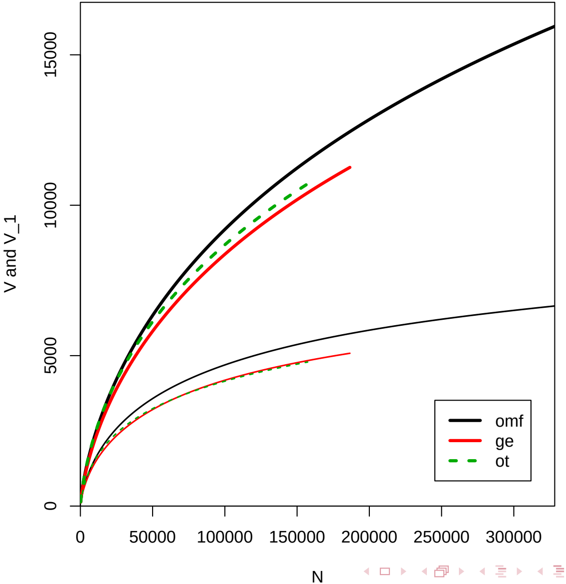
The Dickens' datasets

- Dickens corpus: collection of 14 works by Dickens, about 2.8 million tokens
- Oliver Twist: early work (1837-1839), about 160k tokens
- Great Expectations: later work (1860-1861), considered one of Dickens' masterpieces, about 190k tokens
- Our Mutual Friend: last completed novel (1864-1865), about 330k tokens

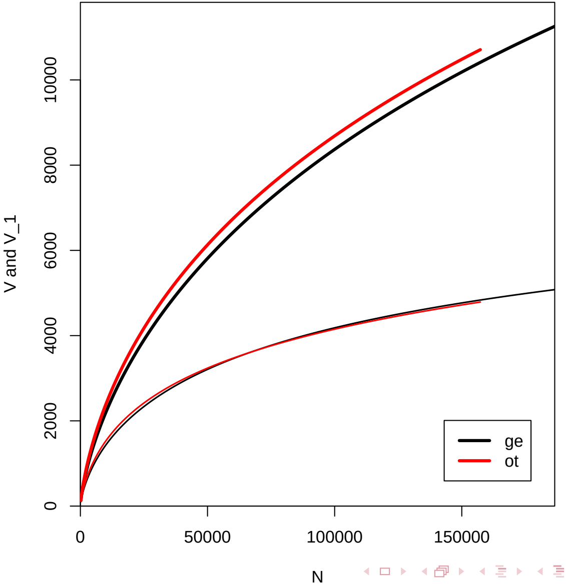
Dickens' V


V and V_1
N
0
10000
20000
30000
40000
0
500000
1000000
1500000
2000000
2500000
dickens
omf
ge
ot

The novels compared



Oliver vs. Great Expectations



Roadmap

Lexical statistics: the basics

Zipf's law
- Typical frequency patterns
- Zipf's law
- Consequences

Applications
- Productivity in morphology
- Productivity beyond morphology
- Lexical richness
- Conclusion and outlook

Conclusion and outlook

- Productivity, lexical richness, extrapolation of type counts for language engineering purposes. . .

Conclusion and outlook

- Productivity, lexical richness, extrapolation of type counts for language engineering purposes. . .
- all applications require a model of the larger **population** of types that our sample comes from

Conclusion and outlook

- Productivity, lexical richness, extrapolation of type counts for language engineering purposes. . .
- all applications require a model of the larger **population** of types that our sample comes from
- Two reasons to construct model of type population distribution:

Conclusion and outlook

- Productivity, lexical richness, extrapolation of type counts for language engineering purposes...
- all applications require a model of the larger **population** of types that our sample comes from
- Two reasons to construct model of type population distribution:
  - Population distribution interesting by itself, for theoretical reasons or in NLP applications

Conclusion and outlook

- Productivity, lexical richness, extrapolation of type counts for language engineering purposes. . .
- all applications require a model of the larger **population** of types that our sample comes from
- Two reasons to construct model of type population distribution:
  - Population distribution interesting by itself, for theoretical reasons or in NLP applications
  - We know how to simulate sampling from population; thus once we have population model we can obtain estimates of type-related quantities (e.g., V and V_1) at arbitrary Ns

Modeling the population
Productivity

- Distribution of types of category of interest necessary to estimate V and V_1 at arbitrary Ns, in order to compare VGCs and $\mathscr{P}$ of different processes

Modeling the population

Productivity

- Distribution of types of category of interest necessary to estimate V and V_1 at arbitrary Ns, in order to compare VGCs and $\mathscr{P}$ of different processes
- However, type population distribution of word formation process (or other category) might be of interest by itself, as model of a part of the mental lexicon of speaker

Modeling the population

Lexical richness

- Lexical richness $= V$ of whole population (how many words did Shakespeare know? Was the lexical repertoire of young Dickens smaller than that of old Dickens? How many words do 5-year-old children know?)

Modeling the population
Lexical richness

- Lexical richness = V of whole population (how many words did Shakespeare know? Was the lexical repertoire of young Dickens smaller than that of old Dickens? How many words do 5-year-old children know?)
- Accurate estimate of population V would solve "variable constant" problem

Roadmap

Lexical statistics: the basics

Zipf's law

Typical frequency patterns

Zipf's law

Consequences

Applications

Productivity in morphology

Productivity beyond morphology

Lexical richness

Conclusion and outlook

Modeling the population

Lexical richness

- Lexical richness = V of whole population (how many words did Shakespeare know? Was the lexical repertoire of young Dickens smaller than that of old Dickens? How many words do 5-year-old children know?)
- Accurate estimate of population V would solve "variable constant" problem
- Sampling from population, in particular to compute VGC, also of interest

Modeling the population

Some NLP applications

- Estimate number (and growth rate) of typos, UNKNOWNs (or other target tokens) in larger samples

Modeling the population

Some NLP applications

- Estimate number (and growth rate) of typos, UNKNOWNs (or other target tokens) in larger samples
  $\longrightarrow$ estimate V and V_1 at arbitrary Ns

Modeling the population
Some NLP applications

- Estimate number (and growth rate) of typos, UNKNOWNs (or other target tokens) in larger samples
  $\longrightarrow$ estimate V and V_1 at arbitrary Ns
- Estimate proportion of OOV words under assumption that lexicon contains top n most frequent types (see zipfR tutorial)

Modeling the population
Some NLP applications

- Estimate number (and growth rate) of typos, UNKNOWNs (or other target tokens) in larger samples $\longrightarrow$ estimate V and V_1 at arbitrary Ns
- Estimate proportion of OOV words under assumption that lexicon contains top n most frequent types (see zipfR tutorial) $\longrightarrow$ requires estimation of V and frequency spectrum at arbitrary Ns (to find out for how many tokens do the top n types account for)

Modeling the population

Some NLP applications

- Estimate number (and growth rate) of typos, UNKNOWNs (or other target tokens) in larger samples $\rightarrow$ estimate V and V_1 at arbitrary Ns
- Estimate proportion of OOV words under assumption that lexicon contains top n most frequent types (see zipfR tutorial) $\rightarrow$ requires estimation of V and frequency spectrum at arbitrary Ns (to find out for how many tokens do the top n types account for)
- Good-Turing estimation, Bayesian priors

Modeling the population

Some NLP applications

- Estimate number (and growth rate) of typos, UNKNOWNs (or other target tokens) in larger samples $\rightarrow$ estimate V and V_1 at arbitrary Ns
- Estimate proportion of OOV words under assumption that lexicon contains top n most frequent types (see zipfR tutorial) $\rightarrow$ requires estimation of V and frequency spectrum at arbitrary Ns (to find out for how many tokens do the top n types account for)
- Good-Turing estimation, Bayesian priors $\rightarrow$ require full type population model

Outlook

- We need model of type population distribution

Outlook

- We need model of type population distribution
- We will use Zipf(-Mandelbrot)'s law as starting point to model how population looks like

Roadmap

Lexical statistics: the basics

Zipf's law

Typical frequency patterns

Zipf's law

Consequences

Applications

Productivity in morphology

Productivity beyond morphology

Lexical richness

Conclusion and outlook

Outlook

- We need model of type population distribution
- We will use Zipf(-Mandelbrot)'s law as starting point to model how population looks like

TO BE CONTINUED