Counting Words:
Pre-Processing and Non-Randomness

Marco Baroni & Stefan Evert

Málaga, 11 August 2006

Outline

Pre-Processing
Non-Randomness
The End

Pre-processing

- IT IS IMPORTANT!!! (Evert and Lüdeling 2001)
- Automated pre-processing often necessary (13,850 types begin with *re-* in BNC, 103,941 types begin with *ri-* in itWaC)
- We can rely on:
  - POS tagging
  - Lemmatization
  - Pattern matching heuristics (e.g., candidate prefixed form must be analyzable as *PRE+VERB*, with *VERB* independently attested in corpus)
- However...

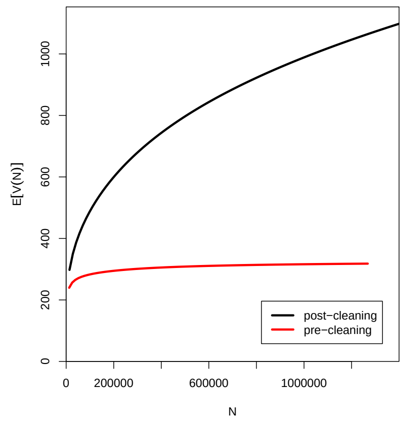
The problem with low frequency words

- Correct analysis of low frequency words is fundamental to measure productivity, estimate LNRE models
- Automated tools will tend to have lowest performance on low frequency forms:
  - Statistical tools will suffer from lack of relevant training data
  - Manually-crafted tools will probably lack the relevant resources
- Problems in both directions (under- and overestimation of hapax counts)
- Part of the more general "95% performance" problem

Underestimation of hapaxes

- The Italian TreeTagger lemmatizer is lexicon-based; out-of-lexicon words (e.g., productively formed words containing a prefix) are lemmatized as UNKNOWN
- No prefixed word with dash (*ri-cadere*) is in lexicon
- Writers are more likely to use dash to mark transparent morphological structure

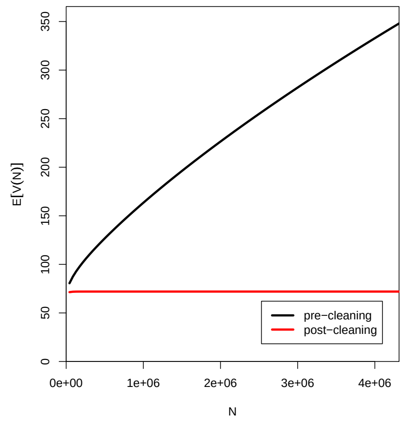
Productivity of *ri-* with and without an extended lexicon



Overestimation of hapaxes

- "Noise" generates hapax legomena
- The Italian TreeTagger thinks that dashed expressions containing pronoun-like strings are pronouns
- Dashed strings can be anything, including full sentences
- This creates a lot of pseudo-pronoun hapaxes: *tu-tu*, *parapaponzi-ponzi-pò*, *altri-da-lui-simili-a-lui*

Productivity of the pronoun class before and after cleaning



$\mathscr{P}$ (and V) with/without correct post-processing

- With:

class	V	V_1	N	$\mathscr{P}$
ri-	1098	346	1,399,898	0.00025
pronouns	72	0	4,313,123	0

- Without:

class	V	V_1	N	$\mathscr{P}$
ri-	318	8	1,268,244	0.000006
pronouns	348	206	4,314,381	0.000048

A final word on pre-processing

- IT IS IMPORTANT
- Often, major roadblock of lexical statistics investigations

Outline

Pre-Processing

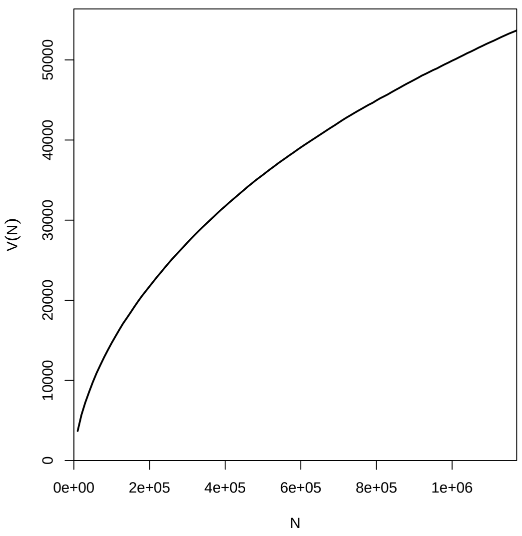
Non-Randomness

The End

Non-randomness

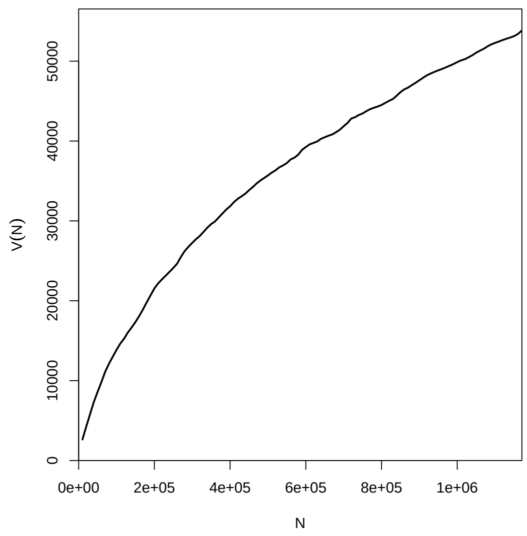
- LNRE modeling based on assumption that our corpora/datasets are **random** samples from the population
- This is obviously not the case
- Can we pretend that a corpus is random?
- What are the consequences of non-randomness?

A Brown-sized random sample from a ZM population estimated with Brown



Pre-processing and non-randomness
Baroni & Evert
Pre-Processing
Non-Randomness
The End

The real Brown



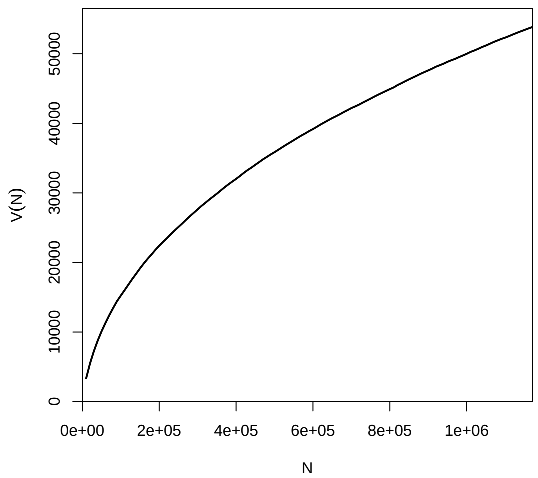
Pre-processing and non-randomness
Baroni & Evert
Pre-Processing
Non-Randomness
The End

Where does non-randomness come from?

- Syntax?
- *the the* should be most frequent English bigram
- If the problem is due to syntax, randomizing by sentence will not get rid of it (Baayen 2001, ch. 5)

Pre-processing and non-randomness
Baroni & Evert
Pre-Processing
Non-Randomness
The End

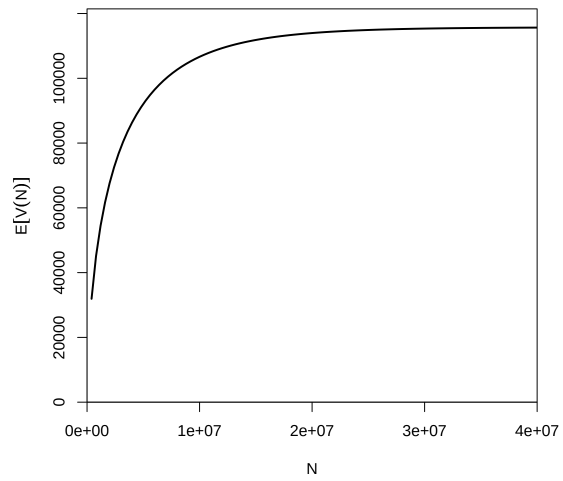
The Brown randomized by sentence



Where does non-randomness come from?

- Not syntax (syntax has short span effect; *the* counts for 10k intervals are OK)
- **Underdispersion** of content-rich words
- The chance of two Noriegas is closer to $\pi/2$ than π^2 (Church 2000)
- *diethylstilbestrol* occurs 3 times in Brown, all in same document (recommendations on feed additives)
- Underdispersion will lead to serious **underestimation** of rare type count
- fZM estimated on Brown predicts $S = 115,539$ in English

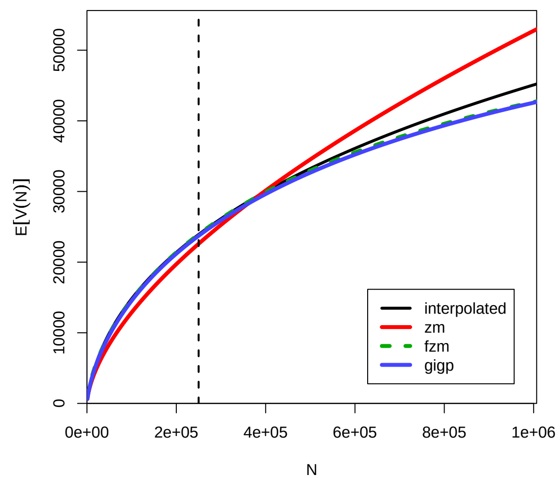
Underestimating types
Extrapolating Brown VGC with fZM



Assessing extrapolation quality

- We have no way to assess goodness of fit of extrapolation from corpus to larger sample from same population
- However, we can estimate models on subset of available data, and extrapolate to full corpus size (Evert and Baroni 2006)
- I.e., use corpus as our population, sample from it

Extrapolation from a **random** sample of 250k Brown tokens

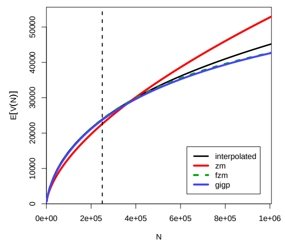


Goodness of fit to spectrum elements

Based on multivariate chi-squared statistic

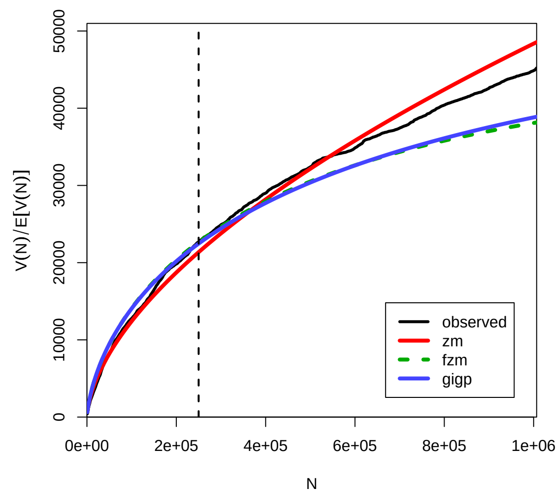
	estimation size			max extrapolation size		
model	X2	df	p	X2	df	p
ZM	7,856	14	$\ll 0.001$	35,346	16	$\ll 0.001$
fZM	539	13	$\ll 0.001$	4,525	16	$\ll 0.001$
GIGP	597	13	$\ll 0.001$	3,449	16	$\ll 0.001$

Compare to V fit:



Pre-processing and non-randomness
Baroni & Evert
Pre-Processing
Non-Randomness
The End

Extrapolation from first 250k tokens in corpus



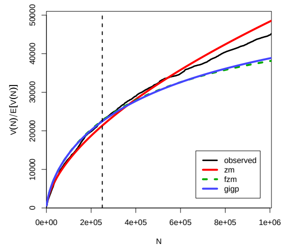
Pre-processing and non-randomness
Baroni & Evert
Pre-Processing
Non-Randomness
The End

Goodness of fit to spectrum elements

Based on multivariate chi-squared statistic

	estimation size			max extrapolation size		
model	X2	df	p	X2	df	p
ZM	8,066	14	$\ll 0.001$	33,6766	16	$\ll 0.001$
fZM	1,011	13	$\ll 0.001$	17,559	16	$\ll 0.001$
GIGP	587	13	$\ll 0.001$	7,815	16	$\ll 0.001$

Compare to V fit:



Pre-processing and non-randomness
Baroni & Evert
Pre-Processing
Non-Randomness
The End

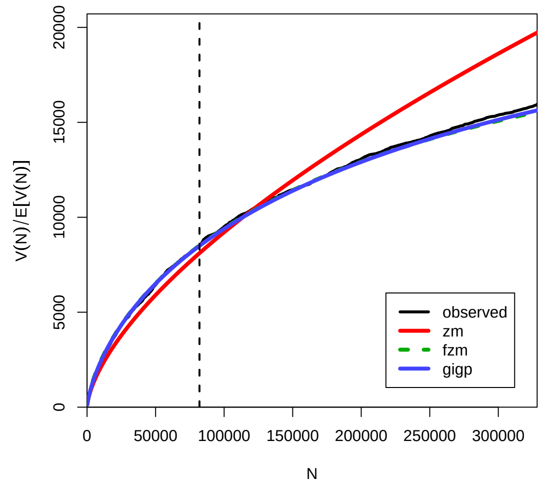
The corpus as a (non-)random sample

- In our experiment, we had access to full population (the Brown) and could take random sample from it
- In real life, full corpus *is* our sample from the population (e.g., “English”, an author’s mental lexicon, all words generated by a wfp)
- If it is not random, there is nothing we can do about it (randomizing the sample will not help!)

What can we do?

- Abandon lexical statistics
- Live with it
- Re-define the population
- Try to account for underdispersion when computing the models (will get mathematically very complicated, but see Baayen 2001, ch. 5)

Not always that bad
Our Mutual Friend



Outline

Pre-Processing
Non-Randomness
The End

What we have done

- **Motivation**: studying distribution and *V* growth rate of type-rich populations (sample captures only small proportion of types in population)
- **LNRE** modeling:
  - **Population model** with limited number of parameters (e.g., ZM), expressed in terms of type density function
  - Equations to calculate expected *V* and frequency spectrum in **random samples** of arbitrary size using population model
  - **Estimation** of population parameters via fit of expected elements to observed frequency spectrum
- **zipfR** package to apply LNRE modeling
- **Problems**

What we (and perhaps some of you?) would like to do next

- Study (and deal with) non-randomness
- Better parameter estimation
- Improve zipfR (any feature request?)
- Use LNRE modeling in applications, e.g.:
  - Good-Turing-style estimation
  - Productivity beyond morphology
  - Better features for machine learning
  - Mixture models

Pre-processing and non-randomness

Baroni & Evert

Pre-Processing

Non-Randomness

The End

That's All, Folks!

THE END