Using Simple Frequencies

Lecture 3: Using Text Analytics to Discover Meaning

Mark Keane, Insight/CSI, UCD

Selling Things

stock-markets

social media

science

news

polls

sentiment-id

sentiment-use

time-series

summaries

VSMs

Classifiers

Clustering

cosine

jaccard

dice

levenschtein

TF-IDF

LLR

PMI

Entropy

*** simple frequencies ***

pre-processed text items of some sort…

The Intuition

- Simplest thing we can do is count words
- Word frequencies may not seem like much but are informative (e.g. distributions can show regularities in the data)
- Technical Issues: proper normalisation, pre-processing of data, modelling deeper reasons

Frequencies can tells us about…

- The opinions of large groups of people (populations) via text sources
- Cultural changes in different countries
- Agreement/disagreement between groups holding different views
- How pop charts change

Overview

- Simple Frequency: Word Clouds/Tags
- Using Frequencies from Corpora:
  - Google N-gram Corpus (Culturomics)
  - Search Term Corpora
  - Other Text Corpora (Books, Pop charts…)
- Using Frequency Distributions
  - Zipf…Moby Dick…Pop Charts…Bubbles

Word Clouds

Word / Tag Clouds

- Simplest frequency usage…visualising common occurrence of a string in some text usually by font size (arranged)

- Used to summarise contents of webpages (Flickr) and navigation aid

President Barack Obama

Inaugural address, 20 January 2009

SOURCE: wordle.net

We asked,

YOU ANSWERED!

How our fans describe
Sutter Home Pink Moscato...

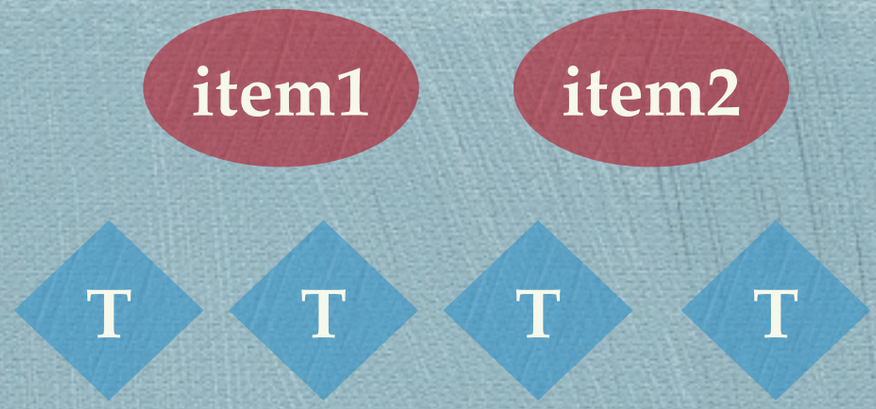
Shaped Word Map
Tweets containing Apple
January 20-21, 2009

Jeff Clark - Neoformix

Word Clouds or Tag Clouds

- Word clouds were used to capture frequencies of words in texts (eg speeches)
- Tag clouds emerged in social media to support user tagging of pictures, shared items, web-page meta-data, and so on

3 Distinct Uses…



- **Frequency of Tag Applied to Single Item:** "democratic vote" for an item (descriptions of the wine, genre attributed to band on last.fm)

- **Frequency Items to which Tag was Applied:** how popular is this tag as descriptor of these photos, web-pages, news stories (cf Flickr)

- **Tag Set Provides Categorisation of Item:** used for web-search for summarising a website

Word Clouds in Speeches

- Reasonable way to visualise the main components of a speech but really does not take you that far
- There is much more that can be done on the shape of a word-set's frequencies (i.e., the distribution)
- But, they are nice…

Tag Clouds

- Seemed a good way to support tagging of photos on social media site, to help select the most commonly used tags by a group
- Early, Halvey & Keane (2007) evaluation showed that word lists were actually better
- Flickr later apologised for use of them…

Doing a Tag Cloud

Creation of a tag cloud [edit]

In principle, the font size of a tag in a tag cloud is determined by its incidence. For a word cloud of categories like weblogs, frequency, for example, corresponds to the number of weblog entries that are assigned to a category. For smaller frequencies one can specify font sizes directly, from one to whatever the maximum font size. For larger values, a scaling should be made. In a linear normalization, the weight t_i of a descriptor is mapped to a size scale of 1 through *f*, where t_{min} and t_{max} are specifying the range of available weights.

$$s_i = \left\lceil \frac{f_{\max} \cdot (t_i - t_{\min})}{t_{\max} - t_{\min}} \right\rceil \text{for } t_i > t_{\min}\text{; else } s_i = 1$$

- s_i: display fontsize
- $f_{\max}$: max. fontsize
- t_i: count
- $t_{\min}$: min. count
- $t_{\max}$: max. count

Since the number of indexed items per descriptor is usually distributed according to a power law,[29] for larger ranges of values, a logarithmic representation makes sense.[30]

Implementations of tag clouds also include text parsing and filtering out unhelpful tags such as common words, numbers, and punctuation.

From Wikipedia June 2014

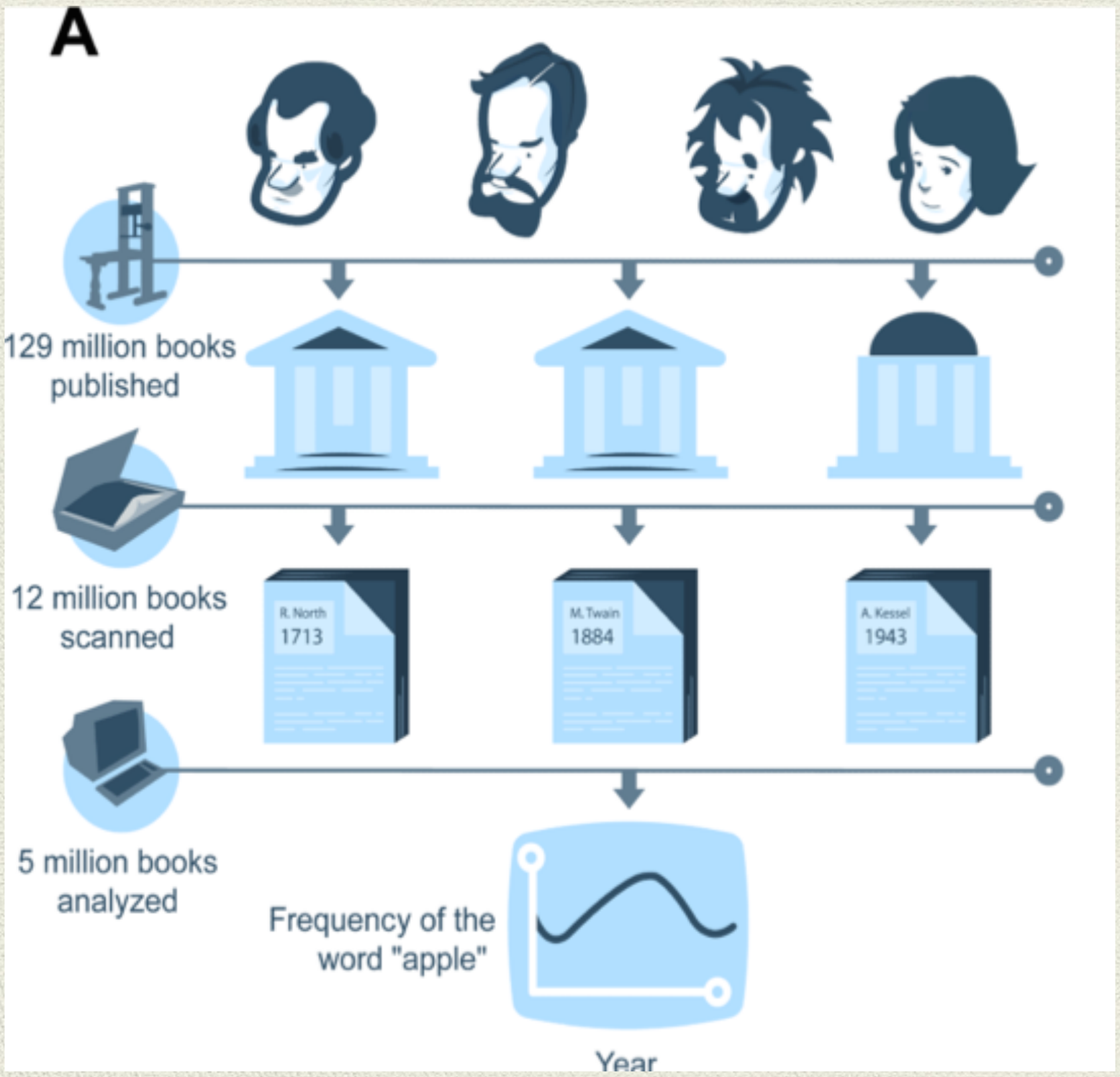
Counting Words in GoogleBooks

Corpora & Word Frequency

- ***Corpus*** is come collection of texts; all the Irish Times articles since 1900, all the books in the world, all readings of US school goers by 25

- (Normalised) Frequencies of words in corpora can tell you a lot, when you look at how they change over time

GoogleBooks Corpus



Michel et al. (2011), Quantitive Analysis of Culture Using Millions of Digitized Books. *Science*, *331*: 176–182.

Culturomics Corpus

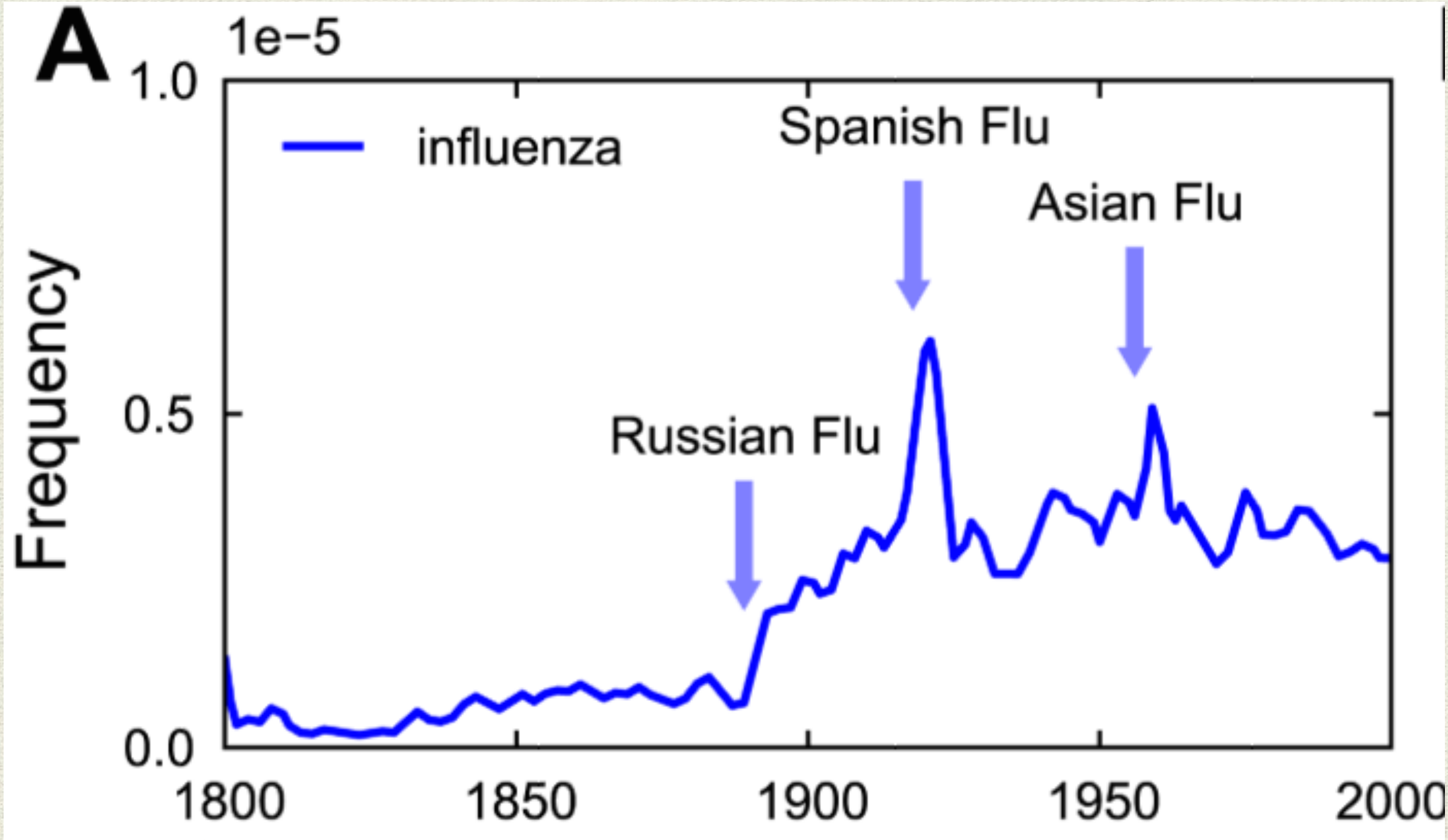
- 4% of all printed books from 1500-2000
- 5,195,769 books aka 500B words
- 1800 (60M p.a.), 1900 (1.4B p.a.), 2000 (8B p.a.)
- Turned into n-grams (up to 5-gram)
  - 1-grams (“stock”, “market”, “home”, “pet”…)
  - 2-grams (“stock market”, “home pet”)…
- Used to track cultural change

Its really BIG…

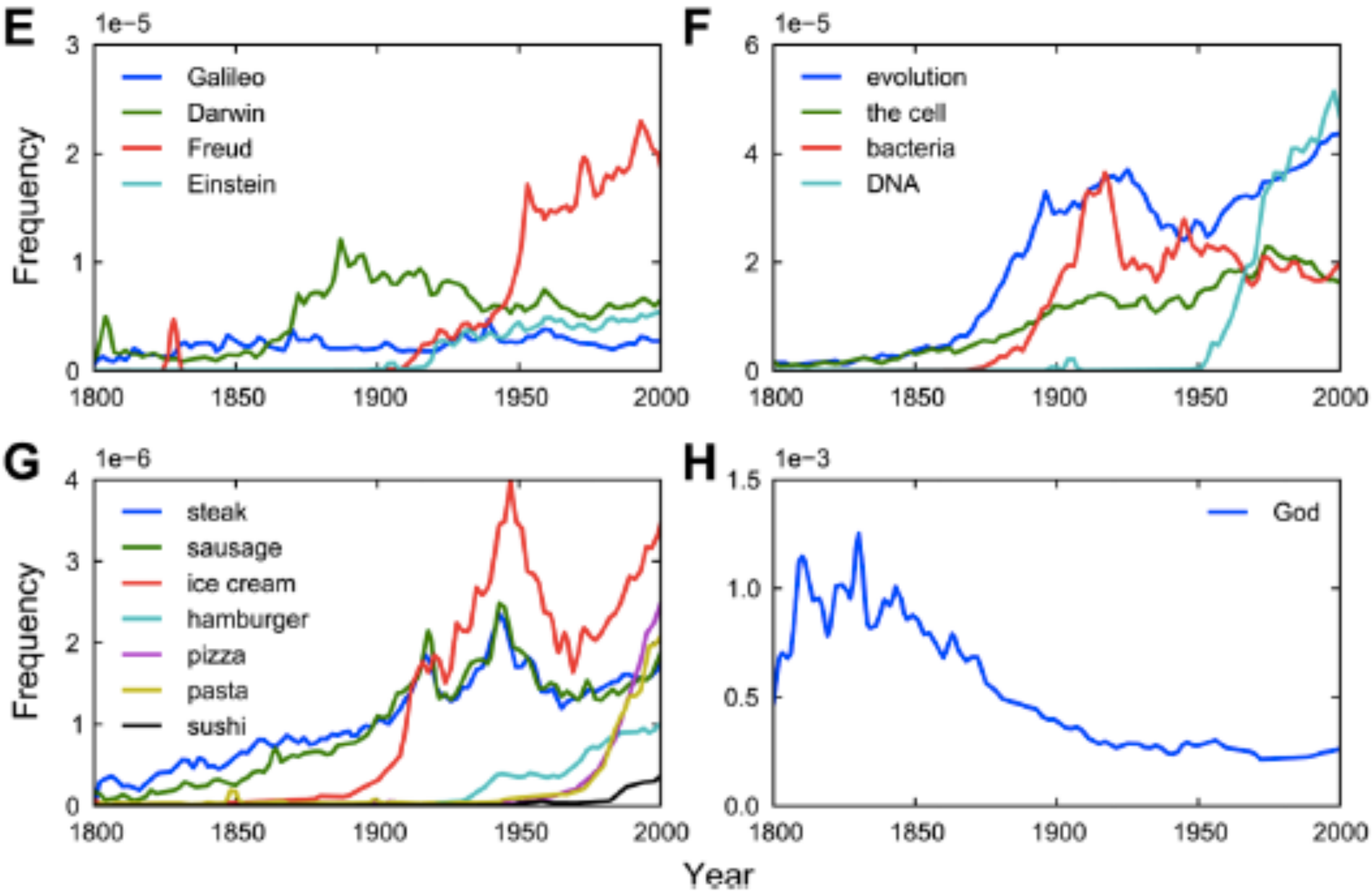
- Reading the books from 2000 alone, non-stop without sleep at 200 words per minute, would take you 80 years
- Corpus characters are 1000 times longer than the human genome
- If you wrote it out as a long sentence…it would go to the moon, and back, 10 times

US Civil War (1861-1865)
Civil Rights Movements (1955-1968)

Michel et al. (2011), Quantitive Analysis of Culture Using Millions of Digitized Books. *Science*, *331*: 176–182.



Michel et al. (2011), Quantitive Analysis of Culture Using Millions of Digitized Books. *Science*, *331*: 176–182.



Michel et al. (2011), Quantitive Analysis of Culture Using Millions of Digitized Books. *Science*, *331*: 176–182.

Simple Frequency (NOT)

- Note, these are not simple frequencies; normalising the raw count is really important
- In compiling n-gram sets, they only consider cases with >40 mentions in the **whole** corpus
- Frequency = N of an n-gram in a given year divided by the total N of words in the corpus for that year (n.b., book numbers differ)
- Note, using moving averages to smoothen graphs !

Simple Frequency (NOT)

- In 1861*, the 1-gram “slavery” appears 21,460 times on 11,687 pages of 1,208 books

- All 1861 books, contain 386,434,758 words

- Frequency of “slavery” for 1861 is:

  21,460/386,434,758 =

  5.5×10^{-5} =

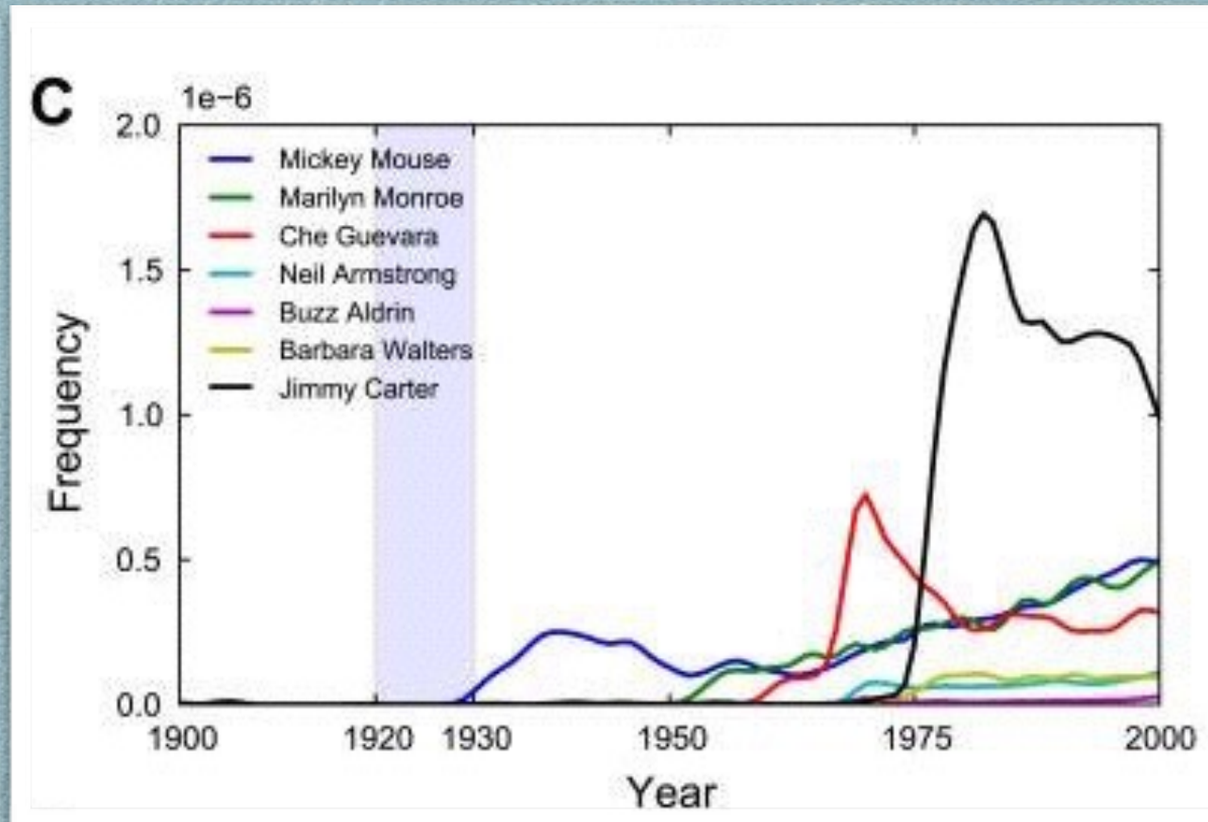
  0.0000555333

* Importance of slavery to US economy at the time, 1863 Emancipation Proclamation

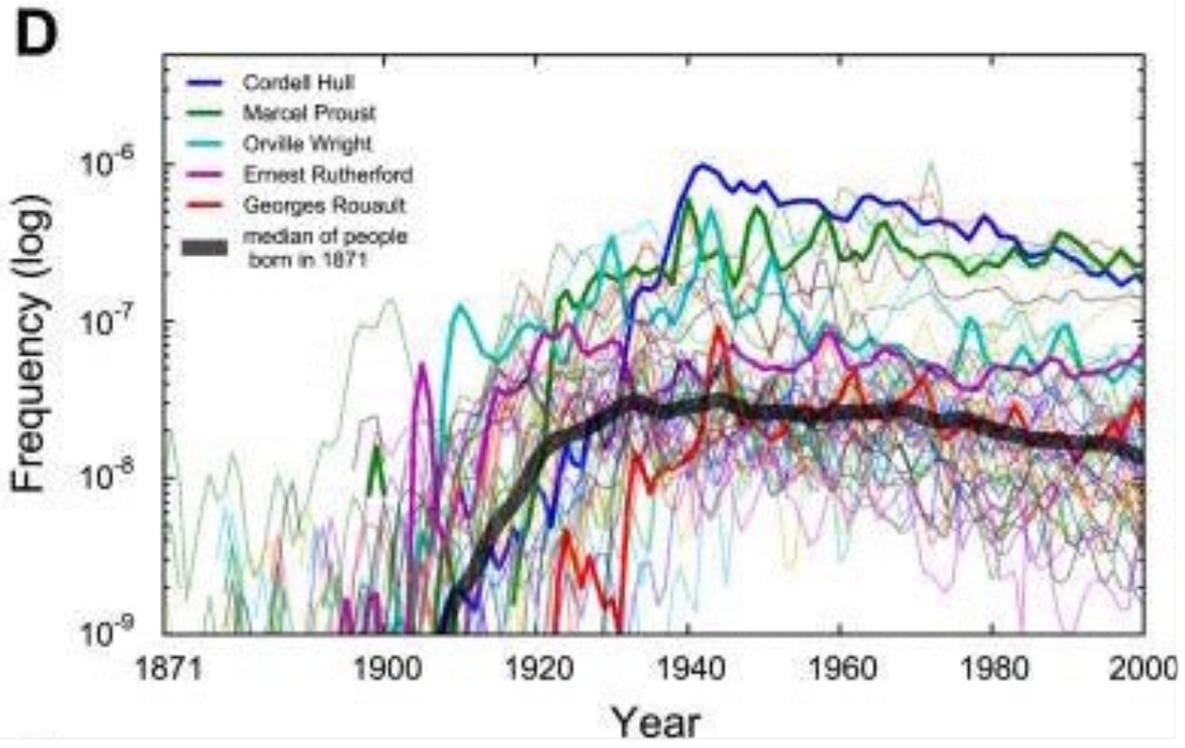
Freq Change = Culture Change

- The rise and fall of celebrity (how its faster)
- Adoption of climate change ideas
- Economic misery of the world

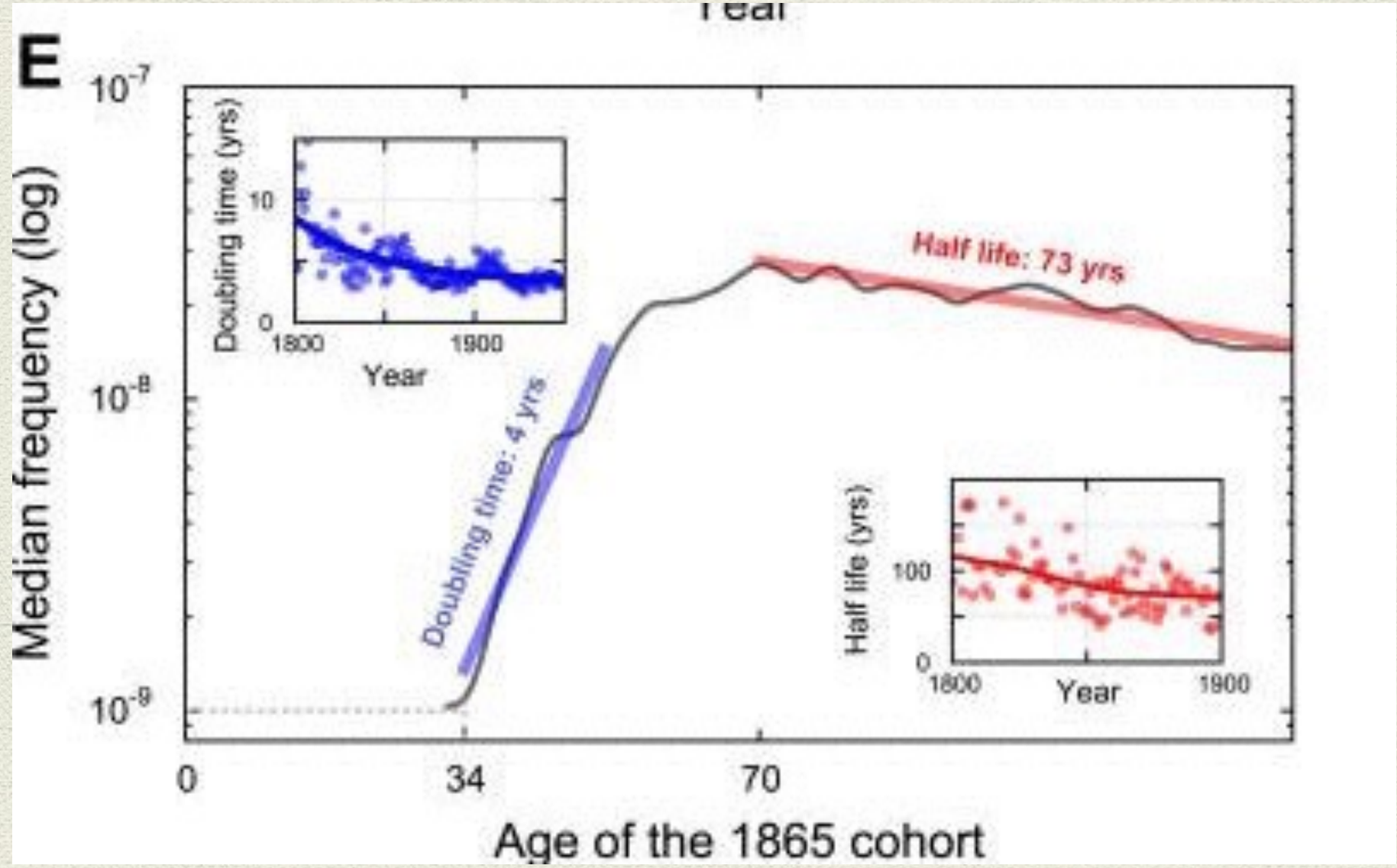
Eg1: Celebrity



C. Fame of People "Born" 'tween 1920-1940



D. 50 Most Famous People Born in 1871



(**E**) The median trajectory of the 1865 cohort has four parameters: (i) initial 'age of celebrity' (34 years old, tick mark); (ii) doubling time of the subsequent rise to fame (4 years, blue line); (iii) 'age of peak celebrity' (70 years after birth, tick mark), and (iv) half-life of the post-peak 'forgetting' phase (73 years, red line). *Inset*: The doubling time and half-life over time.

PAUSE to think…

Important Point

REM

- Pre-processing is not just about taking things out; stripping off stems, removing stops etc…
- It may also be about putting things in; like POS tags, syntax, entity tags, lexical chains

Finally, we have assumed…

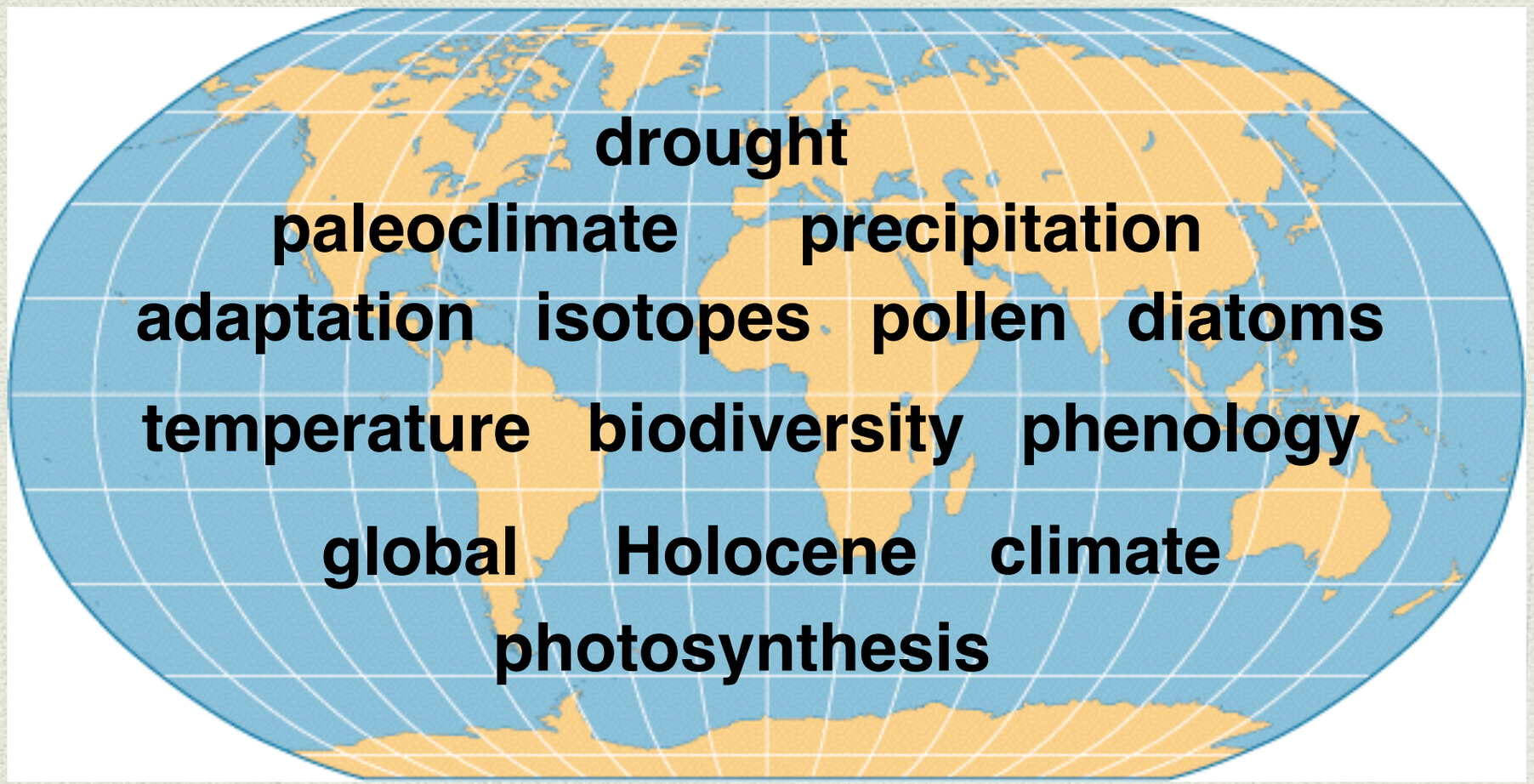
- That you just know which texts to pre-process; but, sometimes you have to think about selecting the texts that make up a corpus
- Is this defined naturally; every debate in the Dail since 1922…(simple case)
- Every news article about stock markets…how do we define this ? (medium case)
- Every tweet that is about senate elections …how do we define this ? (hard case)

REM

- GoogleBooks: selection of books non-trivial; multiple editions
- Ngram choice important and <40 cut-off, plus normalisation
- Also stop-word removal and POS tags added

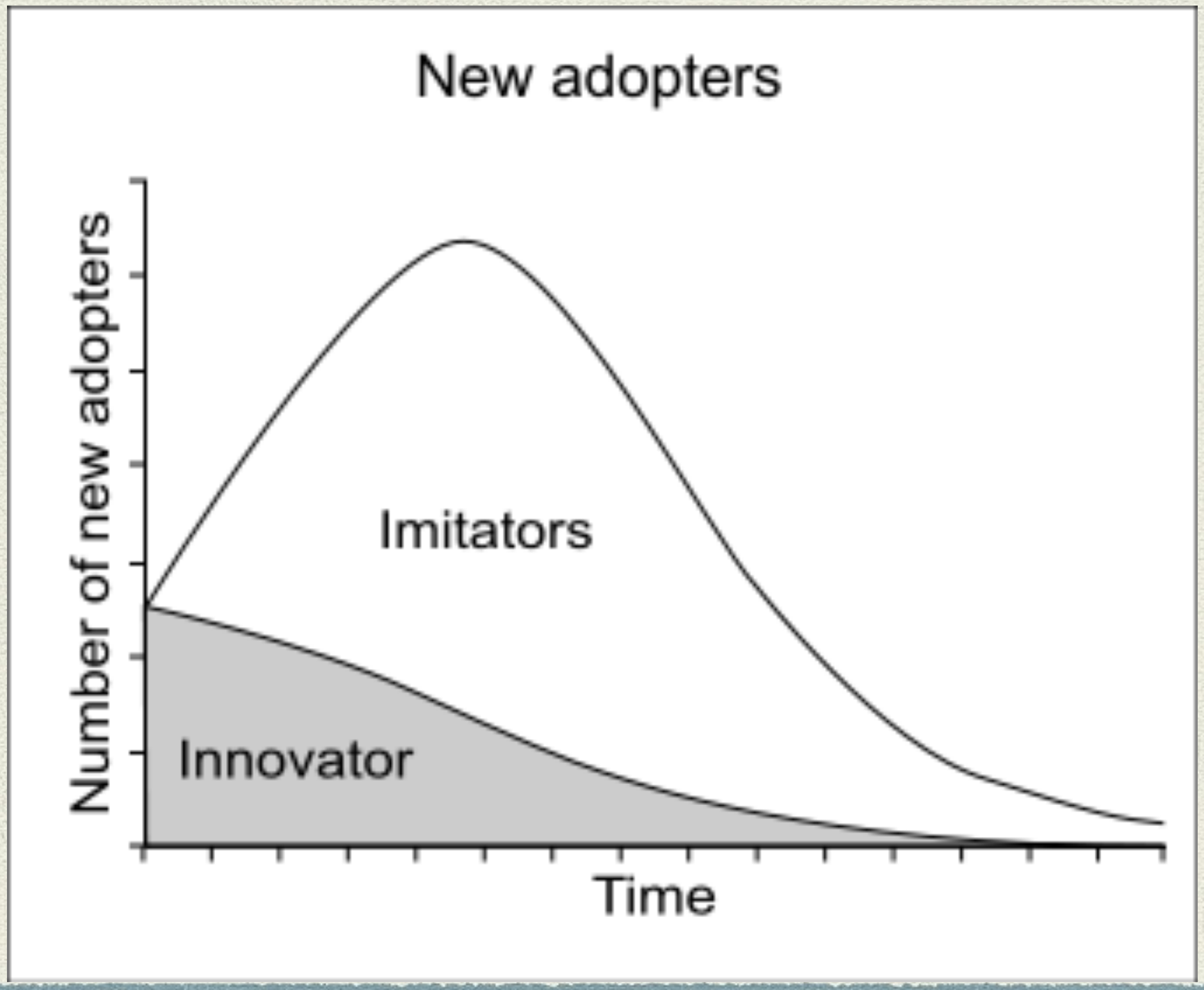
Eg2 Climate Change

- Culturomics-type analysis of climate-change word use
- Word-frequencies fitted to Bass Diffusion Model

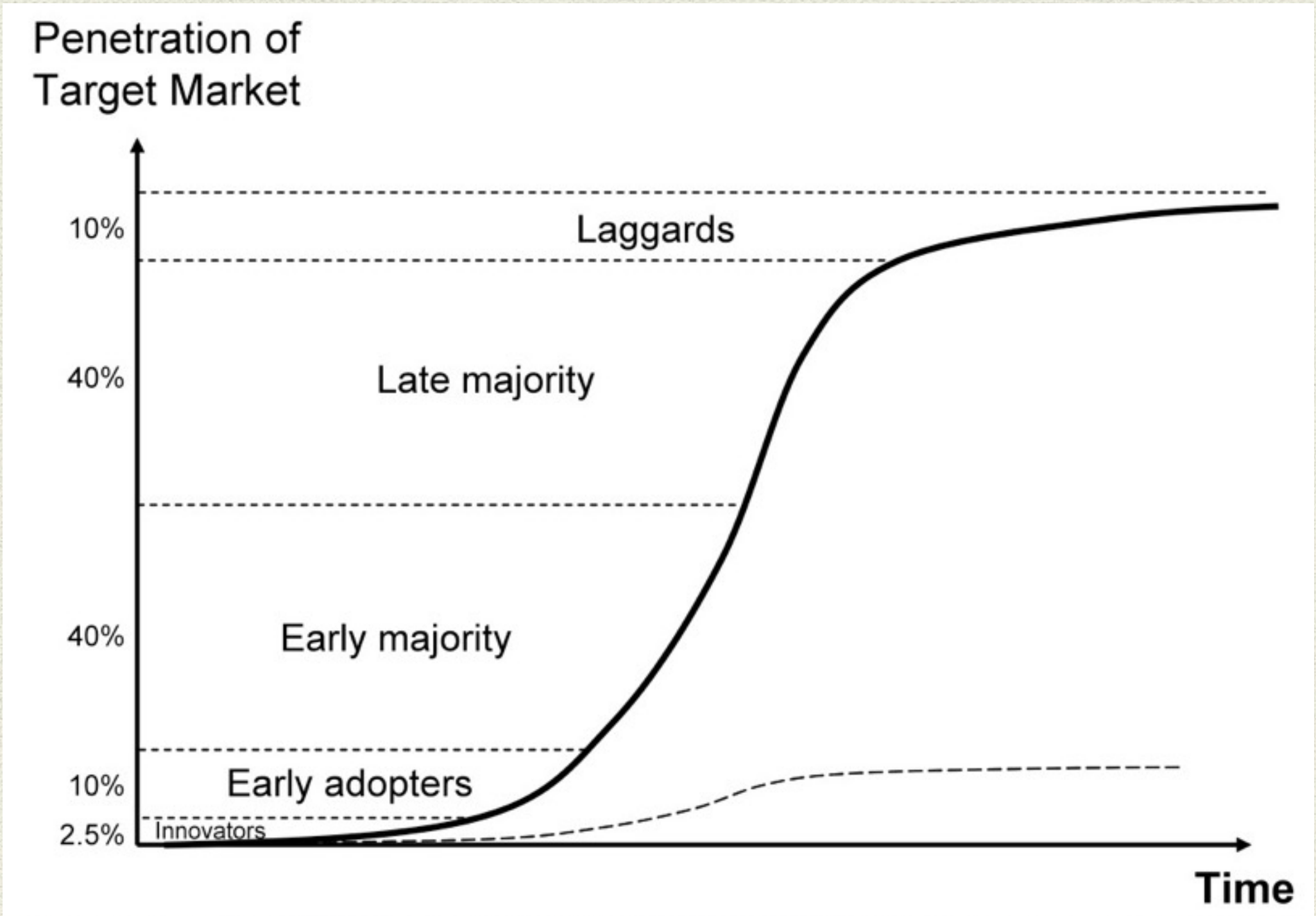


Bentley, R. A., Garnett, P., O'Brien, M. J., & Brock, W. A. (2012). Word diffusion and climate science. *PloS one, 7*, e47966.

Bass Diffusion:
Ideas/Products Are Spread by Different People



Bass Diffusion:
Creates a Standard Freq/Sales Curve



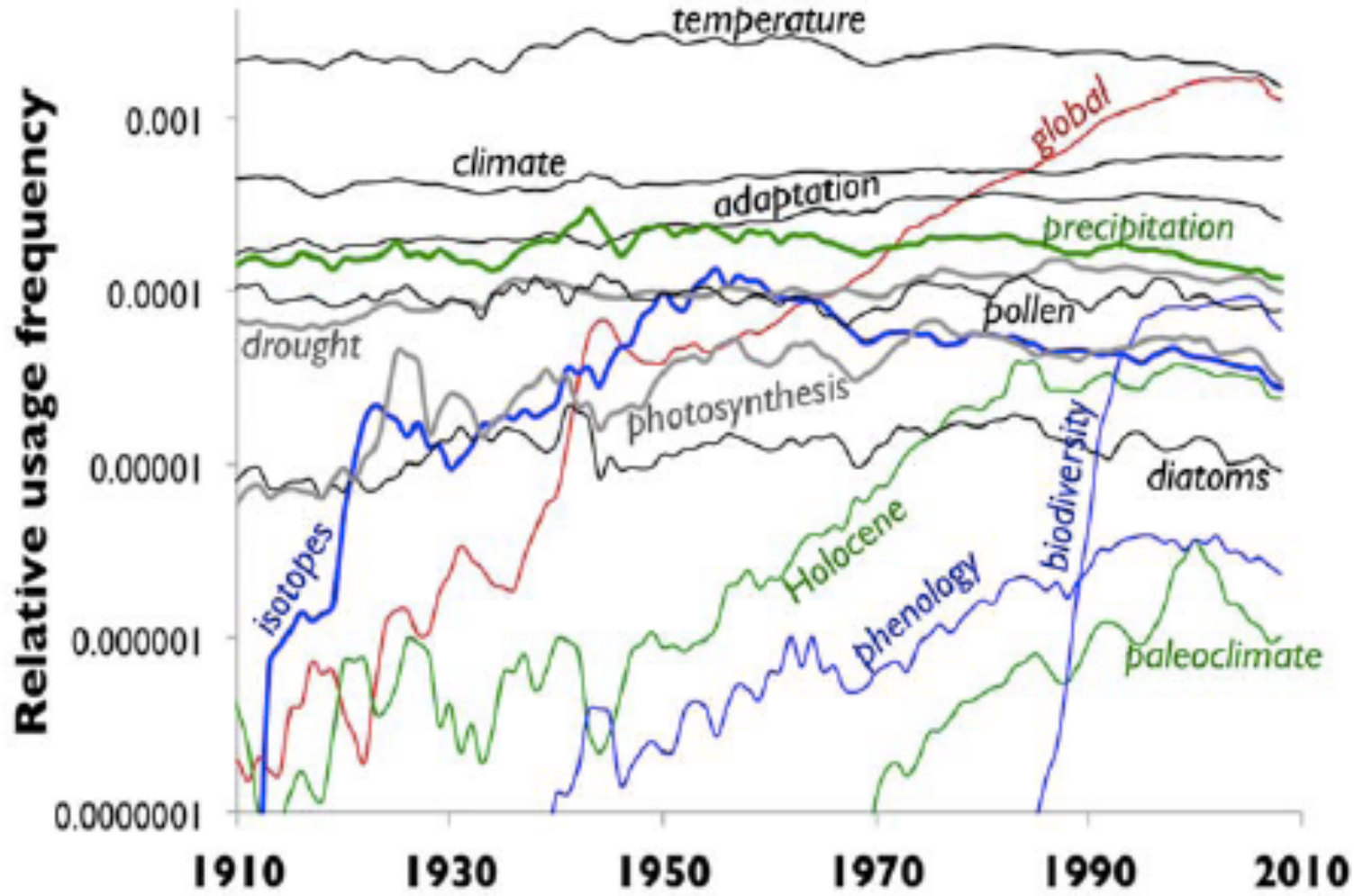


Figure 1. The popularities of the top climate change 1-grams in the Google Ngrams database, normalized to the word *the* and using a logarithmic scale. Shown here is the last century of public usage of a set of the top climate-change keywords in recent scientific publications [24], which include: *adaptation, biodiversity, climate, diatoms, drought, global, Holocene, isotopes, paleoclimate, phenology, photosynthesis, pollen, precipitation*, and *temperature*.
doi:10.1371/journal.pone.0047966.g001

Bentley, R. A., Garnett, P., O'Brien, M. J., & Brock, W. A. (2012). Word diffusion and climate science. *PloS one, 7*, e47966.

Eg3: Economic Misery

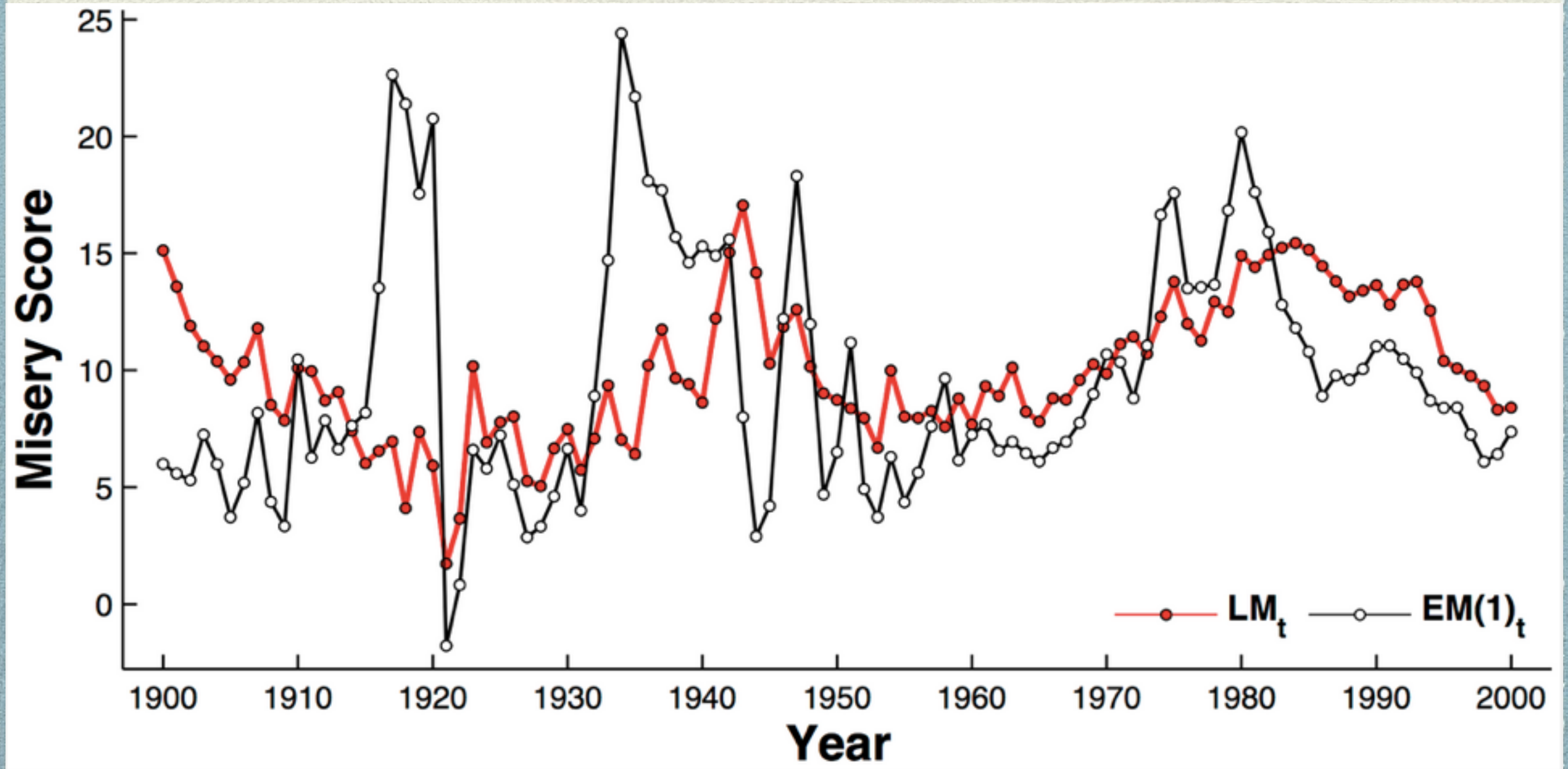
- Analysis of misery words in ngram corpus (literary misery, LM) correlates closely economic misery (EM)

- LM: Literary Misery Index uses a Wordnet Affect (Strapparava & Valitutti, 2004, 2008)

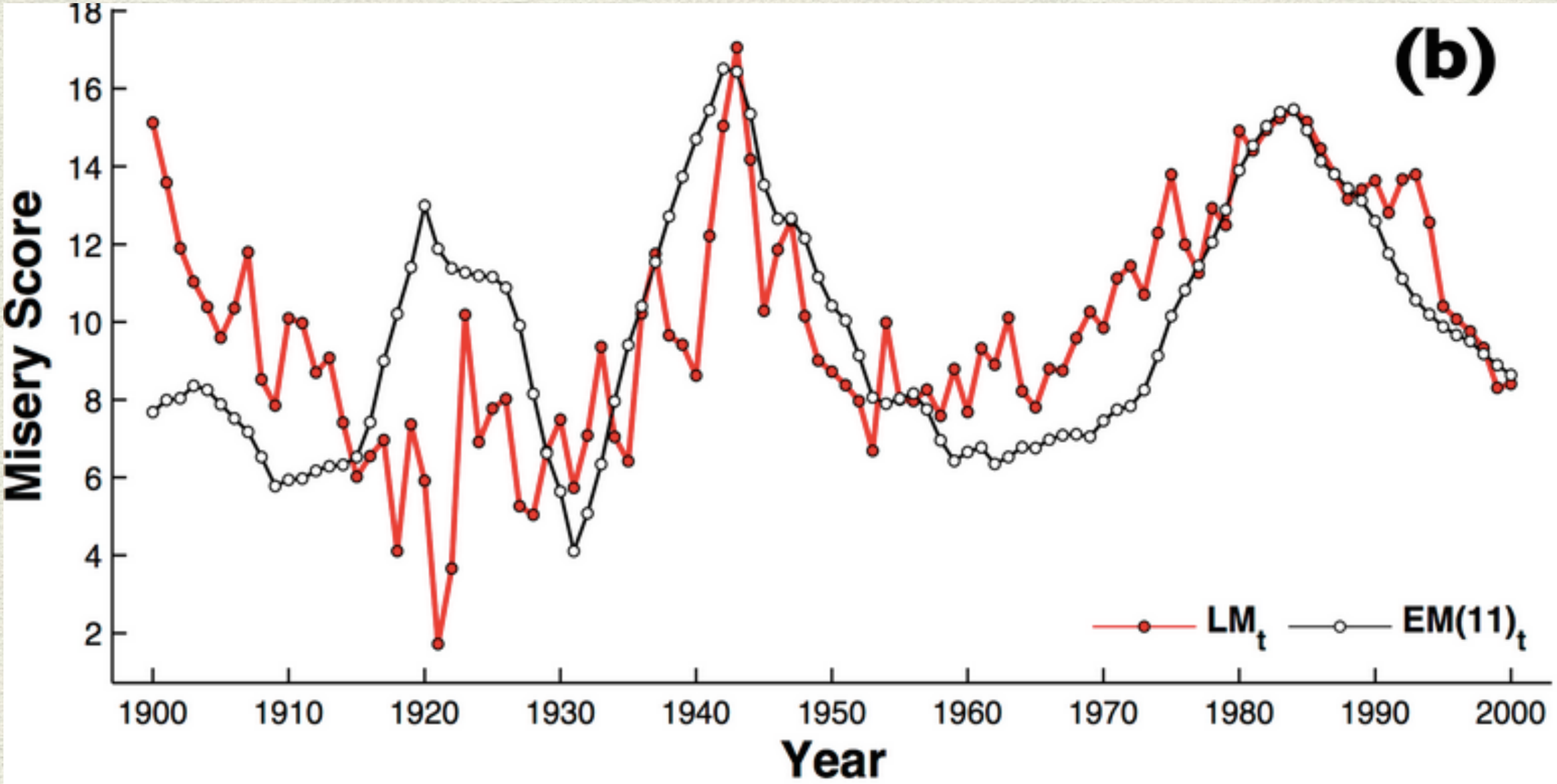
- EM: standard economic misery index, unemployment rate + inflation rate

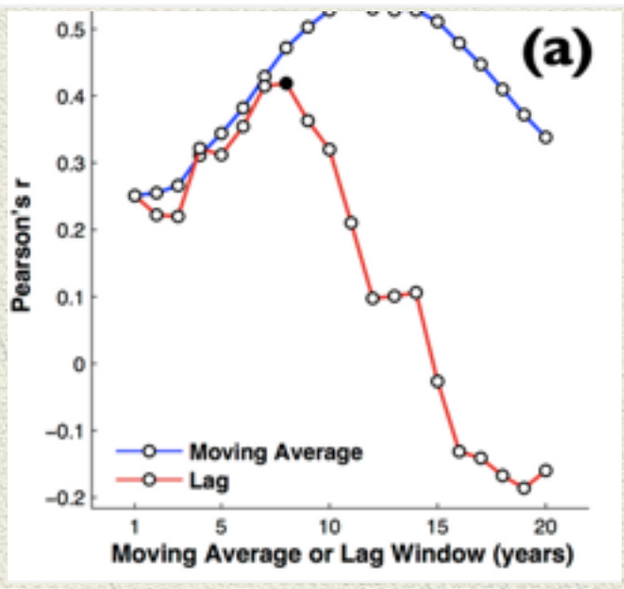
- Predicted LM would be proportional to moving average of annual US EM(x) where *x* is mean-time-window

1-year lag: Em(1)



Bentley, A., Acerbi, A., Ormerod, P., Lampos, V. (2014). Books Average Previous Decade of Economic Misery. *PLoS ONE 9 (1)*





Em(11) shows the best correlation

Different Normalisation…

- Bentley et al (2014) normalised each stemmed word in the sad/happy lists:
  - the size of the happy/sad list

> We used the WordNet Affect (henceforth WNA) text analysis tool which groups synonymous terms into lists related to mood states to perform text analysis on these words after they had been stemmed using Porter's Algorithm [66]. This method is consistent with numerous other text-mining studies such as [22], [67]. We considered six distinct main emotions: anger ($N = 146$), disgust ($N = 30$), fear ($N = 92$), joy ($N = 224$), sadness ($N = 115$), and surprise ($N = 41$).

  - number of "the" words in the given year; not on total number of words
- They also changed them into z-scores

For each stemmed word we collected the amount of 1-gram occurrences (case insensitive) in each year from 1900 to 2000 (both included). Following [38], we excluded data from years after 2000 because books published recently are still being included in the data set, and therefore latest records are incomplete and possibly biased. We normalized the frequency of the words in these word lists, then computed the average normalized frequency per year:

$$w_t = \frac{1}{n}\sum_{i=1}^{n}\frac{c_i}{c_{\text{the}}}, \tag{1}$$

where n is the number of words in the list, c_i is the word count for word i in the list in year t, normalized by c_{the}, which is the count of the most frequent English word, 'the', in year t. We normalized the yearly amount of occurrences using the occurrences, for each year, of the word 'the', rather than by the annual total number of words scanned, to avoid the effect of the influx of data, special characters, etc. that may have changed considerably over the 20th century [40], [68].

These normalized frequency scores were then converted to their z-score equivalents as

$$(w_t - \mu_w)/\sigma_w, \tag{2}$$

where μ_w and σ_w are the mean and standard deviation of w_t over the 100 years of the 20th century. We denote the z-score equivalents in year t for the 224 listed 'joy' words and the 115 listed 'sadness' words as J_t and S_t, respectively, and the difference between them, $LM_t = S_t - J_t$ as our literary misery index, LM_t.

Counting Search Terms

Counting in Other Corpora

- In theory, you can take any corpus of words/sentences/documents and do frequency counts on them
- Your corpus could be (anything once its make sense):
  - search terms (e.g., used in the last month)
  - a single book as a corpus
  - album-names in the billboard charts
  - some selection of news articles
- All you need to do is figure out what to count, how to normalise and capture some regularities in the data

Two Examples

- Search Terms for:
  - Predicting Car Buying
  - Predicting Flu Trends*

* Dealt with in detail in *Words In Time*

Eg1: Search Queries

- GoogleTrends: recording search queries rates over a corpus of search terms for a given day/week/month (Choi & Varian, 2009)
- Not predicting the future but the “present” !
- Daily and weekly reports of volume of queries in different industries

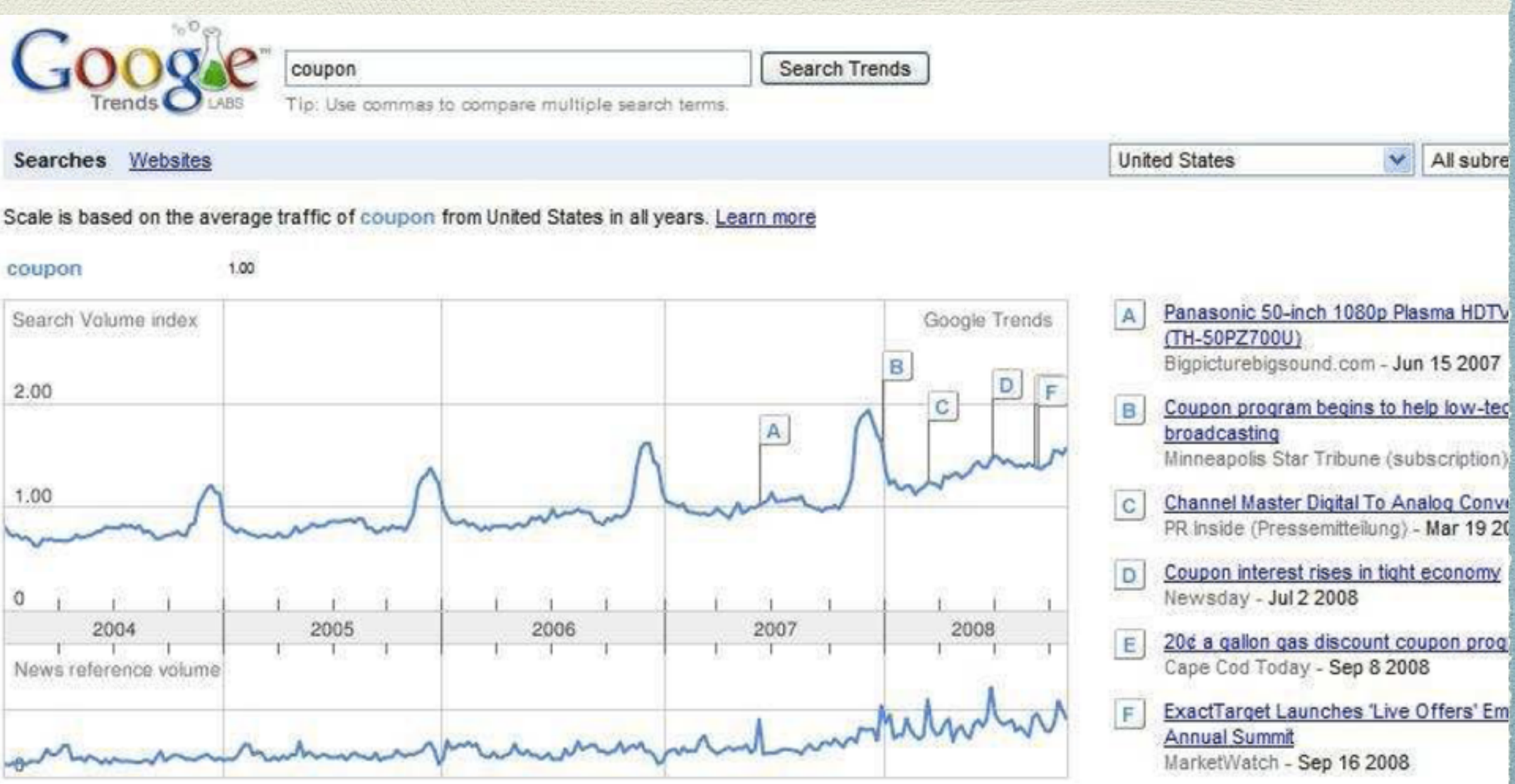
Search Queries: GoogleTrends

- Not raw queries but *query index*: based on query share and baseline normalisation

- *Query share*: the total query volume for search term in a geo-region (e.g. AZ, USA) divided by total no of queries in that region at time *t*

- Baseline: Zero @ Jan1 2004; figures are % deviation from query share on that date

coupon trends up with economic downturn and peaks around the pre-xmas period of each year



Choi, H. & Varian, H. (2009, 2012) Predicting the Present with Google Trends. *Economic Record*, 88(s1), 2-9.

Car Sales: The Model

Denote Ford sales in the t-th month as $\{y_t : \ t = 1, 2, \cdots, T\}$ and the Google Trends index in the k-th week of the t-th month as $\{x_t^{(k)} : \ t = 1, 2, \cdots, T; k = 1, \cdots, 4\}$. The first step in our analysis is to plot the data in order to look for seasonality and structural trends. Figure 1.2 shows a declining trend and strong seasonality in both Ford Sales and the Ford Query index.

We start with a simple baseline forecasting model: sales this month are predicted using sales last month and 12 months ago.

$$\text{Model 0:} \ \log(y_t) \sim \log(y_{t-1}) + \log(y_{t-12}) + e_t, \tag{1.1}$$

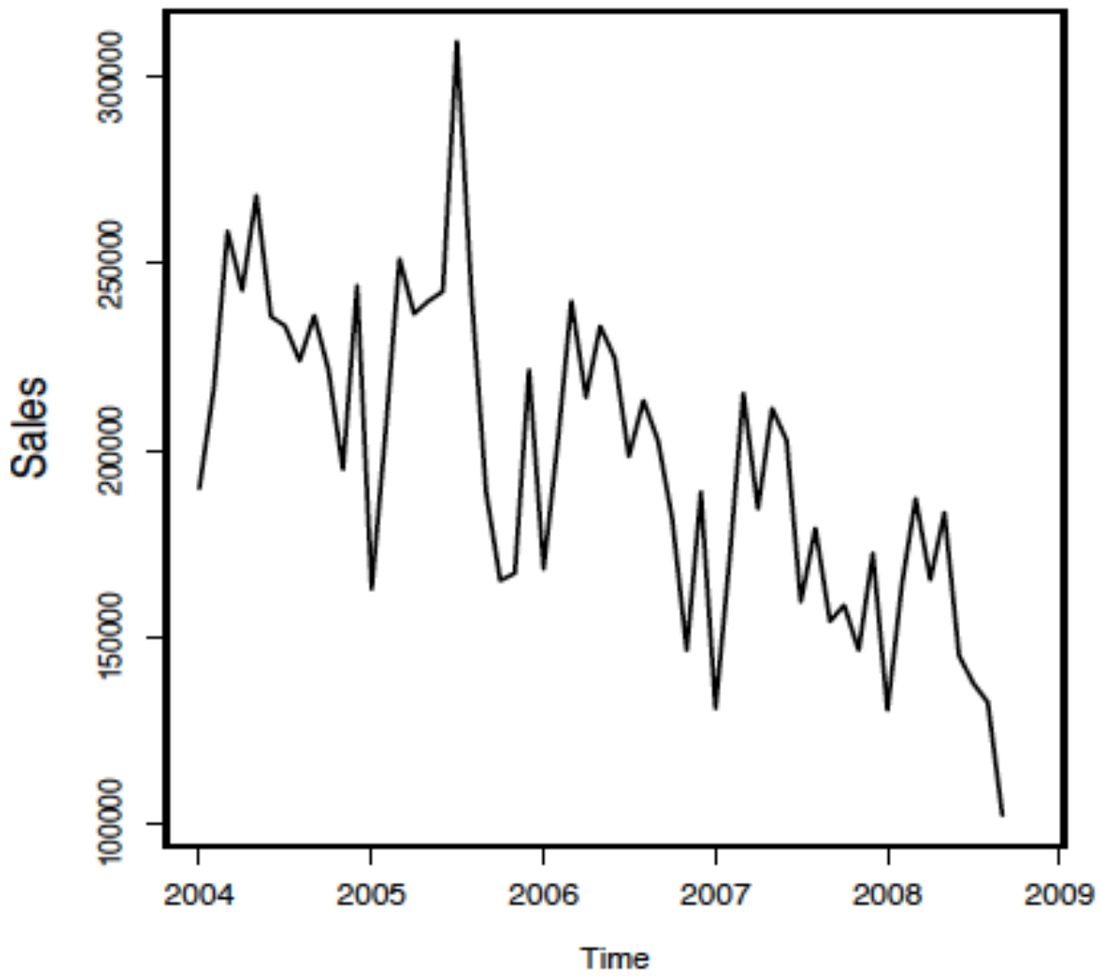
The variable e_t is an error term. This type of model is known in the literature as a *seasonal autoregressive* model or a *seasonal AR model.*

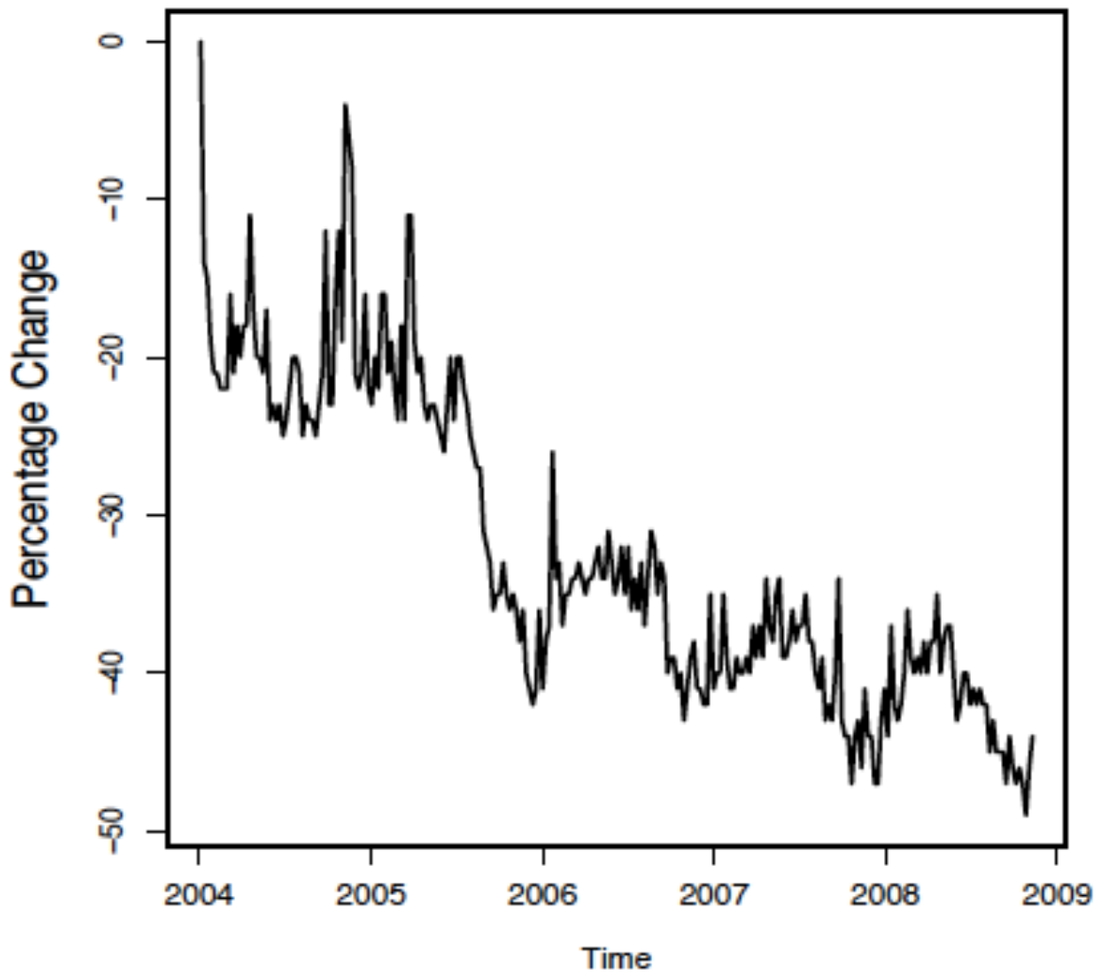
We next add the query index for 'Ford' during the first week of each month to this model. Denoting this variable by $x_t^{(1)}$, we have

$$\text{Model 1:} \ \log(y_t) \sim \log(y_{t-1}) + \log(y_{t-12}) + x_t^{(1)} + e_t \tag{1.2}$$

Choi, H. & Varian, H. (2009, 2012) Predicting the Present with Google Trends.Economic Record, 88(s1), 2-9.

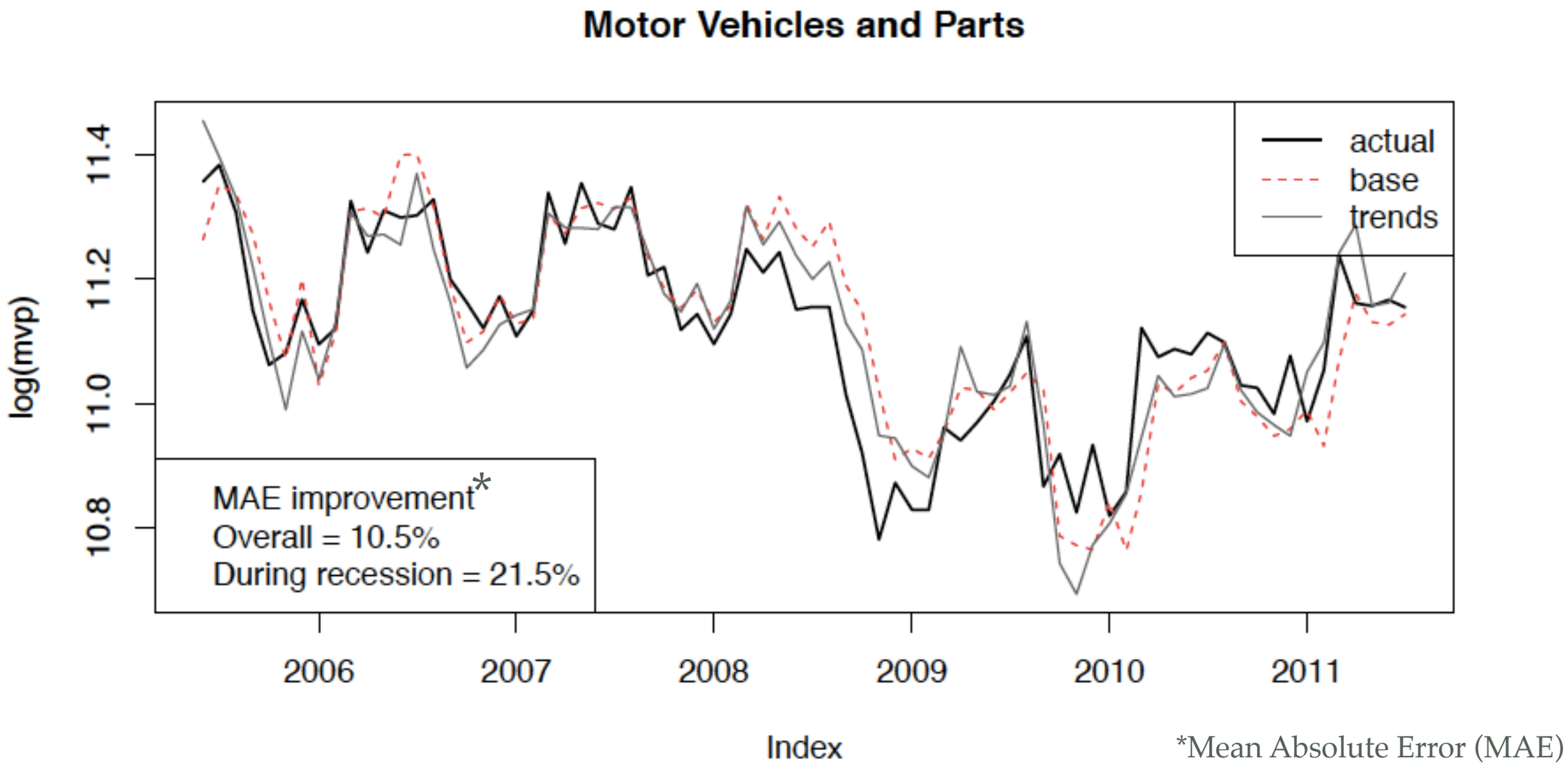
GoogleTrends: Car Sales





Choi, H. & Varian, H. (2009, 2012) Predicting the Present with Google Trends.Economic Record, 88(s1), 2-9.

Parts: Model Results



*Mean Absolute Error (MAE)

Choi, H. & Varian, H. (2011) Predicting the Present with Google Trends.Economic Record, 88(s1), 2-9.

Search Queries: Lessons

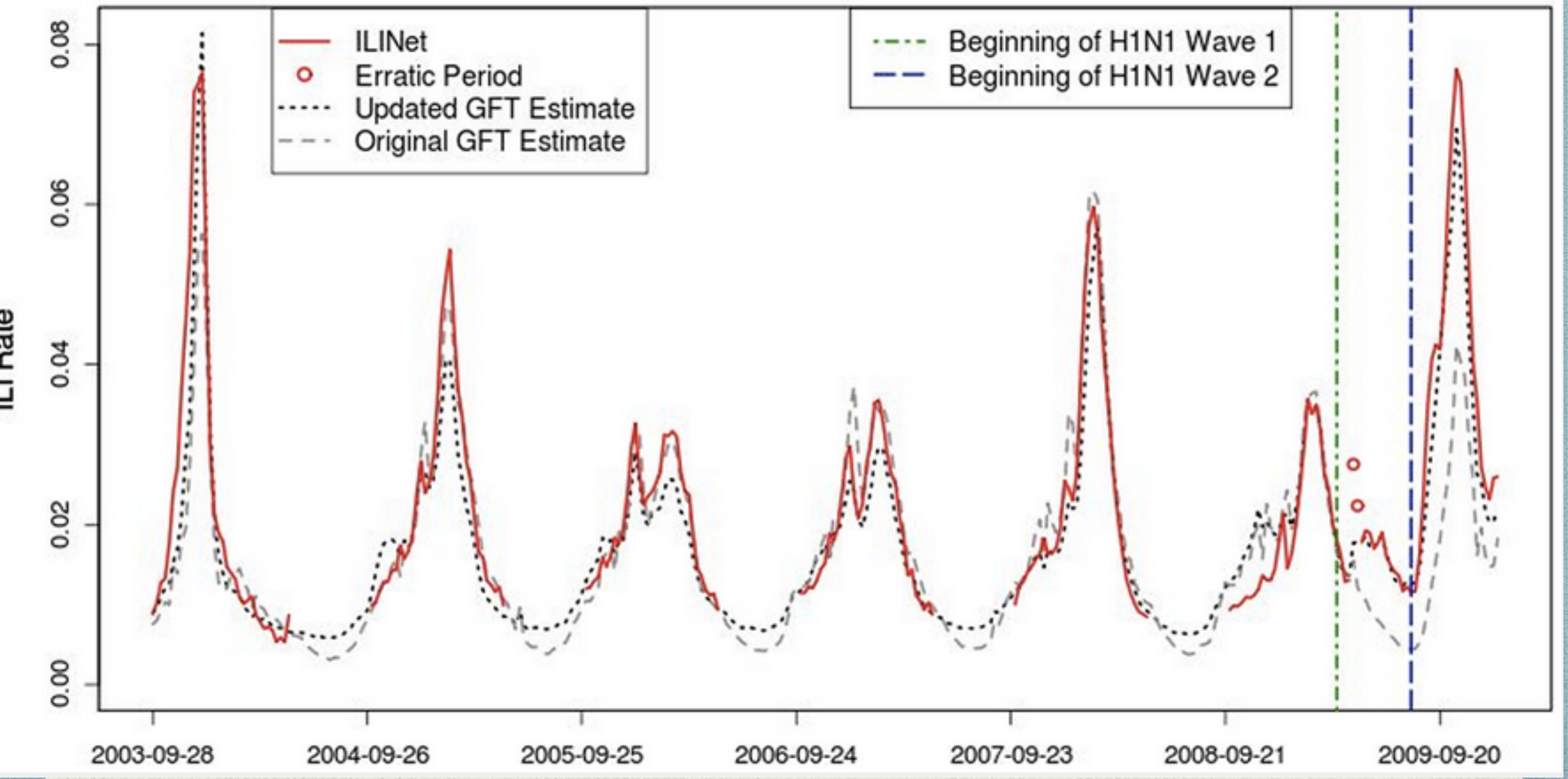
- Using standard economic modelling, simple linear models
- Simple normalisation of query-term volume, supported by NLP classifier
- Is applied to several categories of purchases
- As we shall see in flu-case, no model of search

Eg2 Search Terms: Flu

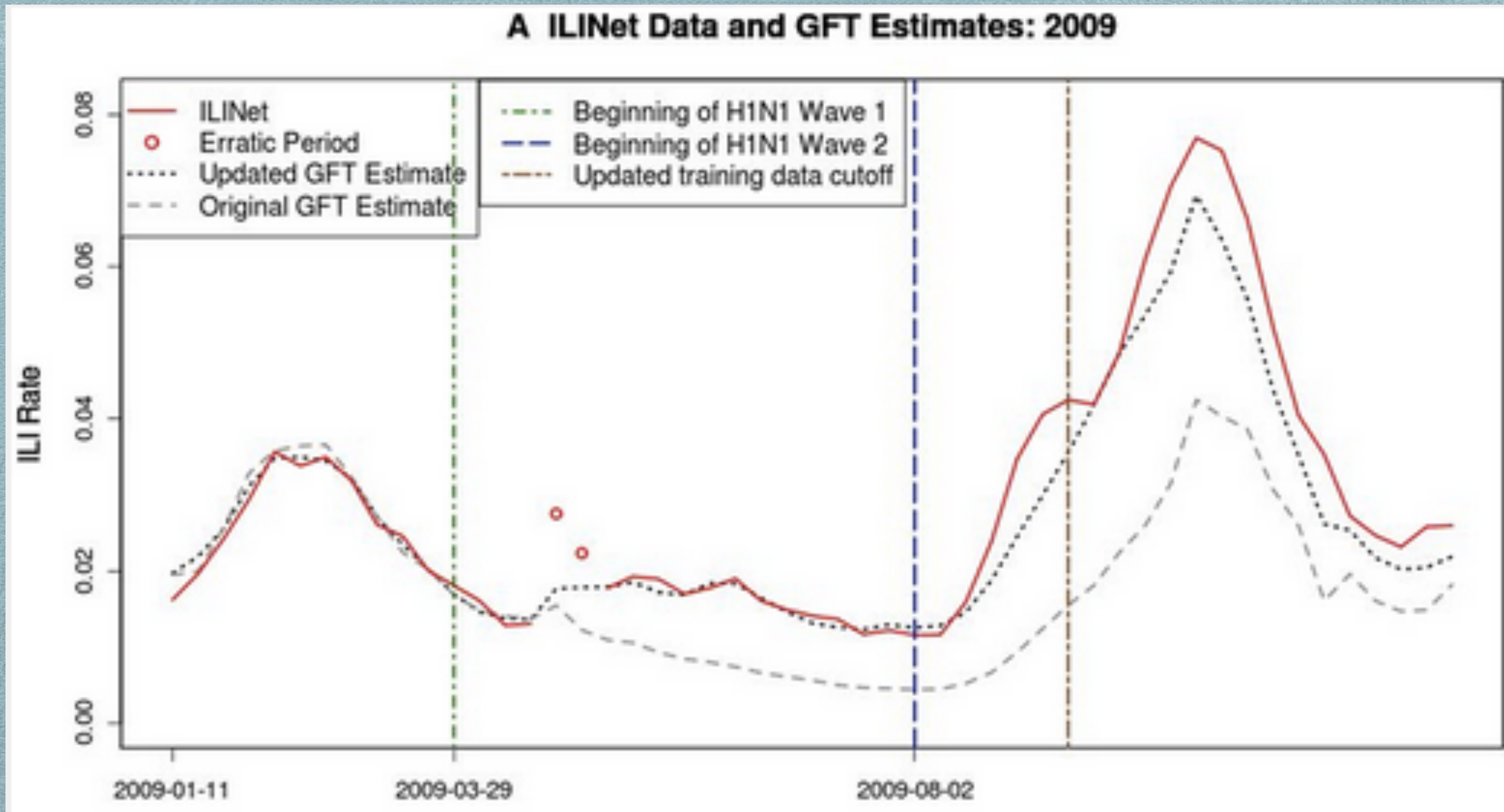
- Google Flu Trends (DFT) aggregates search data, count in flu keywords

- US Centre for Disease Control (ILIs) tracks influenza like illnesses in outpatient data

- From 2003-2009 GFT shows high correlation with ILI stats (ILINet) until 2009 H1N1 Pandemic (pH1N1)



Cook, S., Conrad, C., Fowlkes, A. L., & Mohebbi, M. H. (2011). Assessing Google flu trends performance in the United States during the 2009 influenza virus A (H1N1) pandemic. PloS one, 6, e23610.



- Google modify model in 2009, changing search terms

Search Terms: Lessons

- Google did not reveal search terms; modified to better fit…
- Raises deep issue about selection of terms to track; in a principled way is not clear
- You need a model of search behaviour; in H1N1 search behaviour changed and prediction failed

Distributions of Word Counts

Eg1: Moby Dick

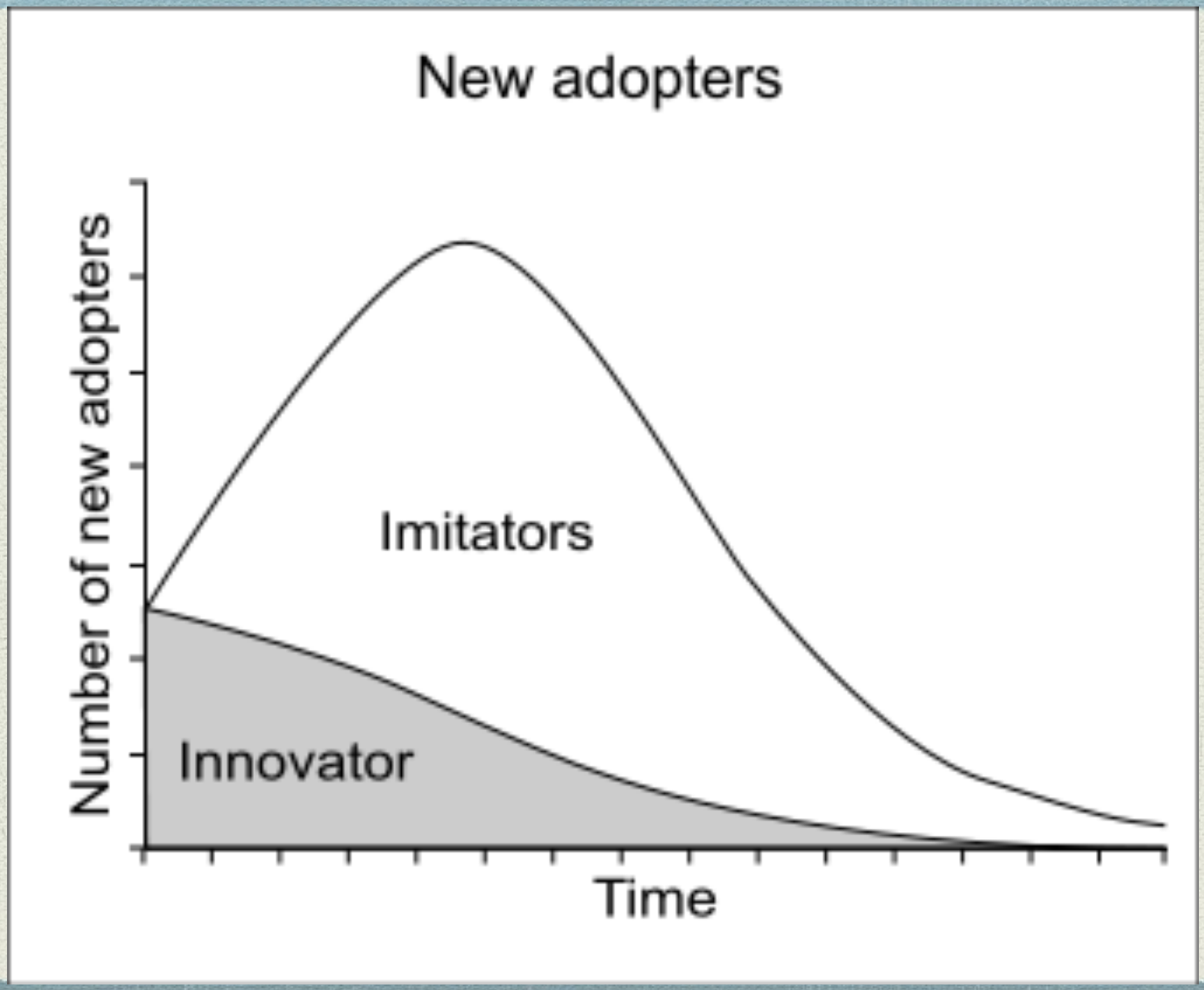
- Using a book as a corpus or a set of plays to determine authorship
- Using all the albums entering the Billboard-100 over 20-30 years
- Using a selected set of news articles

REM

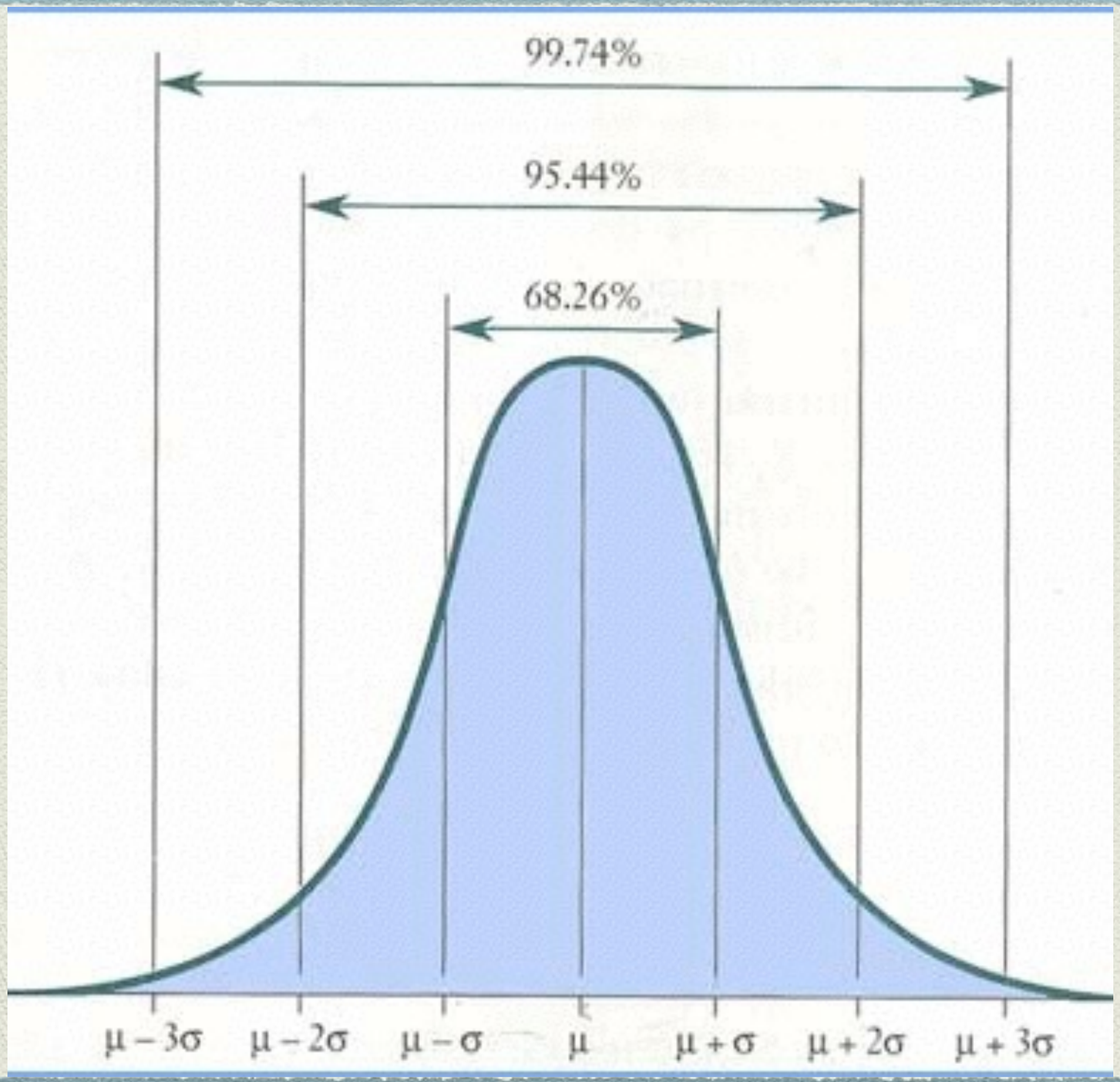
Counting in Other Corpora

- In theory, you can take any corpus of words/sentences/documents and do frequency counts on them
- Your corpus could be (anything once its make sense):
  - search terms (e.g., used in the last month)
  - a single book as a corpus
  - album-names in the billboard charts
  - some selection of news articles
- All you need to do is figure out what to count, how to normalise and capture some regularities in the data

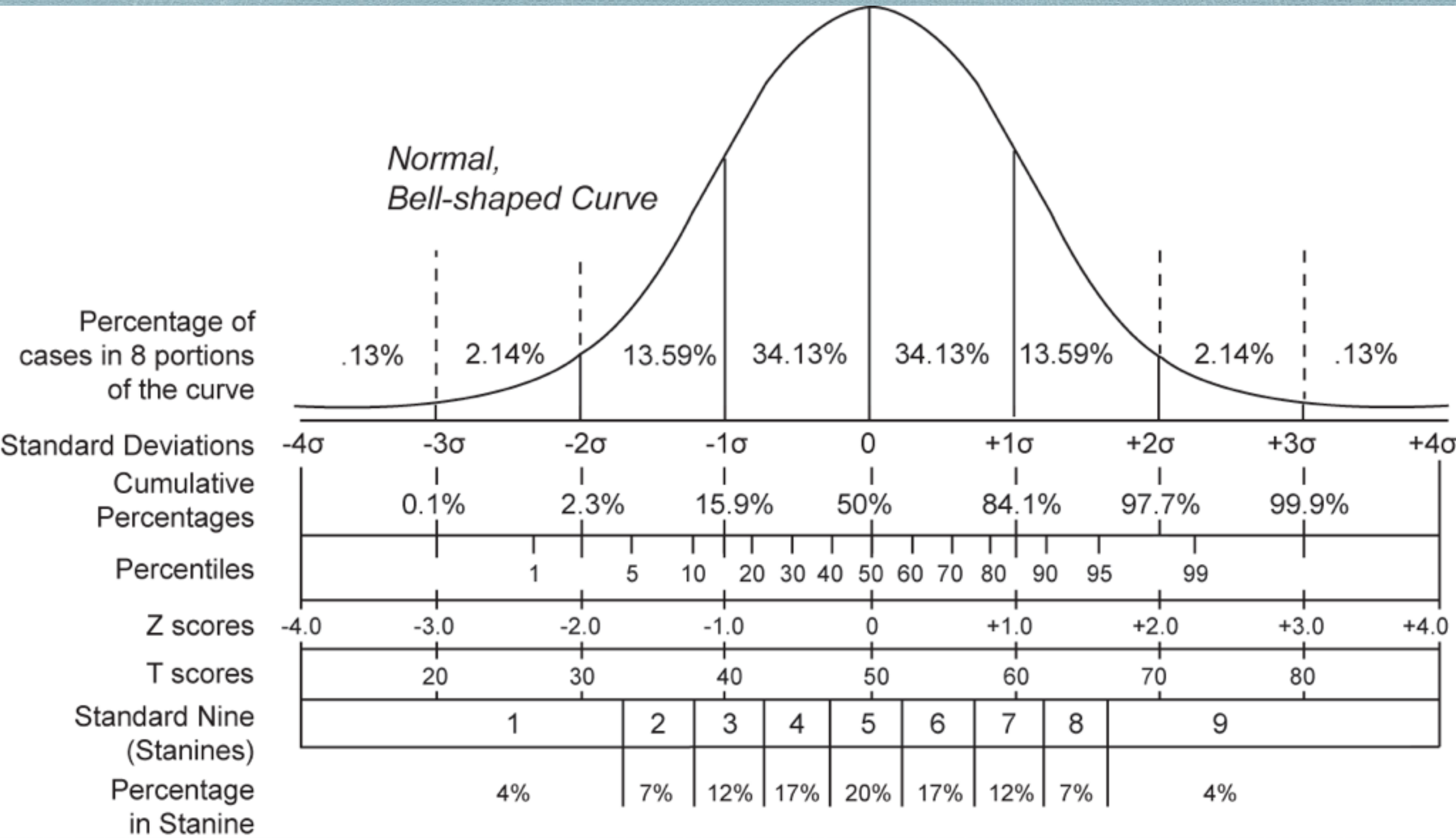
Bass Distributions



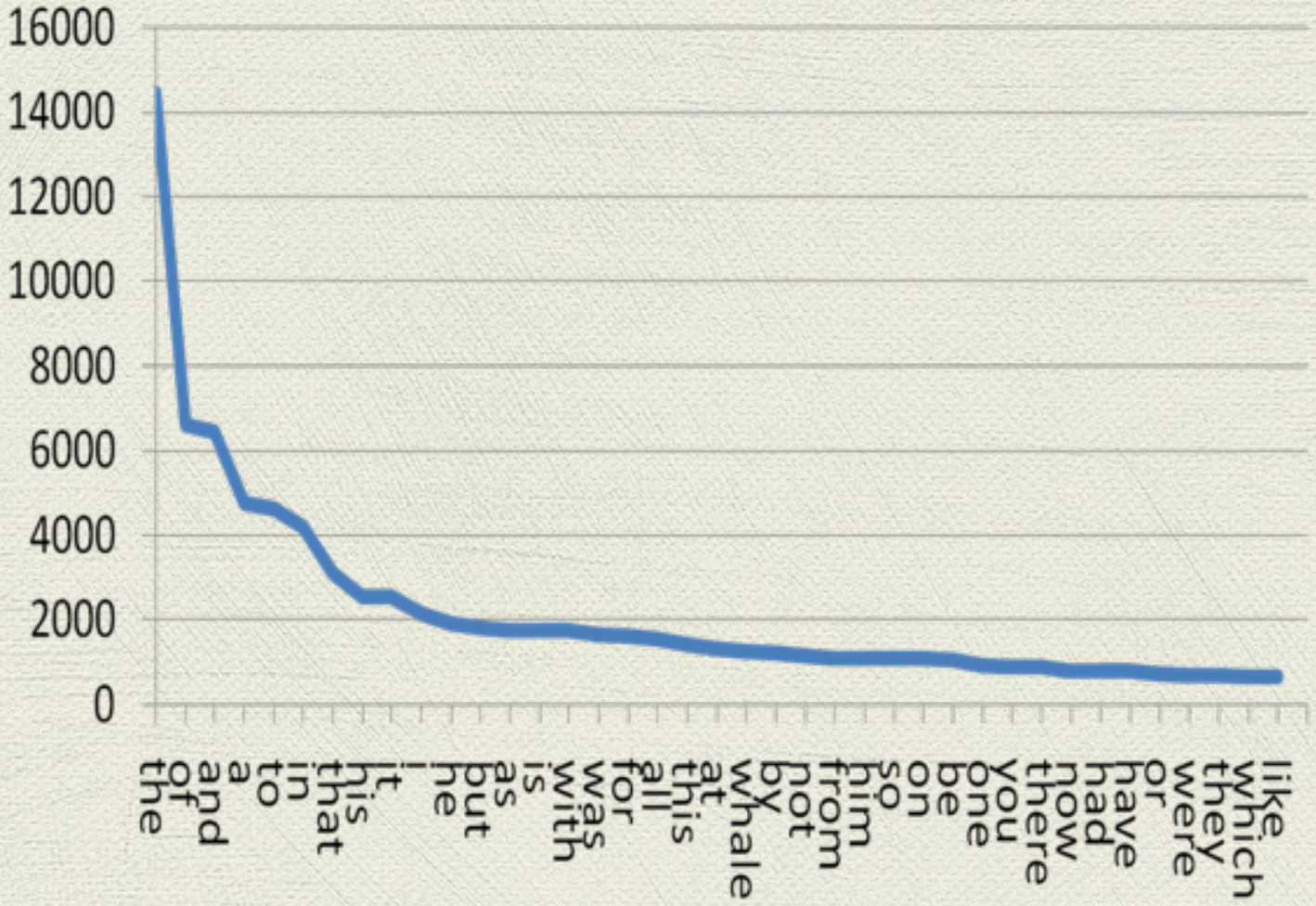
Normal Distribution

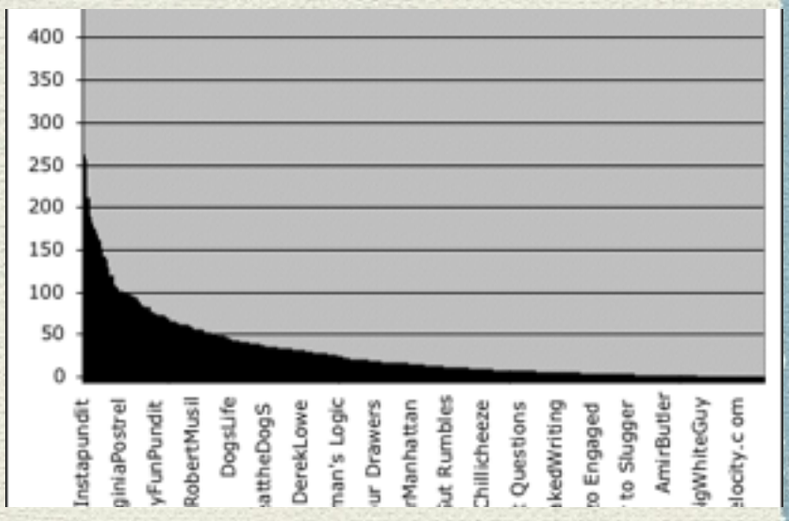


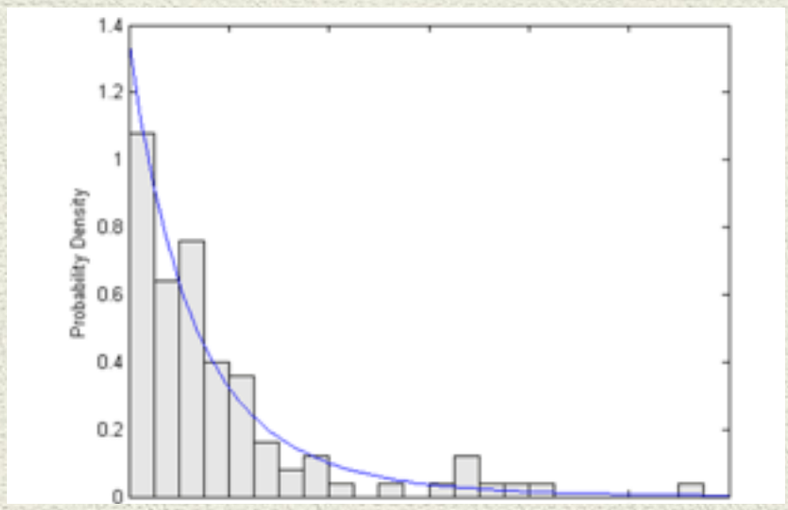
Normal Distribution

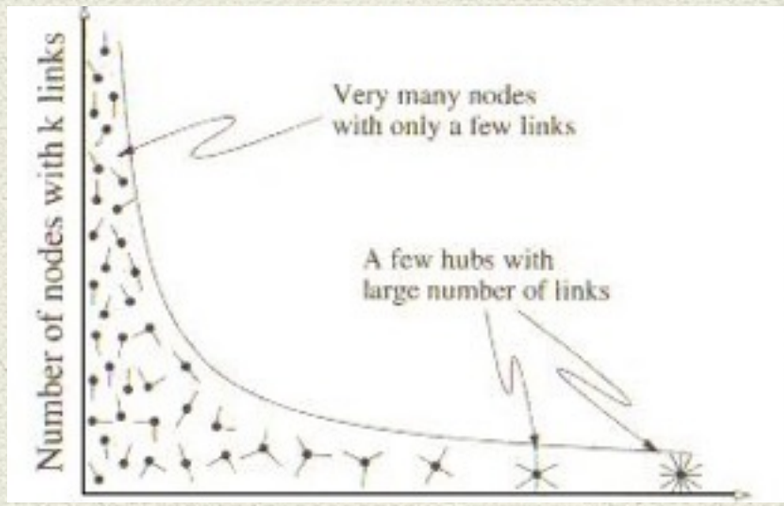


Power Law Distributions









Book as a Corpus

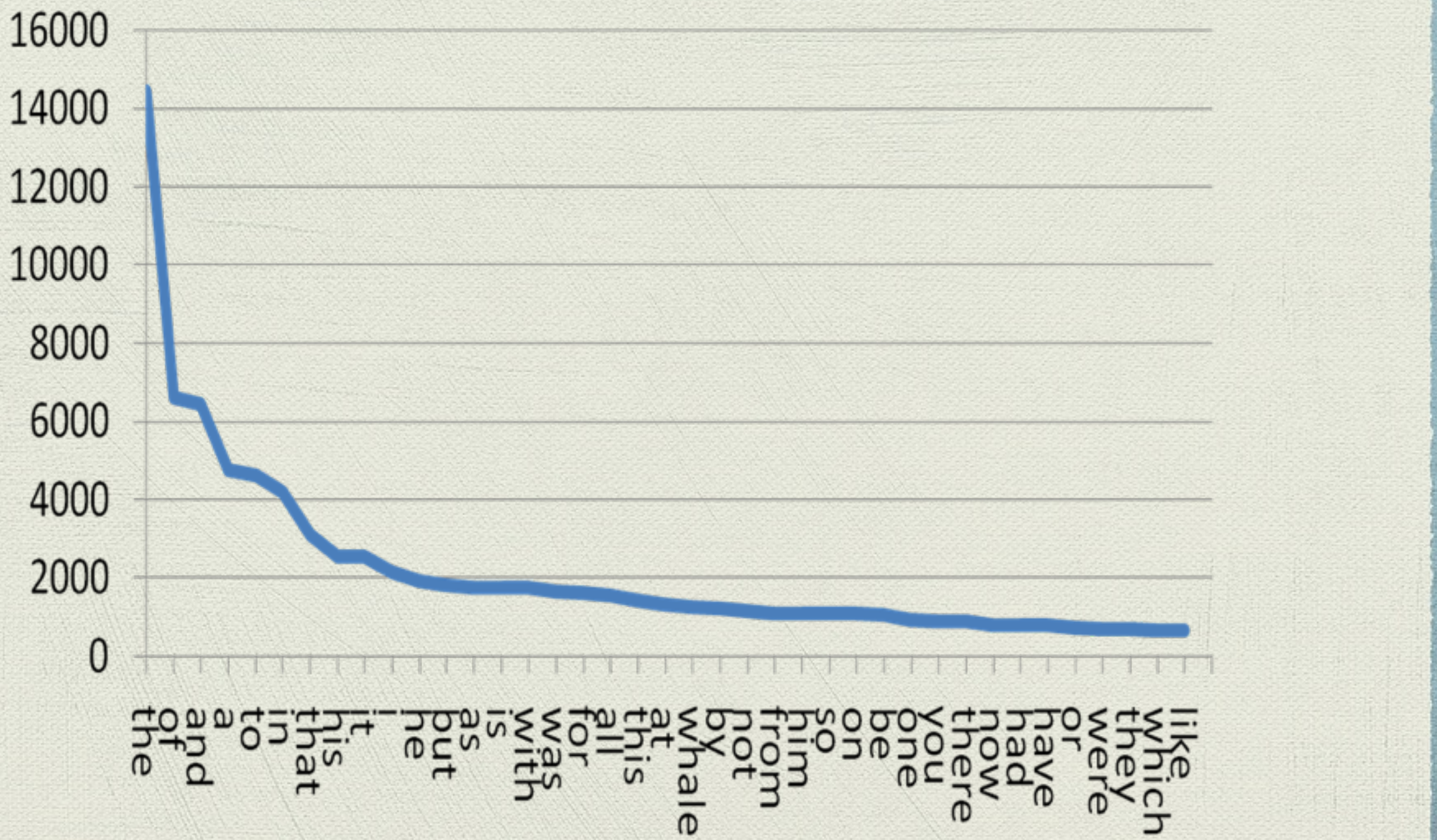
- Linguist Zipf (1949) studied the frequency of words in Merville's *Moby Dick* (1851)

- Any communication system (language) has constraints*: brevity, memorability, simplicity

- Zipf ended up with a law with universal applicability across many comm. systems

* 2-3 rules define sound changes in language evolution

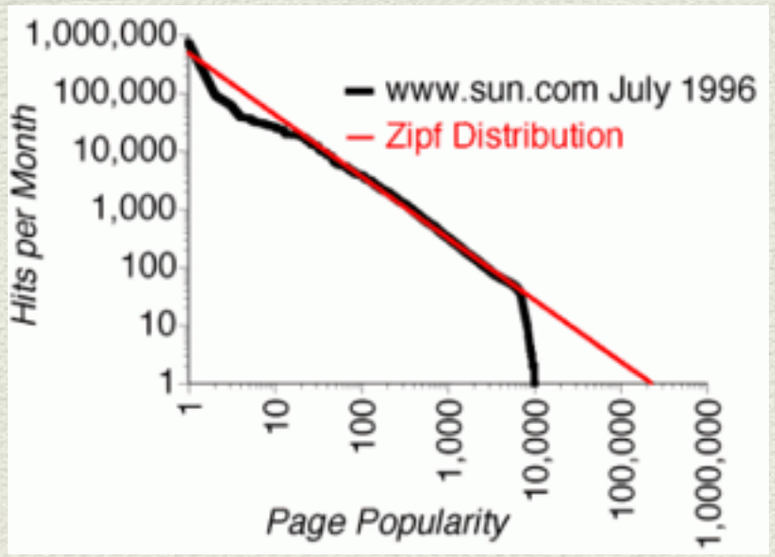
Words in a Book = Power Law



Zipf's Law

- Zipf's law, while only a few words are used very often, many or most are used rarely:

$$P_n \sim 1/n^a$$



- where Pn is the frequency of a word ranked nth and the exponent *a* is almost 1. 2nd is 1/2 of 1st, 3rd is 1/3 as often as first, and so on

- Power-law comes from plotting as ranks or getting logs of frequencies, which logs as a straight line

Zipf's Law

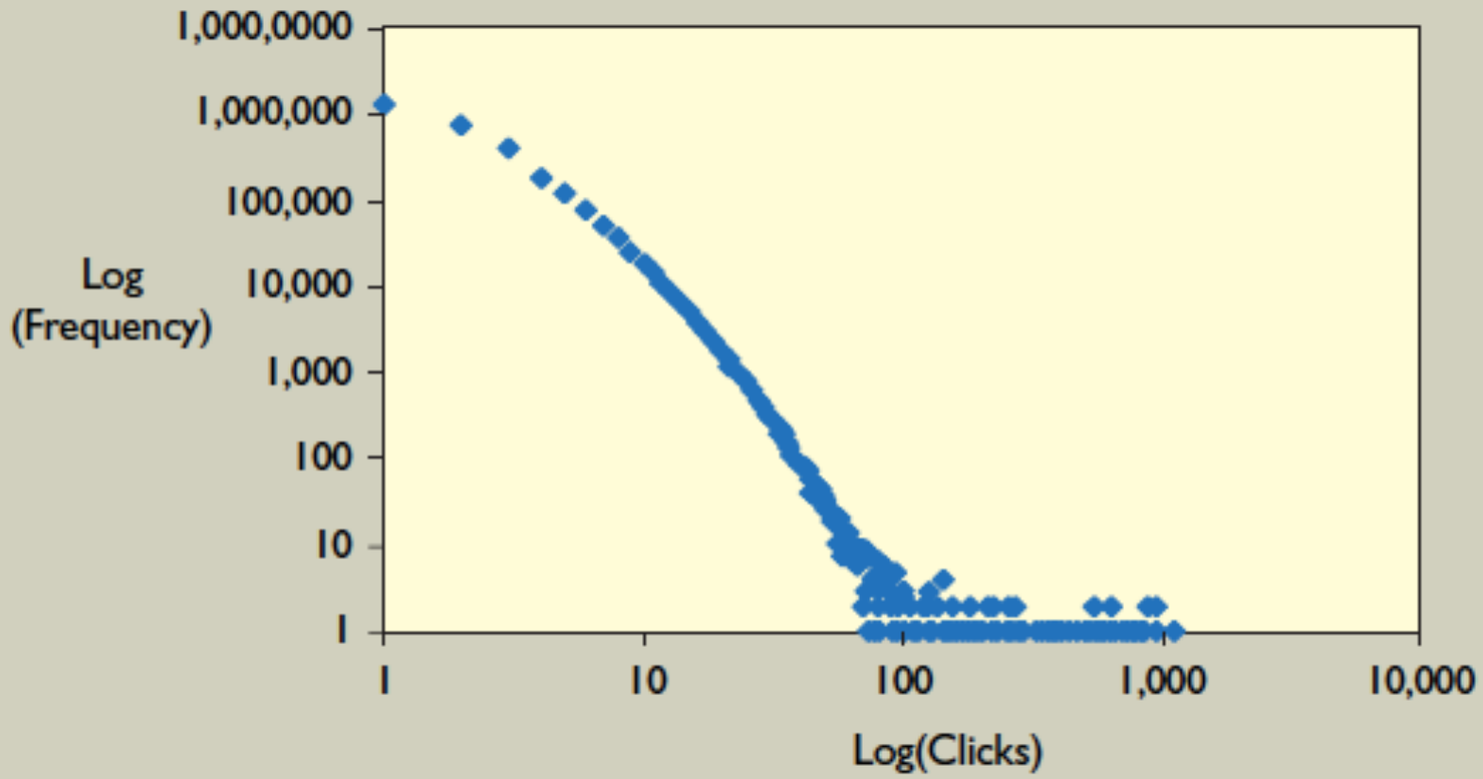
As early as the 1940s, George Zipf found that the frequency distribution of words in Moby Dick [Melville, 1851], and other corpora, follow a regular power-law with the generalized form:

$$y = Cx^{-\alpha} \qquad (1)$$

with $C = e^{c}$ [Estoup, 1916; Newman, 2006; Zipf, 1949]. When power-law distributions are plotted in log-rank, log-frequency form, the data exhibit a linear slope equal to α; in Zipf's Law for English, α is near 1.

Zipf's Law: Wider…

- Rank vs. frequency distribution of individual incomes in a unified nation approximates this law
- Universal laws of surfing (Huberman & Adamic, 1999; Halvey et al., 2006) on web and mobile-web



By Martin Halvey, Mark T. Keane, *and* Barry Smyth

Mobile Web Surfing *is the* SAME *as* Web Surfing

Even in the new context, users surf in the usual way.

80/20 Rule…

- Pareto Principle
- 80% of the wealth is owned by 20% of people; 80% of your sales come from 20% of your clients
- 80/20 is really a short hand for describing power laws
- See also Piketty (2012), *Capital in the 21st Century*

V.F. PARETO (1848-1923)

VILFREDO FEDERICO DAMASO PARETO (ITALIAN PRONUNCIATION: [VIL'FRE:DO PA'RE:TO]; 15 JULY 1848 – 19 AUGUST 1923), BORN **WILFRIED FRITZ PARETO**, WAS AN ITALIAN INDUSTRIALIST, SOCIOLOGIST, ECONOMIST, AND PHILOSOPHER.

REM

Counting in Other Corpora

- In theory, you can take any corpus of words/sentences/documents and do frequency counts on them
- Your corpus could be (anything once its make sense):
  - search terms (e.g., used in the last month)
  - a single book as a corpus
  - album-names in the billboard charts
  - some selection of news articles
- All you need to do is figure out what to count, how to normalise and capture some regularities in the data

Eg 2: Pop Charts

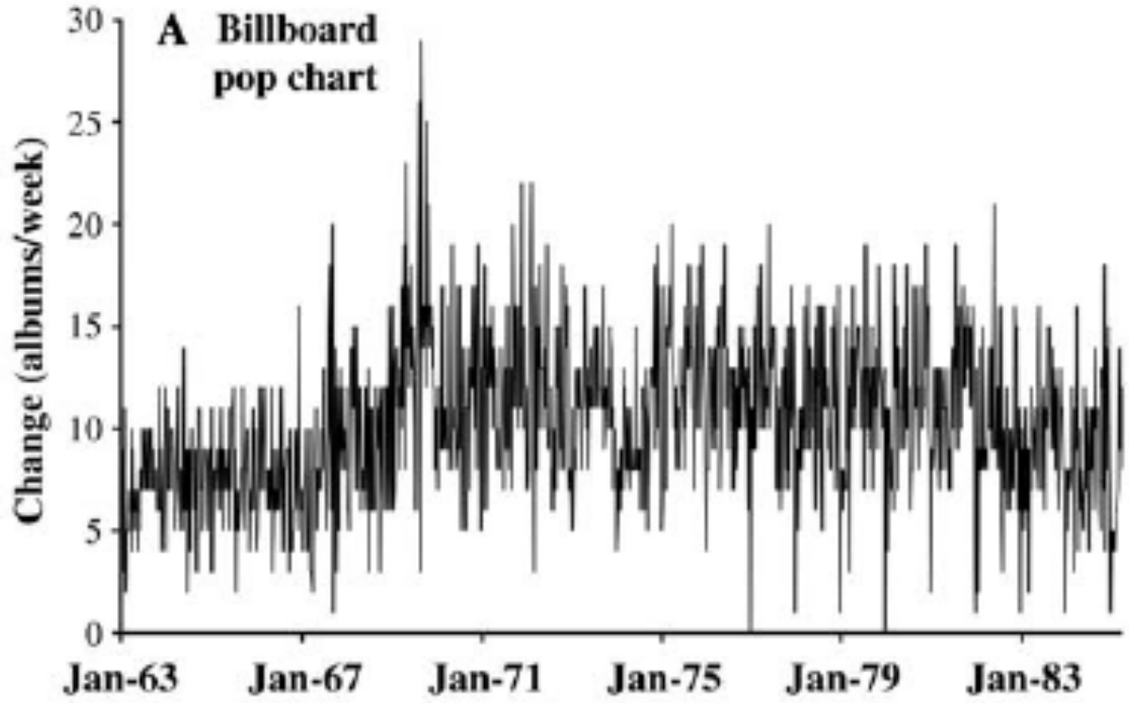
- Corpus: 12,005 albums from Billboard Chart (1963-1985) varying size, 200 since 1967

- 5.6% turnover per week, 10-20 week life for most (DSM, Pink Floyd, 566 weeks pre-85)

- Modelled using random-copying model

Alexander Bentley, R., & Maschner, H. D. (1999). Subtle nonlinearity in popular album charts. Advances in Complex Systems, 2(03), 197-208.

Bentley, R. A., Lipo, C. P., Herzog, H. A., & Hahn, M. W. (2007). Regular rates of popular culture change reflect random copying. Evolution and Human Behavior, 28(3), 151-158.



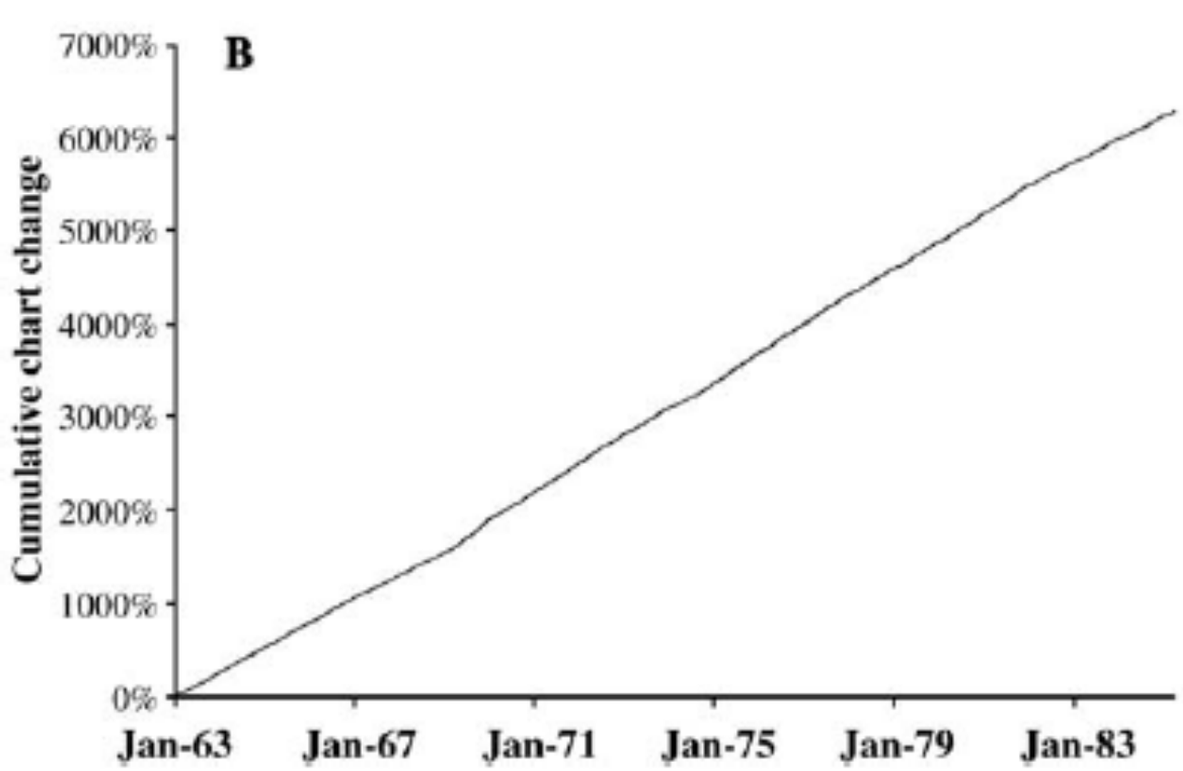


Fig. 3. (A) Weekly turnover on the Billboard Pop Chart, 1963–1985, in terms of the numbers of albums exiting the chart each week. (B) Cumulative change on the Pop Chart, in terms of the fraction of albums on the chart (z_y/y) exiting each week. The denominator y in calculating this fraction was variable, as the chart was expanded from 150 to 200 in mid-1967, and the actual value of y varies slightly from week to week (due to albums' shared positions on the chart, etc.). The turnover rate averaged 5.6% per week for over 20 years. Adapted from Bentley and Maschner (1999).

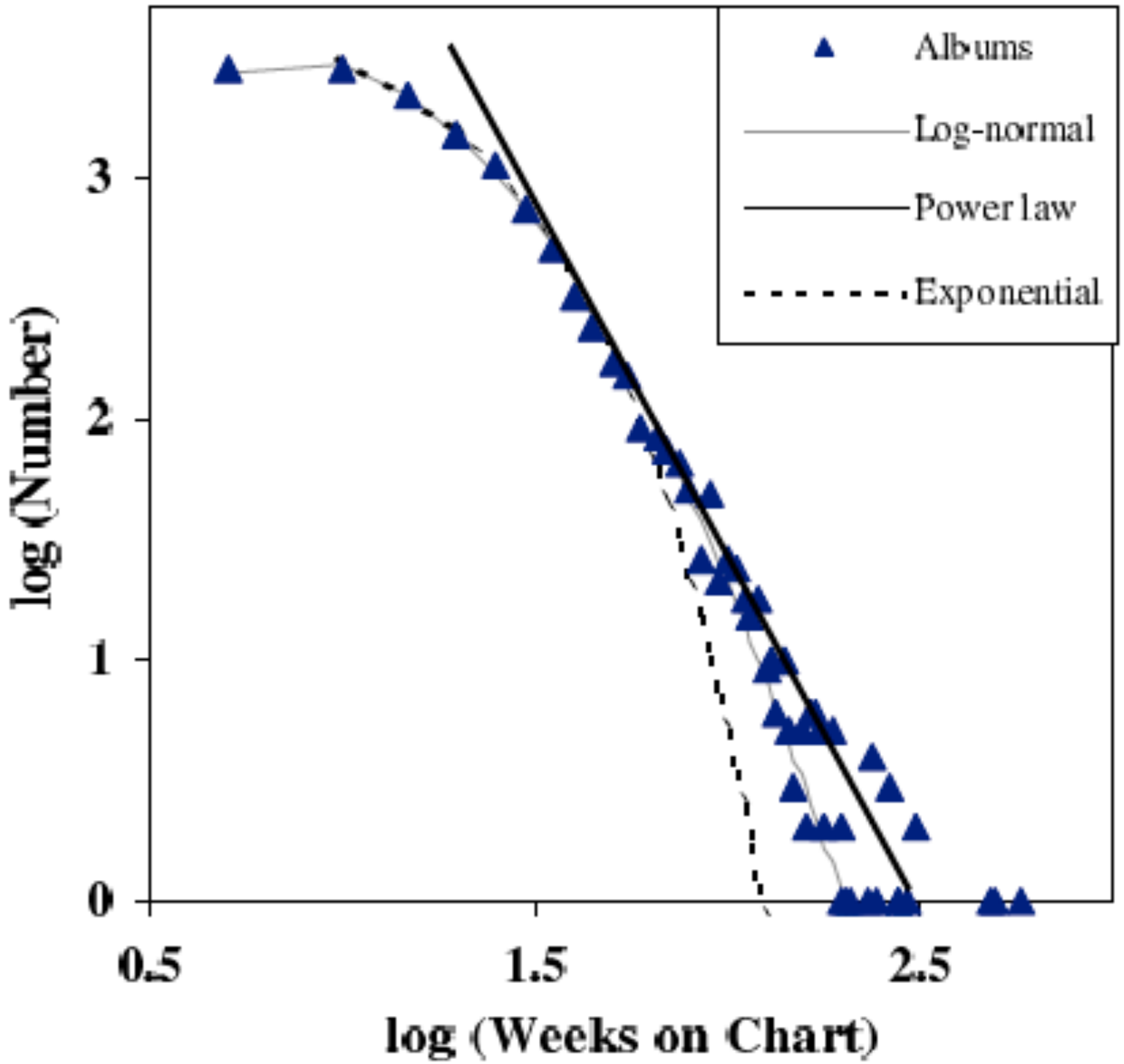


Fig. 4. Lifespans of albums on the Billboard Chart, 1963–1985. The bin size is five albums. Beyond 20 weeks, the slope converges asymptotically to -2.70 ± 0.12, $r^2 = 0.913$. The power-law fit for albums living 20 weeks or more is better than for the exponential relationship, but the log-normal distribution fits the entire distribution.

REM

Counting in Other Corpora

- In theory, you can take any corpus of words/sentences/documents and do frequency counts on them
- Your corpus could be (anything once its make sense):
  - search terms (e.g., used in the last month)
  - a single book as a corpus
  - album-names in the billboard charts
  - some selection of news articles
- All you need to do is figure out what to count, how to normalise and capture some regularities in the data

Eg3: Stock Market Changes

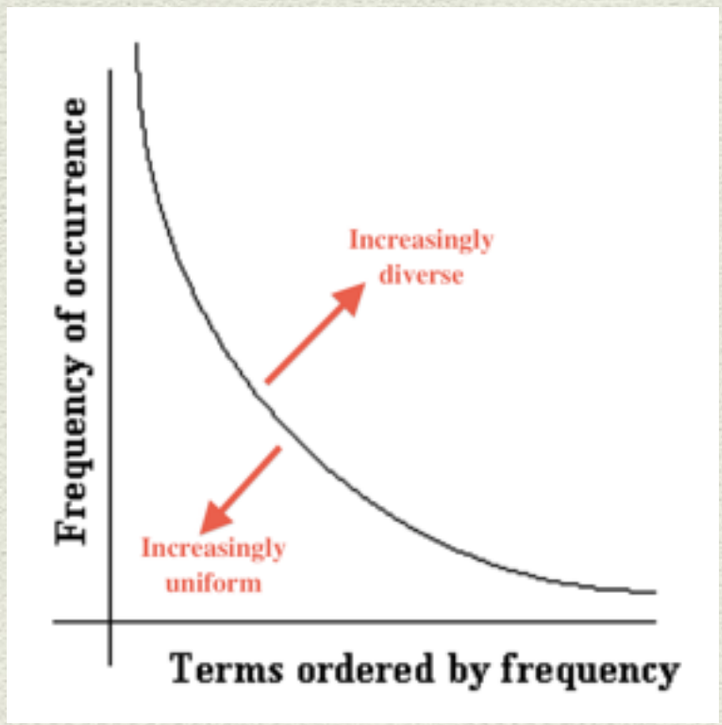
- Trying to predict stock-market changes from the language of news (financial articles)

- Four years of all financial articles from BBC, NYT and Financial Times sites mentioning markets (Dow Jones, FT-100, NIKKEI)

- Full parse, lemmatisation of articles to identify verbs, nouns, lemma-object-pairs

Voice of Herd: Iron-Bar Theory

- If we are in a room talking about the same thing, a power-law of the words will look one way, if we are all talking about different things it will be different



- A bubble is a period when everyone is in agreement; so they are using the same words, all being positive, mentioning the same companies

- This should appear in the counts of nouns/verbs used; looking at changes in alpha

$$y = Cx^{-\alpha}$$

with $C = e^{c}$

Eg3: The Corpus

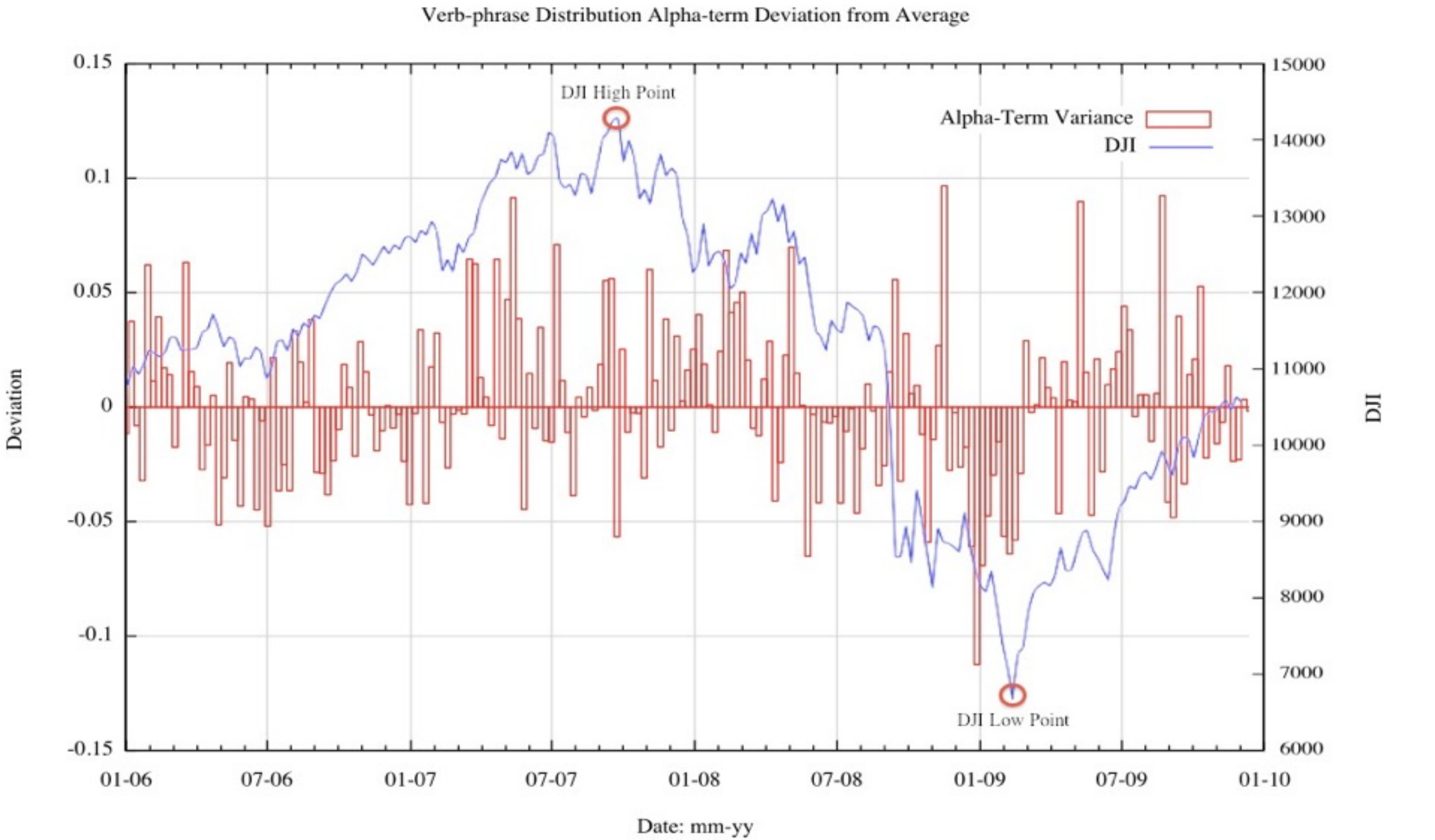
17,716 articles	**1 Jan 2006 – 10 Jan 2010**
Financial Times	13,286 (ft.com)
New York Times	2,425 (nyt.com)
BBC	2,001 (bbc.co.uk/news)
2006	3,869
2007	4,704
2008	5,044
2009	3,960
2010	136

10.5 M words
300 K sentences
4 M noun-phrases
1.5 M verb-phrases

Eg3: Text Processing

- Articles coded for source, author, publication date
- Shallow parsed (Apple Pie Parser)
- Lemmatised & POS tagged (Sketch Engine, TreeTagger)
- Regress for α in every week's verb-phrase distribution
- Plot weekly differences from corpus average

Eg3: Results


Verb-phrase Distribution Alpha-term Deviation from Average
DJI High Point
DJI Low Point
Alpha-Term Variance
DJI
Deviation
DJI
Date: mm-yy
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
15000
14000
13000
12000
11000
10000
9000
8000
7000
6000
01-06
07-06
01-07
07-07
01-08
07-08
01-09
07-09
01-10

Eg3: Results II

- Need to smoothen alpha-change and window it to see correlation

Geometric mean respects non-linearity of residuals:

$$\left(\prod_{i=1}^{n} a_i\right)^{1/n} = \sqrt[n]{a_1 a_2 \cdots a_n}.$$

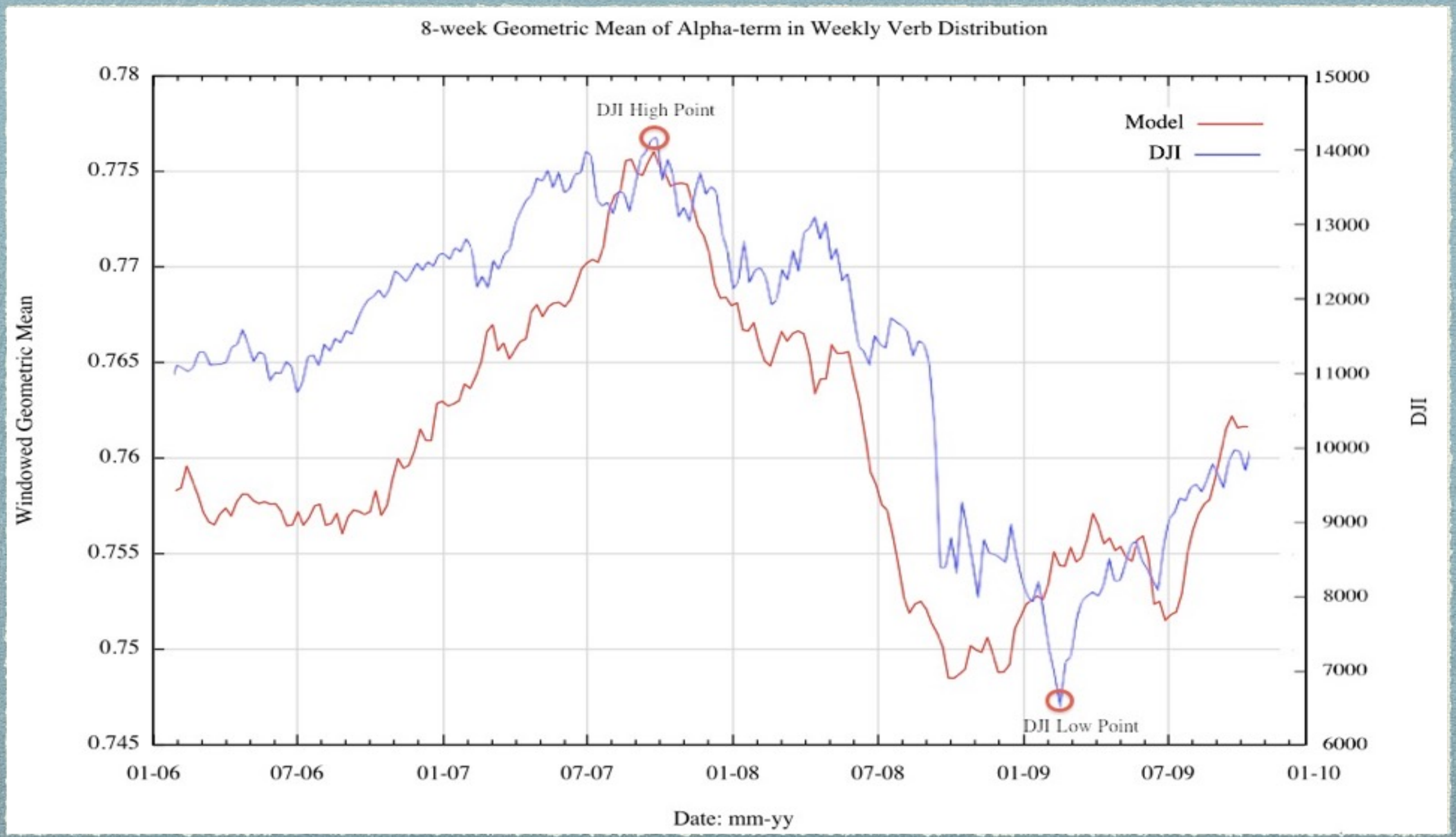
The model: take a windowed average of *α*'s

$$week_i = \frac{(\sqrt[8]{a_{i-3} \ldots a_i \ldots a_{i+4}} + \sqrt[8]{a_{i-4} \ldots a_i \ldots a_{i+3}})}{2}$$

Eg3: Results III

$$y = e^{10.8407\ -0.8137\ x}$$

$$r = .79\ ;\ p < .001$$



Eg3: Issues

- Very interesting, that you can capture regularities in the views of a population of journalists
- But, it is not predictive, as it relies on the deviation from the mean α for the whole four years
- More work required for prediction; e.g. autoregression, stationarity, and extrapolation

Conclusions

We Have Seen…

- Using Simple Frequency as Word Clouds
- Counting different categories of words (search terms, a;bum names) from different corpora (books, a-book, billboard, etc)
- Looking at Distributions of Frequencies

Simple Frequency

- Word Clouds are fine as a visualisation
- Relatively simple as analytics
- Need to extend to corpora, time, distributions

Summary: Corpora I

- Using GoogleNgram Corpus to count:
  - People mentions, idea mentions and celebrity
  - Climate term changes
  - Misery terms in an economy
- Using GoogleTrends Search Queries Corpora:
  - to predict purchases
  - to track flu outbreaks

Summary: Corpora II

- Book Corpus:
  - to predict authorship
- Billboard Album Corpus:
  - to capture turnover to model it
- New Article Corpus:
  - to predict stock-market changes

Techniques Used

- Simple counting
- Building frequency distributions
- Pre-processing text: removal, pos, stemming
- Normalising word / text items
- Later, we will look at frequencies over time

Remember Lect2…

Important Point

REM

- Pre-processing is not just about taking things out; stripping off stems, removing stops etc…
- It may also be about putting things in; like POS tags, syntax, entity tags, lexical chains

Pre-processing Non-Trivial

Normalisation Non-Trivial

Corpus Selection Non-Trivial

Item Selection Non-Trivial

Finally, we have assumed…

REM

- That you just know which texts to pre-process; but, sometimes you have to think about selecting the texts that make up a corpus
- Is this defined naturally; every debate in the Dail since 1922…(simple case)
- Every news article about stock markets…how do we define this ? (medium case)
- Every tweet that is about senate elections …how do we define this ? (hard case)