Text as Data

Justin Grimmer

Professor
Department of Political Science
Stanford University

September 10th, 2019

Text and Social Science

A pre-2000's view of text in social science

- Social interaction often occurs in texts

Text and Social Science

A pre-2000's view of text in social science

- Social interaction often occurs in texts
- Social Scientists avoided studying texts/speech

Text and Social Science

A pre-2000's view of text in social science

- Social interaction often occurs in texts
- Social Scientists avoided studying texts/speech
- Why?

Text and Social Science

A pre-2000's view of text in social science

- Social interaction often occurs in texts
- Social Scientists avoided studying texts/speech
- Why?
    - Hard to find

Text and Social Science

A pre-2000's view of text in social science

- Social interaction often occurs in texts
- Social Scientists avoided studying texts/speech
- Why?
  - Hard to find
  - Time Consuming

Text and Social Science

A pre-2000's view of text in social science

- Social interaction often occurs in texts
- Social Scientists avoided studying texts/speech
- Why?
    - Hard to find
    - Time Consuming
    - Not generalizable (each new data set...new coding scheme)

Text and Social Science

A pre-2000's view of text in social science

- Social interaction often occurs in texts
- Social Scientists avoided studying texts/speech
- Why?
  - Hard to find
  - Time Consuming
  - Not generalizable (each new data set...new coding scheme)
  - Difficult to store/search

Text and Social Science

A pre-2000's view of text in social science

- Social interaction often occurs in texts
- Social Scientists avoided studying texts/speech
- Why?
    - Hard to find
    - Time Consuming
    - Not generalizable (each new data set...new coding scheme)
    - Difficult to store/search
    - Idiosyncratic to coders/researcher

Text and Social Science

A pre-2000's view of text in social science

- Social interaction often occurs in texts
- Social Scientists avoided studying texts/speech
- Why?
    - Hard to find
    - Time Consuming
    - Not generalizable (each new data set...new coding scheme)
    - Difficult to store/search
    - Idiosyncratic to coders/researcher
    - Statistical methods/algorithms, computationally intensive

A post-2000's view of text in social science:

A post-2000's view of text in social science:

Massive collections of texts are increasingly used as a data source in social science:

A post-2000's view of text in social science:

Massive collections of texts are increasingly used as a data source in social science:

- Congressional speeches, press releases, newsletters, ...

A post-2000's view of text in social science:

Massive collections of texts are increasingly used as a data source in social science:

- Congressional speeches, press releases, newsletters, ...
- Facebook posts, tweets, emails, cell phone records, ...

A post-2000's view of text in social science:

Massive collections of texts are increasingly used as a data source in social science:

- Congressional speeches, press releases, newsletters, ...
- Facebook posts, tweets, emails, cell phone records, ...
- Newspapers, magazines, news broadcasts, ...

A post-2000's view of text in social science:

Massive collections of texts are increasingly used as a data source in social science:

- Congressional speeches, press releases, newsletters, ...
- Facebook posts, tweets, emails, cell phone records, ...
- Newspapers, magazines, news broadcasts, ...
- Foreign news sources, treaties, sermons, fatwas, ...

Why?

Why?

- Massive increase in availability of unstructured text (10 minutes of worldwide email $= 1$ LOC)

Why?

- Massive increase in availability of unstructured text (10 minutes of worldwide email = 1 LOC)
- Cheap storage: 1956: $10,000 megabyte. 2014: $<<<<<$ $0.0001 per megabyte (Unless you're sending an SMS)

Why?

- Massive increase in availability of unstructured text (10 minutes of worldwide email $= 1$ LOC)
- Cheap storage: 1956: \$10,000 megabyte. 2014: $<<<<<$ \$0.0001 per megabyte (Unless you're sending an SMS)
- Explosion in methods and programs to analyze texts

Why?

- Massive increase in availability of unstructured text (10 minutes of worldwide email $= 1$ LOC)
- Cheap storage: 1956: \$10,000 megabyte. 2014: $<<<<<$ \$0.0001 per megabyte (Unless you're sending an SMS)
- Explosion in methods and programs to analyze texts
  - Generalizable: one method can be used across many methods and to unify collections of texts

Why?

- Massive increase in availability of unstructured text (10 minutes of worldwide email = 1 LOC)
- Cheap storage: 1956: \$10,000 megabyte. 2014: $<<<<<$ \$0.0001 per megabyte (Unless you're sending an SMS)
- Explosion in methods and programs to analyze texts
  - Generalizable: one method can be used across many methods and to unify collections of texts
  - Systematic: parameters/statistics demonstrate how models make coding decisions

Why?

- Massive increase in availability of unstructured text (10 minutes of worldwide email $= 1$ LOC)
- Cheap storage: 1956: \$10,000 megabyte. 2014: $<<<<<$ \$0.0001 per megabyte (Unless you're sending an SMS)
- Explosion in methods and programs to analyze texts
    - Generalizable: one method can be used across many methods and to unify collections of texts
    - Systematic: parameters/statistics demonstrate how models make coding decisions
    - Cheap: easily applied to many new collections of texts, computing power is inexpensive

Why?

- Massive increase in availability of unstructured text (10 minutes of worldwide email $= 1$ LOC)
- Cheap storage: 1956: \$10,000 megabyte. 2014: $<<<<<$ \$0.0001 per megabyte (Unless you're sending an SMS)
- Explosion in methods and programs to analyze texts
  - Generalizable: one method can be used across many methods and to unify collections of texts
  - Systematic: parameters/statistics demonstrate how models make coding decisions
  - Cheap: easily applied to many new collections of texts, computing power is inexpensive
- Unchanged Demand: Social life (politics, economic exchanges, social interactions) occurs in texts

Why?

- Massive increase in availability of unstructured text (10 minutes of worldwide email = 1 LOC)
- Cheap storage: 1956: \$10,000 megabyte. 2014: $<<<<<$ \$0.0001 per megabyte (Unless you're sending an SMS)
- Explosion in methods and programs to analyze texts
  - Generalizable: one method can be used across many methods and to unify collections of texts
  - Systematic: parameters/statistics demonstrate how models make coding decisions
  - Cheap: easily applied to many new collections of texts, computing power is inexpensive
- Unchanged Demand: Social life (politics, economic exchanges, social interactions) occurs in texts
  - Laws

Why?

- Massive increase in availability of unstructured text (10 minutes of worldwide email = 1 LOC)
- Cheap storage: 1956: $10,000 megabyte. 2014: $<<<<<$ $0.0001 per megabyte (Unless you're sending an SMS)
- Explosion in methods and programs to analyze texts
  - Generalizable: one method can be used across many methods and to unify collections of texts
  - Systematic: parameters/statistics demonstrate how models make coding decisions
  - Cheap: easily applied to many new collections of texts, computing power is inexpensive
- Unchanged Demand: Social life (politics, economic exchanges, social interactions) occurs in texts
  - Laws
  - Treaties

Why?

- Massive increase in availability of unstructured text (10 minutes of worldwide email = 1 LOC)
- Cheap storage: 1956: $10,000 megabyte. 2014: $<<<<<$ $0.0001 per megabyte (Unless you're sending an SMS)
- Explosion in methods and programs to analyze texts
  - Generalizable: one method can be used across many methods and to unify collections of texts
  - Systematic: parameters/statistics demonstrate how models make coding decisions
  - Cheap: easily applied to many new collections of texts, computing power is inexpensive
- Unchanged Demand: Social life (politics, economic exchanges, social interactions) occurs in texts
  - Laws
  - Treaties
  - News media

Why?

- Massive increase in availability of unstructured text (10 minutes of worldwide email $= 1$ LOC)
- Cheap storage: 1956: \$10,000 megabyte. 2014: $<<<<<$ \$0.0001 per megabyte (Unless you're sending an SMS)
- Explosion in methods and programs to analyze texts
  - Generalizable: one method can be used across many methods and to unify collections of texts
  - Systematic: parameters/statistics demonstrate how models make coding decisions
  - Cheap: easily applied to many new collections of texts, computing power is inexpensive
- Unchanged Demand: Social life (politics, economic exchanges, social interactions) occurs in texts
  - Laws
  - Treaties
  - News media
  - Campaigns

Why?

- Massive increase in availability of unstructured text (10 minutes of worldwide email = 1 LOC)
- Cheap storage: 1956: \$10,000 megabyte. 2014: $<<<<<$ \$0.0001 per megabyte (Unless you're sending an SMS)
- Explosion in methods and programs to analyze texts
  - Generalizable: one method can be used across many methods and to unify collections of texts
  - Systematic: parameters/statistics demonstrate how models make coding decisions
  - Cheap: easily applied to many new collections of texts, computing power is inexpensive
- Unchanged Demand: Social life (politics, economic exchanges, social interactions) occurs in texts
  - Laws
  - Treaties
  - News media
  - Campaigns
  - Political pundits

Why?

- Massive increase in availability of unstructured text (10 minutes of worldwide email = 1 LOC)
- Cheap storage: 1956: \$10,000 megabyte. 2014: $<<<<<$ \$0.0001 per megabyte (Unless you're sending an SMS)
- Explosion in methods and programs to analyze texts
  - Generalizable: one method can be used across many methods and to unify collections of texts
  - Systematic: parameters/statistics demonstrate how models make coding decisions
  - Cheap: easily applied to many new collections of texts, computing power is inexpensive
- Unchanged Demand: Social life (politics, economic exchanges, social interactions) occurs in texts
  - Laws
  - Treaties
  - News media
  - Campaigns
  - Political pundits
  - Petitions

Why?

- Massive increase in availability of unstructured text (10 minutes of worldwide email = 1 LOC)
- Cheap storage: 1956: \$10,000 megabyte. 2014: $<<<<<$ \$0.0001 per megabyte (Unless you're sending an SMS)
- Explosion in methods and programs to analyze texts
  - Generalizable: one method can be used across many methods and to unify collections of texts
  - Systematic: parameters/statistics demonstrate how models make coding decisions
  - Cheap: easily applied to many new collections of texts, computing power is inexpensive
- Unchanged Demand: Social life (politics, economic exchanges, social interactions) occurs in texts
  - Laws
  - Treaties
  - News media
  - Campaigns
  - Political pundits
  - Petitions
  - Press Releases

Why?

- Massive increase in availability of unstructured text (10 minutes of worldwide email $=$ 1 LOC)
- Cheap storage: 1956: \$10,000 megabyte. 2014: $<<<<<$ \$0.0001 per megabyte (Unless you're sending an SMS)
- Explosion in methods and programs to analyze texts
  - Generalizable: one method can be used across many methods and to unify collections of texts
  - Systematic: parameters/statistics demonstrate how models make coding decisions
  - Cheap: easily applied to many new collections of texts, computing power is inexpensive
- Unchanged Demand: Social life (politics, economic exchanges, social interactions) occurs in texts
  - Laws
  - Treaties
  - News media
  - Campaigns
  - Political pundits
  - Petitions
  - Press Releases

What Can Text Methods Do?

Haystack metaphor:

What Can Text Methods Do?

Haystack metaphor: Improve Reading

What Can Text Methods Do?

Haystack metaphor: Improve Reading

- Interpreting the meaning of a sentence or phrase $\rightsquigarrow$ Analyzing a straw of hay

What Can Text Methods Do?

Haystack metaphor: Improve Reading

- Interpreting the meaning of a sentence or phrase $\rightsquigarrow$ Analyzing a straw of hay
  - Humans: amazing (Straussian political theory, analysis of English poetry)
  - Computers: struggle

What Can Text Methods Do?

Haystack metaphor: Improve Reading

- Interpreting the meaning of a sentence or phrase $\rightsquigarrow$ Analyzing a straw of hay
  - Humans: amazing (Straussian political theory, analysis of English poetry)
  - Computers: struggle
- Comparing, Organizing, and Classifying Texts $\rightsquigarrow$ Organizing hay stack

What Can Text Methods Do?

Haystack metaphor: Improve Reading

- Interpreting the meaning of a sentence or phrase $\rightsquigarrow$ Analyzing a straw of hay
  - Humans: amazing (Straussian political theory, analysis of English poetry)
  - Computers: struggle
- Comparing, Organizing, and Classifying Texts $\rightsquigarrow$ Organizing hay stack
  - Humans: terrible. Tiny active memories
  - Computers: amazing $\rightsquigarrow$ largely what we'll discuss today

What Can Text Methods Do?

Haystack metaphor: Improve Reading

- Interpreting the meaning of a sentence or phrase $\rightsquigarrow$ Analyzing a straw of hay
    - Humans: amazing (Straussian political theory, analysis of English poetry)
    - Computers: struggle
- Comparing, Organizing, and Classifying Texts$\rightsquigarrow$ Organizing hay stack
    - Humans: terrible. Tiny active memories
    - Computers: amazing$\rightsquigarrow$ largely what we'll discuss today

What automated text methods don't do:

What Can Text Methods Do?

Haystack metaphor: Improve Reading

- Interpreting the meaning of a sentence or phrase $\rightsquigarrow$ Analyzing a straw of hay
  - Humans: amazing (Straussian political theory, analysis of English poetry)
  - Computers: struggle
- Comparing, Organizing, and Classifying Texts $\rightsquigarrow$ Organizing hay stack
  - Humans: terrible. Tiny active memories
  - Computers: amazing $\rightsquigarrow$ largely what we'll discuss today

What automated text methods don't do:

- Develop a comprehensive statistical model of language
- Replace the need to read
- Develop a single tool + evaluation for all tasks

Texts are Deceptively Complex

```
We've got some difficult days ahead.  But it doesn't matter
with me now.  Because I've been to the mountaintop.  And I
don't mind.  Like anybody, I would like to live a long
life.  Longevity has its place.  But I'm not concerned
about that now.
```

Texts are Deceptively Complex

`We've got some difficult days ahead. But it doesn't matter with me now. Because I've been to the mountaintop. And I don't mind. Like anybody, I would like to live a long life. Longevity has its place. But I'm not concerned about that now.`

- Who is the `I` ?

Texts are Deceptively Complex

`We've got some difficult days ahead. But it doesn't matter with me now. Because I've been to the mountaintop. And I don't mind. Like anybody, I would like to live a long life. Longevity has its place. But I'm not concerned about that now.`

- Who is the `I` ?
- Who is the `We`?

Texts are Deceptively Complex

`We've got some difficult days ahead. But it doesn't matter with me now. Because I've been to the mountaintop. And I don't mind. Like anybody, I would like to live a long life. Longevity has its place. But I'm not concerned about that now.`

- Who is the `I` ?
- Who is the `We`?
- What is the `mountaintop` (literal?)

Texts are Deceptively Complex

`We've got some difficult days ahead. But it doesn't matter with me now. Because I've been to the mountaintop. And I don't mind. Like anybody, I would like to live a long life. Longevity has its place. But I'm not concerned about that now.`

- Who is the `I` ?
- Who is the `We`?
- What is the `mountaintop` (literal?)

Texts$\rightsquigarrow$ high dimensional, not self contained

Texts are Surprisingly Simple

(Lamar Alexander (R-TN) Feb 10, 2005)

Word	No. Times Used in Press Release
department	12
grant	9
program	7
firefight	7
secure	5
homeland	4
fund	3
award	2
safety	2
service	2
AFGP	2
support	2
equip	2
applaud	2
assist	2

Texts are Surprisingly Simple (?)

`US Senators Bill Frist (R-TN) and Lamar Alexander (R-TN) today applauded the U S Department of Homeland Security for awarding a $8,190 grant to the Tracy City Volunteer Fire Department under the 2004 Assistance to Firefighters Grant Program's (AFGP) Fire Prevention and Safety Program...`

Not just for "big data"

Not just for "big data"

Manually develop categorization scheme for partitioning small (100) set of documents

Not just for "big data"

Manually develop categorization scheme for partitioning small (100) set of documents

- Bell(n) = number of ways of partitioning n objects

Not just for "big data"

Manually develop categorization scheme for partitioning small (100) set of documents

- Bell(n) $=$ number of ways of partitioning n objects
- Bell(2) $=$ 2 (AB, A B)

Not just for "big data"

Manually develop categorization scheme for partitioning small (100) set of documents

- Bell(n) = number of ways of partitioning n objects
- Bell(2) = 2 (AB, A B)
- Bell(3) = 5 (ABC, AB C, A BC, AC B, A B C)

Not just for "big data"

Manually develop categorization scheme for partitioning small (100) set of documents

- Bell(n) $=$ number of ways of partitioning n objects
- Bell(2) $= 2$ (AB, A B)
- Bell(3) $= 5$ (ABC, AB C, A BC, AC B, A B C)
- Bell(5) $= 52$

Not just for "big data"

Manually develop categorization scheme for partitioning small (100) set of documents

- Bell(n) $=$ number of ways of partitioning n objects
- Bell(2) $= 2$ (AB, A B)
- Bell(3) $= 5$ (ABC, AB C, A BC, AC B, A B C)
- Bell(5) $= 52$
- Bell(100)

Not just for "big data"

Manually develop categorization scheme for partitioning small (100) set of documents

- Bell(n) $=$ number of ways of partitioning n objects
- Bell(2) $= 2$ (AB, A B)
- Bell(3) $= 5$ (ABC, AB C, A BC, AC B, A B C)
- Bell(5) $= 52$
- Bell(100)$\approx 4.75 \times 10^{115}$ partitions

Not just for "big data"

Manually develop categorization scheme for partitioning small (100) set of documents

- Bell(n) = number of ways of partitioning n objects
- Bell(2) = 2 (AB, A B)
- Bell(3) = 5 (ABC, AB C, A BC, AC B, A B C)
- Bell(5) = 52
- Bell(100)$\approx 4.75 \times 10^{115}$ partitions
- Big Number:

Not just for "big data"

Manually develop categorization scheme for partitioning small (100) set of documents

- Bell(n) = number of ways of partitioning n objects
- Bell(2) = 2 (AB, A B)
- Bell(3) = 5 (ABC, AB C, A BC, AC B, A B C)
- Bell(5) = 52
- Bell(100)$\approx 4.75 \times 10^{115}$ partitions
- Big Number:
  7 Billion RAs

Not just for "big data"

Manually develop categorization scheme for partitioning small (100) set of documents

- Bell(n) $=$ number of ways of partitioning n objects
- Bell(2) $= 2$ (AB, A B)
- Bell(3) $= 5$ (ABC, AB C, A BC, AC B, A B C)
- Bell(5) $= 52$
- Bell(100)$\approx 4.75 \times 10^{115}$ partitions
- Big Number:
  7 Billion RAs
  Impossibly Fast (enumerate one clustering every millisecond)

Not just for "big data"

Manually develop categorization scheme for partitioning small (100) set of documents

- Bell(n) $=$ number of ways of partitioning n objects
- Bell(2) $= 2$ (AB, A B)
- Bell(3) $= 5$ (ABC, AB C, A BC, AC B, A B C)
- Bell(5) $= 52$
- Bell(100)$\approx 4.75 \times 10^{115}$ partitions
- Big Number:
  7 Billion RAs
  Impossibly Fast (enumerate one clustering every millisecond)
  Working around the clock (24/7/365)

Not just for "big data"

Manually develop categorization scheme for partitioning small (100) set of documents

- Bell(n) = number of ways of partitioning n objects
- Bell(2) = 2 (AB, A B)
- Bell(3) = 5 (ABC, AB C, A BC, AC B, A B C)
- Bell(5) = 52
- Bell(100)$\approx 4.75 \times 10^{115}$ partitions
- Big Number:
  7 Billion RAs
  Impossibly Fast (enumerate one clustering every millisecond)
  Working around the clock (24/7/365)
  $\approx 1.54 \times 10^{84} \times$

Not just for "big data"

Manually develop categorization scheme for partitioning small (100) set of documents

- Bell(n) = number of ways of partitioning n objects
- Bell(2) = 2 (AB, A B)
- Bell(3) = 5 (ABC, AB C, A BC, AC B, A B C)
- Bell(5) = 52
- Bell(100)$\approx 4.75 \times 10^{115}$ partitions
- Big Number:
  7 Billion RAs
  Impossibly Fast (enumerate one clustering every millisecond)
  Working around the clock (24/7/365)
  $\approx 1.54 \times 10^{84} \times (14,000,000,000)$

Not just for "big data"

Manually develop categorization scheme for partitioning small (100) set of documents

- Bell(n) = number of ways of partitioning n objects
- Bell(2) = 2 (AB, A B)
- Bell(3) = 5 (ABC, AB C, A BC, AC B, A B C)
- Bell(5) = 52
- Bell(100)$\approx 4.75 \times 10^{115}$ partitions
- Big Number:
  7 Billion RAs
  Impossibly Fast (enumerate one clustering every millisecond)
  Working around the clock (24/7/365)
  $\approx 1.54 \times 10^{84} \times (14,000,000,000)$ years

Not just for "big data"

Manually develop categorization scheme for partitioning small (100) set of documents

- Bell(n) = number of ways of partitioning n objects
- Bell(2) = 2 (AB, A B)
- Bell(3) = 5 (ABC, AB C, A BC, AC B, A B C)
- Bell(5) = 52
- Bell(100)$\approx 4.75 \times 10^{115}$ partitions
- Big Number:
  7 Billion RAs
  Impossibly Fast (enumerate one clustering every millisecond)
  Working around the clock (24/7/365)
  $\approx 1.54 \times 10^{84} \times (14,000,000,000)$ years

Automated methods can help with even small problems

Goal for this morning: Document-Term Matrices

$$
\boldsymbol{X} = \begin{pmatrix} 1 & 0 & 0 & \ldots & 3 \\ 0 & 2 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 5 \end{pmatrix}
$$

$\boldsymbol{X} = N \times J$ matrix

Goal for this morning: Document-Term Matrices

$$\boldsymbol{X} = \begin{pmatrix} 1 & 0 & 0 & \dots & 3 \\ 0 & 2 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 5 \end{pmatrix}$$

$\boldsymbol{X} = N \times J$ matrix

- N = Number of documents

Goal for this morning: Document-Term Matrices

$$\boldsymbol{X} = \begin{pmatrix} 1 & 0 & 0 & \ldots & 3 \\ 0 & 2 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 5 \end{pmatrix}$$

$\boldsymbol{X} = N \times J$ matrix

- N = Number of documents
- J = Number of features

Learning From Text

A plan for using texts

1) Acquiring text data
2) Regular expression search in text
3) Creating document-term matrices (term-document matrices)

Finding Text Data

Many places to find text

Finding Text Data

Many places to find text
Goal: plain text (.txt) file. (UTF-8, ASCII)

Finding Text Data

Many places to find text
Goal: plain text (.txt) file. (UTF-8, ASCII)
(May also want to create an `XML` or `JSON` file)

Plain Text

```
September 19, 2010 Sunday 10:46 AM EST
REP. FOXX VISITS LOCAL SCHOOLS, TALKS WITH STUDENTS ON
CONSTITUTION DAY
LENGTH: 320 words
CLEMMONS, N.C., Sept.  17 -- Rep.  Virginia Foxx, R-N.C.
(5th CD), issued the following press release:
Congresswoman Virginia Foxx is celebrating Constitution Day
today by visiting several schools in her district to talk
with students about the Constitution and the individuals
who helped create our charter document.  She will visit
Davie County High School, Forbush High School in Yadkin
County and Piney Creek School in Alleghany County.
```

XML

```
<DOC>
<DOCNO>101-levin-mi-1-19901027< /DOCNO>
<TEXT>
Mr.  LEVIN. Mr.  President, today the House passed and sent
to the President the Great Lakes Critical Programs Act.
...  Mr.  President, I commend and thank Ms.  Bean for her
exceptional efforts on the Great Lakes Critical Programs
Act
< /TEXT>
< /DOC>
```

JSON


```
{"id":"tag:search.twitter.com,2005:287886850381713411",
"objectType":"activity"...displayName":"Linda Bowersox",
"postedTime":"2010-03-10T05:16:14.000Z"...
"body":"@JeffFlake thank you for standing firm and voting
NO on the #FiscalCliff (via #PJNET)","object"...
```

Regular Expressions (from Jurafsky Slides)

Systematic Searches

A language for searching texts:

- Count mentions of a person
- Calculate amount of money discussed
- Prepare texts for analysis: Identify where to "split" a document
- ...

Provide a quick introduction here, with some examples

Regular Expressions, Some Basics (from Jurafsky Slides)

- Disjunctions

RE	**Match**	**Example Patterns Matched**
`[mM]oney`	Money or money	"Money"
`[abc]`	'a', 'b', *or* 'c'	"Investing in Iran"
		"is dangerous business"
`[1234567890]`	any digit	"sitting on $7.5 billion dollars"
		"2005 and 2006, more than "
		"$150 million dollars"
`[\.]`	A period	" 'Run!', he screamed."

Regular Expressions, Some Basics (from Jurafsky Slides)

- Ranges

RE	Match	Example Patterns Matched
[A-Z]	an upper case letter	"Rep. Anthony Weiner (D-Brooklyn & Queens)"
[a-z]	a lower case letter	"ACORN's"
[0-9]	a single digit	"(9th CD) "

Regular Expressions, Some Basics (from Jurafsky Slides)

- Negations

RE	Match	Example Patterns Matched
`[^A-Z]`	not an upper case letter	"ACORN<u>'s</u>"
`[^Ss]`	neither 'S' nor 's'	"<u>ACORN'</u>s"
`[^\.]`	not a period	" '<u>Run!', he screamed</u>."

Regular Expressions, Some Basics (from Jurafsky Slides)

- Optional Characters: `?`, `*`, `+`

RE	**Match**	**Example Patterns Matched**
`colou?r`	Words with `u` 0 or 1 times	"color" or "colour "
`oo*h!`	Words with `o` 0 or more times	"oh!" or "ooh!" or "oooh!"
`o+h!`	Words with `o` 1 or more times	"oh!" or "ooh!" or "oooooh!" or

Regular Expressions, Some Basics (from Jurafsky Slides)

- Wild Cards .

RE	Match	Example Patterns Matched
`beg.n`	Any word with "beg" then "n"	"begin" or
		"began" or
		"begun" or
		"beggn" (Poor grammar!)

Regular Expressions, Some Basics (from Jurafsky Slides)

- Start of the line anchor ^, end of the line anchor $

RE	Match	Example Patterns Match
^[A-Z]	Upper case start of line	"Palo Alto"
		"the town of Palo Alto"
^[^A-Z]	Not upper case start of line	"the town of Palo Alto"
		"Palo Alto"
^.	Start of line	"Palo Alto"
		"the town of Palo Alto"
.$	Identify character that ends a line	"Wait!"
		"This is the end."

Regular Expressions, Some Basics (from Jurafsky Slides)

- "Or" | statements, Useful short hand

RE	**Match**	**Example Patterns Matched**
yours|mine	Matches "yours" or "mine"	"it's either yours or mine"
\d	Any digit	"1-Mississippi"
\D	Any non-digit	"1-Mississippi"
\s	Any whitespace character	"1, 2"
\S	Any non-whitespace character	"1, 2"
\w	Any alpha-numeric	"1-Mississippi "
\W	Any non-alpha numeric	"1-Mississippi"

Regular Expressions, Some Basics (from Jurafsky Slides)

Quick Example to Illuminate Differences:
A “simple” example: identify all instances of `the`.

Regular Expressions, Some Basics (from Jurafsky Slides)

Quick Example to Illuminate Differences:
A “simple” example: identify all instances of `the`.

- `the`

Regular Expressions, Some Basics (from Jurafsky Slides)

Quick Example to Illuminate Differences:
A "simple" example: identify all instances of `the`.

- `the`

  Misses capitalized examples

Regular Expressions, Some Basics (from Jurafsky Slides)

Quick Example to Illuminate Differences:
A "simple" example: identify all instances of `the`.

- `the`

  Misses capitalized examples

- `[tT]he`

Regular Expressions, Some Basics (from Jurafsky Slides)

Quick Example to Illuminate Differences:
A "simple" example: identify all instances of `the`.

- `the`

  Misses capitalized examples

- `[tT]he`

  Returns words that are too long (`theocrat`, `theme`)

Regular Expressions, Some Basics (from Jurafsky Slides)

Quick Example to Illuminate Differences:
A "simple" example: identify all instances of `the`.

- `the`

  Misses capitalized examples

- `[tT]he`

  Returns words that are too long (`theocrat`, `theme`)

- [^a-zA-Z][tT]he[^a-zA-Z]

Regular Expressions, Some Basics (from Jurafsky Slides)

Quick Example to Illuminate Differences:
A "simple" example: identify all instances of `the`.

- `the`

  Misses capitalized examples

- `[tT]he`

  Returns words that are too long (`theocrat`, `theme`)

- [^a-zA-Z][tT]he[^a-zA-Z]

  Misses the first "the" in a sentence

Regular Expressions, Some Basics (from Jurafsky Slides)

Quick Example to Illuminate Differences:
A "simple" example: identify all instances of `the`.

- `the`

  Misses capitalized examples

- `[tT]he`

  Returns words that are too long (`theocrat`, `theme`)

- [^a-zA-Z][tT]he[^a-zA-Z]

  Misses the first "the" in a sentence

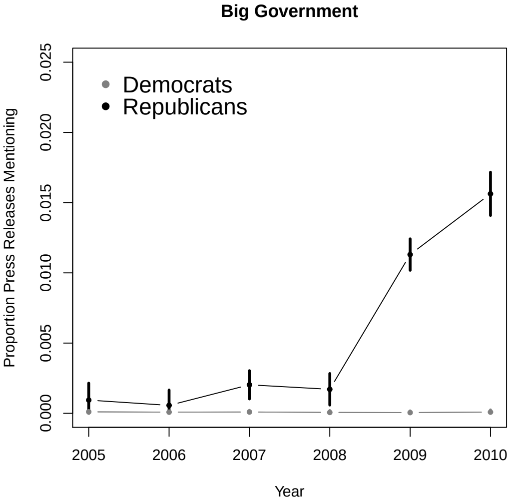
- (^ | [^ a-zA-Z])[tT]he[^ a-zA-Z]

An Example: Searching for Tea Party Language

Grimmer, Westwood, and Messing (2014): Criticism and credit

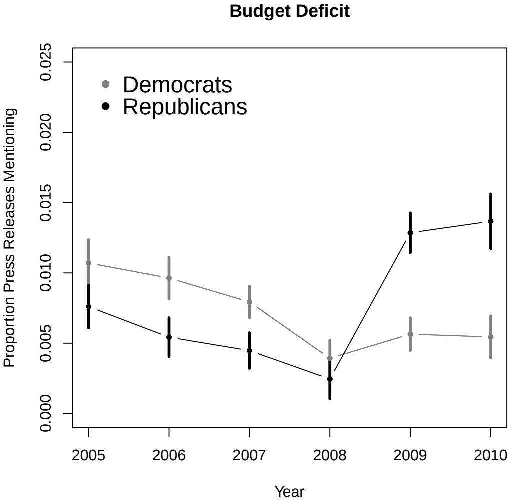
An Example: Searching for Tea Party Language

Grimmer, Westwood, and Messing (2014): Criticism and credit



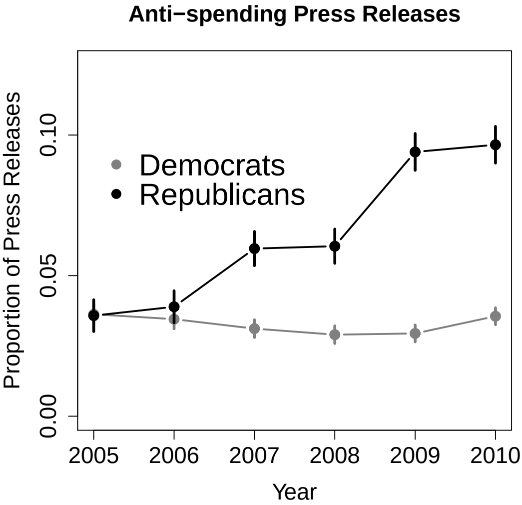
An Example: Searching for Tea Party Language

Grimmer, Westwood, and Messing (2014): Criticism and credit



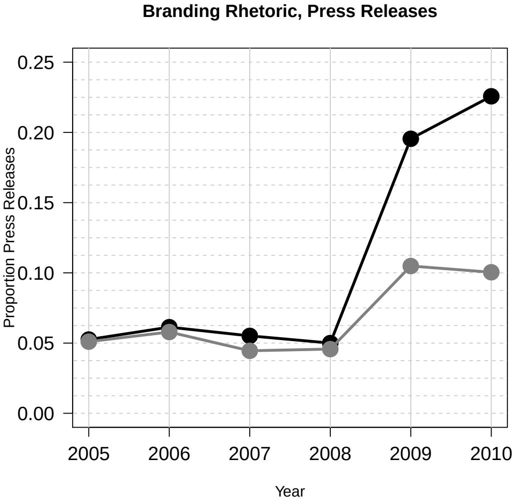
An Example: Searching for Tea Party Language

Grimmer, Westwood, and Messing (2014): Criticism and credit



An Example: Searching for Tea Party Language

Goodman, Grimmer, Parker, Zlotnik (2015): Criticism



Regular Expressions on Steroids: Cheating Detection Software

- WCopyFind:
  `http://plagiarism.bloomfieldmedia.com/z-wordpress/software/wcopyfind/`

Regular Expressions on Steroids: Cheating Detection Software

- WCopyFind:
  `http://plagiarism.bloomfieldmedia.com/z-wordpress/software/wcopyfind/`

- What constitutes plagiarism?

Regular Expressions on Steroids: Cheating Detection Software

- WCopyFind:
  `http://plagiarism.bloomfieldmedia.com/z-wordpress/software/wcopyfind/`

- What constitutes plagiarism?
- Edit distance:

Regular Expressions on Steroids: Cheating Detection Software

- WCopyFind:
  `http://plagiarism.bloomfieldmedia.com/z-wordpress/software/wcopyfind/`
- What constitutes plagiarism?
- Edit distance:
  - Heuristically: how many letters to change from *a* to *b*

Regular Expressions on Steroids: Cheating Detection Software

- WCopyFind:
  `http://plagiarism.bloomfieldmedia.com/z-wordpress/software/wcopyfind/`
- What constitutes plagiarism?
- Edit distance:
  - Heuristically: how many letters to change from *a* to *b*
- Sets many parameters:

Regular Expressions on Steroids: Cheating Detection Software

- WCopyFind:
  `http://plagiarism.bloomfieldmedia.com/z-wordpress/software/wcopyfind/`
- What constitutes plagiarism?
- Edit distance:
  - Heuristically: how many letters to change from *a* to *b*
- Sets many parameters:
  - Number of differences between pair of "strings"

Regular Expressions on Steroids: Cheating Detection Software

- WCopyFind:
  `http://plagiarism.bloomfieldmedia.com/z-wordpress/software/wcopyfind/`
- What constitutes plagiarism?
- Edit distance:
  - Heuristically: how many letters to change from *a* to *b*
- Sets many parameters:
  - Number of differences between pair of "strings"
  - Length of character strings to consider

Regular Expressions on Steroids: Cheating Detection Software

- WCopyFind:
  `http://plagiarism.bloomfieldmedia.com/z-wordpress/software/wcopyfind/`
- What constitutes plagiarism?
- Edit distance:
  - Heuristically: how many letters to change from *a* to *b*
- Sets many parameters:
  - Number of differences between pair of "strings"
  - Length of character strings to consider
  - Number of matching strings to constitute match

Regular Expressions on Steroids: Cheating Detection Software

- WCopyFind:
  `http://plagiarism.bloomfieldmedia.com/z-wordpress/software/wcopyfind/`
- What constitutes plagiarism?
- Edit distance:
  - Heuristically: how many letters to change from *a* to *b*
- Sets many parameters:
  - Number of differences between pair of "strings"
  - Length of character strings to consider
  - Number of matching strings to constitute match
- Useful:

Regular Expressions on Steroids: Cheating Detection Software

- WCopyFind:
  `http://plagiarism.bloomfieldmedia.com/z-wordpress/software/wcopyfind/`
- What constitutes plagiarism?
- Edit distance:
  - Heuristically: how many letters to change from *a* to *b*
- Sets many parameters:
  - Number of differences between pair of "strings"
  - Length of character strings to consider
  - Number of matching strings to constitute match
- Useful:
  - Media uptake

Regular Expressions on Steroids: Cheating Detection Software

- WCopyFind:
  `http://plagiarism.bloomfieldmedia.com/z-wordpress/software/wcopyfind/`
- What constitutes plagiarism?
- Edit distance:
  - Heuristically: how many letters to change from *a* to *b*
- Sets many parameters:
  - Number of differences between pair of "strings"
  - Length of character strings to consider
  - Number of matching strings to constitute match
- Useful:
  - Media uptake
  - Joint Press Releases

Document Term Matrices

Regular expressions and search are useful

Document Term Matrices

Regular expressions and search are useful
We want to use statistics/algorithms to characterize text

Document Term Matrices

Regular expressions and search are useful

We want to use statistics/algorithms to characterize text

We'll put it in a document-term matrix

Document Term Matrices

Preprocessing$\rightsquigarrow$Simplify text, make it useful

Document Term Matrices

Preprocessing$\rightsquigarrow$Simplify text, make it useful
Lower dimensionality

Document Term Matrices

Preprocessing$\rightsquigarrow$Simplify text, make it useful
Lower dimensionality

- For our purposes

Document Term Matrices

Preprocessing $\rightsquigarrow$ Simplify text, make it useful

Lower dimensionality

- For our purposes

Remember: characterize the Hay stack

Document Term Matrices

Preprocessing $\rightsquigarrow$ Simplify text, make it useful

Lower dimensionality

- For our purposes

Remember: characterize the Hay stack

- If you want to analyze a straw of hay, these methods are unlikely to work

Document Term Matrices

Preprocessing$\rightsquigarrow$Simplify text, make it useful

Lower dimensionality

- For our purposes

Remember: characterize the Hay stack

- If you want to analyze a straw of hay, these methods are unlikely to work
- But even if you want to closely read texts, characterizing hay stack can be useful

Preprocessing for Quantitative Text Analysis

One (of many) recipe for preprocessing: retain useful information

Preprocessing for Quantitative Text Analysis

One (of many) recipe for preprocessing: retain useful information

1) Remove capitalization, punctuation

Preprocessing for Quantitative Text Analysis

One (of many) recipe for preprocessing: retain useful information

1) Remove capitalization, punctuation
2) Discard Word Order (Bag of Words Assumption)

Preprocessing for Quantitative Text Analysis

One (of many) recipe for preprocessing: retain useful information

1) Remove capitalization, punctuation
2) Discard Word Order (Bag of Words Assumption)
3) Discard stop words

Preprocessing for Quantitative Text Analysis

One (of many) recipe for preprocessing: retain useful information

1) Remove capitalization, punctuation
2) Discard Word Order (Bag of Words Assumption)
3) Discard stop words
4) Create Equivalence Class: Stem, Lemmatize, or synonym

Preprocessing for Quantitative Text Analysis

One (of many) recipe for preprocessing: retain useful information

1) Remove capitalization, punctuation
2) Discard Word Order (Bag of Words Assumption)
3) Discard stop words
4) Create Equivalence Class: Stem, Lemmatize, or synonym
5) Discard less useful features $\rightsquigarrow$ depends on application

Preprocessing for Quantitative Text Analysis

One (of many) recipe for preprocessing: retain useful information

1) Remove capitalization, punctuation
2) Discard Word Order (Bag of Words Assumption)
3) Discard stop words
4) Create Equivalence Class: Stem, Lemmatize, or synonym
5) Discard less useful features $\rightsquigarrow$ depends on application
6) Other reduction, specialization

Preprocessing for Quantitative Text Analysis

One (of many) recipe for preprocessing: retain useful information

1) Remove capitalization, punctuation
2) Discard Word Order (Bag of Words Assumption)
3) Discard stop words
4) Create Equivalence Class: Stem, Lemmatize, or synonym
5) Discard less useful features $\rightsquigarrow$ depends on application
6) Other reduction, specialization

Output: Count vector, each element counts occurrence of stems

Preprocessing for Quantitative Text Analysis

One (of many) recipe for preprocessing: retain useful information

1) Remove capitalization, punctuation
2) Discard Word Order (Bag of Words Assumption)
3) Discard stop words
4) Create Equivalence Class: Stem, Lemmatize, or synonym
5) Discard less useful features $\rightsquigarrow$ depends on application
6) Other reduction, specialization

Output: Count vector, each element counts occurrence of stems

Provide tools to preprocess via this recipe

Preprocessing Texts

We're going to use the `Natural Language Toolkit (nltk)` to work with texts

- Built in functionality
- Ensures we can customize our feature spaces

Text Loaded into Python

Gettysburg Address

```
from BeautifulSoup import BeautifulSoup
from urllib import urlopen
import re, os
url =
urlopen('http://avalon.law.yale.edu/19th_century/gettyb.asp').read()
soup = BeautifulSoup(url)
text = soup.p.contents[0]
```

Preprocessing Texts

Removing capitalization:

- Python : `string.lower()`
- R : `tolower('string')`

Removing punctuation

- Python: `re.sub('\W', ' ', string)`
- R : `gsub('\\W', ' ', string)`

Preprocessing Texts

```
text_1 = text.lower()
text_2 = re.sub('\W', ' ', text_1)
```

The Bag of Words Assumption

Assumption: Discard Word Order

`Now we are engaged in a great civil war, testing whether that nation, or any nation`

The Bag of Words Assumption

Assumption: Discard Word Order

`now we are engaged in a great civil war testing whether that nation or any nation`

The Bag of Words Assumption

Assumption: Discard Word Order

Unigrams

Unigram	Count
a	1
any	1
are	1
civil	1
engaged	1
great	1
in	1
nation	2
now	1
or	1
testing	1
that	1
war	1
we	1
whether	1

The Bag of Words Assumption

Assumption: Discard Word Order

Bigrams

Bigram	Count
now we	1
we are	1
are engaged	1
engaged in	1
in a	1
a great	1
great civil	1
civil war	1
war testing	1
testing whether	1
whether that	1
that nation	1
nation or	1
or any	1
any nation	1

The Bag of Words Assumption

Assumption: Discard Word Order

Trigrams

Trigram	Count
now we are	1
we are engaged	1
are engaged in	1
engaged in a	1
in a great	1
a great civil	1
great civil war	1
civil war testing	1
war testing whether	1
whether that nation	1
that nation or	1
nation or any	1
or any nation	1

How Could This Possibly Work?

Speech is:

- Ironic

  `A real strength of the Bears is their place kicking and I'm super glad they didn't give Robbie Gould a reasonable raise`

- Subtle Negation (Source: Janyce Wiebe) :

  `They have not succeeded, and will never succeed, in breaking the will of this valiant people`

- Order Dependent (Source: Arthur Spirling):

  `Peace, no more war`

  `War, no more peace`

How Could This Possibly Work?

Three answers

1) It might not: Validation is critical (task specific)
2) Central Tendency in Text: Words often imply what a text is about `war, civil, union` or tone `consecrate, dead, died, lives`.

   Likely to be used repeatedly: create a theme for an article
3) Human supervision: Inject human judgement (coders): helps methods identify subtle relationships between words and outcomes of interest

   Dictionaries

   Training Sets

Discarding Word Order in Python

```
from nltk import word_tokenize
from nltk import bigrams
from nltk import trigrams
from nltk import ngrams

text_3 = word_tokenize(text_2)
text_3_bi = bigrams(text_3)
text_3_tri = trigrams(text_3)
text_3_n = ngrams(text_3, 4)
```

Stop Words

- Stop Words: English Language place holding words

Stop Words

- Stop Words: English Language place holding words

  `the, it, if, a, able, at, be, because...`

Stop Words

- Stop Words: English Language place holding words

  `the, it, if, a, able, at, be, because...`
- Add "noise" to documents (without conveying much information)

Stop Words

- Stop Words: English Language place holding words

  `the, it, if, a, able, at, be, because...`
- Add "noise" to documents (without conveying much information)
- Discard stop words: focus on substantive words

Stop Words

- Stop Words: English Language place holding words

  `the, it, if, a, able, at, be, because...`
- Add “noise” to documents (without conveying much information)
- Discard stop words: focus on substantive words

Note of Caution: Monroe, Colaresi, and Quinn (2008)

Stop Words

- Stop Words: English Language place holding words

  `the, it, if, a, able, at, be, because...`
- Add "noise" to documents (without conveying much information)
- Discard stop words: focus on substantive words

Note of Caution: Monroe, Colaresi, and Quinn (2008)
`she, he, her, his`

Stop Words

- Stop Words: English Language place holding words

  `the, it, if, a, able, at, be, because...`
- Add “noise” to documents (without conveying much information)
- Discard stop words: focus on substantive words

Note of Caution: Monroe, Colaresi, and Quinn (2008)

`she, he, her, his`

Many English language stop lists include gender pronouns

Stop Words

- Stop Words: English Language place holding words

  `the, it, if, a, able, at, be, because...`
- Add "noise" to documents (without conveying much information)
- Discard stop words: focus on substantive words

Note of Caution: Monroe, Colaresi, and Quinn (2008)

`she, he, her, his`

Many English language stop lists include gender pronouns

- Exercise caution when discarding stop words

Stop Words

- Stop Words: English Language place holding words

  `the, it, if, a, able, at, be, because...`
- Add "noise" to documents (without conveying much information)
- Discard stop words: focus on substantive words

Note of Caution: Monroe, Colaresi, and Quinn (2008)

`she, he, her, his`

Many English language stop lists include gender pronouns

- Exercise caution when discarding stop words
- You may need to customize your stop word list $\rightsquigarrow$ abbreviations, titles, etc

Stop Words

- Stop Words: English Language place holding words

  `the, it, if, a, able, at, be, because...`
- Add "noise" to documents (without conveying much information)
- Discard stop words: focus on substantive words

Note of Caution: Monroe, Colaresi, and Quinn (2008)
`she, he, her, his`
Many English language stop lists include gender pronouns

- Exercise caution when discarding stop words
- You may need to customize your stop word list$\rightsquigarrow$ abbreviations, titles, etc

```
stop_words =
urlopen('http://www.ai.mit.edu/projects/jmlr/papers/volume5/lewis04a/a11-smart-stop-list/english.stop').read().
n')
```

```
text_4 = [x for x in text_3 if x not in stop_words]

text_rem = [x for x in text_3 if x not in text_4]
```

Creating an Equivalence Class of Words

Reduce dimensionality further

Creating an Equivalence Class of Words

Reduce dimensionality further $\rightsquigarrow$ create equivalence class between words

Creating an Equivalence Class of Words

Reduce dimensionality further $\rightsquigarrow$ create equivalence class between words

- Words used to refer to same basic concept

Creating an Equivalence Class of Words

Reduce dimensionality further $\rightsquigarrow$ create equivalence class between words

- Words used to refer to same basic concept

  `family, families, familial` $\rightarrow$ `famili`

Creating an Equivalence Class of Words

Reduce dimensionality further $\rightsquigarrow$ create equivalence class between words

- Words used to refer to same basic concept

  `family, families, familial`$\rightarrow$ `famili`

- Stemming/Lemmatizing algorithms: Many-to-one mapping from words to stem/lemma

Comparing Stemming and Lemmatizing

Stemming algorithm:

Comparing Stemming and Lemmatizing

Stemming algorithm:

- Simplistic algorithms

Comparing Stemming and Lemmatizing

Stemming algorithm:

- Simplistic algorithms
- Chop off end of word

Comparing Stemming and Lemmatizing

Stemming algorithm:

- Simplistic algorithms
- Chop off end of word
- Porter stemmer, Lancaster stemmer, Snowball stemmer

Comparing Stemming and Lemmatizing

Stemming algorithm:

- Simplistic algorithms
- Chop off end of word
- Porter stemmer, Lancaster stemmer, Snowball stemmer

Lemmatizing algorithm:

Comparing Stemming and Lemmatizing

Stemming algorithm:

- Simplistic algorithms
- Chop off end of word
- Porter stemmer, Lancaster stemmer, Snowball stemmer

Lemmatizing algorithm:

- Condition on part of speech (noun, verb, etc)

Comparing Stemming and Lemmatizing

Stemming algorithm:

- Simplistic algorithms
- Chop off end of word
- Porter stemmer, Lancaster stemmer, Snowball stemmer

Lemmatizing algorithm:

- Condition on part of speech (noun, verb, etc)
- Verify result is a word

Comparing Stemming and Lemmatizing

Stemming algorithm:

- Simplistic algorithms
- Chop off end of word
- Porter stemmer, Lancaster stemmer, Snowball stemmer

Lemmatizing algorithm:

- Condition on part of speech (noun, verb, etc)
- Verify result is a word

Key comparison: equivalence classes

Comparing Stemming and Lemmatizing

Stemming algorithm:

- Simplistic algorithms
- Chop off end of word
- Porter stemmer, Lancaster stemmer, Snowball stemmer

Lemmatizing algorithm:

- Condition on part of speech (noun, verb, etc)
- Verify result is a word

Key comparison: equivalence classes

```
from nltk.stem.lancaster import LancasterStemmer
st = LancasterStemmer()
from nltk.stem import PorterStemmer
pt = PorterStemmer()
from nltk.stem.snowball import EnglishStemmer
sb = EnglishStemmer()
from nltk.stem.wordnet import WordNetLemmatizer
wn = WordNetLemmatizer()
```

```
>>> st.stem('better')
'bet'
>>> pt.stem('better')
'better'
>>> sb.stem('better')
'better'
>>> wn.lemmatize('better', 'a')
'good'
>>> wn.lemmatize('families', 'n')
'family'
text_5 = map(pt.stem, text_4)
```

All together now...

```
Four score and seven years ago our fathers brought forth on
this continent a new nation, conceived in liberty, and
dedicated to the proposition that all men are created
equal.
```

All together now...

```
Four score and seven years ago our fathers brought forth on
this continent a new nation, conceived in liberty, and
dedicated to the proposition that all men are created
equal.
```

Step 1: Remove capitalization and punctuation:

:

:

All together now...

`Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty, and dedicated to the proposition that all men are created equal.`

Step 1: Remove capitalization and punctuation:

`four score and seven years ago our fathers brought forth on this continent a new nation conceived in liberty and dedicated to the proposition that all men are created equal`

⋮

⋮

All together now...

Step 1: Remove capitalization and punctuation:

`four score and seven years ago our fathers brought forth on this continent a new nation conceived in liberty and dedicated to the proposition that all men are created equal`

Step 2: Discard word order:

All together now...

Step 1: Remove capitalization and punctuation:

`four score and seven years ago our fathers brought forth on this continent a new nation conceived in liberty and dedicated to the proposition that all men are created equal`

Step 2: Discard word order:

`four, score, and, seven, years, ago, our, fathers, brought, forth, on, this, continent, a, new, nation, conceived, in, liberty, and, dedicated, to, the, proposition, that, all, men, are, created, equal`

⋮

All together now...

Step 1: Remove capitalization and punctuation:
Step 2: Discard word order:
`four, score, and, seven, years, ago, our, fathers, brought, forth, on, this, continent, a, new, nation, conceived, in, liberty, and, dedicated, to, the, proposition, that, all, men, are, created, equal`
Step 3: Remove stop words :

All together now...

Step 1: Remove capitalization and punctuation:
Step 2: Discard word order:
`four, score, and, seven, years, ago, our, fathers, brought, forth, on, this, continent, a, new, nation, conceived, in, liberty, and, dedicated, to, the, proposition, that, all, men, are, created, equal`
Step 3: Remove stop words :
`four, score, seven, years, ago, fathers, brought, forth, continent, new, nation, conceived, liberty, dedicated, proposition, men, created, equal`

All together now...

Step 1: Remove capitalization and punctuation:
Step 2: Discard word order:
Step 3: Remove stop words :
`four, score, seven, years, ago, fathers, brought, forth, continent, new, nation, conceived, liberty, dedicated, proposition, men, created, equal`
Step 4: Applying Stemming Algorithm

All together now...

Step 1: Remove capitalization and punctuation:
Step 2: Discard word order:
Step 3: Remove stop words :
`four, score, seven, years, ago, fathers, brought, forth, continent, new, nation, conceived, liberty, dedicated, proposition, men, created, equal`
Step 4: Applying Stemming Algorithm
`four, score, seven, year, ago, father, brought, forth, contin, new, nation, conceiv, liberti, dedic, proposit, men, creat, equal`

All together now...

Step 1: Remove capitalization and punctuation:

Step 2: Discard word order:

Step 3: Remove stop words :

Step 4: Applying Stemming Algorithm

`four, score, seven, year, ago, father, brought, forth, contin, new, nation, conceiv, liberti, dedic, proposit, men, creat, equal`

Step 5: Create Count Vector

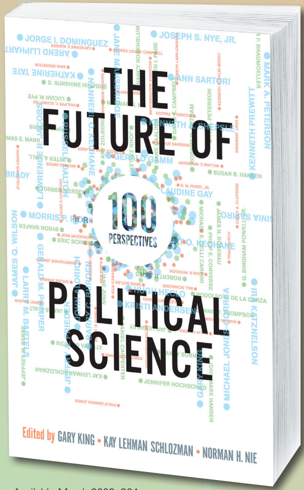
Stem	Count
ago	1
brought	1
seven	1
creat	1
conceiv	1
men	1
father	1
⋮	⋮

All together now...

Step 1: Remove capitalization and punctuation:
Step 2: Discard word order:
Step 3: Remove stop words :
Step 4: Applying Stemming Algorithm
Step 5: Create Count Vector

Stem	Count
ago	1
brought	1
seven	1
creat	1
conceiv	1
men	1
father	1
⋮	⋮

This Can Actually Work!



Available March 2009: 304pp
Pb: 978-0-415-99701-0: **$24.95**
www.routledge.com/politics

THE FUTURE OF POLITICAL SCIENCE

100 Perspectives

Edited by **Gary King**, Harvard University, **Kay Lehman Schlozman**, Boston College and **Norman H. Nie**, Stanford University

"The list of authors in *The Future of Political Science* is a 'who's who' of political science. As I was reading it, I came to think of it as a platter of tasty hors d'oeuvres. It hooked me thoroughly."

—Peter Kingstone, University of Connecticut

"In this one-of-a-kind collection, an eclectic set of contributors offer short but forceful forecasts about the future of the discipline. The resulting assortment is captivating, consistently thought-provoking, often intriguing, and sure to spur discussion and debate."

—Wendy K. Tam Cho, University of Illinois at Urbana-Champaign

"King, Schlozman, and Nie have created a visionary and stimulating volume. The organization of the essays strikes me as nothing less than brilliant. . . It is truly a joy to read."

—Lawrence C. Dodd, Manning J. Dauer Eminent Scholar in Political Science, University of Florida

Routledge
Taylor & Francis Group

Evaluators' Rate Machine Choices Better Than Their Own (Grimmer and King)

Generate pairs of similar documents: Humans vs Machines

- Scale: (1) unrelated, (2) loosely related, or (3) closely related
- Table reports: mean(scale)

Pairs from	Overall Mean	Evaluator 1	Evaluator 2

Evaluators' Rate Machine Choices Better Than Their Own (Grimmer and King)

Generate pairs of similar documents: Humans vs Machines

- Scale: (1) unrelated, (2) loosely related, or (3) closely related
- Table reports: mean(scale)

Pairs from	Overall Mean	Evaluator 1	Evaluator 2
Random Selection	1.38	1.16	1.60

Evaluators' Rate Machine Choices Better Than Their Own (Grimmer and King)

Generate pairs of similar documents: Humans vs Machines

- Scale: (1) unrelated, (2) loosely related, or (3) closely related
- Table reports: mean(scale)

Pairs from	Overall Mean	Evaluator 1	Evaluator 2
Random Selection	1.38	1.16	1.60
Hand-Coded Clusters	1.58	1.48	1.68

Evaluators' Rate Machine Choices Better Than Their Own (Grimmer and King)

Generate pairs of similar documents: Humans vs Machines

- Scale: (1) unrelated, (2) loosely related, or (3) closely related
- Table reports: mean(scale)

Pairs from	Overall Mean	Evaluator 1	Evaluator 2
Random Selection	1.38	1.16	1.60
Hand-Coded Clusters	1.58	1.48	1.68
Hand-Coding	2.06	1.88	2.24

Evaluators' Rate Machine Choices Better Than Their Own (Grimmer and King)

Generate pairs of similar documents: Humans vs Machines

- Scale: (1) unrelated, (2) loosely related, or (3) closely related
- Table reports: mean(scale)

Pairs from	Overall Mean	Evaluator 1	Evaluator 2
Random Selection	1.38	1.16	1.60
Hand-Coded Clusters	1.58	1.48	1.68
Hand-Coding	2.06	1.88	2.24
Machine	2.24	2.08	2.40

Evaluators' Rate Machine Choices Better Than Their Own (Grimmer and King)

Generate pairs of similar documents: Humans vs Machines

- Scale: (1) unrelated, (2) loosely related, or (3) closely related
- Table reports: mean(scale)

Pairs from	Overall Mean	Evaluator 1	Evaluator 2
Random Selection	1.38	1.16	1.60
Hand-Coded Clusters	1.58	1.48	1.68
Hand-Coding	2.06	1.88	2.24
Machine	2.24	2.08	2.40

p.s. The hand-coders did the evaluation!

Pre-existing word weights $\rightsquigarrow$ Dictionaries

Pre-existing word weights⇝ Dictionaries

DICTION

DICTION is a computer-aided text analysis program for Windows® and Mac® that uses a series of dictionaries to search a passage for five semantic features—Activity, Optimism, Certainty, Realism and Commonality—as well as thirty-five sub-features. DICTION uses predefined dictionaries and can use up to thirty custom dictionaries built with words that the user has defined, such as topical or negative words, for particular research needs.

Pre-existing word weights$\rightsquigarrow$ Dictionaries

DICTION

DICTION 7, now with *Power Mode,* can read a variety of text formats and can accept a large number of files within a single project. Projects containing over 1000 files are analyzed using *power analysis* for enhanced speed and reporting efficiency, with results automatically exported to .csv-formatted spreadsheet file.

Pre-existing word weights⇝ Dictionaries

DICTION

On an average computer, DICTION can process over 20,000 passages in about five minutes. DICTION requires 4.9 MB of memory and 38.4 MB of hard disk space.

Pre-existing word weights⇝ Dictionaries

DICTION

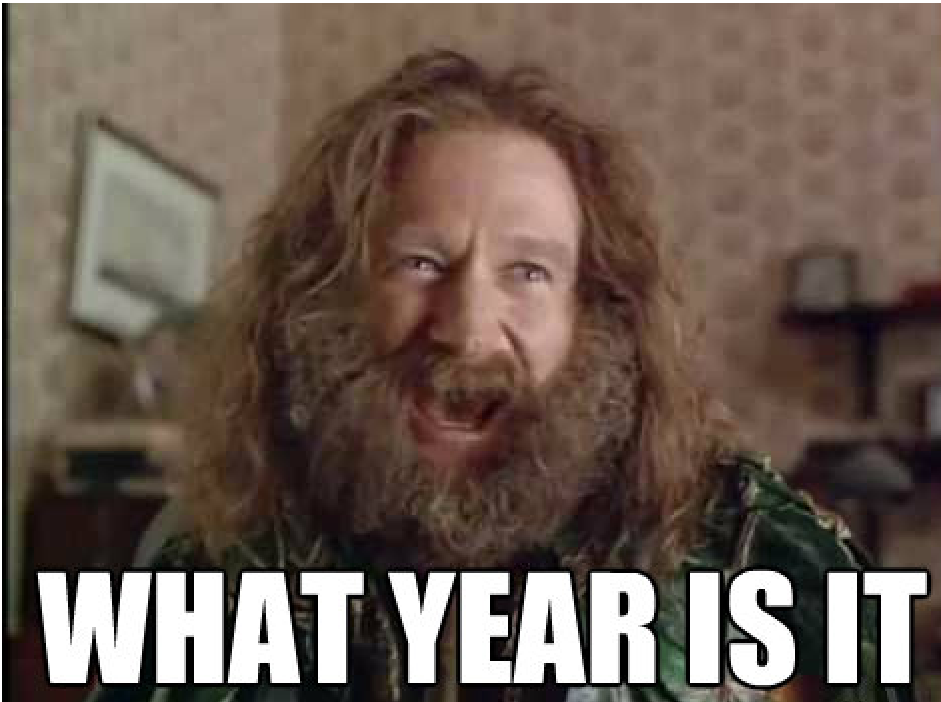
> "provides both social scientific and humanistic understandings"
>
> —Don Waisanen, Baruch College

Pre-existing word weights⇝ Dictionaries

DICTION

DICTION 7 for Mac (Educational) ($219.00)

This is the educational edition of DICTION Version 7 for Mac. You
purchase on the following page.


WHAT YEAR IS IT

Dictionary Methods

Many Dictionary Methods (like DICTION)

Dictionary Methods

Many Dictionary Methods (like DICTION)

1) Proprietary

Dictionary Methods

Many Dictionary Methods (like DICTION)

1) Proprietary$\rightsquigarrow$ wrapped in GUI

Dictionary Methods

Many Dictionary Methods (like DICTION)

1) Proprietary $\rightsquigarrow$ wrapped in GUI
2) Basic tasks:

Dictionary Methods

Many Dictionary Methods (like DICTION)

1) Proprietary$\rightsquigarrow$ wrapped in GUI
2) Basic tasks:
    a) Count words

Dictionary Methods

Many Dictionary Methods (like DICTION)

1) Proprietary $\rightsquigarrow$ wrapped in GUI
2) Basic tasks:
    a) Count words
    b) Weighted counts of words

Dictionary Methods

Many Dictionary Methods (like DICTION)

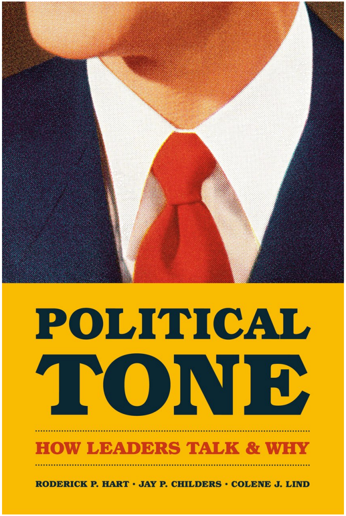
1) Proprietary$\rightsquigarrow$ wrapped in GUI
2) Basic tasks:
    a) Count words
    b) Weighted counts of words
    c) Some graphics

Dictionary Methods

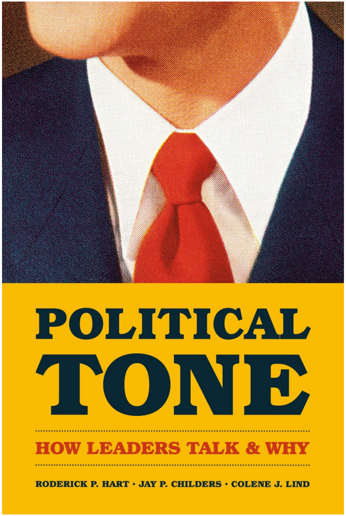
Many Dictionary Methods (like DICTION)

1) Proprietary $\rightsquigarrow$ wrapped in GUI
2) Basic tasks:
    a) Count words
    b) Weighted counts of words
    c) Some graphics
3) Pricey $\rightsquigarrow$ inexplicably

DICTION

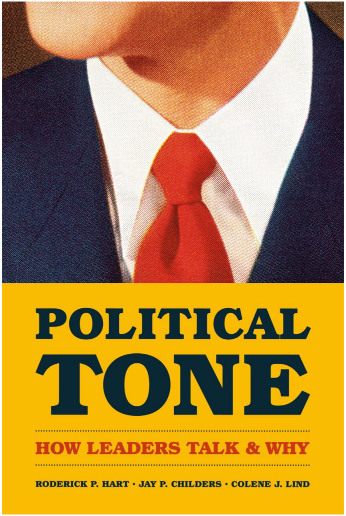

POLITICAL
TONE
HOW LEADERS TALK & WHY
RODERICK P. HART · JAY P. CHILDERS · COLENE J. LIND

DICTION



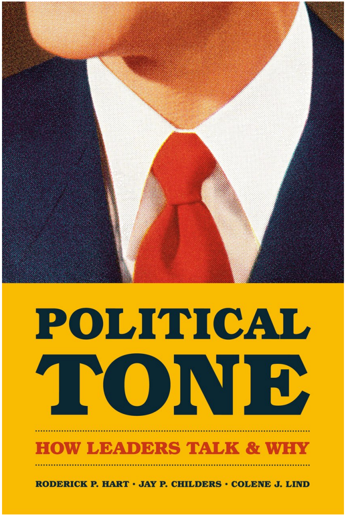
- { Certain, Uncertain }

DICTION



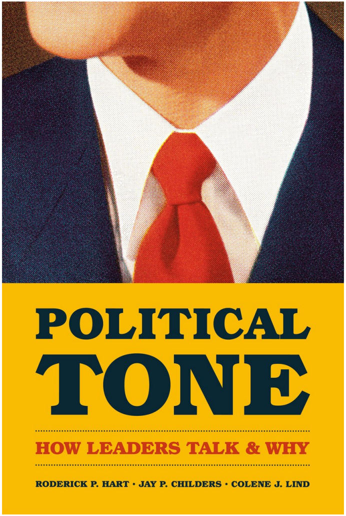
- { Certain, Uncertain }
, { Optimistic, Pessimistic }

DICTION



- { Certain, Uncertain }
  , { Optimistic, Pessimistic }
- $\approx$ 10,000 words

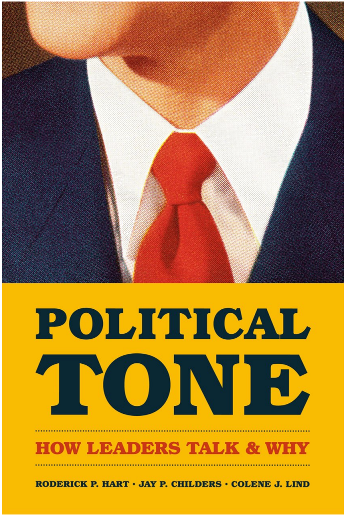
DICTION



- { Certain, Uncertain } , { Optimistic, Pessimistic }
- $\approx$ 10,000 words

Applies DICTION to a wide array of political texts

DICTION



- { Certain, Uncertain } , { Optimistic, Pessimistic }
- $\approx$ 10,000 words

Applies DICTION to a wide array of political texts
Examine specific periods of American political history

Other Dictionaries

1) General Inquirer Database
   (http://www.wjh.harvard.edu/~inquirer/)

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }
   - 3627 negative and positive word strings

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }
   - 3627 negative and positive word strings
   - Workhorse for classification across many domains/papers

Other Dictionaries

1) General Inquirer Database (http://www.wjh.harvard.edu/~inquirer/)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }
   - 3627 negative and positive word strings
   - Workhorse for classification across many domains/papers
2) Linguistic Inquiry Word Count (LIWC)

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }
   - 3627 negative and positive word strings
   - Workhorse for classification across many domains/papers
2) Linguistic Inquiry Word Count (LIWC)
   - Creation process:

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }
   - 3627 negative and positive word strings
   - Workhorse for classification across many domains/papers
2) Linguistic Inquiry Word Count (LIWC)
   - Creation process:
     1) Generate word list for categories$\rightsquigarrow$ " We drew on common emotion rating scales...Roget's Thesaurus...standard English dictionaries. [then] brain-storming sessions among 3-6 judges were held" to generate other words

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
    - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
    - { Positive, Negative }
    - 3627 negative and positive word strings
    - Workhorse for classification across many domains/papers
2) Linguistic Inquiry Word Count (LIWC)
    - Creation process:
        1) Generate word list for categories$\rightsquigarrow$ " We drew on common emotion rating scales...Roget's Thesaurus...standard English dictionaries. [then] brain-storming sessions among 3-6 judges were held" to generate other words
        2) Judge round$\rightsquigarrow$ (a) Does the word belong? (b) What other categories might it belong to?

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }
   - 3627 negative and positive word strings
   - Workhorse for classification across many domains/papers
2) Linguistic Inquiry Word Count (LIWC)
   - Creation process:
     1) Generate word list for categories$\rightsquigarrow$ " We drew on common emotion rating scales...Roget's Thesaurus...standard English dictionaries. [then] brain-storming sessions among 3-6 judges were held" to generate other words
     2) Judge round$\rightsquigarrow$ (a) Does the word belong? (b) What other categories might it belong to?
   - { Positive emotion, Negative emotion }

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }
   - 3627 negative and positive word strings
   - Workhorse for classification across many domains/papers
2) Linguistic Inquiry Word Count (LIWC)
   - Creation process:
     1) Generate word list for categories$\rightsquigarrow$ " We drew on common emotion rating scales...Roget's Thesaurus...standard English dictionaries. [then] brain-storming sessions among 3-6 judges were held" to generate other words
     2) Judge round$\rightsquigarrow$ (a) Does the word belong? (b) What other categories might it belong to?
   - { Positive emotion, Negative emotion }
   - 2300 words grouped into 70 classes

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
    - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
    - { Positive, Negative }
    - 3627 negative and positive word strings
    - Workhorse for classification across many domains/papers
2) Linguistic Inquiry Word Count (LIWC)
    - Creation process:
        1) Generate word list for categories $\rightsquigarrow$ " We drew on common emotion rating scales...Roget's Thesaurus...standard English dictionaries. [then] brain-storming sessions among 3-6 judges were held" to generate other words
        2) Judge round $\rightsquigarrow$ (a) Does the word belong? (b) What other categories might it belong to?
    - { Positive emotion, Negative emotion }
    - 2300 words grouped into 70 classes

- Harvard-IV-4

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }
   - 3627 negative and positive word strings
   - Workhorse for classification across many domains/papers
2) Linguistic Inquiry Word Count (LIWC)
   - Creation process:
     1) Generate word list for categories$\rightsquigarrow$ " We drew on common emotion rating scales...Roget's Thesaurus...standard English dictionaries. [then] brain-storming sessions among 3-6 judges were held" to generate other words
     2) Judge round$\rightsquigarrow$ (a) Does the word belong? (b) What other categories might it belong to?
   - { Positive emotion, Negative emotion }
   - 2300 words grouped into 70 classes

- Harvard-IV-4
- Affective Norms for English Words (we'll discuss this more later)

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - { Positive, Negative }
   - 3627 negative and positive word strings
   - Workhorse for classification across many domains/papers
2) Linguistic Inquiry Word Count (LIWC)
   - Creation process:
     1) Generate word list for categories $\rightsquigarrow$ " We drew on common emotion rating scales...Roget's Thesaurus...standard English dictionaries. [then] brain-storming sessions among 3-6 judges were held" to generate other words
     2) Judge round $\rightsquigarrow$ (a) Does the word belong? (b) What other categories might it belong to?
   - { Positive emotion, Negative emotion }
   - 2300 words grouped into 70 classes

- Harvard-IV-4
- Affective Norms for English Words (we'll discuss this more later)
- ...

Generating New Words

Three ways to create dictionaries (non-exhaustive):

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods (Separating methods)

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods (Separating methods)
- Manual generation

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods (Separating methods)
- Manual generation
  - Careful thought (prayer? epiphanies? divine intervention?) about useful words

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods (Separating methods)
- Manual generation
  - Careful thought (prayer? epiphanies? divine intervention?) about useful words
- Populations of people who are surprisingly willing to perform ill-defined tasks

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods (Separating methods)
- Manual generation
    - Careful thought (prayer? epiphanies? divine intervention?) about useful words
- Populations of people who are surprisingly willing to perform ill-defined tasks
    a) Undergraduates: Pizza $\rightarrow$ Research Output

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods (Separating methods)
- Manual generation
  - Careful thought (prayer? epiphanies? divine intervention?) about useful words
- Populations of people who are surprisingly willing to perform ill-defined tasks
  a) Undergraduates: Pizza $\rightarrow$ Research Output
  b) Mechanical turkers

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods (Separating methods)
- Manual generation
  - Careful thought (prayer? epiphanies? divine intervention?) about useful words
- Populations of people who are surprisingly willing to perform ill-defined tasks
  a) Undergraduates: Pizza $\rightarrow$ Research Output
  b) Mechanical turkers
     - Example: $\{$ Happy, Unhappy $\}$

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods (Separating methods)
- Manual generation
  - Careful thought (prayer? epiphanies? divine intervention?) about useful words
- Populations of people who are surprisingly willing to perform ill-defined tasks
  a) Undergraduates: Pizza $\rightarrow$ Research Output
  b) Mechanical turkers
     - Example: $\{$ Happy, Unhappy $\}$
     - Ask turkers: how happy is

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods (Separating methods)
- Manual generation
  - Careful thought (prayer? epiphanies? divine intervention?) about useful words
- Populations of people who are surprisingly willing to perform ill-defined tasks
  a) Undergraduates: Pizza $\rightarrow$ Research Output
  b) Mechanical turkers
     - Example: $\{$ Happy, Unhappy $\}$
     - Ask turkers: how happy is `elevator`, `car`, `pretty`, `young`

Generating New Words

Three ways to create dictionaries (non-exhaustive):

- Statistical methods (Separating methods)
- Manual generation
  - Careful thought (prayer? epiphanies? divine intervention?) about useful words
- Populations of people who are surprisingly willing to perform ill-defined tasks
  a) Undergraduates: Pizza $\rightarrow$ Research Output
  b) Mechanical turkers
     - Example: $\{$ Happy, Unhappy $\}$
     - Ask turkers: how happy is
       `elevator`, `car`, `pretty`, `young`
       Output as dictionary

Applying Methods to Documents

Applying the model:

Applying Methods to Documents

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK})$, $(i = 1, \ldots, N)$

Applying Methods to Documents

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK})$, $(i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$

Applying Methods to Documents

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK})$, $(i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
  - $\theta_k \in \{0, 1\}$

Applying Methods to Documents

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK})$, $(i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
    - $\theta_k \in \{0, 1\}$
    - $\theta_k \in \{-1, 0, 1\}$

Applying Methods to Documents

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK})$, $(i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
  - $\theta_k \in \{0, 1\}$
  - $\theta_k \in \{-1, 0, 1\}$
  - $\theta_k \in \{-2, -1, 0, 1, 2\}$

Applying Methods to Documents

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK})$, $(i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
    - $\theta_k \in \{0, 1\}$
    - $\theta_k \in \{-1, 0, 1\}$
    - $\theta_k \in \{-2, -1, 0, 1, 2\}$
    - $\theta_k \in \Re$

Applying Methods to Documents

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK})$, $(i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
    - $\theta_k \in \{0, 1\}$
    - $\theta_k \in \{-1, 0, 1\}$
    - $\theta_k \in \{-2, -1, 0, 1, 2\}$
    - $\theta_k \in \Re$

For each document i calculate score for document

Applying Methods to Documents

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK})$, $(i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
    - $\theta_k \in \{0, 1\}$
    - $\theta_k \in \{-1, 0, 1\}$
    - $\theta_k \in \{-2, -1, 0, 1, 2\}$
    - $\theta_k \in \Re$

For each document i calculate score for document

$$Y_i = \frac{\sum_{k=1}^{K} \theta_k X_{ik}}{\sum_{k=1}^{K} X_k}$$

Applying Methods to Documents

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK})$, $(i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
    - $\theta_k \in \{0, 1\}$
    - $\theta_k \in \{-1, 0, 1\}$
    - $\theta_k \in \{-2, -1, 0, 1, 2\}$
    - $\theta_k \in \Re$

For each document i calculate score for document

$$
\begin{aligned}
Y_i &= \frac{\sum_{k=1}^{K} \theta_k X_{ik}}{\sum_{k=1}^{K} X_k} \\
Y_i &= \frac{\boldsymbol{\theta}^{'} \boldsymbol{X}_i}{\boldsymbol{X}_i^{'} \mathbf{1}}
\end{aligned}
$$

Applying Methods to Documents

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK})$, $(i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
    - $\theta_k \in \{0, 1\}$
    - $\theta_k \in \{-1, 0, 1\}$
    - $\theta_k \in \{-2, -1, 0, 1, 2\}$
    - $\theta_k \in \Re$

For each document i calculate score for document

$$Y_i = \frac{\sum_{k=1}^{K} \theta_k X_{ik}}{\sum_{k=1}^{K} X_k}$$

$$Y_i = \frac{\boldsymbol{\theta}^{'} \boldsymbol{X}_i}{\boldsymbol{X}_i^{'} \mathbf{1}}$$

$Y_i \approx$ continuous $\rightsquigarrow$ Classification

Applying Methods to Documents

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK})$, $(i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
  - $\theta_k \in \{0, 1\}$
  - $\theta_k \in \{-1, 0, 1\}$
  - $\theta_k \in \{-2, -1, 0, 1, 2\}$
  - $\theta_k \in \Re$

For each document i calculate score for document

$$
\begin{aligned}
Y_i &= \frac{\sum_{k=1}^{K} \theta_k X_{ik}}{\sum_{k=1}^{K} X_k} \\
Y_i &= \frac{\boldsymbol{\theta}^{'} \boldsymbol{X}_i}{\boldsymbol{X}_i^{'} \mathbf{1}}
\end{aligned}
$$

$Y_i \approx$ continuous $\rightsquigarrow$ Classification

$Y_i > 0 \Rightarrow$ Positive Category

Applying Methods to Documents

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK})$, $(i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
    - $\theta_k \in \{0, 1\}$
    - $\theta_k \in \{-1, 0, 1\}$
    - $\theta_k \in \{-2, -1, 0, 1, 2\}$
    - $\theta_k \in \Re$

For each document i calculate score for document

$$
\begin{aligned}
Y_i &= \frac{\sum_{k=1}^{K} \theta_k X_{ik}}{\sum_{k=1}^{K} X_k} \\
Y_i &= \frac{\boldsymbol{\theta}^{'} \boldsymbol{X}_i}{\boldsymbol{X}_i^{'} \boldsymbol{1}}
\end{aligned}
$$

$Y_i \approx$ continuous $\rightsquigarrow$ Classification

$Y_i > 0 \Rightarrow$ Positive Category

$Y_i < 0 \Rightarrow$ Negative Category

Applying Methods to Documents

Applying the model:

- Vector of word counts: $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{iK})$, $(i = 1, \ldots, N)$
- Weights attached to words $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$
    - $\theta_k \in \{0, 1\}$
    - $\theta_k \in \{-1, 0, 1\}$
    - $\theta_k \in \{-2, -1, 0, 1, 2\}$
    - $\theta_k \in \Re$

For each document i calculate score for document

$$
\begin{aligned}
Y_i &= \frac{\sum_{k=1}^{K} \theta_k X_{ik}}{\sum_{k=1}^{K} X_k} \\
Y_i &= \frac{\boldsymbol{\theta}^{'} \boldsymbol{X}_i}{\boldsymbol{X}_i^{'} \mathbf{1}}
\end{aligned}
$$

$Y_i \approx$ continuous $\rightsquigarrow$ Classification

$Y_i > 0 \Rightarrow$ Positive Category

$Y_i < 0 \Rightarrow$ Negative Category

$Y_i \approx 0$ Ambiguous

Applying a Dictionary to Press Releases

Applying a Dictionary to Press Releases

- Collection of 169,779 press releases (US House members 2005-2010)

Applying a Dictionary to Press Releases

- Collection of 169,779 press releases (US House members 2005-2010)
- Dictionary from Neal Caren's website $\rightsquigarrow$ Theresa Wilson, Janyce Wiebe, and Paul Hoffman's dictionary

Applying a Dictionary to Press Releases

- Collection of 169,779 press releases (US House members 2005-2010)
- Dictionary from Neal Caren's website $\rightsquigarrow$ Theresa Wilson, Janyce Wiebe, and Paul Hoffman's dictionary
- Create positive/negative score for press releases.

Applying a Dictionary to Press Releases

- Collection of 169,779 press releases (US House members 2005-2010)
- Dictionary from Neal Caren's website $\rightsquigarrow$ Theresa Wilson, Janyce Wiebe, and Paul Hoffman's dictionary
- Create positive/negative score for press releases.

Examining Positive and Negative Statements in Press Releases

Examining Positive and Negative Statements in Press Releases

Least positive members of Congress:

Examining Positive and Negative Statements in Press Releases

Least positive members of Congress:

1) Dan Burton, 2008

Examining Positive and Negative Statements in Press Releases

Least positive members of Congress:

1) Dan Burton, 2008
2) Nancy Pelosi, 2007

Examining Positive and Negative Statements in Press Releases

Least positive members of Congress:

1) Dan Burton, 2008
2) Nancy Pelosi, 2007
3) Mike Pence 2007

Examining Positive and Negative Statements in Press Releases

Least positive members of Congress:

1) Dan Burton, 2008
2) Nancy Pelosi, 2007
3) Mike Pence 2007
4) John Boehner, 2009

Examining Positive and Negative Statements in Press Releases

Least positive members of Congress:

1) Dan Burton, 2008
2) Nancy Pelosi, 2007
3) Mike Pence 2007
4) John Boehner, 2009
5) Jeff Flake, (basically all years)

Examining Positive and Negative Statements in Press Releases

Least positive members of Congress:

1) Dan Burton, 2008
2) Nancy Pelosi, 2007
3) Mike Pence 2007
4) John Boehner, 2009
5) Jeff Flake, (basically all years)
6) Eric Cantor, 2009

Examining Positive and Negative Statements in Press Releases

Least positive members of Congress:

1) Dan Burton, 2008
2) Nancy Pelosi, 2007
3) Mike Pence 2007
4) John Boehner, 2009
5) Jeff Flake, (basically all years)
6) Eric Cantor, 2009
7) Tom Price, 2010

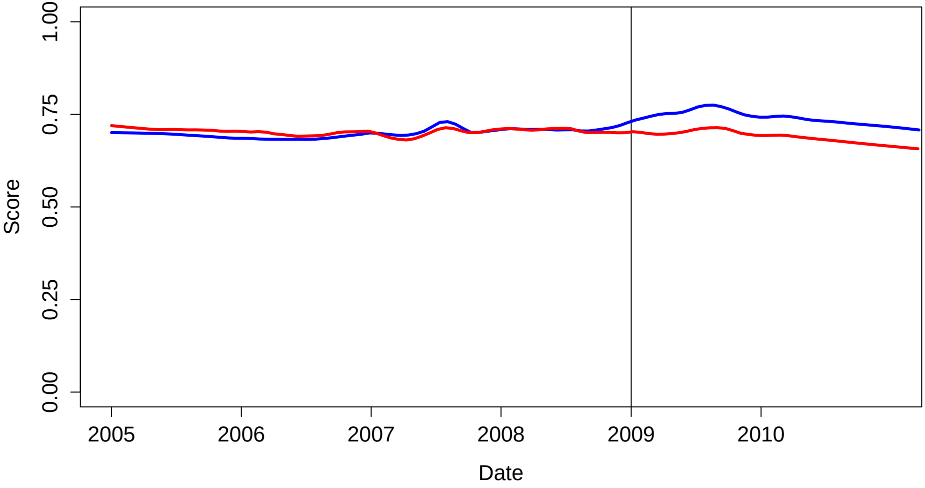
Examining Positive and Negative Statements in Press Releases

Least positive members of Congress:

1) Dan Burton, 2008
2) Nancy Pelosi, 2007
3) Mike Pence 2007
4) John Boehner, 2009
5) Jeff Flake, (basically all years)
6) Eric Cantor, 2009
7) Tom Price, 2010

Legislators who are more extreme $\rightsquigarrow$ less positive in press releases

Examining Positive and Negative Statements in Press Releases



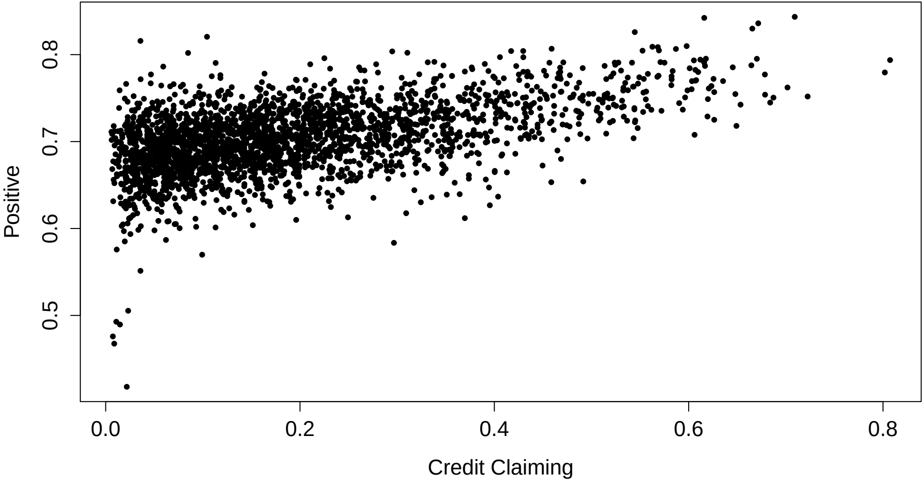
Examining Positive and Negative Statements in Press Releases

- Credit Claiming press release: 9.1 percentage points "more positive" than a non-credit claiming press release

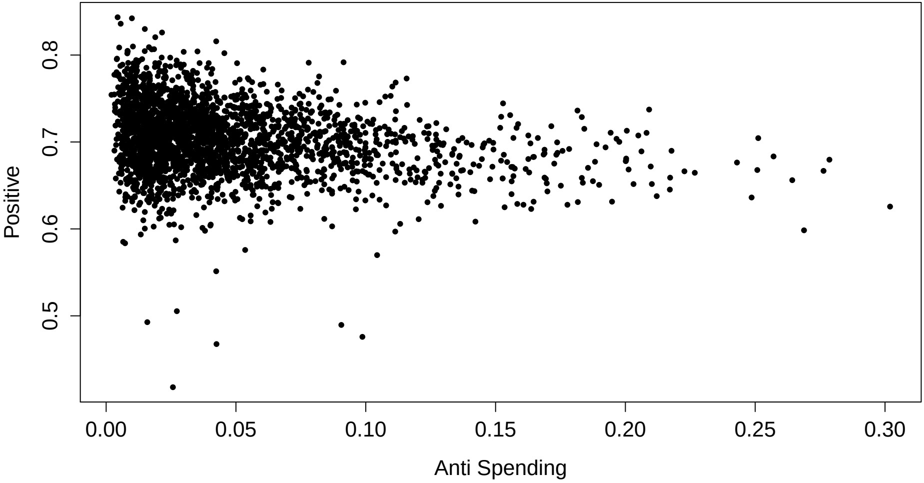
Examining Positive and Negative Statements in Press Releases

- Credit Claiming press release: 9.1 percentage points "more positive" than a non-credit claiming press release
- Anti-spending press release: 10.6 percentage points "less positive" than a non-anti spending press release

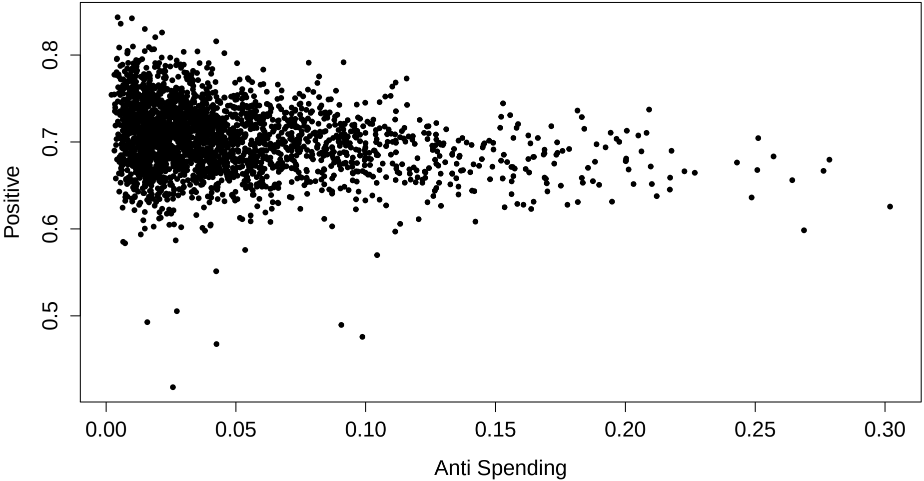
Examining Positive and Negative Statements in Press Releases



Examining Positive and Negative Statements in Press Releases



Examining Positive and Negative Statements in Press Releases



Methodological Issues/Problems with Dictionaries

Dictionary methods are context invariant

Methodological Issues/Problems with Dictionaries

Dictionary methods are context invariant

- No optimization step $\rightsquigarrow$ same word weights regardless of texts

Methodological Issues/Problems with Dictionaries

Dictionary methods are context invariant

- No optimization step $\rightsquigarrow$ same word weights regardless of texts
- Optimization$\rightsquigarrow$ incorporate information specific to context

Methodological Issues/Problems with Dictionaries

Dictionary methods are context invariant

- No optimization step $\rightsquigarrow$ same word weights regardless of texts
- Optimization$\rightsquigarrow$ incorporate information specific to context
- Without optimization$\rightsquigarrow$ unclear about dictionaries performance

Methodological Issues/Problems with Dictionaries

Dictionary methods are context invariant

- No optimization step $\rightsquigarrow$ same word weights regardless of texts
- Optimization$\rightsquigarrow$ incorporate information specific to context
- Without optimization$\rightsquigarrow$ unclear about dictionaries performance

Just because dictionaries provide measures labeled "positive" or "negative" it doesn't mean they are accurate measures in your text (!!!!)

Methodological Issues/Problems with Dictionaries

Dictionary methods are context invariant

- No optimization step $\rightsquigarrow$ same word weights regardless of texts
- Optimization$\rightsquigarrow$ incorporate information specific to context
- Without optimization$\rightsquigarrow$ unclear about dictionaries performance

Just because dictionaries provide measures labeled "positive" or "negative" it doesn't mean they are accurate measures in your text (!!!!)

Validation

Validation

Classification Validity:

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents
- Hand coded documents useful for other reasons

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents
- Hand coded documents useful for other reasons
  - Is the classification scheme well defined for your texts?

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents
- Hand coded documents useful for other reasons
  - Is the classification scheme well defined for your texts?
  - Can humans accomplish the coding task?

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents
- Hand coded documents useful for other reasons
  - Is the classification scheme well defined for your texts?
  - Can humans accomplish the coding task?
  - Is the dictionary your using appropriate?

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents
- Hand coded documents useful for other reasons
  - Is the classification scheme well defined for your texts?
  - Can humans accomplish the coding task?
  - Is the dictionary your using appropriate?

Replicate classification exercise

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents
- Hand coded documents useful for other reasons
    - Is the classification scheme well defined for your texts?
    - Can humans accomplish the coding task?
    - Is the dictionary your using appropriate?

Replicate classification exercise

- How well does our method perform on held out documents?

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents
- Hand coded documents useful for other reasons
  - Is the classification scheme well defined for your texts?
  - Can humans accomplish the coding task?
  - Is the dictionary your using appropriate?

Replicate classification exercise

- How well does our method perform on held out documents?
- Why held out?

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents
- Hand coded documents useful for other reasons
    - Is the classification scheme well defined for your texts?
    - Can humans accomplish the coding task?
    - Is the dictionary your using appropriate?

Replicate classification exercise

- How well does our method perform on held out documents?
- Why held out? Over fitting

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents
- Hand coded documents useful for other reasons
  - Is the classification scheme well defined for your texts?
  - Can humans accomplish the coding task?
  - Is the dictionary your using appropriate?

Replicate classification exercise

- How well does our method perform on held out documents?
- Why held out? Over fitting
- Using off-the-shelf dictionary: all labeled documents to test

Validation

Classification Validity:

- Training: build dictionary on subset of documents with known labels
- Test: apply dictionary method to other documents with known labels
- Requires hand coded documents
- Hand coded documents useful for other reasons
  - Is the classification scheme well defined for your texts?
  - Can humans accomplish the coding task?
  - Is the dictionary your using appropriate?

Replicate classification exercise

- How well does our method perform on held out documents?
- Why held out? Over fitting
- Using off-the-shelf dictionary: all labeled documents to test
- Supervised learning classification: (Cross)validation

Hand Coding: A Brief Digression

Humans should be able to classify documents into the categories you want the machine to classify them in

Hand Coding: A Brief Digression

Humans should be able to classify documents into the categories you want the machine to classify them in

- This is hard

Hand Coding: A Brief Digression

Humans should be able to classify documents into the categories you want the machine to classify them in

- This is hard
- Why?

Hand Coding: A Brief Digression

Humans should be able to classify documents into the categories you want the machine to classify them in

- This is hard
- Why?
  - Ambiguity in language

Hand Coding: A Brief Digression

Humans should be able to classify documents into the categories you want the machine to classify them in

- This is hard
- Why?
  - Ambiguity in language
  - Limited working memory

Hand Coding: A Brief Digression

Humans should be able to classify documents into the categories you want the machine to classify them in

- This is hard
- Why?
  - Ambiguity in language
  - Limited working memory
  - Ambiguity in classification rules

Hand Coding: A Brief Digression

Humans should be able to classify documents into the categories you want the machine to classify them in

- This is hard
- Why?
  - Ambiguity in language
  - Limited working memory
  - Ambiguity in classification rules
- A procedure for training coders:

Hand Coding: A Brief Digression

Humans should be able to classify documents into the categories you want the machine to classify them in

- This is hard
- Why?
  - Ambiguity in language
  - Limited working memory
  - Ambiguity in classification rules
- A procedure for training coders:
  1) Coding rules
  2) Apply to new texts
  3) Assess coder agreement (we'll discuss more in a few weeks)
  4) Using information and discussion, revise coding rules

Assessing Classification

Measures of classification performance

	Actual Label	
Guess	Liberal	Conservative
Liberal	True Liberal	False Liberal
Conservative	False Conservative	True Conservative

Assessing Classification

Measures of classification performance

	Actual Label	
Guess	Liberal	Conservative
Liberal	True Liberal	False Liberal
Conservative	False Conservative	True Conservative

$$\text{Accuracy} = \frac{\text{TrueLib} + \text{TrueCons}}{\text{TrueLib} + \text{TrueCons} + \text{FalseLib} + \text{FalseCons}}$$

Assessing Classification

Measures of classification performance

	Actual Label	
Guess	Liberal	Conservative
Liberal	True Liberal	False Liberal
Conservative	False Conservative	True Conservative

$$\text{Accuracy} = \frac{\text{TrueLib} + \text{TrueCons}}{\text{TrueLib} + \text{TrueCons} + \text{FalseLib} + \text{FalseCons}}$$

$$\text{Precision}_{\text{Liberal}} = \frac{\text{True Liberal}}{\text{True Liberal} + \text{False Liberal}}$$

Assessing Classification

Measures of classification performance

	Actual Label	
Guess	Liberal	Conservative
Liberal	True Liberal	False Liberal
Conservative	False Conservative	True Conservative

$$\text{Accuracy} = \frac{\text{TrueLib} + \text{TrueCons}}{\text{TrueLib} + \text{TrueCons} + \text{FalseLib} + \text{FalseCons}}$$

$$\text{Precision}_{\text{Liberal}} = \frac{\text{True Liberal}}{\text{True Liberal} + \text{False Liberal}}$$

$$\text{Recall}_{\text{Liberal}} = \frac{\text{True Liberal}}{\text{True Liberal} + \text{False Conservative}}$$

Assessing Classification

Measures of classification performance

	Actual Label	
Guess	Liberal	Conservative
Liberal	True Liberal	False Liberal
Conservative	False Conservative	True Conservative

$$\text{Accuracy} = \frac{\text{TrueLib} + \text{TrueCons}}{\text{TrueLib} + \text{TrueCons} + \text{FalseLib} + \text{FalseCons}}$$

$$\text{Precision}_{\text{Liberal}} = \frac{\text{True Liberal}}{\text{True Liberal} + \text{False Liberal}}$$

$$\text{Recall}_{\text{Liberal}} = \frac{\text{True Liberal}}{\text{True Liberal} + \text{False Conservative}}$$

$$F_{\text{Liberal}} = \frac{2\text{Precision}_{\text{Liberal}}\text{Recall}_{\text{Liberal}}}{\text{Precision}_{\text{Liberal}} + \text{Recall}_{\text{Liberal}}}$$

Assessing Classification

Measures of classification performance

	Actual Label	
Guess	Liberal	Conservative
Liberal	True Liberal	False Liberal
Conservative	False Conservative	True Conservative

$$\text{Accuracy} = \frac{\text{TrueLib} + \text{TrueCons}}{\text{TrueLib} + \text{TrueCons} + \text{FalseLib} + \text{FalseCons}}$$

$$\text{Precision}_{\text{Liberal}} = \frac{\text{True Liberal}}{\text{True Liberal} + \text{False Liberal}}$$

$$\text{Recall}_{\text{Liberal}} = \frac{\text{True Liberal}}{\text{True Liberal} + \text{False Conservative}}$$

$$F_{\text{Liberal}} = \frac{2\text{Precision}_{\text{Liberal}}\text{Recall}_{\text{Liberal}}}{\text{Precision}_{\text{Liberal}} + \text{Recall}_{\text{Liberal}}}$$

Under reported for dictionary classification

What about continuous measures?

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

- Go back to hand coding exercise

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

- Go back to hand coding exercise
- Imagine asking undergraduates to rate document on a continuous scale (0-100)

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

- Go back to hand coding exercise
- Imagine asking undergraduates to rate document on a continuous scale (0-100)
- Difficult to create classifications with agreement

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

- Go back to hand coding exercise
- Imagine asking undergraduates to rate document on a continuous scale (0-100)
- Difficult to create classifications with agreement
- Precisely the point$\rightsquigarrow$ merely creating a gold standard is hard, let alone computer classification

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

- Go back to hand coding exercise
- Imagine asking undergraduates to rate document on a continuous scale (0-100)
- Difficult to create classifications with agreement
- Precisely the point $\rightsquigarrow$ merely creating a gold standard is hard, let alone computer classification

Lower level classification

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

- Go back to hand coding exercise
- Imagine asking undergraduates to rate document on a continuous scale (0-100)
- Difficult to create classifications with agreement
- Precisely the point $\rightsquigarrow$ merely creating a gold standard is hard, let alone computer classification

Lower level classification $\rightsquigarrow$ label phrases and then aggregate

$\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

- Go back to hand coding exercise
- Imagine asking undergraduates to rate document on a continuous scale (0-100)
- Difficult to create classifications with agreement
- Precisely the point $\rightsquigarrow$ merely creating a gold standard is hard, let alone computer classification

Lower level classification $\rightsquigarrow$ label phrases and then aggregate
Modifiable areal unit problem in texts $\rightsquigarrow$

What about continuous measures?

Necessarily more complicated

- Go back to hand coding exercise
- Imagine asking undergraduates to rate document on a continuous scale (0-100)
- Difficult to create classifications with agreement
- Precisely the point $\rightsquigarrow$ merely creating a gold standard is hard, let alone computer classification

Lower level classification $\rightsquigarrow$ label phrases and then aggregate
Modifiable areal unit problem in texts $\rightsquigarrow$ aggregating destroys information, conclusion may depend on level of aggregation

Validation, Dictionaries from other Fields

Validation, Dictionaries from other Fields

Accounting Research: measure tone of 10-K reports

Validation, Dictionaries from other Fields

Accounting Research: measure tone of 10-K reports

- tone matters ($)

Validation, Dictionaries from other Fields

Accounting Research: measure tone of 10-K reports

- tone matters ($)

Previous state of art: Harvard-IV-4 Dictionary applied to texts

Validation, Dictionaries from other Fields

Accounting Research: measure tone of 10-K reports

- tone matters ($)

Previous state of art: Harvard-IV-4 Dictionary applied to texts

Loughran and McDonald (2011): Financial Documents are Different, polysemes

Validation, Dictionaries from other Fields

Accounting Research: measure tone of 10-K reports

- tone matters ($)

Previous state of art: Harvard-IV-4 Dictionary applied to texts

Loughran and McDonald (2011): Financial Documents are Different, polysemes

- Negative words in Harvard, Not Negative in Accounting:

Validation, Dictionaries from other Fields

Accounting Research: measure tone of 10-K reports

- tone matters ($)

Previous state of art: Harvard-IV-4 Dictionary applied to texts

Loughran and McDonald (2011): Financial Documents are Different, polysemes

- Negative words in Harvard, Not Negative in Accounting: `tax,cost,capital,board,liability,foreign, cancer, crude(oil),tire`

Validation, Dictionaries from other Fields

Accounting Research: measure tone of 10-K reports

- tone matters ($)

Previous state of art: Harvard-IV-4 Dictionary applied to texts

Loughran and McDonald (2011): Financial Documents are Different, polysemes

- Negative words in Harvard, Not Negative in Accounting: `tax,cost,capital,board,liability,foreign, cancer, crude(oil),tire`
- 73% of Harvard negative words in this set(!!!!!)

Validation, Dictionaries from other Fields

Accounting Research: measure tone of 10-K reports

- tone matters ($)

Previous state of art: Harvard-IV-4 Dictionary applied to texts
Loughran and McDonald (2011): Financial Documents are Different, polysemes

- Negative words in Harvard, Not Negative in Accounting: `tax,cost,capital,board,liability,foreign, cancer, crude(oil),tire`
- 73% of Harvard negative words in this set(!!!!!)
- Not Negative Harvard, Negative in Accounting:

Validation, Dictionaries from other Fields

Accounting Research: measure tone of 10-K reports

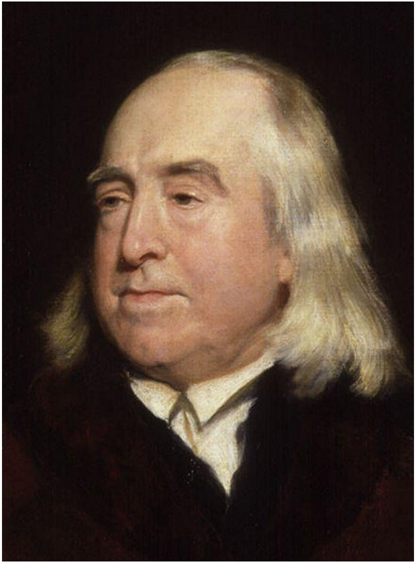
- tone matters ($)

Previous state of art: Harvard-IV-4 Dictionary applied to texts

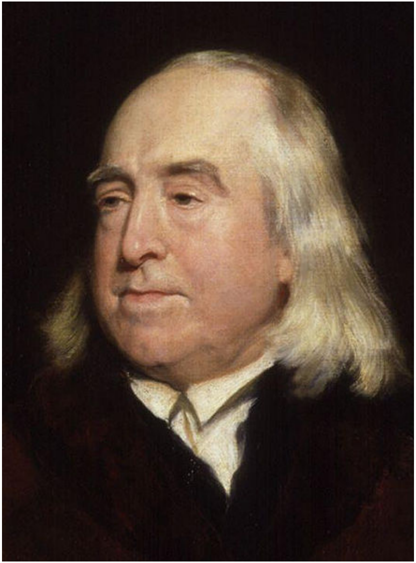
Loughran and McDonald (2011): Financial Documents are Different, polysemes

- Negative words in Harvard, Not Negative in Accounting: `tax,cost,capital,board,liability,foreign, cancer, crude(oil),tire`
- 73% of Harvard negative words in this set(!!!!!)
- Not Negative Harvard, Negative in Accounting: `felony,litigation,restated,misstatement, unanticipated`

Measuring Happiness

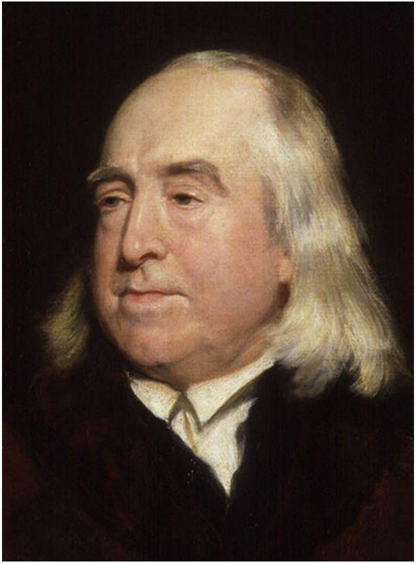


Measuring Happiness



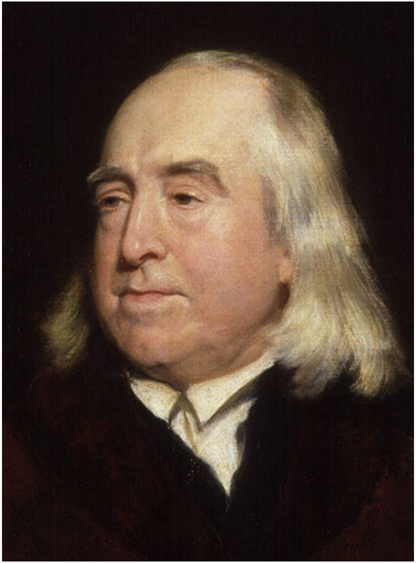
- Quantifying Happiness: How happy is society?

Measuring Happiness



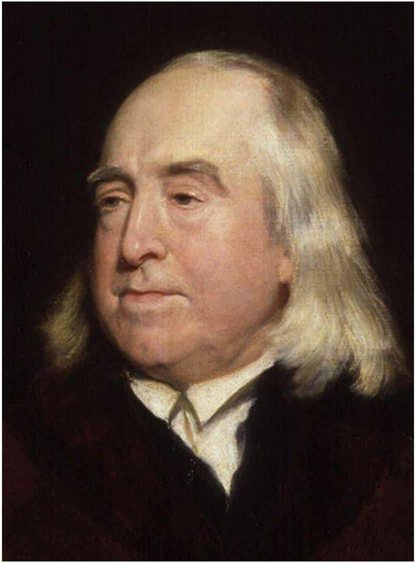
- Quantifying Happiness: How happy is society?
- How Happy is a Song?

Measuring Happiness



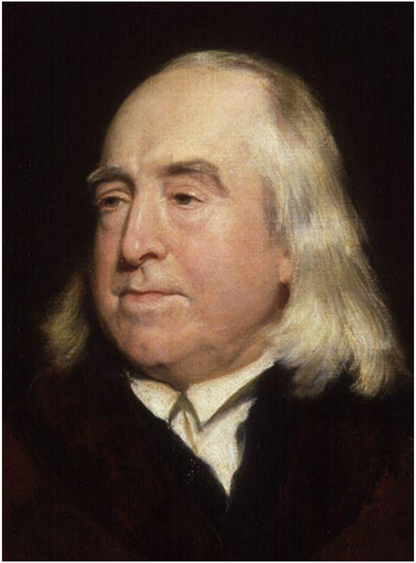
- Quantifying Happiness: How happy is society?
- How Happy is a Song?
- Blog posts?

Measuring Happiness



- Quantifying Happiness: How happy is society?
- How Happy is a Song?
- Blog posts?
- Facebook posts? (Gross National Happiness)

Measuring Happiness



- Quantifying Happiness: How happy is society?
- How Happy is a Song?
- Blog posts?
- Facebook posts? (Gross National Happiness)

Use Dictionary Methods

Measuring Happiness

Dodds and Danforth (2009): Use a dictionary method to measure happiness

Measuring Happiness

Dodds and Danforth (2009): Use a dictionary method to measure happiness

- Affective Norms for English Words (ANEW)

Measuring Happiness

Dodds and Danforth (2009): Use a dictionary method to measure happiness

- Affective Norms for English Words (ANEW)
- Bradley and Lang 1999: 1034 words, Affective reaction to words

Measuring Happiness

Dodds and Danforth (2009): Use a dictionary method to measure happiness

- Affective Norms for English Words (ANEW)
- Bradley and Lang 1999: 1034 words, Affective reaction to words
    - On a scale of 1-9 how happy does this word make you?

Measuring Happiness

Dodds and Danforth (2009): Use a dictionary method to measure happiness

- Affective Norms for English Words (ANEW)
- Bradley and Lang 1999: 1034 words, Affective reaction to words
    - On a scale of 1-9 how happy does this word make you?
      Happy : triumphant (8.82)/paradise (8.72)/ love (8.72)

Measuring Happiness

Dodds and Danforth (2009): Use a dictionary method to measure happiness

- Affective Norms for English Words (ANEW)
- Bradley and Lang 1999: 1034 words, Affective reaction to words
    - On a scale of 1-9 how happy does this word make you?
      Happy : triumphant (8.82)/paradise (8.72)/ love (8.72)
      Neutral: street (5.22)/ paper (5.20)/ engine (5.20)

Measuring Happiness

Dodds and Danforth (2009): Use a dictionary method to measure happiness

- Affective Norms for English Words (ANEW)
- Bradley and Lang 1999: 1034 words, Affective reaction to words
  - On a scale of 1-9 how happy does this word make you?
    Happy : triumphant (8.82)/paradise (8.72)/ love (8.72)
    Neutral: street (5.22)/ paper (5.20)/ engine (5.20)
    Unhappy : cancer (1.5)/funeral (1.39)/ rape (1.25) /suicide (1.25)

Measuring Happiness

Dodds and Danforth (2009): Use a dictionary method to measure happiness

- Affective Norms for English Words (ANEW)
- Bradley and Lang 1999: 1034 words, Affective reaction to words
  - On a scale of 1-9 how happy does this word make you?
    Happy : triumphant (8.82)/paradise (8.72)/ love (8.72)
    Neutral: street (5.22)/ paper (5.20)/ engine (5.20)
    Unhappy : cancer (1.5)/funeral (1.39)/ rape (1.25) /suicide (1.25)
- Happiness for text i (with word j having happiness θ_j and document frequence X_{ij})

Measuring Happiness

Dodds and Danforth (2009): Use a dictionary method to measure happiness

- Affective Norms for English Words (ANEW)
- Bradley and Lang 1999: 1034 words, Affective reaction to words
  - On a scale of 1-9 how happy does this word make you?
    Happy : triumphant (8.82)/paradise (8.72)/ love (8.72)
    Neutral: street (5.22)/ paper (5.20)/ engine (5.20)
    Unhappy : cancer (1.5)/funeral (1.39)/ rape (1.25) /suicide (1.25)
- Happiness for text i (with word j having happiness θ_j and document frequence X_{ij})

$$\text{Happiness}_i = \frac{\sum_{k=1}^{K} \theta_k X_{ik}}{\sum_{k=1}^{K} X_{ik}}$$

Lyrics for
Michael Jackson's Billie Jean

"She was more like a beauty queen
from a movie scene.
⋮
And mother always told me,
be careful who you love.
And be careful of what you do
'cause the lie becomes the truth.

Billie Jean is not my lover,
She's just a girl who claims
that I am the one.
⋮

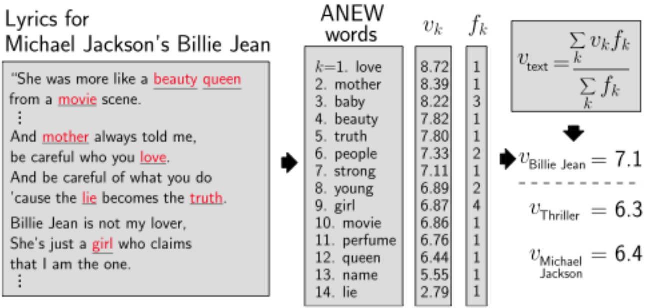
ANEW words	v_k	f_k
k=1. love	8.72	1
2. mother	8.39	1
3. baby	8.22	3
4. beauty	7.82	1
5. truth	7.80	1
6. people	7.33	2
7. strong	7.11	1
8. young	6.89	2
9. girl	6.87	4
10. movie	6.86	1
11. perfume	6.76	1
12. queen	6.44	1
13. name	5.55	1
14. lie	2.79	1

$$v_{\text{text}} = \frac{\sum_k v_k f_k}{\sum_k f_k}$$

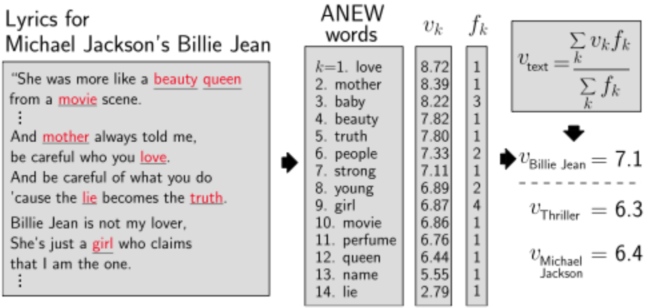
$$v_{\text{Billie Jean}} = 7.1$$

$$v_{\text{Thriller}} = 6.3$$

$$v_{\text{Michael Jackson}} = 6.4$$

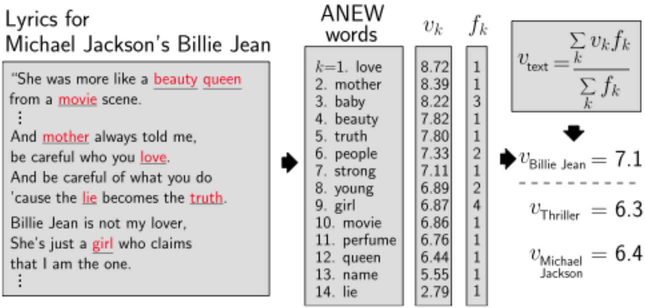


Homework Hints: One approach: write a `for` loop searching for words in dictionary (caution: is dictionary stemmed?)



Homework Hints: One approach: write a `for` loop searching for words in dictionary (caution: is dictionary stemmed?)

Happiest Song on Thriller?

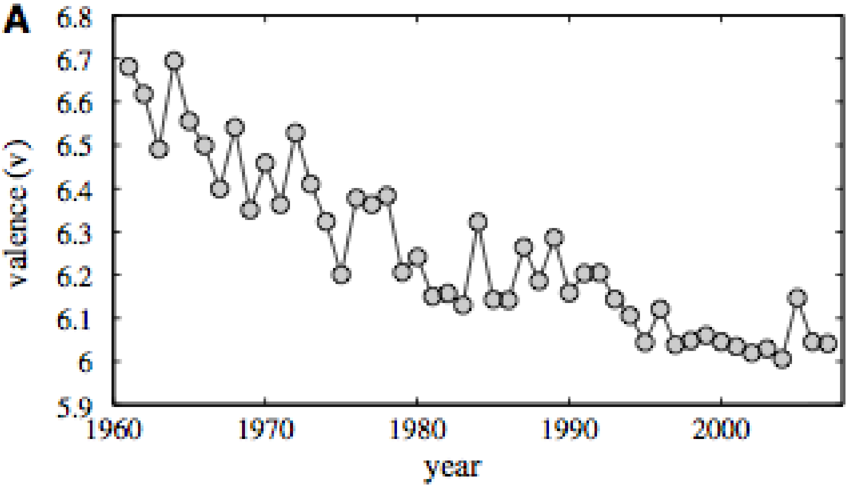


Homework Hints: One approach: write a `for` loop searching for words in dictionary (caution: is dictionary stemmed?)

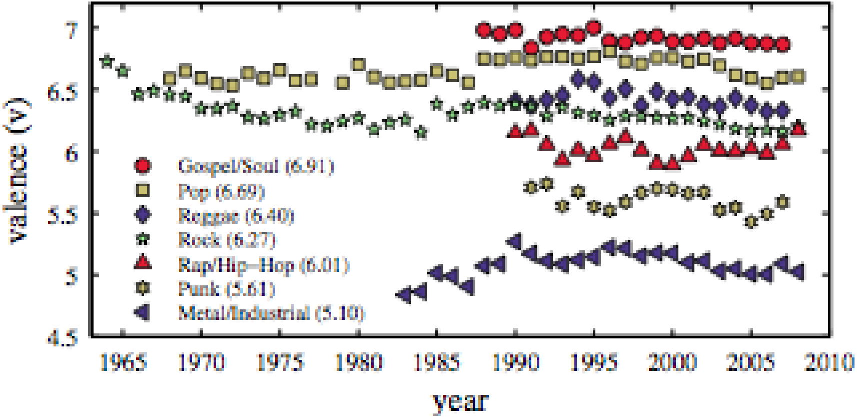
Happiest Song on Thriller?

P.Y.T. (Pretty Young Thing) (This is the right answer!)

Happiness in Society



Happiness in Society



Happiness in Society