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

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

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

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

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

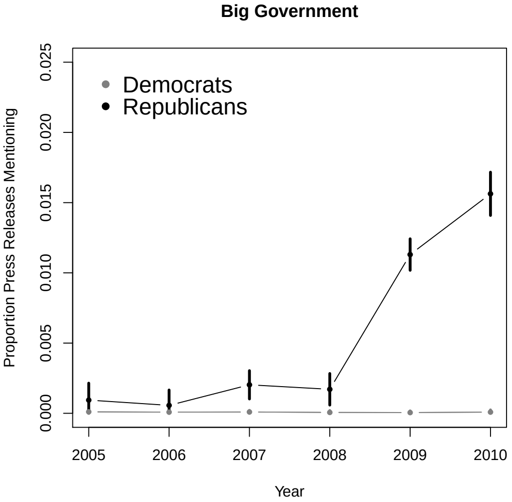
- (^ | [^ a-zA-Z])[tT]he[^ a-zA-Z]

An Example: Searching for Tea Party Language

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

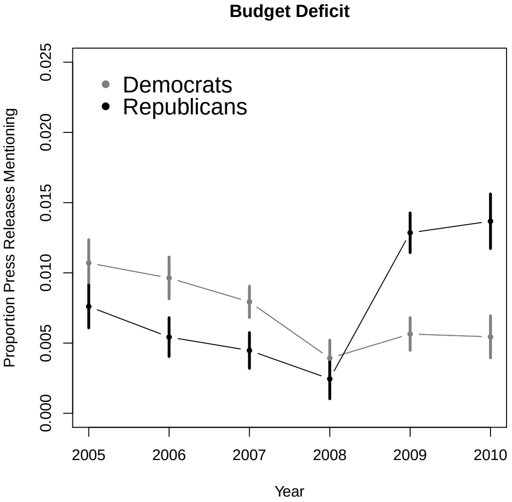
An Example: Searching for Tea Party Language

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



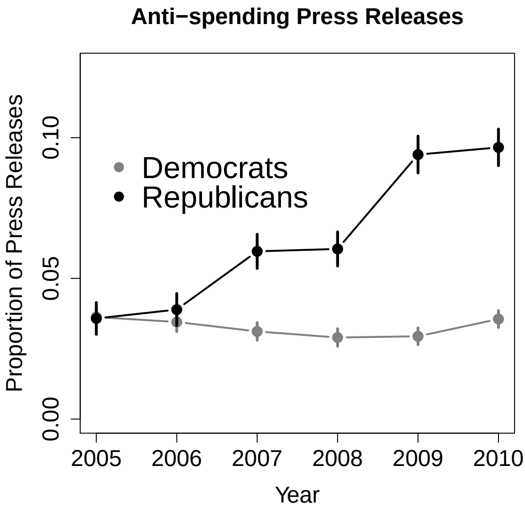
An Example: Searching for Tea Party Language

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



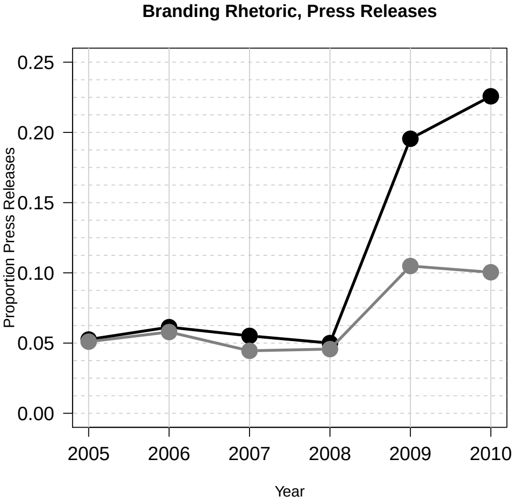
An Example: Searching for Tea Party Language

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 (`Python Code!`)

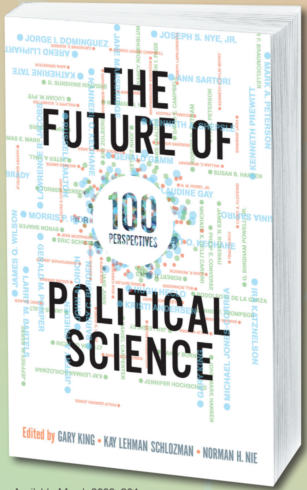
Stem	Count
ago	1
brought	1
seven	1
creat	1
conceiv	1
men	1
father	1
⋮	⋮

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 (`Python Code!`)

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!