

# Text as Data

Justin Grimmer

Professor  
Department of Political Science  
Stanford University

May 23rd, 2019

# Text and Political Science

- A pre-2000's view of text in social science
- Social interaction often occurs in texts

# Text and Political 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 Political 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 Political 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 Political 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 Political 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 Political 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 Political 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 Political 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

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

# Not just for “big data”

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

- $\text{Bell}(n)$  = number of ways of partitioning  $n$  objects
- $\text{Bell}(2) = 2$  (AB, A B)
- $\text{Bell}(3) = 5$  (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

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

# Not just for “big data”

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

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

# Not just for “big data”

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

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

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

# Plan for the Course

- 1) 5/23, Morning: Acquiring, preprocessing, and comparing text
- 2) 5/23, Afternoon: **Discovery**: Vector Space Model of Text, Clustering Methods, Topic Models, Separating Words
- 3) 5/24, Morning: **Measurement**: Dictionary Methods, Hand Coding, Supervised Methods Part 1
- 4) 5/24, Afternoon: **Causal Inference**: Train/Test Split, Analyst Induced SUTVA, Text as Dependent and Independent Variable



# Four Principles of Automated Text Analysis

Principle 1: All Quantitative Models of Language are Wrong—But Some are Useful

# Four Principles of Automated Text Analysis

Principle 1: All Quantitative Models of Language are Wrong—But Some are Useful

- Data generation process for text  $\rightsquigarrow$  unknown

# Four Principles of Automated Text Analysis

Principle 1: All Quantitative Models of Language are Wrong—But Some are Useful

- Data generation process for text  $\rightsquigarrow$  unknown
- Complexity of language:

# Four Principles of Automated Text Analysis

Principle 1: All Quantitative Models of Language are Wrong—But Some are Useful

- Data generation process for text  $\rightsquigarrow$  unknown
- Complexity of language:
  - Time flies like an arrow

# Four Principles of Automated Text Analysis

Principle 1: All Quantitative Models of Language are Wrong—But Some are Useful

- Data generation process for text  $\rightsquigarrow$  unknown
- Complexity of language:
  - Time flies like an arrow, fruit flies like a banana

# Four Principles of Automated Text Analysis

## Principle 1: All Quantitative Models of Language are Wrong—But Some are Useful

- Data generation process for text  $\rightsquigarrow$  unknown
- Complexity of language:
  - Time flies like an arrow, fruit flies like a banana
  - Make peace, not war

# Four Principles of Automated Text Analysis

## Principle 1: All Quantitative Models of Language are Wrong—But Some are Useful

- Data generation process for text  $\rightsquigarrow$  unknown
- Complexity of language:
  - Time flies like an arrow, fruit flies like a banana
  - Make peace, not war , Make war not peace (Spirling, 2013)

# Four Principles of Automated Text Analysis

## Principle 1: All Quantitative Models of Language are Wrong—But Some are Useful

- Data generation process for text  $\rightsquigarrow$  unknown
- Complexity of language:
  - Time flies like an arrow, fruit flies like a banana
  - Make peace, not war , Make war not peace (Spirling, 2013)
  - “Years from now, you’ll look back and you’ll say that this was the moment, this was the place where America remembered what it means to hope. ”



# Four Principles of Automated Text Analysis

## Principle 1: All Quantitative Models of Language are Wrong—But Some are Useful

- Data generation process for text  $\rightsquigarrow$  unknown
- Complexity of language:
  - Time flies like an arrow, fruit flies like a banana
  - Make peace, not war , Make war not peace (Spirling, 2013)
  - “Years from now, you’ll look back and you’ll say that this was the moment, this was the place where America remembered what it means to hope. ”
- Models **necessarily** fail to capture language  $\rightsquigarrow$  useful for specific tasks

# Four Principles of Automated Text Analysis

## Principle 1: All Quantitative Models of Language are Wrong—But Some are Useful

- Data generation process for text  $\rightsquigarrow$  unknown
- Complexity of language:
  - Time flies like an arrow, fruit flies like a banana
  - Make peace, not war , Make war not peace (Spirling, 2013)
  - “Years from now, you’ll look back and you’ll say that this was the moment, this was the place where America remembered what it means to hope. ”
- Models **necessarily** fail to capture language  $\rightsquigarrow$  useful for specific tasks
- **Validation**  $\rightsquigarrow$  demonstrate methods perform task

# Four Principles of Automated Text Analysis

Principle 2: Quantitative Methods Augment Humans, Not Replace Them

# Four Principles of Automated Text Analysis

Principle 2: Quantitative Methods Augment Humans, Not Replace Them

- Computer-Assisted Reading

# Four Principles of Automated Text Analysis

## Principle 2: Quantitative Methods Augment Humans, Not Replace Them

- Computer-Assisted Reading
- Quantitative methods organize, direct, and suggest

# Four Principles of Automated Text Analysis

## Principle 2: Quantitative Methods Augment Humans, Not Replace Them

- **Computer-Assisted** Reading
- Quantitative methods organize, direct, and suggest
- Humans: read and interpret

# Four Principles of Automated Text Analysis

Principle 3: There is no Globally Best Method for Automated Text Analysis

# Four Principles of Automated Text Analysis

Principle 3: There is no Globally Best Method for Automated Text Analysis

- Supervised methods  $\rightsquigarrow$  known categories



# Four Principles of Automated Text Analysis

## Principle 3: There is no Globally Best Method for Automated Text Analysis

- Supervised methods  $\rightsquigarrow$  known categories
- Unsupervised methods  $\rightsquigarrow$  discover categories

# Four Principles of Automated Text Analysis

## Principle 3: There is no Globally Best Method for Automated Text Analysis

- Supervised methods  $\rightsquigarrow$  known categories
- Unsupervised methods  $\rightsquigarrow$  discover categories
- Debate  $\rightsquigarrow$  acknowledge differences, resolved

# Four Principles of Automated Text Analysis

## Principle 4: Validate, Validate, Validate

# Four Principles of Automated Text Analysis

## Principle 4: Validate, Validate, Validate

- Quantitative methods  $\rightsquigarrow$  variable performance across tasks

# Four Principles of Automated Text Analysis

## Principle 4: Validate, Validate, Validate

- Quantitative methods  $\rightsquigarrow$  variable performance across tasks
- Few theorems to guarantee performance

# Four Principles of Automated Text Analysis

## Principle 4: Validate, Validate, Validate

- Quantitative methods  $\rightsquigarrow$  variable performance across tasks
- Few theorems to guarantee performance
- Apply methods  $\rightsquigarrow$  validate

# Four Principles of Automated Text Analysis

## Principle 4: Validate, Validate, Validate

- Quantitative methods  $\rightsquigarrow$  variable performance across tasks
- Few theorems to guarantee performance
- Apply methods  $\rightsquigarrow$  validate
- Avoid: blind application of methods

# Goal for Today: Document-Term Matrices

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

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



# Goal for Today: Document-Term Matrices

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

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

- $N$  = Number of documents

# Goal for Today: Document-Term Matrices

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

$\mathbf{X} = N \times J$  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"...
```

# Prepackaged Data Sources

<http://dfr.jstor.org>

[illegible]

Search

Search:

**Selected: 9,290,921**

Sort By: Most Recent to Oldest

« Prev

1

2

3

... 929.093

[Next](#)

### **Narrow results by:**

#### **+ Year of Publication**

#### **Content Type**

## Article Type

### Key terms

**+** Journal

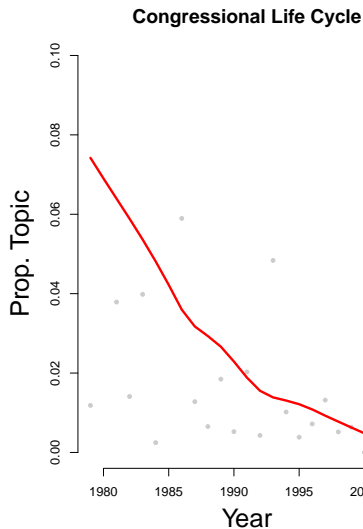
+ Publisher

#### **✚ Author**

## Language

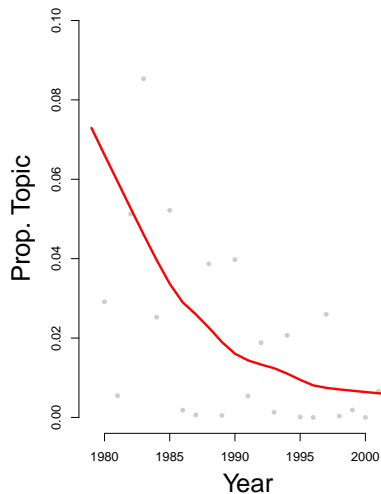
1. **Evaluating International Research Ethics Capacity Development: An En**  
Joseph Ali, Nancy E. Kass, Nelson K. Sewankambo, Tara D. White, Adnan A  
*Journal of Empirical Research on Human Research Ethics: An International J*  
More Info
2. **Age of Fracture, by Daniel T. Rodgers**  
Michael J. Kramer  
*American Political Thought*, Vol. 3, No. 1 (Spring 2014), pp. 193-196  
More Info
3. **The Failure of Popular Sovereignty: Slavery, Manifest Destiny, and the I**  
Martin H. Quitt  
*American Political Thought*, Vol. 3, No. 1 (Spring 2014), pp. 190-193  
More Info

# History of Home Style



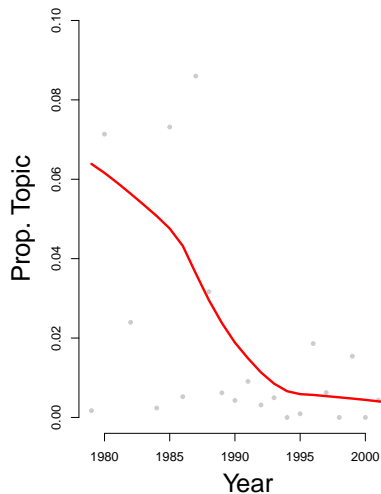
# History of Home Style

Comparative Study of Home Style

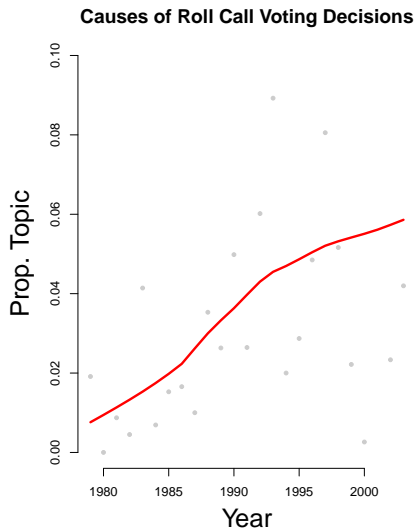


# History of Home Style

**Casework and the Incumbency Advantage**

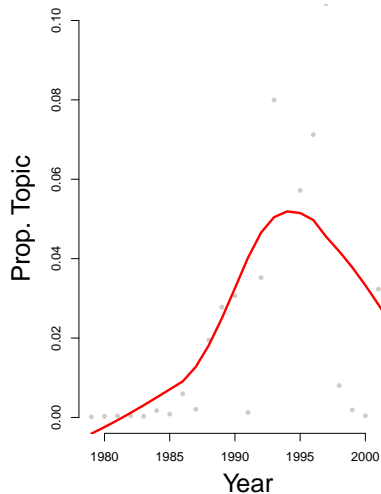


# History of Home Style



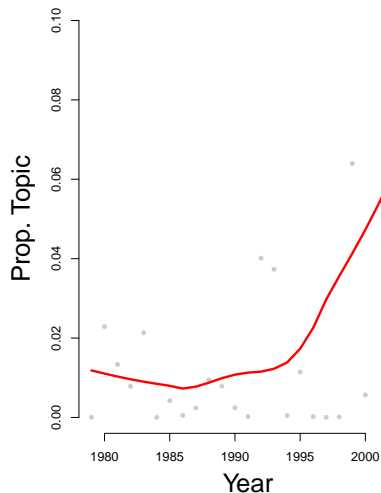
# History of Home Style

**Ideological Shirking**



# History of Home Style

**Biases in Congressional Communication**





# Prepackaged Data Sources

Lexis Nexis (and other data base sources)

# Prepackaged Data Sources

Lexis Nexis (and other data base sources)

- 1) Batch search and download

# Prepackaged Data Sources

Lexis Nexis (and other data base sources)

- 1) Batch search and download
- 2) **Do not try to scrape Lexis Nexis(!!!!)**

# Prepackaged Data Sources

Lexis Nexis (and other data base sources)

- 1) Batch search and download
- 2) **Do not try to scrape Lexis Nexis(!!!!)**

Application Programming Interface (APIs)

# Prepackaged Data Sources

Lexis Nexis (and other data base sources)

- 1) Batch search and download
- 2) **Do not try to scrape Lexis Nexis(!!!!)**

Application Programming Interface (APIs)

- Facilitate interaction with applications (like Twitter)

# Prepackaged Data Sources

Lexis Nexis (and other data base sources)

- 1) Batch search and download
- 2) **Do not try to scrape Lexis Nexis(!!!!)**

Application Programming Interface (APIs)

- Facilitate interaction with applications (like Twitter)
- Download data (often in JSON format)~→ Twitter, Data.gov, ...

# Books, Archives, and Other Non-Digital Material

befit the servant towards the master; and he will not behave like many, who on meeting any great prince, with whom if only they have spoken but once, press forward with a certain smiling and friendly look, as if they" wished to caress an equal or show favour to an inferior.

"He will very rarely or almost never ask anything of his lord j for himself, lest his lord, being reluctant to deny it to him "directly, may sometimes grant it with an ill grace, which is much worse. Even in asking for others he will choose his time discreetly and ask proper and reasonable things; and he will so frame his request, by omitting what he knows may displease and by skilfully doing away with difficulties, that his lord shall always grant it, or shall not think him offended by refusal even if it be denied; for when lords have denied a favour to an importunate suitor, they often reflect that he who asked it with such eagerness, must have desired it greatly, and so having failed to obtain it, must feel ill will towards him who denied it; and believing this, they begin to hate the man and can never more look upon him with favour., 19.-" He will not seek to intrude unasked into his masters chamber or private retreats, even though he be of great consequence; for when great lords are in private, they often like a little liberty to say and do what they please, and do not wish to be seen or heard by any who may criticise them; and it is very proper. Hence I think those men do ill who blame great lords for consorting privately with persons who are of little worth save in matters of personal service, for I do not see why lords should not have the same freedom to relax their minds that we fain would have to relax ours. But if a Courtier accustomed to deal with important matters, chances to find himself in private with his lord, he must put on another face, postpone grave concerns to another place and time, and give the conversation a cast that shall amuse and please his lord, so as not to

# Books, Archives, and Other Non-Digital Material

## 1) Create images of texts



# Books, Archives, and Other Non-Digital Material

- 1) Create images of texts
- 2) Optical Character Recognition

# Books, Archives, and Other Non-Digital Material

- 1) Create images of texts
- 2) Optical Character Recognition
  - Built in Adobe Pro

# Books, Archives, and Other Non-Digital Material

- 1) Create images of texts
- 2) Optical Character Recognition
  - Built in Adobe Pro
  - Abbyy FineReader (Batch processing)

# Books, Archives, and Other Non-Digital Material

- 1) Create images of texts
- 2) Optical Character Recognition
  - Built in Adobe Pro
  - Abbyy FineReader (Batch processing)
  - Tesseract (Google, command line tool)

# Books, Archives, and Other Non-Digital Material

- 1) Create images of texts
- 2) Optical Character Recognition
  - Built in Adobe Pro
  - Abbyy FineReader (Batch processing)
  - Tesseract (Google, command line tool)
- 3) Also use, e-book formats...

# Acquiring Data from Web: Automated Web Collection

## HAGAMAN VOLUNTEER FIRE DEPARTMENT RECEIVES FEDERAL GRANT



08/22/12

HAGAMAN, N.Y. – Congressman Paul Tonko announced today that the federal government has awarded a grant of \$61,332 to the Hagaman Volunteer Fire Department, Inc. through the Department of Homeland Security's Assistance to Firefighters Grant Program in the eighth round of Fire Prevention & Safety (FP&S) announcements. The grant will help the company purchase a new safety trailer to provide fire prevention and life safety training to residents throughout Montgomery County.

**"Our first responders not only help us in times of need, they educate our communities on safety and prevention," said Congressman Paul Tonko. "I want to congratulate Hagaman on receiving this award. These are the sort of investments that are worth making – bettering our communities and improving our quality of life."**

"The Hagaman Volunteer Fire Department is humbled and honored to receive this Fire Prevention and Safety grant award to help improve our educational programs not only in the Village of Hagaman and the Town of Amsterdam but throughout our neighboring communities as well," said Hagaman Volunteer Fire Department, Inc. Chief Donald Reksc. "Our department is currently able to provide educational programs to about 1,800 children and adults annually. Through the use of this award, we'll be able to purchase a new safety trailer that will also allow us to accommodate students we currently haven't been able to include such as disabled and special needs students. We're very excited for the opportunity to extend our program further to our communities."

[Here are the details of the award:](#)

# Acquiring Data from Web: Automated Web Collection

```
<table border="0" cellpadding="0" cellspacing="0" align="right"><tr><td valign="top">
<a href="#" onclick="return popup('/common/popup/popup.cfm?action=item.print&itemID=715',600,500);"></a>
</td></tr></table>
```

**HAGAMAN VOLUNTEER FIRE DEPARTMENT RECEIVES FEDERAL GRANT**

08/22/12

HAGAMAN, N.Y. — Congressman Paul Tonko announced today that the federal government has awarded a grant of \$61,332 to the Hagaman Volunteer Fire Department, Inc. through the Department of Homeland Security's Assistance to Firefighters Grant Program in the eighth round of Fire Prevention & Safety (FP&S) announcements. The grant will help the company purchase a new safety trailer to provide fire prevention and life safety training to residents throughout Montgomery County.

"Our first responders not only help us in times of need, they educate our communities on safety and prevention," said Congressman Paul Tonko. "I want to congratulate Hagaman on receiving this award. These are the sort of investments that are worth making — bettering our communities and improving our quality of life."

The Hagaman Volunteer Fire Department is humbled and honored to receive this Fire Prevention and Safety grant award to help improve our educational programs not only in the Village of Hagaman and the Town of Amsterdam but throughout our neighboring communities as well," said Hagaman Volunteer Fire Department, Inc. Chief Donald Rekac. "Our department is currently able to provide educational programs to about 1,800 children and adults annually. Through the use of this award, we'll be able to purchase a new safety trailer that will also allow us to accommodate students we currently haven't been able to include such as disabled and special needs students. We're very excited for the opportunity to extend our program further to our communities."

Here are the details of the award:

· Hagaman Volunteer Fire Department, Inc: \$61,332 to purchase a new safety trailer for fire prevention and training purposes.

On April 11, 2012 Congressman Tonko wrote a letter of support on behalf of the Hagaman Volunteer Fire Department for the grant to the Assistant Administrator of Federal Emergency Management Agency (FEMA) Grant Programs.

The Fire Prevention and Safety Grants (FP&S) are part of the Assistance to Firefighters Grants (AFG), and are under the purview of the Grant Programs Directorate at FEMA. FP&S Grants support projects that enhance the safety of the public and firefighters from fire and related hazards. The primary goal is to target high-risk populations and reduce injury and prevent death. In 2005, Congress reauthorized funding for FP&S and expanded the eligible uses of funds to include Firefighter Safety Research and Development.

For more information on the FP&S grants program, click [here](http://www.fema.gov/fire-prevention-safety-grants).

**FP&S Grants**

# Acquiring Data from Web: Automated Web Collection

```
base = 'http://tonko.house.gov'

for j in range(len(html)):#
    out = urlopen(html[j]).read()
    soup = BeautifulSoup(out)
    h3s = soup.findAll('h3')
    fr = []
    date = []
    for m in range(len(h3s)):
        dd = h3s[m].findNext('a')
        dd = dd['href']
        dd2 = base + dd.encode('UTF-8')
        fr.append(dd2)
        temp = h3s[m].findNext('span')
        temp2 = util.clean_html(str(temp)).split('/')
        mons = mon_key[temp2[0]]
        day = temp2[1]
        year = '20' + temp2[2]
        temp3 = day + mons + year
        date.append(temp3)
    for num in range(len(fr)):
        out2 = urlopen(fr[num]).read()
        soup2 = BeautifulSoup(out2)
        divs = soup2.findAll('div')
        content = ''
        for m in range(len(divs)):
            if divs[m].has_key('class'):
                if divs[m]['class']=='contentdata':
                    stuff = util.clean_html(str(divs[m]))
                    content += stuff
        names = date[num] + 'Tonko' + str(num) + '.txt'
        files = open(names, 'w')
        files.write(content)
    files.close()
```



# Acquiring Data from Web: Automated Web Collection

WASHINGTON, D.C. -- Rep. Paul Tonko (NY-21) released the following statement on the passage of H.R. 3962, the Affordable Health Care for America Act:

"Today the House of Representatives took a giant step towards fixing our broken health care system by passing legislation that will provide coverage for millions of uninsured Americans, strengthen Medicare for our seniors, lower costs for businesses and individuals, and provide protections for those who already have health care coverage. In the 21st Congressional District alone, this bill will cover 22,000 of the uninsured and close the Medicare Part "donut hole" that currently has 7,300 of our seniors paying out of pocket for prescription drug costs.

"As I traveled throughout the district over the past 10 months, I heard heartbreaking stories of families thrust into bankruptcy because they had been denied coverage when they became ill, heard from people who've had to decide between buying food and prescription drugs, and from small business owners who cannot provide coverage to their employees because it's too expensive. The overwhelming number of voices have told me we need to fix the system, and that's what we are doing today.

# Acquiring Data from Web: Automated Web Collection

WASHINGTON, D.C. -- Rep. Paul Tonko (NY-21) released the following statement on the passage of H.R. 3962, the Affordable Health Care for America Act:

"Today the House of Representatives took a giant step towards fixing our broken health care system by passing legislation that will provide coverage for millions of uninsured Americans, strengthen Medicare for our seniors, lower costs for businesses and individuals, and provide protections for those who already have health care coverage. In the 21st Congressional District alone, this bill will cover 22,000 of the uninsured and close the Medicare Part "donut hole" that currently has 7,300 of our seniors paying out of pocket for prescription drug costs.

"As I traveled throughout the district over the past 10 months, I heard heartbreaking stories of families thrust into bankruptcy because they had been denied coverage when they became ill, heard from people who've had to decide between buying food and prescription drugs, and from small business owners who cannot provide coverage to their employees because it's too expensive. The overwhelming number of voices have told me we need to fix the system, and that's what we are doing today.

Exercise: Scraping a Presidential Speech: HW1 on the github :

- <http://www.crummy.com/software/BeautifulSoup/>
- Parse paragraphs, label speakers

# Acquiring Data from Web: Distributed Human Computing

## Amazon.com's Mechanical Turk

- Marketplace for Human Intensive Tasks
- Requester (you): create HITs, offer \$ (about \$0.05 per task)
- Workers (bored + broke people): complete task
- Requester: evaluate and pay

Odesk, elance, ...

# You have text, now what?

# Regular Expressions (from Jurafsky Slides)

## REGULAR EXPRESSIONS

<

< PREV

RANDOM

NEXT >

>

WHENEVER I LEARN A NEW SKILL I CONCOCT ELABORATE FANTASY SCENARIOS WHERE IT LETS ME SAVE THE DAY.

OH NO! THE KILLER MUST HAVE FOLLOWED HER ON VACATION!



BUT TO FIND THEM WE'D HAVE TO SEARCH THROUGH 200 MB OF EMAILS LOOKING FOR SOMETHING FORMATTED LIKE AN ADDRESS!



IT'S HOPELESS!

EVERYBODY STAND BACK.



I KNOW REGULAR EXPRESSIONS.



# 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	" <u>Money</u> "
[abc]	'a', 'b', or 'c'	"Investing in <u>Ira</u> n" "is <u>dan</u> gerous <u>bu</u> siness"
[1234567890]	any digit	"sitting on \$ <u>7.5</u> billion dollars" " <u>2005</u> and <u>2006</u> , more than " "\$ <u>150</u> million dollars"
[\.]	A period	" 'Run!', he screamed <u>.</u> "

# Regular Expressions, Some Basics (from Jurafsky Slides)

## - Ranges

RE	Match	Example Patterns Matched
[A-Z]	an upper case letter	" <u>R</u> ep. <u>A</u> nthony <u>W</u> einer ( <u>D</u> - <u>B</u> rooklyn & <u>Q</u> ueens)"
[a-z]	a lower case letter	"ACORN' <u>s</u> "
[0-9]	a single digit	"( <u>9</u> th 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	<u>“color”</u> or <u>“colour ”</u>
oo*h!	Words with o 0 or more times	<u>“oh!”</u> or <u>“ooh!”</u> or <u>“oooh!”</u>
o+h!	Words with o 1 or more times	<u>“oh!”</u> or <u>“ooh!”</u> or <u>“oooooh!”</u> 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 “begn” (Poor grammar!)

# Regular Expressions, Some Basics (from Jurafsky Slides)

- Start of the line anchor  $\wedge$ , end of the line anchor  $\$$

RE	Match	Example Patterns Match
$\wedge[A-Z]$	Upper case start of line	" <u>P</u> alo Alto" "the town of Palo Alto"
$\wedge[\wedge A-Z]$	Not upper case start of line	" <u>t</u> he town of Palo Alto" "Palo Alto"
$\wedge.$	Start of line	" <u>P</u> alo Alto" " <u>t</u> he town of Palo Alto"
$.\$$	Identify character that ends a line	"Wait <u>!</u> " "This is the end <u>.</u> "

# 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 <u>yours</u> or <u>mine</u> ”
\ d	Any digit	“ <u>1</u> -Mississippi”
\ D	Any non-digit	“1- <u>Mississippi</u> ”
\ s	Any whitespace character	“1, <u>2</u> ”
\ S	Any non-whitespace character	“ <u>1</u> , <u>2</u> ”
\ w	Any alpha-numeric	“ <u>1</u> - <u>Mississippi</u> ”
\ W	Any non-alpha numeric	“1- <u>Mississippi</u> ”

# 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

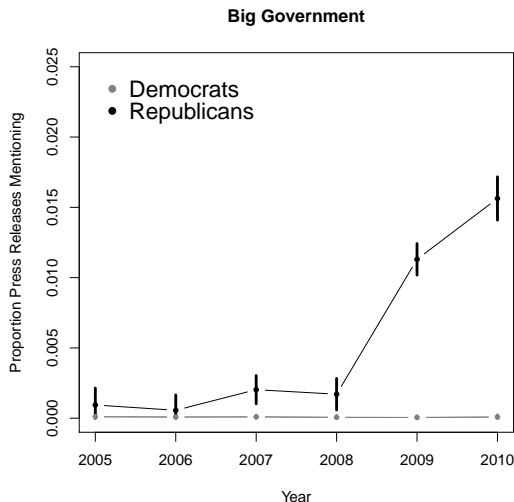
- **(^ | [^ a-zA-Z])[tT]he[^ a-zA-Z]**

# An Example: Searching for Tea Party Language

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

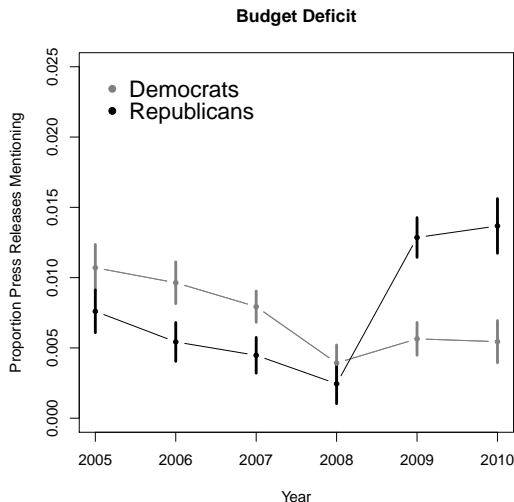
# An Example: Searching for Tea Party Language

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



# An Example: Searching for Tea Party Language

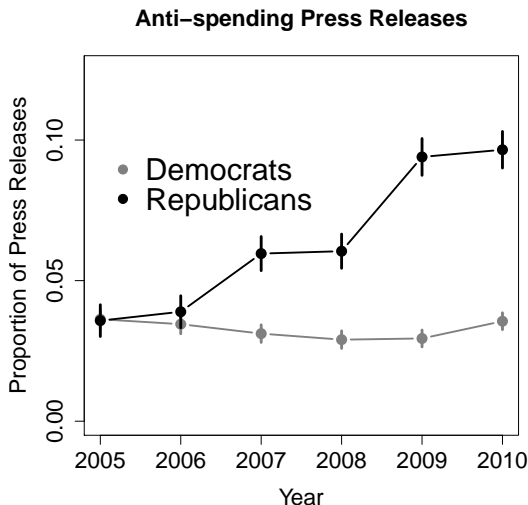
Grimmer, Westwood, and Messing (2014): Criticism and credit





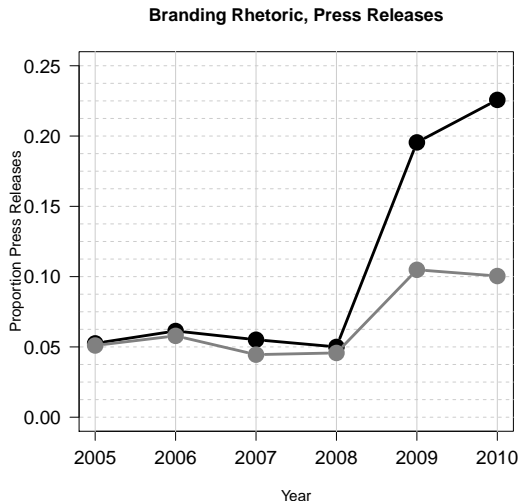
# An Example: Searching for Tea Party Language

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

WUSTL\_1.py

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

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
---------	---

Unigrams

# The Bag of Words Assumption

## Assumption: Discard Word Order

	Bigram	Count
	now we	1
	we are	1
	are engaged	1
	engaged in	1
	in a	1
	a great	1
Bigrams	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

	Trigram	Count
Trigrams	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 → 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↪ abbreviations, titles, etc

To the Python code!

# 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**

Python Code!

# 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:

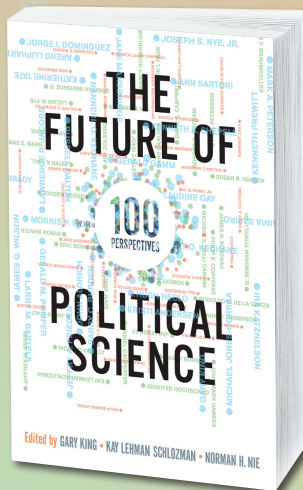
Step 3: Remove stop words :

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](http://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

# 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
<b>Machine</b>	<b>2.24</b>	<b>2.08</b>	<b>2.40</b>

# 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
<b>Machine</b>	<b>2.24</b>	<b>2.08</b>	<b>2.40</b>

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