# Advanced Quantitative Techniques
(Week 6)

Gregory M. Eirich
QMSS

# Agenda

1. Sensitizing models for context and semantics
2. Topic models
3. How do I get text again?

# Finalize Python results to come …

master ▾ QMSS-adv-analytic-techniques / Class6 - Context for texts.ipynb  Go to file  ···

melsyt upload updated files  Latest commit 4ebf9f1 on Jan 18, 2019  History

1 contributor

1968 lines (1968 sloc) | 152 KB  Raw  Blame

## Advanced quantitative techniques - Class 6 - Context for texts

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
files = ['Trump.Inaug.2017', 'Obama.Inaug.2009']
paper_words = {}

for d in files:
    file_path = 'Data/' + str(d) + '.txt'
    with open(file_path, 'rb') as f:
        f = [f.read().decode('utf8', 'ignore').replace('\r\n', " ").replace('\ufeff', '')]
        paper_words[d] = f
```

# 1. Sensitizing models for context and semantics

# Making our analysis more subtle

- What can we do to make our analysis more reflective of the lived reality of text usage and communication?

# 1-- Thinking happens in n-grams

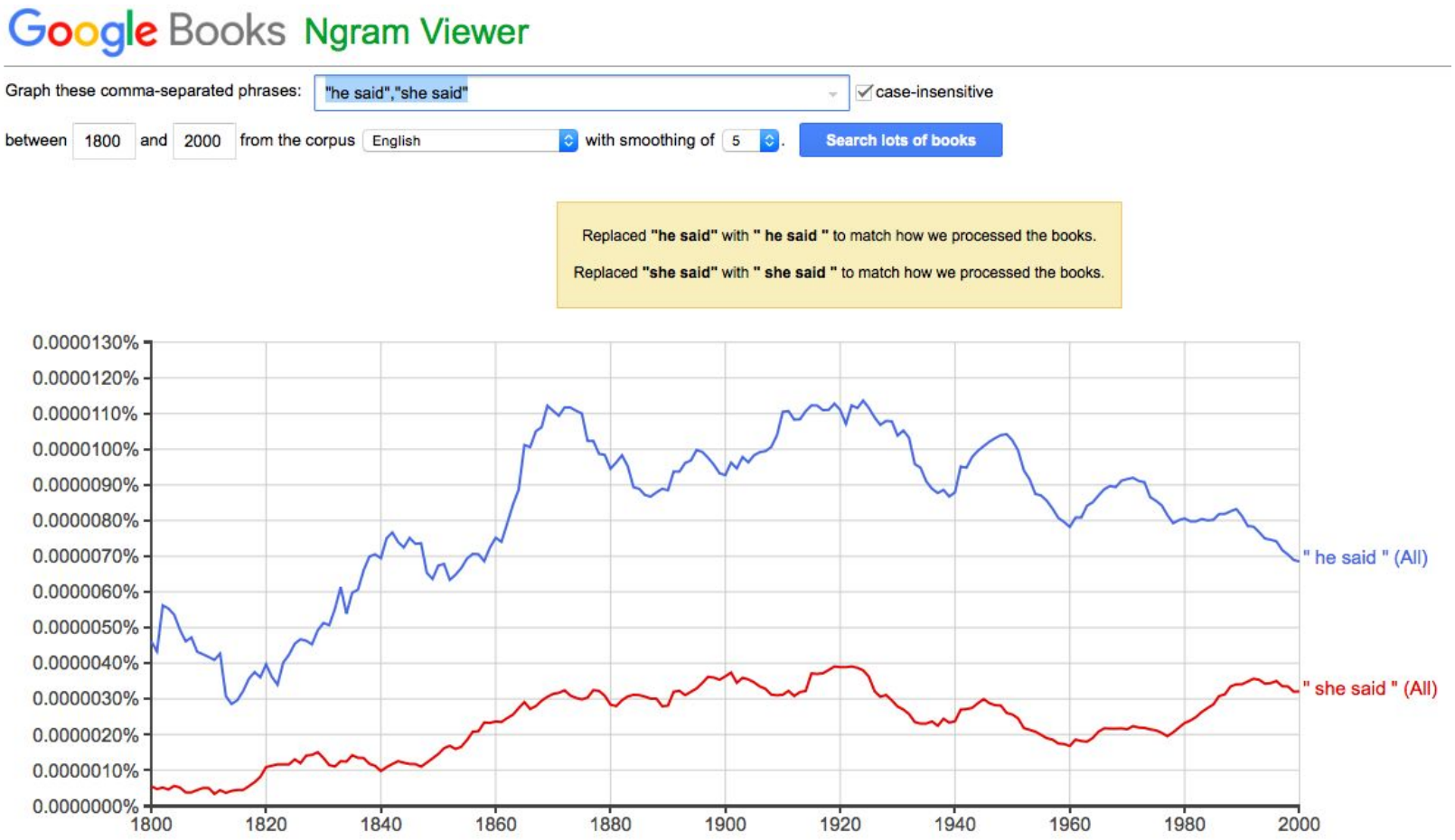
- N-grams are simply phrases at level n (1, 2, 3, etc.)

- Concepts often need modification to be more meaningful or understood

# Some examples

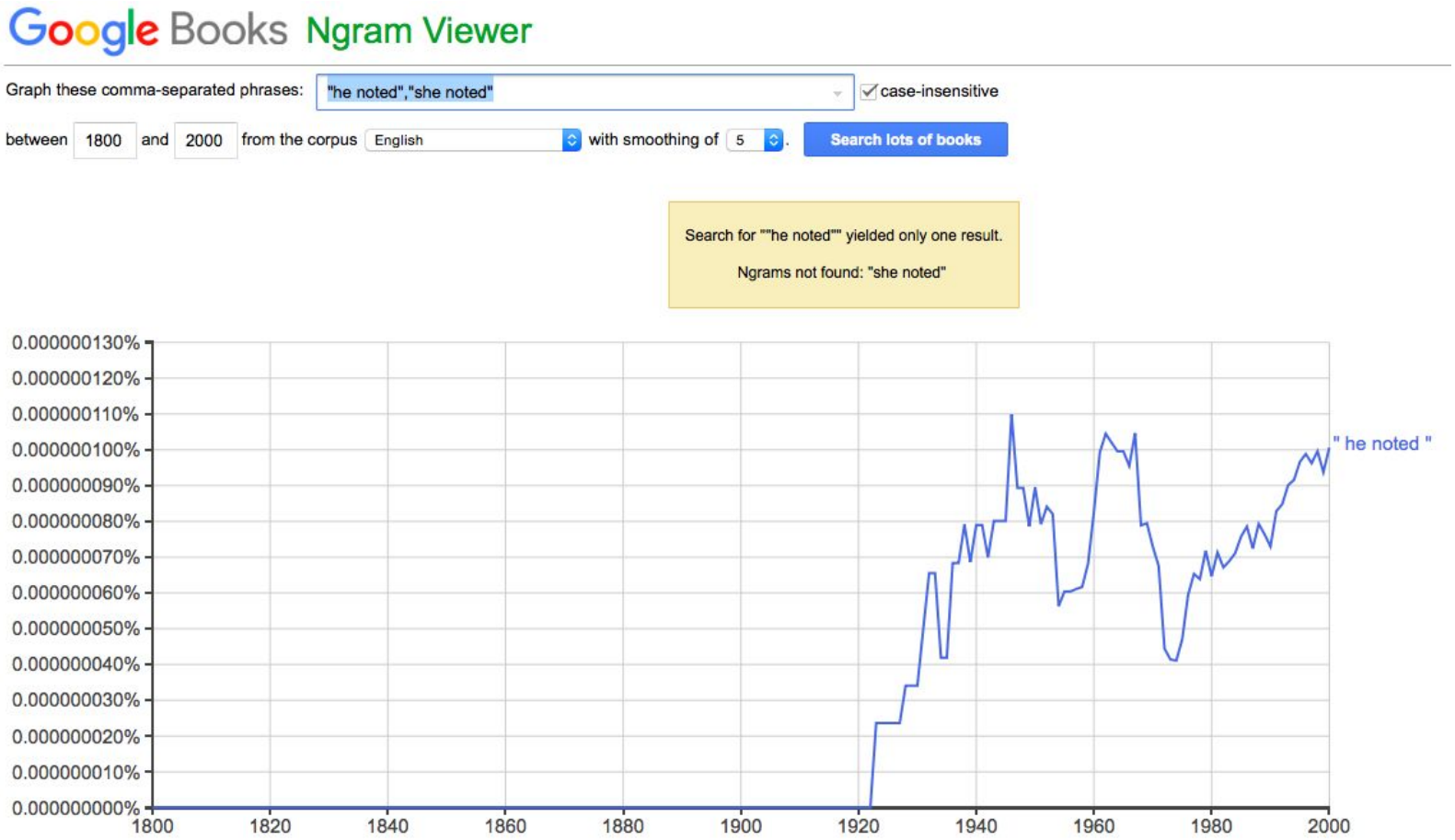
# He said, she said

- Movement of voices over 200 years in English



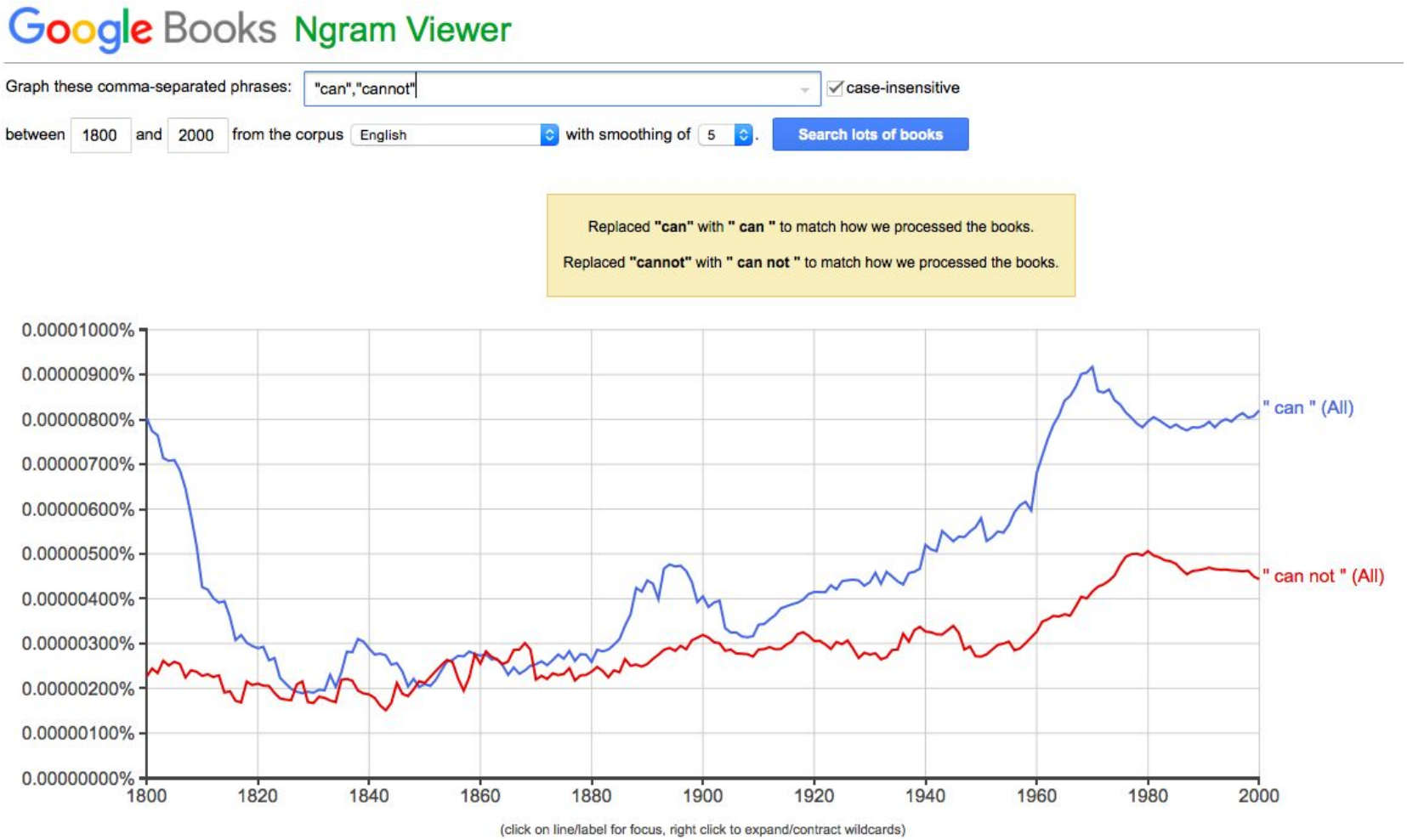
# He noted, she noted

- But “she noted” doesn’t even register



# We cannot, but we can!

- The growth of the “can do” attitude?



# Remember this paper?

- These guys use it as one stage of their work

**Detecting Latent Ideology in Expert Text: Evidence From Academic Papers in Economics**

**Zubin Jelveh[1], Bruce Kogut[2], and Suresh Naidu[3]**

[1]Dept. of Computer Science & Engineering, New York University
[2]Columbia Business School and Dept. of Sociology, Columbia University
[3]Dept. of Economics and SIPA, Columbia University
zj292@nyu.edu, bruce.kogut@columbia.edu, sn2430@columbia.edu

**Abstract**

Previous work on extracting ideology from text has focused on domains where expression of political views is expected, but it's unclear if current technology can work in domains where displays of ideology are considered inappropriate. We

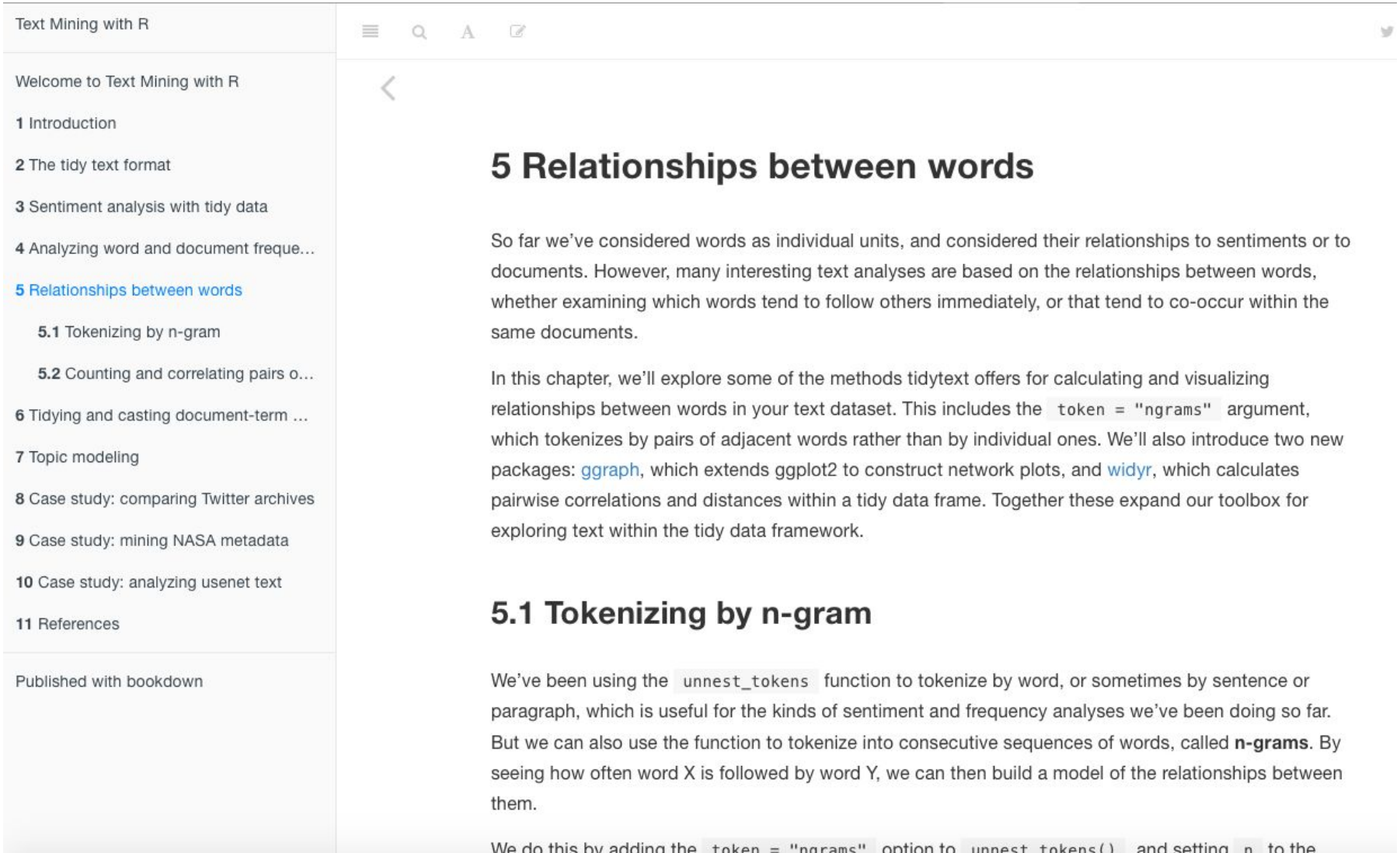
views itself as a science (Chetty, 2013) carefully applying rigorous methodologies and using institutionalized safe-guards such as peer review. The field's most prominent research organization explicitly prohibits researchers from making policy recommendations in papers that it releases (National Bureau of Economic Research, 2010). Despite these measures, economics' close proximity

# They use bi-grams as their unit of analysis

| Left-Leaning Bigrams | Right-Leaning Bigrams |
|---|---|
| mental_health | public_choic |
| post_keynesian | stock_return |
| child_care | feder_reserv |
| labor_market | yes_yes |
| health_care | market_valu |
| work_time | journal_financi |
| keynesian_econom | bank_note |
| high_school | money_suppli |
| polici_analys | free_bank |
| analys_politiqu | liquid_effect |
| politiqu_vol | journal_financ |
| birth_weight | median_voter |
| labor_forc | law_econom |
| journal_post | vote_share |
| latin_america | war_spend |
| mental_ill | journal_law |
| medic_care | money_demand |
| labour_market | gold_reserv |
| social_capit | anna_j |
| singl_mother | switch_cost |

# How do I do this in R?

Tidytext allows us to do this pretty straightforwardly ...





## 5 Relationships between words

So far we've considered words as individual units, and considered their relationships to sentiments or to documents. However, many interesting text analyses are based on the relationships between words, whether examining which words tend to follow others immediately, or that tend to co-occur within the same documents.

In this chapter, we'll explore some of the methods tidytext offers for calculating and visualizing relationships between words in your text dataset. This includes the `token = "ngrams"` argument, which tokenizes by pairs of adjacent words rather than by individual ones. We'll also introduce two new packages: ggraph, which extends ggplot2 to construct network plots, and widyr, which calculates pairwise correlations and distances within a tidy data frame. Together these expand our toolbox for exploring text within the tidy data framework.

### 5.1 Tokenizing by n-gram

We've been using the `unnest_tokens` function to tokenize by word, or sometimes by sentence or paragraph, which is useful for the kinds of sentiment and frequency analyses we've been doing so far. But we can also use the function to tokenize into consecutive sequences of words, called **n-grams**. By seeing how often word X is followed by word Y, we can then build a model of the relationships between them.

We do this by adding the `token = "ngrams"` option to `unnest_tokens()`, and setting `n` to the

# Obama's 2009 Inaugural Address

## The beginning verbatim:

**Vice President Biden, Mr. Chief Justice,members of the United States Congress, distinguished guests, and fellow citizens:**

**Each time we gather to inaugurate a President we bear witness to the enduring strength of our Constitution. We affirm the promise of our democracy. We recall that what binds this nation together is not the colors of our skin or the tenets of our faith or the origins of our names. What makes us exceptional -- what makes us American -- is our allegiance to an idea articulated in a declaration made more than two centuries ago:**
**"We hold these truths to be self-evident, that all men are created equal; that they are endowed by their Creator with certain unalienable rights; that among these are life, liberty, and the pursuit of happiness."**

**Today we continue a never-ending journey to bridge the meaning of those words with the realities of our time. For history tells us that while these truths may be self-evident, they've never been self-executing; that while freedom is a gift from God, it must be secured by His people here on Earth. The patriots of 1776 did not fight to replace the**

# Trump's 2017 Inaugural Address

## The beginning verbatim:

**Chief Justice Roberts, President Carter, President Clinton, President Bush, President Obama, fellow Americans, and people of the world: thank you.**

**We, the citizens of America, are now joined in a great national effort to rebuild our country and to restore its promise for all of our people. Together, we will determine the course of America and the world for years to come.**

**We will face challenges. We will confront hardships. But we will get the job done.**

**Every four years, we gather on these steps to carry out the orderly and peaceful transfer of power, and we are grateful to President Obama and First Lady Michelle Obama for their gracious aid throughout this transition. They have been magnificent.**

**Today's ceremony, however, has very special meaning. Because today we are not merely transferring power from one Administration to another, or from one party to another – but we are transferring power from**

# How do I do this in R?

## Getting started with “ngrams”

```
paper words <- data frame(file = paste0("/Users/gregoryeirich/Downloads/",
                                        c("Trump.Inaug.2017.txt", "Obama.Inaug.2009.txt")))
%>%
  mutate(text = map(file, read_lines)) %>%
  unnest() %>%
  group by(file = str sub(basename(file), 1, -5)) %>%
  mutate(line number = row_number()) %>%
  ungroup() %>%
  ## unnest tokens(word, text)  %>% ##
  unnest tokens(word, text, token = "ngrams", n = 2) ## how to get bigrams instead
## mutate(word = wordStem(word)) ## how to stem words

head(paper_words)
```

# How do I do this in R?

These are the first bi-grams

```
> head(paper words)
# A tibble: 6 × 3
              file line number            word
             <chr>       <int>           <chr>
1 Obama.Inaug.2009           1  vice president
2 Obama.Inaug.2009           1 president biden
3 Obama.Inaug.2009           1        biden mr
4 Obama.Inaug.2009           1        mr chief
5 Obama.Inaug.2009           1   chief justice
6 Obama.Inaug.2009           2      members of
```

# How do I do this in R?

Let's separate bigrams into their components and remove stops words and then put them back together ...

```
## bigram stuff now ##
paper words %>%
  count(word, sort = TRUE)

library(tidyr)

bigrams separated <- paper words %>%
  separate(word, c("word1", "word2"), sep = " ")

bigrams filtered <- bigrams separated %>%
  filter(!word1 %in% stop words$word) %>%
  filter(!word2 %in% stop_words$word)

# new bigram counts:
bigram counts <- bigrams filtered %>%
  count(word1, word2, sort = TRUE)

bigram_counts

bigrams united <- bigrams filtered %>%
  unite(word, word1, word2, sep = " ")

bigrams_united
```

# How do I do this in R?

Let's separate bigrams into their components and remove stops words and then put them back together ...

```
> bigrams united
# A tibble: 313 × 3
                file line number                    word
*               <chr>       <int>                   <chr>
1  Obama.Inaug.2009           1          vice president
2  Obama.Inaug.2009           1         president biden
3  Obama.Inaug.2009           1           chief justice
4  Obama.Inaug.2009           2 congress distinguished
5  Obama.Inaug.2009           2   distinguished guests
6  Obama.Inaug.2009           2         fellow citizens
7  Obama.Inaug.2009           4            bear witness
8  Obama.Inaug.2009           4       enduring strength
9  Obama.Inaug.2009           4        idea articulated
10 Obama.Inaug.2009           4           centuries ago
```

# How do I do this in R?

Let's make these bigrams into a Document-Term Matrix

```
pw = bigrams united[,c("file","word")]
d=  count_(pw, c("file", "word"))

pwdtm = d %>%
  cast_dtm(file, word, n)

## make the dtm into a dataframe ##

mpwdtm=as.matrix(pwdtm)
df.mpwdtm=as.data.frame(mpwdtm)

## make the dtm into a tdm instead ##

t.t = t(mpwdtm)
head(t.t, 50)


df.t.t = as.data.frame(t.t)
```

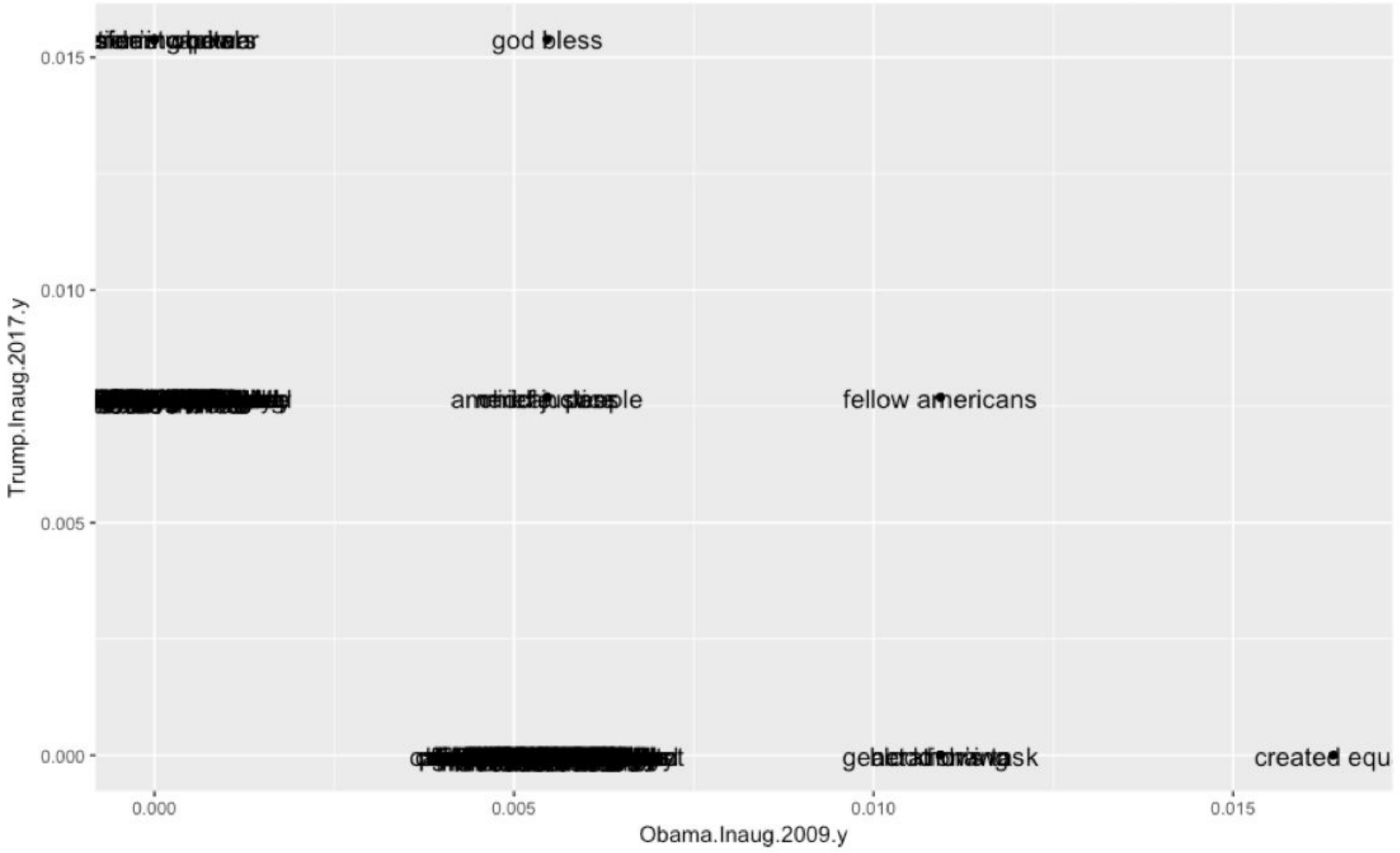
# How do I do this in R?

Let's make these bigrams into a Document-Term Matrix

```
summing = function(x) x/sum(x, na.rm=T)

df.t.t.2 = apply(df.t.t, 2, summing)

df.t.t$names<-rownames(df.t.t)
df.t.t = as.data.frame(t.t)
df.t.t$names<-rownames(df.t.t)
head(df.t.t)

df.t.t.2 = as.data.frame(df.t.t.2)
df.t.t.2$names<-rownames(df.t.t.2)
df.t.t.2 = as.data.frame(df.t.t.2)

total <- merge(df.t.t,df.t.t.2,by="names")

head(total)

total$obama.over.trump = (total$Obama.Inaug.2009.y) - (total$Trump.Inaug.2017.y)
sort.OT <- total[order(-total$obama.over.trump) , ]
sort.OT[1:30, ]

q = qplot(Obama.Inaug.2009.y, Trump.Inaug.2017.y, data = total)
q + geom_text(aes(label=names), size = 4.5)
```

# How do I do this in R?

Let's scatter the bi-grams



# Which bigrams sound most Obama-like?

Created equal; act knowing; blood drawn; generation's task; etc...

```
> sort.OT <- total[order(-total$obama.over.trump) , ]
> sort.OT[1:30, ]
                    names Obama.Inaug.2009.x Trump.Inaug.2017.x Obama.Inaug.2009.y Trump.Inaug.2017.y oba
68           created equal                  3                  0        0.016393443                   0
3              act knowing                  2                  0        0.010928962                   0
35             blood drawn                  2                  0        0.010928962                   0
123      generation's task                  2                  0        0.010928962                   0
5             afford delay                  1                  0        0.005464481                   0
10         america thrives                  1                  0        0.005464481                   0
13  america's possibilities                 1                  0        0.005464481                   0
14    america's prosperity                  1                  0        0.005464481                   0
23       american soldiers                  1                  0        0.005464481                   0
25          ancient values                  1                  0        0.005464481                   0
26             awesome joy                  1                  0        0.005464481                   0
27           basic measure                  1                  0        0.005464481                   0
28            bear witness                  1                  0        0.005464481                   0
30         begun america's                  1                  0        0.005464481                   0
33        bleakest poverty                  1                  0        0.005464481                   0
36      boundaries demands                  1                  0        0.005464481                   0
37         broad shoulders                  1                  0        0.005464481                   0
40            capped peaks                  1                  0        0.005464481                   0
43       central authority                  1                  0        0.005464481                   0
44           centuries ago                  1                  0        0.005464481                   0
45          change knowing                  1                  0        0.005464481                   0
50        citizen deserves                  1                  0        0.005464481                   0
51         citizens seared                  1                  0        0.005464481                   0
53          climate change                  1                  0        0.005464481                   0
55             code reform                  1                  0        0.005464481                   0
```

# Which bigrams sound most Trump-like?

American workers; nation's capital; transferring power; God bless; etc...

```
> sort.OT <- total[order(total$obama.over.trump) , ]
> sort.OT[1:30, ]
                      names Obama.Inaug.2009.x Trump.Inaug.2017.x Obama.Inaug.2009.y Trump.Inaug.2017.y ob
24          american workers                  0                  2        0.000000000        0.015384615
183         nation's capital                  0                  2        0.000000000        0.015384615
215          president obama                  0                  2        0.000000000        0.015384615
279       transferring power                  0                  2        0.000000000        0.015384615
127                god bless                  1                  2        0.005464481        0.015384615
1                  20th 2017                  0                  1        0.000000000        0.007692308
2         accept politicians                  0                  1        0.000000000        0.007692308
4          action constantly                  0                  1        0.000000000        0.007692308
6           almighty creator                  0                  1        0.000000000        0.007692308
7              america proud                  0                  1        0.000000000        0.007692308
8               america safe                  0                  1        0.000000000        0.007692308
9             america strong                  0                  1        0.000000000        0.007692308
11           america wealthy                  0                  1        0.000000000        0.007692308
12  america's infrastructure                  0                  1        0.000000000        0.007692308
15          american carnage                  0                  1        0.000000000        0.007692308
16          american destiny                  0                  1        0.000000000        0.007692308
17         american families                  0                  1        0.000000000        0.007692308
18             american flag                  0                  1        0.000000000        0.007692308
19            american hands                  0                  1        0.000000000        0.007692308
20         american industry                  0                  1        0.000000000        0.007692308
21            american labor                  0                  1        0.000000000        0.007692308
29        beautiful students                  0                  1        0.000000000        0.007692308
31          benefit american                  0                  1        0.000000000        0.007692308
32               bible tells                  0                  1        0.000000000        0.007692308
34             bless america                  0                  1        0.000000000        0.007692308
```

# How do I do this in R?

Trump and Obama bigrams are highly negatively correlated.  Why?

```
> cor(t.t, method="pearson")
                 Obama.Inaug.2009 Trump.Inaug.2017
Obama.Inaug.2009        1.0000000       -0.8737025
Trump.Inaug.2017       -0.8737025        1.0000000
```

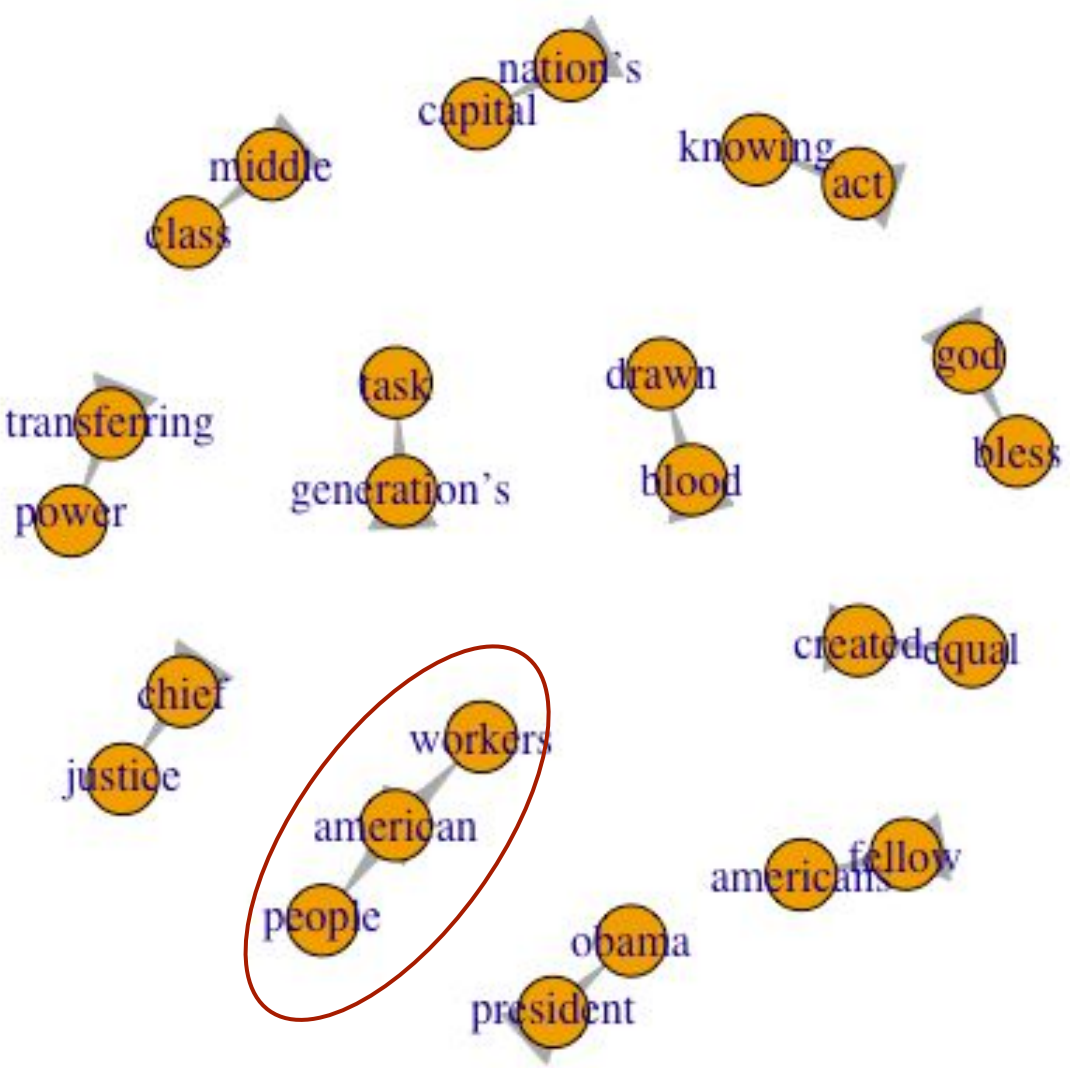
# How are words linked together?

We can do a networked graph of these bigrams (that appear at least 2 times)

```
bigram graph <- bigram_counts %>%
  filter(n > 1) %>%
  graph_from_data_frame()


layout <-layout.fruchterman.reingold(bigram_graph)
plot(bigram_graph, layout=layout)
```

# How are words linked together?

We can do a networked graph of these bigrams (that appear at least 2 times)



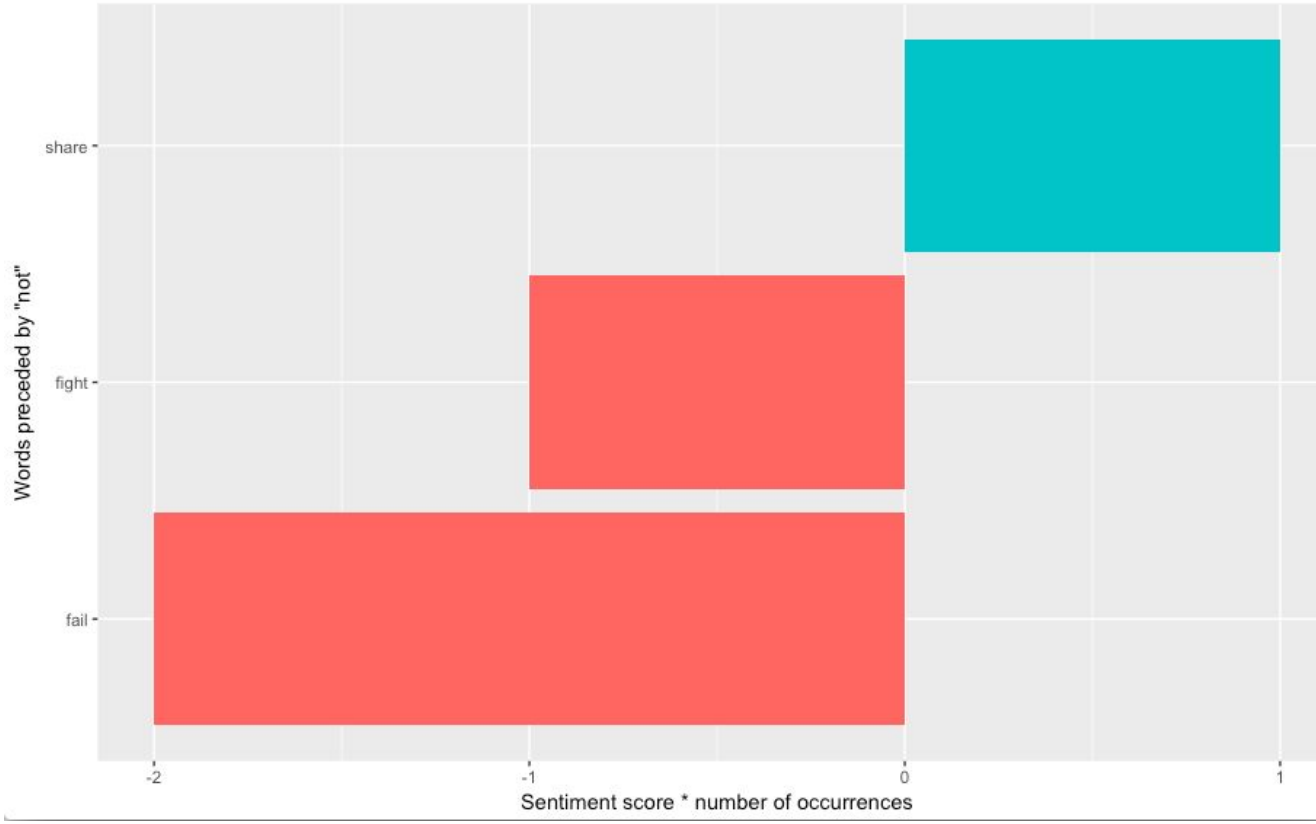
# What about improving sentiment analysis?

Where is “not” preceeding some words?

```
bigrams separated %>%
  filter(word1 == "not") %>%
  count(word1, word2, sort = TRUE)

AFINN <- get_sentiments("afinn")

not words <- bigrams separated %>%
  filter(word1 == "not") %>%
  inner join(AFINN, by = c(word2 = "word")) %>%
  count(word2, score, sort = TRUE) %>%
  ungroup()

not_words

# A tibble: 3 × 3
  word2 score     n
  <chr> <int> <int>
1  fail    -2     1
2 fight    -1     1
3 share     1     1
```

# What about improving sentiment analysis?

Which words change the most because they change their meaning with “not” in front of them?




```
not_words %>%
  mutate(contribution = n * score) %>%
  arrange(desc(abs(contribution))) %>%
  head(20) %>%
  mutate(word2 = reorder(word2, contribution)) %>%
  ggplot(aes(word2, n * score, fill = n * score > 0)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  xlab("Words preceded by \"not\"") +
  ylab("Sentiment score * number of occurrences") +
  coord_flip()
```

# What about improving sentiment analysis?

Where is “not” plus other *negatives* preceeding some words?

```
negation_words <- c("not", "no", "never", "without")

negated words <- bigrams separated %>%
  filter(word1 %in% negation words) %>%
  inner join(AFINN, by = c(word2 = "word")) %>%
  count(word1, word2, score, sort = TRUE) %>%
  ungroup()

negated_words

# A tibble: 7 × 4
  word1     word2 score     n
  <chr>     <chr> <int> <int>
1 never    forget    -1     1
2    no challenge    -1     1
3    no      fear    -2     1
4    no    matter     1     1
5   not      fail    -2     1
6   not     fight    -1     1
7   not     share     1     1
```

# What about tri-grams?

Note, I was lazy, so it says “bigrams” but it means “trigrams” here. Sorry!

```
paper words <- data frame(file = paste0("/Users/gregoryeirich/Downloads/",
                                        c("Trump.Inaug.2017.txt", "Obama.Inaug.2009.txt"))) %>%
  mutate(text = map(file, read_lines)) %>%
  unnest() %>%
  group by(file = str sub(basename(file), 1, -5)) %>%
  mutate(line number = row_number()) %>%
  ungroup() %>%
  ## unnest tokens(word, text)  %>% ##
  unnest tokens(word, text, token = "ngrams", n = 3) ## how to get bigrams instead
## mutate(word = wordStem(word)) ## how to stem words

head(paper_words)


## bigram stuff now ##
paper words %>%
  count(word, sort = TRUE)

library(tidyr)

bigrams separated <- paper words %>%
  separate(word, c("word1", "word2", "word3"), sep = " ")
```

# What about tri-grams?

Note, I was lazy, so it says “bigrams” but it means “trigrams” here. Sorry!

```
bigrams filtered <- bigrams separated %>%
  filter(!word1 %in% stop words$word) %>%
  filter(!word2 %in% stop words$word) %>%
  filter(!word3 %in% stop_words$word)

# new bigram counts:
bigram counts <- bigrams filtered %>%
  count(word1, word2, word3, sort = TRUE)

bigram_counts

bigrams united <- bigrams filtered %>%
  unite(word, word1, word2, word3, sep = " ")

bigrams_united

## filter(sum(n) >= 10) %>% ##
```

# Trump's most common tri-grams

## Substantive tri-grams happen most once for Trump

```
> sort.OT <- total[order(total$obama.over.trump) , ]
> sort.OT[1:30, ]
                          names Obama.Inaug.2009.x Trump.Inaug.2017.x Obama.Inaug.2009.y Trump.Inaug.2017.
1  action constantly complaining                 0                  1                 0         0.0333333
2         american carnage stops                 0                  1                 0         0.0333333
3          american workers left                 0                  1                 0         0.0333333
4    beautiful students deprived                 0                  1                 0         0.0333333
6       benefit american workers                 0                  1                 0         0.0333333
8           bush president obama                 0                  1                 0         0.0333333
9       carter president clinton                 0                  1                 0         0.0333333
10         chief justice roberts                 0                  1                 0         0.0333333
11   children safe neighborhoods                 0                  1                 0         0.0333333
13        clinton president bush                 0                  1                 0         0.0333333
17        decades we've enriched                 0                  1                 0         0.0333333
20        education system flush                 0                  1                 0         0.0333333
21     enriched foreign industry                 0                  1                 0         0.0333333
24             god bless america                 0                  1                 0         0.0333333
25             god's people live                 0                  1                 0         0.0333333
28             january 20th 2017                 0                  1                 0         0.0333333
29     justice roberts president                 0                  1                 0         0.0333333
31           lady michelle obama                 0                  1                 0         0.0333333
34           minds openly debate                 0                  1                 0         0.0333333
36        obama fellow americans                 0                  1                 0         0.0333333
38      president bush president                 0                  1                 0         0.0333333
39    president carter president                 0                  1                 0         0.0333333
40   president clinton president                 0                  1                 0         0.0333333
41        president obama fellow                 0                  1                 0         0.0333333
42     radical islamic terrorism                 0                  1                 0         0.0333333
```

# Obama’s most common tri-grams

## Substantive tri-grams happen most once for Obama too

```
> sort.OT <- total[order(-total$obama.over.trump) , ]
> sort.OT[1:30, ]
                          names Obama.Inaug.2009.x Trump.Inaug.2017.x Obama.Inaug.2009.y Trump.Inaug.2017
5   begun america’s possibilities                 1                   0         0.03448276         0.000000
7        boundaries demands youth                 1                   0         0.03448276         0.000000
12         climate change knowing                 1                   0         0.03448276         0.000000
14         common creed describes                 1                   0         0.03448276         0.000000
15  congress distinguished guests                 1                   0         0.03448276         0.000000
16      creed describes tolerance                 1                   0         0.03448276         0.000000
18         durably lift suspicion                 1                   0         0.03448276         0.000000
19     economy requires railroads                 1                   0         0.03448276         0.000000
22   founding principles requires                 1                   0         0.03448276         0.000000
23   freedoms ultimately requires                 1                   0         0.03448276         0.000000
26          honest labor liberate                 1                   0         0.03448276         0.000000
27 individual freedoms ultimately                 1                   0         0.03448276         0.000000
30        labor liberate families                 1                   0         0.03448276         0.000000
32           life’s worst hazards                 1                   0         0.03448276         0.000000
33           manage crisis abroad                 1                   0         0.03448276         0.000000
35        modern economy requires                 1                   0         0.03448276         0.000000
37      opportunity human dignity                 1                   0         0.03448276         0.000000
44          require perpetual war                 1                   0         0.03448276         0.000000
45     requires collective action                 1                   0         0.03448276         0.000000
46            rising middle class                 1                   0         0.03448276         0.000000
49         science teachers we’ll                 1                   0         0.03448276         0.000000
51              snow capped peaks                 1                   0         0.03448276         0.000000
52        spare philadelphia hall                 1                   0         0.03448276         0.000000
53    striving hopeful immigrants                 1                   0         0.03448276         0.000000
54             survive half slave                 1                   0         0.03448276         0.000000
```

# “Cliches”

- What about F. Scott Fitzgerald’s “Flappers and Philosophers”?
- Using Nick Beauchamp’s “Cliche Scorer” app...

Northeastern University

Department of Political Science
NULab for Texts, Maps and Networks

C.V. Research Teaching Text Tools Contact

## Nick Beauchamp

I am an Assistant Professor at Northeastern University in the Department of Political Science, the NULab for Text, Maps and Networks, and the Network Science Institute. I received my PhD from the NYU Department of Politics, specializing in U.S. politics (political behavior, campaigns, opinion, political psychology, social media) and political methodology (quantitative text analysis, machine learning, bayesian methods, agent-based models, networks).

My research examines how political opinions form and change as a result of discussion, deliberation and argument in domains such as legislatures, campaigns, social media, and the judiciary, using techniques from machine learning, automated text analysis, and social network analysis. My recent projects explore deliberative quality in online political forums; predict elections using Twitter textual data; test the representativeness of UK legislators based on their text-derived ideology; and visualize

# “Cliches”

Cliché Score

Start over

| | |
|---|---|
| western half of the | 0.73 |
| back to new york | 4.12 |
| he paused and looked | 0.49 |
| paused and looked at | 0.36 |
| in the air like | 2.68 |
| trying to give you | 0.93 |
| lifted her head and | 0.73 |
| the women and children | 2.89 |
| said the young man | 0.49 |
| corners of his mouth | 2.54 |
| come over here and | 2.02 |
| members of the crew | 1.11 |
| closed his eyes and | 3.8 |
| the men and women | 5.04 |
| from time to time | 6.82 |
| always wanted to be | 3.75 |
| years of his life | 3.83 |
| made all the difference | 2.14 |
| from day to day | 3.37 |
| of men and women | 4.85 |
| he shook his head | 6.09 |
| shook his head no | 1.07 |
| looked up and saw | 3.55 |
| he raised his voice | 1.04 |
| big enough to be | 1.21 |

- The higher the score, the more common is the 4-gram

# 2-- Parts of speech

- What part of speech words are can affect what they are doing and how much they matter

# An example

- Burscher, Bjorn, Rens Vliegenthart, and Claes H. de Vreese. "Frames Beyond Words Applying Cluster and Sentiment Analysis to News Coverage of the Nuclear Power Issue." *Social Science Computer Review* (2015): 0894439315596385.

# What they are interested in

- They write: “News media can shape public opinion regarding an issue by emphasizing some elements of the broader controversy over others. Research shows that aspects of an issue, which are more salient in the media, cause individuals to focus on these aspects when constructing their opinions.

- Shah, Watts, Domke, and Fan (2002), for example, showed that public approval of President Clinton depended on whether news coverage during the Lewinsky sex scandal focused on the sexual nature of Clinton’s indiscretion or the attacks of Republicans on Clinton’s behavior

# What about parts of speech?

- One thing they note is focusing on titles is really important to understand framing
- They note: “Related research on topic clustering has shown that giving higher weight to the title of a news article can increase the accuracy of topic clusters, because the title is more representative of the topic than the main text.

# What about parts of speech?

- They do part-of-speech tagging to select words that are a noun, an adjective, or adverb.
- They note: They believe that words from the selected classes (nouns, adjectives, and adverbs) are most indicative of frames. This is because other word classes, like verbs, conjunctions, or pronouns, are much less likely than the selected classes to add meaning to a frame.
- They note: Previous research has shown that giving higher weights to nouns than other word classes can increase the quality of topic clusters.
- BTW, they also apply named-entity recognition to remove all names of persons, organizations, and locations as well as times and dates. Names of countries and organizations, for example, refer to very specific events, while frames are more abstract semantic concepts. Therefore, it is more likely to get clusters, which actually discriminate between emphasis frames, when we remove named entities

# What about parts of speech?

- They note: To our knowledge, this has not been tested before in document clustering. When representing articles in the cluster analysis, we only use the above-mentioned parts of each article as features and ignore all other words. We call this the selection approach.
- In order to see whether this way of selecting features improves the validity of the cluster analysis, we conduct a baseline analysis where we use all words from each article as features (baseline approach). We compare cluster centers in the selection approach with clusters centers in the baseline approach.
- Furthermore, we conduct a manual content analysis to compare the accuracy of frame codings in both approaches. This leads to the following research question: To what extent does selecting frame related document features improve the construct validity and coding accuracy of statistical frame analysis?

# The typical baseline approach

**Table 2.** Clusters Baseline Approach.

| B1 | B2 | B3 | B4 | B5 | B6 | B7 |
|---|---|---|---|---|---|---|
| British | Commission | Indian | Carbon | Iran | Japan | India |
| Company | Chernobyl | Point | Gas | Iranian | Fukushima | Korea |
| Pound | Safety | Entergy | Emission | Russia | Tokyo | North |
| EDF | Waste | County | Wind | Weapon | Radiation | Treaty |
| Station | Company | Emergency | Climate | Uranium | Tepco | Weapon |
| Industry | Station | Westchester | Electricity | Program | Japanese | China |
| Price | Utility | Buchanan | Coal | Tehran | Water | Test |
| Cost | Official | Plan | Industry | Enrichment | Tsunami | Pakistani |
| Share | Site | Commission | Oil | Bushehr | Daichi | United |
| Million | Fuel | Siren | Station | Russian | Accident | Korean |
| Billion | Radiation | Steets | Renewable | United | Earthquake | Ban |
| Electricity | People | Federal | Cost | Sanction | Disaster | Official |
| Britain | Million | Evacuation | Waste | International | Safety | Agreement |
| BNFL | Industry | Official | Fuel | Official | Radioactive | Administration |
| Market | Public | Hudson | Renewables | Ahmadinejad | Worker | South |
| $N = 558$ | $N = 1{,}918$ | $N = 250$ | $N = 648$ | $N = 233$ | $N = 383$ | $N = 296$ |

# The parts-of-speech etc approach

**Table 3.** Clusters Selection Approach.

| S1 | S2 | S3 | S4 | S5 | S6 | S7 |
|---|---|---|---|---|---|---|
| Station | Energy | Company | Weapon | Commission | Fuel | Reactor |
| State | Gas | Government | Program | Regulatory | Uranium | Radiation |
| Mile | Government | Price | State | Federal | Waste | Radioactive |
| First | Oil | Industry | President | Reactor | Plutonium | Accident |
| Official | Source | Pound | Country | Safety | Radioactive | Safety |
| Government | Renewable | Cost | Official | Regulator | Spent | Water |
| Plan | Climate | Reactor | Agreement | State | Reactor | Disaster |
| Security | Electricity | Electricity | Test | License | Rod | Leak |
| People | Policy | Share | Energy | Agency | State | Level |
| World | Emission | Plan | Foreign | Company | Enrichment | Worker |
| Last | Change | State | Treaty | Official | Storage | Exposure |
| Federal | Carbon | Utility | Nation | Problem | Company | Earthquake |
| Reactor | Coal | Generation | International | Utility | Material | Official |
| Attack | Minister | Last | World | Attack | Site | Station |
| Former | Generation | Energy | Uranium | Mile | Government | Operator |
| $N = 1,296$ | $N = 645$ | $N = 609$ | $N = 568$ | $N = 548$ | $N = 328$ | $N = 292$ |

# Comparing them

**Table 4.** Identified Frames Baseline and Selection Approach.

| | |
|---|---|
| Baseline approach | |
| Frame 1 | Economic aspects of nuclear power production |
| Frame 2 | Safety of nuclear plants, nuclear waste, nuclear power accidents and radiation risks |
| Frame 3 | Nuclear power and weapon development |
| Frame 4 | Role of nuclear power in electricity production and effects on climate change |
| Frame 5 | Evacuation of nuclear reactors |
| Selection approach | |
| Frame 1 | *Safety* of nuclear plants |
| Frame 2 | Role of nuclear power in electricity production and effects on *climate* change |
| Frame 3 | *Economic* aspects of nuclear power production |
| Frame 4 | Nuclear power and *weapon* development |
| Frame 5 | Processing of nuclear materials and nuclear *waste* |
| Frame 6 | Nuclear power *accidents* and radiation risks |

# Another example--



## AMERICAN MOVIE

*January 20, 2015*

· Visualization

*Update, January 22nd:* Now with plots standardized per thousand films released that year.

It's time for another episode of Data Analysis on the Bus. This one follows from an exchange on Twitter, prompted by the coverage of *American Sniper* about the tendency to use the word "American" in film titles, especially when you want things to sound terribly serious. This led to a bit of freewheeling and it has to be said perhaps tendentious cultural theorizing on my part. Rather more usefully, it also prompted Benjamin Schmidt to send me some IMDB data containing film titles with the word "American" in them. I did a correspondence analysis with it yesterday at lunchtime, but to be honest I wouldn't lean too heavily on it if I were you. On the bus to work this morning I couldn't resist messing around with it a bit more. I was going to say that this evening's State of the Union Address provides the thinnest of pretexts for posting these pictures, but Michael Lunny just informed me that Alex Godfrey at the Guardian had more or less the same thought today. So here are the pictures—just some trends, and this time I'll lay off the CultStud speculation. Mostly.

When cleaning the data I first threw out all the porn (the "Adult" genre). Here's the time series for films with "America" or "American" in the title. I don't have IMDB's annual data on total films released so I'm sorry to say, Reviewer C, that these numbers aren't standardized. Why are you peer reviewing a blog post like this in the first place?

*Update:* I couldn't leave well enough alone, so after some very helpful advice from Gabriel Rossman, who knows a hell of a lot about this data set, I went and pulled the most recent version of the IMDB Genres file and cleaned it quickly, more or less as before. As always, cleaning data with the attempt to replicate prior results proved to be a sobering experience. The plots aren't strictly comparable, but I'm keeping the originals rather than replacing

# A final example ---

CULTUREBOX | ARTS, ENTERTAINMENT, AND MORE. | NOV. 20 2013 11:51 PM

Slate

## A Textual Analysis of *The Hunger Games*

Suzanne Collins' favorite adjectives, adverbs, and ways of starting a sentence.

15.8k 53

*By Ben Blatt*

CONNEC

Terms and Limitatic

Katniss Everdeen (Jennifer Lawrence) in *The Hunger Games*. She'll probably be "weak," "wild," or "furious."

Photo by Murray Close/Lionsgate, Inc.

# -ly words

## Most Distinctive '-ly' Adverbs by Author

| SUZANNE COLLINS<br>*Hunger Games Series* | STEPHENIE MEYER<br>*Twilight Series* | J.K. ROWLING<br>*Harry Potter Series* |
|---|---|---|
| Repeatedly | Amazingly | Feebly |
| Genuinely | Intently | Promptly |
| Genetically | Deliberately | Forcefully |
| Intensely | Crookedly | Grumpily |
| Basically | Anxiously | Kindly |
| Currently | Physically | Coldy |
| Severely | Furiously | Miserably |
| Exclusively | Strangely | Dreamily |
| Obediently | Wildly | Resolutely |
| Voluntarily | Slowly | Apprehensively |

# Adjectives

## Most Distinctive Adjectives by Author

| SUZANNE COLLINS<br>*Hunger Games* Series | STEPHENIE MEYER<br>*Twilight* Series | J.K. ROWLING<br>*Harry Potter* Series |
|---|---|---|
| Drunk | Unwilling | Nasty |
| Good-natured | Unreadable | Considerable |
| Functional | Instinctive | Terrified |
| Preliminary | Impatient | White-hot |
| Initial | Hesitant | Open-mouth |
| Last-ditch | Irresistible | Gleeful |
| Despicable | Compatible | Magical |
| Diagonal | Carefree | Squashy |
| Lethal | Wistful | Spiral |
| Rich | White-haired | Famous |

Created by @BenBlatt of Slate.com
Source: *Harry Potter* 1-7, *Hunger Games* 1-3, *Twilight* 1-4
Distinctive = Freq(Word |Author 1)/Freq(Word | Any of the Three Authors)
Words not used by multiple authors and less than ten times were excluded

# BTW-- Sentences

- News media can shape public opinion regarding an issue by emphasizing some elements of the broader controversy over others (Iyengar...

## Most Common Sentences By Each Author

| SUZANNE COLLINS<br>*Hunger Games Series* | STEPHENIE MEYER<br>*Twilight Series* | J.K. ROWLING<br>*Harry Potter Series* |
|---|---|---|
| My name is Katniss Everdeen. | I sighed. | Nothing happened. |
| I don't know. | He sighed. | Harry looked around. |
| I shake my head. | I shrugged. | Harry stared. |
| I am seventeen years old. | I frowned. | He waited. |
| My home is District 12. | He chuckled. | Harry said nothing. |
| Now I wish I had. | I laughed. | They looked at each other. |
| I swallowed hard. | He shrugged. | Harry blinked. |
| He hesitates. | I flinched. | He looked around. |
| I'm not really surprised. | I took a deep breath. | Something he didn't have last time. |
| Something is wrong. | He didn't answer. | He stood up. |

Created by @BenBlatt of Slate.com
Source: *Harry Potter* 1-7, *Hunger Games* 1-3, *Twilight* 1-4

# How do I do this in R?

Parts of speech tagging in R is objectively difficult.  Sorry!

# Parts of speech

- What part of speech words are can affect what they are doing and how much they matter

# How do I do this in R?

Parts of speech tagging in R is objectively difficult.  There are a few packages designed for this:

- NLPclean
- RDRPOSTagger

# Adverbs

- I am going to use a (1) static table of parts of speech and (2) regular expression rules to look at adverbs in the States of the Union

- This is obviously oversimplifying because words can mean different things by being used differently in sentences, but it is a start ...

# Adverbs

Bringing in some text and code; thanks to "“I Have The Best Words.” How Trump's First SOTU Compares To All The Others."

```
library(tidyverse)
library(tidytext)
library(stringr)
library(SnowballC)
library(lubridate)
library(readr)
library(stringr)
library(quanteda)
library(ggplot2)
library(DT)
```

# Adverbs

## Getting started

```
sou <- read_csv(file.choose()) ## load SOU csv

presidents <- read_csv(file.choose()) ## load presidential party affiliation

sou <- sou %>%
    left_join(presidents)  ## merge them together

sou$order = 1:nrow(sou)  ## order the SOUs by date

sou$ID <- paste0(sou$president,"-",sou$date)  ## make an ID for each speech

parts_of_speech2 = read_csv(file.choose())
```

# Adverbs

## Merging the POS with the SOUs

```
parts_of_speech2 = read_csv(file.choose()) ## read in POS dictionary

parts <- sou %>%
    unnest_tokens(word, text) %>%
    filter(str_detect(word, "[a-z]")) %>%
    # match to lexicon
    inner_join(parts_of_speech2, by = "word") ## merge POS and SOUs together
```

# Adverbs

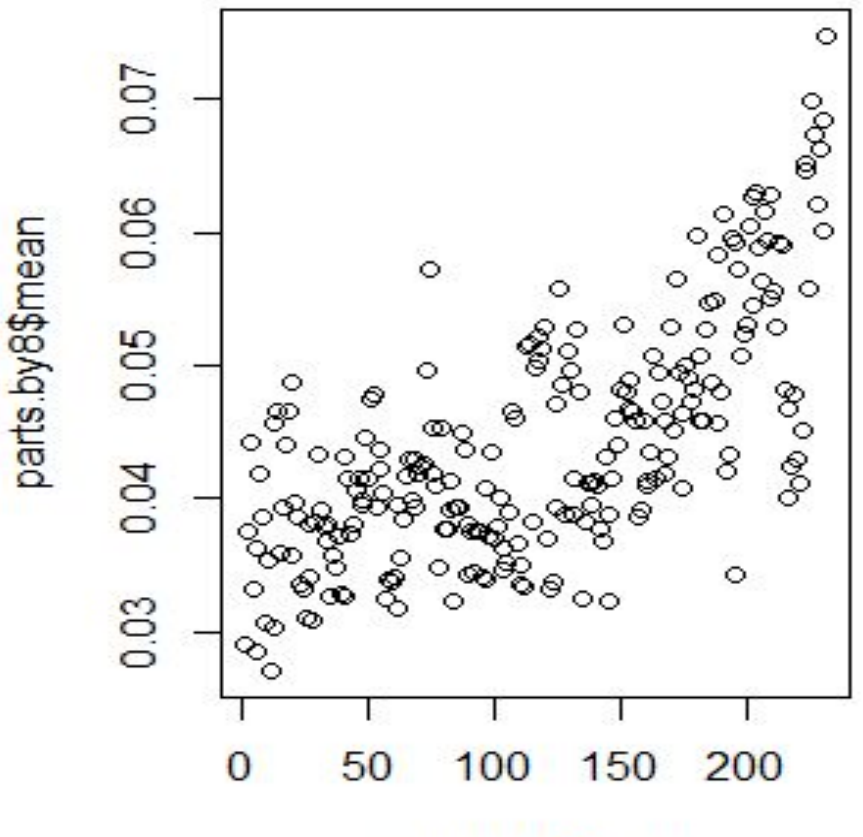
Getting overall proportions of adverb usage

```
parts$adverb = ifelse(parts$pos=="Adverb",1,0) ## find all the adverbs


parts.by = parts %>%  ## proportion of adverbs by speech
    group_by(ID) %>%
    summarise(mean = mean(adverb, na.rm=T), ord=mean(order, na.rm=T))
```

# Adverbs

An upward trend

```
plot(parts.by8$ord, parts.by8$mean)
```



# What are the biggest adverbs?

## Overall

```
> parts %>% filter(adverb==1) %>% count(word, sort = TRUE)
# A tibble: 992 x 2
   word         n
   <chr>    <int>
 1 so        3967
 2 now       3367
 3 only      2477
 4 most      2345
 5 also      1862
 6 well      1722
 7 over      1633
 8 out       1475
 9 very      1289
10 just      1261
11 long      1232
12 right     1207
13 far       1141
14 up        1117
15 where     1097
16 still     1075
17 already    972
18 yet        958
19 even       943
20 less       933
21 further    923
```

# What are the biggest adverbs?

## Early SOUs

```
> parts %>% filter(adverb==1 &
order<120) %>% count(word, sort =
TRUE)
# A tibble: 798 x 2
   word          n
   <chr>     <int>
 1 so         2397
 2 now        1638
 3 only       1470
 4 most       1427
 5 well       1023
 6 also        910
 7 over        790
 8 far         735
 9 thus        731
10 however     673
11 very        666
12 just        656
13 out         655
14 yet         646
15 right       638
16 long        627
17 less        624
18 still       569
19 where       562
```

## Late SOUs

```
> parts %>% filter(adverb==1 &
order>120) %>% count(word, sort =
TRUE)
# A tibble: 782 x 2
   word         n
   <chr>    <int>
 1 now       1684
 2 so        1507
 3 only       954
 4 also       939
 5 most       879
 6 over       820
 7 out        797
 8 up         702
 9 well       681
10 very       600
11 just       594
12 long       593
13 together   560
14 tonight    549
15 right      548
16 better     528
17 where      520
18 here       514
19 still      497
```

# Adverbs via -ly

Do the same thing, but for only -ly words (Sorry this is mixed in with sentiment analysis dataframe below!)

```
bing <- get_sentiments("bing")

sentiments <- sou %>%
    unnest_tokens(word, text) %>%
    filter(str_detect(word, "[a-z]")) %>%
    # match to lexicon
    inner_join(bing, by = "word")

sentiments$pos = ifelse(sentiments$sentiment=="positive",1,0)

sentiments$ID <- paste0(sentiments$president,"-",sentiments$date)

sentiments.by = sentiments %>%
    group_by(ID) %>%
    summarise(mean = mean(pos, na.rm=T))

sentiments.by$order = 1:nrow(sentiments.by)
```

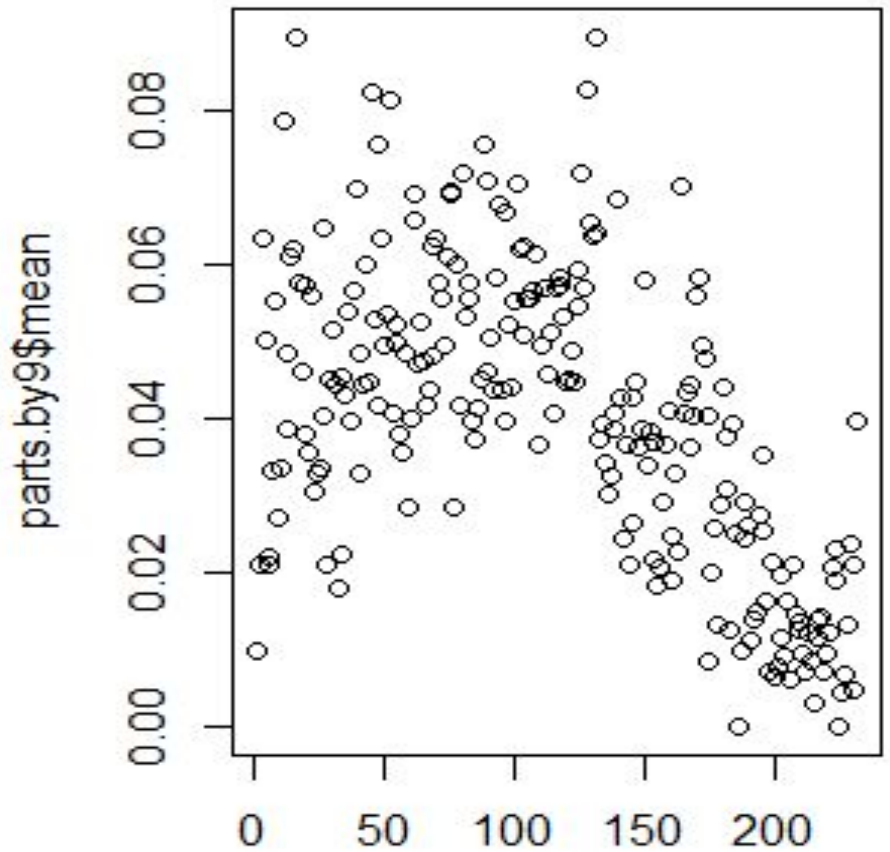
# Adverbs via -ly

Use regular expression to find -ly words

```
sentiments$ly = str_detect(sentiments$word, "ly$")

sentiments.by = sentiments %>%
  group_by(ID) %>%
  summarise(mean = mean(ly, na.rm=T))

str_view(sentiments$word, "ly$", match = T)

parts.by9 = sentiments %>%
  group_by(ID) %>%
  summarise(mean = mean(ly, na.rm=T), ord=mean(order, na.rm=T))

plot(parts.by9$ord, parts.by9$mean)
```

# Adverbs via -ly

A more complex pattern (this is correlated -.38 against the other adverb method)



# What are the biggest -ly adverbs

## Overall

```
> sentiments %>% filter(ly=="TRUE") %>% count(word) %>% arrange(-n)
# A tibble: 287 x 2
   word              n
   <chr>         <int>
 1 friendly        448
 2 earnestly       282
 3 promptly        234
 4 clearly         183
 5 properly        178
 6 successfully    137
 7 justly          128
 8 wisely          109
 9 effectively     106
10 scarcely        104
11 speedily        101
12 fairly           99
13 happily          94
14 sufficiently     91
15 respectfully     90
16 safely           83
17 unfortunately    81
18 faithfully       76
19 inevitably       76
20 readily          76
21 orderly          74
```

# What are the biggest -ly adverbs

## Early SOUs

```
> sentiments %>% filter(ly=="TRUE"
& order<120) %>% count(word) %>%
arrange(-n)
# A tibble: 233 x 2
   word                n
   <chr>           <int>
 1 friendly          389
 2 earnestly         216
 3 promptly          147
 4 justly            119
 5 properly          119
 6 scarcely           99
 7 clearly            90
 8 speedily           87
 9 respectfully       82
10 happily            78
11 successfully       77
12 sufficiently       71
13 fairly             67
14 faithfully         67
15 wisely             65
16 readily            62
17 safely             62
18 reasonably         50
19 unfortunately      49
```

## Late SOUs

```
> sentiments %>% filter(ly=="TRUE" &
order>120) %>% count(word) %>%
arrange(-n)
# A tibble: 210 x 2
   word                n
   <chr>           <int>
 1 clearly            92
 2 promptly           86
 3 effectively        64
 4 successfully       60
 5 earnestly          59
 6 friendly           59
 7 properly           57
 8 costly             44
 9 orderly            44
10 wisely             44
11 fairly             32
12 unfortunately      31
13 inevitably         26
14 exceedingly        23
15 deadly             21
16 safely             21
17 slowly             20
18 sufficiently       19
19 consistently       18
```

# 3-- Emoticons

- The rise of a very special set of text-as-picture

DataGenetics

Home | Blog | About Us | Work | Content | Contact Us | Save to Facebook

## Emoticon Analysis in Twitter

Probably the most often quoted words of Andy Warhol are *"In the future, everyone will be world-famous for 15 minutes".*

Were he alive today, I expect he might have said something like, *"In the future, everyone will be world-famous in 140 characters".*

Twitter is a phenomenon. Hundreds of millions of people broadcast their status for, anyone who cares, to read. Some messages are even read! The popular ones are rebroadcast to an even wider audience. Twitter has short-circuited the communication channels of our species.

The strict 140 character limit, imposed by Twitter, is a blessing and a curse. We are forced to abbreviate words, concepts and sentences. Prolific use is made of acronyms, creative contractions and the stylistic use of punctuation is common. Prominent in Tweets is the copious use of emoticons.


*"In the future, everyone will be world-famous ~~for 15 minutes~~ in 140 characters"*

### Emoticons

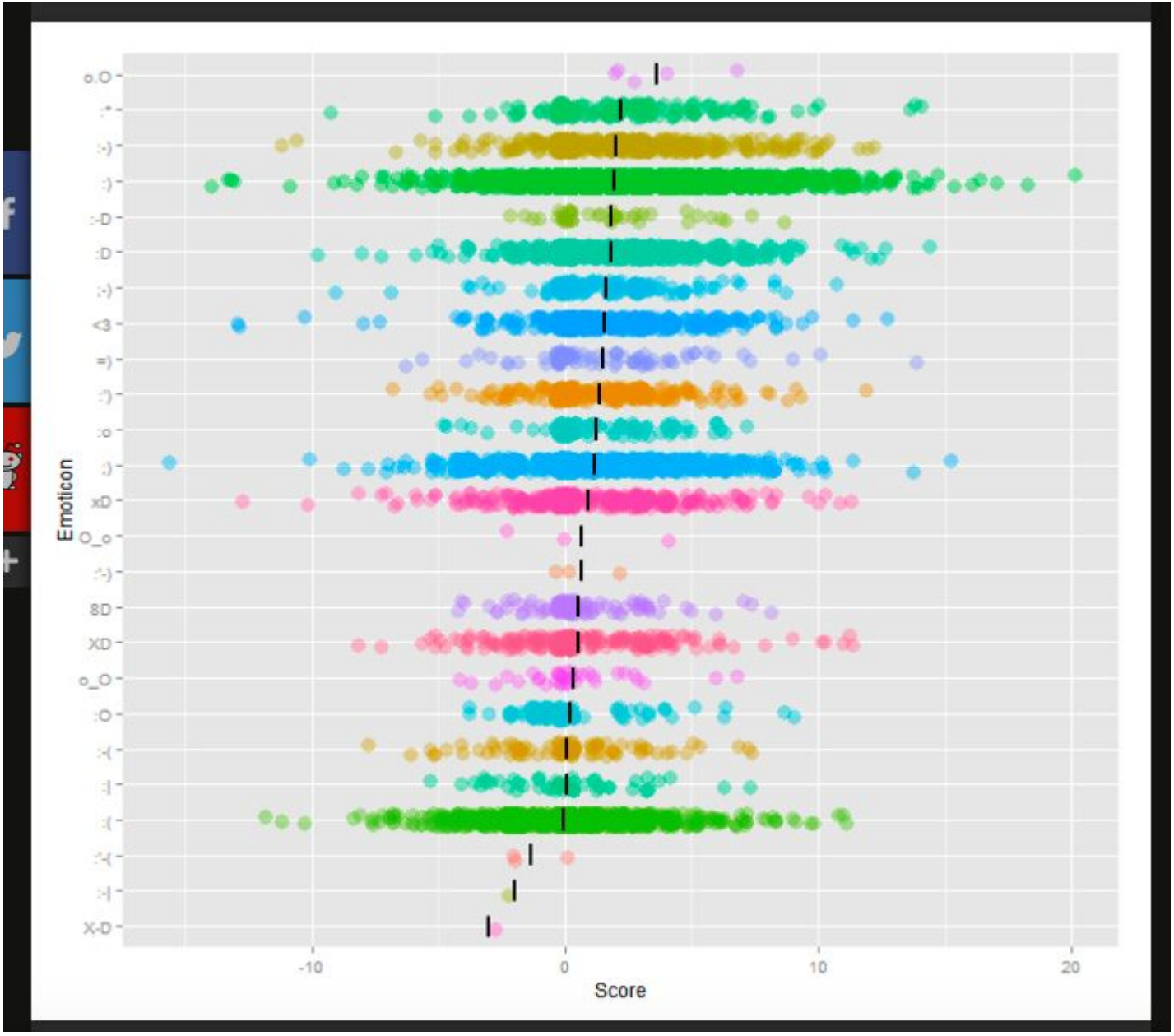
# Emoticons on Twitter

## The top emoticons

It's no big surprise, but the popular emoticons dominate the usage patterns. Of the **96,269,892** Tweets that contained emoticons, the top 20 smileys accounted for 90% of all occurences. Here they are:

| | Emoticon | Usage | Percent | Notes |
|---|---|---|---|---|
| #1 | :) | 32,115,789 | 33.360% | *Happy face* |
| #2 | :D | 10,595,385 | 11.006% | *Laugh* |
| #3 | :( | 7,613,014 | 7.908% | *Sad face* |
| #4 | ;) | 7,238,295 | 7.519% | *Wink* |
| #5 | :-) | 4,254,708 | 4.420% | *Happy face (with nose)* |
| #6 | :P | 3,588,863 | 3.728% | *Tongue out* |
| #7 | =) | 3,564,080 | 3.702% | *Happy face* |
| #8 | (: | 2,720,383 | 2.826% | *Happy face (mirror)* |
| #9 | ;-) | 2,085,015 | 2.166% | *Wink (with nose)* |
| #10 | :/ | 1,840,827 | 1.912% | *Uneasy, undecided, skeptical, annoyed?* |
| #11 | XD | 1,795,792 | 1.865% | *Big grin* |
| #12 | =D | 1,434,004 | 1.490% | *Laugh* |
| #13 | :o | 1,077,124 | 1.119% | *Shock, Yawn* |

# Emoticons on Twitter




From:

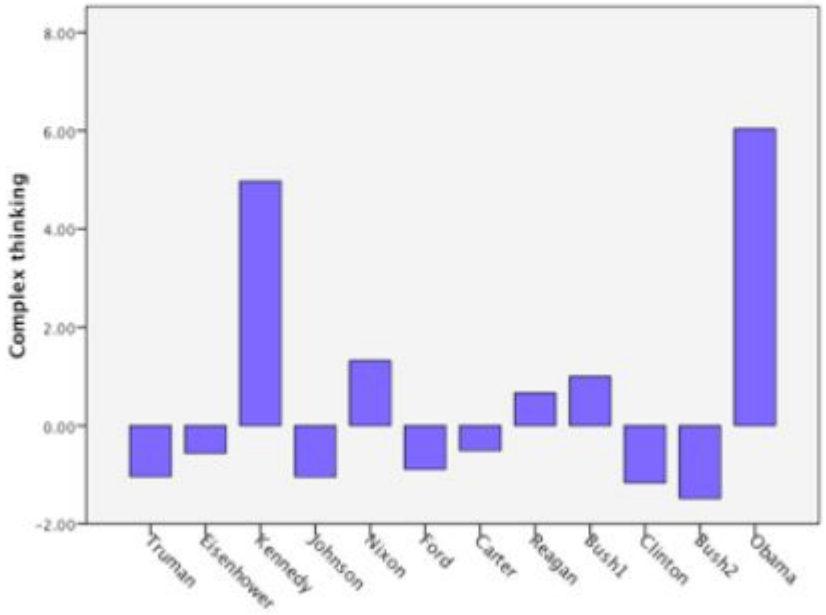
[The Happiest Emoticons]

*September 10, 2013*

By Eeshan Malhotra

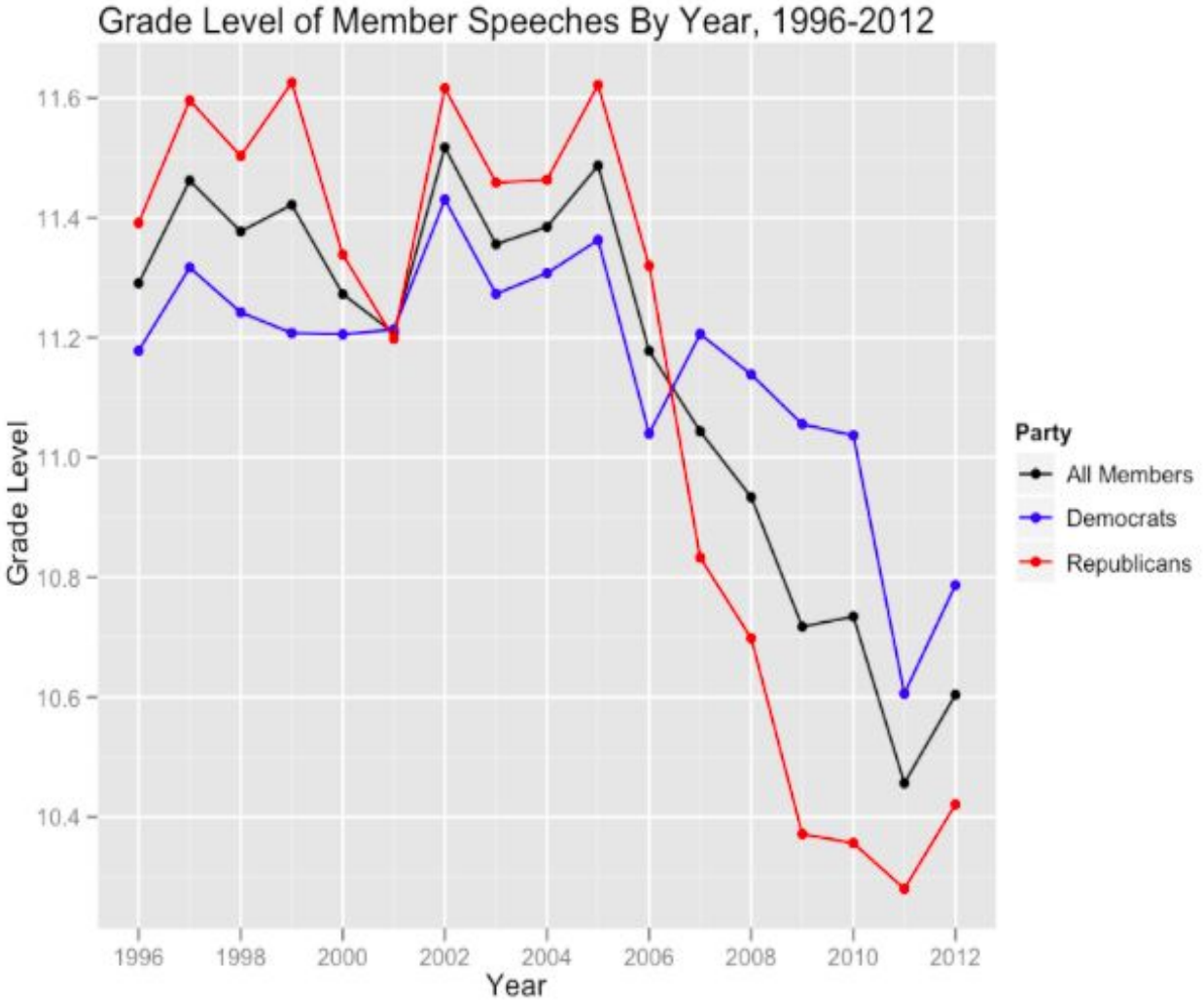
# 4a-- Complexity of thought

**Figure 3**. Complexity of thinking. The higher the number, the more complex and nuanced the language in the presentation of arguments.




- Obama was a law professor, you know

# 4a-- Another example of complexity of thought



# 4b-- Diversity of thought

- Ben Blatt also notes: “Looking at Meyer’s sentences, you get the sense that she may be using similar sentences again and again—a pitfall, perhaps, of writing a series of such length. To determine the relative repetitiveness of each author, I first made a list of the 50 most common three-word sentence openings each author used throughout her work.
- Then I calculated the percentage of all sentence openings (not necessarily the whole sentence) that come from each author's most used opening list:
  - J.K. Rowling: 2.35 percent
  - Suzanne Collins: 2.69 percent
  - Stephenie Meyer: 4.42 percent
- Repeatedly starting sentences off the same way doesn’t mean the prose is bad—it just means it’s repetitive. Hemingway uses the same 50 openings (“There was a,” “He did not”) in 5.3 percent of all sentences in The Sun Also Rises.”

# 4c-- Readability example

## Publishing while Female Gender Differences in Peer Review Scrutiny, by Erin Hengel

- Using five well-known “readability” tests, I analyse every article abstract published in the top four economics journals since 1950. I) Abstracts written by women are 1–6 percent more readable than those by men. II) The gap is up to three times higher in published articles than in earlier, draft versions of the same papers. III) Women’s writing gradually improves but men’s does not—meaning the readability gap grows over authors’ careers. I explore many interpretations; the simplest and most persuasive is that referees apply higher standards to women’s writing, subject- ing them to an added time tax. This last hypothesis is confirmed by submit-accept times at Econometrica: female-authored papers take six months longer to complete peer review.

# 4c-- Readability example

## Different scores

TABLE 2: Readability scores

| Score | Formula |
|---|---|
| Flesch Reading Ease | $206.835 - 1.015 \times AWS - 84.6 \times ASW$ |
| Flesch-Kincaid | $-15.59 + 0.390 \times AWS + 11.8 \times ASW$ |
| Gunning Fog | $0.4 \times AWS + 100 \times PWW$ |
| SMOG | $3.1291 + 5.7127 \times \sqrt{APS}$ |
| Dale-Chall | $3.6365 + 0.0496 \times AWS + 15.79 \times DWW$ |

*Notes.* $AWS$: average number of words per sentence; $ASW$: average number of syllables per word; $PWW$: ratio of polysyllabic words (3+ syllables) to word count; $APS$: average number of polysyllabic words per sentence; $DWW$: ratio of difficult words (not on Dale-Chall list) to word count.

# 4c-- Readability example

## A new Python module on readability

- “To transparently handle these issues and eliminate ambiguity in how the readability scores were calculated, I wrote the Python module Textatistic. Its code and detailed documentation is available at GitHub. A brief description is provided here.”

# 4c-- Readability example

## Women improve readability scores over time

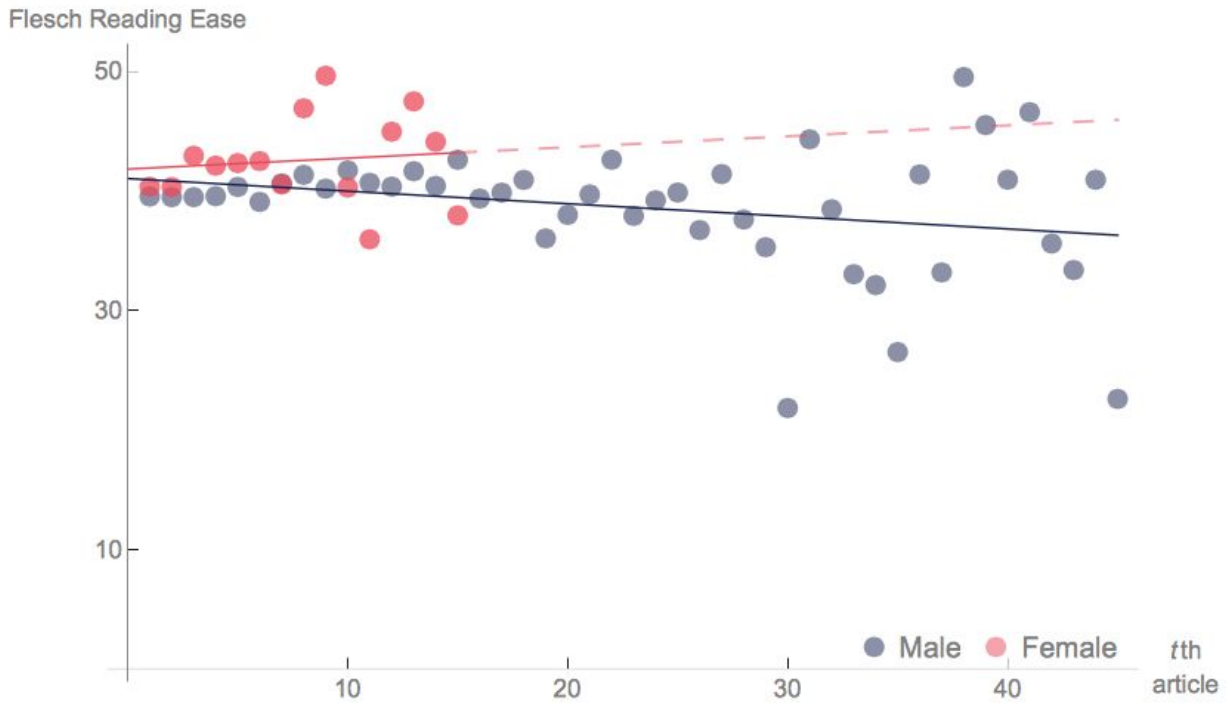



FIGURE 4: Readability of authors' *t*th publication

*Notes.* Mean Flesch Reading Ease scores grouped by authors' first, second, …, *t*th, … publication in the data. Lines of best fit are estimated separately for men and women on the grouped averages using OLS. Dotted line indicates out-of-sample forecast (the largest *t* for a woman is 15; for a man it's 45).

# 4d-- One more readability example

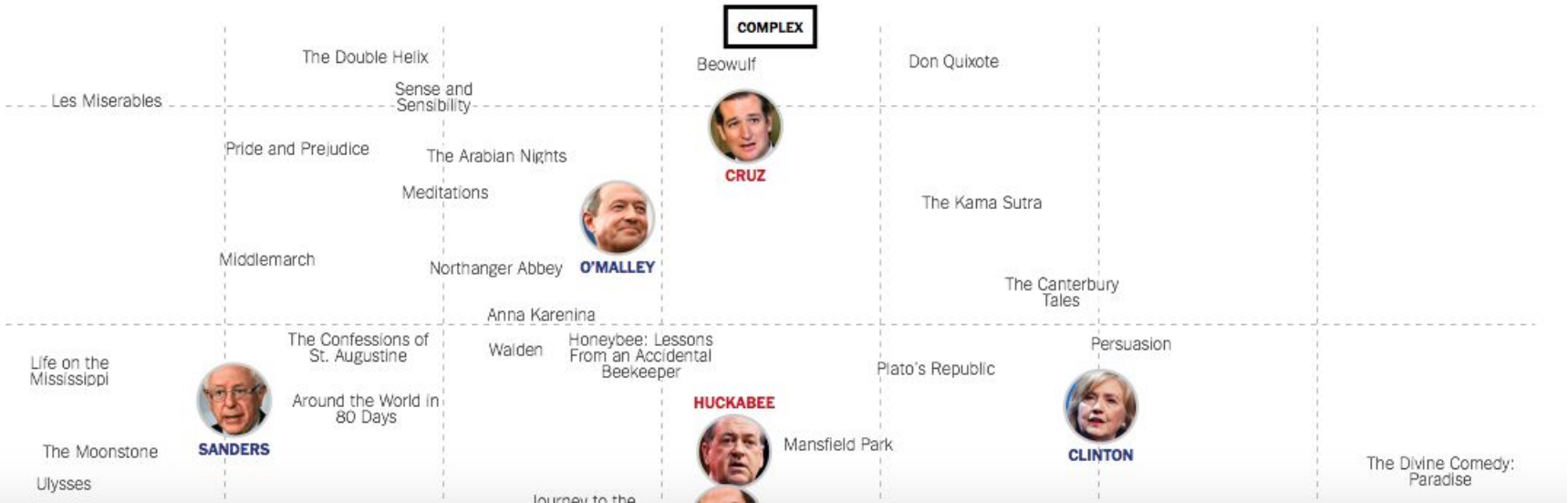


TheUpshot

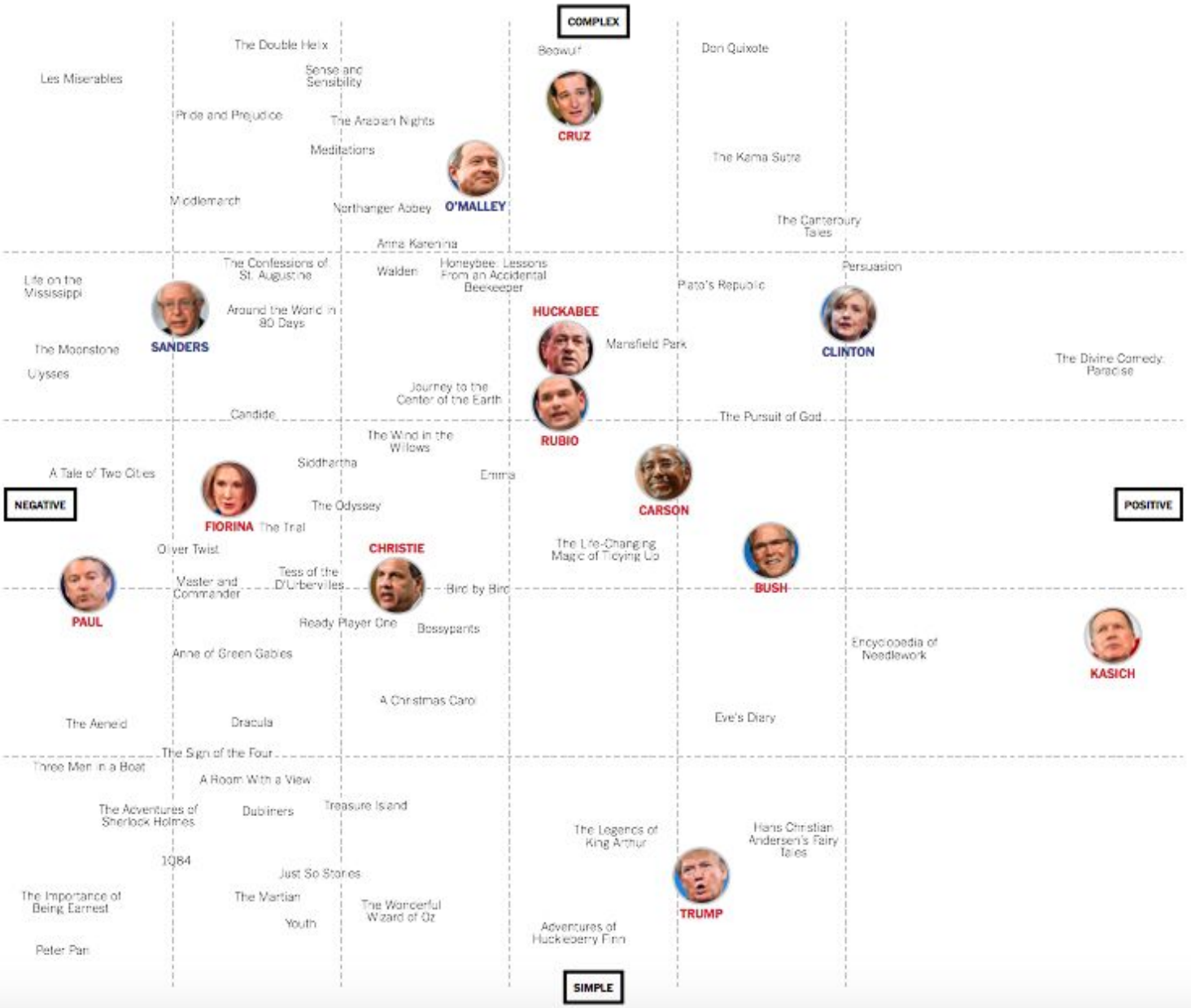


## Ted Cruz as Beowulf: Matching Candidates With the Books They Sound Like

By JOSH KATZ OCT. 28, 2015



# 4d-- One more readability example



# 2. Topic models

# What are topic models?

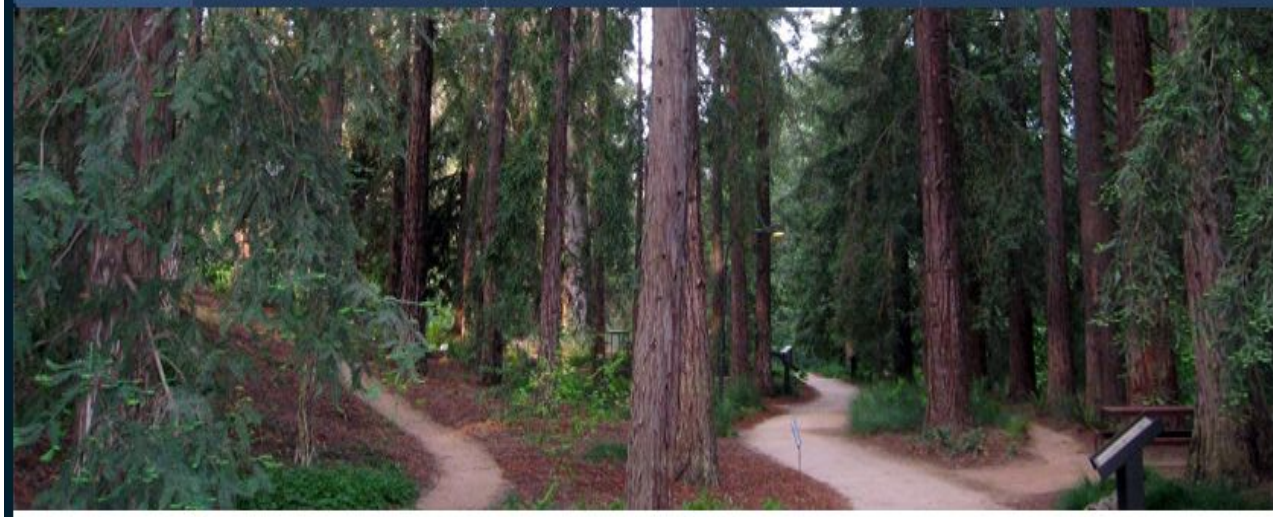
# Why study topic models?

# What are topic models?

# A simple example

# RTextTools: a machine learning library for text classification

Blog | About the Project | Install | How to Cite | Documentation



## Getting Started with Latent Dirichlet Allocation using RTextTools + topicmodels

8/30/2011 20 Comments

RTextTools bundles a host of functions for performing supervised learning on your data, but what about other methods like latent Dirichlet allocation? With some help from the topicmodels package, we can get started with LDA in just five steps. Text in green can be executed within R.

**Step 1: Install RTextTools + topicmodels**

We begin by installing and loading RTextTools and the topicmodels package into our R workspace.

```
install.packages(c("RTextTools","topicmodels"))
```

## Develop

Updates on RT
progress, tips,

## By Autho

All
Loren Collingwo
Timothy P. Jurk

## By Date

# Topic models of NYT headlines

1. Start with the RTextTools and topicmodels packages

```
install.packages(c("RTextTools","topicmodels"))
library(RTextTools)
library(topicmodels)
```

# Topic models of NYT headlines

2. Pull some data from within the RTextTools package, namely, a bunch of headlines from the NYT, say, a sample of 1000 headlines

```
data(NYTimes)
data <- NYTimes[sample(1:3100,size=1000,replace=FALSE),]
```

# Topic models of NYT headlines

3. Make a document-term-matrix (documents along the rows, with columns of words along the columns), stemmed and made lowercase

```
matrix <- create matrix(cbind(as.vector(data$Title),as.vector(data$Subject)),
language="english", removeNumbers=TRUE, stemWords=TRUE, weighting=weightTf)
```

# Topic models of NYT headlines

4. Run a DLA topic model, with 27(=k) topics assumed

```
k <- length(unique(data$Topic.Code))
lda <- LDA(matrix, k)
```

# Topic models of NYT headlines

## 5. Examine the results of the topic model solution

```
terms(lda,6)

nyt.topics = posterior(lda, matrix)$topics
df.nyt.topics = as.data.frame(nyt.topics)
df.nyt.topics = cbind(email=as.character(rownames(df.nyt.topics)),
                      df.nyt.topics, stringsAsFactors=F)

sample(which(df.nyt.topics$"1" > .6), 3)
sample(which(df.nyt.topics$"12" > .6), 3)

data$Title[[1]]
```

# Topic models of NYT headlines

The result of some of the code, around the most probable topic to be put into ...

```
> nyt.topics = posterior(lda, matrix)$topics
> df.nyt.topics = as.data.frame(nyt.topics)
> df.nyt.topics = cbind(email=as.character(rownames(df.nyt.topics)),
+                       df.nyt.topics, stringsAsFactors=F)

> sample(which(df.nyt.topics$"1" > .6), 3)
[1] 709 326 514
> sample(which(df.nyt.topics$"3" > .6), 3)
[1] 645 282 104
```

# Topic models of NYT headlines

## The top 6 terms associated with each topic

```
> terms(lda,6)
     Topic 1      Topic 2   Topic 3    Topic 4    Topic 5   Topic 6 Topic 7  Topic 8   Topic 9
[1,] "bush"       "aid"     "citi"     "american" "colleg"  "drug"  "hous"   "program" "school"
[2,] "elect"      "crisi"   "battl"    "nation"   "student" "day"   "budget" "new"     "china"
[3,] "presidenti" "korea"   "deal"     "iraqi"    "abus"    "new"   "show"   "mexico"  "republican"
[4,] "kerri"      "west"    "europ"    "forc"     "guilti"  "use"   "ban"    "iran"    "gop"
[5,] "judg"       "russia"  "busi"     "compani"  "immigr"  "guard" "hospit" "war"     "economi"
[6,] "propos"     "north"   "struggl"  "pataki"   "begin"   "call"  "end"    "job"     "seri"

     Topic 15   Topic 16   Topic 17  Topic 18   Topic 19 Topic 20  Topic 21 Topic 22   Topic 23
[1,] "plan"     "democrat" "chang"   "campaign" "cut"    "new"     "kill"   "state"    "use"
[2,] "militari" "parti"    "problem" "senat"    "tax"    "clinton" "bomb"   "death"    "baghdad"
[3,] "debat"    "effort"   "polit"   "terror"   "agre"   "york"    "talk"   "health"   "crime"
[4,] "big"      "critic"   "oil"     "race"     "ask"    "crash"   "peac"   "governor" "war"
[5,] "insur"    "town"     "saudi"   "mayor"    "gene"   "privat"  "india"  "die"      "olymp"
[6,] "loss"     "polit"    "base"    "fbi"      "dont"   "group"   "trial"  "ground"   "power"
```

# Topic models of NYT headlines

Topic 1 is about Bush election/administration, while Topic 3 is business

```
> data$Title[[709]]
[1] U.S. Troops Get a Warm 'Thank You' From President Bush

> data$Title[[326]]
[1] Britain Raises Barriers High Against the Asylum Seekers

> data$Title[[514]]
[1] THE ELECTION

> data$Title[[645]]
[1] At the Core of the Milky Way, The Brightest Star Ever Seen
```

```
> data$Title[[282]]
[1] THE MEDIA BUSINESS; European Officials Agree to Ban On Most Cigarette Ads by 2006

> data$Title[[104]]
[1] Messy Free-Market Plunge Rattling China's Businesses
```

# What are topic models?

# An intuitive explanation

# Edwin Chen

Hanging. MIT, MSR, Clarium, Twitter, Google, Dropbox.

Email
Twitter
Github
Google+
LinkedIn
Quora

Atom / RSS


## Recent Posts

Moving Beyond CTR: Better Recommendations Through

# Introduction to Latent Dirichlet Allocation

by Edwin Chen on Mon 22 August 2011

## Introduction

Suppose you have the following set of sentences:

- I like to eat broccoli and bananas.
- I ate a banana and spinach smoothie for breakfast.
- Chinchillas and kittens are cute.
- My sister adopted a kitten yesterday.
- Look at this cute hamster munching on a piece of broccoli.

What is latent Dirichlet allocation? It's a way of automatically discovering **topics** that these sentences contain. For example, given these sentences and asked for 2 topics, LDA might produce something like

- **Sentences 1 and 2**: 100% Topic A
- **Sentences 3 and 4**: 100% Topic B
- **Sentence 5**: 60% Topic A, 40% Topic B
- **Topic A**: 30% broccoli, 15% bananas, 10% breakfast, 10% munching, … (at which point, you could interpret topic A to be about food)
- **Topic B**: 20% chinchillas, 20% kittens, 20% cute, 15% hamster, … (at which point, you could interpret topic B to be about cute animals)

# Some sentences as “documents”

Suppose you have the following set of sentences:

- I like to eat broccoli and bananas.
- I ate a banana and spinach smoothie for breakfast.
- Chinchillas and kittens are cute.
- My sister adopted a kitten yesterday.
- Look at this cute hamster munching on a piece of broccoli.

# What we ask LDA to do

What is latent Dirichlet allocation? It's a way of automatically discovering **topics** that these sentences contain. If we were to find 2 topics in these sentences, LDA would say this:

# How to do LDA

- **Sentences 1 and 2**: 100% Topic A
- **Sentences 3 and 4**: 100% Topic B
- **Sentence 5**: 60% Topic A, 40% Topic B

---

- **Topic A**: 30% broccoli, 15% bananas, 10% breakfast, 10% munching, … (at which point, you could interpret topic A to be about food)
- **Topic B**: 20% chinchillas, 20% kittens, 20% cute, 15% hamster, … (at which point, you could interpret topic B to be about cute animals)

# How to do LDA

How does LDA perform this discovery?

# The writing process LDA assumes 1

LDA represents documents as **mixtures of topics** that spit out words with certain probabilities. It assumes that documents are produced in the following fashion: when writing each document, you

- Decide on the number of words N the document will have (say, according to a Poisson distribution).

# The writing process LDA assumes 2

- Then choose a topic mixture for the document (according to a Dirichlet distribution over a fixed set of K topics). For example, assuming that we have the two food and cute animal topics above, you might choose the document to consist of 1/3 food and 2/3 cute animals.

# The writing process LDA assumes 3

- Then generate each word w_i in the document by:
  - First picking a topic (according to the multinomial distribution that you sampled above; for example, you might pick the food topic with 1/3 probability and the cute animals topic with 2/3 probability).
  - Using the topic to generate the word itself (according to the topic's multinomial distribution). For example, if we selected the food topic, we might generate the word "broccoli" with 30% probability, "bananas" with 15% probability, and so on.

# The writing process LDA assumes

Assuming this generative model for a collection of documents, LDA then tries to backtrack from the documents to find a set of topics that are likely to have generated that collection.

# What we ask LDA to do

Let's make an example. According to the above process, when generating some particular document D, you might

- Pick 5 to be the number of words in D.
- Decide that D will be 1/2 about food and 1/2 about cute animals.
- Pick the first word to come from the food topic, which then gives you the word "broccoli".
- Pick the second word to come from the cute animals topic, which gives you "panda".
- Pick the third word to come from the cute animals topic, giving you "adorable".
- Pick the fourth word to come from the food topic, giving you "cherries".
- Pick the fifth word to come from the food topic, giving you "eating".

So the document generated under the LDA model will be "broccoli panda adorable cherries eating" (note that LDA is a bag-of-words model).

# The writing process LDA assumes

- So now suppose you have a set of documents. You've chosen some fixed number of K topics to discover, and want to use LDA to learn the topic representation of each document and the words associated to each topic.

- Go through each document, and randomly assign each word in the document to one of the K topics, and then improve on that until you'll eventually reach a roughly steady state.

- Use these assignments to estimate the topic mixtures of each document (by counting the proportion of words assigned to each topic within that document) and the words associated to each topic (by counting the proportion of words assigned to each topic overall).

# An analogy for LDA

Suppose you've just moved to a new city. You're a hipster and an anime fan, so you want to know where the other hipsters and anime fans tend to hang out. Of course, as a hipster, you know you can't just *ask*, so what do you do?

Here's the scenario: you scope out a bunch of different establishments (**documents**) across town, making note of the people (**words**) hanging out in each of them (e.g., Alice hangs out at the mall and at the park, Bob hangs out at the movie theater and the park, and so on). Crucially, you don't know the typical interest groups (**topics**) of each establishment, nor do you know the different interests of each person.

# An analogy, continued

So you pick some number K of categories to learn (i.e., you want to learn the K most important kinds of categories people fall into), and start by making a guess as to why you see people where you do.

For example, you initially guess that Alice is at the mall because people with interests in X like to hang out there; when you see her at the park, you guess it's because her friends with interests in Y like to hang out there; when you see Bob at the movie theater, you randomly guess it's because the Z people in this city really like to watch movies; and so on.

# An analogy, continued

Of course, your random guesses are very likely to be incorrect (they're random guesses, after all!), so you want to improve on them. One way of doing so is to:

- Pick a place and a person (e.g., Alice at the mall).
- Why is Alice likely to be at the mall? Probably because other people at the mall with the same interests sent her a message telling her to come.
- In other words, the more people with interests in X there are at the mall and the stronger Alice is associated with interest X (at all the other places she goes to), the more likely it is that Alice is at the mall because of interest X.
- So make a new guess as to why Alice is at the mall, choosing an interest with some probability according to how likely you think it is.

# An analogy, continued

Go through each place and person over and over again. Your guesses keep getting better and better (after all, if you notice that lots of good students hang out at the bookstore, and you suspect that Alice is pretty good student herself, then it's a good bet that Alice is at the bookstore because her good student friends told her to go there; and now that you have a better idea of why Alice is probably at the bookstore, you can use this knowledge in turn to improve your guesses as to why everyone else is where they are), and eventually you can stop updating.

# An analogy, continued

Then take a snapshot (or multiple snapshots) of your guesses, and use it to get all the information you want:

- For each category, you can count the people assigned to that category to figure out what people have this particular interest. By looking at the people themselves, you can interpret the category as well (e.g., if category X contains lots of tall people wearing jerseys and carrying around basketballs, you might interpret X as the “basketball players” group).

# An analogy, continued

Then take a snapshot (or multiple snapshots) of your guesses, and use it to get all the information you want:

- For each place P and interest category C, you can compute the proportions of people at P because of C (under the current set of assignments), and these give you a representation of P. For example, you might learn that the people who hang out at Barnes & Noble consist of 10% hipsters, 50% anime fans, 10% sports, and 30% college students.

# Chen's example

LDA applied to Sarah Palin's emails:

**Trig/Family/Inspiration**: family, web, mail, god, son, from, congratulations, children, life, child, down, trig, baby, birth, love, you, syndrome, very, special, bless, old, husband, years, thank, best, …

**Wildlife/BP Corrosion**: game, fish, moose, wildlife, hunting, bears, polar, bear, subsistence, management, area, board, hunt, wolves, control, department, year, use, wolf, habitat, hunters, caribou, program, denby, fishing, …

**Energy/Fuel/Oil/Mining:** energy, fuel, costs, oil, alaskans, prices, cost, nome, now, high, being, home, public, power, mine, crisis, price, resource, need, community, fairbanks, rebate, use, mining, villages, …

# Chen's example

LDA applied to Sarah Palin's emails, continued:

**Gas**: gas, oil, pipeline, agia, project, natural, north, producers, companies, tax, company, energy, development, slope, production, resources, line, gasline, transcanada, said, billion, plan, administration, million, industry, …

**Education/Waste**: school, waste, education, students, schools, million, read, email, market, policy, student, year, high, news, states, program, first, report, business, management, bulletin, information, reports, 2008, quarter, …

**Presidential Campaign/Elections**: mail, web, from, thank, you, box, mccain, sarah, very, good, great, john, hope, president, sincerely, wasilla, work, keep, make, add, family, republican, support, doing, p.o, …

# A Trig/Family/Inspiration email

Here's an example of an email which fell 99% into the Trig/Family/Inspiration category (particularly representative words are highlighted in blue):

Hello Governor Palin, Our **family** wanted to congratulate **you** and your **family** on the **birth** of your **son**, **Trig**. Our fourth **child**, Daniel, was **born** with **Down Syndrome**, and we can't imagine our **family** without him. Recently, I met a mom with a 34-year-old **daughter** with DS and she said it best: "Don't **you** feel like you've been chosen to be a member of a **very special** club?" **God** bless your **family**, what a **beautiful** example of **love** you are to all who see you! the Paul & Tricia Pietig **family**, Des Moines, Iowa

# A mixed topic email

And here's an excerpt from an email which fell 10% into the Presidential Campaign/Election category (in red) and 90% into the Wildlife/BP Corrosion category (in green):

We understand that you have been discussed as a possible choice for the **Vice Presidency**.

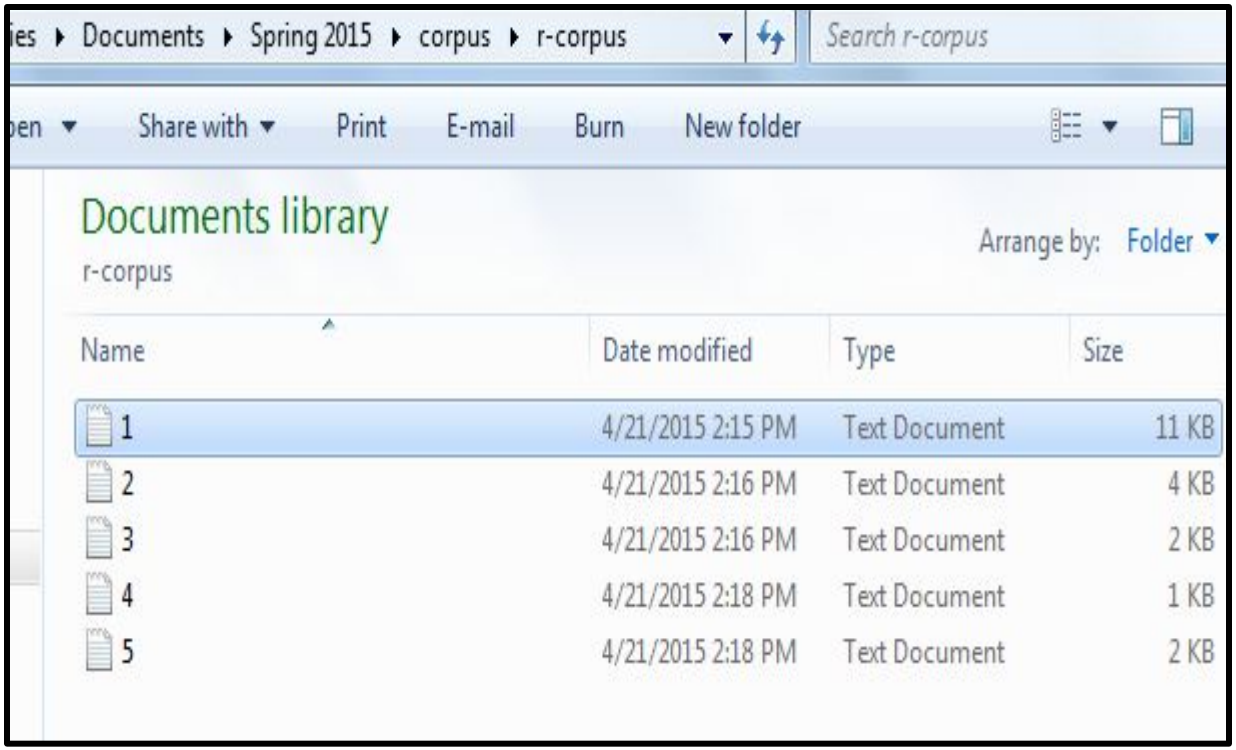
As **people** who **support** the democratic process and care about protecting our **wildlife** for future generations, we want **you** to know that we don't believe **people** in our states would vote for **you** for any office if they knew your record on these issues.

It is troubling that **you** are **now** working to deny more than 50,000 Alaskans a vote on **aerial** killing of **wolves** and **bears** with legislation now **being** considered in the Alaska legislature.

# Another example

# Topic models of Amazon reviews

```
Sys.setenv(NOAWT=TRUE)
##install.packages("tm")
library(tm)
my.corpus <- Corpus(DirSource("C:/Users/gme2101/Documents/Spring
2015/corpus/r-corpus"))  ## just a set of documents like this: ##
```



# Topic models of Amazon reviews

```
my.corpus <- tm map(my.corpus, removePunctuation)
my.corpus <- tm_map(my.corpus, removeWords, stopwords("english"))

my.tdm <- TermDocumentMatrix(my.corpus)
inspect(my.tdm)

##install.packages("Snowball")
library(Snowball)
my.corpus <- tm_map(my.corpus, stemDocument)

##my.dtm <- DocumentTermMatrix(my.corpus, control = list(weighting = weightTf,
stopwords = TRUE))
##inspect(my.dtm)
```

# Topic models of Amazon reviews

```
findFreqTerms(my.tdm, 2)

findAssocs(my.tdm, 'my', 0.20)

my.df <- as.data.frame(inspect(my.tdm))
my.df.scale <- scale(my.df)
d <- dist(my.df.scale,method="euclidean")
fit <- hclust(d, method="ward")
plot(fit)

install.packages("topicmodels")
library(topicmodels)
lda.model = LDA(my.dtm, 2)

terms(lda.model,10)
```

# Amazon reviews ...

## Review #1

```
> my.corpus[[1]]
<<PlainTextDocument (metadata: 7)>>
An Android User Review   iPhon 6 Read     think  switch  Android
By Jay  Januari 28 2015
Color Name GraySiz Name 16 GB
So I made  switch  use Android   iPhon back  October  Ive  use  iPhon 6   past  month
now  can give  detail review    like  switch  Befor  switch Ive use  Samsung Galaxi S2
first smartphon ever  also  Nexus 4 Sinc Im  tech enthusiast Im well vers   play
around  mani  Android devic includ   big name Galaxi S5 HTC One M8  M7 One Plus One
Here   thought

Thing   iPhon  realli well  hardwar  softwarewis
1 Camera The behind  scene softwar  digit captur  imag  definit  strongest sell
iPhon Other   S5  Note 4  smartphon realli come close     kind  imag qualiti  matter
megapixel compar   iPhon This  one   reason    switch  sinc Ive start  dabbl
photographi  want  realli good camera   smartphon Side note   read  lot  tech blog
notion    near futur  smartphon wont accur describ  devic anymor sinc make  phone call
probabl one   least common use featur   smartphon   look   averag user Camera social
media email  take  higher usag
```

# Amazon reviews ...

## Review #5

```
> my.corpus[[5]]
<<PlainTextDocument (metadata: 7)>>
I realli like  phone
By Ken M  Juli 9 2014
Color Name Blue Verifi Purchas
So far I realli like  phone Its  first real smart phone I resist get one  mani year
bit   learn experi      appear    job  well If youv got     latest stuff I dont think
phone   allaround simpl communic  sure seem    job Its also small enough  overcom one
argument   smart phone size  weight This   slight larger   basic phone  I  held
mani year yet    mani  thing quit easili The camera  good  front  rear face   flash
avail   front face one  selfi one Batteri life  just OK   bad  smart phone standard
Im told Im  light user  I get two day    charg although one   pay attent  shut
function  your  use   data  your use WiFi   revers  well All   Id certain recommend
phone  someon like   doesnt    let alon know   use   bell  whistl     The price  right
one get plenti  bang   buck   phone
>
```

# Which words go with which topics

A lot of overlap here, but Topic 1 is dominated by “app,” while Topic 2 is dominated by the term phone itself

```
> terms(lda.model,10)
      Topic 1   Topic 2
 [1,] "app"     "phone"
 [2,] "android" "iphon"
 [3,] "use"     "like"
 [4,] "iphon"   "android"
 [5,] "one"     "ive"
 [6,] "can"     "use"
 [7,] "phone"   "one"
 [8,] "devic"   "get"
 [9,] "ios"     "app"
[10,] "realli"  "much"
```

# Amazon reviews ...

Which reviews go with which topics?

```
> df.emails.topics
      email            1            2
1.txt 1.txt 0.9999413959 5.860408e-05
2.txt 2.txt 0.0001729985 9.998270e-01
3.txt 3.txt 0.9996323854 3.676146e-04
4.txt 4.txt 0.0007892080 9.992108e-01
5.txt 5.txt 0.7275410094 2.724590e-01
```

# Another example

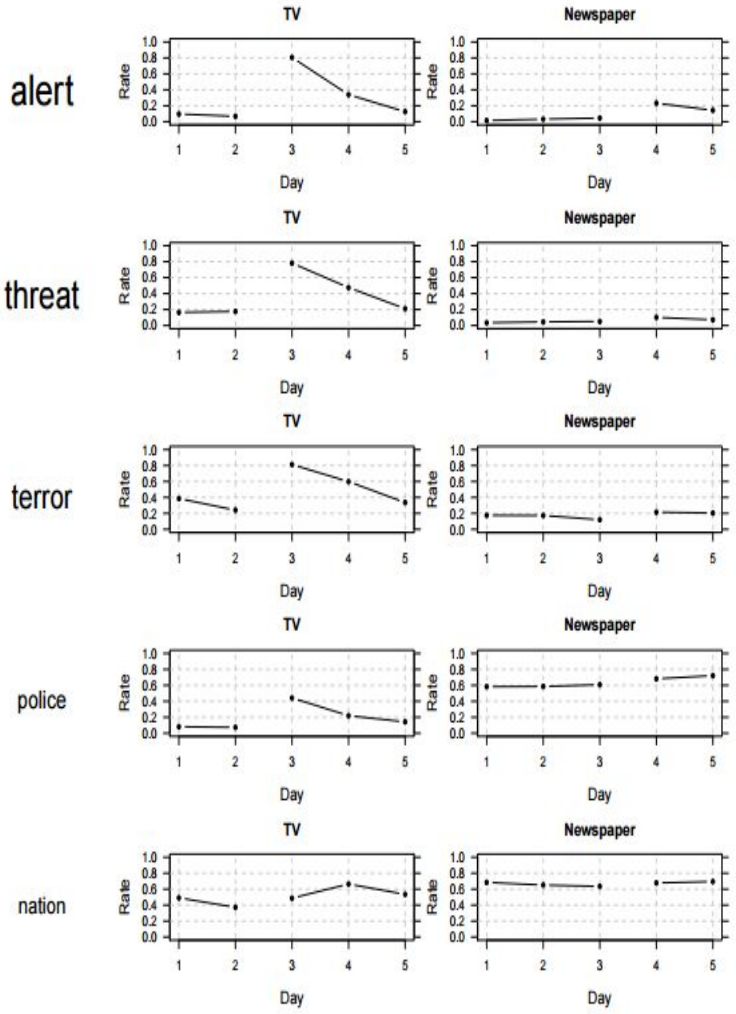
# Topics shrinking and growing ...

- Bonilla, Tabitha, and Justin Grimmer. "Elevated threat levels and decreased expectations: How democracy handles terrorist threats." *Poetics* 41.6 (2013): 650-669.

- To more credibly estimate the effects of terror threats, they exploit elevations of the U.S. government's color coded alert system. Using this design, a statistical model for texts and a new collection of news stories, they show that media outlets allocate substantially more attention to terrorism after an alert.

# Some descriptive results

Figure 1: Changes in Word Rates Around Terror Alerts



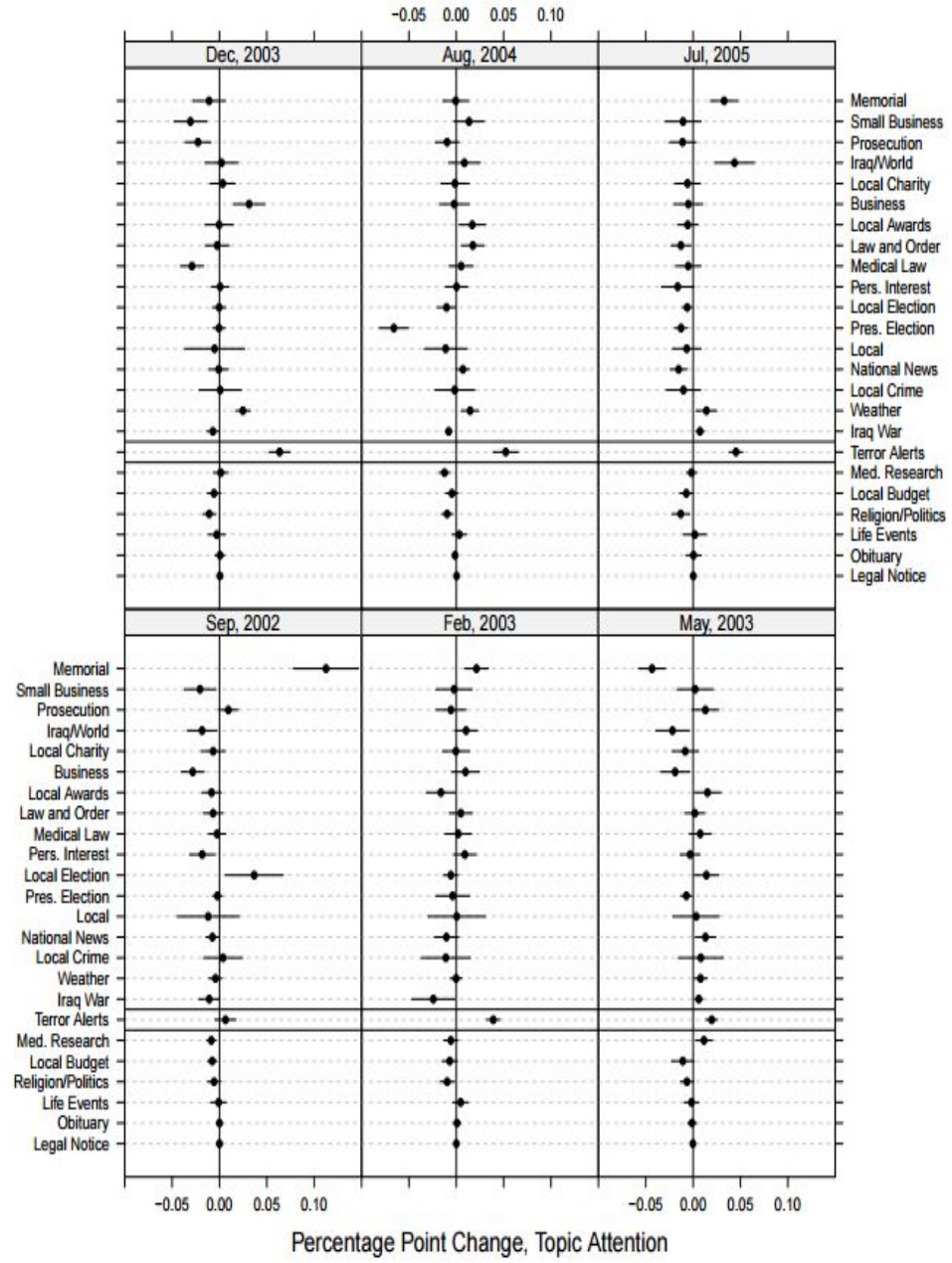

This figure shows that words related to terrorism are used much more often after an alert than before. The left-hand column shows the words with usage rates that change most drastically after an alert. The middle column shows how the word usage rates change for television news, while the right hand column shows

# Topics in the media

Table 2: Topics of News Stories and Evening News Broadcasts

| Label | Discriminating Stems | % |
|---|---|---|
| Memorial | peopl,time,famili,work,world,live,home,like | 11.28 |
| Local Small Business | citi,plan,project,000,develop,counti,properti,million | 7.83 |
| Criminal Prosecution | polic,offic,arrest,old,sentenc,prison,prosecutor,told | 6.96 |
| Iraq/World | al,attack,iraq,offici,palestinian,bomb,american,peopl | 6.86 |
| Local Philanthropy | section,load,date,type,photo,newspap,english,music | 6.83 |
| Business | compani,busi,market,million,price,peopl,work,time | 6.40 |
| Local Awards | educ,board,teacher,budget,fund,colleg,high,million | 4.92 |
| Law and Order | polic,man,car,arrest,offic,vehicl,old,accid | 4.64 |
| Medical Law | compani,law,drug,feder,lawsuit,million,worker,medic | 4.38 |
| Personal Interest Stories | team,game,player,run,win,sport,season,second | 4.04 |
| State/Local Elections | vote,candid,voter,democrat,campaign,race,counti,ballot | 3.93 |
| 2004 President Campaign | bush,nation,convent,columbia,presid,space,war,democrat | 3.79 |
| Local | school,center,group,inform,denver,program,island,week | 3.67 |
| National Political News | bush,law,senat,rule,feder,govern,commiss,court | 3.55 |
| Local Crime | school,town,island,polic,committe,charg,hous,case | 3.28 |
| Weather | hurrican,storm,wind,rain,flood,area,damag,beach | 3.22 |
| Iraq War | iraq,bush,unit,war,saddam,nuclear,powel,nation | 2.75 |
| Terror Alerts | alert,attack,offici,terrorist,terror,al,homeland,threat | 2.68 |
| Medical Research | hospit,drug,studi,research,percent,dr,care,medic | 2.45 |
| Local Budgeting | budget,million,citi,increas,fund,revenu,fiscal,cost | 2.14 |
| Religion and Politics | church,law,appeal,rule,court,marriag,sentenc,order | 1.98 |
| Life Events | funer,church,home,memori,son,daughter,sister | 1.67 |
| Obituary | funer,late,servic,burial,sister,wife,born,niec | 0.64 |
| Legal Notice | record,hear,island,probat,sold,certifi,clerk,notic | 0.09 |

Figure 2: Terror Alerts Shift Attention to Terrorism, Though the Shift Varies Across Alerts




This figure shows the within-outlet change in attention to topics that occurs in response to alerts. This

# Results

- The change in the amount of writing in the terrorism topic after an alert

# Another example

# LDA in arts funding articles

- DiMaggio, Paul, Manish Nag, and David Blei. "Exploiting affinities between topic modeling and the sociological perspective on culture: Application to newspaper coverage of US government arts funding." *Poetics* 41.6 (2013): 570-606.

- They use Latent Dirichlet Allocation (LDA) to analyze how one policy domain, government assistance to artists and arts organizations, was framed in almost 8000 articles. These comprised all articles that referred to government support for the arts in the U.S. published in five U.S. newspapers between 1986 and 1997—a period during which such assistance, once noncontroversial, became a focus of contention.

# Some benefits of LDA

- Many topics may be viewed as *frames* (semantic contexts that prime particular associations or interpretations of a phenomenon in a reader) and employed accordingly

- Another particular strength of topic modeling is its ability to capture polysemy and disambiguate different uses of a term, based on the context (other terms) in which it appears. In their emphasis on *relationality*, topic models capture the insight, shared by linguistics and much cultural sociology, that meanings emerge out of relations rather than residing within words. Thus many terms may appear in more than one topic within a given corpus of documents.

# Some benefits of LDA

- A third virtue of topic modeling is its deep affinity to the central insight in the sociology of culture that texts do not necessarily reflect a singular perspective but are often characterized by *heteroglossia*, the copresence of competing "voices"—perspectives or styles of expression—within a single text.

- Blei (2012, p. 78) writes that the fundamental "intuition behind LDA is that documents exhibit multiple topics." The results that LDA produces can be useful in examining heteroglossia empirically.

# 3. How do I get text again?

# 4. Another way to get at semantics: Word to vector models (for next time)