EC999: Vector Space Representation

Thiemo Fetzer

University of Chicago & University of Warwick

April 20, 2017

Quantitative Text Analysis as process

Above all, there needs to be a formulatedresearch question or **goal** to be achieved.

When I presented the supplementary budget to this House last April, I said we could work our way through this period of severe economic distress. Today, I can report that notwithstanding the difficulties of the past eight months, we are now on the road to economic recovery.

In this next phase of the Government's plan we must stabilise the deficit in a fair way, safeguard those worst hit by the recession, and stimulate crucial sectors of our economy to sustain and create jobs. The worst is over.

This Government has the moral authority and the well-grounded optimism rather than the cynicism of the Opposition. It has the imagination to create the new jobs in energy, agriculture, transport and construction that this green budget will

docs \ words	made	because	had	into	get	some	through	next	where	many	irish
t06_kenny_fg	12	11	5	4	8	4	3	4	5	7	10
t05_cowen_ff	9	4	8	5	5	5	14	13	4	9	8
t14_ocaolain_sf	3	3	3	4	7	3	7	2	3	5	6
t01_lenihan_ff	12	1	5	4	2	11	9	16	14	6	9
t11_gormley_green	0	0	0	3	0	2	0	3	1	1	2
t04_morgan_sf	11	8	7	15	8	19	6	5	3	6	6
t12_ryan_green	2	2	3	7	0	3	0	1	6	0	0
t10_quinn_lab	1	4	4	2	8	4	1	0	1	2	0
t07_odonnell_fg	5	4	2	1	5	0	1	1	0	3	0
t09_higgins_lab	2	2	5	4	0	1	0	0	2	0	0
t03_burton_lab	4	8	12	10	5	5	4	5	8	15	8
t13_cuffe_green	1	2	0	0	11	0	16	3	0	3	1
t08_gilmore_lab	4	8	7	4	3	6	4	5	1	2	11
t02_bruton_fg	1	10	6	4	4	3	0	6	16	5	3

Descriptive statistics on words

Scaling documents

Classifying documents

Extraction of topics

Vocabulary analysis

Sentiment analysis

Bag of Words Language Model

- For most of what we will do, we will represent documents as vectors of word frequency counts.
- This is called the bag of words language model, as the order in which terms appear is disregarded.
- This implies grammatical structure is disregarded.
- Central to this language model is the representation of text as vectors of weighted word counts combined into document term matrices (dtm's) or their transpose tdm's.

Sparsity revisited

- As we noted, Zipfs Law and Heap's law imply an exploding vocabulary space the more text is added.
- Storing large matrices in memory is an issue - it simply becomes not feasible.
- Thats why most text packages in R work with a construct called *sparse* matrix.
- Sparse matrices are arranged as *triplets* consisting of three arrays (A,B,C).
  - A contains all of the nonzero entries reading top to bottom one column after the other
  - B contains indices of/pointers to A indicating where each new column begins
  - C contains the row index of each element in A.
- Most statistical packages for machine learning/ text analysis in R support sparse matrices.

Sparse Matrix Storage efficiency vs assignment inefficiency

```r
library('Matrix')

m1 <- matrix(0, nrow = 1000, ncol = 100)
m2 <- Matrix(0, nrow = 1000, ncol = 100, sparse = TRUE)

#storage efficiency
object.size(m1)
```

```
## 800200 bytes
```

```r
object.size(m2)
```

```
## 1824 bytes
```

```r
#assignment can take more time
system.time(m1[, 1:10] <- 1)
```

```
##    user  system elapsed
##   0.001   0.000   0.001
```

```r
system.time(m2[, 1:10] <- 1)
```

```
##    user  system elapsed
##   0.008   0.000   0.009
```

Building a document-term matrix

- In the quanteda package, building a dfm is easy.

```
myCorpus <- corpus_subset(data_corpus_inaugural, Year > 1990)
#stemming, stopword removal
myStemMat <- dfm(myCorpus, remove = stopwords("english"), stem = TRUE, removePunct = TRUE)
myStemMat[, 1:5]
```

```
## Document-feature matrix of: 7 documents, 5 features (17.1% sparse).
## 7 x 5 sparse Matrix of class "dfmSparse"
##               features
## docs            fellow citizen today celebr mysteri
##   1993-Clinton       5       2    10      4       1
##   1997-Clinton       7       8     6      1       0
##   2001-Bush          1      10     2      0       0
##   2005-Bush          3       7     3      2       0
##   2009-Obama         1       1     6      2       0
##   2013-Obama         3       8     6      1       0
##   2017-Trump         1       4     5      3       1
```

```
#top features
topfeatures(myStemMat, 20)
```

```
##     will  america       us   nation american     must      new    world    peopl     time
##      161      103      100       84       79       68       67       63       60       54
##    everi  freedom      can  citizen     work  countri    today      one   govern      now
##       52       48       48       40       39       39       38       38       36       34
```

Building a document-term matrix

- In the tm package, building a dfm is easy as well.

```
library(tm)
reut21578 <- system.file("texts", "crude", package = "tm")
reuters <- VCorpus(DirSource(reut21578), readerControl = list(reader = readReut21578XMLasPlain))
reuters

## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 20

##tm_map function to manipulate corpus / dfm matrix objects
reuters <- tm_map(reuters, removeWords, stopwords("english"))
reuters <-tm_map(reuters, stemDocument)
dtm <- DocumentTermMatrix(reuters)
inspect(dtm[5:10, 740:743])

## <<DocumentTermMatrix (documents: 6, terms: 4)>>
## Non-/sparse entries: 3/21
## Sparsity           : 88%
## Maximal term length: 6
## Weighting          : term frequency (tf)
##
##      Terms
## Docs  polici polit popul port
##   211      0     0     0    0
##   236      1     0     0    0
##   237      0     1     1    0
##   242      0     0     0    0
##   246      0     0     0    0
##   248      0     0     0    0

findFreqTerms(dtm, lowfreq=10)

##  [1] "accord" "barrel" "bpd"    "crude"  "dlrs"   "kuwait" "last"   "market" "meet"
## [10] "mln"    "new"    "offici" "oil"    "one"    "opec"   "pct"    "price"  "reuter"
## [19] "said"   "said."  "saudi"  "sheikh" "the"    "u.s."   "will"
```

Working with DTMs / DFMs

- In the coming weeks, we will mostly work with DTMs/ DFMs as main data objects.
- Thus we can think of moving beyond the initial focus, which was on short fragments of text, and start to discuss how we can treat documents represented as vectors.
- At the end of the day, our $\mathbf{X}$ document term matrices are just data matrices that you would analyze using statistical methods, such as regression techniques.
- Specific nature of text data means that we can not translate methods one to one.
- We start by defining concept of measuring distance in high dimensional vector spaces

Distance between Texts?

- The idea is that (weighted) features form a vector for each document, and that these vectors can be judged using metrics of similarity.
- Most often, you want to know how similar or dissimilar text is from one another.
- So we need to have a metric to capture distance.
- A documents vector for us is simply (for us) the row of the document-feature matrix

Characteristics of similarity measures

Let A and B be any two documents in a set and $d(A, B)$ be the distance between A and B.

1. $d(A, B) \geq 0$ (the distance between any two points must be non-negative)
2. $d(A, B) = 0$ iff $A = B$ (the distance between two documents must be zero if and only if the two objects are identical)
3. $d(A, B) = d(B, A)$ (distance must be symmetric: A to B is the same distance as from B to A)
4. $d(A, C) \leq d(A, B) + d(B, C)$ (the measure must satisfy the triangle inequality)

Euclidian Distance between two vectors

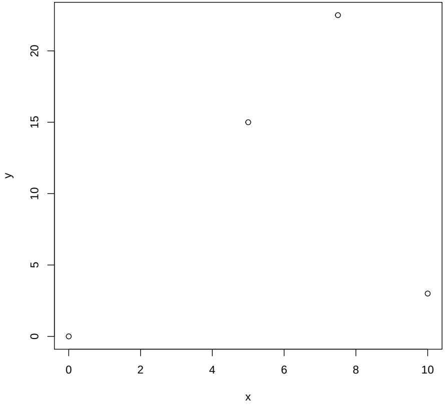
Euclidian distance is defined as

$$d^{Euclidian}(A, B) = \sqrt{\sum_{j=1}^{p}(y_{Aj} - y_{Bj})^2}$$

where p is the set of distinct words (the number of columns in our document-term matrix).
Another common notation is

$$\|\mathbf{y_A} - \mathbf{y_B}\|$$

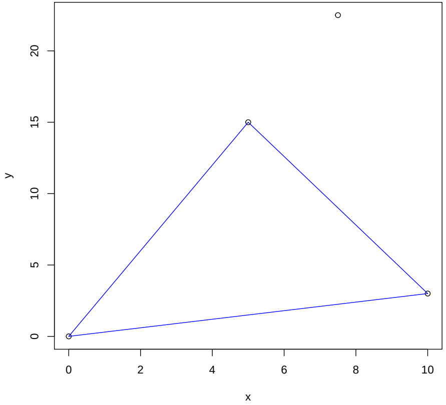
Euclidian Distance between two vectors



- Euclidian distance $\|\mathbf{y_A} - \mathbf{y_B}\| = 13$
- Transformed $\|\mathbf{1.5y_A} - \mathbf{y_B}\| = 19.6596$

  $\Rightarrow$ What does that imply for textual vectors?

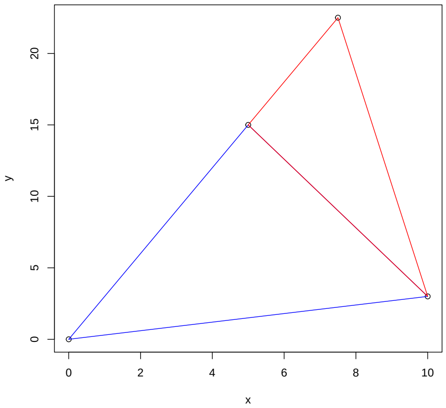
Euclidian Distance between two vectors



- Euclidian distance $\|\mathbf{y_A} - \mathbf{y_B}\| = 13$
- Transformed $\|\mathbf{1.5y_A} - \mathbf{y_B}\| = 19.6596$

  $\Rightarrow$ What does that imply for textual vectors?

Euclidian Distance between two vectors



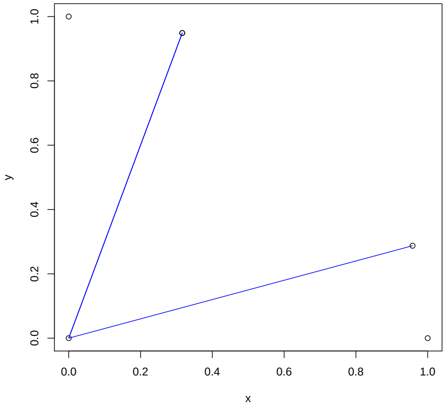
- Euclidian distance $\|\mathbf{y_A} - \mathbf{y_B}\| = 13$
- Transformed $\|\mathbf{1.5y_A} - \mathbf{y_B}\| = 19.6596$

  $\Rightarrow$ What does that imply for textual vectors?

Issue with Euclidian Distance

- Textual data may consist of documents that are *very similar* in their usage of vocabular, but could have different length.
- In the extreme case, the Euclidian distance between two identical documents where one document just repeates all words would be large.
- Euclidian distance does not measure degree of *linear dependence*.
- So $d(A, 2B)$ will be much larger than $d(A, B)$, even though the angle between the vectors stay the same.
- Similarly, $d(2A, 2B)$ will have same angle, but their Euclidian distance will be twice as long as distance $d(A, B)$
- A large Euclidian distance could be due to documents having different length, not because they are using different vocabulary.
- So what can be done? Lets normalize the length of the vector to 1.

Cosine Similarity: Measuring Angle between two unit lenght vectors



Cosine Similarity: Measuring Angle between two unit length vectors

- What is the length of a vector A and B? its simply the Euclidian distance from origin, i.e. $\|\mathbf{y_A}\|$,$\|\mathbf{y_B}\|$
- So the vectors $\mathbf{y'_A} = \frac{\mathbf{y_A}}{\|\mathbf{y_A}\|}$ and $\mathbf{y'_B} = \frac{\mathbf{y_B}}{\|\mathbf{y_B}\|}$ both have length 1.
- What is the angle between the vectors $\frac{\mathbf{y_A}}{\|\mathbf{y_A}\|}$ and $\frac{\mathbf{y_B}}{\|\mathbf{y_B}\|}$?

$$\cos\left(\mathbf{y_A}, \mathbf{y_B}\right) = \frac{\mathbf{y_A} \cdot \mathbf{y_B}}{\|\mathbf{y_A}\|\|\mathbf{y_B}\|} = \frac{\sum_{i=1}^{n} y_{iA} y_{iB}}{\sqrt{\sum_{i=1}^{n} y_{iA}^2}\sqrt{\sum_{i=1}^{n} y_{iB}^2}}$$

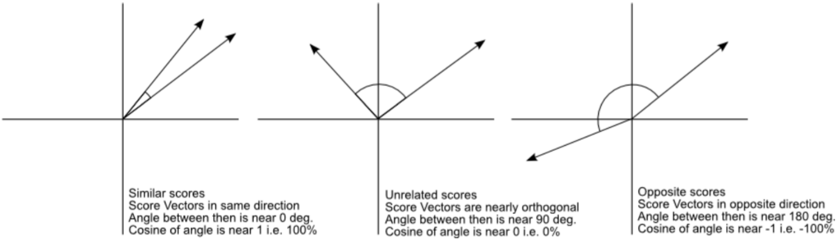
Cosine Similarity: Relationship to Euclidian Distance

$$\|\tilde{\mathbf{y}}_{\mathbf{A}} - \tilde{\mathbf{y}}_{\mathbf{B}}\|^2 = (\tilde{\mathbf{y}}_{\mathbf{A}} - \tilde{\mathbf{y}}_{\mathbf{B}})^{'}(\tilde{\mathbf{y}}_{\mathbf{A}} - \tilde{\mathbf{y}}_{\mathbf{B}}) = \|\tilde{\mathbf{y}}_{\mathbf{A}}\|^2 + \|\tilde{\mathbf{y}}_{\mathbf{B}}\|^2 - 2\tilde{\mathbf{y}}_{\mathbf{A}}^{'}\tilde{\mathbf{y}}_{\mathbf{B}}$$

Note that the normalization of the vectors $\tilde{\mathbf{y}}_{\mathbf{A}}, \tilde{\mathbf{y}}_{\mathbf{B}}$ to length one imply that

$$\|\tilde{\mathbf{y}}_{\mathbf{A}} - \tilde{\mathbf{y}}_{\mathbf{B}}\|^2 = (\tilde{\mathbf{y}}_{\mathbf{A}} - \tilde{\mathbf{y}}_{\mathbf{B}})^{'}(\tilde{\mathbf{y}}_{\mathbf{A}} - \tilde{\mathbf{y}}_{\mathbf{B}}) = 2(1 - \tilde{\mathbf{y}}_{\mathbf{A}}^{'}\tilde{\mathbf{y}}_{\mathbf{B}}) = 2(1 - \cos(\tilde{y}_A, \tilde{y}_B))$$

Cosine Similarity Examples



So Cosine similarity ranges from -1.0 to 1.0 for term frequencies; or 0 to 1.0 for normalized term frequencies (or tf-idf) - why?

Cosine (dis) similarity

- When introducing the dot product, we introduced the idea of angles between vectors as a measure of linear dependence.
- For two vectors x and x', the angle was given as

$$\theta = \cos^{-1}\left(\frac{\langle x_i, x_{i'} \rangle}{||x_i||_2||x_{i'}||_2}\right)$$

- We can define a dissimilarity function as

$$1 - \cos\left(\frac{\langle x_i, x_{i'} \rangle}{||x_i||_2||x_{i'}||_2}\right)$$

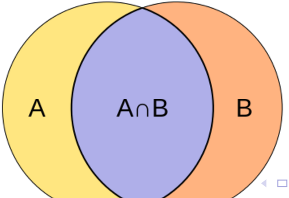
- We saw how this measure behaves with perfectly positively correlated vectors.
- Cosine similarity is widely used in text clustering because two documents with the same proportions of term occurrences but different lengths are often considered identical.

Binary Jaccard Dissimilarity

- Jaccard Similarity is the simplest of the similarities and is nothing more than a combination of binary operations of set algebra.
- To calculate the Jaccard Distance or similarity is treat our document as a set of tokens.
- Formally:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

- For example, given two sets' binary indicator vectors $\mathbf{y}_A = (0,1,1,0)^\dagger$ and $\mathbf{y}_B = (1,1,0,0)^\dagger$, the cardinality of their intersect is 1 and the cardinality of their union is 3, rendering their Jaccard coefficient $1/3$.



Extended Jaccard Distance

$$J(\mathbf{y_A}, \mathbf{y_B}) = \frac{\mathbf{y_A'}\mathbf{y_B}}{\|\mathbf{y_A}\|^2 + \|\mathbf{y_B}\|^2 - \mathbf{y_A'}\mathbf{y_B}}$$

The extended Jaccard coefficient allows elements of vectors $\mathbf{y}_A$ and $\mathbf{y}_B$ to be arbitrary positive real numbers. This coefficient captures a *vector-length-sensitive measure of similarity*.

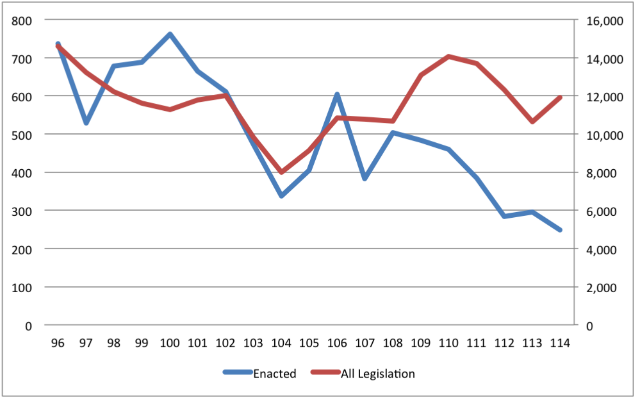
However, it is scale invariant:

$$J(\mathbf{2y_A}, \mathbf{2y_B}) = J(\mathbf{y_A}, \mathbf{y_B})$$

But not length invariant:

$$J(\mathbf{2y_A}, \mathbf{y_B}) \neq J(\mathbf{y_A}, \mathbf{y_B})$$

Sample Application: Is there a decline in legislative output?



Number of enacted bills across different congresses, starting from 1979 to 2016. There is a declining trend in the number of bills being passed, while the total bills considered stayed reasonably stable.

Mapping Legislative Influence or Productivity

- Most proposed bills do never make it into actual law.
- It seems that over time, the number of bills that get passed has been going down.
- However, the actually passed bills may still contain a lot of information from bills that did not pass.
- Consider the following examples for the US:
  - H.R. 1060 (105th): Pharmacy Compounding Act
  - S. 830 (105th): Food and Drug Administration Modernization Act of 1997

Mapping Legislative Influence

S. 830 (105th): Food and Drug Administration Modernization Act of 1997

Introduced: **Jun 5, 1997**
105th Congress, 1997–1998

Status: **Enacted — Signed by the President** on **Nov 21, 1997**
This bill was enacted after being signed by the President on November 21, 1997.

Law: Pub.L. 105-115

Sponsor:

James "Jim" Jeffords
Senator from Vermont
Republican

Text:

Read Text »
Last Updated: Nov 9, 1997
Length: 85 pages

Mapping Legislative Influence

H.R. 1060 (105th): Pharmacy Compounding Act

Introduced: **Mar 13, 1997**
105th Congress, 1997–1998

Status: **Died in a previous Congress**
This bill was introduced on March 13, 1997, in a previous session of Congress, but was not enacted.

Sponsor:

Richard Burr
Representative for North Carolina's 5th congressional district
Republican

Text:


Read Text »
Last Updated: Mar 13, 1997
Length: 4 pages

Mapping Legislative Influence

SEC. 2. APPLICATION OF FEDERAL LAW TO THE PRACTICE OF PHARMACY COMPOUNDING.

(a) IN GENERAL- Section 503 (21 U.S.C. 353) is amended by adding at the end the following:

> '(h)(1) Sections 501(a)(2)(B), 501(f), 501(h), 502(f)(1), 502(l), 502(o), 502(s), 502(t), 505, and sections 510 through 520 shall not apply to a drug or device that is compounded by a licensed pharmacist or licensed physician or other licensed practitioner authorized by State law to prescribe drugs or devices or both--
>
> '(A) on the order of such a licensed physician or other licensed practitioner for an individual patient; or
>
> '(B) in limited quantities, as determined by the principal State agency of jurisdiction which regulates the practice of pharmacy for that pharmacist, before receiving a valid order for an individual patient if the compounding of the drug or device is based on a history of receiving valid orders that have been generated solely within an established relationship between the pharmacist, and (i) the patient for whom the order will be given, or (ii) the physician or other licensed practitioner who will write such order.

Such sections shall not apply to a drug or device if such pharmacist or physician or other licensed practitio does no more than advertise or otherwise promote the compounding service and does not advertise or otherwise promote the compounding of a particular drug or device.

'SEC. 503A. PHARMACY COMPOUNDING.

'(a) IN GENERAL- Sections 501(a)(2)(B), 502(f)(1), and 505 shall not apply to a drug product if the drug product is compounded for an identified individual patient based on the unsolicited receipt of a valid prescription order or a notation, approved by the prescribing practitioner, on the prescription order that a compounded product is necessary for the identified patient, if the drug product meets the requirements of this section, and if the compounding--

'(1) is by--

'(A) a licensed pharmacist in a State licensed pharmacy or a Federal facility, or

'(B) a licensed physician,

on the prescription order for such individual patient made by a licensed physician or other licensed practitioner authorized by State law to prescribe drugs; or

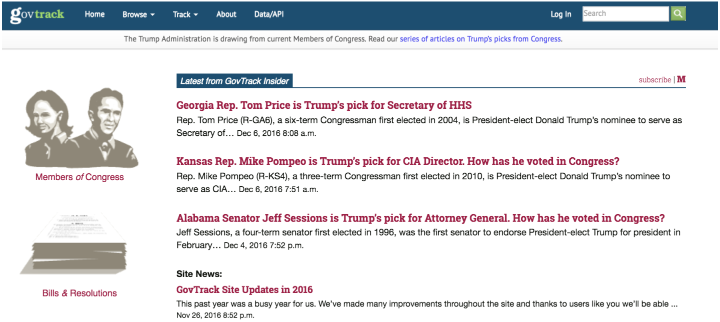
'(2)(A) is by a licensed pharmacist or licensed physician in limited quantities before the receipt of a valid prescription order for such individual patient; and

'(B) is based on a history of the licensed pharmacist or licensed physician receiving valid prescription orders for the compounding of the drug product, which orders have been generated solely within an established relationship between--

Can Cosine Similarity be used to identify other "intellectual owners"?

- The idea here is that bills that are enacted are combinations of bills that have been introduced by a range of politicians, the vast majority of which never got enacted or passed.
- Build two different corpora:
  1. texts of all bills that were introduced in a congress
  2. texts of all bills that were enacted
- Perform cosine similarity analysis at the bill level, the "section" or "paragraph" level.

Introducing govtrack.us



govtrack.us provides an **API** as well as *bulk data downloads*.

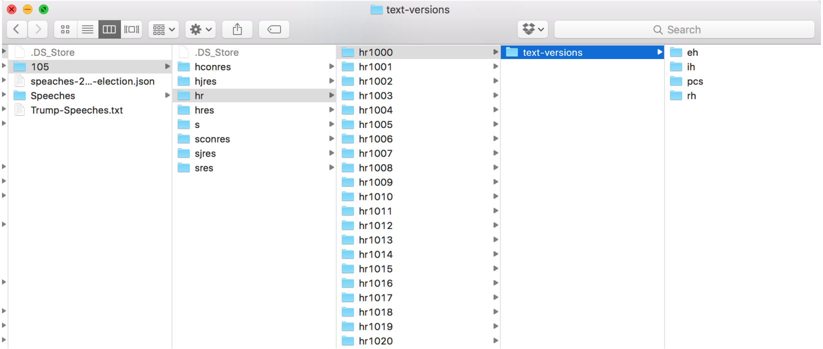
Bulk downloading legislative text versions

- You can browse the data structure here: `https://www.govtrack.us/data/`
- Bills go through multiple stages: IS - Introduced in Senate, IH - Introduced in House to being an Enrolled Bill (ENR)
- Bulk download is possible
- E.g. using rsync on Mac/ *nix computers or cwrsync (`https://www.itefix.net/cwrsync` on windows)
- Alternatively could download using HTTP, but they dont like that.

Bulk downloading legislative text versions

Downloading all bill versions of the 105th congress - roughly 13k documents. On a Mac/*nix machine just type in Terminal:

```
rsync -avz --include='*.txt' --include='*/' --exclude='*'
govtrack.us::govtrackdata/congress/105/bills/hr/
/Users/...
```



Plain Text Files

```
[Congressional Bills 105th Congress]
[From the U.S. Government Printing Office]
[H.R. 1000 Introduced in House (IH)]





105th CONGRESS
  1st Session
                                H. R. 1000

To require States to establish a system to prevent prisoners from being
     considered part of any household for purposes of determining
eligibility of the household for food stamp benefits and the amount of
  food stamp benefits to be provided to the household under the Food
                         Stamp Act of 1977.


_______________________________________________________________________


                    IN THE HOUSE OF REPRESENTATIVES

                             March 10, 1997

  Mr. Goodlatte (for himself, Mr. Smith of Oregon, and Mr. Stenholm)
 introduced the following bill; which was referred to the Committee on
                              Agriculture

_______________________________________________________________________

                                 A BILL


To require States to establish a system to prevent prisoners from being
     considered part of any household for purposes of determining
eligibility of the household for food stamp benefits and the amount of
  food stamp benefits to be provided to the household under the Food
                         Stamp Act of 1977.

    Be it enacted by the Senate and House of Representatives of the
United States of America in Congress assembled,

SECTION 1. STATES REQUIRED TO ESTABLISH SYSTEM TO PREVENT PRISONERS
              FROM BEING CONSIDERED PART OF ANY HOUSEHOLD UNDER THE
              FOOD STAMP ACT OF 1977.
```

Building a corpus

```
# devtools::install_github('kbenoit/readtext') library(readtext) TEXT <-
# readtext(list.files(path = '../../Data/105', pattern = '*.txt', full.names = TRUE,
# recursive = TRUE))
CORP <- corpus(TEXT, docnames = list.files(path = "../../Data/105", pattern = "*.txt", full.nheadames = FALSE
    recursive = TRUE))
docvars(CORP)[["id"]] <- docnames(CORP)
docvars(CORP)[["bill"]] <- gsub("([a-z]+)/([a-z]+[0-9]+)/text-versions/([a-z]+)/document.txt$",
    "\\2", docnames(CORP))
docvars(CORP)[["version"]] <- gsub("([a-z]+)/([a-z]+[0-9]+)/text-versions/([a-z]+)/document.txt$",
    "\\3", docnames(CORP))
docvars(CORP)[["doctype"]] <- gsub("([a-z]+)/([a-z]+[0-9]+)/text-versions/([a-z]+)/document.txt$",
    "\\1", docnames(CORP))
docvars(CORP)[["congress"]] <- 106
# preserve introduced and engrossed
CORP <- subset(CORP, version %in% c("ih", "enr"))
CORP.dfm <- dfm(CORP, ignoredFeatures = c("will", stopwords("english")), stem = TRUE)

# cosine similarity computation
SIMS <- similarity(CORP.dfm, "hr/hr1060/text-versions/ih/document.txt", margin = "documents",
    method = "cosine")[[1]]
```

Distribution of Cosine Similarity Across Whole Corpus

```
plot(density(SIMS), main = "Cosine Dissimilarity") + abline(v = SIMS[["s/s830/text-versions/enr/document.txt"]
    col = "red")
```

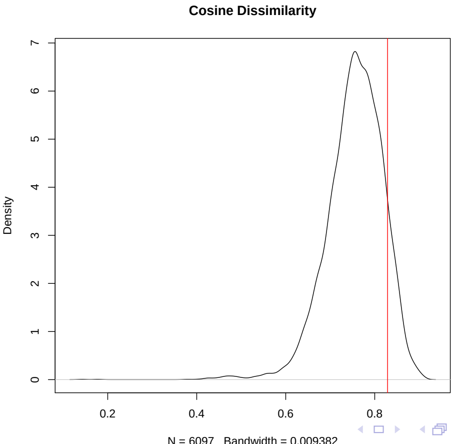
```
## numeric(0)
```

```
SIMS[["s/s830/text-versions/enr/document.txt"]]
```

```
## [1] 0.829
```

```
## many docs with similar score
sum(SIMS >= SIMS[["s/s830/text-versions/enr/document.txt"]])
```

```
## [1] 626
```



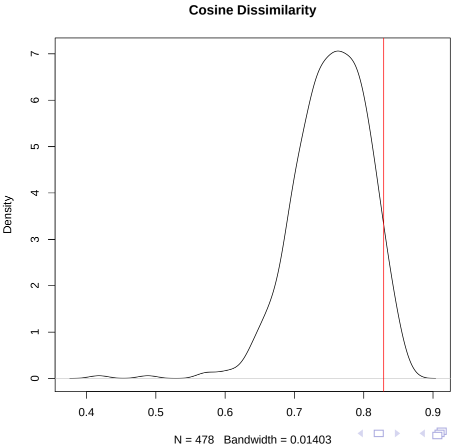
Distribution of Cosine Similarity Across Corpus of Enacted Bills

```
plot(density(SIMS[grep("/enr/", names(SIMS))]), main = "Cosine Dissimilarity") + abline(v = SIMS[["s/s830/text
    col = "red")

## numeric(0)

## many docs with similar score
sum(SIMS[grep("/enr/", names(SIMS))] >= SIMS[["s/s830/text-versions/enr/document.txt"]])

## [1] 24
```



Refinement?

- We see that cosine similarity is able to detect significant overlap between the much smaller docment HR1060 and the much longer document S830.
- Can refine this a bit further by chunking text into paragraphs and remove very short section titles

```
CORP.PARA <- changeunits(CORP, to = "paragraphs")
```

- Alternative segmenting using the `segment` function
- Reducing the unit of analysis to capture individual bill sections may improve performance but can blow up dimensionality, so may be best to proceed iteratively.

Extended Jaccard Similarity

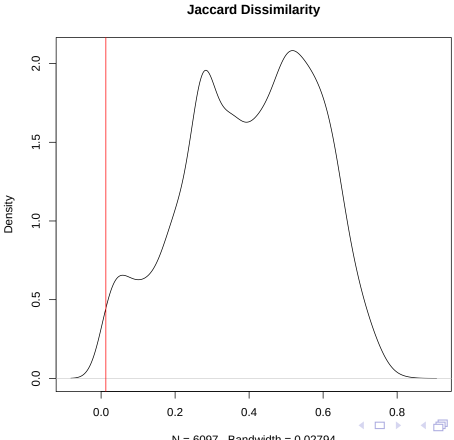
```
SIMS <- similarity(CORP.dfm, "hr/hr1060/text-versions/ih/document.txt", margin = "documents",
    method = "eJaccard")[[1]]

plot(density(SIMS), main = "Jaccard Dissimilarity") + abline(v = SIMS[["s/s830/text-versions/enr/document.txt"
    col = "red")

## numeric(0)

## many docs with similar score
sum(SIMS >= SIMS[["s/s830/text-versions/enr/document.txt"]])

## [1] 6061
```



Lots of other distance metrics

```r
library(proxy)
```

```
##
## Attaching package:  'proxy'
## The following object is masked from 'package:Matrix':
##
##     as.matrix
## The following objects are masked from 'package:stats':
##
##     as.dist, dist
## The following object is masked from 'package:base':
##
##     as.matrix
```

```r
lapply(pr_DB$get_entries(), function(x) x$names)
```

```
## $Jaccard
## [1] "Jaccard" "binary"  "Reyssac" "Roux"
##
## $Kulczynski1
## [1] "Kulczynski1"
##
## $Kulczynski2
## [1] "Kulczynski2"
##
## $Mountford
## [1] "Mountford"
##
## $Fager
## [1] "Fager"   "McGowan"
##
## $Russel
## [1] "Russel" "Rao"
##
## $`simple matching`
## [1] "simple matching" "Sokal/Michener"
##
## $Hamman
```