Diving Deeper into Text Analysis in R

Alison Blaine, Erica Hayes & Markus Wust
NCSU Libraries

workspace: go.ncsu.edu/textr
slides: go.ncsu.edu/textrslides

Welcome!

Learning environment

- Questions are welcomed
- Requests to repeat/re-explain are welcomed
- Collaborators in learning - help your neighbor if you can
- We get imposter syndrome too

Goals

Learn & implement some R functions for preprocessing text data

Learn and implement some text analysis techniques using R

Know how to seek help & resources in the future

DON'T PANIC

Text Analysis

Train computers to find similarities between texts, extract themes, and analyze expressed sentiments

Reading: Human - Machine

Human Reader: Con - Pro

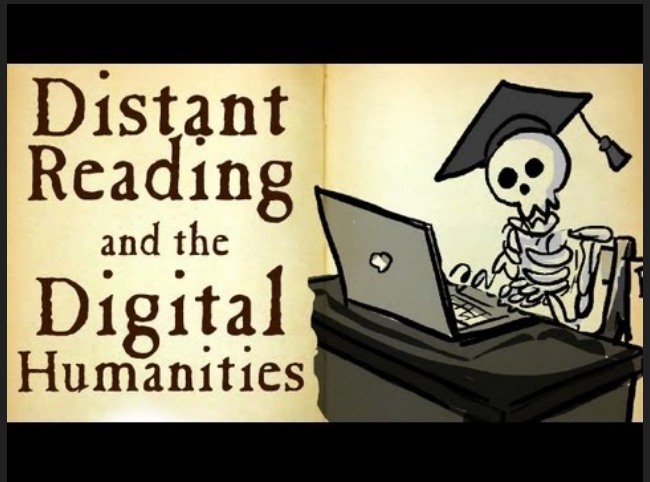
- Slow
- Limited memory
- Limited factual background knowledge
- Limited linguistic knowledge

- Can understand "meaning"
- Understand linguistic nuances and tone (e.g., sarcasm), allusions, emotions, etc.

Machine Reader: Con - Pro

- So far no actual comprehension
- Usually no deeper understanding
- No understanding of linguistic nuances, emotions, etc.

- Fast
- (Almost) unlimited memory
- (Almost) instant access to supplementary information



Text Analysis: What Is It Good For?

- Computerized text analysis can help you get an overview of large text corpora (from tweets to novels)
- Indicate general trends
- Extract data

On the other hand:

- For a more in-depth analysis of a text, you still have to read it yourself
- Text analysis less suited for poetic language

R for Text Analysis

Popularity of R

[source]

August 07, 2018

IEEE Language Rankings 2018

Python retains its top spot in the fifth annual IEEE Spectrum top programming language rankings, and also gains a designation as an "embedded language". Data science language R remains the only domain-specific slot in the top 10 (where it as listed as an "enterprise language") and drops one place compared to its 2017 ranking to take the #7 spot.

Looking at other data-oriented languages, Matlab as at #11 (up 3 places), SQL is at #24 (down 1), Julia at #32 (down 1) and SAS at #40 (down 3). Click the screenshot below for an interactive version of the chart where you can also explore the top 50 rankings.

Language Rank	Types	Spectrum Ranking
1. Python		100.0
2. C++		98.4
3. C		98.2
4. Java		97.5
5. C#		89.8
6. PHP		85.4
7. R		83.3
8. JavaScript		82.8
9. Go		76.7
10. Assembly		74.5

The IEEE Spectrum rankings are based on search, social media, and job listing trends, GitHub repositories, and mentions in journal articles. You can find details on the ranking methodology here, and discussion of the trends behind the 2018 rankings at the link below.

IEEE Spectrum: The 2018 Top Programming Languages

Why is R so popular?

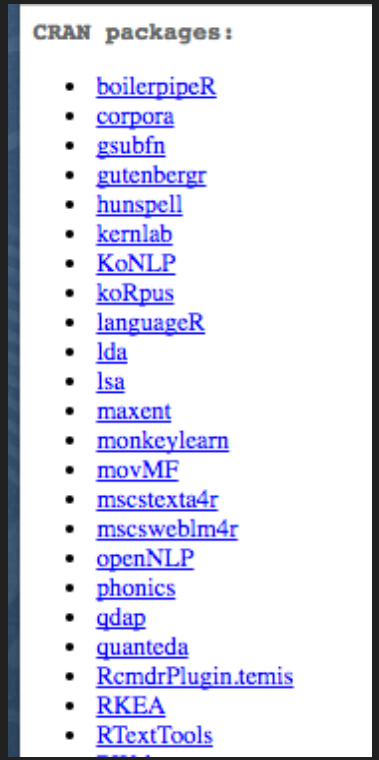
- Versatility
- Over 16,000 Packages
- RStudio
- tidyverse

The tidyverse

Components

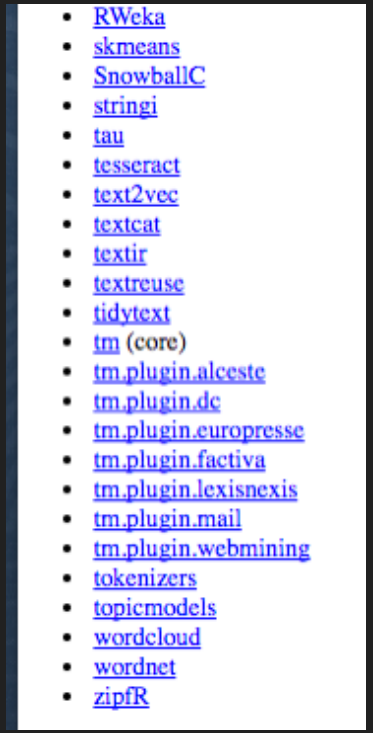
R Packages for Text Analysis

There are more than this -- see link!



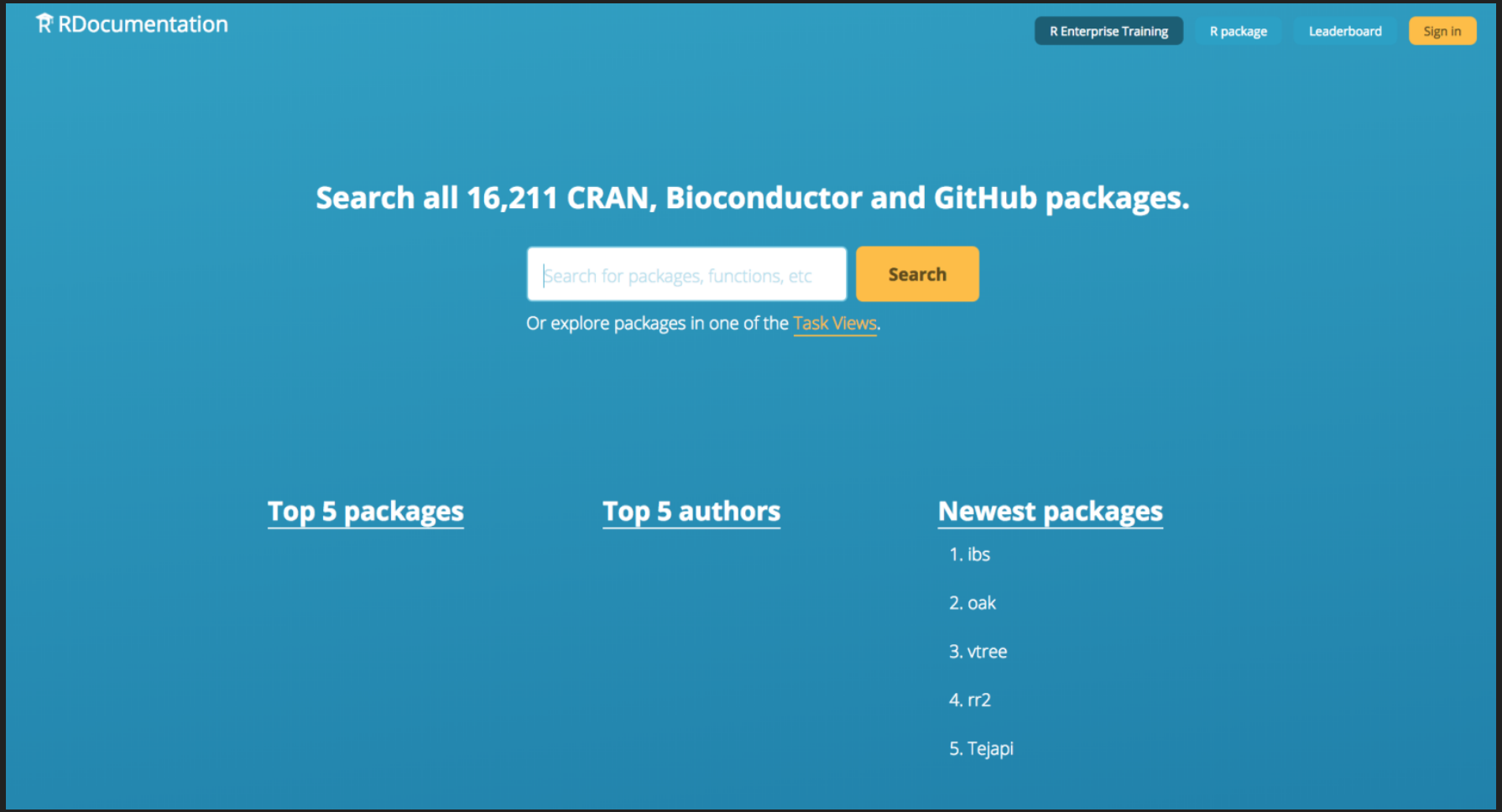
CRAN packages:

- boilerpipeR
- corpora
- gsubfn
- gutenbergr
- hunspell
- kernlab
- KoNLP
- koRpus
- languageR
- lda
- lsa
- maxent
- monkeylearn
- movMF
- mscstexta4r
- mscsweblm4r
- openNLP
- phonics
- qdap
- quanteda
- RcmdrPlugin.temis
- RKEA
- RTextTools



- RWeka
- skmeans
- SnowballC
- stringi
- tau
- tesseract
- text2vec
- textcat
- textir
- textreuse
- tidytext
- tm (core)
- tm.plugin.alceste
- tm.plugin.dc
- tm.plugin.europresse
- tm.plugin.factiva
- tm.plugin.lexisnexis
- tm.plugin.mail
- tm.plugin.webmining
- tokenizers
- topicmodels
- wordcloud
- wordnet
- zipfR

https://cran.r-project.org/web/views/NaturalLanguageProcessing.html


RDocumentation
R Enterprise Training
R package
Leaderboard
Sign in
Search all 16,211 CRAN, Bioconductor and GitHub packages.
Search for packages, functions, etc
Search
Or explore packages in one of the Task Views.
Top 5 packages
Top 5 authors
Newest packages
1. ibs
2. oak
3. vtree
4. rr2
5. Tejapi

RStudio:

a graphical user interface for R

available as a desktop application or as an online application

~/machine-learning-test - RStudio

Go to file/function | Addins | machine-learning-test

Ex_1.R | Ex_2.R

Source on Save | Run | Source

```
1  install.packages("ISLR")
2  install.packages("randomForest")
3  library(ISLR)
4  library("randomForest")
5  library(ggplot2)
6  library(caret)
7
8  # Know more about the Wage dataset
9  data(Wage)
10 summary(Wage)
11 dim(Wage)
12
13
14 # Some feature engineering, we don't need logwage
15 Wage<- subset(Wage, select=- c(logwage))
16
17 # Some Exploratory data analysis a.k.a. visualization
18 qplot(age, wage, data=Wage, colour = race)
19 qplot(age, wage, data=Wage, colour=education)
20
21 Wage_x <- data.frame(as.numeric(Wage$age), as.numeric(Wage$education), as.numeric(Wage$jobclass),as.nu
22 cor(Wage_x, Wage$wage)
23
24
25 inTrain<- createDataPartition(y = Wage$wage, p=0.7, list= FALSE)
26
27 training<- Wage[inTrain,]
```

20:1 (Top Level) | R Script

script

Console ~/machine-learning-test/

```
307460    281.74597 128.97951
156087    134.70538 100.35238
12157      75.35568  82.67135
10832     118.88436 101.01245
160170     27.14058  76.07576
453712    182.02062 147.03898
305387    141.77517 146.90206
380031    135.59806 130.24479
154617     96.37065 128.80519
447647     54.59815  96.90569
451555     96.86670  92.74230
153403     95.23071 109.92425
```

output

Environment | History

Import Dataset | List

Global Environment

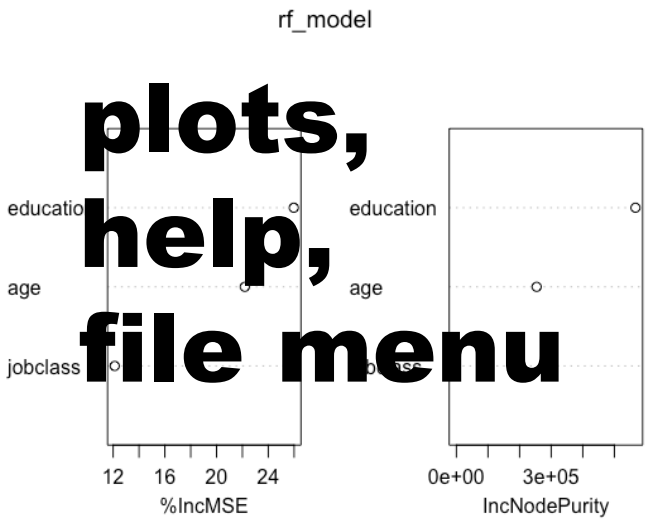
Data

compare_result	898 obs. of 2 variables
emp_data	20 obs. of 2 variables
earnings: num	
s_rating: num	
inTrain	int [1:2102, 1] 1 2 5 6 7 10 11 12 14 17 ...
test	7 obs. of 2 variables
testing	obs. of variables
training	obs. of variables
Wage	3000 obs. of 11 variables

session info

Files | Plots | Packages | Help | Viewer

Zoom | Export | Publish



plots, help, file menu

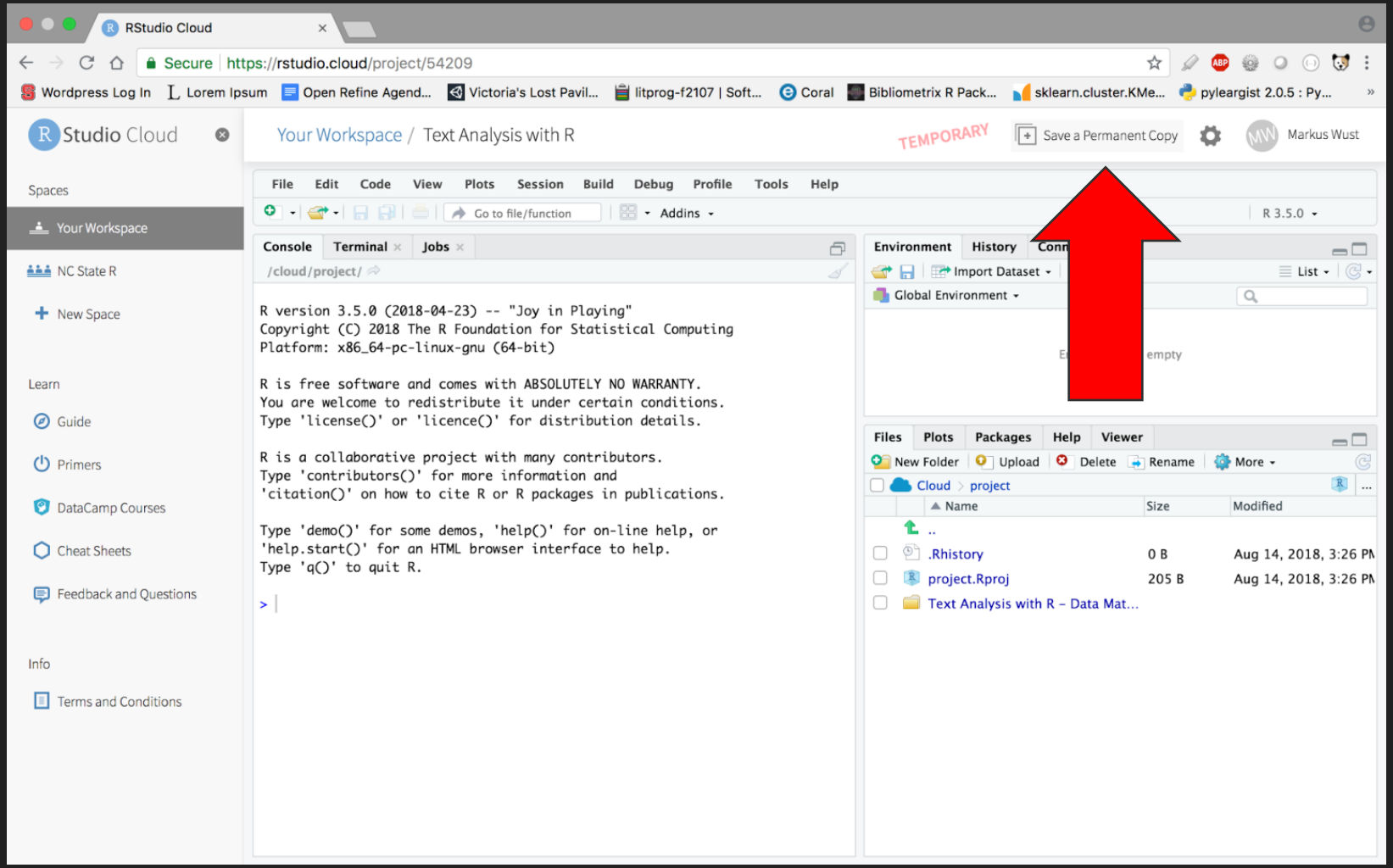
RStudio Cloud:

a web-based version of R Studio

same features, but also sharing options

RStudio Cloud login

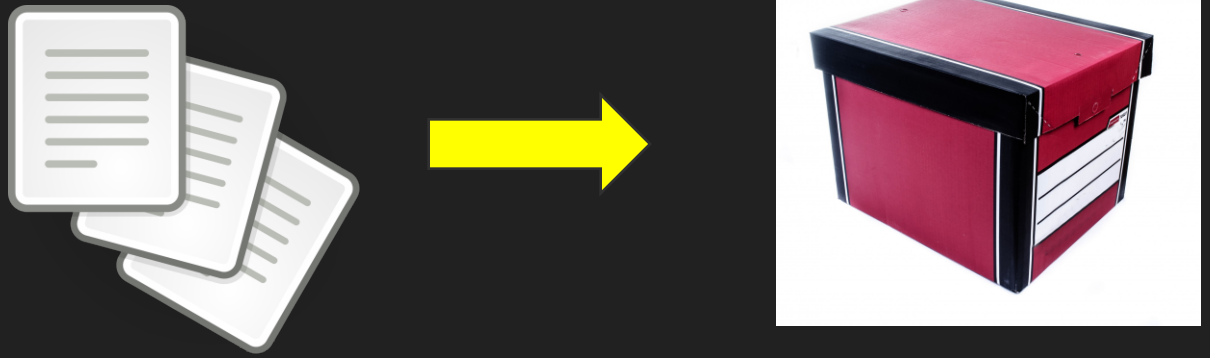
go.ncsu.edu/textr


RStudio Cloud
Secure https://rstudio.cloud/project/54209
Wordpress Log In
Lorem Ipsum
Open Refine Agend...
Victoria's Lost Pavil...
litprog-f2107 | Soft...
Coral
Bibliometrix R Pack...
sklearn.cluster.KMe...
pyleargist 2.0.5 : Py...
Studio Cloud
Your Workspace / Text Analysis with R
TEMPORARY
Save a Permanent Copy
Markus Wust
Spaces
Your Workspace
NC State R
New Space
Learn
Guide
Primers
DataCamp Courses
Cheat Sheets
Feedback and Questions
Info
Terms and Conditions
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
R 3.5.0
Console
Terminal
Jobs
/cloud/project/
R version 3.5.0 (2018-04-23) -- "Joy in Playing"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
>
Environment
History
Import Dataset
List
Global Environment
empty
Files
Plots
Packages
Help
Viewer
New Folder
Upload
Delete
Rename
More
Cloud
project
Name
Size
Modified
..
.Rhistory
0 B
Aug 14, 2018, 3:26 PM
project.Rproj
205 B
Aug 14, 2018, 3:26 PM
Text Analysis with R – Data Mat...

Relaxing Dog Picture

Preprocessing: Preparing Text for Analysis

Step 1:
Create a corpus from your raw text file(s)



Corpus Object in R

Corpus is a data structure in R that represents a collection of documents that you import into R or access using an API.

tm package

SimpleCorpus
VCorpus
PCorpus

Why Create a Corpus?

Creating a corpus object gives you access to text mining and analysis functions from various text analysis R packages: tm, quanteda, openNLP, topicmodels, etc.
From a corpus object you can:

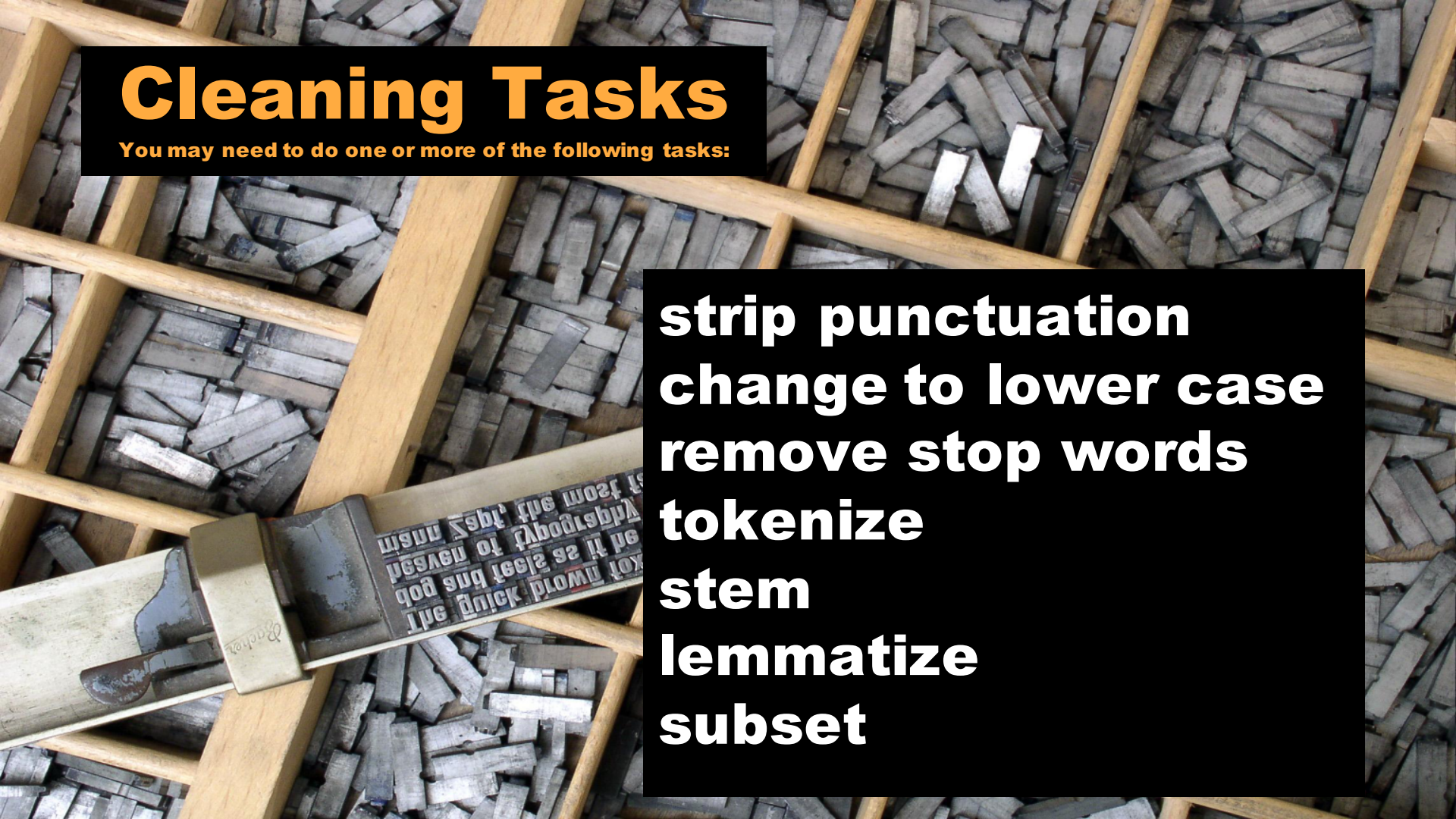
- data cleaning and transforming
- filtering
- document-term-matrix OR
- term-document-matrix

```
<<DocumentTermMatrix (documents: 20, terms: 1183)>>
Non-/sparse entries: 1908/21752
Sparsity           : 92%
Maximal term length: 17
Weighting          : term frequency (tf)
Sample             :
     Terms
Docs  crude dlrs last mln oil opec prices reuter said saudi
  144     0    0    1   4  11   10      3      1    9     0
  236     1    2    4   4   7    6      2      1    6     0
  237     0    1    3   1   3    1      0      1    0     0
  242     0    0    0   0   3    2      1      1    3     1
  246     0    0    2   0   4    1      0      1    4     0
  248     0    3    1   3   9    6      7      1    5     5
  273     5    2    7   9   5    5      4      1    5     7
  489     0    1    0   2   4    0      2      1    2     0
  502     0    1    0   2   4    0      2      1    2     0
  704     0    0    0   0   3    0      2      1    3     0
```

Step 2:
Clean the corpus

Cleaning Tasks

You may need to do one or more of the following tasks:



strip punctuation
change to lower case
remove stop words
tokenize
stem
lemmatize
subset

Data cleaning terms

stopwords - [the, a, an, and, not, for, but, so...]

tokenize - the/cat/sat/on/the/mat/

stem - playing, played, player → play*

lemmatize -- good, better, best → good

trying, tried, tries → try

Alternate approach:

Structuring text in a dataframe

(as opposed to a corpus)

Tidy the Data

Put unstructured text in a tabular structure

Take the text and turn it into a **data frame** with columns and rows

Use tidy data principles

Tidy data principles

One variable per column

One observation per row

One value per cell

Lat, Long	Address
35.7796, -78.6382	3100 Hillsborough Street, Raleigh, NC

Lat	Long	Street	City	State
35.7796	-78.6382	3100 Hillsborough Street	Raleigh	NC

Tidy text structure

“a table with one token per row” - Julia Silge, *Text Mining with R*

A token is a meaningful unit, such as a word, word pair, or sentence

Converting a text file to tidy text data frame

Raw text file

```
CHAPTER I

JONATHAN HARKER'S JOURNAL

(_Kept in shorthand._)


_3 May. Bistritz._--Left Munich at 8:35 P. M., on 1st May, arriving at
Vienna early next morning; should have arrived at 6:46, but train was an
hour late. Buda-Pesth seems a wonderful place, from the glimpse which I
got of it from the train and the little I could walk through the
streets. I feared to go very far from the station, as we had arrived
late and would start as near the correct time as possible. The
impression I had was that we were leaving the West and entering the
East; the most western of splendid bridges over the Danube, which is
here of noble width and depth, took us among the traditions of Turkish
rule.
```

Tidy text

	gutenberg_id	title	word
208	345	Dracula	_3
209	345	Dracula	bistritz
210	345	Dracula	left
211	345	Dracula	munich
212	345	Dracula	8
213	345	Dracula	35
214	345	Dracula	1st
215	345	Dracula	arriving
216	345	Dracula	vienna
217	345	Dracula	morning
218	345	Dracula	arrived

Text Analysis Methods

Word frequency

Discover the most frequent words in a text

The mouse ate the cheese.

The 2

mouse 1

ate 1

cheese 1

Collocation

Conventional expressions of two or more words. Reveal patterns of word usage.

“time”

save time

make some **time**

nick of **time**

bide my **time**

N-grams (bi, tri-, etc)

Example: “The merry month of May”

Where n = 2:

The merry

merry month

month of

of May

These are bigrams!

Concordance

key word in context

‘Will you be my own true love?’ the musician sang…”
“took the dusty books with love and dusted them…”

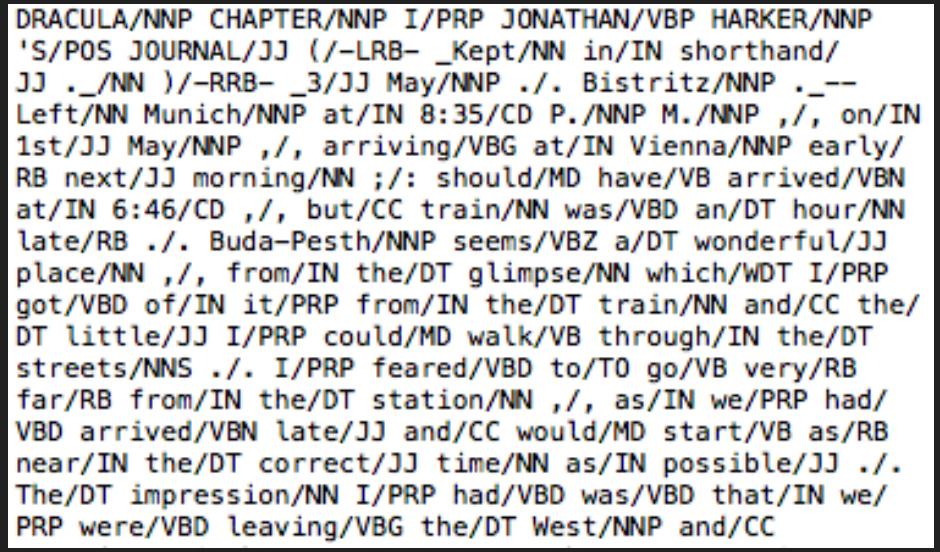
Natural Language Processing

- Part of speech tagging
- Topic modeling
- Sentiment analysis

Part of speech tagging

Purpose: to be able to parse strings by part of speech

What are the most frequent nouns used in A Novel?



Topic modeling

Purpose: discover “topics” in a collection of documents… using statistics!

Topic modeling

- You tell the computer to find 5 "topics" (gut feeling)
- Computer tries to figure out 5 "topics" based on distribution of words -> doesn't know what those topics are about
- Computer shows words that make up "topic"
- You figure out connection between words

Topic modeling with LDA

Latent Dirichlet Allocation Model

Uses probabilities to generate topics assumed hidden in a text

- You give the computer X number of topics to discover.
- The computer gives you X groups of words that statistically seem to have been generated by the hidden topics.
- You don’t get the actual topics.

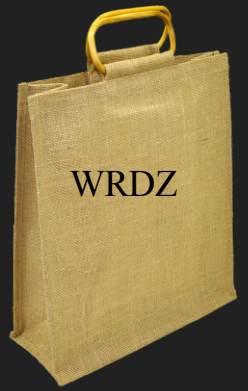
LDA model example

You: Okay computer, find me 5 topics from this set of documents I have here. I don't know what the topics are -- figure it out for me!

(you, the human, simply want to discover topics in a collection of documents. You pick the number 5 because it feels right)

Computer: OK. I assume that this text corpus is a random selection of words generated from exactly 5 topics. Let me figure out the five topics that generated this corpus.

(the computer model is not a human. It uses math to determine 5 original “topics” used to generate a text corpus, which it sees as a bag of words.)



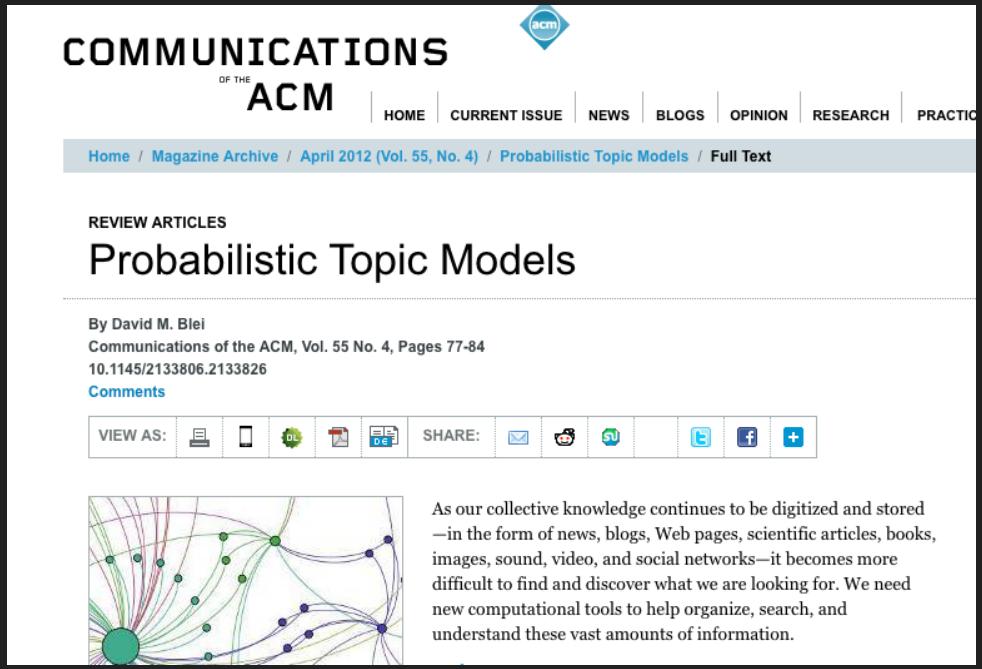
Computer: here are the 5 topics and the words that have the highest probability of being associated with those topics:

Topic 1: house / horsemen / ride / night
Topic 2: window / bed / night / sleep
Topic 3: lantern / sea / mate / farthing
Topic 4: dogs / river / woods / darkness
Topic 5: shallow / november / hand / time

Human: Hmm...ok. Maybe topic one is “journey.” Maybe topic two is “home.” Maybe topic 3 is “seafaring.” Maybe topic 4 is “hunting.”

Topic models resource

Probabilistic Topic Models
By David M. Blei (2012)



COMMUNICATIONS OF THE ACM

HOME | CURRENT ISSUE | NEWS | BLOGS | OPINION | RESEARCH | PRACTIC

Home / Magazine Archive / April 2012 (Vol. 55, No. 4) / Probabilistic Topic Models / Full Text

REVIEW ARTICLES

Probabilistic Topic Models

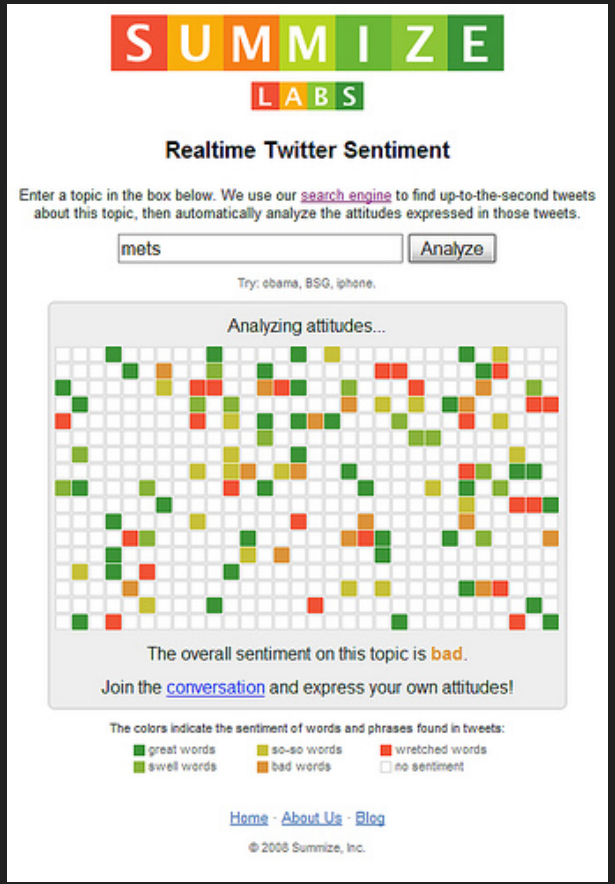
By David M. Blei
Communications of the ACM, Vol. 55 No. 4, Pages 77-84
10.1145/2133806.2133826
Comments

VIEW AS: SHARE:

As our collective knowledge continues to be digitized and stored —in the form of news, blogs, Web pages, scientific articles, books, images, sound, video, and social networks—it becomes more difficult to find and discover what we are looking for. We need new computational tools to help organize, search, and understand these vast amounts of information.

Sentiment analysis

Purpose: discover sentiment in a text (positive, negative, neutral, etc.)



Sentiment analysis

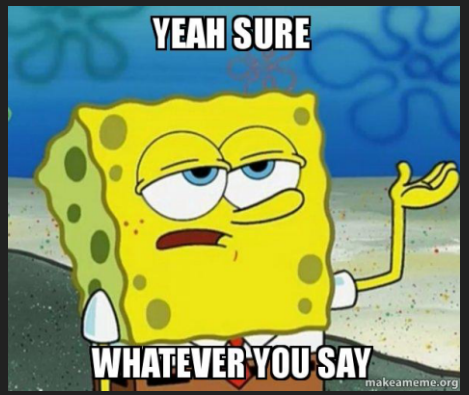
I don’t like seagulls. They creep me out.

I don’t dislike seagulls. Some think they are creepy but I don’t.

Sentiment analysis

Limited:

- **Sentiment is objective**
- **Sentiment is not always expressed explicitly**
- **Words can have multiple meanings**



Text Analysis: Glossary

tf-idf - how important a term is in distinguishing a text from others in a collection

document-term matrix - matrix of whether a term exists in a document. rows: documents, columns: terms

Workshop activities

Data import & cleaning

word frequency

tf-idf

also in the script but not covered:

topic modeling with LDA model

sentiment analysis

Resources

Julia Silge and David Robinson (2017), Text Mining with R: A Tidy Approach

Matthew Jockers (2014), Text Analysis with R for Students of Literature

Evaluation

go.ncsu.edu/libeval

Another Relaxing Dog Picture