Chapter 20 – Text Mining

Data Mining for Business Analytics in R

Shmueli, Bruce & Patel

- *Video seminar Monday after class. GPS 2015*
  - *Two areas where employment opportunities are high for GPS students.*

The first is the traditional data analyst role. based on exploratory analytics. Honestly, most firms, including Deloitte, are looking for people that can build dashboards from disparate data sources.
The power of being good at cleaning and preparing a dataset should not be under emphasized. THERE WILL ALWAYS BE A NEED AND DEMAND FOR THOSE THAT CAN MANIPULATE DATA TO FIT TO A CLIENT'S NEED.

- *Uber self-driving crash: Software set to ignore objects on road*
  - *Sensitivity settings can really matter!*

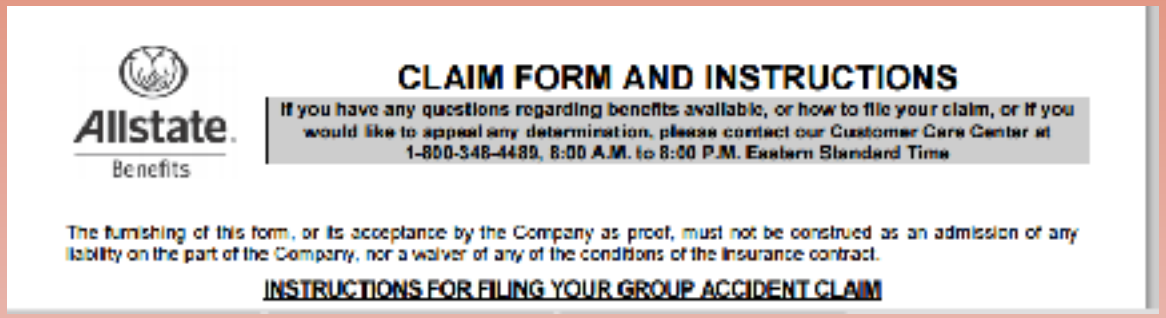
Humans communicate in words

- *Machines use numbers*
- *Result: Text mining = watching how people communicate with each other.*

- *Up to now we have been dealing with structured quantitative data*
  - *Numerical*
  - *Binary (yes/no)*
  - *Multicategory*
- *Now we turn to unstructured text*

Applications of Text Mining

- *Insurance fraud – notes in claim forms can be mined and transformed into predictor variables for a predictive model*
- *The model is trained on prior claims in two classes – found to be fraudulent, and not found to be fraudulent*
- *The model is then applied to new claims*


Allstate.
Benefits

CLAIM FORM AND INSTRUCTIONS

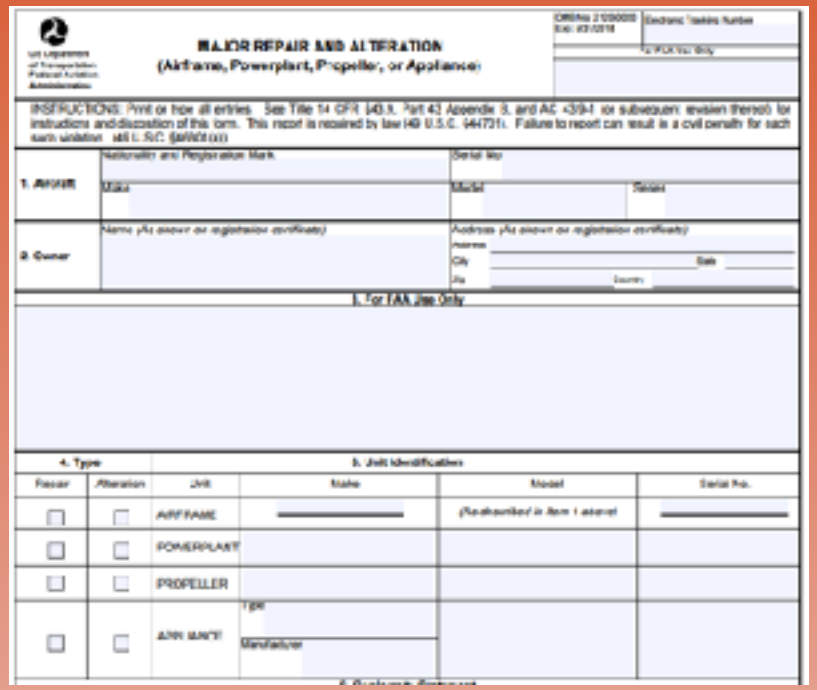
If you have any questions regarding benefits available, or how to file your claim, or if you would like to appeal any determination, please contact our Customer Care Center at 1-800-348-4489, 8:00 A.M. to 8:00 P.M. Eastern Standard Time

The furnishing of this form, or its acceptance by the Company as proof, must not be construed as an admission of any liability on the part of the Company, nor a waiver of any of the conditions of the insurance contract.

INSTRUCTIONS FOR FILING YOUR GROUP ACCIDENT CLAIM

Applications, cont.

- *Maintenance or support tickets often contain text fields*
- *These fields could be mined to classify ticket in several ways:*
  - *How urgent?*
  - *How much time to fix?*
  - *What category of technician is needed to fix?*



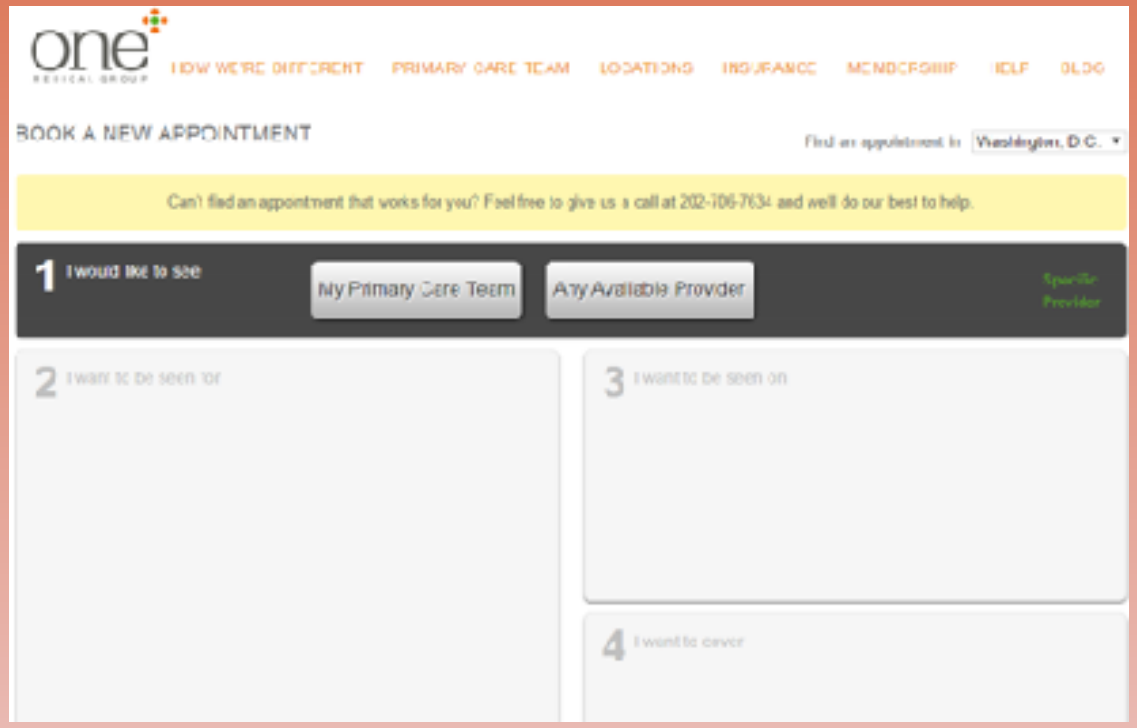
MAJOR REPAIR AND ALTERATION
(Airframe, Powerplant, Propeller, or Appliance)

INSTRUCTIONS: Print or type all entries. See Title 14 CFR §43.9, Part 43 Appendix B, and AC 43.9-1 (or subsequent revision thereto) for instructions and discussion of this form. This report is required by law (49 U.S.C. 44721). Failure to report can result in a civil penalty for each such violation (49 U.S.C. §46301(a)).

1. Aircraft — Nationality and Registration Mark; Serial No.; Make; Model; Series
2. Owner — Name (As shown on registration certificate); Address (As shown on registration certificate); Number; City; State; Zip; Country
3. For FAA Use Only
4. Type — Repair; Alteration
5. Job Identifications — Job; Make; Model; Serial No.
AIRFRAME (Nacelle/pod(s) in Item 1 above)
POWERPLANT
PROPELLER
ANY UNIT — Type; Manufacturer
6. Description of Repair or Alteration

Applications, cont.

- *Medical triage/diagnosis*
- *Clinics could use patient online appointment request forms to route requests*
  - *Admin asst.*
  - *Nurse*
  - *Doctor*



What exactly is text mining?

- *Classify (label) thousands of documents*
  - *Extension of predictive modeling (our focus)*
- *Extract meaning from a single document – interpreting it like a human reads language*
  - *"Natural language processing" (very ambitious, not predictive modeling, not our focus)*

Classification (labeling) and clustering

- *No attempt to extract overall document meaning from a single document*
- *Focus is on assigning a label or class to numerous documents*
- *As with numerical data mining, the goal is to do better than guessing*

"Bag-of-words"

- *Grammar, syntax, punctuation, word order are ignored*
- *The document is considered as a "bag of words"*
- *This approach is, nonetheless, effective when the goal is to decide which category or cluster a document falls in*
- *A typical application is supervised learning*
- *Requires lots of documents (a corpus)**
- *Do not need 100% accuracy*

**"Corpus" often refers to a fixed standard set of documents that many researchers can use to develop and tune text mining algorithms.*

The spreadsheet model of text

- *Columns are terms*
- *Rows are documents*
- *Cells indicate presence/absence (or frequency) of terms in documents*
- *Consider the two sentences:*
  - *S1 First we consider the spreadsheet model*
  - *S2 Then we consider another model*

Here is the resulting spreadsheet, using presence/absence:

	first	we	consider	the	spreadsheet	model	then	another
S1	1	1	1	1	1	1	0	0
S2	0	1	1	0	0	1	1	0

Need to turn text into a matrix

- *For the two documents (sentences S1 and S2) that we looked at earlier, the process of producing a matrix is simple. We had*
  - *Words*
  - *Spaces*
  - *Periods*
- *Each word is preceded or followed by a space or period – a delimiter.*
- *Real text is more complicated*

Lots of things to process besides words…

- *Numbers (including dates, percents, monetary amounts), e.g. from Google Annual Report 2014:*

We considered the historical trends in currency exchange rates and determined that it was reasonably possible that changes in exchange rates of 20% could be experienced in the near term. If the U.S. dollar weakened by 20% at December 31, 2013 and 2014, the amount recorded in AOCI related to our foreign exchange options before tax effect would have been approximately $4 million and $686 million lower at December 31, 2013 and December 31, 2014, and the total amount of expense recorded as interest and other income, net, would have been approximately $123 million and $90 million higher in the years ended December 31, 2013 and December 31, 2014. If the U.S. dollar strengthened by 20% at December 31, 20013 and December 31, 2014, the amount recorded in accumulated AOCI related to our foreign exchange options before tax effect would have been approximately $1.7 billion and $2.5 billion higher at December 31, 2013 and December 31, 2014, and the total amount of expense recorded as interest and other income, net, would have been approximately $120 million and $164 million higher in the years ended December 31, 2013 and December 31, 2014.

Email addresses, url's, stray characters introduced by file conversions, …

Sender: Distribution list for statistical items of interest <WSS-ELECTRONIC-MAIL-LIST@LISTS.MITRE.ORG> From: "Massimini, Vince" <svm@mitre.org> Subject: Comparing the Maximal Procedure to Permuted Blocks Randomization To: <WSS-ELECTRONIC-MAIL-LIST@LISTS.MITRE.ORG> Precedence: list List-Help: <mailto:LISTSERV@LISTS.MITRE.ORG?body=INFO%20WSS-ELECTRONIC-MAIL-LIST>

For more WSS events, see washstat.org WSS Public Health/Biostatistics Section and NCI Division of Cancer Preventi= on Jointly Sponsored Event: =20

SPEAKER: Vance W. Berger, PhD National Cancer Institute and University of = Maryland Baltimore County and Klejda Bejleri, BS Biometry and Statistics, D= epartment of Biological Statistics and Computational Biology, Cornell Unive= rsity, Ithaca, NY 14853 =20 TITLE: Comparing the Maximal Procedure to Permuted Blocks Randomization

TIME AND PLACE: Monday, June 8th NCI Shady Grove, 9609 Medical Center Drive= , Rockville MD Room 5E30/32. Bring photo ID, allow time to get through secu= rity

Proper nouns & terms specific to a particular field

From Techsmith corporate information: All-In-One Capture, Camtasia, Camtasia Studio, Camtasia Relay, Coach's Eye, DubIt, EnSharpen, Enterprise Wide, Expressshow, Jing, Morae, Rich Recording Technology (RRT), Snagit, Screencast.com, ScreenChomp, Show The World, SmartFocus, TechSmith, TechSmith and T Design logo, TechSmith Fuse, TechSmith Relay, TSCC, and UserVue are marks or registered marks of TechSmith Corporation.

From medical journal: Eight hundred elderly women and men from the population-based Framingham Osteoporosis Study had BMD assessed in 1988-1989 and again in 1992-1993. BMD was measured at femoral neck, trochanter, Ward's area, radial shaft, ultradistal radius, and lumbar spine using Lunar densitometers. (Risk Factors for Longitudinal Bone Loss in Elderly Men and Women: The Framingham Osteoporosis Study, Journal of Bone and Mineral Research Volume 15, Issue 4, pages 710–720, April 2000)

Tokenization

- *We need to move from a mass of text to useful predictor information*
- *The first step is to separate out and identify individual terms*
- *The process by which you identify delimiters and use them to separate terms is called tokenization. The resulting terms are also called tokens.*

Preprocessing

- *Goal – reduction of text (also called vocabulary reduction) without losing meaning or predictive power*
- *Stemming*
  - *Reducing multiple variants of a word to a common core*
  - *Travel, traveling, traveled, etc. -> travel*
- *Ignore case*
- *Frequency filters can eliminate terms that*
  - *Appear in nearly all documents*
  - *Appear in hardly any documents*

Preprocessing, cont.

- *Punctuation characters and extra white space can be removed, and treated as delimiters*
- *Remove terms that are on a stoplist (stopwords)*
  - *Typically is done to reduce size and noise by getting rid of very common terms*
  - *Illustrated with the default "cnglish" stoplist that comes with R's* `tm`

Preprocessing, cont.

- *Frequency vs. presence/absence*
- *Normalization, when the presence of a type of term might be important but we don't need the specific term. For example*
  - *Replace john@domain.com with "email token"*
  - *Replace www.domain.com with "url token"*

Post-reduction matrix

- *Columns are documents, rows are terms*
- *Options for cell entries:*
  - *0/1 (presence absence)*
  - *Frequency count*
  - *TF-IDF (term frequency – inverse document frequency)*
  - *TF = frequency of term*
  - *IDF = log of inverse of the frequency with which documents have that term*
  - *There are varying definitions of both TF and IDF, hence of TF-IDF*
  - *Bottom line:*
    - *TF-IDF is high where a rare term is present or frequent in a document*
    - *TF-IDF is near zero where a term is absent from a document, or abundant across all documents*

Using R's `tm` package with simple example

S1. this is the first sentence.
S2. this is a second sentence.
S3. the third sentence is here.

enter the text as a corpus:

```
text <- c("this is the first sentence",
     "this is a second sentence",
     "the third sentence is here")

corp <- Corpus(VectorSource(text))
```

Producing the term-document matrix

```
tdm <- TermDocumentMatrix(corp)
inspect(tdm)
Output

<<TermDocumentMatrix (terms: 7, documents: 3)>>
Non-/sparse entries: 12/9
Sparsity : 43%
Maximal term length: 8
Weighting : term frequency (tf)
        Docs
Terms    1 2 3
first    1 0 0
here     0 0 1
second   0 1 0
sentence 1 1 1
the      1 1 1
third    0 0 1
this     1 1 0
```

The term “first” does not occur in document 3

Basic Text Mining in R

- *Philip Murphy, PhD 4/4/2017*
- *docs <- tm_map(docs,removePunctuation)*
- *writeLines(as.character(docs[1])) # Check to see if it worked.*

- *docs <- tm_map(docs, tolower)*
- *docs <- tm_map(docs, PlainTextDocument)*

May have 1000s of terms

- *Too big to analyze easily*
- *With numeric data, we have ways around this*
- *With text data, additional methods*
- *Many tokens are "junk"*
  - *Spelling errors*
  - *Only occur once*
  - *Not relevant (but are you* sure*?)*
- *Method 1: Filter out the junk*
- *Method 2: Select tokens likely to be important*
- *Method 3: Build a new matrix in "reduced form"*
  - *Latent Semantic Indexing*

Getting rid of some junk

- Cleaning
  - Corp <- tm_map(Corp, removeNumbers)
  - getTransformations( ) # plus more from murphy
    - [1] "removeNumbers" "removePunctuation" "removeWords"
    - [4] "stemDocument" "stripWhitespace"
    - tolower
    - PlainTextDocument
    - Corp <- tm_map(Corp, removeNumbers)
  -

From terms to concepts – Latent Semantic Indexing (LSI)

- *The post-reduction term/document matrix is often still huge – too big for easy processing*
- *Similar to principal components.*
  - *Create small set of synthetic predictor variables,*
  - *Each a linear combination of "like-minded" original variables.*
- *Latent semantic indexing similar for text –*
  - *maps multiple terms to a small set of concepts.*
  - *Based on terms that co-occur*

Intuitive explanation - LSI

each time the term “alternator” appeared in an automobile document, the document also included the terms “battery” and “headlights.”

Or each time the term “brake" appeared in an automobile document, the terms “pads” and “squeaky” also appeared.

However, there is no detectable pattern regarding the use of the terms “alternator” and “brake” together. Documents including “alternator” might or might not include “brake” and documents including “brake” might or might not include “alternator.”

*Our four terms, b**attery, headlights, pads, and squeaky** describe two different automobile repair issues:*

failing brakes and a bad alternator.

Analytic Solver Platform, XLMiner Platform, Data Mining User Guide, 2014, Frontline Systems, p. 245}

Extracting Meaning?

- *It may be possible to use the concepts to identify themes in the document corpus, and clusters of documents sharing those themes.*
- *Often, however, the concepts do not map in obvious fashion to meaningful themes.*
- *Their key contribution is simply reducing the vocabulary – instead of a matrix with thousands of columns, we can deal with just 10 to 30.*
- *Depends on diversity of documents*
-

Alternative for reducing variables

- *Remove junk*
- *Keep most important terms*
- *Latent Semantic Analysis*

- *Also* add *some variables! (Almost always)*
  - *Numeric, easy to measure*
  - *Size of document*
  - *Count words of special types (swear words, etc)*
  - *Count special tokens*

Iterative improvement strategy

- Run the basic default stemming
- Look at top 100 terms
  - Trace back to originals
  - Make changes as needed (see below)
- Specialized vocabulary lists
  - Synonym lists
  - Stemming rules
- Use bigrams, trigrams ”not very large”, “mis-identified”
- Proper nouns, phrases: convert to tokens *qqdonaldt*, *qqtedcruz*, *qqhilary*, *qqRepublicanParty*
- What is important for meaning, in *your* project?

A predictive model

- *Now we have a clean, structured dataset similar to what we have used in our numerical data mining:*
  - *Class identifications (labels) for training*
  - *Numerical predictors*

Example: Classify posts as auto-related, or electronics-related

ALWAYS: Look at the *raw* data!! Study it!

```
From: smith@logos.asd.sgi.com (Tom Smith) Subject: Ford Explorer 4WD -
do I need performance axle?
We're considering getting a Ford Explorer XLT with 4WD and we have the
following questions (All we would do is go skiing - no off-roading):
1. With 4WD, do we need the "performance axle" - (limited slip axle). Its
purpose is to allow the tires to act independently when the tires are on
different terrain.
2. Do we need the all-terrain tires (P235/75X15) or will the all-season
(P225/70X15) be good enough for us at Lake Tahoe?
Thanks,
Tom
—
==============================================
Tom Smith Silicon Graphics smith@asd.sgi.com 2011 N. Shoreline Rd. MS
8U-815 415-962-0494 (fax) Mountain View, CA 94043
==============================================
```

How raw is "raw"??

Answer: Very!

multiple levels, from laptop screen to row in your corpus

Tip: Study your raw data!

It will always have surprises

Especially with text (you can read it easily!)

1. ***Original source, in the wild***
2. ***Gathered into a dataset somewhere eg Excel***
    1. ***Most time spent here, especially at start***
3. *Before it's read into R*
4. *When first read into R*
5. *After cleaning*
6. *After each transformation - to make sure the transform did what you expected*
    1. *Minimum: check variable size, do str()*

Import the data, create document labels:

```
library(tm)
# step 1: import and label records
# read zip file into a corpus
corp <- Corpus(ZipSource("AutoElectronics.zip",
    recursive = T))

# create an array of records labels
label <- c(rep(1, 1000), rep(0, 1000))
```

"reach into subdirectories while getting documents"

1000 1's which will be used later to label the first 1000 documents, which we know are auto, and 1000 0's to label the remainder, which are electronic

Preprocess the text
(function `tm_map`)

```
# tokenization
corp <- tm_map(corp, stripWhitespace)
corp <- tm_map(corp, removePunctuation)
corp <- tm_map(corp, removeNumbers)

# stopwords
corp <- tm_map(corp, removeWords, stopwords("english"))

# stemming
corp <- tm_map(corp, stemDocument)
```

Produce the Concept Matrix

```
# step 3: TF-IDF and latent semantic analysis

# compute TF-IDF
tdm <- TermDocumentMatrix(corp)
tfidf <- weightTfIdf(tdm)

# extract (20) concepts
library(lsa)
lsa.tfidf <- lsa(tfidf, dim = 20)

# convert to data frame
words.df <- as.data.frame(as.matrix(lsa.tfidf$dk))
```

Now run a standard predictive model on the Concept Matrix

```
# sample 60% training data
training <- sample(c(1:2000), 0.6*2000)

# run logistic model on training
trainData = cbind(label = label[training], words.df[training,])
reg <- glm(label ~ ., data = trainData, family = 'binomial')

# compute accuracy on validation set
validData = cbind(label = label[-training], words.df[-training,])
pred <- predict(reg, newdata = validData, type = "response")

# produce confusion matrix
```

```
          Reference
Prediction 0   1
0          385 16
1          9   390
Accuracy : 0.9688
```

The high accuracy shows the posts are very separable

functions learned today

- library(tm) #text mining.
- docs <- **VCorpus**(DirSource(cname))
  - corp <- **Corpus**(ZipSource("AutoElectronics.zip", recursive = T)). #similar, different file source
- Corp <- **tm_map**(Corp, removeNumbers)
  - ●"removePunctuation" "removeWords" tolower
- **writeLines**(as.character(Corp[1])) # Check to see if it worked.
- Don't forget: Look at samples of RAW data
- Tip: Add some variables
- **str**(data). #work with any data
- **rep**(1, 1000) #repeat 1000 times
-
- training <- **sample**(c(1:2000), 0.6*2000). #take 1200 samples from 1:2000
- tdm <- **TermDocumentMatrix**(corp)
- tfidf <- **weightTfIdf(**tdm)
- Sparse2 <- **removeSparseTerms**(Tdm, .98) #keep only 2% approximately
- **head**(Sparse2$dimnames$Terms, 50). #first 50 elements.
  - # could also use sample
  - **sample**(Sparse2$dimnames$Terms, 20, replace=TRUE)
- lsa.tfidf <- **lsa**(tfidf, dim = 20) #Doing LSA analysis. Careful, can take minutes
- lsa.tfidf$dk # extracts variable (column) called 'dk'
- words.df <- **as.data.frame**(as.matrix(lsa.tfidf$dk))
- reg <- **glm**(label ~ ., data = trainData, family = 'binomial')
- pred <- **predict**(reg, newdata = validData, type = "response")