UCLA Anderson
SCHOOL of MANAGEMENT

MGMTMFE 431:

Data Analytics and Machine Learning

Topic 7: Unstructured Data
Introduction to Textual Analysis

Spring 2019

Professor Lars A. Lochstoer

Unstructured Data and Introduction to Textual Analysis

a. What is unstructured data?

b. Introduction to textual analysis
- Example: reading financial reports using EDGAR data

c. Mapping unstructured data to numerical signals

d. Ravenpack and sentiment scores

UCLA Anderson
SCHOOL of MANAGEMENT

a. What is unstructured data?

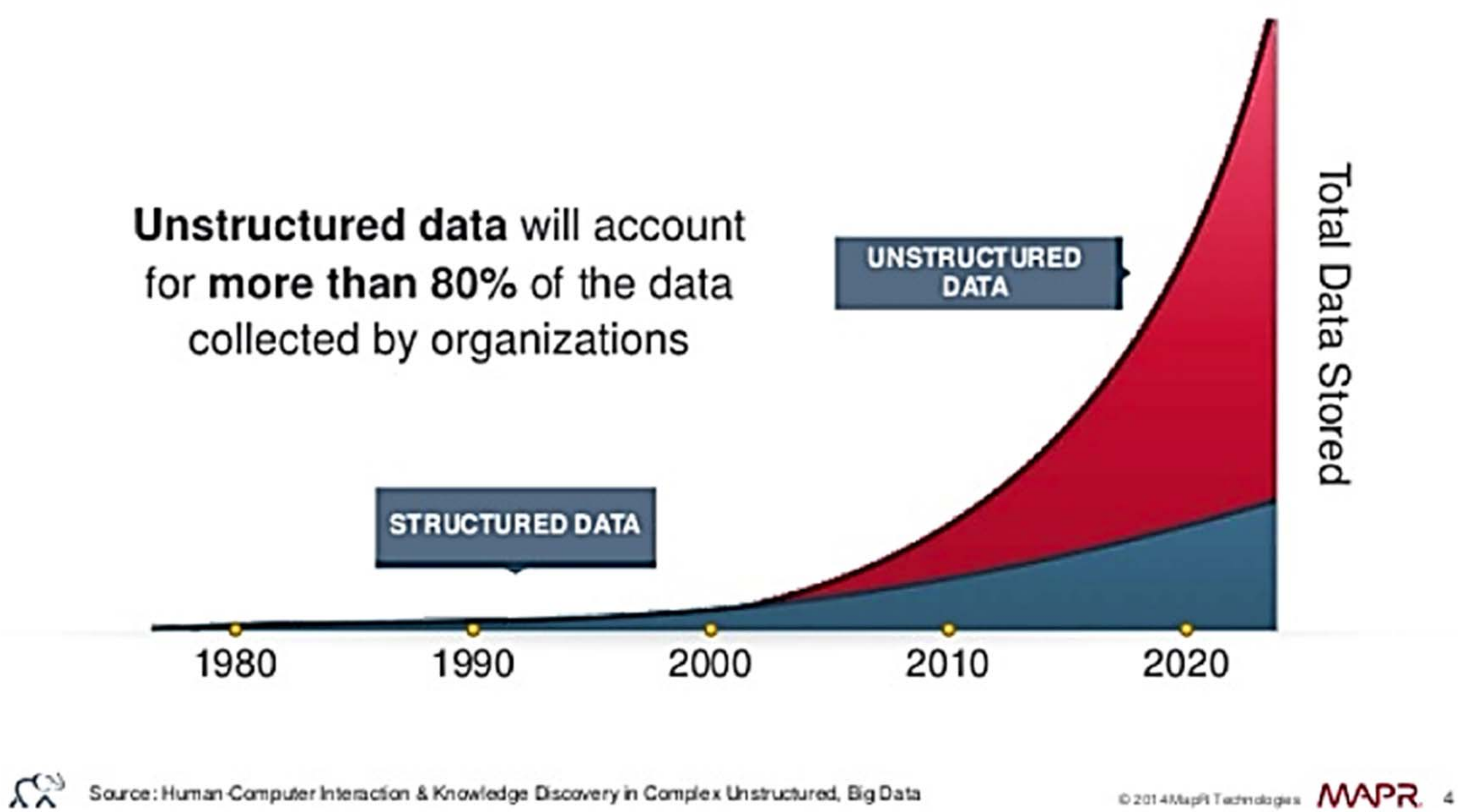
Structured data

- Asset prices, returns, accounting numbers, macroeconomic series, volume, inventory, earnings, dividends, etc.
- Easy to read and input into models
- Most data used historically are of this form

Unstructured data

- Newspaper articles, blogs, internet search data, text components of financial reports (both firms' reports and analyst reports)
- Most data is in this form
  - Note: this does not necessarily mean most of information content is in this form…! A lot is likely captured within existing structured data, including asset prices
- More qualitative in nature, harder to analyze

a. Growth of unstructured data



UCLA Anderson SCHOOL of MANAGEMENT

a. Challenge of unstructured data

1. Filter out the (large amount of) noise
   - …but don't throw out the baby with the bath water…

2. Create ***informative*** numerical signal
   - Based on data that typically displays strong trends (see last slide)
   - Erratic behavior over time (e.g., less information over weekends, lots in weekdays)
   - Text does not equal text: ***Content*** provider matters! (Bloomberg might me more informative than a random blog, etc.)
   - ***Context*** matters as well. Language might mean different things in a legal document, financial report, news article, etc.

b. Introduction to textual analysis

Text is unstructured

- The context matters for interpretation
- There are many ways to express the same meaning
  - For instance:

  1. "We do not expect high growth"
  2. "High growth is unrealistic"

- An algorithm that focuses on unigrams (single words) might pick up "growth" and "high" as the most informative, but the inference likely would be the opposite of the true meaning
- Bigrams (adjacent two-word combinations) would not fare much better…
- Even a four-gram would require *<u>very</u>* sophisticated code to get at the right interpretation

Thus, noise is a big problem when trying to find meaningful classifications of text data

b. Today's plan

Modest goal

- Learn basics of dealing with text in R (through the tm-package; pdf posted under Week 7)
- Think about how to create meaningful signals out of text
- Starting point for your future learning (could easily have a whole class dedicated to this topic alone)

Use 10-K (annual reports) from the SEC's EDGAR database as our data example

- https://www.sec.gov/edgar/searchedgar/companysearch.html
- One can design a web-crawler to download data (ftp-access will be closed as of end-of-year)
- I have downloaded all of Apple's (AAPL) available annual reports and posted them to a folder on CCLE under Week 7

b. An example of an annual report

Apple Inc.
Form 10-K
For the Fiscal Year Ended September 24, 2016

UNITED STATES
SECURITIES AND EXCHANGE COMMISSION
Washington, D.C. 20549

Form 10-K

(Mark One)
☒ **ANNUAL REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE SECURITIES EXCHANGE ACT OF 1934**
For the fiscal year ended September 24, 2016
or
☐ **TRANSITION REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE SECURITIES EXCHANGE ACT OF 1934**
For the transition period from __________ to __________ __________________________
Commission File Number: 001-36743

Apple Inc.

(Exact name of Registrant as specified in its charter)

UCLA Anderson
SCHOOL of MANAGEMENT

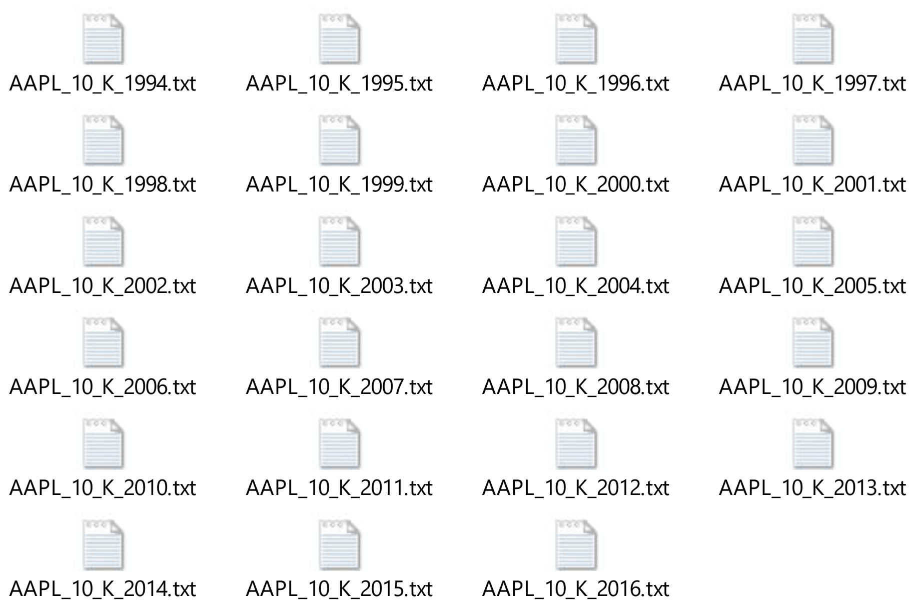
b. An example of an annual report

Note: over 70 pages of dense financial information. And that's just one report…

b. The Corpus

The ***Corpus*** is the full set of text data (in this case, 10-K's) you are working with

```
> # need text mining package
> require(tm)

> # Read in text files
> text_dir <- file.path("D:/llochsto/Dropbox/Data Analytics/Data", "TextData")
> main_corpus <- Corpus(DirSource(text_dir))
```



main_corpus[3] corresponds to the annual report in 1996

main_corpus[23] corresponds to the annual report in 2016

etc.

b. The Corpus needs to be pre-processed

- The computer does not know that “investment” is a word and that “\t @yyp&&$” is not.
- The computer sees “these” as an equally significant word as “gains”
- The computer sees “invest” as different from “invest.” and “invest,” and “invests” and “ invest” and “invest “ and “in-vest” and “Invest” etc.
- The computer sees numbers and letters all as asci code.
- The computer sees “ “ as a two-letter word and “   “ as a three-letter word
- Etc.

UCLA Anderson SCHOOL of MANAGEMENT

b. Pre-processing through the tm-package

```
➢ # remove punctuation
➢ main_corpus <- tm_map(main_corpus, removePunctuation)

➢ # remove special characters
➢ main_corpus <- tm_map(main_corpus, content_transformer(gsub),
                  pattern = "\t", replacement = " ")
➢ main_corpus <- tm_map(main_corpus, content_transformer(gsub),
                  pattern = "[^a-zA-Z0-9 ]", replacement = " ")
```

removePunctuation removes "." "," ";" ":" "?" etc.

You can remove any undesired characters using the gsub function

- Examples: html-markups such as "\t"
- Regular expression [^a-zA-Z0-9] means remove everything that is not (^) a letter or number or space

b. Pre-processing (cont'd)

```
➢ # remove numbers (we have those better represented in CompuStat)
➢ main_corpus <- tm_map(main_corpus, removeNumbers)

➢ # convert all to lowercase, so word is recognized as the same with arbitrary capitalization
➢ main_corpus <- tm_map(main_corpus, tolower)

➢ # remove particular words that you know are irrelevant noise
➢ main_corpus <- tm_map(main_corpus, removeWords, c("table of contents","sec","securities exchange commission",
  + "united states"))
➢ main_corpus <- tm_map(main_corpus, removeWords, c("company","company's","financial", "september", "net",
  + "securities", "including", "inc", "billion", "million", "assets", "operating", "statements", "tax"))
➢ main_corpus <- tm_map(main_corpus, removeWords, c("may","notes","can","changes","cost","will","also","rate",
  + "rates","equity","available","certain","results","relative"))
```

The words above are all words that occur very often in financial reports without any special significance, along with some SEC-filing-specific word combinations

In general, you want to clean up the document as much as possible to remove words or combinations of words that have little signal-value. Pro-forma language is a good example of such noise.

Also, if you have a particular goal (for instance, trying to predict future firm growth) it makes sense to delete words that are unrelated to this. For instance, there are chapters of the report that are more important than others, such as Item 7 – Manager's discussion of financial condition and outlook, that one may want to pay special attention to, while Item 4, Mine Safety Disclosures, is probably unimportant.

UCLA Anderson SCHOOL of MANAGEMENT

b. Pre-processing: Stopwords and Stemming

```
➢ # remove "stopwords" (e.g., and, to, a, as, the, ...)
➢ main_corpus <- tm_map(main_corpus, removeWords, stopwords("english"))

➢ # stemming words, i.e., keep only the stem so as not to differentially count
➢ # investing, invest, invests
➢ # taking out common word endings such as 'ing', 'es', and 's'
➢ require(SnowballC)
➢ main_corpus <- tm_map(main_corpus, stemDocument)
```

Stopwords are standard prepositions, identifiers, and other very common words that are typically not useful and just adds noise

- Examples include "a", "the", "it", etc.
- Different *stopword* libraries exist, the above is just the one that comes with the tm-package for "English"

Stemming

- A *very* important task
- Keeps only the stem of the word (invest is kept for invest, invests, invested, investment, investing)

UCLAAnderson
SCHOOL of MANAGEMENT

b. Pre-processing: extra space and saving

```
➢ # finally, let's get rid of all the extra white space in the document so all words are only
➢ # separated by one space
➢ main_corpus <- tm_map(main_corpus, stripWhitespace)

➢ # if you want to save as one big corpus (in mycorpus.txt), do the below
➢ #writeLines(as.character(main_corpus), con="TextData/mycorpus.txt")
```

Extra whitespace (“ “) occurs very often and stripWhitespace gets rid of a lot of it, though not all

You may want to save the pre-prosessed data as one big corpus, not retaining the information about what article/document each word came from. writeLines achieves this.

We don’t want this, however, as we will be interested in 10-K information as it is revealed over time.

UCLA Anderson SCHOOL of MANAGEMENT

b. Text analysis: DocumentTermMatrix

```
➢ # next, organize words into matrix, which then can be used for analysis
➢ corpus_matrix <- DocumentTermMatrix(main_corpus)
➢ inspect(corpus_matrix[1:10,1:10])
```

Docs	aapl	abil	accept	access	accompani	accord	account	accru	accrual	accu
AAPL_10_K_1994.txt	1	17	8	3	7	14	52	14	2	1
AAPL_10_K_1995.txt	1	22	8	1	7	21	56	9	1	0
AAPL_10_K_1996.txt	1	27	7	6	9	27	69	18	2	0
AAPL_10_K_1997.txt	0	0	0	1	0	8	0	0	0	0
AAPL_10_K_1998.txt	1	29	9	4	10	20	79	16	1	0
AAPL_10_K_1999.txt	1	21	17	5	9	38	103	21	3	0
AAPL_10_K_2000.txt	1	22	12	2	8	12	88	14	1	0
AAPL_10_K_2001.txt	1	25	14	8	7	23	107	14	2	0
AAPL_10_K_2002.txt	1	31	16	15	9	17	113	15	5	0
AAPL_10_K_2003.txt	1	32	17	15	9	25	144	19	13	0

corpus_matrix contains all words along with their count, rows give the document, columns give the terms

Some words have zero frequency, meaning they do not appear in the 10 documents we inspect above

b. Text analysis: Overall frequency

```
➢ # organize words by frequency
➢ freq <- colSums(as.matrix(corpus_matrix))
➢ ord_corpus <- order(freq)

➢ # see most common words
➢ freq[tail(ord_corpus)]
   option share stock compani sale product
    3228  3674  4360   4774   5961  6421
```

Taking the colSums, we get the word count across all documents

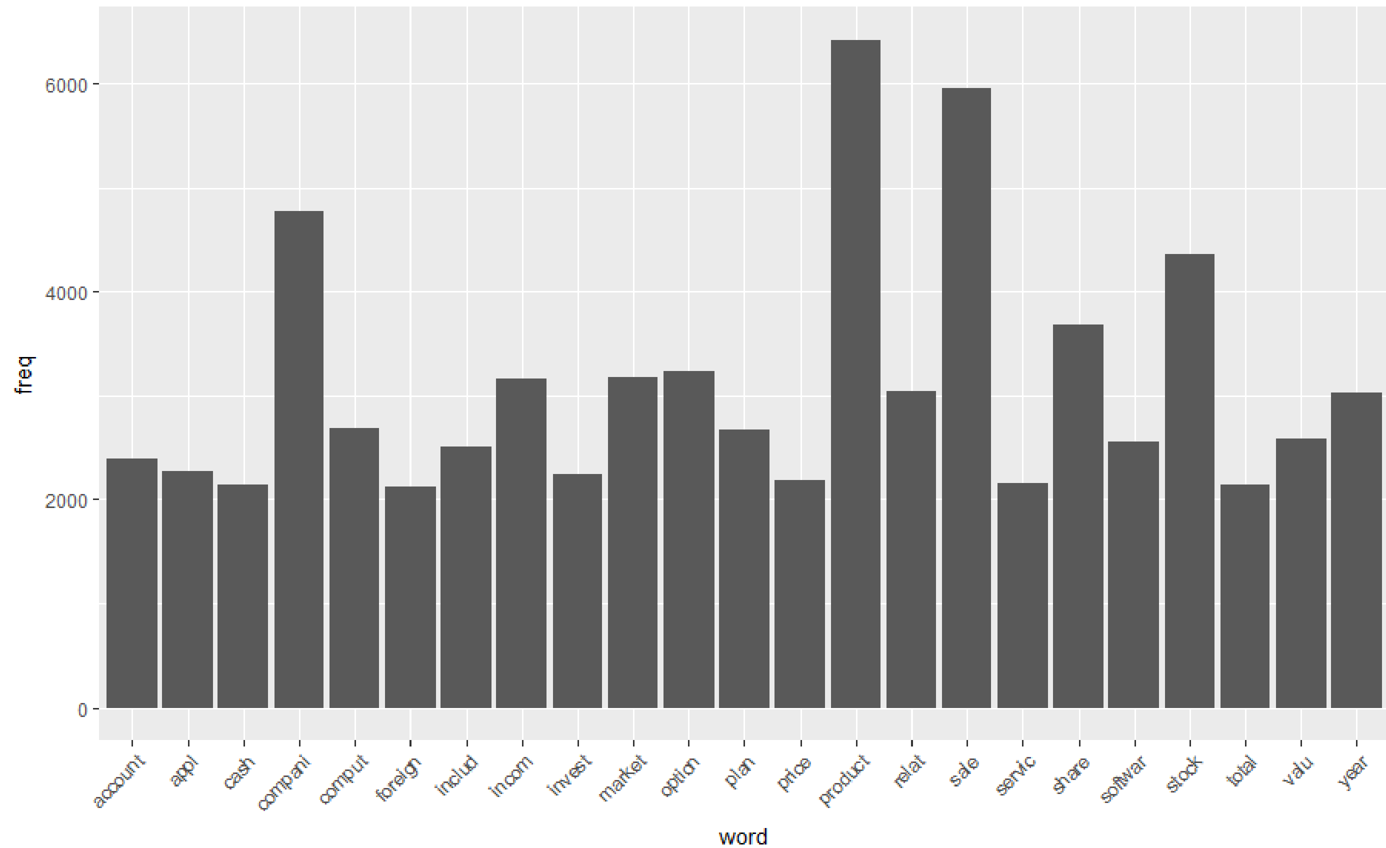
Are these words informative?

- Seem mainly to reflect the most common discussions in an annual report
- While counting frequency may tell you something, we have to get a little more sophisticated it would seem…
- But, let's continue with this a little more to learn more about the nature of the data

UCLA Anderson
SCHOOL of MANAGEMENT

b. Text analysis: Frequency plots

```
# identify words that appear frequently
# create a convenient data.frame
word_freq <- data.frame(word=names(freq), freq=freq)

# plot most frequent words along with frequency
p <- ggplot(subset(word_freq, freq>2100), aes(word, freq))
p <- p + geom_bar(stat = "identity")
p <- p + theme(axis.text.x=element_text(angle=45,hjust=1))
p
```



b. Text analysis: Wordcloud

```
➢ # plot wordcloud, a net way to express the data, size of font relates to frequency
➢ require(wordcloud)
➢ # plot 100 most frequent words, and add some color
➢ wordcloud(names(freq), freq, max.words=100, rot.per = 0.2, colors=brewer.pal(6, "Dark2"))
```

UCLA Anderson SCHOOL of MANAGEMENT

c. Mapping to signal: Growth expectations

```
➢ # let's search for frequency of a pre-set list of words we perceive as related to high growth expectations
➢ freq[c("invest","growth","grow","high","strong","lead","good")]
   invest   growth   grow   high   strong   lead   good
    2239     472      14    508     281      85     122
➢ freq[c("loss","weak","low","poor","uncertain","under","disappoint")]
    loss     weak     low     poor uncertain  under disappoint
    1939     116      59      29       85      270       2

➢ # let's loop through the reports by year
➢ higrowth_words = NULL
➢ logrowth_words = NULL
➢ for (j in 1:23)
  + {corpus_matrix <- DocumentTermMatrix(main_corpus[j])
  + # organize words by frequency
  + freq <- colSums(as.matrix(corpus_matrix))
  + higrowth_words = rbind(higrowth_words,freq[c("invest","growth","grow","high","strong","lead")])
  + logrowth_words = rbind(logrowth_words,freq[c("loss","weak","low","poor","uncertain","under")]) + }

➢ # add rows to get a score by year
➢ hi_growth <- rowSums(higrowth_words, na.rm = TRUE)
➢ lo_growth <- rowSums(logrowth_words, na.rm = TRUE)
```

Generally, if we have an idea of how to classify words, we will get much stronger results

Here, we try to classify based on high or low growth expectations, as expressed by management through the annual report

“Letting the data speak” requires lots of data and very high-level code, easy to end up in a black-box setting and get lots of noise in the end…

c. Mapping to signal: Scaling

```
➢ # display scores
➢ hi_growth
  [1] 83 91 72 3 106 148 153 190 263 252 244 225 190 169 185 117 176 163 172 175 160 150 124
➢ lo_growth
  [1] 33 48 88 6 139 138  84 134 194 204 140 118 111 101 102  86 102 104 104 110 116 119 127
```

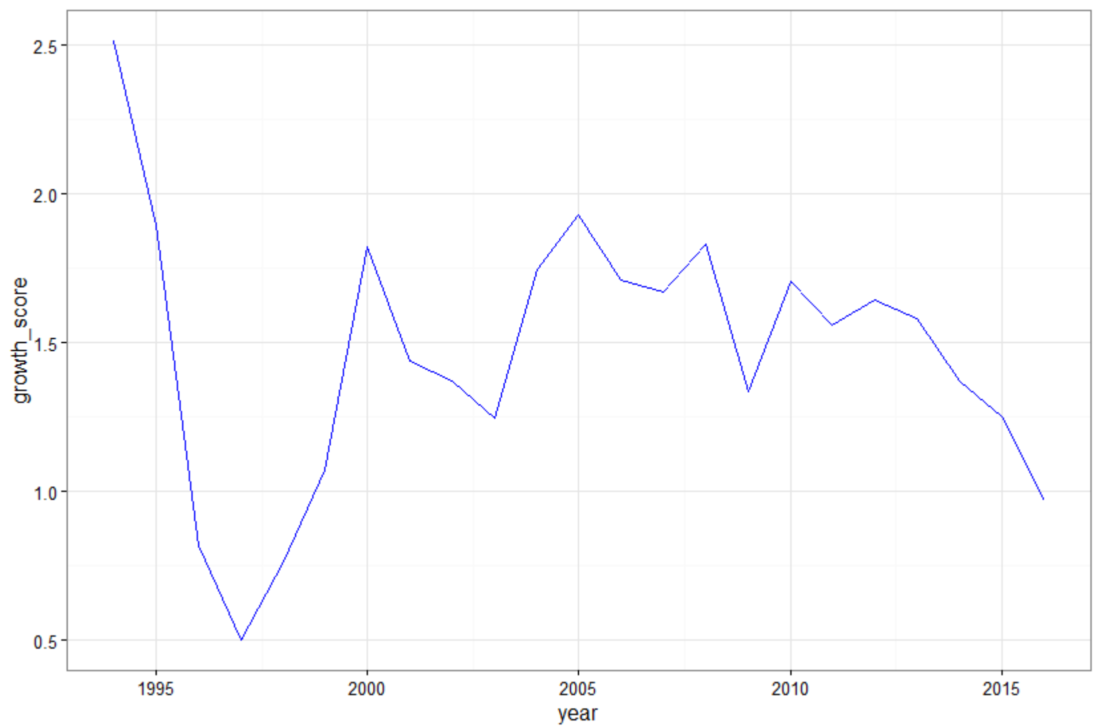
Often, there are nonstationarities in text data

- In our case, annual reports may have gotten longer over time, there may have been particular sections added or deleted.
- This makes it hard to compare levels of word counts across years -> we don't know if we are comparing apples to apples..!
- Solution: scale using a within-year word count benchmark
- Generally, creating *word-rates* can be helpful where the normalization is based on belong to a group that one ex-ante believes is comparable
- On the next slide, we create a high over low growth score ratio each year and use this as our normalized signal

UCLA Anderson SCHOOL of MANAGEMENT

c. Mapping to signal: Creating a growth score

```
➢ # create hi over lo metric
➢ growth_score = hi_growth / lo_growth
➢ growth_score
[1] 2.5151515 1.8958333 0.8181818 0.5000000 0.7625899 1.0724638 1.8214286 1.4179104 1.3556701 1.2352941
    1.7428571 1.9067797 1.7117117 1.6732673 1.8137255 1.3604651 1.7254902 1.5673077 1.6538462 1.5909091
    1.3793103 1.2605042 0.9763780

➢ # plot growth score versus year
➢ year <- c(1994:2016)
➢ qplot(year,growth_score, col=I("blue"), geom = "line") + theme_bw()
```



c. Is the predictive relation significant?

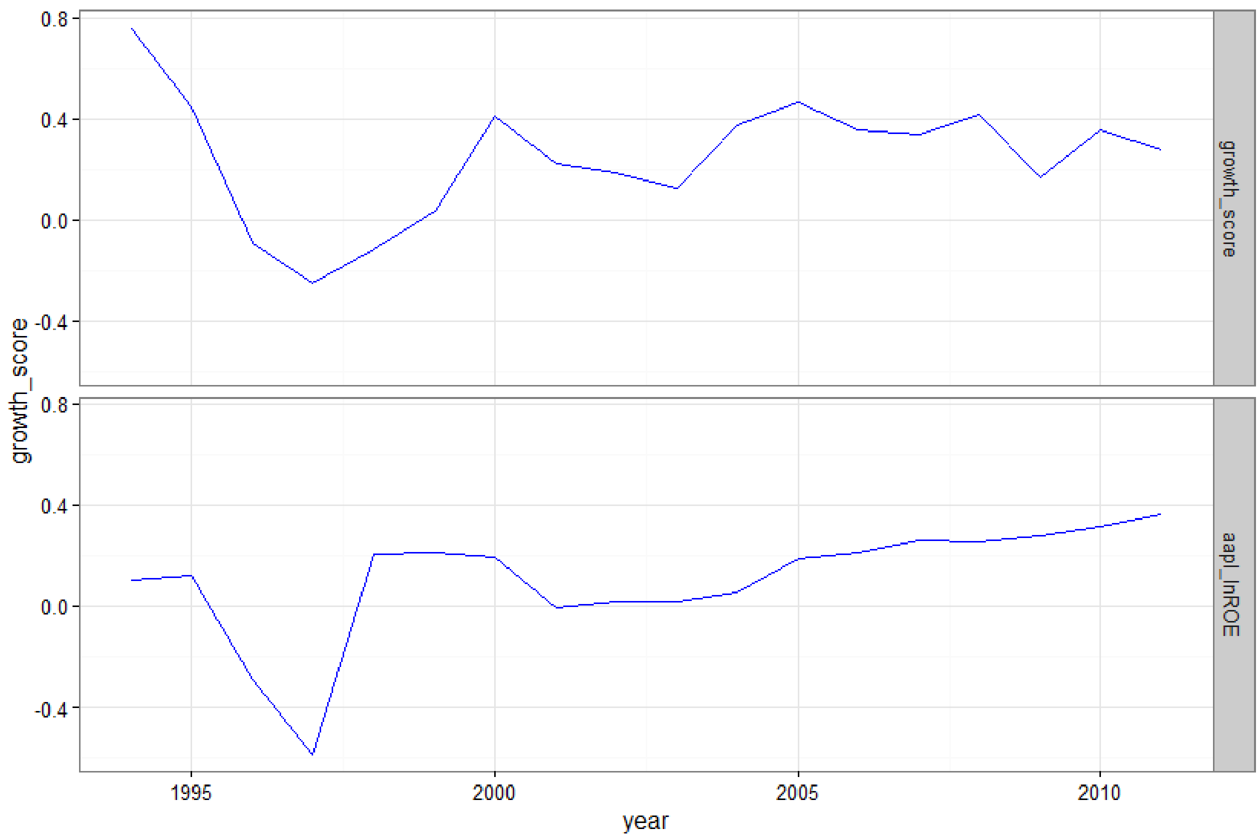
```
➢ # regress to see if a statistically significant forecasting relation
➢ reg <- lm(aapl_lnROE[2:19]~growth_score[1:18])
➢ summary(reg)
```

```
Call:
lm(formula = aapl_lnROE[2:19] ~ growth_score[1:18])

Residuals:
     Min       1Q   Median       3Q      Max 
-0.42665 -0.09162 -0.01116  0.13602  0.29382 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)  
(Intercept)        -0.29175    0.15215  -1.918   0.0732 .
growth_score[1:18]  0.26648    0.09705   2.746   0.0144 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1953 on 16 degrees of freedom
Multiple R-squared:  0.3203,   Adjusted R-squared:  0.2778 
F-statistic:  7.54 on 1 and 16 DF,  p-value: 0.01436
```

UCLA Anderson
SCHOOL of MANAGEMENT

c. Compare growth score to next year Apple ROE

```
➢ # plot growth score in year t versus lnROE of Apple in year t + 1
➢ aapl_lnROE <- c(0.0324513, 0.1050476, 0.1230894, -0.2899466, -.5851619,.2052859,.2108069,
 + .1922505,-.0059545,.0151465,.016474,.0559076,.1884116,.2130643,.2623312,.2519465,.2787039,
 + .3133555,.3649631,.3712058,.2820229)
➢ reg_data <- data.frame(year = year[1:18], growth_score = growth_score[1:18] / 2 - 0.5, series = "growth_score")
➢ reg_data <- rbind(reg_data, data.frame(year = year[1:18], growth_score = aapl_lnROE[2:19], series = "aapl_lnROE"))
➢  > qplot(year,growth_score,data=reg_data,facets=series~.,col=I("blue"), geom="line") + theme_bw()
```



UCLAAnderson
SCHOOL of MANAGEMENT

c. Is the predictive relation significant?

There appears that we have been able to extract a useful signal from the word data

However, it is not clear from this result whether this is useful above and beyond standard metrics such as the b/m ratio (which also predicts future growth)

That is the bar a textual analysis product must clear…

Useful resource: https://sraf.nd.edu/textual-analysis/resources/

UCLA Anderson SCHOOL of MANAGEMENT

d. Ravenpack and sentiment scores

News Sentiment: An attitude or opinion expressed in media (DJ Newswires, WSJ, social media)

Ravenpack:

- Processes unstructured media reports/news into structured data feeds
- Provides a number of quantitative Sentiment scores

MSCI

- Takes Ravenpack Sentiment scores and creates equity factors for risk models

Key Questions:

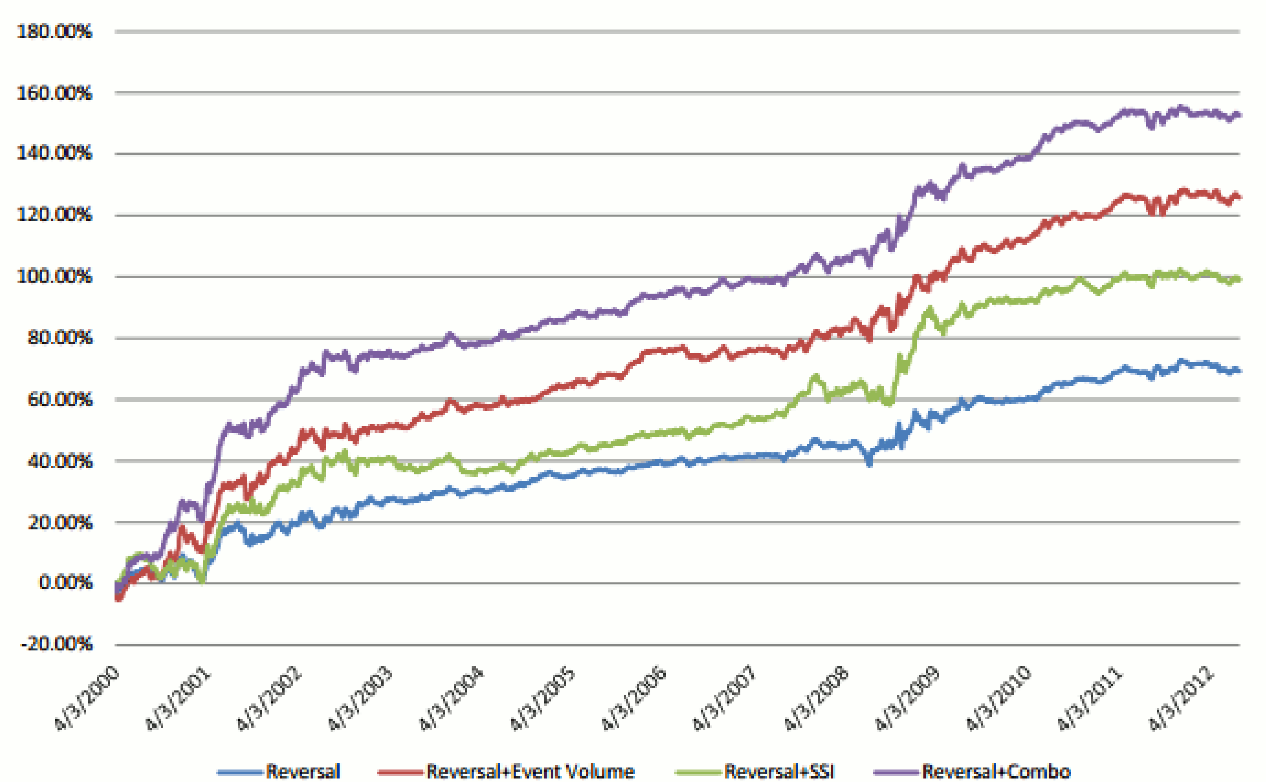
- Does Ravenpack data provide additional information for predicting risk/returns in presence of other well known risk factors?
- What is the investment horizon for using the data (long-term, medium-term, short-term)?

https://www.ravenpack.com/page/ravenpack-news-analytics/

https://www.youtube.com/watch?time_continue=26&v=HDwf_tfYCv4

d. Ravenpack “Overlay” strategies

Fig 5: Cumulative Return of Short-Term Reversal Strategy Enhanced by RavenPack News Analytics-Zero Transaction Cost



This figure shows the cumulative return of the short-term reversal strategy enhanced by RavenPack news analytics without considering the transaction cost. The stock universe includes all S&P 500 index constituents.

SOURCE: RavenPack, 2013

UCLA Anderson
SCHOOL of MANAGEMENT

e. The group projects

- Up to 5 students per group
- Find an idea for a predicting returns, return variances, or return covariances
- Obtain and clean relevant data (whatever you want: 10-K's, news, other)
- Show econometric techniques used (regularizations, decision trees and boosting) and explain why well-suited for problem. Show forecasting results and, if relevant, any trading profits (e.g., Sharpe ratio)
- Present in-class assuming presentation is for the fund management as a pitch to use this new model for a new fund or as an overlay on an existing strategy
- Presentations are to be 15 mins per group + 5 min class discussion on Friday May 31st and Monday June 3rd.