UCLA Anderson
SCHOOL of MANAGEMENT

MGMTMFE 431:

Data Analytics and Machine Learning

Topic 6:
Textual Analysis and Trading Strategies

Spring 2025

Professor Lars A. Lochstoer

Structured vs. unstructured data

Structured data

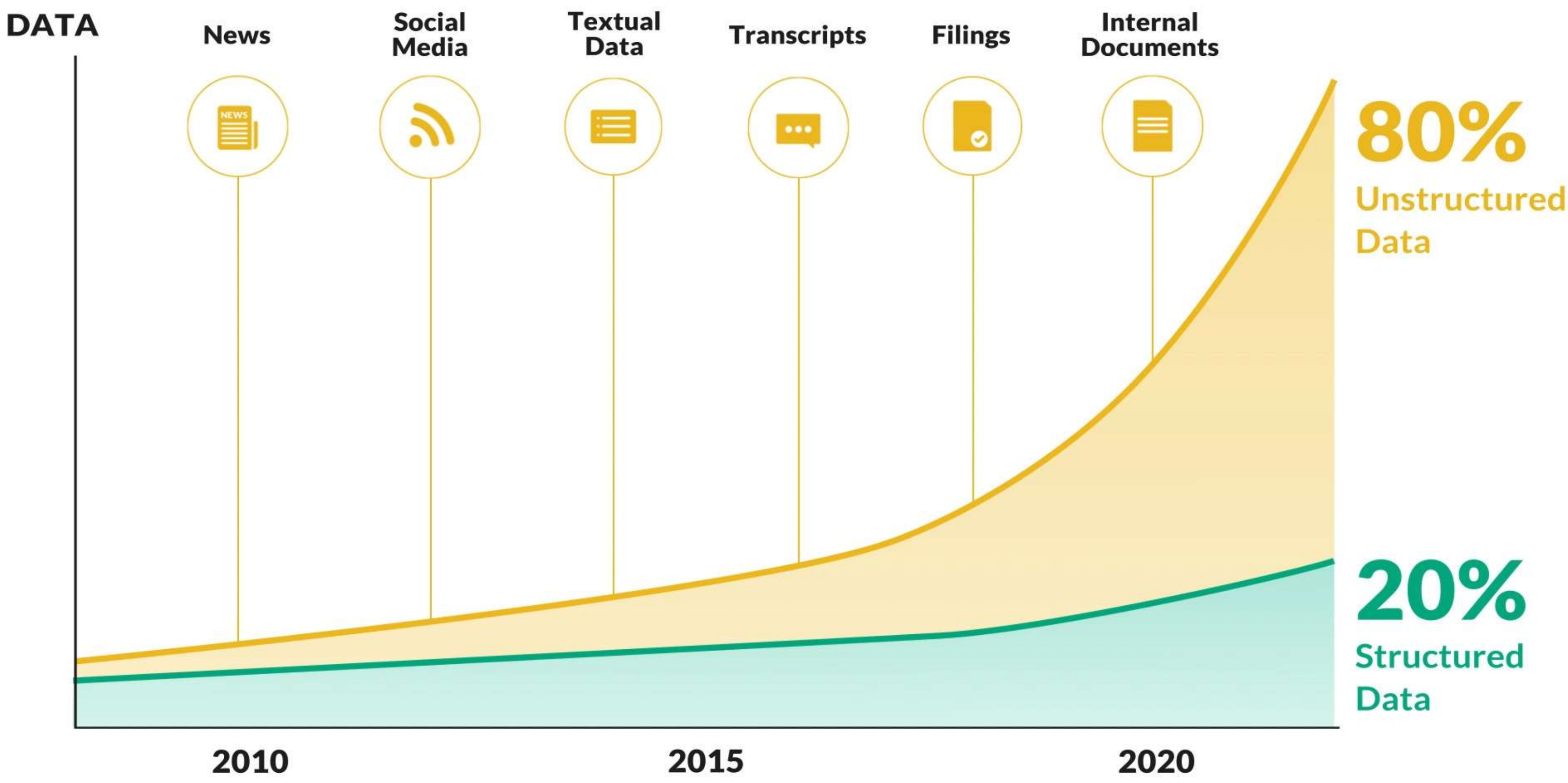
- Asset prices, returns, accounting numbers, macroeconomic series, volume, inventory, earnings, dividends, etc.
- Easy to read and input into models
- Most data used historically are of this form

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

UCLAAnderson
SCHOOL of MANAGEMENT

Growth of unstructured data



UCLA Anderson SCHOOL of MANAGEMENT

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.

UCLA Anderson SCHOOL of MANAGEMENT

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 sophisticated code to get at the right interpretation

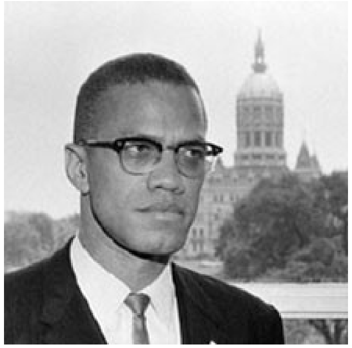
Thus, noise is a big problem when trying to find meaningful classifications of text data

First: A Review of Research on News and the Stock Market

UCLA Anderson SCHOOL of MANAGEMENT

Motivation

- “The media’s the most powerful entity on earth.”
    - Malcolm X, American human rights activist



- “I fear three newspapers more than a hundred thousand bayonets.”
    - Napoleon Bonaparte, first emperor of France

- “Whoever controls the media controls the mind.”
    - Jim Morrison, lead singer of The Doors

Importance of Media in Finance

- Old view: Financial media doesn’t matter
  - Capital and product markets are highly efficient
  - Prices and quantities already reflect all information

- Modern view: Media affects and reflects behavior
  - Investors, managers, and consumers are human
  - Humans have limited information processing abilities
    - The way information is transmitted matters
  - Judgmental errors can affect market outcomes
    - Limits to arbitrage in asset markets; frictions in other markets

Roles of Media in Finance

- Attracts attention
  - To important current events
- Conveys information
  - About the macroeconomy, industries, and firms
  - About politics, laws, and regulations
- Influences beliefs
  - Provides compelling and memorable stories

News Selection and Promotion

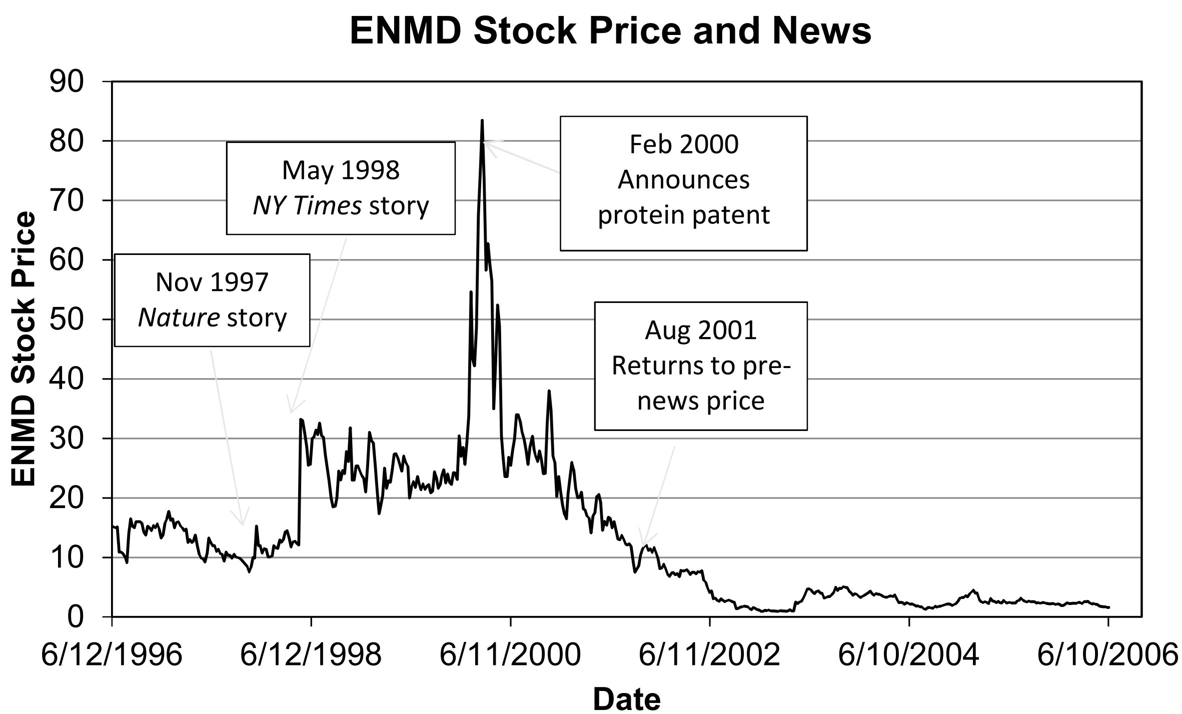
- People notice and remember only a few events
  - We have finite attention and imperfect memories
  - Millions of events occur around the world every day
- Media focuses attention and aids memory by exploiting cognitive heuristics
  - We attend to <u>salient</u> stimuli that stand out
  - We recall memories that are easily <u>available</u>
    - Journalists try to find or construct **dramatic stories**

Anatomy of a Headline

- Salience
  - Big and bright
  - Evocative language
    - Strips, churn, squirm
- Availability
  - Story-telling
    - Last-minute standoff
    - Wild ride
  - Drama
    - Unprecedented

The Impact of Media Attention

- Huberman and Regev (2001, *JF*) study EntreMed
  - Possible cure for cancer elicits *many* reactions
  - EntreMed investors seem to ignore past reactions



UCLA Anderson SCHOOL of MANAGEMENT

Investor Overreaction

- Attention promotes overreaction
  - "Nothing is as important as you think it is while you're thinking about it." - Daniel Kahneman, Nobel laureate

- EntreMed investors focused on salient good news from newspapers and ads
  - They ignored subtle statistical and economic info
    - EntreMed: 90% of new drugs don't receive FDA approval
  - But it's difficult to <u>know</u> whether prices were inefficient

Staying “Abreast of the Market”

- Many journalists (and traders) claim to know why the market moved—at least, in hindsight

Fed policy

Housing market

Oil supply

Exchange rates

Innovation

War

UCLA Anderson SCHOOL of MANAGEMENT

A Simple *Content* Measure

- Tetlock (2007, *JF*) measures the frequency of positive and negative words in a daily column
  - WSJ – "Abreast of the Market" column
  - Negative words in psychosocial dictionaries include:
    - "fear," "worry," "disappoint," "collapse," "flaw," and "ruin"
  - Compute the relative frequency of each category
    - E.g., **negativity** = negative words / total words

- Stock prices react more to negative words
- Does the market respond appropriately?

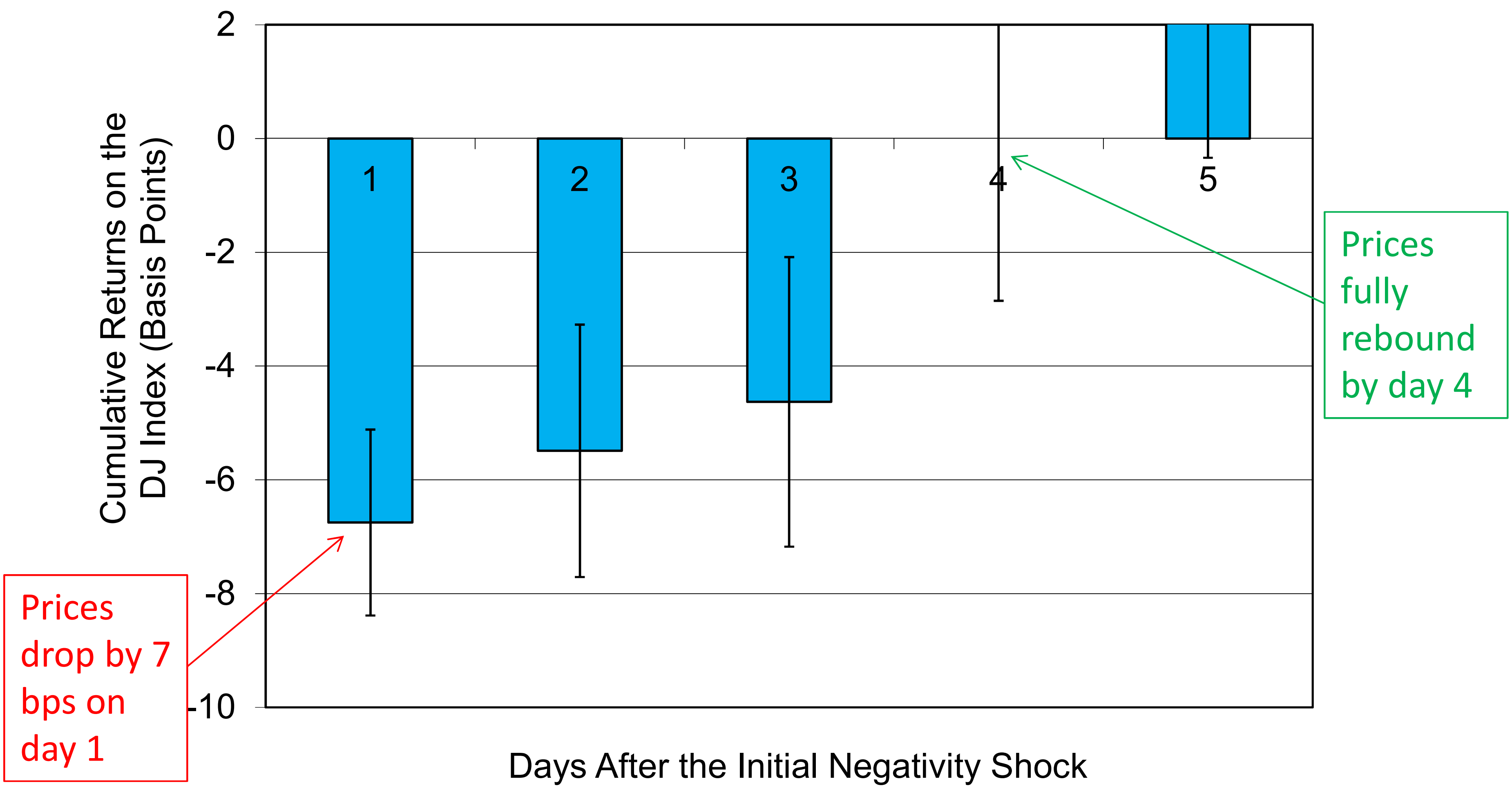
Example: Quantifying Content

- WSJ "Abreast of the Market" column on Feb 17, 2009
  - Headline: *Market's `Hope Balloon'* ***Loses*** *Air; Tepid Upturns Haven't Stopped the Slide*
  - Financial markets are supposedly driven by two **competing** forces: **fear** and greed. **Fear** just made another **grab** for the steering wheel.
  - **Disappointment** with the government's planned credit-market bailout and **concerns** that the $787 billion stimulus plan won't jolt the economy fast enough snuffed out the budding stock-market rally. Now investors are **worried** that stocks could fall back to their November **lows** -- and possibly even farther.

- Method: Compute negativity in each day's column
  - E.g., 9 negative / 82 words = 11.0% -- much higher than usual

UCLA Anderson
SCHOOL of MANAGEMENT

Negativity Predicts DJ Index



Why Do Investors Overreact?

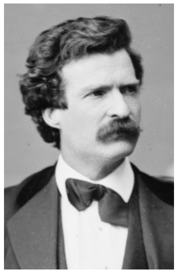
- Journalists’ powerful techniques influence beliefs
  - Use evocative imagery
  - Use emotional language
  - Focus on people

- Study of *WSJ* “Abreast of the Market” (1970-2007)
  - Different journalists write the column each day
  - Journalists differ in their writing styles (e.g., optimism)
  - ***Stock prices increase after days with an optimistic author***
  - See Dougal, Engelberg, Garcia, and Parsons (2012, *RFS*)

UCLA Anderson SCHOOL of MANAGEMENT

Which News Is Informative?

- “If you don’t read the newspaper, you are uninformed; if you do … , you are misinformed.”
    - Mark Twain, writer



- “It’s amazing that the amount of news that happens in the world every day just exactly fits the newspaper.”
    - Jerry Seinfeld, comedian

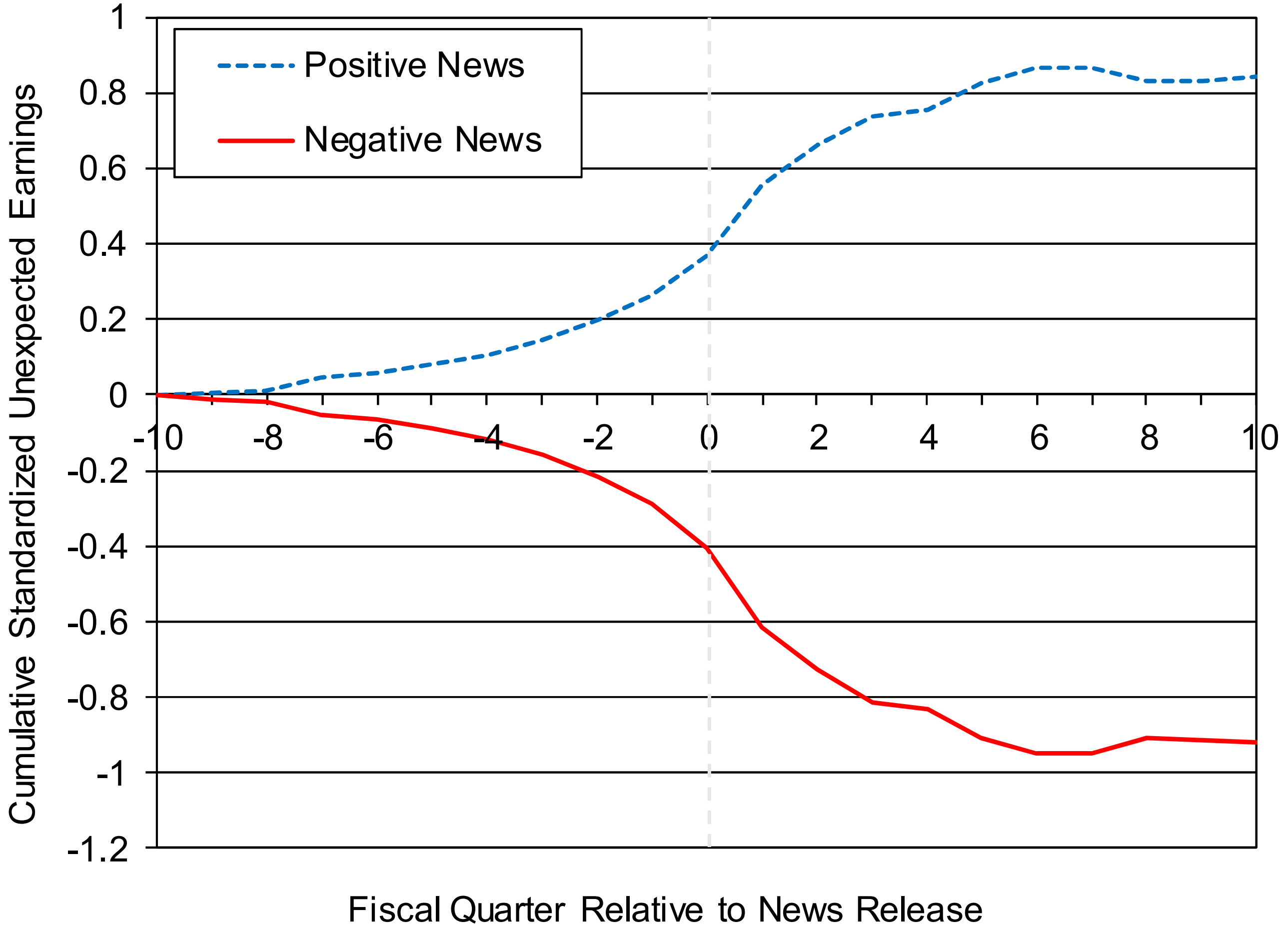
UCLA Anderson SCHOOL of MANAGEMENT

How Informative Is Firm News?

- Much financial news is genuinely informative
  - May be related to firms' fundamental values (*x*)
  - Most news about firms doesn't make the front page

- Tetlock et al. (2008, *JF*) analyze firm news (cross-sectional analysis)
  - *DJ* newswire and *WSJ* stories about S&P 500 firms
  - Compute daily negativity scores for these stories
  - Examine outcomes before and after negative stories
    - Firms' earnings
    - Firms' stock prices

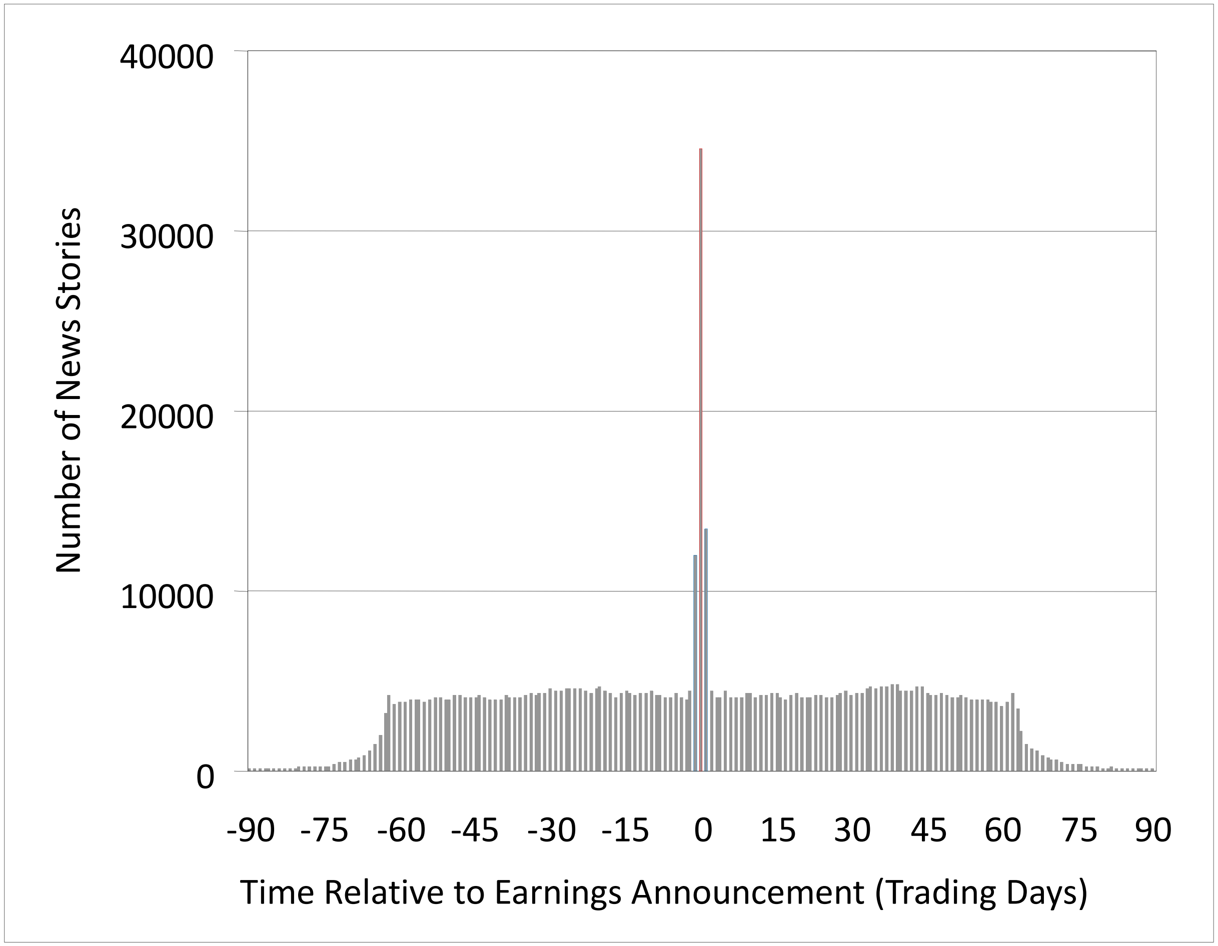
News Content Predicts Firm Earnings



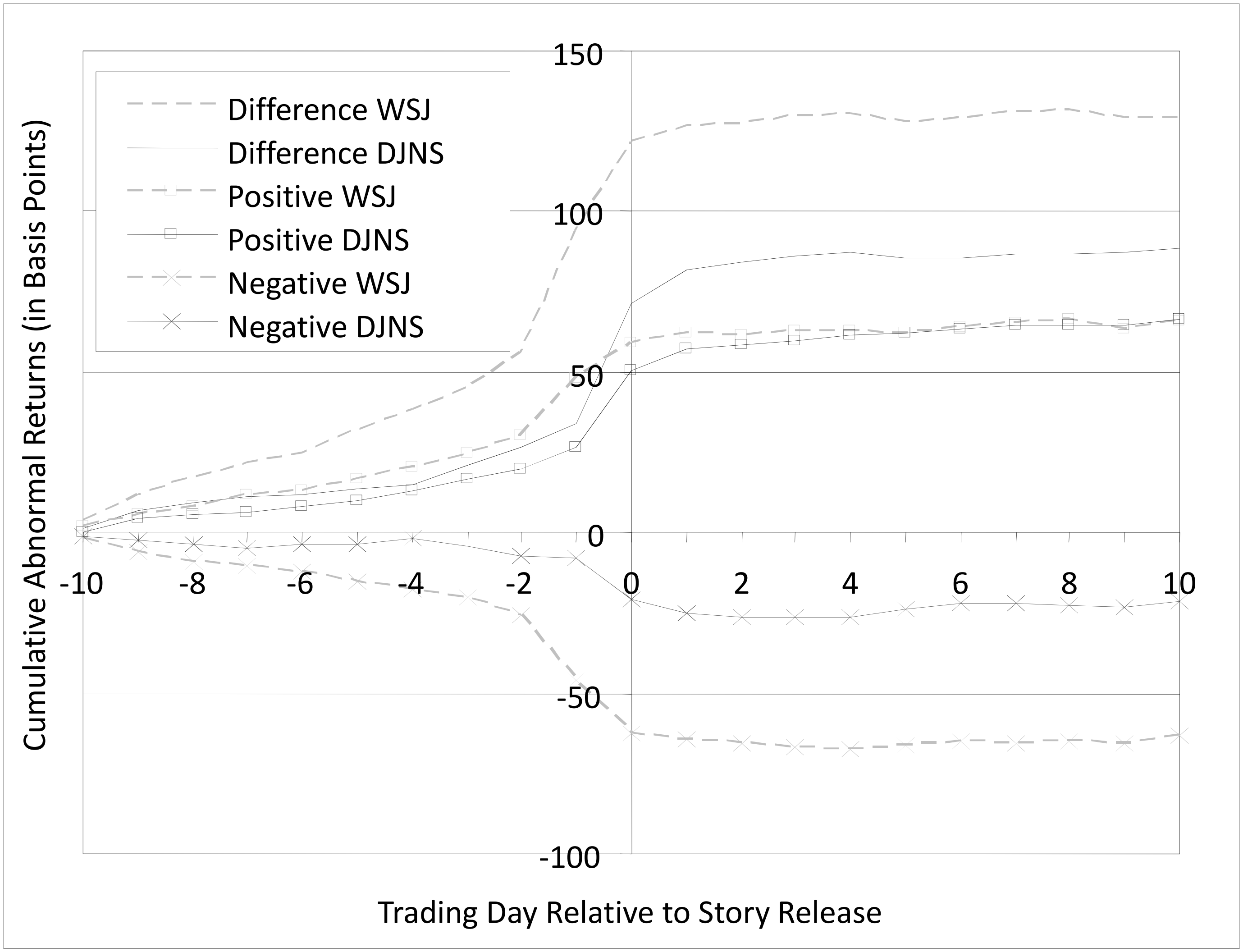
UCLA Anderson SCHOOL of MANAGEMENT

Fundamental Content Is Important

- Coverage patterns suggest earnings news is important



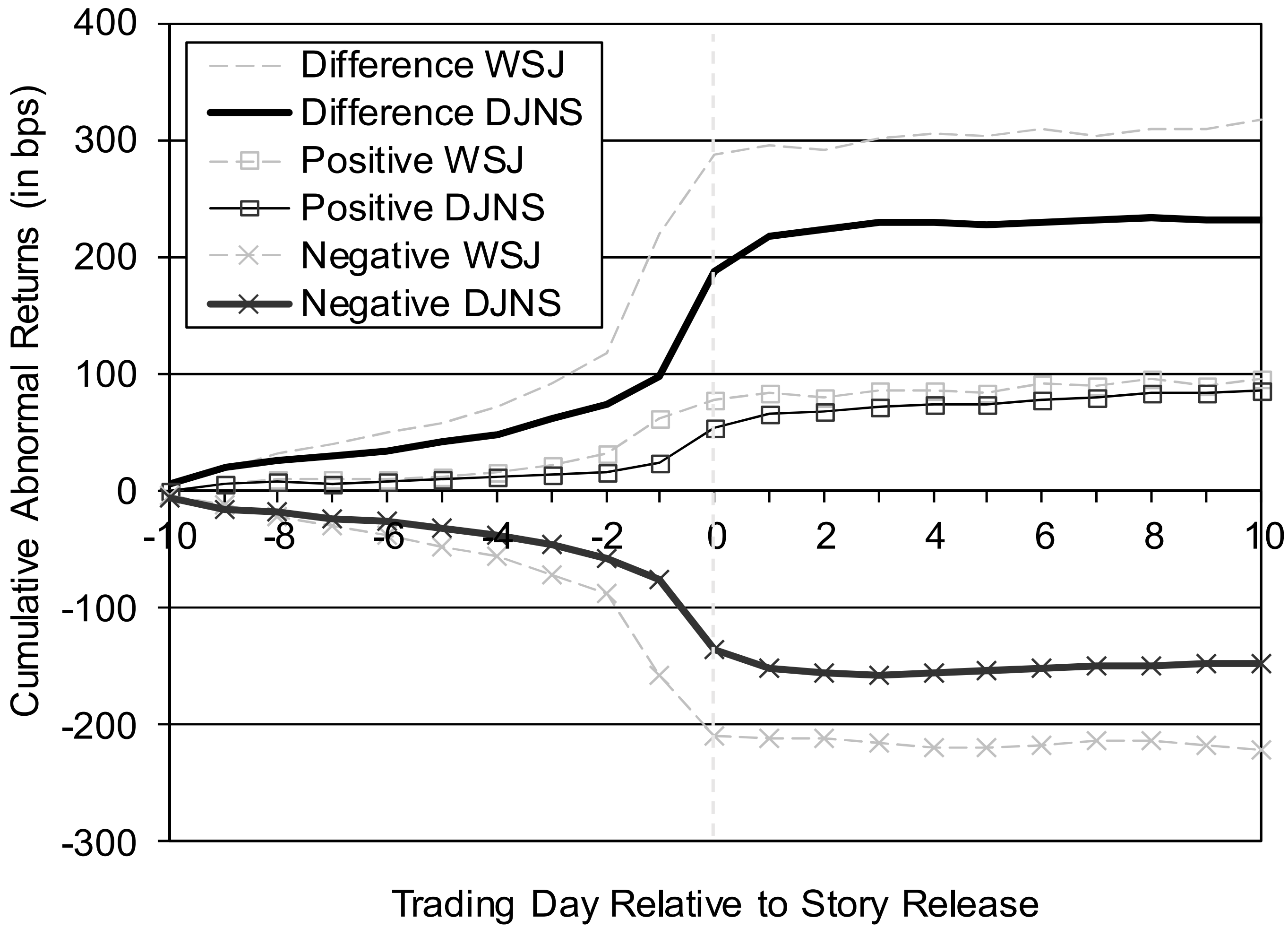
Brief Underreaction to Information



UCLA Anderson
SCHOOL of MANAGEMENT

Underreaction to Earnings News

- Focus on stories that mention "earnings"



Interpreting Firm News

- Investors can't attend to all relevant news
  - Underreact to relevant news that's not featured
  - Underreaction increases with news relevance
    - Stories about earnings are especially relevant for firm value

- Lack of attention can cause additional biases
  - Failure to distinguish new news from "stale" news
    - Market prices should react more to new news
    - A.J. Liebling – "People everywhere confuse what they read in the newspapers with news."

Recognizing Stale News

- Tetlock (2011, *RFS*) study of stale news
  - Data: DJ news archive from 1996 to 2008
  - Staleness = similarity of a story to previous stories
    - E.g., 90 words overlap / 150 words = 60% staleness

- Key findings
  - Stock prices react less to stale stories
    - Presumably, stale stories are less informative
  - Still, prices overreact to stale stories
    - Price reactions to stale stories tend to reverse

UCLA Anderson
SCHOOL of MANAGEMENT

Extreme Case of Stale News

- Consider market activity in United Airlines' stock
  - United Airlines filed for bankruptcy in 2002
    - Two published studies of the market reaction(s) to this event

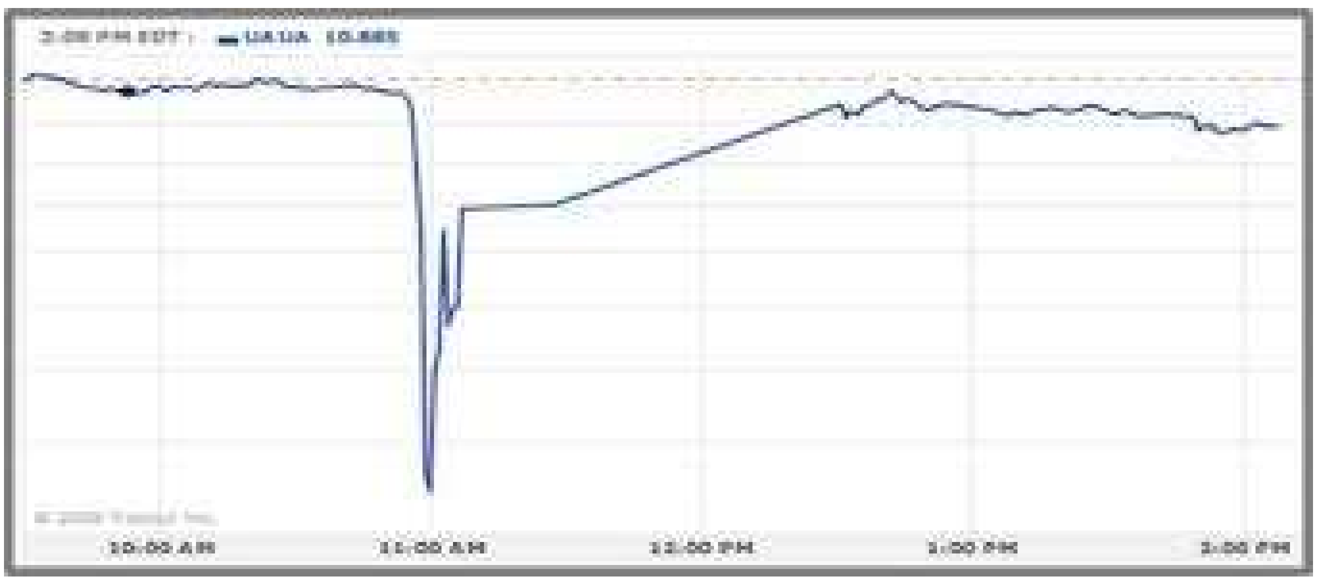
- 2002 United bankruptcy story was new
  - ~100% stock price decline; no rebound

- The firm exited bankruptcy in 2006
  - On Sept. 7, 2008, United's stock market cap is $1.6B

UCLA Anderson SCHOOL of MANAGEMENT

United Stock on Sept 8, 2008

- *Google News* posts a 6-year-old *Chicago Tribune* story about United's 2002 bankruptcy
  - United's stock falls 76% within minutes
    - United rebounds, but remains down 11% on the day



Key Lessons from Research

- Trading activity and price movements are related to news, but it's hard to link them

- Market prices reflect both news and noise
  - Overreact to non-information
    - Sensationalist news that grabs investor attention
    - False or stale news when investors aren't paying attention
  - Underreact to genuine information
    - Substantive news—e.g., news about earnings
    - News that's not featured—e.g., firm news in the back pages

Rest of today

a. Sentiment

b. Using text in regression-based forecasting models

Lecture note 6b: predicting mergers with text

UCLA Anderson
SCHOOL of MANAGEMENT

a. Sentiment

Investor sentiment

- Baker and Wurgler (Journal of Finance, 2006) construct an index intended to capture overall "investor sentiment"

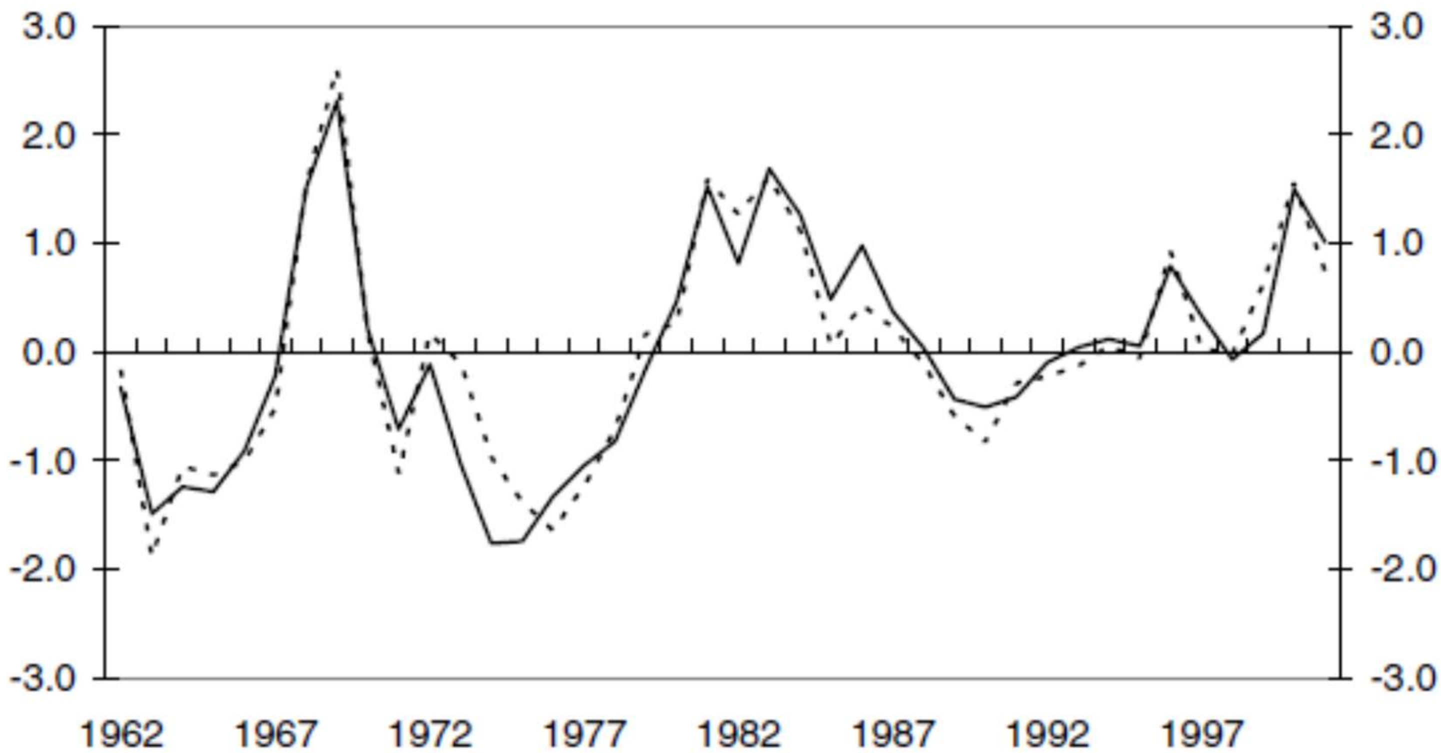
- Roughly speaking, are investors optimistic or pessimistic about business activity and growth?
  - When sentiment is high, they are too optimistic (irrational exuberance)

- Index is a composite of:
  - The closed-end fund discount, NYSE share turnover, the number and average first-day returns on IPOs, the equity share in new issues, and the dividend premium (see paper, pages 1655-1656, for more information)

- The authors find that when sentiment is low, subsequent returns on small stocks, young stocks, high volatility stocks, unprofitable stocks, non-dividend-paying stocks, extreme growth stocks, and distressed stocks are high.
  - High returns are relative to the return on stocks with 'opposite' characteristics; e.g. large stocks, old stocks, low vol, etc.

a. B&W Sentiment Index

Panel E. Sentiment index (SENTIMENT)

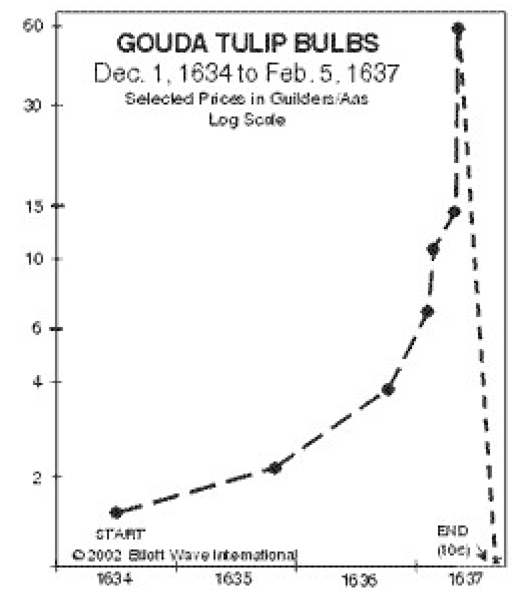


a. Sentiment as a general concept

- The idea of investor sentiment affecting valuations (and thus subsequent returns) is an old one, including mentions by Keynes (1936)
  - The Tulip Mania (1636 to 1937 things went truly nuts) is an early example of bubble driven by investor expectations

- Many investors now seek to identify investor sentiment at higher frequencies than B&W's index using alternative data sources
  - We mentioned Ravenpack in the first lecture as a provider of such sentiment indexes.

- In the following, we will create a sentiment indicator based on Dow Jones Newswire headlines.
  - This is Problem Set 6
  - Effectively, we are trying to create an index that helps predicting returns using Supervised Learning (with our usual regression-based forecasting methods)



UCLA Anderson SCHOOL of MANAGEMENT

a. News Data

- Many papers have used news data to capture investor mood and information
  - Presumably (a) investors get information in part from the news, and (b) presumably the news reflect in part what investors are discussing as current topics

- I have downloaded DJIA Headline News from www.kaggle.com
  - Kaggle.com is a really fun web-site that has cool data and machine learning and analytics code to analyze. If you haven't yet, I encourage you to check it out.

- The data is in DJIA_Headline_News.csv on BruinLearn under Week 6.
  - Data is from 8/8/2008 until 7/1/2016
  - There are 25 headlines each day
  - There is also an indicator of whether the Dow Jones go up or down that day
  - Next slide shows what data looks like

UCLA Anderson SCHOOL of MANAGEMENT

a. News Data

- An example of a headline:
  - "*Georgia 'downs two Russian warplanes' as countries move to brink of war*"

- Data example: (Top1 is top headline, Top2 is second to top headline, etc., until Top25)

Date	Label	Top1	Top2	Top3	Top4	Top5
8/8/2008	0	b"Georgia	b'BREAKIN	b'Russia T	b'Russian	b"Afghan
8/11/2008	1	b'Why wo	b'Bush pu	b"Jewish (	b'Georgiai	b"Olympic
8/12/2008	0	b'Rememl	b"Russia '	b'"If we h	b"Al-Qa'e	b'Ceasefir
8/13/2008	0	b' U.S. ref	b"When tl	b' Israel cl	b'Britain\'	b'Body of
8/14/2008	1	b'All the e	b'War in S	b'Swedish	b'Russia e	b'Missile

- Label is 0 if Dow Jones Index goes down during the same day as the headline, 1 otherwise

- Ultimate goal: can we construct a model that uses text (headline) data to predict Dow Jones returns?
  - Sentiment-related idea: If people become more optimistic (pessimistic), they buy (sell), and prices will go up (down).

a. The NLTK package

- NLTK: Natural Language Toolkit
  - https://www.nltk.org/

- Package with functions for natural language processing

Important concepts

- ***Tokenize***: split sentences up into individual works as elements in string vector
- ***Stopwords***: A list of words to exclude from document
  - Standards are words like: "a", "the", "then", "and", etc.
- ***Dictionaries*** and ***word lists***: pre-defined lists of words that are classified as a type of word (e.g., "English", or "positive sentiment")
  - See: https://sraf.nd.edu/loughranmcdonald-master-dictionary/
  - The ***lm*** option in ***pysentiment2*** contains this library
- ***Stemming***: get the stem of a word, e.g. "invest" could be stem of "investing", "investment", "investor", etc.
  - Use ***PorterStemmer*** in NLTK package

a. News Data: Download and Cleaning

```
data = pd.read_csv('DJIA_Headline_News.csv')

def create_df(dataset):
   stop_words = set(stopwords.words('english'))
   lm         = ps.LM() # Loughran and MacDonald
sentiment word list
   dataset    = dataset.drop(columns=['Date', 'Label'])
   dataset.replace("[^a-zA-Z]", " ", regex=True,
inplace=True)

   for col in dataset.columns:
      dataset[col] = dataset[col].str.lower()
      dataset[col] = dataset[col].str.replace('b ','')
   headlines  = []
   head_clean = []
   sentscore  = []
   porter = PorterStemmer()
```

```
for row in range(len(dataset.index)):
        document = ' '.join(str(x) for x in dataset.iloc[row, 0:25])
        headlines.append(document)
        tokens  =  word_tokenize(document)
        stemmed = [porter.stem(word) for word in tokens]
        words   = [w for w in stemmed if not w in stop_words]
        head_clean.append(' '.join(word for word in words))
        tokens  = lm.tokenize(' '.join(word for word in words))
        sentscore.append(lm.get_score(tokens)['Negative'])
    df          = pd.DataFrame(headlines, columns=['All'])
    df['processed'] = head_clean
    df['score']     = sentscore
   df['label'] = data.Label
    df['date']  = data.Date

    entire_processed_text = ' '.join(doc for doc in head_clean)
    return df[['date','label','All','processed','score']],
entire_processed_text

df_full, entire_text = create_df(data)
```

UCLA Anderson SCHOOL of MANAGEMENT

a. News Data: Create Corpus and Clean

- Next, we will create a "Corpus," which is the set of text we consider
- Then we will create a Document Term Matrix (DTM), which is a common concept and input to several routines
- First, run the file "Snippets topic 6 _ initialization.py" to create folder containing each headline as individual .txt file.
- Next:

```
# here, set the directory you defined for the Corpus in the initialization script
newcorpus = PlaintextCorpusReader('YOURDIRECTORY', '.*')

# create a DTM from corpus
def dtm_from_corpus(xCorpus):
    s = 0
    fd_list = []
    for x in range(s, len(xCorpus.fileids())):
        fd_list.append(nltk.FreqDist(xCorpus.words(xCorpus.fileids()[x])))
    dtm = pd.DataFrame(fd_list, index = xCorpus.fileids()[s:])
    dtm.fillna(0,inplace = True)
    return dtm

dtm = dtm_from_corpus(newcorpus)
```

UCLA Anderson SCHOOL of MANAGEMENT

a. News Data: Create DTM

- DTM is a matrix of all the unique words in the first row, where following rows are each file in the corpus giving the frequency of that word in that file

```
dtm
Out[5]:
                     georgia  two  russian  ...  medit  writh  curbia
file_2008-08-08.txt      9.0  2.0      5.0  ...    0.0    0.0     0.0
file_2008-08-11.txt      4.0  0.0      2.0  ...    0.0    0.0     0.0
file_2008-08-12.txt     10.0  1.0      2.0  ...    0.0    0.0     0.0
file_2008-08-13.txt      8.0  1.0      4.0  ...    0.0    0.0     0.0
file_2008-08-14.txt      5.0  0.0      3.0  ...    0.0    0.0     0.0
                         ...  ...      ...  ...    ...    ...     ...
file_2016-06-27.txt      0.0  1.0      1.0  ...    0.0    0.0     0.0
file_2016-06-28.txt      0.0  1.0      0.0  ...    0.0    0.0     0.0
file_2016-06-29.txt      0.0  0.0      0.0  ...    0.0    0.0     0.0
file_2016-06-30.txt      0.0  0.0      0.0  ...    0.0    0.0     0.0
file_2016-07-01.txt      0.0  0.0      0.0  ...    1.0    1.0     1.0

[1989 rows x 22388 columns]
```

UCLA Anderson
SCHOOL of MANAGEMENT

a. Get a sense of data using word frequency

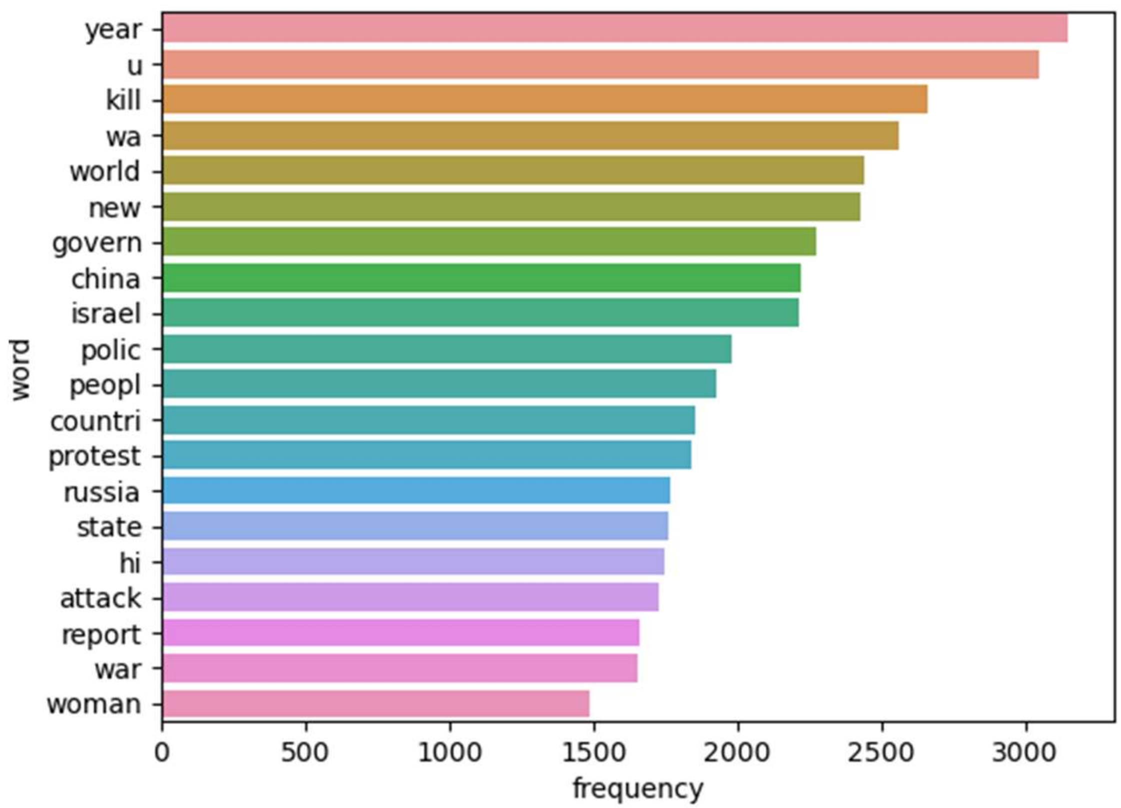
Important concepts:

- ***Unigram***: one word, e.g., "stock"
- ***Bigram***: two consecutive words, "tech stock"
- ***Trigram***: three consecutive words, "tech stock bubble"
- ***Lemmatize***: similar to stemming, but gets more at overall meaning. Examples
  beans -> bean
  better -> good

```
s_words = stopwords.words('english')
additional_stopwords = ['u','ha','say']
s_words.extend(additional_stopwords)
def word_frequency(sentence,stopwords):
#joins all the sentence, creates tokens, creates lower case,
#removes numbers and lemmatizes the words
new_tokens = word_tokenize(sentence)
    new_tokens = [t.lower() for t in new_tokens]
    new_tokens =[t for t in new_tokens if t not in s_words]
    new_tokens = [t for t in new_tokens if t.isalpha()]
    lemmatizer = WordNetLemmatizer()
    new_tokens =[lemmatizer.lemmatize(t) for t in new_tokens]
    #counts the words, pairs and trigrams
    counted   = Counter(new_tokens)
    counted_2 = Counter(ngrams(new_tokens,2))
    counted_3 = Counter(ngrams(new_tokens,3))
    #creates 3 data frames and returns them
    word_freq =
pd.DataFrame(counted.items(),columns=['word','frequency']).sort_value
s(by='frequency',ascending=False)
    word_pairs
=pd.DataFrame(counted_2.items(),columns=['pairs','frequency']).sort_v
alues(by='frequency',ascending=False)
    trigrams
=pd.DataFrame(counted_3.items(),columns=['trigrams','frequency']).sor
t_values(by='frequency',ascending=False)
    return word_freq,word_pairs,trigrams

data2, data3, data4 = word_frequency(entire_text,s_words)
```

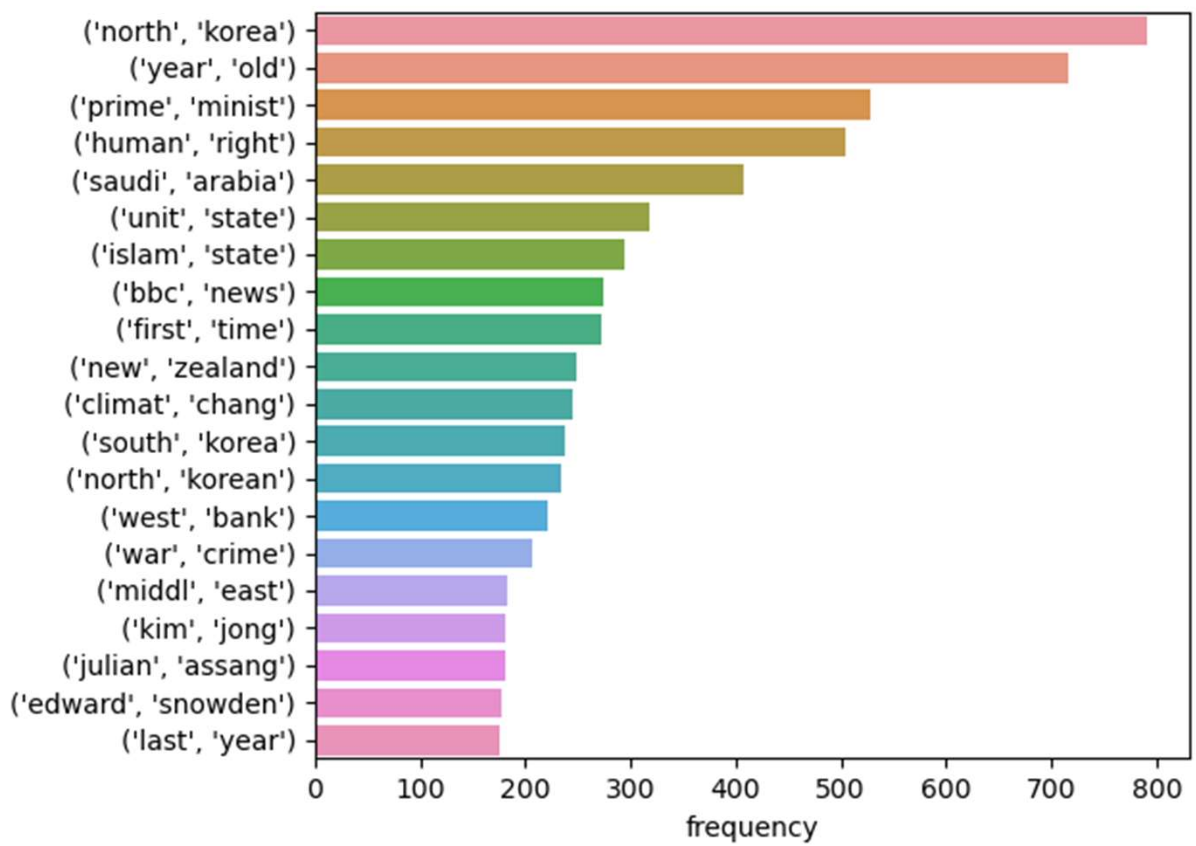
a. Get a sense of data using word frequency

```python
# top unigrams
ax1 = plt.figure()
sns.barplot(x='frequency',y='word',data=data2.head(50))
```



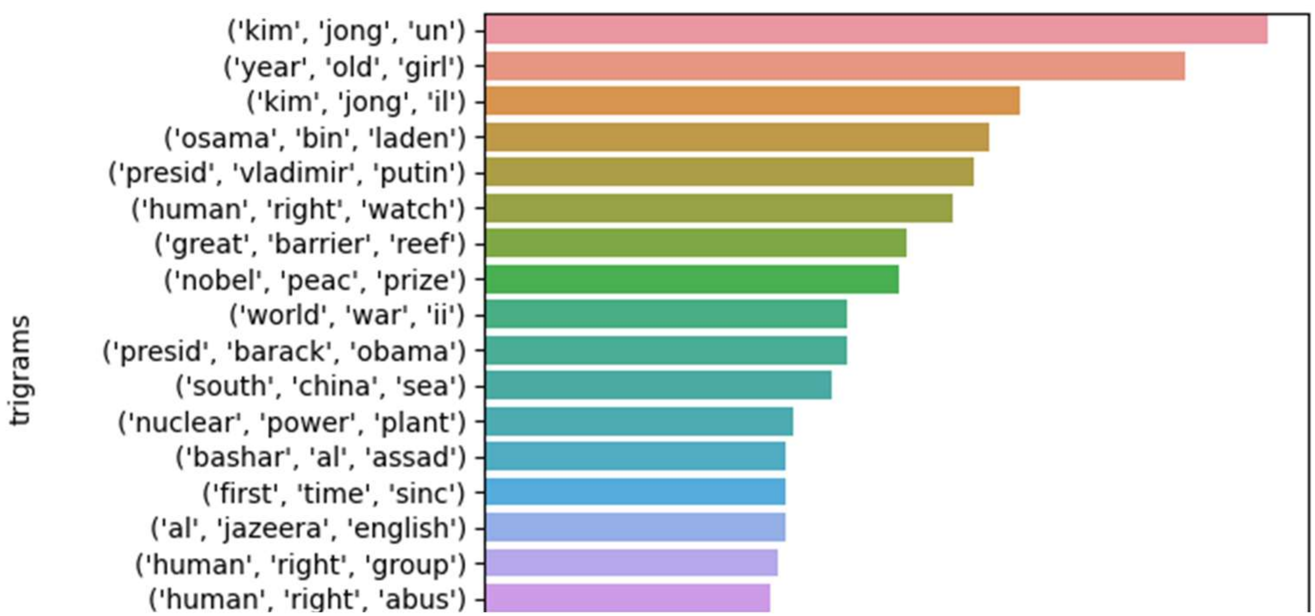
a. Get a sense of data using word frequency

```
# top bigrams
ax2 = plt.figure()
sns.barplot(x='frequency',y='pairs',data=data3.head(20))
```



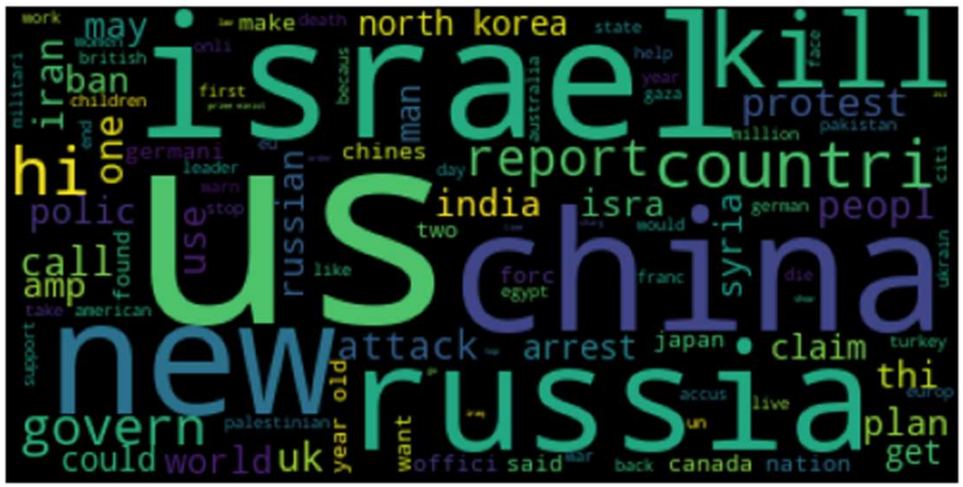
a. Get a sense of data using word frequency

```
# top trigrams
ax3 = plt.figure()
sns.barplot(x='frequency',y='trigrams',data=data4.head(20))
```



a. Create a WordCloud to get sense of data

```
# Plot 100 most frequent words
wc = WordCloud(max_words=100,stopwords={'say','ha','wa','u'}).generate_from_text(entire_text)
plt.figure()
plt.imshow(wc)
plt.axis('off')
plt.show()
```

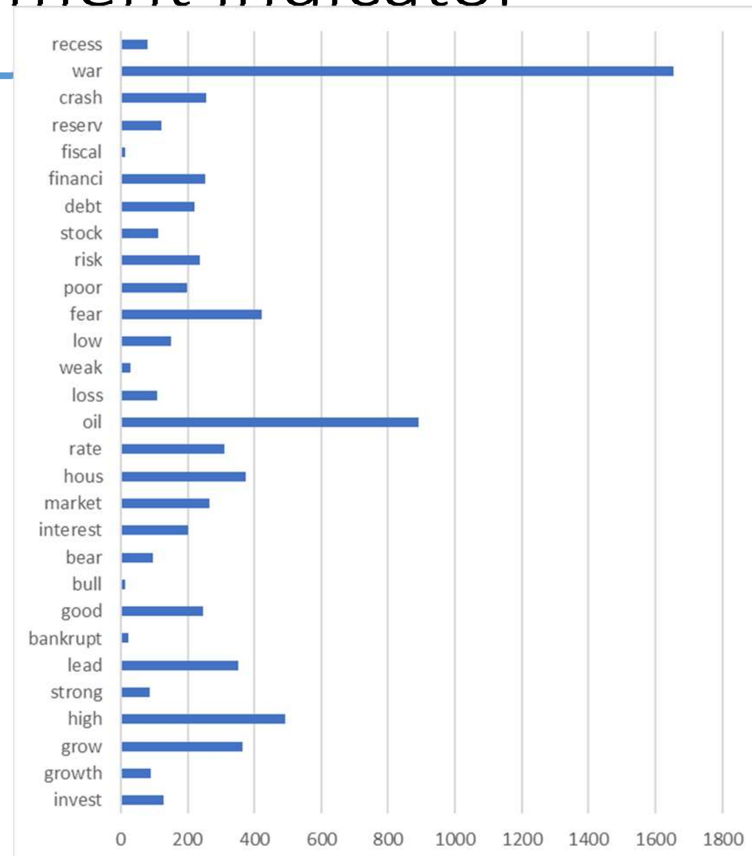


Pretty clear that typical DJIA headline is about political events, such as wars, foreign affairs, legal issues, etc. Could potentially be related to stock returns and sentiment, but far from obvious

UCLA Anderson
SCHOOL of MANAGEMENT

b. Create a Sentiment Indicator

- Baker and Wurgler created an index based on variables they ex ante thought to be related to investor sentiment. We can take a similar approach by pre-defining a set of 'sentiment'-related words.
  - Note: words are expressed in their stemmed form as we have stemmed the document
  - Also note: I just made up a set of words that made sense to me, you can do better!




```
sent_words = ["invest","growth","grow","high","strong","lead","bankrupt",
              "good","bull","bear","interest","market","hous","rate","oil",
              "loss","weak","low","fear","poor","risk","stock","debt",
              "financi","fiscal","reserv","crash","war","recess"]
dtm_sentiment = dtm[sent_words]
dtm_sentiment_sum = dtm_sentiment.sum()
dtm_sentiment_sum
```

UCLA Anderson SCHOOL of MANAGEMENT

b. A regression-based text model

- The outcome-variable (label) is binary, so a logistic regression is natural.

```
# create y and x data for regressions
y_data       = df_full['label']
x_data       = dtm_sentiment
x_data.index = y_data.index

# run logit regression
glm_binom = sm.GLM(y_data,sm.add_constant(x_data),family=sm.families.Binomial())
fitted    = glm_binom.fit()
fitted.summary()
```

- Note that DTM gives word frequency of each word, each period. Thus x_data are integers with typical values 0, 1, and 2, though higher number of occurrences may happen.
- Output is on next slide. Note that only coefficients on *low* and *stock* have t-statistics higher than 2. In addition, *oil* is marginally significant

UCLA Anderson SCHOOL of MANAGEMENT

b. Logistic regression results

Generalized Linear Model Regression Results

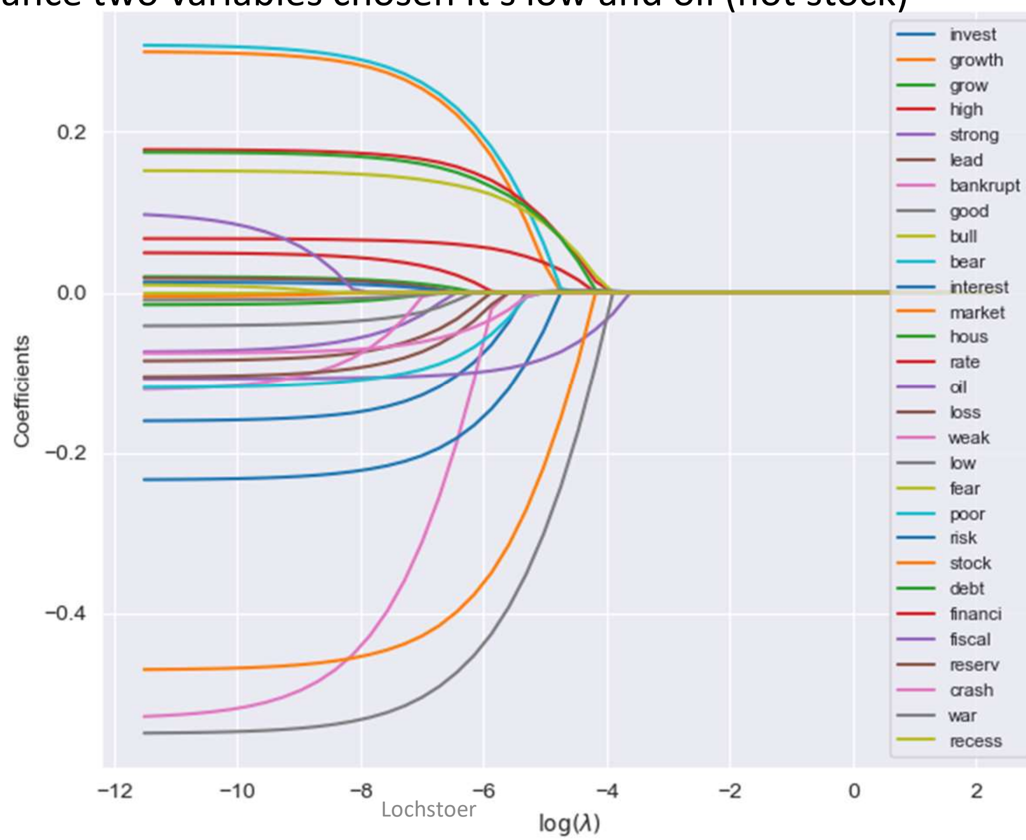
Dep. Variable:	label	No. Observations:	1989

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.2138	0.090	2.376	0.018	0.037	0.390
invest	-0.1606	0.173	-0.929	0.353	-0.500	0.178
growth	0.3000	0.205	1.461	0.144	-0.102	0.702
grow	0.0197	0.109	0.181	0.856	-0.194	0.233
high	0.0671	0.091	0.739	0.460	-0.111	0.245
strong	-0.0744	0.215	-0.345	0.730	-0.497	0.348
lead	0.0179	0.105	0.170	0.865	-0.189	0.225
bankrupt	-0.5314	0.438	-1.214	0.225	-1.389	0.326
good	-0.0417	0.124	-0.336	0.737	-0.285	0.202
bull	0.0098	0.566	0.017	0.986	-1.099	1.119
bear	0.3082	0.195	1.579	0.114	-0.074	0.691
interest	-0.2340	0.145	-1.615	0.106	-0.518	0.050
market	-0.0053	0.126	-0.042	0.967	-0.253	0.242
hous	-0.0154	0.101	-0.152	0.879	-0.214	0.183
rate	0.1778	0.107	1.658	0.097	-0.032	0.388
oil	**-0.1076**	**0.056**	**-1.925**	**0.054**	**-0.217**	**0.002**
loss	-0.0855	0.188	-0.454	0.650	-0.455	0.284
weak	-0.1220	0.365	-0.334	0.738	-0.837	0.593
low	**-0.5499**	**0.166**	**-3.303**	**0.001**	**-0.876**	**-0.224**
fear	0.1517	0.100	1.522	0.128	-0.044	0.347
poor	-0.1185	0.139	-0.855	0.392	-0.390	0.153
risk	0.0126	0.127	0.099	0.921	-0.236	0.261
stock	**-0.4707**	**0.185**	**-2.548**	**0.011**	**-0.833**	**-0.109**
debt	0.1747	0.124	1.412	0.158	-0.068	0.417
financi	0.0496	0.122	0.407	0.684	-0.189	0.289
fiscal	0.1006	0.553	0.182	0.856	-0.983	1.184
reserv	-0.1055	0.176	-0.601	0.548	-0.450	0.239
crash	-0.0758	0.109	-0.698	0.485	-0.289	0.137
war	-0.0096	0.043	-0.222	0.824	-0.094	0.075
recess	0.0099	0.216	0.046	0.964	-0.414	0.434

UCLA Anderson
SCHOOL of MANAGEMENT

b. A regularized regression-based text model

- Next, let’s consider adding regularization (or, equivalently, a prior) to the logistic regression.
  - In particular, let’s use elastic net with alpha = 0.5
- First, let’s look at regularized coefficients versus lambda (constraint)
  - Note: if for instance two variables chosen it’s low and oil (not stock)



b. A regularized regression-based text model

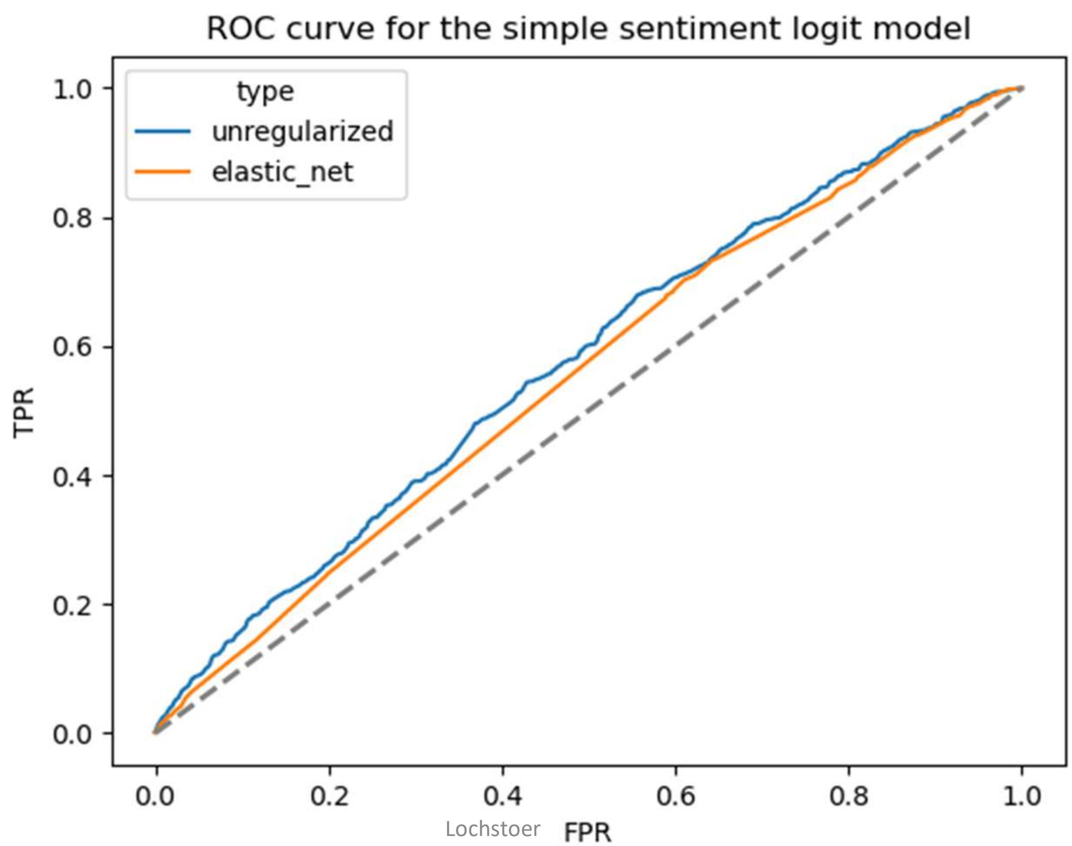
- Let’s do a 10-fold cross-validation exercise to find lambda

```
# Elastic net with cross validation
mod1 = LogisticRegressionCV(penalty='elasticnet',solver = 'saga',
            l1_ratios=[0.5], cv = 10, max_iter = 1000,fit_intercept =
False,refit=True)
mod1 = mod1.fit(x_data, y_data)

# Best penalizing term
best_alpha_cv = mod1.C_

# Estimate the model again using this "best" alpha
mod1 = LogisticRegression(penalty='elasticnet',solver = 'saga',
                          l1_ratio=0.5, max_iter = 1000,fit_intercept = True)
mod1.C=best_alpha_cv[0]
mod1.fit(x_data, y_data)
# look at estimated coefficients
mod1.coef_
```

```
array([[ 0.        ,  0.        ,  0.        ,  0.        ,  0.        ,
         0.        ,  0.        ,  0.        ,  0.        ,  0.        ,
         0.        ,  0.        ,  0.        ,  0.03384449, -0.07897115,
         0.        ,  0.        , -0.20226243,  0.03215747,  0.        ,
         0.        , -0.11615796,  0.0332111 ,  0.        ,  0.        ,
         0.        ,  0.        ,  0.        ,  0.        ]])
```

b. The ROC Curve for Regularized Logistic Reg.

- Get the ROC curves for both unregularized and regularized models
- Note the elastic net does worse, but this is in-sample
    - We expected this!



UCLA Anderson SCHOOL of MANAGEMENT

b. Proper out-of-sample testing

- Split data into training data-set and proper out-of-sample (not cross-validation) data set. Let training data be data up until 2014-12-31)

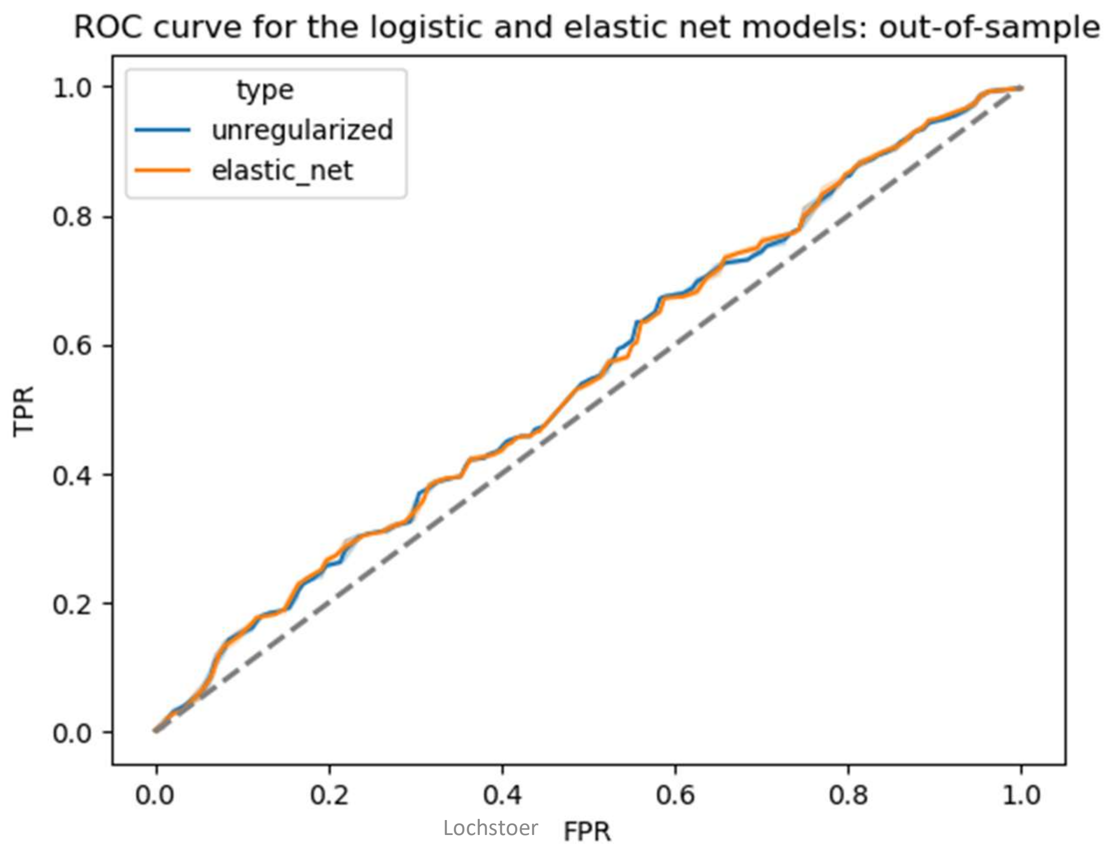
```
# Out-of-sample testing
split_index = np.where(df_full.date=='2014-12-31')[0]
x_data_trans=pd.DataFrame(x_data)
x_train = x_data.iloc[:split_index[0],:]
x_test  = x_data.iloc[split_index[0]:,:]

y_train = y_data.iloc[:split_index[0]]
y_test  = y_data.iloc[split_index[0]:]
```

- Run CV exercise on training data, then test ROC on testing data
    - See code snippets

UCLA Anderson SCHOOL of MANAGEMENT

b. Proper out-of-sample testing

- ROC Curves
  - Elastic net performs very similarly to unregularized
  - Note, we are not guaranteed that the regularized will do better out of sample, it’s just more likely to given CV exercise



UCLA Anderson SCHOOL of MANAGEMENT

b. Lower frequency co-movement

- This model is estimated daily, but may be a good description of more low frequent stock returns

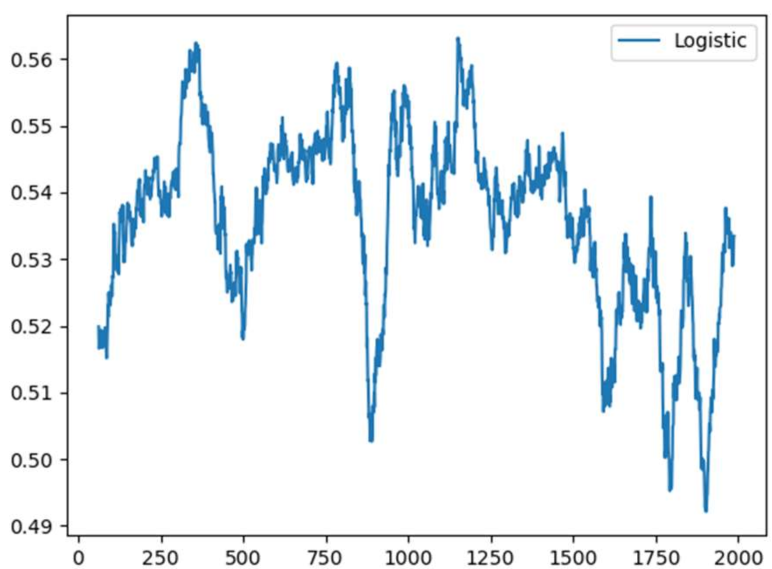
- A natural way to go to, say, the quarterly frequency is to sum up the predictions over the quarter and relate to stock returns over the quarter
    - Note: we are here not trying to actual forecast quarterly stock returns, but instead see if there is comovement between our sentiment indicator and stock returns at the quarterly frequency
    - Of course, you can yourself see if there is a statistically significant forecasting relation as well (probably not, though; textual sentiment data is usually short-lived).

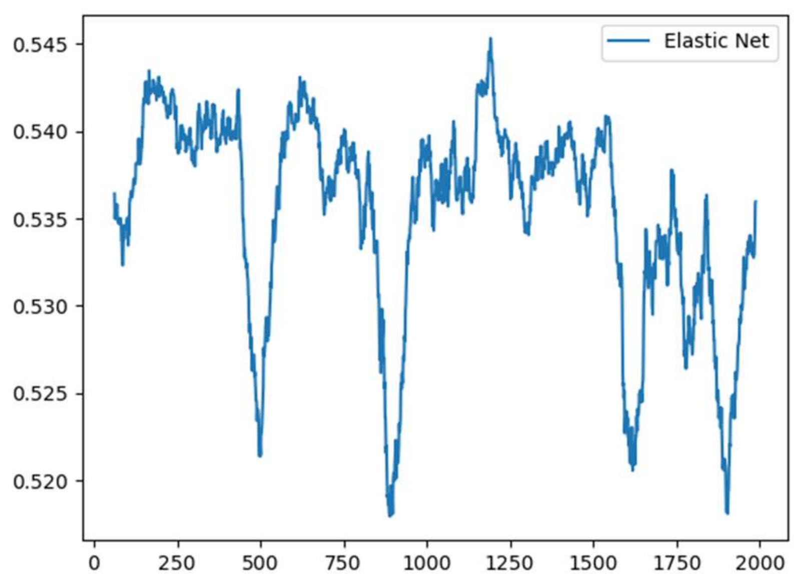
- Create 63-day (3 months) moving average of returns are model predictions, overlapping at the daily frequency

```
preds_ma_elnet = pd.DataFrame(mod1.predict_proba(x_data)[:,1],columns=['Elastic
Net']).rolling(63).mean().dropna()
preds_ma_logit =
pd.DataFrame(logit_model.predict_proba(x_data)[:,1],columns=['Logistic']).rolling(63).mean().dropna()
y_ma           = pd.DataFrame(np.array(y_data),columns=['y_label']).rolling(63).mean().dropna()
```

b. Lower frequency co-movement

- Plot low frequency versions of sentiment measures
  - Looks more like Baker and Wurgler series.
  - Lower frequency plots like this are visually more intuitive and easier to relate to particular historical events
  - Notice much higher volatility of unregularized regression predictions





UCLA Anderson SCHOOL of MANAGEMENT

b. Lower frequency co-movement

```
# regress low frequency ma components to assess contemporaneous relationship
# note how important it is to adjust for autocorrelation in residuals!
nw_lag = 90
reg21 = sm.OLS(y_ma,preds_ma_logit).fit().summary()
reg22  =
sm.OLS(y_ma,preds_ma_logit).fit(cov_type='HAC',cov_kwds={'maxlags':nw_lag}).summary()
```

Covariance Type: nonrobust

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.2141	0.052	-4.127	0.000	-0.316	-0.112
Logistic	1.4007	0.097	14.469	0.000	1.211	1.591

Covariance Type: HAC

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.2141	0.316	-0.677	0.498	-0.834	0.406
Logistic	1.4007	0.592	2.365	0.018	0.240	2.562

UCLA Anderson SCHOOL of MANAGEMENT

b. Lower frequency co-movement

```
# Comparison for elastic net
reg41 = sm.OLS(y_ma,preds_ma_elnet).fit().summary()
reg41
reg42 = sm.OLS(y_ma,preds_ma_elnet).fit(cov_type='HAC',cov_kwds={'maxlags':nw_lag}).summary()
reg42
```

Covariance Type: nonrobust

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.8640	0.125	-6.892	0.000	-1.110	-0.618
Elastic Net	2.6152	0.234	11.170	0.000	2.156	3.074

Covariance Type: HAC

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.8640	0.743	-1.164	0.245	-2.319	0.591
Elastic Net	2.6152	1.392	1.879	0.060	-0.113	5.344

- Note how taking into account overlap (autocorrelation) is critical for correct standard errors.
- 6-10% of the variation in quarterly stock returns (R2) are reflected in DJIA headlines over this sample
- Casuality is likely mainly from stock returns to news, rather than news to stock returns

UCLA Anderson
SCHOOL of MANAGEMENT

c. Intro to text analysis: After thoughts…

Difficult to construct robust predictor based on text

Using the cross-section likely better as increases amount of data, power to detect underlying mechanism

In general, a good idea of how to filter the data is needed