Corporate political influence:

How firms use earnings calls to respond to politics

Anthony Cozart

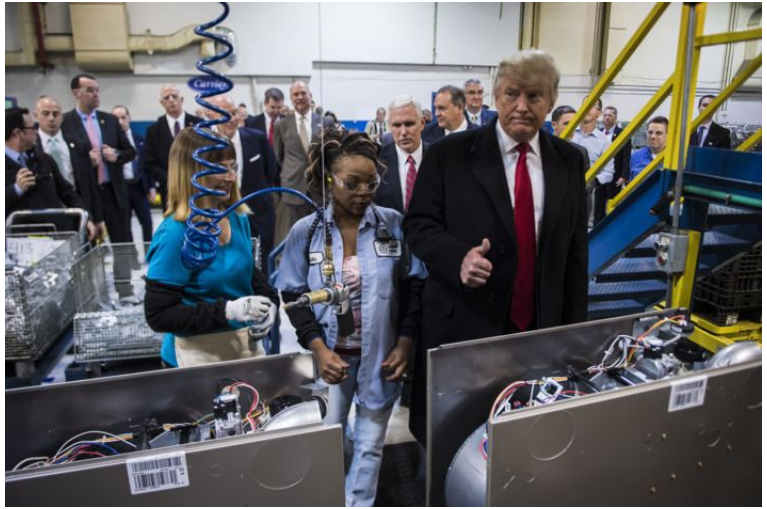
SF-based jobs website interview (originally for SI 618, W18)

April 16, 2018

Motivation

From 2008-2016, political issues like healthcare and immigration became increasingly important to businesses.

In the 2016 election, Trump sought to win voters by criticizing trade policies, and by pressuring business executives to stop sending factory jobs abroad.



Trump touring a Carrier factory in Indiana.

Research Questions

How do business executives use investor calls to respond to political threats, and to new policies (e.g., Obamacare)?

How do these responses affect stock valuations? And why do the valuations change? (i.e., do they reflect new information, including the signal of the response itself?)

Data

Every quarter, executives of publicly listed companies present results to their investors.

These calls are transcribed, and posted to www.SeekingAlpha.com.

We can look to these transcripts to see how companies are talking about presidential candidates, their proposals, and existing policies.

Roadblock #1

Seeking Alpha has roughly 20k transcripts for S&P500 companies (4 per year, over ten years).

I wrote a script to see if it was possible to retrieve the transcripts, but was rate-limited immediately.

(Why? Web-scraping violates their terms and conditions.)

A different approach

I couldn't get the data I wanted, but found transcripts on GitHub that allow me to do a "proof of concept."

Data: 130 transcripts from 28 industrial companies in 2016.

This meant I couldn't examine or exploit variation across time, candidate, or industry.

Link to data: https://github.com/Carlossn/Python/tree/master/NLP_Prof_Warning_Prediction/Data/transcripts/trans

Workflow

Cleaning ⇨ Exploratory analysis ⇨ LDA classification ⇨ Prediction

Cleaning

Step 1: Load and understand JSON files

```
In [312]: df = pd.read_json('transcripts.json')
          df.head()
```

Out[312]:

	Date	Month	Name	Period	Symbol	Target	Text	Year
0	2016-01-26	1	Parker-Hannifin Corp.	Q22016	Parker-Hannifin Corp. (NYSE:	0	[Parker-Hannifin Corp. (NYSE:, Q2 2016 Earning...	2016
1	2016-04-26	4	Parker Hannifin Corporation	Q32016	Parker Hannifin Corporation (NYSE:	0	[Parker Hannifin Corporation (NYSE:, Q3 2016 E...	2016
2	2016-08-20	8	Parker-Hannifin Corp.	Q42016	Parker-Hannifin Corp. (NYSE:	0	[Parker-Hannifin Corp. (NYSE:, Q4 2016 Earning...	2016

What's in an earnings call transcript?

```
In [359]: df.loc[1, 'Text']
```

```
Out[359]: '[\'Parker Hannifin Corporation (NYSE:\', \'Q3 2016 Earnings Conference Call\', \'April 26, 2
016, 11:00 AM ET\', \'Executives\', \'Thomas L. Williams - Chairman and Chief Executive Offic
er\', \'Jon P. Marten - Executive Vice President and Chief Financial Officer\', \'Lee C. Bank
s - President and Chief Operating Officer\', \'Analysts\', \'Jamie Cook - Credit Suisse\', \'
Joseph Ritchie - Goldman, Sachs & Co.\', \'Jeffrey Hammond - KeyBanc Capital Markets, Inc.\',
\'Ann Duignan - JPMorgan Chase\', \'David Raso - Evercore ISI Group.\', \'Eli Lustgarten - Lo
ngbow Research LLC.\', \'Joshua Pokrzywinski - Buckingham Research Group Inc.\', \'Andrew Cas
ey - Wells Fargo Securities, LLC.\', \'Nathan Jones - Stifel, Nicolaus & Co.\', \'Joseph Gior
dano - Cowen & Co.\', \'Operator\', \'Good day, ladies and gentlemen, and welcome to the Park
er-Hannifin Corp. Quarter Three 2016 Earnings Conference Call. At this time, all participants
are in a listen-only mode. Later, we will host a question-and-answer session, and instruction
s will follow at that time [Operator instructions] As a reminder, this conference is being re
corded.\', \'Now I will hand the floor over to Jon Marten, Chief Financial Officer. Sir you h
ave the floor.\', \'Jon P. Marten\', "Good morning, and welcome to Parker-Hannifin\'s third q
uarter FY 2016 earnings release teleconference. Joining me today is Chairman and Chief Execut
ive Officer, Tom Williams; and President and Chief Operating Officer, Lee Banks.", "Today\'s
presentation slides together with the audio webcast replay will be accessible on the company\
's Investor Information website at phstock.com for one-year following today\'s call.", "On sl
ide number two, you\'ll find the company\'s Safe Harbor disclosure statement addressing forwa
```

What's in a clean earnings call transcript?

```
In [358]: df.loc[1, 'Clean Text']
```

```
Out[358]: '\', \'Good day, ladies and gentlemen, and welcome to the Parker-Hannifin Corp. Quarter Three
2016 Earnings Conference Call. At this time, all participants are in a listen-only mode. Late
r, we will host a question-and-answer session, and instructions will follow at that time [Ope
rator instructions] As a reminder, this conference is being recorded.\', \'Now I will hand th
e floor over to Jon Marten, Chief Financial Officer. Sir you have the floor.\', \'Jon P. Mart
en\', "Good morning, and welcome to Parker-Hannifin\'s third quarter FY 2016 earnings release
teleconference. Joining me today is Chairman and Chief Executive Officer, Tom Williams; and P
resident and Chief Operating Officer, Lee Banks.", "Today\'s presentation slides together wit
h the audio webcast replay will be accessible on the company\'s Investor Information website
at phstock.com for one-year following today\'s call.", "On slide number two, you\'ll find the
company\'s Safe Harbor disclosure statement addressing forward-looking statements as well as
non-GAAP financial measures. Reconciliations for any reference to non-GAAP financial measures
are included in this morning\'s press release and are posted on Parker\'s website at phstock.
com.", "Today\'s call agenda appears on slide number three, to begin, our Chairman and Chief
Executive Officer, Tom Williams, will provide highlights for the third quarter of fiscal year
2016. Following Tom\'s comments, I will provide a review of the company\'s third quarter FY 2
016 performance together with the revised guidance for FY 2016. Tom will provide a few summar
y comments and then we\'ll open the call for a Q&A.", "At this time, I\'ll turn it over to To
m and ask that you refer to slide number four.", \'Thomas L. Williams\', "Thanks Jon, good mo
```

Cleaning

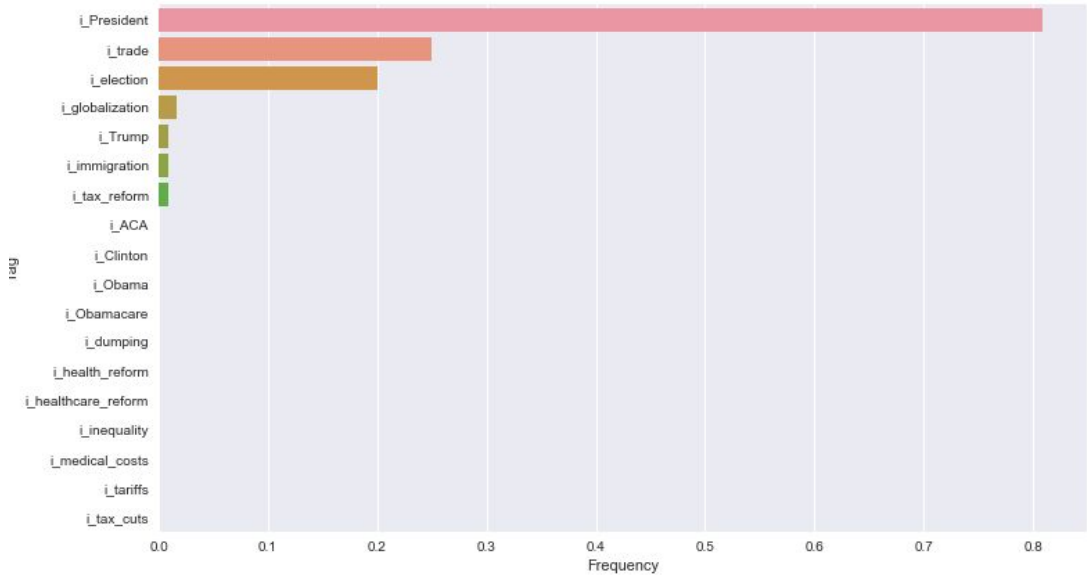
Step 2: Extract relevant info from “Text”

```
# Split text using 'operator' -- only on the first instance. This keeps analyst calls.
df.dtypes
df['Text'] = df['Text'].astype(str)
df['Clean Text'] = df['Text'].str.split("Operator", 1).str[1]
# Four rows don't have an operator, and are relatively clean (they're a bit different
df['Clean Text'] = np.where(df['Clean Text'].isnull(), df['Text'], df['Clean Text'])
```

Steps 3 & 4: Create new variables, and join post-election change in stock price data from Yahoo Finance

Exploratory Analysis

Figure 1. Frequency of political words



The top bar, an indicator for “President”, is an outlier. More on this later.

Roadblock #2

Businesses aren’t talking about most political issues.

The words “trade” and “election” come up in roughly 20% of transcripts.

Executives mentioned that customers were delaying purchases until after the election, and that political uncertainty was affecting their own investment decisions.

Change in direction

Simply counting the “buzzwords” in Figure 1, given the lack of variation, doesn’t say much about how firms are talking about political issues.

But, I have 120 transcripts that are a rich source of data.

Can I apply some of the algorithms and tools I’ve been learning to understand these transcripts a bit more?

LDA classification

I use a Latent Dirichlet Allocation (“LDA”) model in PySpark to:

1. understand the topics executives are discussing in their calls,
2. and to classify each transcript by a “dominant” topic.

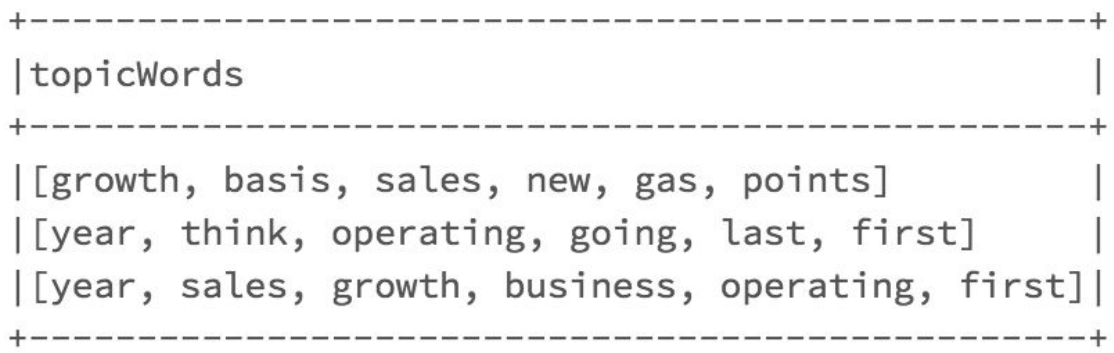
Intuitively, what is LDA?

Link to PySpark documentation: https://spark.apache.org/docs/2.2.0/ml-clustering.html#latent-dirichlet-allocation-lda

LDA: clustering words in text data by topic, and then calculating the dominant topic

cluding the contribution year-to-date, operating cash to sales was 10.5%.\
', \'During the second quarter, we repurchased $50 million in shares, brin
ging our year-to-date total to $450 million. We have now repurchased $1.8
billion in shares since October 2014. However, the most impressive accompl
ishment of the third quarter was our margin performance. I'm very pleased
to be delivered total segment operating margins of 13.8% or 14.7% on adjus
ted basis. This represents a 30 basis points improvement year-over-year in
adjusted margins which is a significant accomplishment given the difficult
economic conditions we are facing.\', \'During the third quarter, we deliv
ered decrementals margin return on sales of 17% or 11.8% adjusted for busi
ness realignment expenses. This demonstrates excellent performance by our
team. This quarter remarks the 5\', \'Year-to-date we have also held SG&A
flat at 12.1% of the sales despite of $1.2 billion drop in sales, another
significant accomplishment by our team. Our performance during such a sust
ained period of lower sales and order rates is unprecedented. It demonstra
tes Parker's ability to create a more adaptable cost structure and deliver
less difficult financial performance.\', \'So now a few brief comments on
key end-markets, so reflecting on the order rates over the past year, we a

What are the topics?



```
+-----------------------------------------------+
|topicWords                                     |
+-----------------------------------------------+
|[growth, basis, sales, new, gas, points]       |
|[year, think, operating, going, last, first]   |
|[year, sales, growth, business, operating, first]|
+-----------------------------------------------+
```

Figure 2. LDA Topic Words
Takeaway: firms talk about gas prices, future sales, and the operating environment.

Dominant Topics, by industrials sub-sectors

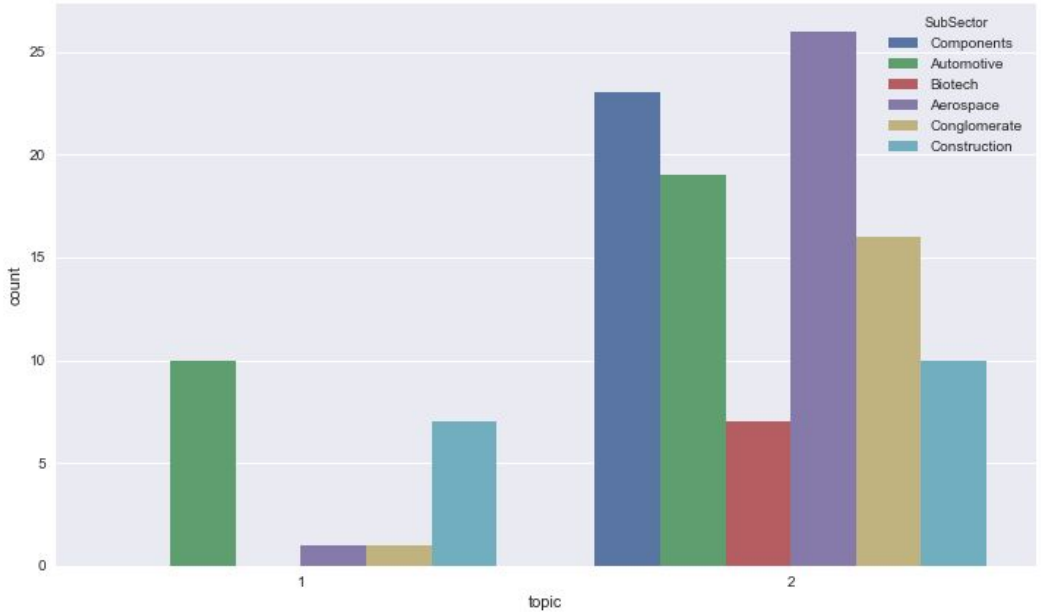


Figure 3. LDA Classification

Takeaway: Some variation in the dominant topic by “sub-sector”

Direction of future research

- Improve the data
- Fine-tune the LDA algorithm given new data (re-calculate optimal # of topics)
- Process financial data before and after calls with political information by executives
- See whether our LDA classifier predicts the size of stock price movements

Questions?

Web-scrape code (not used)

```
ticker = []
headline = []
text = []

for symbol in symbol_list:
    site = 'http://seekingalpha.com/symbol/'+symbol+'/earnings/transcripts'
    print(symbol)
    hdr = {'User-Agent':'Mozilla/5.0'}
    # update args in hdr to avoid 403 forbidden errors. See: https://stackoverflow.com/questions/13303449/url
    req = urllib.request.Request(site, headers=hdr)
    try:
        page = urllib.request.urlopen(req)
        soup = BeautifulSoup(page, 'lxml')
        # now loop through all of the linked transcripts on the page to find transcripts
        for link in soup.find_all('a'):
            x = link.get('href')
            # print(x)
            if isinstance(x, str):
                # to start, just look for 'transcript' and 'article' in the links.
                # the links we want look like: /article/4132150-3m-mmm-2018-outlook-meeting-conference-transc
                # with just 'transcript', we get things like: http://seekingalpha.com//earnings/earnings-call
                wordlist = ['transcript', 'article']
                if all(x.find(s) >= 0 for s in wordlist):
                    parse_site = 'http://seekingalpha.com/'+x+'?part=single'
                    print(parse_site)
                    parse_req = urllib.request.Request(parse_site, headers=hdr)
                    try:
                        parse_page = urllib.request.urlopen(parse_req)
                        parse_soup = BeautifulSoup(parse_page,'lxml')
                        # print(parse_soup.prettify())
                        ticker.append(symbol)
                        headline.append(parse_soup.title.string)
                        text.append(parse_soup.text) # look at data structure to get rid of the header
                    except urllib.error.URLError as e:
                        print ('Error code:',e.code)
                    time.sleep(5)
    except urllib.error.URLError as e:
        print('Error code:', e.code)
```

LDA Code (PySpark, in DataBricks)

```
from pyspark.ml.feature import HashingTF, IDF, Tokenizer, CountVectorizer, RegexTokenizer
from pyspark.ml.feature import StopWordsRemover
from pyspark.ml.clustering import LDA
from pyspark.ml.pipeline import Pipeline
import numpy as np
import pandas as pd
import nltk
from nltk.corpus import stopwords as nltkstopwords
nltk.download("book")

transcripts = spark.table("clean_transcripts_csv3")

#  This is a helper function that looks up the words associated with indices.
from pyspark.sql.types import ArrayType, StringType

def indices_to_terms(vocabulary):
    def indices_to_terms(xs):
        return [vocabulary[int(x)] for x in xs]
    return udf(indices_to_terms, ArrayType(StringType()))
```

```
# Create custom StopWords dictionary, combining NLTK english words with a few additional.
# Why? We have a lot of titles (aka Chief Executive, President,) that are becoming topics
# If we remove numerical characters, then we should also remove millions/thousands (also topics)
custom_stopwords = nltkstopwords.words('english')
to_add = ['chief', 'executive', 'vice', 'president', 'officer', 'director', 'chairman', 'chair', 'million', 'billion',
'thousand','quarter']
# ,'year',
for w in to_add:
  custom_stopwords.append(w)

tokenizer = RegexTokenizer(inputCol="Remarks", outputCol="words", pattern="[a-zA-Z]*", gaps=False)
stopWordsRemover = StopWordsRemover(inputCol="words", outputCol="filtered",stopWords=custom_stopwords)
stopWordsRemover.loadDefaultStopWords("english")
vectorizer = CountVectorizer(inputCol="filtered", outputCol="features", minDF=3) # if = 2 we start to get names as topics
# Check:
# stopWordsRemover.getStopWords()
```

```
k = []
log_perplexity = []
log_likelihood = []

for kay in range(3,10,1):
  print ("k = ",kay)
  lda = LDA(k=kay, maxIter=10, seed=123)
  pipeline = Pipeline(stages=[tokenizer, stopWordsRemover, vectorizer, lda])
  pipelineModel = pipeline.fit(transcripts)

  countVectorModel = pipelineModel.stages[-2]
  cmv = countVectorModel.vocabulary

  ldaModel = pipelineModel.stages[-1]
  transcripts_lda = pipelineModel.transform(transcripts)

  k.append(kay)
  lp = ldaModel.logPerplexity(transcripts_lda)
  lp = round(lp, 4)
  log_perplexity.append(lp)
  ll = ldaModel.logLikelihood(transcripts_lda)
  ll = round(ll, 4)
  log_likelihood.append(ll)

lda_scores = pd.DataFrame({'perplexity': log_perplexity, 'likelihood': log_likelihood}, index=k)
lda_scores.sort_values(by=['likelihood','perplexity'], ascending=[False,True])

# Re-run with optimal k
```

```
# Re-run with optimal k
optimal_k = 3

lda = LDA(k=optimal_k, maxIter=10, seed=123)
pipeline = Pipeline(stages=[tokenizer, stopWordsRemover, vectorizer, lda])
pipelineModel = pipeline.fit(transcripts)

countVectorModel = pipelineModel.stages[-2]
cmv = countVectorModel.vocabulary

ldaModel = pipelineModel.stages[-1]
transcripts_lda = pipelineModel.transform(transcripts)

topics = ldaModel.describeTopics(6)
topics = topics.withColumn("topicWords", indices_to_terms(countVectorModel.vocabulary)("termIndices"))
topics.select("topicWords").show(optimal_k,truncate=False)

from pyspark.sql.types import IntegerType, FloatType
from pyspark.sql.functions import udf,col
func=udf(lambda v: int(np.argmax(v)+1), IntegerType()) #add one since index starts at 0
transcripts_lda = transcripts_lda.withColumn("topic", func(col("topicDistribution")))
display(transcripts_lda.select('topic'))
#display(transcripts_lda.select("topicDistribution"))

# Instead of fussing with Databricks, export columns by selecting and using the GUI.
to_export = transcripts_lda.select(['topic','SubSector','Change','AboveMedian'])
display(to_export)
```