Twitter Analysis: Project 3 Presentation

Ryan Downing, Stefan Obradovic, Sahil Goel, Candace Austin, Rohan Samy, Kafana Ouattara

Dataset Description

We used our dataset from Project 1, which consisted of 313 tweets from individuals on Twitter regarding the technology sector

- Twitter posts made between March 01, 2020 and March 12, 2022
- At most 10 tweets per analyst
- Tweets specifically mentioning the following tickers: AAPL, MSFT, GOOG, AMZN, TSLA, FB, NVDA, AVGO, CSCO, ADBE, CRM, ORCL, ITC, AMD, NFLX, TXN, INTU, PYPL, NOW, IBM, QQQ, TQQQ, ARKK

Collecting the Dataset

Used Python's tweepy Twitter API

- Collected list of twitter usernames from active users in the financial analyst community
- Obtained all tweets for each user in the set time range
- Filtered tweets which did not contain any cashtags (i.e. $AAPL, $MSFT)

```
def get_users_tweets_between_dates(self, user, start_date, end_date, **kwargs):
        start_date_str = start_date.replace(tzinfo=pytz.UTC).isoformat()
        end_date_str = end_date.replace(tzinfo=pytz.UTC).isoformat()
        tweets = []
        while True:
            new_tweets = self.get_users_tweets(
              user, max_results=100, end_time=end_date_str,
              tweet_fields=["id", "text", "created_at", "author_id"], **kwargs
            )
            if new_tweets is None:
                return tweets

            new_tweets = list(reversed(new_tweets))
            if len(new_tweets) < 100:
                return tweets + new_tweets

            oldest_tweet_date = new_tweets[0].created_at.replace(tzinfo=None)
            if oldest_tweet_date < start_date:
                for i, tweet in enumerate(new_tweets, 1):
                    date = tweet.created_at.replace(tzinfo=None)
                    if date >= start_date:
                        return tweets + new_tweets[i:]
                return tweets

            end_date_str = oldest_tweet_date.replace(tzinfo=pytz.UTC).isoformat()
            tweets.extend(new_tweets)
```

Author	author_id	created_at	id	Text	Date	Stock(s) Mentioned	Tweet Link	Misc.
DanielTNiles	1948086848	2020-05-04 03:49:04+00:00	1257155255852085249	On @CNBC talking about managing positions dail...	2020-05-04 03:49:04+00:00	AMZN	https://twitter.com/twitter/statuses/125715525...	
DanielTNiles	1948086848	2020-07-17 19:21:26+00:00	1284206597179228160	On @YahooFinance w/ @ZGuz about rotating into ...	2020-07-17 19:21:26+00:00	TSLA	https://twitter.com/twitter/statuses/128420659...	
DanielTNiles	1948086848	2021-05-12 15:24:59+00:00	1392501064193122304	As I said on @CNBC, I still like $VIAC. Q1 res...	2021-05-12 15:24:59+00:00	NFLX	https://twitter.com/twitter/statuses/139250106...	
DanielTNiles	1948086848	2020-07-30 18:32:12+00:00	1288905248241676288	Covering $AMZN short & getting long again....	2020-07-30 18:32:12+00:00	MSFT, AMZN	https://twitter.com/twitter/statuses/128890524...	
DanielTNiles	1948086848	2020-10-20 20:25:34+00:00	1318649583405309953	The power of social media/internet meeting tal...	2020-10-20 20:25:34+00:00	FB	https://twitter.com/twitter/statuses/131864958...	
...	...	...	...	...	...	...	...	...
The_RockTrading	21764428	2022-03-05 16:40:00+00:00	1500149136645050377	$FB Closing Friday below critical $200 support...	2022-03-05 16:40:00+00:00	FB	https://twitter.com/twitter/statuses/150014913...	
The_RockTrading	21764428	2022-02-16 13:13:23+00:00	1493936546591870986	$FB 👀	2022-02-16 13:13:23+00:00	FB	https://twitter.com/twitter/statuses/149393654...	
The_RockTrading	21764428	2022-03-05 16:20:00+00:00	1500144103488737288	$TSLA Doesn't look terrible like the rest of t...	2022-03-05 16:20:00+00:00	TSLA	https://twitter.com/twitter/statuses/150014410...	

Cleaning and Tokenizing the Text

Cleaning text before applying analysis

- Converted each tweet to a list of sentences using sent_tokenize function
- Removed all characters which are not alphabetical or whitespace (including emojis and cashtags) and then lowercased remaining characters
- Splitted on whitespace to get tokens, removed stopwords, and recreated sentences, resulting in a list of lists for each tweet

```
data["sentences"] = data["Text"].apply(sent_tokenize)
data.head()
def generate_tokens(sentence_list):
    final_tokens = []
    for sentence in sentence_list:
        new_text = re.sub("[^A-Za-z]"," ", sentence)
        tokens = new_text.lower().split()
        tokens = [el for el in tokens if el not in sw]
        if tokens!=[]:
            final_tokens.append(tokens)
    return final_tokens

sw = set(stopwords.words("english"))
data["clean_tokens"] = data["sentences"].apply(generate_tokens)
data.head()
```

Generate 300-dimensional word embeddings

- Used Word2Vec model from Sci-kit Learn library to estimate word embeddings

- Chose Skip-Gram model as a parameter to predict surrounding words given the target words

```
model = Word2Vec(data["clean_tokens"].sum(), size = 300, sg = 1, window = 5, min_count = 1, seed=1234)
```

Interesting words

Apple Most Similar Words

```
[('market', 0.456509470939636 23),
 ('earnings', 0.44108909368515015),
 ('amzn', 0.4371196925640106),
 ('aapl', 0.4283718168735504),
 ('tsla', 0.4207943081855774),
 ('stocks', 0.41929757595062256),
 ('fb', 0.41373759508132935),
 ('trend', 0.4108179807662964),
 ('year', 0.40746599435806274),
 ('googl', 0.4067089259624481)]
```

Facebook Most Similar Words

```
[('fb', 0.434287786483764 65),
 ('tsla', 0.40036195516586304),
 ('aapl', 0.3944771885871887),
 ('pypl', 0.38249915838241577),
 ('would', 0.37551170587539673),
 ('largest', 0.37386929988861084),
 ('nasdaq', 0.37191933393478394),
 ('tech', 0.3715526759624481),
 ('stocks', 0.3595224618911743),
 ('earnings', 0.35839802026748 66)]
```

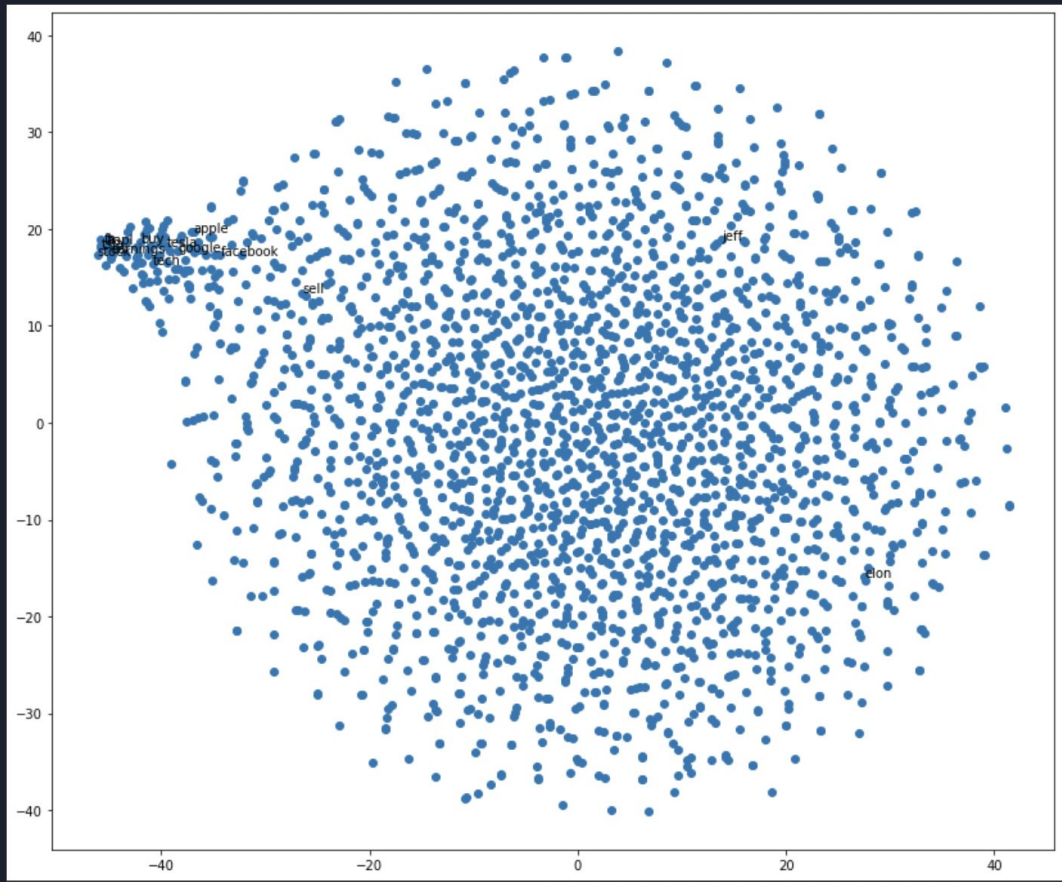
Tesla Most Similar Words

```
[('aapl', 0.5454764366149902),
 ('fb', 0.5433405637741089),
 ('amzn', 0.5176622867584229),
 ('tsla', 0.4932287931442261),
 ('googl', 0.4873512387275696),
 ('earnings', 0.48251521587371826),
 ('bitcoin', 0.48001861572265625),
 ('see', 0.47951188683509827),
 ('new', 0.4773513078689575),
 ('stock', 0.4764426350593567)]
```

300-Dimensional Word Embedding Projected

Words Labeled

apple, aapl, tesla, tsla, facebook, fb, stock, tech, google, jeff, elon, earnings, buy, sell



Word Relationships - Window Size of 5

```
# Model Identifies Market Sentiment
print(model.wv.most_similar(negative=["risk"],positive=["volatility","bitcoin"], topn=10))
print(model.wv.most_similar(negative=["tesla"],positive=["long","facebook"], topn=10))
print('--------------------------')
# Model does not catch synonyms for negative events
print(model.wv.most_similar(negative=["long"],positive=["increase","short"], topn=10))
print(model.wv.most_similar(negative=["buy"],positive=["bullish","sell"], topn=10))
print(model.wv.most_similar(negative=["good"],positive=["bad","buy"], topn=10))
print('--------------------------')
# Model catches information about market indices, stocks/tickers, & CEOs
print(model.wv.most_similar(negative=["nasdaq"],positive=["market","tesla"], topn=10))
print(model.wv.most_similar(negative=["facebook"],positive=["fb","tesla"], topn=10))
print(model.wv.most_similar(negative=["elon"],positive=["tesla","jeff"], topn=10))
print('--------------------------')
# Model does not catch information about company businesses
print(model.wv.most_similar(negative=["tesla"],positive=["car","amazon"], topn=10))
print(model.wv.most_similar(negative=["netflix"],positive=["streaming","tesla"], topn=10))
```

```
[('high', 0.2376978099346161), ('prices', 0.22642448544502258), ('tsla', 0.22160924971103668), ('amazon',
[('high', 0.24155691266059875), ('fb', 0.22594892978668213), ('results', 0.21478328108787537), ('like', 0
--------------------------
[('q', 0.26205214858055115), ('aapl', 0.2620397210121155), ('see', 0.2433638721704483), ('new', 0.2316661
[('months', 0.18381169438362122), ('amat', 0.174531489610672), ('example', 0.16746389865875244), ('muted'
[('v', 0.28860074281692505), ('see', 0.2848662734031677), ('us', 0.27734076976776123), ('back', 0.2640242
--------------------------
[('stock', 0.4615834951400757), ('tsla', 0.4566470980644226), ('new', 0.44742727279663086), ('earnings',
[('stock', 0.5244141817092896), ('tsla', 0.5210899114608765), ('earnings', 0.49599573016166687), ('new',
[('amzn', 0.3422868549823761), ('stock', 0.3141649067401886), ('earnings', 0.30252301692962646), ('tsla',
--------------------------
[('mean', 0.19809049367904663), ('android', 0.19797280430793762), ('credits', 0.19227232038974762), ('dvc
[('tsla', 0.43583664298057556), ('company', 0.42317134141921997), ('x', 0.4096143841743469), ('bitcoin',
```

We find that the embedding is good at associating market sentiment, and associating exchange, CEO, and ticker information with companies.

The embedding is not good at associating products and industries to companies and is also not good at associating synonyms for negative sentiment.

The last finding is likely due to a biased sample: Twitter users often know the industries of companies they discuss without mentioning it explicitly. Users often only use one sentiment word in a tweet so hard to catch synonyms.

Repeated for Window Size of 10

```
model = Word2Vec(data["clean_tokens"].sum(), size = 300, sg = 1, window = 10, min_count = 1, seed=1234)
```

Apple Most Similar Words

```
('tsla', 0.9487775564193726)
('market', 0.9484710097312927)
('amzn', 0.9446272850036621)
('like', 0.9444700479507446)
('stocks', 0.9443486928939819)
('bitcoin', 0.943225085735321)
('new', 0.9424498081207275)
('week', 0.9423635601997375)
('billion', 0.9415996670722961)
('fb', 0.9399840831756592)
```

Facebook Most Similar Words

```
('largest', 0.8963330984115601)
('aapl', 0.8947012424468994)
('p', 0.8910495042800903)
('new', 0.8909953236579895)
('amzn', 0.8893800377845764)
('tsla', 0.8892759680747986)
('year', 0.8891743421554565)
('nasdaq', 0.8889255523681641)
('stocks', 0.8870859146118164)
('like', 0.8858124017715454)
```

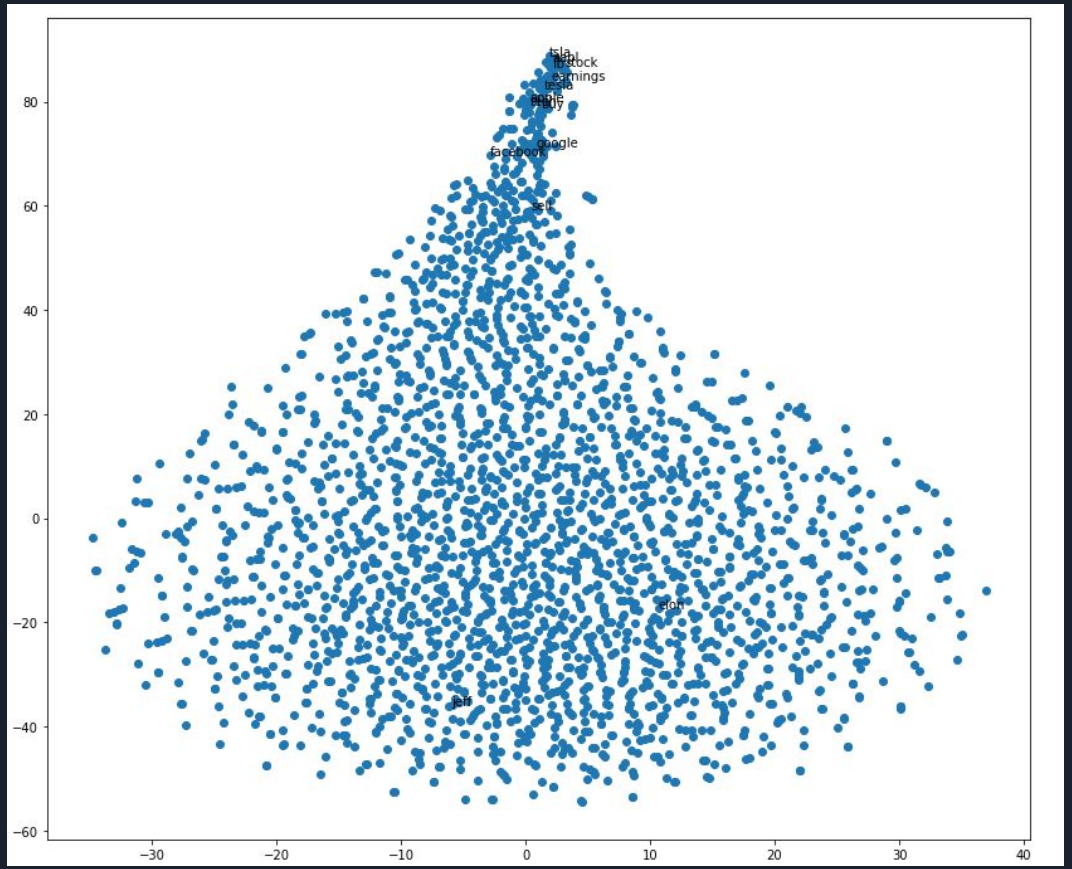
Tesla Most Similar Words

```
('tsla', 0.9587001800537109)
('like', 0.9552483558654785)
('amzn', 0.9539690017700195)
('aapl', 0.9529201984405518)
('billion', 0.9522708058357239)
('bitcoin', 0.9519350528717041)
('market', 0.9518647193908691)
('see', 0.9496039152145386)
('week', 0.9495173692703247)
('stock', 0.9478133916854858)
```

Word Embedding Dimensionality Reduction

Words Labeled

apple, aapl, tesla, tsla, facebook, fb, stock, tech, google, jeff, elon, earnings, buy, sell



Word Relationships - Window Size of 10

```
# Model Identifies Market Sentiment
print(model.wv.most_similar(negative=["risk"],positive=["volatility","bitcoin"], topn=10))
print(model.wv.most_similar(negative=["tesla"],positive=["long","facebook"], topn=10))
print('--------------------------')
# Model does not catch synonyms for negative events
print(model.wv.most_similar(negative=["long"],positive=["increase","short"], topn=10))
print(model.wv.most_similar(negative=["buy"],positive=["bullish","sell"], topn=10))
print(model.wv.most_similar(negative=["good"],positive=["bad","buy"], topn=10))
print('--------------------------')
# Model catches information about market indices, stocks/tickers, & CEOs
print(model.wv.most_similar(negative=["nasdaq"],positive=["market","tesla"], topn=10))
print(model.wv.most_similar(negative=["facebook"],positive=["fb","tesla"], topn=10))
print(model.wv.most_similar(negative=["elon"],positive=["tesla","jeff"], topn=10))
print('--------------------------')
# Model does not catch information about company businesses
print(model.wv.most_similar(negative=["tesla"],positive=["car","amazon"], topn=10))
print(model.wv.most_similar(negative=["netflix"],positive=["streaming","tesla"], topn=10))
```

```
[('prices', 0.37570518255233765), ('high', 0.3660631775856018), ('saturday', 0.3641311526298523), ('tes
[('year', 0.8209222555160522), ('amzn', 0.8176488280296326), ('growth', 0.8162639141082764), ('fb', 0.8
--------------------------
[('aapl', 0.7196254134178162), ('tsla', 0.7170669436454773), ('q', 0.7139182686805725), ('see', 0.7139
[('fed', 0.555022656917572), ('w', 0.5470331907272339), ('like', 0.5446999073028564), ('big', 0.542337
[('back', 0.45502030849456787), ('u', 0.45290103554725647), ('v', 0.45129960775375366), ('us', 0.45027
--------------------------
[('tsla', 0.9210186004638672), ('stock', 0.9167298078536987), ('new', 0.913523256778717), ('like', 0.91
[('tsla', 0.8942036628723145), ('aapl', 0.8907740712165833), ('see', 0.8866246938705444), ('new', 0.885
[('trillion', 0.5202465057373047), ('amzn', 0.5201020240783691), ('nflx', 0.5146331787109375), ('shop',
--------------------------
[('finally', 0.35021376609802246), ('think', 0.3340514004230499), ('p', 0.3186957836151123), ('e', 0.31
[('bitcoin', 0.795515775680542), ('company', 0.7905641794204712), ('tsla', 0.7904973030090332), ('x', (
```

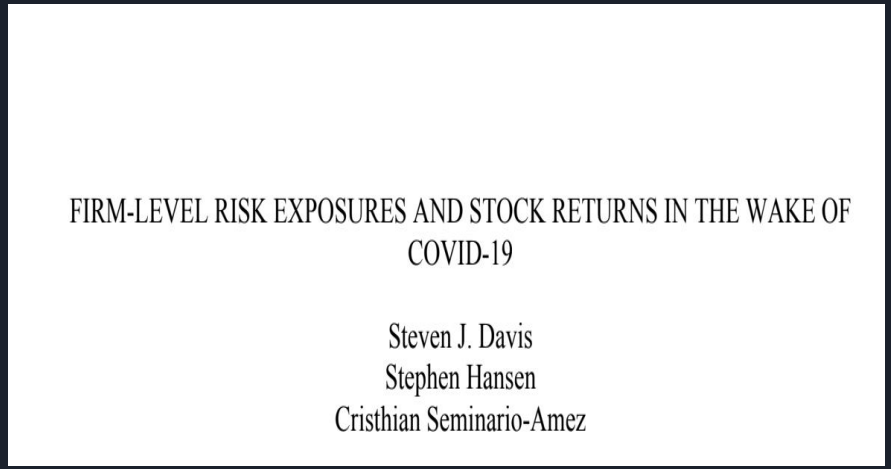
The embedding with a window size of 10 is actually worse than the previous embedding with a window size of 5.

Worse performance on sentiment association, equally bad at finding synonyms for negative sentiment words, slightly worse at catching company CEO associations. Maybe better at catching Tesla business operations by predicting (bitcoin).

The lower performance is likely due to the fact that Tweets are short-form text with a maximum of 280 characters in length so sentences tend to have shorter long-term semantic associations than other writing forms.

Paper Summary - Overview

- The global economic activity has been severely affected by the COVID-19 shock. Its heterogeneity reveals important details about the nature of the shock.
- The authors used different approaches to explain what drives firm level return reactions to common stocks
- The approaches are the 10-K Risk Factor discussion and the two text analytics (Expert-Curated dictionaries and Supervised Machine learning)


FIRM-LEVEL RISK EXPOSURES AND STOCK RETURNS IN THE WAKE OF COVID-19

Steven J. Davis
Stephen Hansen
Cristhian Seminario-Amez

Paper Summary - Approach & Methodology

- Analyze daily returns for 2,155 equity securities on the 20 **jump days** from February 24 to March 27 that Baker et al. (2020a) identify and classify
- Process **Risk Factors** from 2010 to 2016
    - Required to be filed by SEC to notify shareholders of any item that could impact future earnings
- Model the relationships using **curated dictionaries** and the **multinomial inverse regression model (MNIR)**

Paper Summary - Approach & Methodology

- Curated dictionaries include 16 dictionaries that cover economic and financial conditions and 20 that pertain to policy areas
    - Judgment of expert economists drawing on textbooks, newspaper articles, 10-K filings, and “their own knowledge of economic matters and input from other economists in seminars and personal communications”
- MNIR treats the RF texts for each firm i as a bag-of-words represented by a V-dimensional (V=18911) vector x_i of terms or “features.” x_i,v is the count of term v for firm i
    - Allows more flexibility in the relationship between returns and terms
    - Terms can have different regression coefficients, rather than per category

Paper Summary - Results (Curated Dictionaries)

- Fit regression models for daily firm-level returns via least-squares estimation

$$\text{Abn}_{it} = \sum_{j=1}^{J} \beta_j \, \text{RExp}_i^j + \beta_{J+1} \, \text{Leverage}_i + \beta_{J+2} \, \log(\text{Mcap}_{it}) + \gamma_{s(i)} + \epsilon_{it}$$

- Adjusted R2 values range from 20% the day after Super Tuesday to 33% on pandemic fallout days and 36% for the March 9 Oil Price Crash
- The large positive coefficient on intellectual property shows that bad news about the COVID-19 pandemic is relatively good for firms that own or develop healthcare-related intellectual property.
- Jumps attributed to monetary policy easing yield large positive return reactions for firms with high exposures to inflation and interest rates but not to intellectual property or transportation, infrastructure and utilities.
- Jump days attributed to fiscal policy news generate the largest return reaction at firms with high exposure to the tax category. Firms exposed to tax-sensitive categories like real estate and business investment also outperform on fiscal policy jump days.
- The precise interpretation of some of these patterns is unclear.
    - The tax category, for example, captures exposures to both high taxes and the potential for large tax credits (e.g., for R&D or investment).

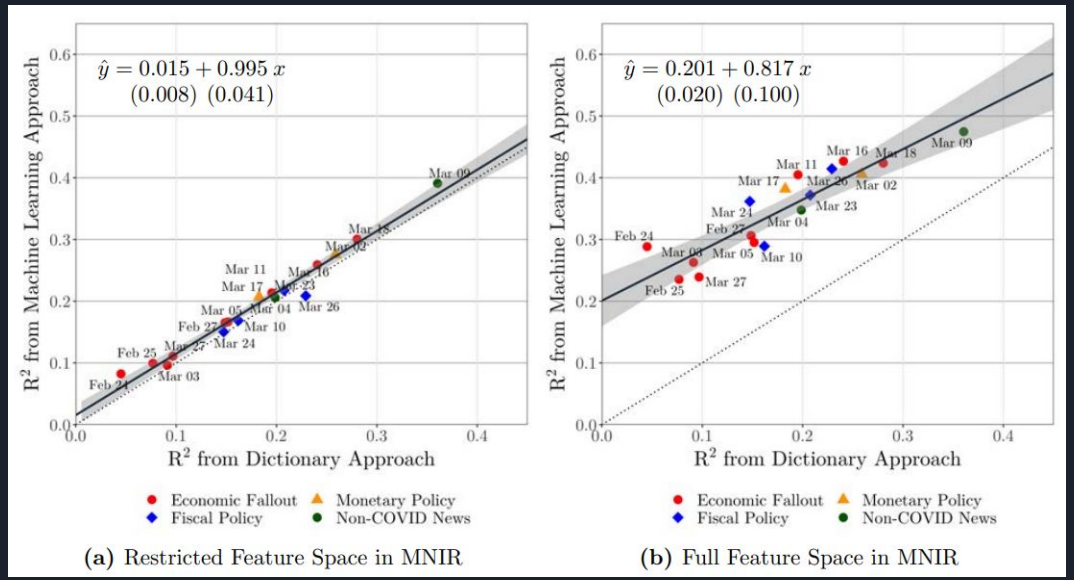
Paper Summary - Results (MNIR)

- Estimate the inverse regression separately for each jump day or collection of days with controls $\mathbf{c}_{it} = (\text{Leverage}_i, \log(\text{Mcap}_{it}), \gamma_{s(i)})$

$$Abn_{it} = \alpha_1 \, z_{it} + \alpha_2 \, N_i + \alpha_3 \, Leverage_i + \alpha_4 \, Log(Mcap_{it}) + \gamma_{s(i)} + \varepsilon_{it}$$

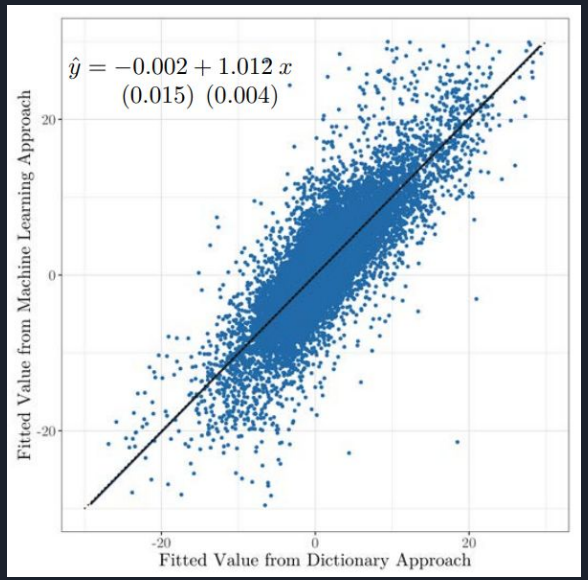
- Test with a full MNIR feature space (V=18911) and a restricted feature space (V=244)
    - Restricted feature space corresponds to the feature representation of the curated dictionaries

Paper Summary - Results (MNIR)



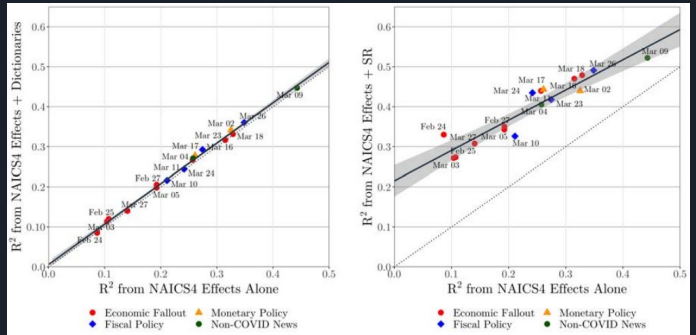
(a) Restricted Feature Space in MNIR

(b) Full Feature Space in MNIR



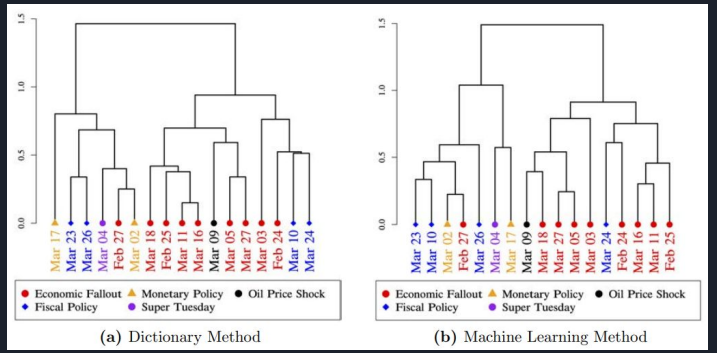
Paper Summary - Results

- Added the 216 unique NAICS4 codes to account for their systematic risks in the model
  - Improved both models and strengthened the correlation between results
- Use a hierarchical clustering algorithm to group similar risk days together
  - With few exceptions, the clustering algorithm groups the “fiscal” and “monetary policy” jumps into distinct blocks, as it does for the “pandemic fallout” jumps



(a) Dictionary Method (b) Machine Learning Method

Figure 5: Improvement in R^2 beyond Narrow Industry Codes



(a) Dictionary Method (b) Machine Learning Method

Paper Summary - Conclusions

- Results provide evidence that COVID-19 accelerated ongoing shifts to digital services and remote interactions
    - The share of new U.S. patent applications that advance technologies to support video conferencing, telecommuting, remote interactivity, and working from home more than doubled in the wake of the pandemic
- Applies a hybrid approach based on a conventional dictionary-based approach
    - Low requirement for domain expertise
    - Avoids the laborious construction of expert-curated dictionaries, reliance on external libraries, and is flexible and adaptable

Thank You!!