FOMC Minutes Analysis

Can minutes forecast interest rate changes?

Group 4D-Intelli
Members: Yuhang Xia, Chonwai Ho, Hyun Jin Cho, Xueyang Wei, Jiahao Sun, Cheng Peng
Mar 9th, 2019

CONTENT

Progress Check

Research

Web Scrapping

Tokenization & Bag of Words

TF-IDF

4D-Intelli

Progress Check

FOMC Minutes Analysis

Web Scrapping	100%: Complete Scraping of FOMC data
Data Merging	100%: Scrapped text data evaluated and merged using pandas dataframe
Tokenization	100%: Completed using package sklearn
Bag of Words	100%: Bag of Word analysis using package sklearn and NLTK Counter packager
Tf-idf	100%: Tf-idf have been applied to minutes
Machine Learning	10%: Research findings and machine learning tools under discussion

4D-Intelli

Research Papers

- The Information Content of FOMC Minutes (Boukus and Rosenberg 2006)
- Latent semantic analysis of the FOMC statements (Panagiotis Mazis, Andrianos Tsekrekos, 2017)

Purpose

- The two papers identifies the textual patterns and themes to examine the impact on the US Treasury market
- The later paper was built on top of Boukus and Rosenberg, 2006 and extends their findings

4D-Intelli

Latent semantic analysis of the FOMC statements

Keep Mitigating Risk & Improving Return

Data Set

- Uses FOMC statements from 2003 to 2014
- A total of 99 minutes

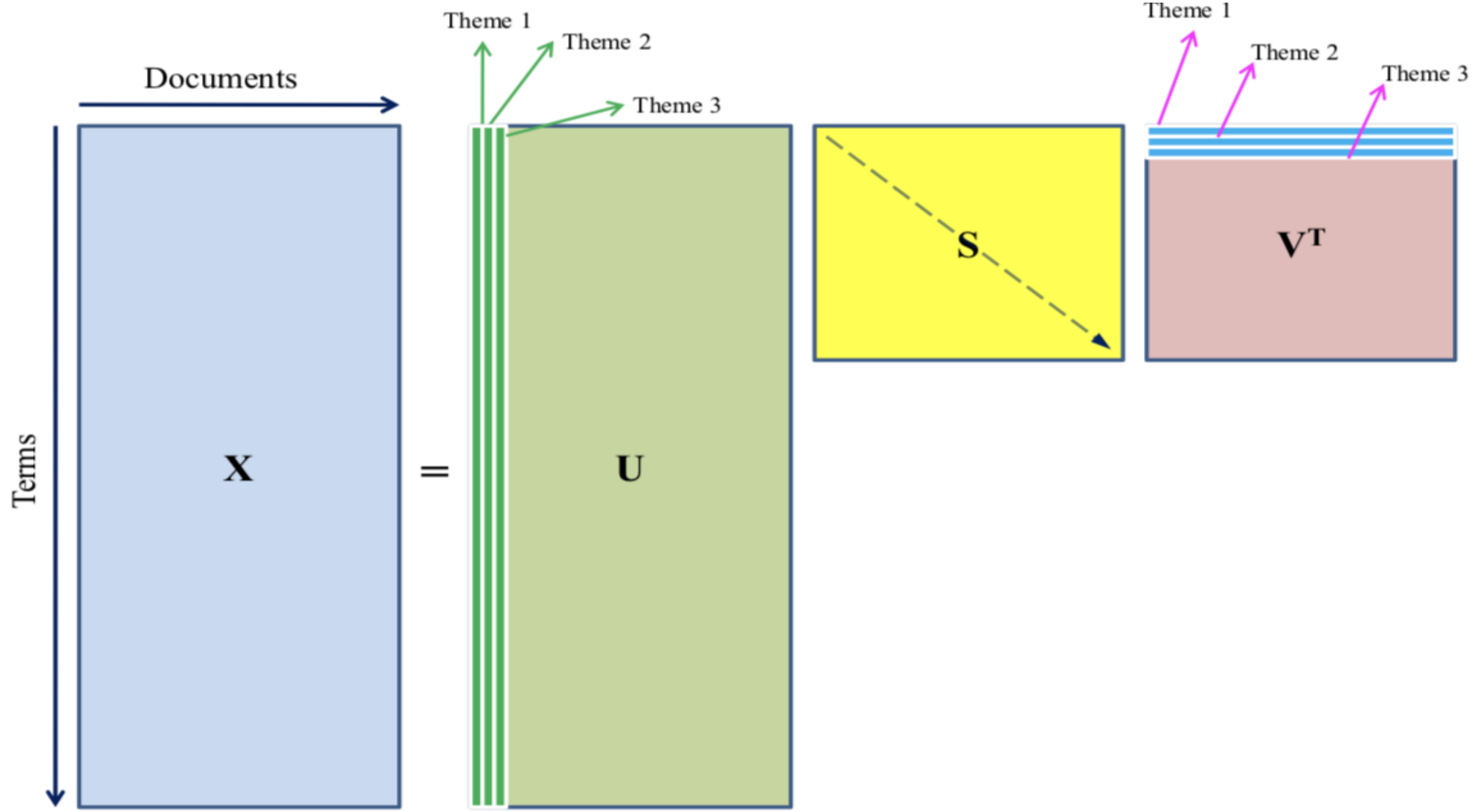
Text Processing

- Eliminate text formatting
- Remove stop words
- Lemmatization
- Porter Stemming
- Filter using LM list

Analysis

- Latent Semantic Analysis(LSA)
- Extract themes using singular value decomposition (SVD)

Latent semantic analysis of the FOMC statements



4D-Intelli

Latent semantic analysis of the FOMC statements

4D-Intelli

Conclusion

- Mazis and Tsekrekos were able to use LSA to explain total policy variation
- Establishes the use of LSA to produce a set of themes to characterize the FOMC statements
- Increased communication and transparency of the FOMC has enhanced the understanding of future Fed actions
- It is a challenging task due to the nuanced and multifaced tone of the minutes!

Other Papers

- Have minutes helped to predict fed funds rate changes? (Jung 2016)
- Evaluating qualitative forecasts: The FOMC minutes, 2006–2010 (Stekler and Symington, 2015)

Web Scrapping

What does this code do? – Beautiful Soup and Requests

```
from bs4 import BeautifulSoup
import requests
import re
import urllib.request
import os

base_url = "https://www.federalreserve.gov/monetarypolicy/fomchistorical"

transcript_links = {}
for year in range(1968, 1993):
  html_doc = requests.get(base_url + str(year) +'.htm')
  soup = BeautifulSoup(html_doc.content, 'html.parser')
  links = soup.find_all("a", string=re.compile('Minutes*'))
  link_base_url = "https://www.federalreserve.gov"
  transcript_links[str(year)] = [link_base_url + link["href"] for link in links]
  print("Year Complete: ", year)
```

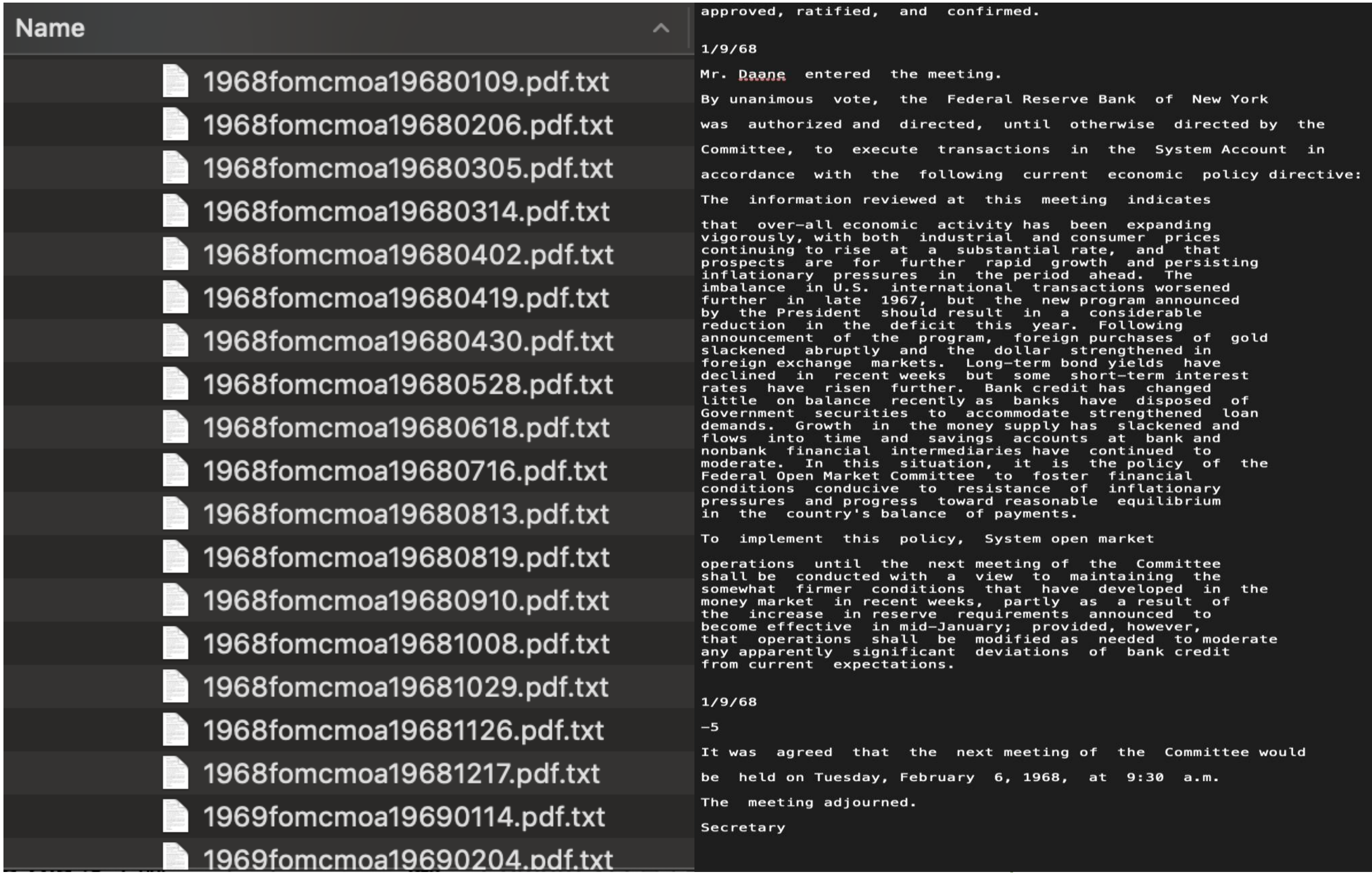
4D-Intelli

Web Scrapping

What does this code do? – Beautiful Soup and Requests

```
for year in transcript_links.keys():
    if not os.path.exists("./feddata/" + year):
        os.makedirs("./feddata/" + year)
    for link in transcript_links[year]:
        response = urllib.request.urlopen(str(link))
        name = re.search("[^/]*$", str(link))
        print(link)
        with open("./feddata/" + year + "/" + name.group(), 'wb') as f:
            f.write(response.read())
        print("file uploaded")
```

Loading Data



Tokenization & Bag of Words(BoW)

Data Processing

```
def lem_term(document):
    document = re.sub('\d+|\_', '', document)
    wnl = WordNetLemmatizer()
    tokens = word_tokenize(document)
    lem_token = [wnl.lemmatize(word) for word in tokens]
    for char in (['.', ',']):
        while char in lem_token:
            lem_token.remove(char)
    document = ' '.join(lem_token)
    re.sub('\+d','',document)
    return document
```

Tokenization & Bag of Words(BoW)

Bag of words

```
#%% bag of words with sklearn
vectorizer = CountVectorizer(stop_words = 'english',lowercase = True)
AnnualBow = vectorizer.fit_transform(corpus)
df_AnnualBow = pd.DataFrame(AnnualBow.A,columns = vectorizer.get_feature_names())

##remove thoes meaningness words
for col in ['year','month','day','mr','meeting','committee']:
    try:
        df_AnnualBow.pop(col)
    except:
        continue
minutes_BoW_sk2 = pd.concat([minutes,df_AnnualBow],axis = 1)
```

4D-Intelli

Tf-Idf

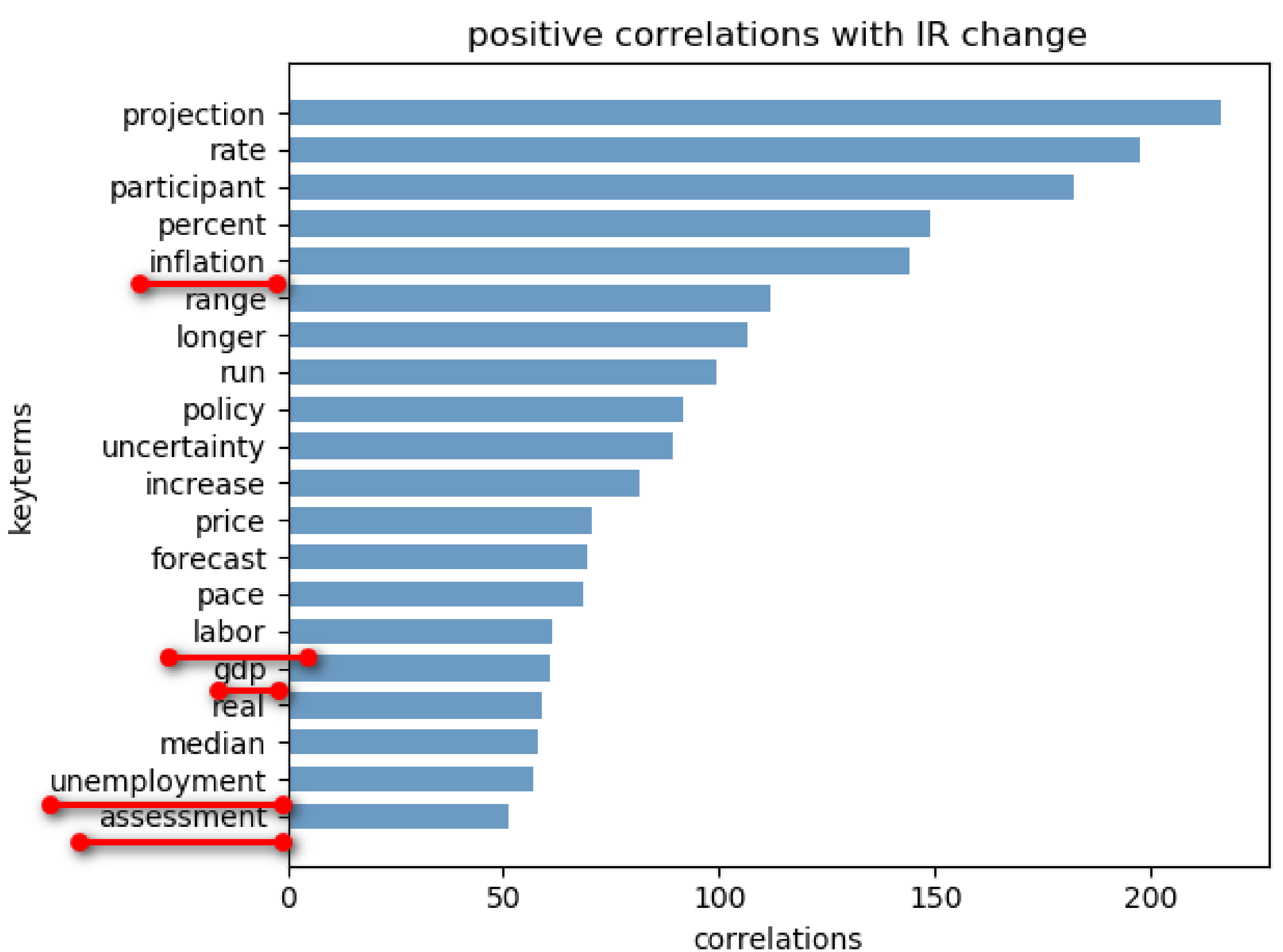
Assign weights by tf-idf model

```
v = TfidfVectorizer(stop_words='english', max_df=0.9)
tfidf = v.fit_transform(corpus)
df_Annualtfidf = pd.DataFrame(tfidf.A,columns = v.get_feature_names())

for col in ['year','month','day','mr','meeting']:
    try:
        df_Annualtfidf.pop(col)
    except:
        continue

minutes_tfidf_sk = pd.concat([minutes,df_Annualtfidf],axis = 1)
```

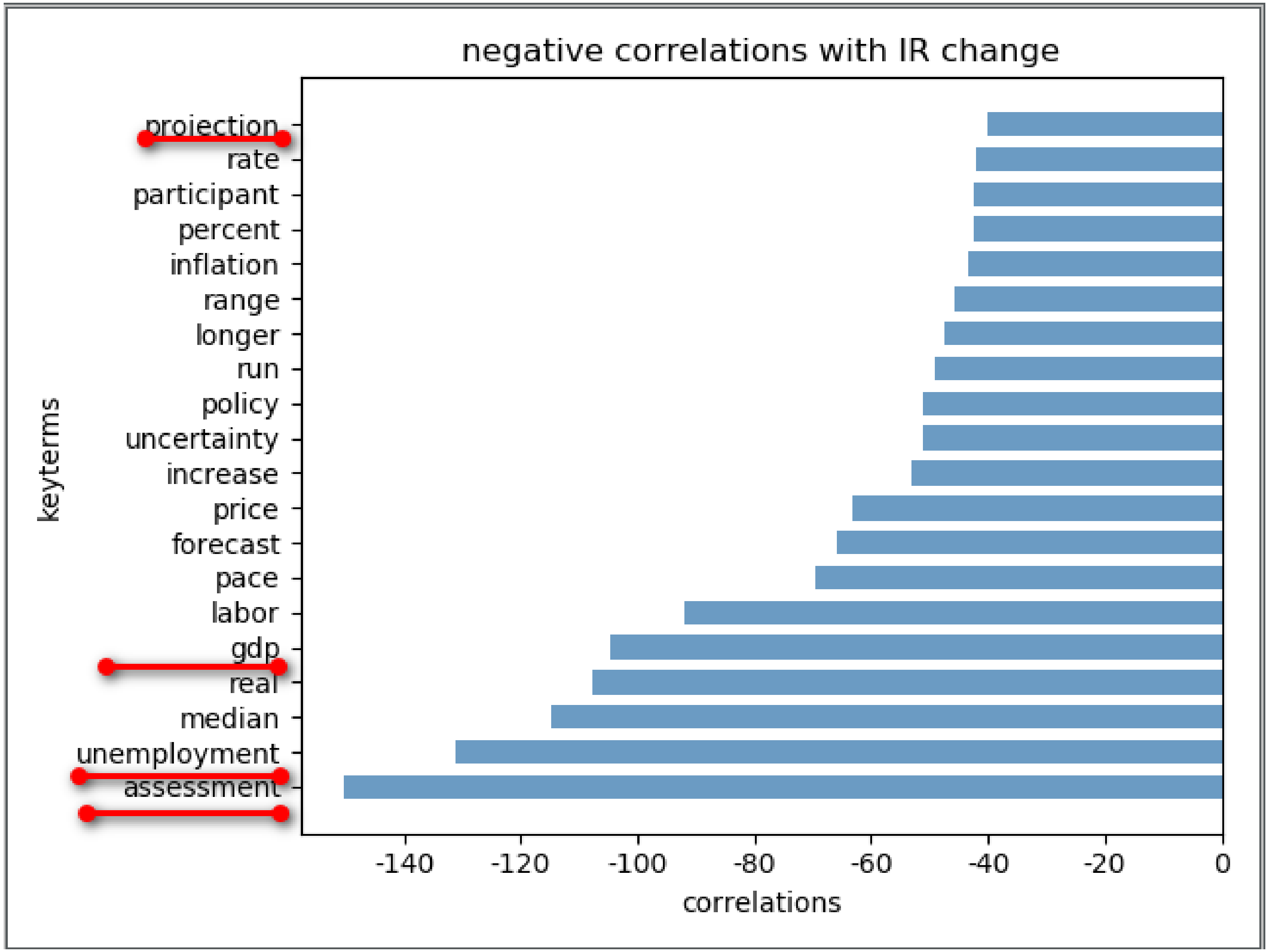
Preliminary analysis and what to do next?

```
correlations = [np.correlate(bow_IR_cor2['IR_Change'],bow_IR_cor2[term])[0]
                for term in list(bow_IR.columns[8:])]

IR_corKeyterm = pd.DataFrame({'keyterms':keyterms,'correlations':correlations})
bottom10 = IR_corKeyterm.sort_values(by = 'correlations') >> head(20)
top10 = IR_corKeyterm.sort_values(by = 'correlations') >> tail(20)
#%% plot graph
## first with positive correlations
import matplotlib.pyplot as plt
import matplotlib
matplotlib.rcParams['axes.unicode_minus'] = False

correlation = top10['correlations']
keyterms = top10['keyterms']
plt.barh(range(len(keyterms)), correlation, height=0.7, color='steelblue', alpha=0.8)
plt.yticks(range(len(keyterms)), keyterms)
plt.xlabel("correlations")
plt.ylabel('keyterms')
plt.title("positive correlations with IR change")
plt.show()
```

Preliminary analysis and what to do next?



Negative results with IR change



OLS Regression Results

Dep. Variable:	IR_Change	R-squared:	0.022
Model:	OLS	Adj. R-squared:	0.012
Method:	Least Squares	F-statistic:	2.121
Date:	Fri, 08 Mar 2019	Prob (F-statistic):	0.0618
Time:	19:26:05	Log-Likelihood:	172.94
No. Observations:	469	AIC:	-333.9
Df Residuals:	463	BIC:	-309.0
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.0212	0.010	-2.178	0.030	-0.040	-0.002
projection	0.0005	0.001	0.350	0.726	-0.002	0.003
inflation	8.921e-05	0.001	0.132	0.895	-0.001	0.001
unemployment	0.0008	0.002	0.341	0.733	-0.004	0.005
gdp	-0.0023	0.004	-0.599	0.550	-0.010	0.005
assessment	0.0045	0.005	0.877	0.381	-0.006	0.015

Omnibus:	311.938	Durbin-Watson:	1.243
Prob(Omnibus):	0.000	Jarque-Bera (JB):	6534.033
Skew:	-2.518	Prob(JB):	0.00
Kurtosis:	20.579	Cond. No.	50.5

OLS Regression Results

Dep. Variable:	IR_Change	R-squared:	1.000
Model:	OLS	Adj. R-squared:	1.000
Method:	Least Squares	F-statistic:	2.141e+28
Date:	Fri, 08 Mar 2019	Prob (F-statistic):	0.00
Time:	19:22:52	Log-Likelihood:	14909.
No. Observations:	469	AIC:	-2.974e+04
Df Residuals:	428	BIC:	-2.957e+04
Df Model:	40		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-7.477e-16	7.23e-16	-1.034	0.302	-2.17e-15	6.73e-16
unemployment	-3.721e-16	1.04e-16	-3.575	0.000	-5.77e-16	-1.68e-16
median	4.476e-16	1.35e-16	3.310	0.001	1.82e-16	7.13e-16
real	4.168e-16	1.03e-16	4.044	0.000	2.14e-16	6.19e-16
gdp	-4.077e-16	1.56e-16	-2.609	0.009	-7.15e-16	-1.01e-16
labor	-2.277e-17	6.51e-17	-0.350	0.727	-1.51e-16	1.05e-16
pace	3.014e-16	5.59e-17	5.389	0.000	1.91e-16	4.11e-16
forecast	-9.541e-17	1.14e-16	-0.835	0.404	-3.2e-16	1.29e-16
price	-3.652e-16	3.51e-17	-10.397	0.000	-4.34e-16	-2.96e-16
increase	7.86e-17	5.65e-17	1.391	0.165	-3.25e-17	1.9e-16
uncertainty	5.727e-16	1.13e-16	5.046	0.000	3.5e-16	7.96e-16
policy	3.253e-17	3.65e-17	0.890	0.374	-3.93e-17	1.04e-16
run	2.043e-16	8.24e-17	2.480	0.014	4.24e-17	3.66e-16
longer	-1.947e-16	7.8e-17	-2.497	0.013	-3.48e-16	-4.15e-17
range	1.897e-17	3.83e-17	0.495	0.621	-5.63e-17	9.43e-17
inflation	2.822e-16	3.22e-17	8.756	0.000	2.19e-16	3.46e-16

Latent Dirichlet allocation (LDA)

Data processing:

- Customize stop words
- Build the bigram and trigram models

```
stop_words = stopwords.words('english')
stop_words.extend(['meeting', 'january', 'mr', 'committee','federal'])

nlp = spacy.load('en_core_web_sm',disable=['parser', 'ner'])
```

...

```
def sent_to_words(sentences):
    for sentence in sentences:
        yield(gensim.utils.simple_preprocess(str(sentence), deacc=True))  # deacc=True removes punctuations
```

```
bigram = gensim.models.Phrases(corpus, min_count=5, threshold=100) # higher threshold fewer phrases.
trigram = gensim.models.Phrases(bigram[corpus], threshold=100)

# Faster way to get a sentence clubbed as a trigram/bigram
bigram_mod = gensim.models.phrases.Phraser(bigram)
trigram_mod = gensim.models.phrases.Phraser(trigram)
```

Latent Dirichlet allocation (LDA)

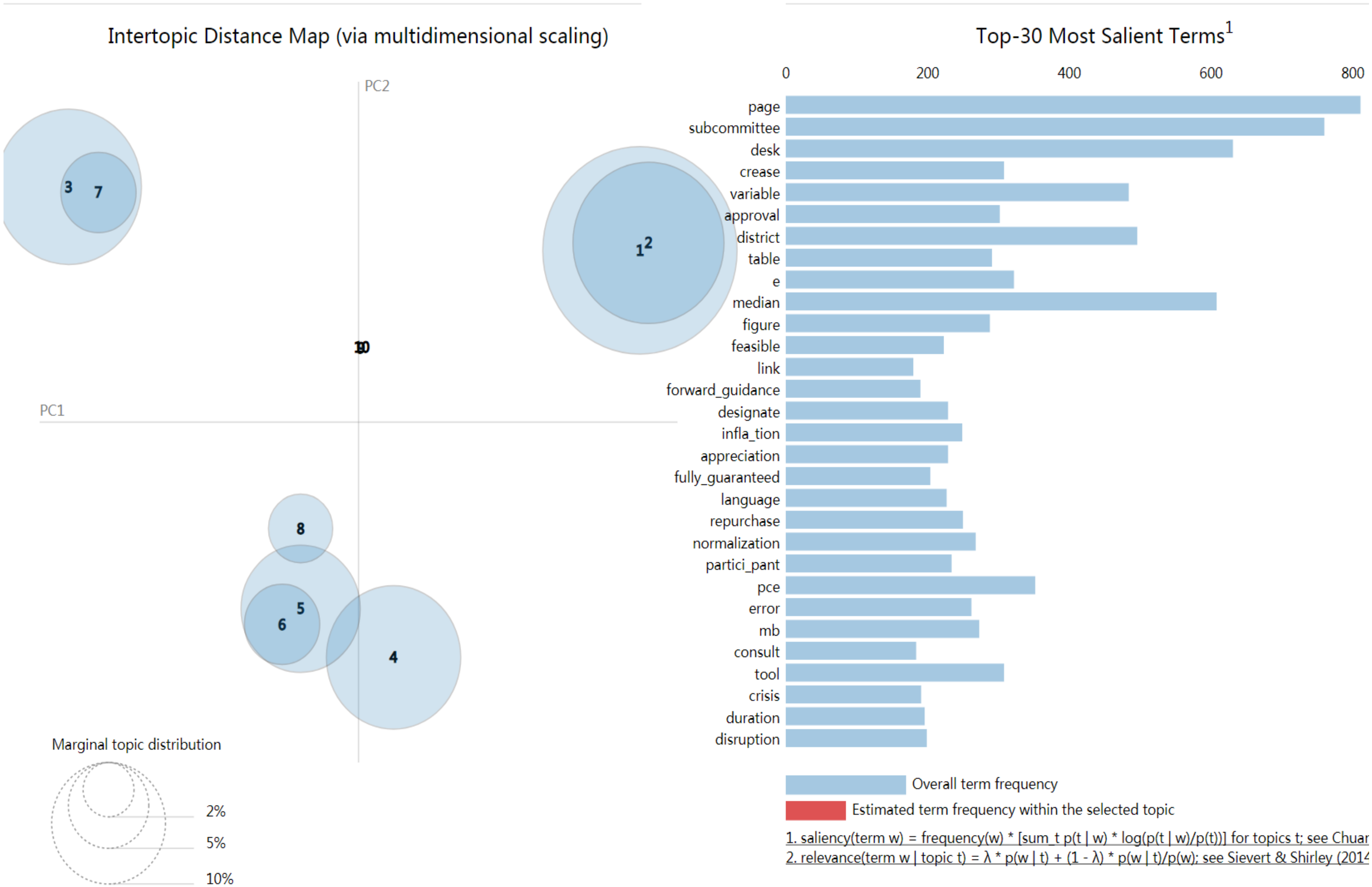
Establish models

C:\Users\Jesse\
:\Downloads\LDA

```
corpus_no_stops = remove_stopwords(corpus)
corpus_bigrams = make_bigrams(corpus_no_stops)
data_lemmatized = lemmatization(corpus_bigrams, allowed_postags=['NOUN', 'ADJ', 'VERB', 'ADV'])
# Create Dictionary
id2word = corpora.Dictionary(data_lemmatized)
id2word.filter_extremes(no_below=3, no_above=0.2)
# Create Corpus
texts = data_lemmatized
# Term Document Frequency
corpus = [id2word.doc2bow(text) for text in texts]
```

...

```
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                            id2word=id2word,
                                            num_topics=10,
                                            random_state=100,
                                            update_every=1,
                                            chunksize=100,
                                            passes=10,
                                            alpha='auto',
                                            per_word_topics=True)
```


Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
3
7
1
2
10
8
5
6
4
Marginal topic distribution
2%
5%
10%
Top-30 Most Salient Terms1
0
200
400
600
800
page
subcommittee
desk
crease
variable
approval
district
table
e
median
figure
feasible
link
forward_guidance
designate
infla_tion
appreciation
fully_guaranteed
language
repurchase
normalization
partici_pant
pce
error
mb
consult
tool
crisis
duration
disruption
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuar
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014

References

Sources:
The Information Content of FOMC Minutes (Boukus and Rosenberg 2006)
Latent semantic analysis of the FOMC statements (Panagiotis Mazis, Andrianos Tsekrekos, 2017)
Have minutes helped to predict fed funds rate changes? (Jung 2016)
Evaluating qualitative forecasts: The FOMC minutes, 2006–2010 (Stekler and Symington, 2015)

Whats Next???

Thank You!!!