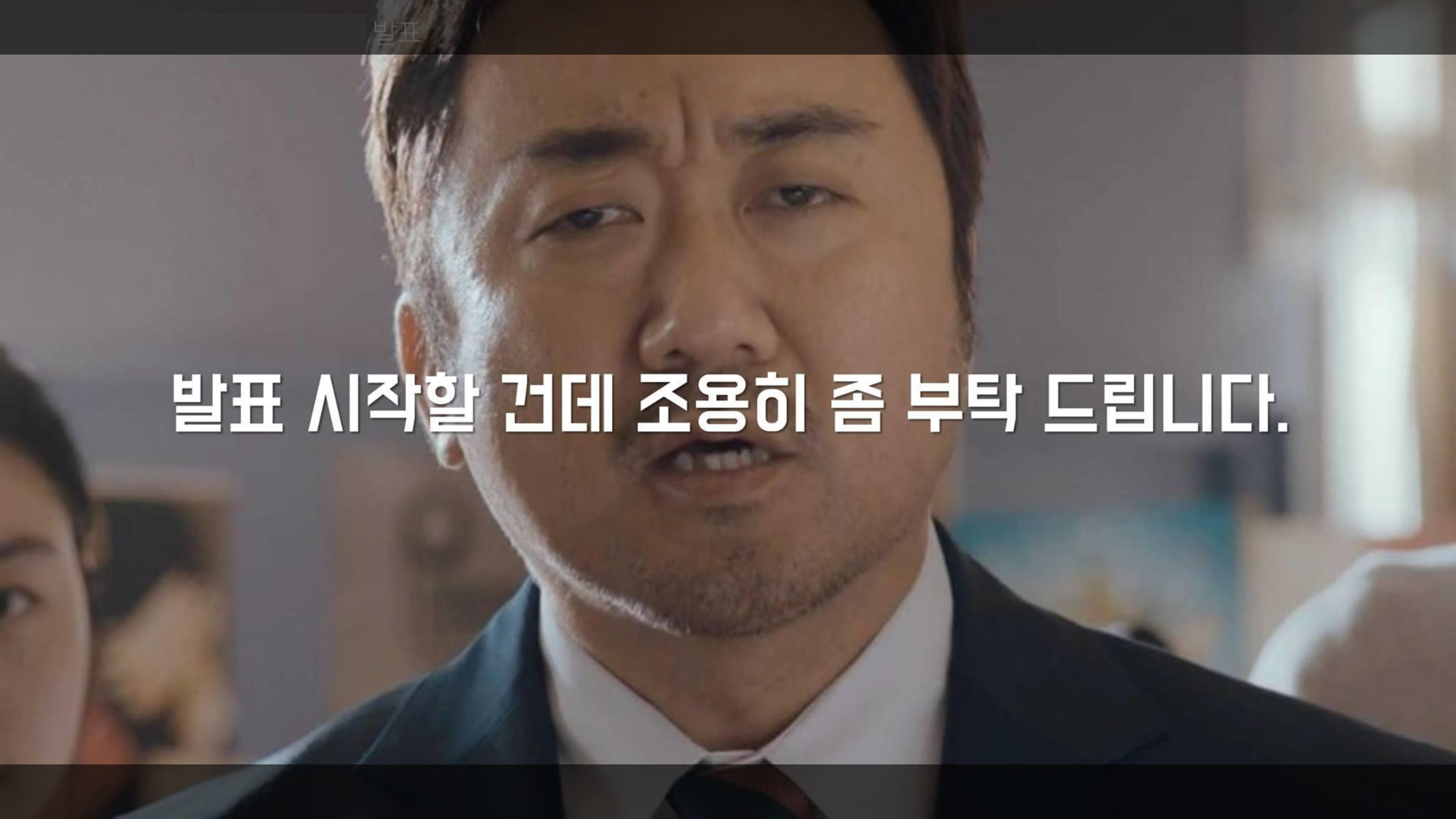
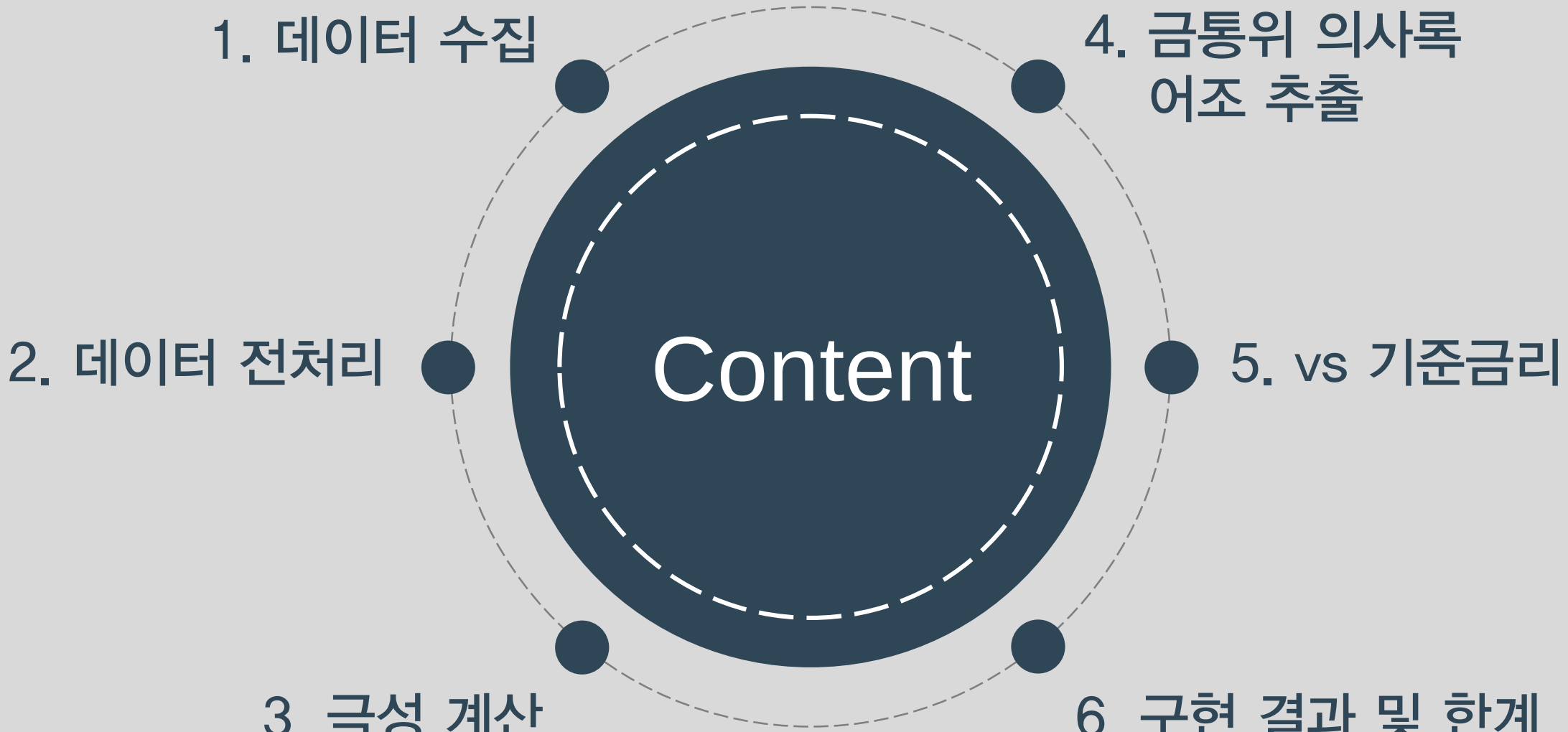

발표 시작할 건데 조용히 좀 부탁 드립니다.

Deciphering Monetary Policy Board Minutes through Text Mining Approach: The Case of Korea

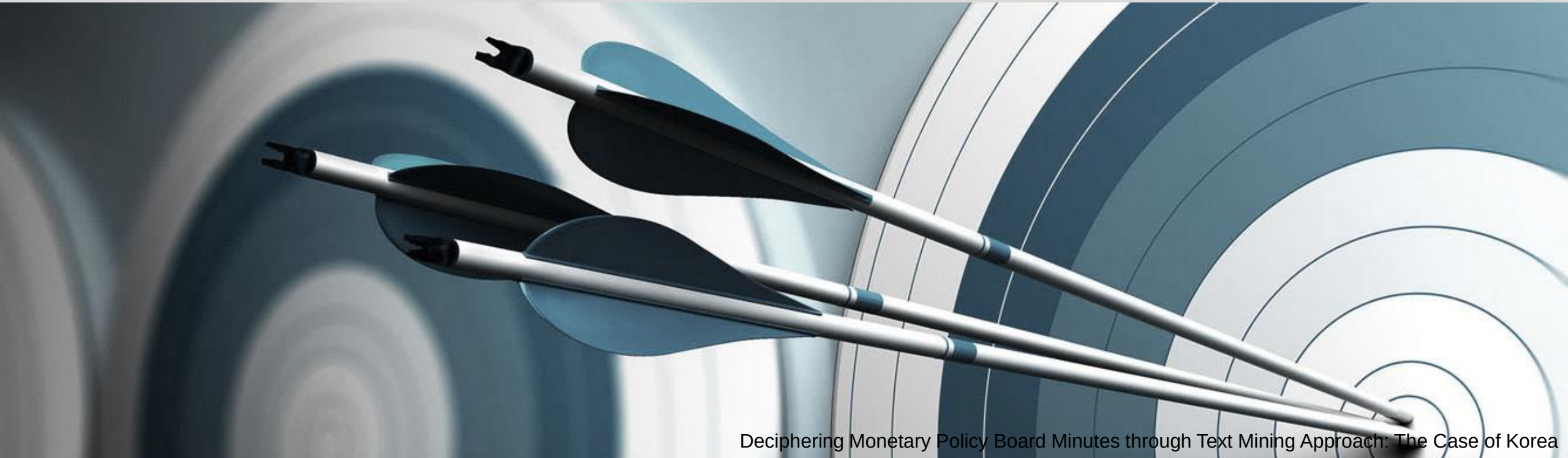
텍스트 마이닝을 활용한 금융통화위원회 의사록 분석

4조
편신범, 강정호, 국승지, 이선민


Content
1. 데이터 수집
2. 데이터 전처리
3. 극성 계산
4. 금통위 의사록
어조 추출
5. vs 기준금리
6. 구현 결과 및 한계

구현 목표

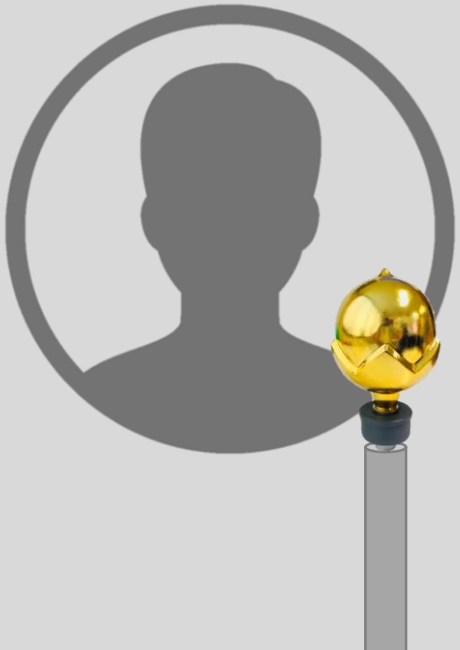
텍스트 마이닝을 활용하여 금통위 의사록에 담긴 어조를 추출하여 지수화(수치화) 하고,
이를 통해 기준 금리의 변동에 얼마나 많은 영향을 미쳤는지에 대한 설명력과 예측력을 검증



역할 분담

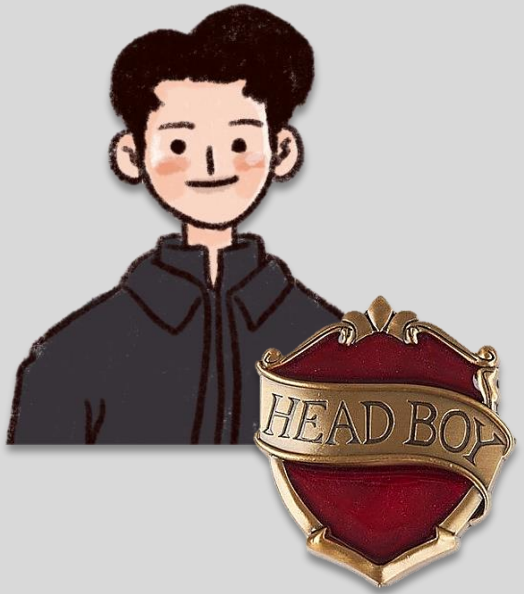
채권 분석 보고서
콜금리
N-gram count
PPT

연합뉴스
연합 인포맥스
불용어 처리
극성 계산



금통위 의사록
불용어 처리
N-gram count
금통위 의사록 어조 추출

이데일리
기준금리
극성 계산
데이터 시각화

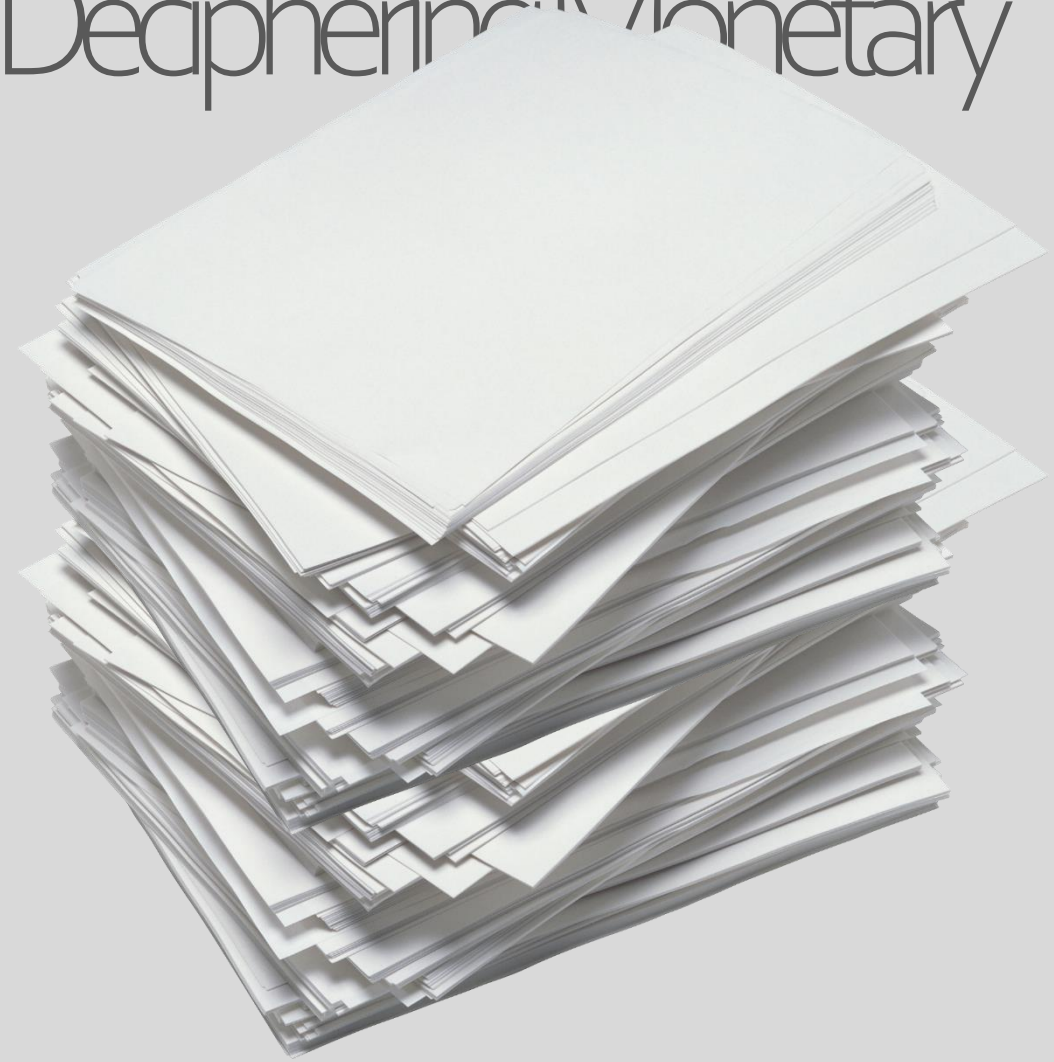


Newspaper, Bondreport, Deciphering Monetary

When?

2005~2017

+2



How many?

230,000

+20,000

Data Crawling

Crawling CallRate

```
# 1
df['old_date'] = df['date'] - timedelta(days = 30)

for i in range(len(callrate_list)):
        while df['old_date'][i] not in list(df['date']):
            df['old_date'][i] = df['date'] - timedelta(days = 1)
            if df['old_date'][i] in list(df['date']):
                df['old_callrate'][i] = df.loc[df['old_date'][i]]['callrate']
```

Crawling CallRate

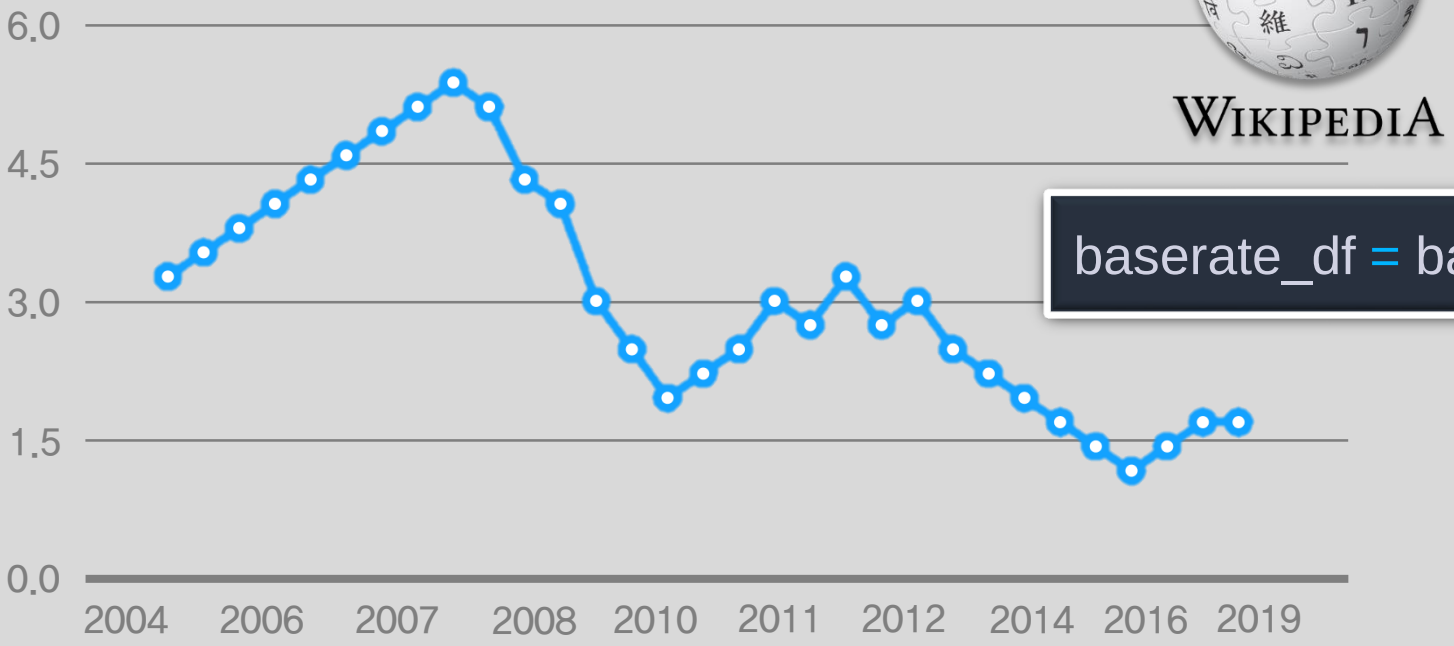
2005.01.01 ~ 2019.07.15

Up 2178
Down 2144
Neutrality 987

2005.01.01 ~ 2017.12.31

Up 1934
Down 1901
Neutrality 913

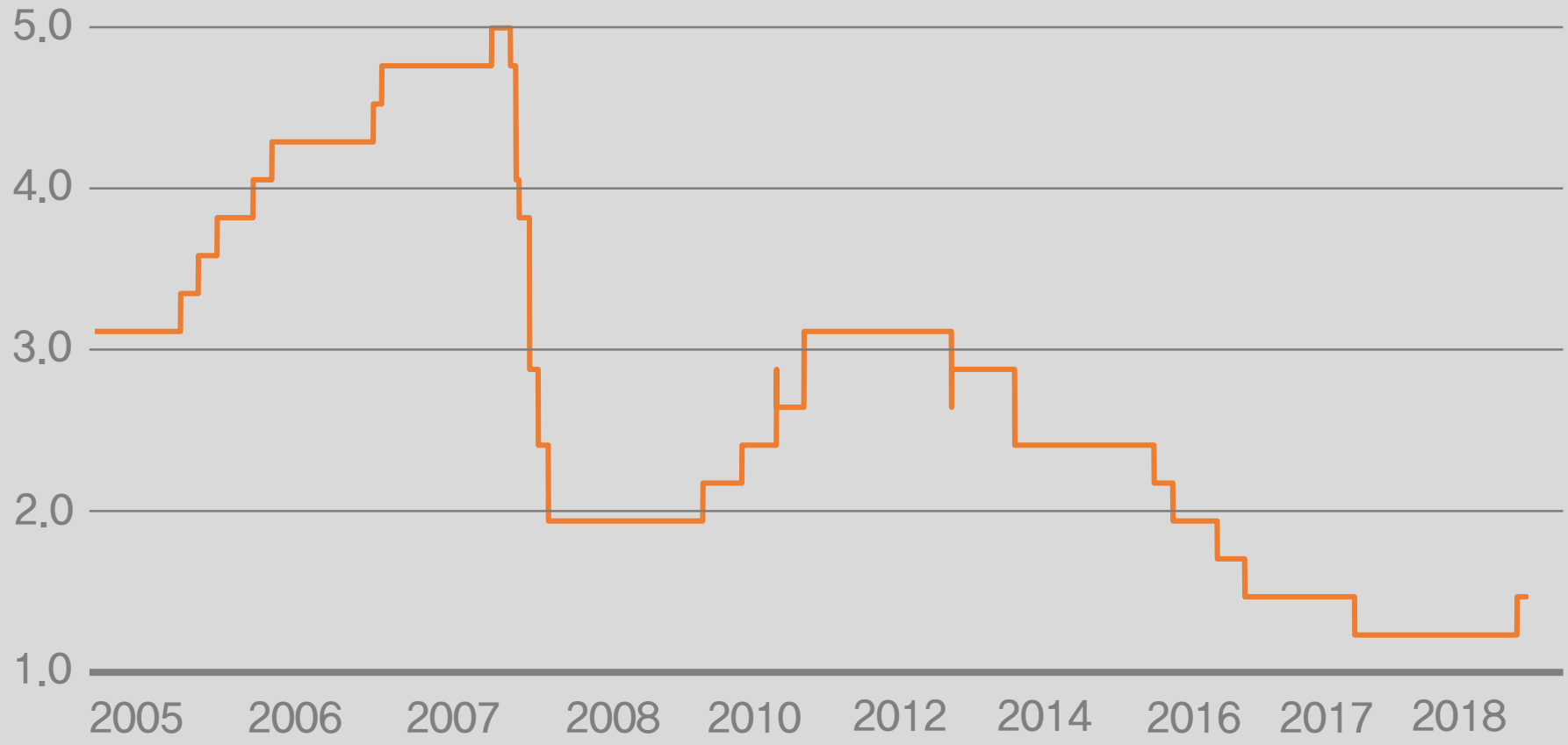
Crawling Base Rate



WIKIPEDIA

```
baserate_df = baserate_df.ffill(axis = 'rows')
```

Crawling Base Rate



Crawling Bond Reports

Request, BeautifulSoup를 사용하여 채권분석보고서 크롤링

```
# 1
if date is not None : # None 값이 뜰 때 손쉽게 제거 가능

# 2
errorcount = 0 # 에러가 발생할 때 마다 Error를 출력하고, errorcount 수를 셈
except Exception as e:
            print("============== {} ==============".format(str(e)))
            print("Error :" + text )
            errorcount += 1
            print("현재까지의 누적 Errors:  ", errorcount)
```

Crawling Bond Reports

pdf parser를 사용하여 pdf to txt 파일로 변환

```
# 1
for page in PDFPage.get_pages(fp):
            interpreter.process_page(page)
            data =  retstr.getvalue()
            text_path = os.path.join(파일경로, 파일이름 + '.txt')
            with open(text_path, 'w', encoding = 'utf-8') as f:
                f.write(data)
```

Crawling Bond Reports

pandas와 os의 listdir, isfile을 사용하여 txt파일들을 병합, csv 형식으로 변환

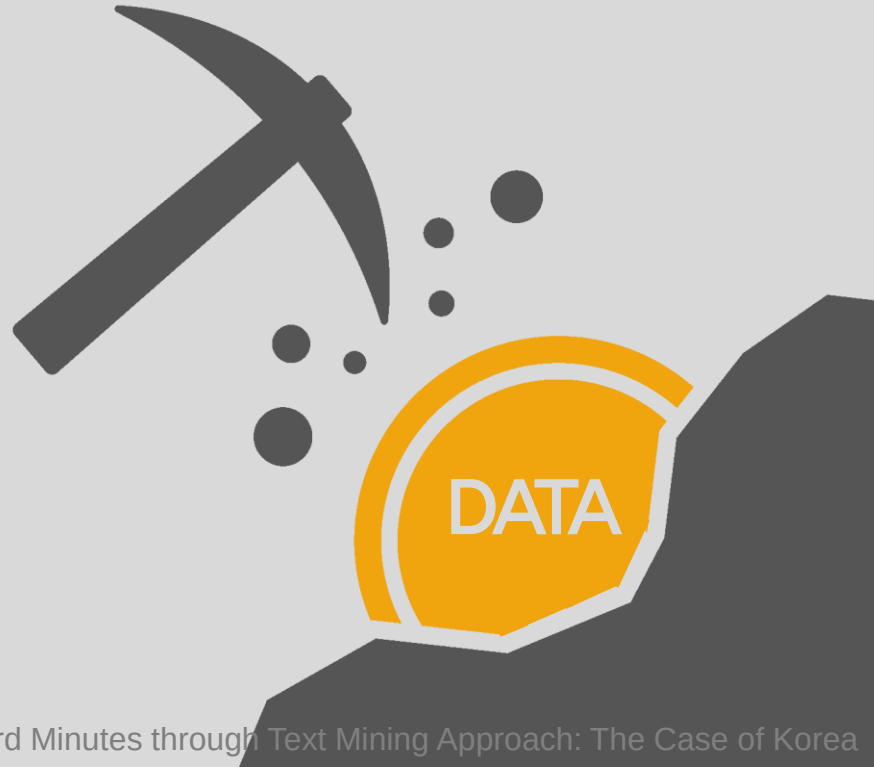
```
# 1
source_folder = "txt 파일이 들어있는 경로"
output_file = "내가 파일을 내보낼 경로₩내보낼 파일 이름.csv"

# 2
# 지정 폴더 내 파일 목록 조회
txt_files = [f for f in listdir(source_folder) if isfile(join(source_folder, f))]
```

Crawling Bond Reports

전체 수집 대상 채권 분석 보고서 수 3466개

PDF Parser 모듈을 사용하는
변환 과정에서 에러 발생, 96개 데이터 손실

채권 분석 보고서 수 3,370개



Crawling DoctorRock

Request, BeautifulSoup를 사용하여 금통위 의사록 크롤링
tika를 사용하여 pdf to txt 파일로 변환

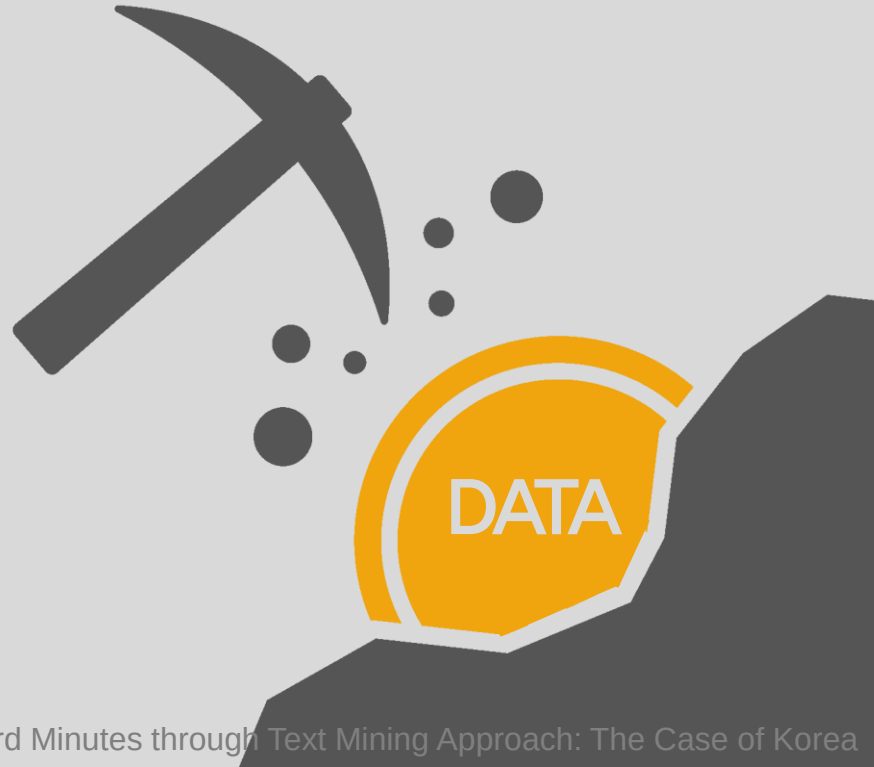
```
# 기존의 .가 아닌, ₩n을 space로 바꿔줌으로써 섹션 구분 성능 향상
parsedPDF = re.sub('₩n', ' ', parser.from_file(pdf_tmp_filepath)["content"])

# 문장구분 성능 향상
sentence_enders = re.compile
(r"((?<=[함음됨임봄짐움])(₩s{2,3}|₩.+|₩()|(?<=다)₩.)₩s*")
```

Crawling DoctorRock

전체 수집 대상 금통위 의사록 수 272 + 1개
실제로 수집된 금통위 의사록 수 273개

실제 논문 보다 10개 더 많은 PDF 수집

총 0개의 데이터 손실 발생



Crawling News

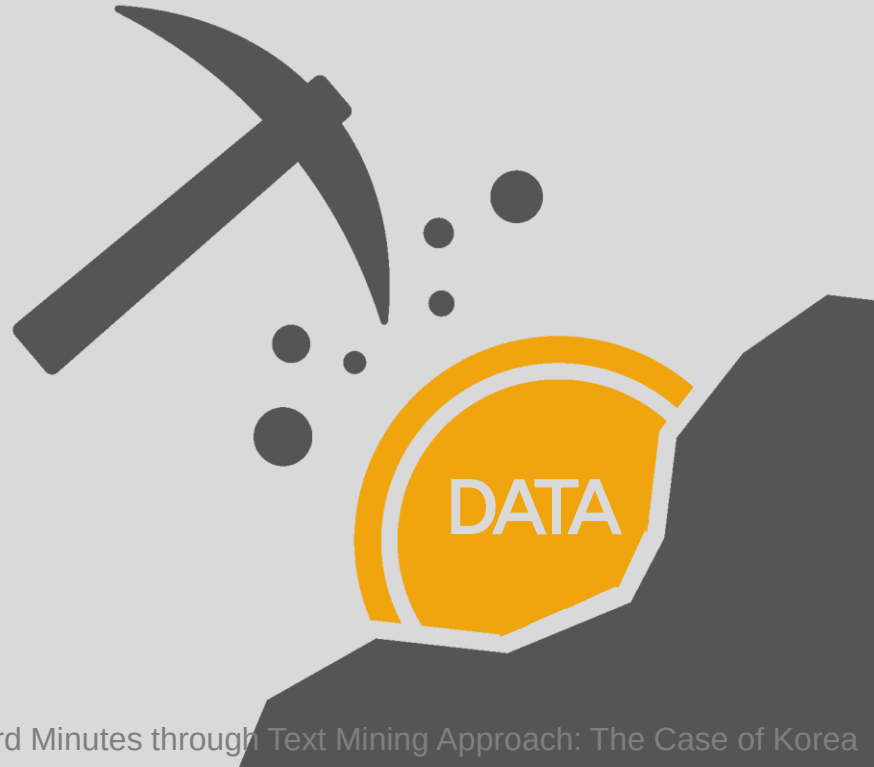
```
try:  # 네이버 뉴스를 먼저 크롤링 한다.
            item['date'] = response.css('span.t11::text').get().split(' ')[0]
            item['title'] = response.css('#articleTitle::text').get()
            item['content'] = response.css('#articleBodyContents::text').get()
            yield item

Except: # 그 다음으로 이데일리 사이트에 접속한다.
            item['date'] =
            response.css('div.dates').xpath('//p[contains(text(),"등록")][1]/text()').get().split(' ')[1]
            item['title'] = response.css('div.news_titles h2::text').get()
            item['content'] = response.css('div.news_body::text').get()
            yield item
```

Crawling News

edaily 키워드로 뽑은 뉴스: 4,577개
이데일리 키워드로 뽑은 뉴스: 81,030개
중복 제거 + 텍스트 변환 후: 84,259 개

연합 인포맥스 뉴스: 140,171개
텍스트 변환 후: 134,448 개

연합 뉴스: 24,929개
텍스트 변환 후: 24,929 개



Crawling
Sum_total

채권분석+뉴스3사+금통위의사록
3,370+243,636+3,594

250,600 개

Crawling Compare

	채권 분석 보고서	금통위 의사록	뉴스
논문	25,325	151	206,223
논문 구현	3,370	150	243,636

Data Preprocessing

데이터전처리
Data Preprocessing

뉴스의 경우, content 정제를 위해 정규표현식 사용

```
# 기자 이름 포함 앞 부분 다 지우기
news_ac['content'] =
news_ac['content'].apply(lambda x: re.sub('.+['
기자']{2}₩=|.+['기자']{2} ₩=|.+['특파원
']{2}₩=|.+['특파원']{2} ₩=','',x,2))

# (서울or뉴욕=연합인포맥스or연합뉴스) 지우기
news_ac['content'] =
news_ac['content'].apply(lambda x:
re.sub('₩([가-    ].=연합인포맥스₩)|₩([가-    ].=
연합뉴스₩)','',x,1))
```

데이터전처리
DataPreprocessing

txt 파일 병합

.csv 파일 변환

token, N-gram 추출

불용어 처리

날짜 형식 통합

콜금리 label 부착

최종 데이터 병합

뉴스의 경우, 데이터가 많아 Sampling 후 진행

```
df_sample =
df.sample(n=1000, random_state = 1)

for_sampling = df_sample.copy()
for_sampling.head(3)
```

형태소 조합
Token, N-gram

N-gram의
최대는 5

+

명사, 동사,
부사, 및 부정어

+

15회 이하
N-gram은 x

불용어 처리

Function & Regular Expression

```
stoppos = [
    'SC','SY','SF','SE','SS','SP','SO','SW',
    'SSC','JKS','JKC','JKG','JKO','JKB','JKV',
    'JKQ','JX','JC','EF','EC','ETN','ETM','JKS',
    'JKC','JKG','JKO','JKB','JKV','JKQ','JC','JX',
    'EP','EF','EC','ETN','ETM','XPN','XSN','XSV',
    'XSA','XR','SF','SE','SSO','SSC','SC','SY','SH',
    'SL','SN'
    ]
```

```
total_df['ngram'] =
total_df['ngram'].apply
(lambda x:re.sub('[가-    ]{1,8}₩/
(JKS|JKC|JKG|JKO|JKB|JKV|JKQ|J
C|JX|EP|EF|EC|ETN|ETM|XPN|XS
N|XSV|XSA|XR|SF|SE|SSO|SSC|S
C|SY|SH|SL|SN)
(₩,?)','',str(x),50))
```

250600개 중, 227개의 N-gram이 추출되지 않음

15회 미만 N-gram 삭제

Value_counts()

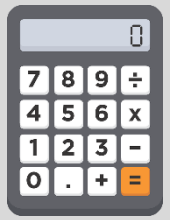
```
total_df = pd.DataFrame(merged_df['ngram'].value_counts())

up_df = pd.DataFrame(merged_df[temp_df.updown == 'up']['ngram'].value_counts())

down_df = pd.DataFrame(merged_df[temp_df.updown == 'down']['ngram'].value_counts())

ne_df = pd.DataFrame(temp_df[temp_df.updown ==  'neutrality']['ngram'].value_counts())
```

15회 미만 N-gram 삭제

Value_counts()

```
for_polarity_df = merged_df[merged_df['total'] >= 15]
# for_polarity_df = for_polarity_df.fillna(0.5)
```

추출된 총 N-gram수: 47,606개 (논문 73,428개)

Data Calculate 

극성 계산
Polarity Calculate

$$\frac{\text{하나의 N-gram의 Pos 개수 / 전체 Pos 개수}}{\text{하나의 N-gram의 Neg 개수 / 전체 Neg 개수}}$$

극성 계산
Polarity Calculate

```
sum_pos = df['up'].sum()
# 전체 pos 개수: 19897829 개

sum_neg = df['down'].sum()
# 전체 neg 개수: 1749713 개
```

```
df['분자'] = df['up'] / sum_pos
df['분모'] = df['down'] / sum_neg
df['polarity'] = df['분자'] / df['분모']
```

극성 계산
Polarity Calculate

```
# 0.76 이상 1.3 이하의 polarity 삭제
df = df[(df['polarity'] > 1.3) | (df['polarity'] < 0.76)]

# # 1.3보다 크면 1('hawkish' 매파), 0.76보다 작으면 -1('dovish' 비둘기파))
df['hawk/dov'][df['polarity'] > 1.3] = 1
df['hawk/dov'][df['polarity'] < 0.76] = -1
```

논문 구현 hawkish: 13,767개, dovish: 12,576개 추출
실제 논문 hawkish: 18,685개, dovish: 21,280개 추출

금통위 의사록 어조 추출
Doctor Rock Tone

```
# 닥터록의 한문장의 n-gram을 word_list로 설정
for idx, val in enumerate(Dr_rock['ngram']):
    word_list = val.replace(' ','').split(',')
    hawk_count = 0
    dov_count = 0

# n-gram이 hawk | dov 에 있으면 count += 1
for i in word_list:
        if i in hawk_list:
            hawk_count += 1
        elif i in dov_list:
            dov_count += 1
```

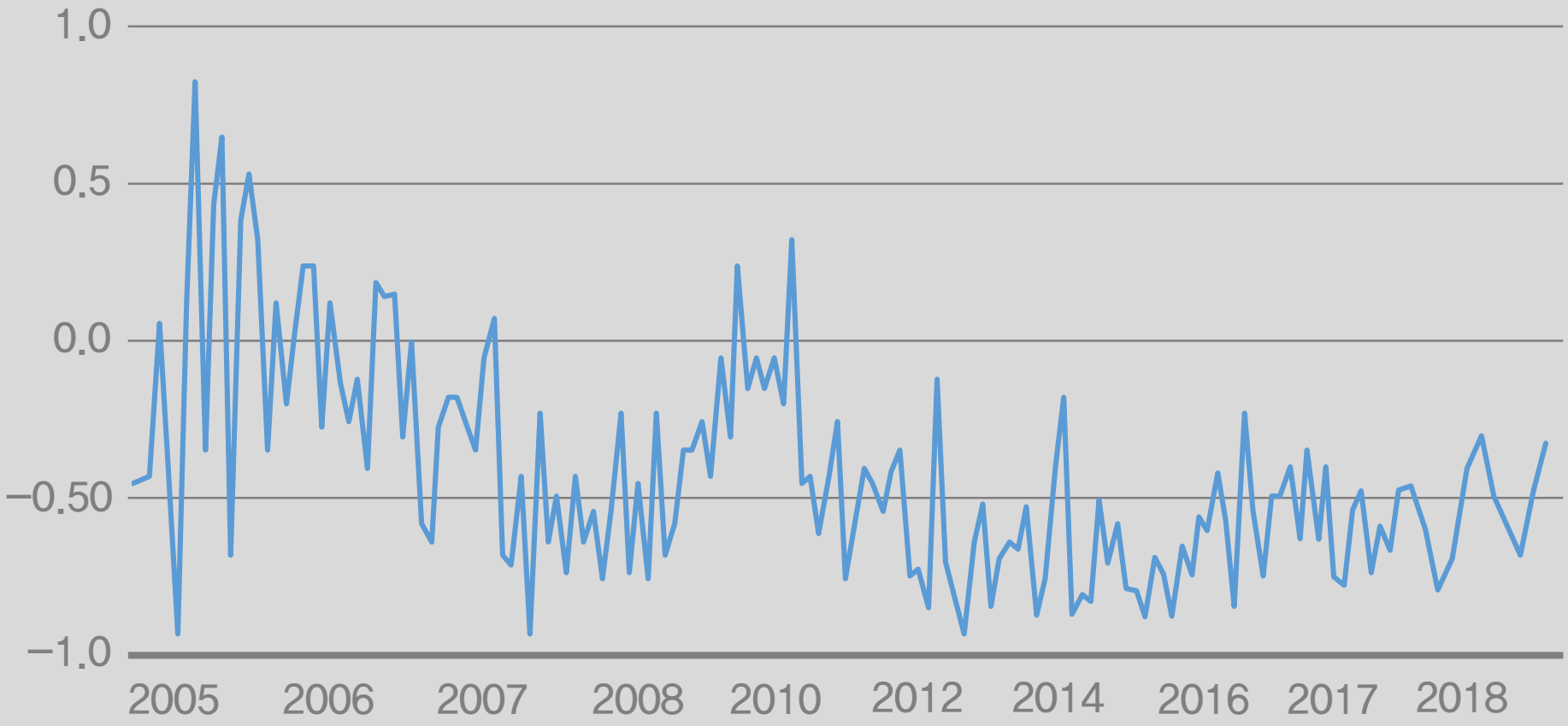
```
# 분자, 분모 설정
bunja = float(h_count - d_count)
bunmo = float(h_count + d_count)

# 문장 점수 계산
sentence_score = float(bunja / bunmo)
```

총 150개의 의사록 데이터 사용
논문 역시 151개의 의사록 데이터 사용

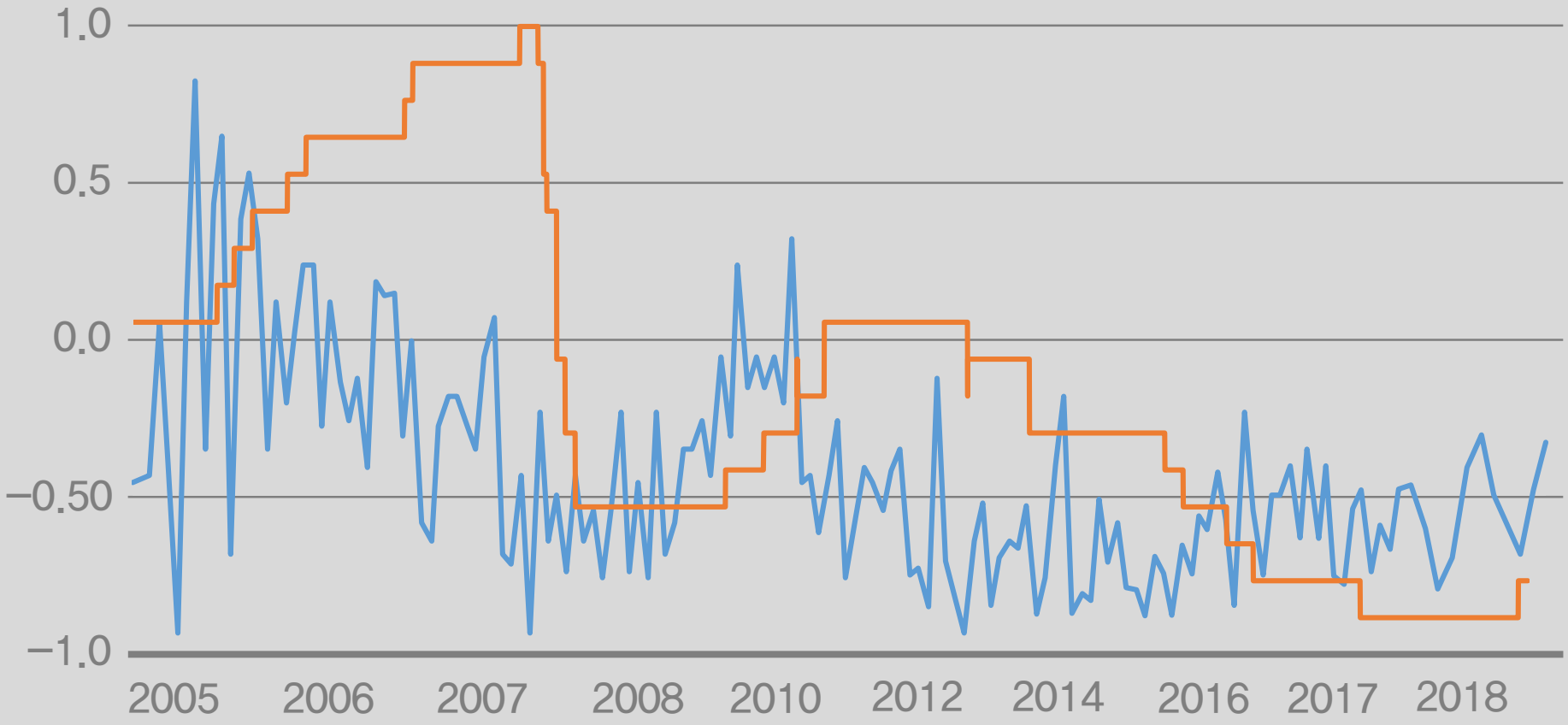
금통위 의사록 어조 추출
Doctor Rock Tone

```
# 문장들을 하나의 의사록으로 합치기
for i in date_list:
    one_Dr = dr_save[dr_save[  'date'  ] == i]
    pos_sentence = len(one_Dr[one_Dr['sentence_score'] > 0])
    neg_sentence = len(one_Dr[one_Dr['sentence_score'] < 0])

    bunja = pos_sentence - neg_sentence
    bunmo = pos_sentence + neg_sentence

    score = float(bunja / bunmo)
```

금통위 의사록 어조 그래프



금통위 의사록 / 기준금리


1.0
0.5
0.0
−0.50
−1.0
2005
2006
2007
2008
2010
2012
2014
2016
2017
2018

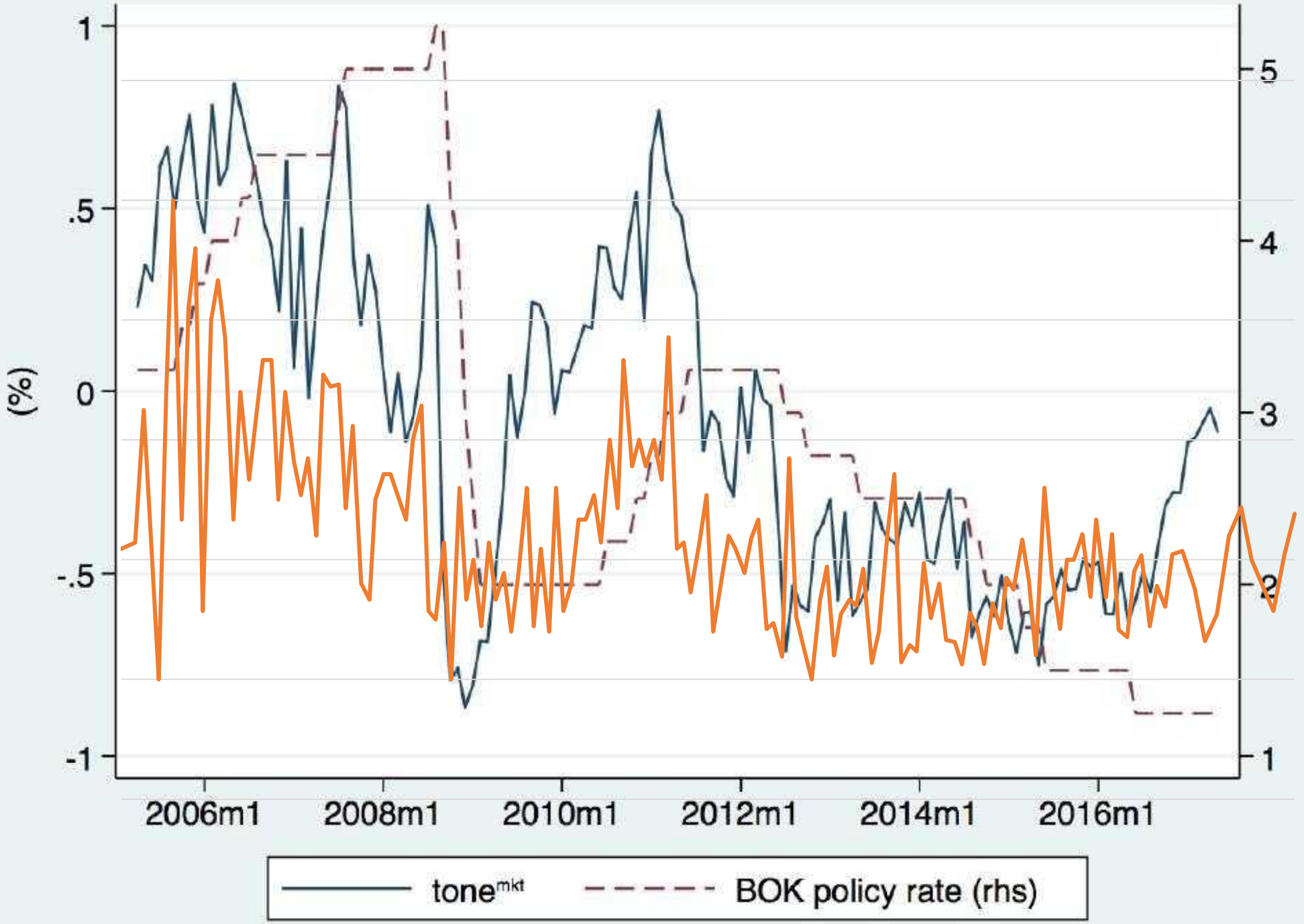
상관관계
Correlation

피어슨 상관 계수(Pearson correlation coefficient 또는 Pearson's r)는
두 변수간의 관련성을 구하기 위해 보편적으로 이용된다. 개념은 다음과 같다.
correlation = X와 Y가 함께 변하는 정도 / X와 Y가 각각 변하는 정도

```
for_corr.corr(method = 'pearson')
```

상관관계

0.549


1
.5
0
-.5
-1
(%)
5
4
3
2
1
2006m1
2008m1
2010m1
2012m1
2014m1
2016m1
tone^mkt
BOK policy rate (rhs)


왜
Why?
의사록 어조 그래프의 트렌드는 유사하다
2012년까지는 마치 y절편 값만 다른 듯 하다
그 이유를 찾아 거슬러 올라가 보자

논문 구현 hawkish: 13,767개, dovish: 12,576개 추출
실제 논문 hawkish: 18,685개, dovish: 21,280개 추출

추출된 총 N-gram수: 47,606 개 (논문 73,428개)

	채권 분석 보고서	금통위 의사록	뉴스
논문	25,325	151	206,223
논문 구현	3,370	150	243,636

결론

완벽하게 구현하지는 못했지만, 트렌드와 유사한 그래프를 확인하였고,
논문 보다 높은 의사록 어조와 기준 금리의 상관계수를 산출하였다.

결론

연구자들의 경우, 뉴스를 더 많이 수집하였다.
만약, 뉴스가 채권분석 보고서보다 조금 더 극성을 가질 것이다,
라고 추측해본다면, 극성의 범위를 조절함으로써
더 나은 결과를 도출할 수 있을 것이다.

연구자들의 Thinking

또한, 실제로 데이터들을 확인해본 결과,
해가 갈수록 기준금리의 그래프는 평탄해지는 경향이 존재하는 것을 확인하였다.
따라서, 금통위에서 발표하는 의사록과 연관성이 있다고 파악되는
다른 변수와의 관계를 살펴보는 것 역시 유의미 할 것이라고 연구자들은 생각하였다.


"발표 들어주셔서 감사합니다."
-마동석-