

# Issue What Show (잇슈왓슈)

한눈에 보는 대한민국 핫이슈

라떼는 말 이야  
Latte is horse

박종선(조장), 김재현, 이희수, 문진한이.

# 목차

---

## □ 필요성

1. 포털사이트 인기 검색어의 아쉬운 점
2. 다른 서비스와의 차별성

## □ 프로젝트 과정

1. Web Scraping
2. 전처리
3. 형태소 분석
4. 통계적 분석
5. 시각화
6. 시스템 구성
7. 완성모델 시연

## □ 추후 개발예정 사항

1. 사이트별 키워드로 뉴스 보기
2. 특정 키워드의 시각별 중요도 보기
3. 월별, 일별, 계절 등으로 변화 보여주기
4. 다양한 기기 환경에서 최적화된 서비스 구현하기

## □ 만든 사람들

## □ 사용기술 및 참고문헌

## □ 마지막.. 느낀점 & Comment..

## □ Q n A

---

## 필요성

- 1. 포털사이트 인기 검색어의 아쉬운 점**
  - 2. 다른 서비스와의 차별성**
-

# 1. 포털사이트 인기 검색어의 아쉬운 점

The screenshot shows the NAVER search history interface. At the top, there are date and time controls: '2019.10.29. (화)' and '09:25'. Below this is a '전체 연령대' (All Age Groups) section listing the top 10 search terms:

1	에어팟 프로
2	오퀴즈 아이패드 매일지급
3	임병수
4	신세경집업
5	vip 드라마
6	249달러
7	함연지
8	vip
9	타다
10	캐리소프트

Below this is a 'NAVER' logo, followed by a list of the top 20 search terms:

11	김철민
12	황사
13	펜벤다졸
14	부의
15	박시온 진태현
16	꽃길만 걸어요
17	박시온
18	2020 스타벅스 다이어리
19	볼리비아
20	에어팟 3세대

Annotations on the right side highlight three aspects of the search results:

- 같은 맥락의 검색어가 여러 개 검색되어 있다.
- 검색어간의 연관성을 알 수 없다.
- 검색어만 봤을 때 그 의미를 한번에 알 수 없다.

## 2. 다른 서비스와의 차별성

### SOME TREND

: 검색 키워드에서 연관된 키워드를 맵 형식으로 보여주는 서비스



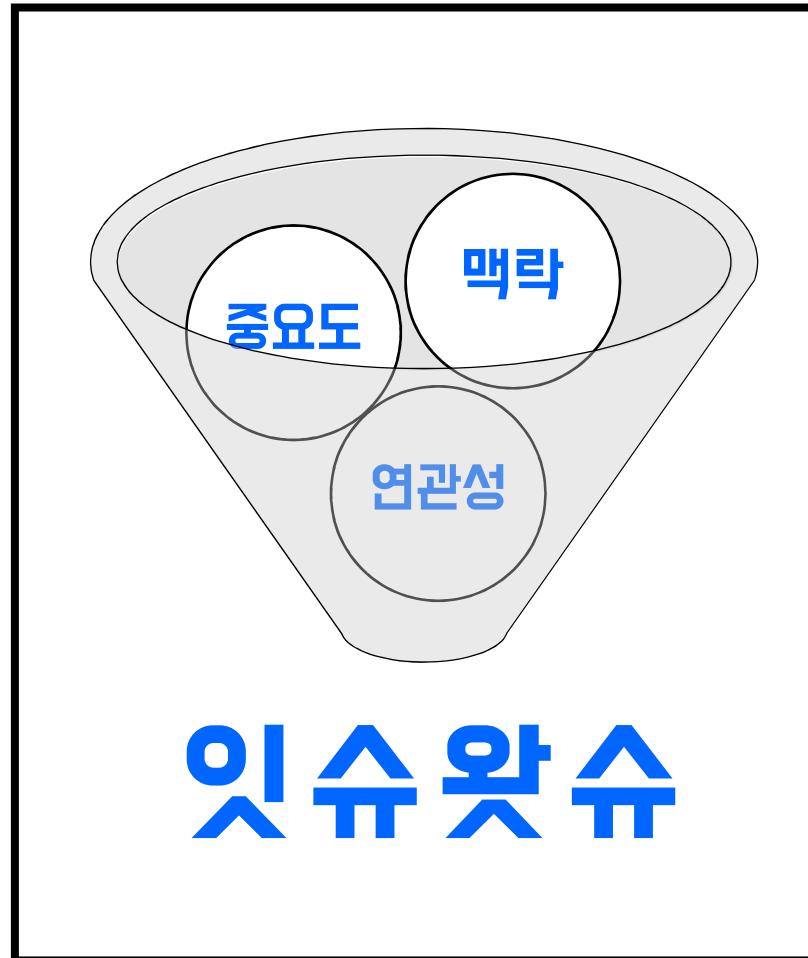
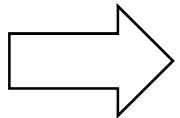
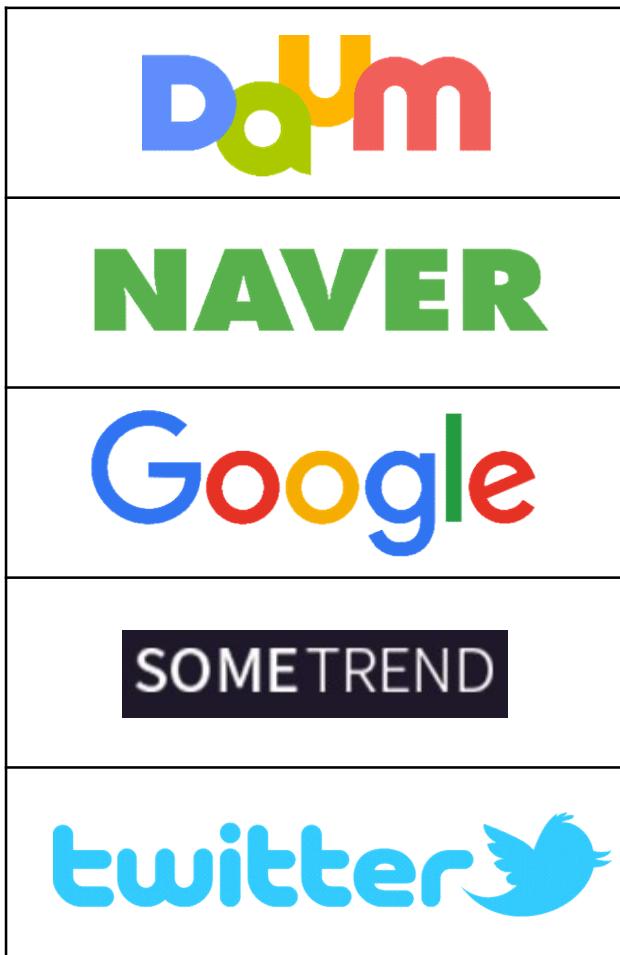
#### 1. 검색 키워드와

연관 키워드간의 관계에서,  
연관 키워드들의 중요도나  
연관 키워드간의 관계를  
알 수 없다.

#### 2. 모든 뉴스들에서

클릭 수가 많은 뉴스기사만  
보여주기 때문에,  
키워드와 관련된 뉴스는  
바로 볼 수 없다.

# 필요성 = 잇슈왓슈



## ▣ 프로젝트 과정

### 1. Web Scraping

- 검색어 가져오기
- 가져온 검색어로 뉴스기사 수집
- 뉴스기사 html 저장

### 2. 전처리

- White List & Black List 방식으로 1차 추출
- 1차 처리 후 불필요한 Text를 제거

### 3. 형태소 분석

- 분석기 모델 분석
- 분석기 모델 선정 및 이유 : Khaiii

### 4. 통계적 분석

- 통계적 분석방법 선정

### - 분석 기법 소개 및 선정 :

- TF - IDF, Word2Vec

### - 분석 기법 적용 :

- TF - IDF, Word2Vec
- TF - IDF + Word2Vec

### 5. 시각화

- 시각화 모델 분석 : 제외한 모델
- 시각화 모델 선정 및 적용 : 키워드 네트워크

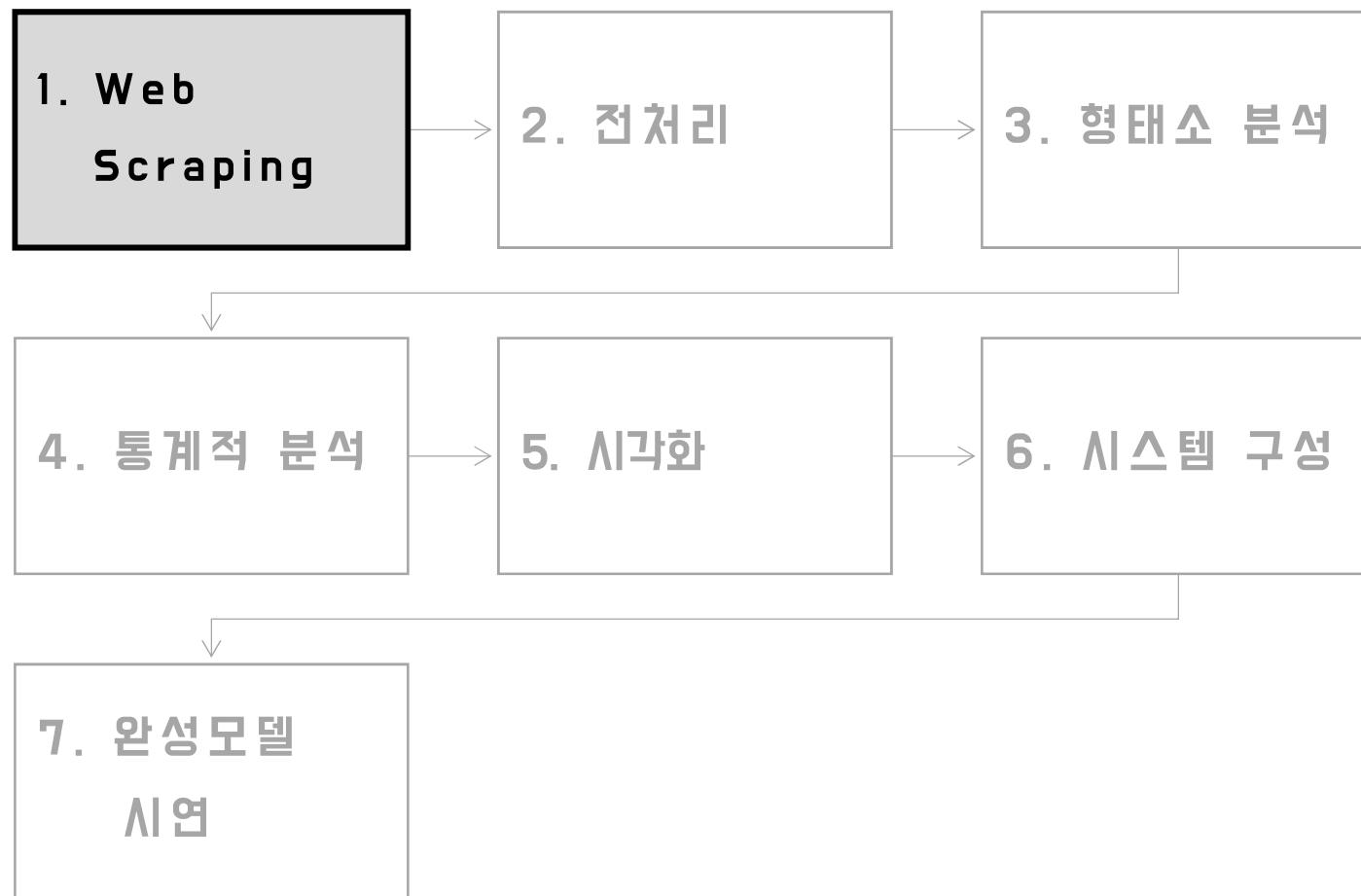
### 6. 시스템 구성

- 시스템 구성 및 전체 프로세스 모식도
- AWS 개발 및 배포환경 설정 / 도메인 네임 설정

### 7. 완성모델 시연

# 프로젝트 과정

---



# 1. Web Scraping



## - 검색어 가져오기

The image shows screenshots from four different platforms:

- Daum**: A Korean search engine showing the top 10 trending keywords for the day.
- NAVER**: A Korean search engine showing the top 10 trending keywords for the day.
- Twitter**: A social media platform showing the top 5 trending hashtags.
- Google**: A search engine showing a message about regional search results.

A large red arrow labeled "Reject" points from the Twitter and Google sections towards the right side of the slide.

Daum	NAVER
1. 임원희	1~10위
2. 정동진	11~20위
3. 전인화	1. 김건모 아버지
4. 정석용	2. 일루미나티
5. 집사부일체	3. 바디피셜 머슬건
6. 이선미 여사	4. 김건모 부친상
7. 권혁수	5. 김건모 엄마
8. 룰드컵	6. 정동진
9. 셔누	7. 세수광미니쿠션
10. 김희철	8. 권혁수
	9. 임원희
	10. 토니엄마 식당

1 · Trending  
#오타쿠\_취향\_호불호  
1,731 Tweets

2 · Trending  
#이\_해시태그\_보는순간\_배경화면\_공개

3 · Trending  
스위치 7만원

4 · Trending  
기류쿠로

5 · Trending  
흐르다네

급상승 검색어

1~10위 11~20위

1. 김건모 아버지  
2. 일루미나티  
3. 바디피셜 머슬건  
4. 김건모 부친상  
5. 김건모 엄마  
6. 정동진  
7. 세수광미니쿠션  
8. 권혁수  
9. 임원희  
10. 토니엄마 식당

2019.11.03. 22:04 기준 DataLab. 급상승

트위터의 경우  
의미가 없는 키워드가 많아 제외하기로 하였음.

구글의 경우 국내는  
실시간 키워드 서비스를 제공하지 않아  
제외하기로 하였음.

네이버, 다음  
검색어 각 10개  
총 20개의 키워드를 수집

트위터의 경우  
의미가 없는 키워드가 많아 제외하기로 하였음.

구글의 경우 국내는  
실시간 키워드 서비스를 제공하지 않아  
제외하기로 하였음.

# 1. Web Scraping



- 수집한 검색어로 뉴스 검색



총 20개의 키워드에서  
각 키워드별로  
네이버, 다음 뉴스 30개와  
트윗 100개를 가져옴

구글의 경우  
구글 자체 알고리즘에 의해,  
**최신순이 아닌 정확도순으로**  
뉴스가 업로드되어  
트랜드와 맞지 않아 제외하기로 하였음.

**20개의 키워드로**  
**약 1200개의**  
**뉴스기사를**  
**가져옴**

현재는 한 사이클이  
2시간마다 실행하도록  
설정하였음.

(지금 이 순간도 홀로 열심히 일하고 있습니다..)

# 1. Web Scraping



## - 뉴스기사 html 저장

```
<어딘가>
| - <YYMMDD>
|   | - <HHmm>           # 이 단위가 한 사이클
|   |   | - <D_K_01>
|   |   |   | - <D_01>.html
|   |   |   | - ...
|   |   |   | - <D_30>.html
|   |   |   ...
|   |   |   | - <G_30>.html
|   |   |   | - <N_01>.html
|   |   |   ...
|   |   |   | - <T_30>.html #총 120개 뉴스
|   |   ...
|   |   | - <D_K_10>
|   |   |   | - ...
|   |   | - <N_K_01>
|   |   |   | - <D_01>.html
|   |   |   | - ...
|   |   ...
|   |   | - <D_K_01>
|   |   |   | - <D_01>.html
|   |   ...
|   |   ...
|   |   | - <T_K_10>          #총 40개 키워드
|   |   |   | - <D_01>.html
|   ...
|   ...
|   | - <HHmm>           # 다음 사이클에 생성될 단위
|   |   | - <D_K_01>
|   ...
|   ...
```

```
-rw-rw-r--. 1 lab03 lab03 5788 10월 17 09:31 log_prepro  
-rw-rw-r--. 1 lab03 lab03 196303 10월 17 09:19 output.csv  
-rw-rw-r--. 1 lab03 lab03 283777 10월 17 09:19 output_tw.  
[lab03@ip-70-12-50-156 0910]$ ls  
D_K_01  D_K_05  D_K_09  N_K_03  N_K_07  log_harvesta.txt  
D_K_02  D_K_06  D_K_10  N_K_04  N_K_08  log_preprocra.txt  
D_K_03  D_K_07  N_K_01  N_K_05  N_K_09  output.csv  
D_K_04  D_K_08  N_K_02  N_K_06  N_K_10  output_tw.csv  
[lab03@ip-70-12-50-156 0910]$ tail log_harvesta.txt  
수집한 문서 개수: 1183개  
걸린 시간: 10m 27s  
[lab03@ip-70-12-50-156 0910]$
```

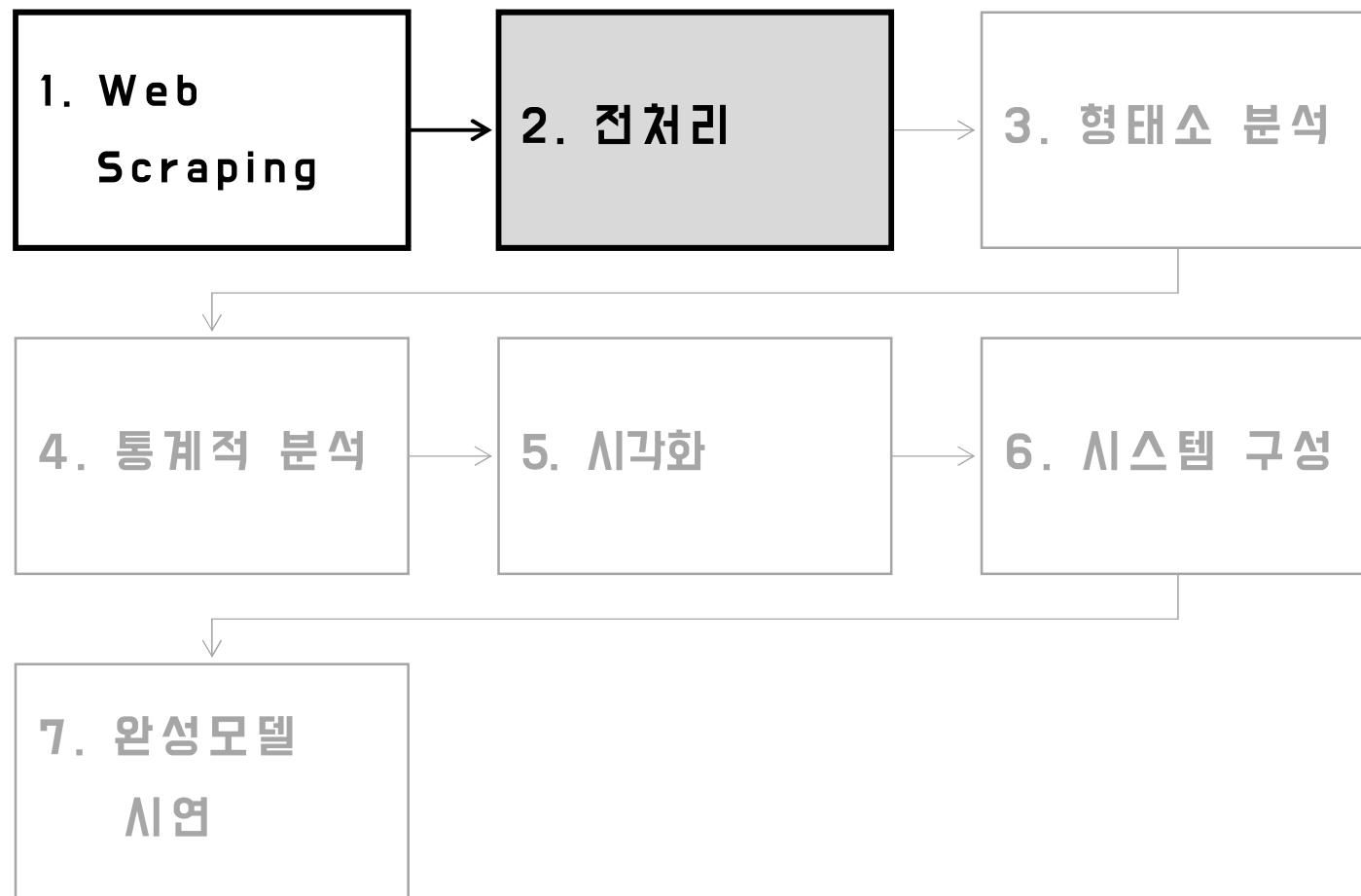
html 문서 저장시.  
어느 사이트의 키워드인지.  
어느 사이트의 뉴스인지 알 수 있도록 하였음.

이는 추후 다른 기능구현시

사용할 수 있는 메타 정보가 될 수 있음.

# 프로젝트 과정

---





## 2. 전처리

### - White List & Black List 방식으로 1차 추출



#### White List 방식

html 형식에 따라 정보추출처리를 할 수 있는 곳들을 선정하여 수집하는 것을  
White List라 정의하였음.

#### Black List 방식

html 형식에 따라 정보추출처리를 할 수 없는 곳들의 예러로그를 분석하였으며,  
그 후 처리가 불가능하다고 판단된 곳을, 사전에 제외하는 것을 Black List라 정의하였음.

## 2. 전처리



### - 1차 처리 후 불필요한 Text를 제거

```
# 화이트리스트 체크
#
# 화이트리스트 셀렉터 문자열들
_selectorList = [
    # DAUM
    "div#harmonyContainer.article_view",
    # NAVER
    "div#articleBody.article_body",
    # NAVER MAIN
    "div#wrap table td.content div#main_content \
        div#articleBody.article_body div#articleBodyContents",
    # MAGIC(아주 많은 뉴스들이 여기에 포함 됨)
    "*[itemprop='articleBody']",
    # TOMATO 뉴스
    "article div.rn_scontent section div.rns_text",
    # ...
]
```

### White List

1

```
[lab03@ip-70-12-115-184 D_K_01]$ cat D_01_DONE.txt
2019 코리아세일페스타 오늘 개막
2019 코리아세일페스타 개막을 하루 앞둔 31일 오전 서울 중구 명동예술극장 앞에서
설문모 산업통상자원부 장관(왼쪽 다섯번째 부단), 김연화 주진위원장, 흥보모델 방송
인 강호동 등이 대형 모양의 플래카드를 들고 행사를 홍보하고 있다. 코리아세일페스
타는 11월 1~22일 전국적으로 열려 유통 제조 서비스 업체가 업체별로 특색 있는 할인
행사를 추진할 예정이다.[lab03@ip-70-12-115-184 D_K_01]$
```

Useless : 57 (4.8%)  
Unknown : 106 (9.0%)  
Hit : 1015 (86.2%)  
걸린 시간: 2m 32s  
[lab03@ip-70-12-50-156 0910]\$

필요한  
Text 정보

만 추출

처리율 :  
최소  
85% 이상  
구현

4

```
skip = False
for s in soup.select("".join([
    # 미디어펜
    "div#HeadMenu div#Default_Warp div#MenuBar ul#mega-menu, ",
    # 더코리아뉴스
    "div#wrap div div#divMenu div table td \
        div[style='z-index:0'], ",
    # YTN 94.5
    "div#wrap div#gnb ul#topLink li.b ul li.YTN_CSA_outlink1, ",
    # WIKI TREE
    "div.wrap div.multi-scroll-wrap div.multi-scroll-inner \
        div.scroll-zone01 div.scroll-start01 div.scroll-i \
        div.article_wrap div.article_byline span.time, "
])):
```

### Black List

2

```
# 모든 과정이 끝난 후 text 형태로 된 문서에서 불필요한 Text 제거
def common_rm_text(text):
    text = text.strip().split("\n")

    newText = []
    for li in text:
        li = li.strip()
        if li == "": continue

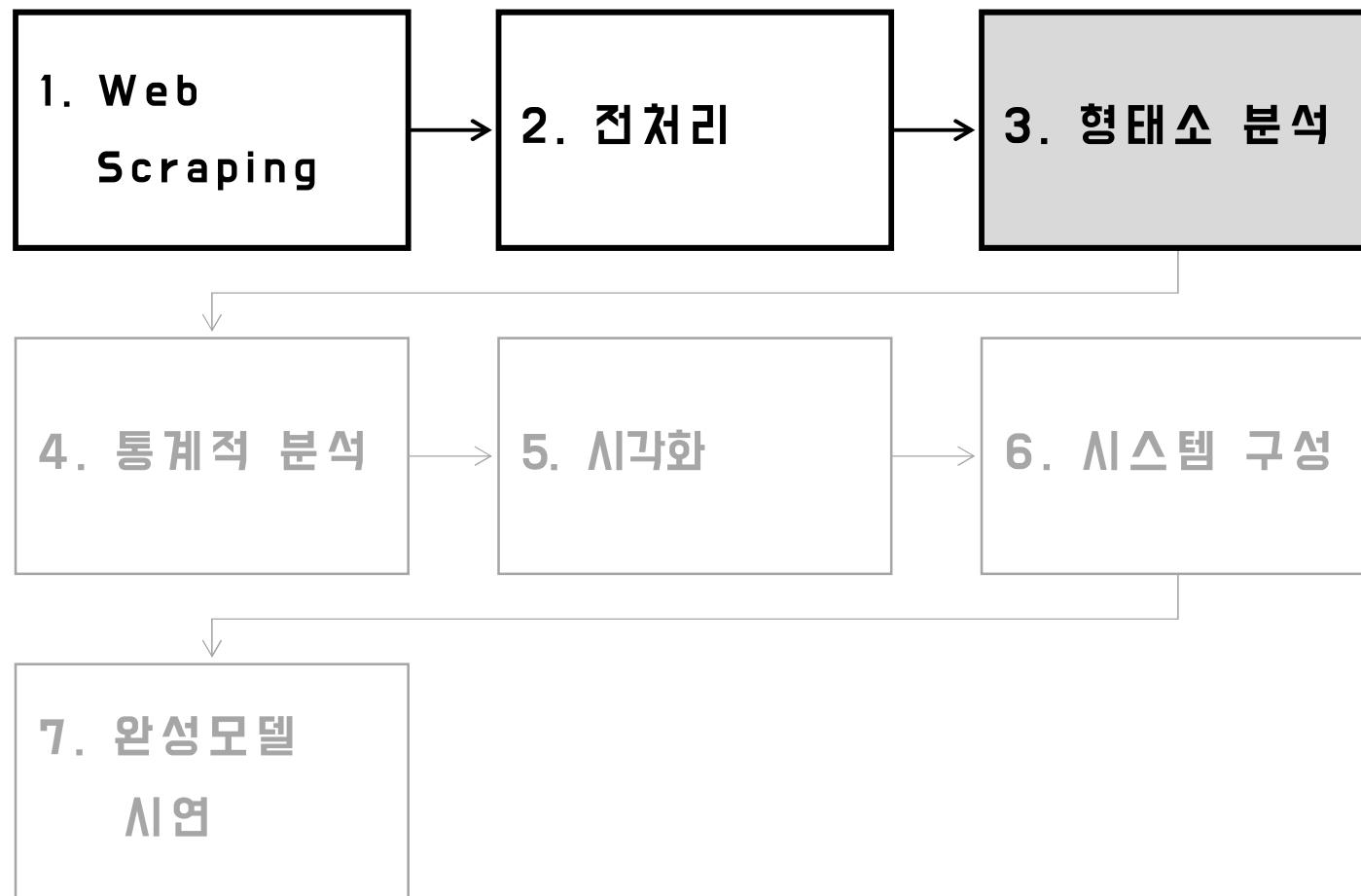
        # 뉴스 페第一部
        li = re.sub(r"^(뉴스엔\s+.*[a-zA-Z0-9_-+]+@[a-zA-Z0-9_-+]+\s+)", "", li)
        li = re.sub(r"\s*(\[\|\].{0,30}\|\]\)\s*", "", li)
        li = re.sub(r"\s*\(\s?.{0,10}\s*\^\s?\)\s*", "", li)
        li = re.sub(r"\s*\[\(\[\<\|.{0,10}-.{0,10}\[\|\]\]>\]\)\s*", "", li)

        # 이메일
        li = re.sub(
            r"^\^([a-zA-Z0-9_-+]+@[a-zA-Z0-9_-+]+\.[a-zA-Z0-9_-+]+)$", "", li)
        li = re.sub(
            r"^\^([a-zA-Z0-9_-+]+@[a-zA-Z0-9_-+]+\.[a-zA-Z0-9_-+]+)$", "", li)
        li = re.sub("", ".join([r"\^\^[\?\\?]{2}\^([a-zA-Z0-9_-+]+@[a-zA-Z0-9_-+]+\.[a-zA-Z0-9_-+]+[\?\\?])"])", "", li)
```

3

# 프로젝트 과정

---



# 3. 형태소 분석

## - 분석기 모델 분석

1. Khaiii(kakao hangul analyzer 3)
2. Open Korea Text(트위터)
3. 은전한닢
4. Komoran
5. Daon
6. ETRI(한국전자통신연구원) 분석기
7. RHINO
8. 한나눔
9. 아리랑
10. 꼬마

같은 예문이라 하여도

분석기마다

분석 결과에 차이가 있다.

여자/NNG+화장실/NNG+서/JKB 일면식/NNG+도/JX 없/VA+는/ETM  
남성/NNG '/SP+문/VV+지/EC+말/VX+아/EC 폭행/NNG+'/SP 뛰쳐나가/VV+L/ETM  
여성/NNG 따라가/VV+아/EC 지속/NNG+하/XSV+어서/EC 폭행/NNG 피해자/NNG  
뇌진탕/NNG 증세/NNG 호소/NNG+/SP 트라우마/NNG+로/JKB 화장실/NNG 못/MAG 가/VV+아/EC  
자료사진/NNG+./SP+사진/NNG+은/JX 기사/NNG 중/NNB 특정표현/NNG+과/JKB 관계없/VA+ㅁ/ETN+./SF

여자 화장실서 일면식도 없는 남성 · 묻지 마 폭행 :

[# 0] = /SP(SP)  
[# 1] 여자 = 여자/NNG(NNG)  
[# 2] 화장실서 = 화장/NNG(NNG)실/NNG(NNG)서/JKB(JKB)  
[# 3] 일면식도 = 일/NNG(NNG)면식/NNG(NNG)도/JX(JX)  
[# 4] 없는 = 없/VA(VA)는/ETM(ETM)  
[# 5] 남성 = 남성/NNG(NNG)  
[# 6] ' = '/SW(SY)  
[# 7] 묻지 = 묻/VV(VV)지/EC(EC)  
[# 8] 마 = 말/VX(VX)아/EC(EC)  
[# 9] 폭행 = 폭행/NNG(NNG)  
[#10] ' = '/SW(SY)

뛰쳐나간 여성 따라가 지속해서 폭행

[# 0] 뛰쳐나간 = 뛰쳐나가/VV(VV) \_/ETM(ETM)  
[# 1] 여성 = 여성/NNG(NNG)  
[# 2] 따라가 = 따라가/VV(VV)아/EC(EC)  
[# 3] 지속해서 = 지속/NNG(NNG)하/XSV(XSV)아서/EC(EC)  
[# 4] 폭행 = 폭행/NNG(NNG)

### 3. 형태소 분석



#### - 분석기 모델 설정 및 이유 : Khaiii

##### 복합명사 추출 가능

(통계적 분석의 오차를 줄 일 수 있다.)

Ex) 여자화장실/NNG + 서/JKB

etc) 여자/NNG + 화장실/NNG + 서/JKB

##### 고유명사 구별 가능

Ex) 한승곤/NNP + 기자/NNG

etc) 한/MM + 승곤/NNP + 기자/NNG

어떤 분석기는 "한승곤 기자"의 성분인 "한"을  
"1" 혹은 관형사로 인식하는 오류가  
발생하기도 했다.

※ 여기서 잠깐! Khaiii(카이) 무엇?

Khaiii 란?

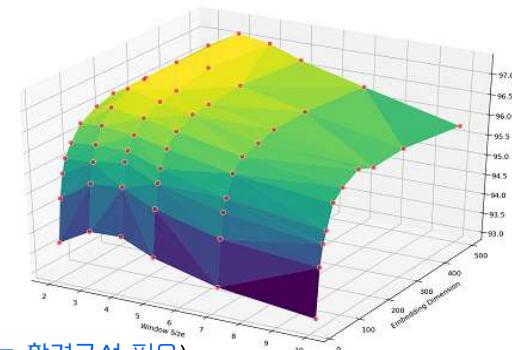
카이는 “[Kakao Hangul Analyzer III](#)”의 첫 글자들만 모아 만든 이름으로  
카카오에서 개발한 세 번째 형태소 분석기.

CNN(Convolutional Neural Network)기술을 적용해 학습한 형태소분석기로,

C++로 구현하여 GPU 없이도 비교적 빠르게 동작하며, Python 바인딩을 제공.

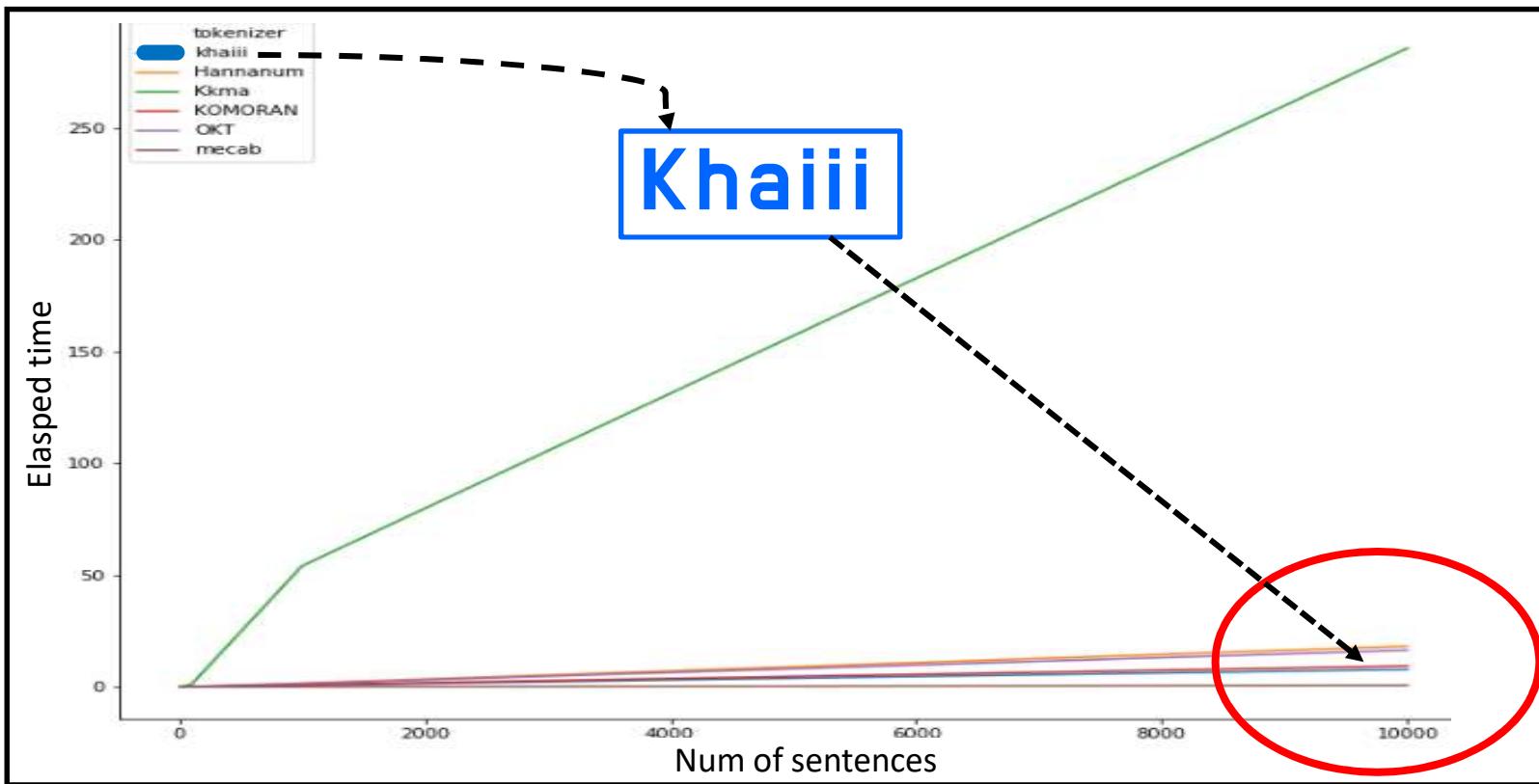
Khaiii 빌드 및 실행 환경

- [Linux only](#)
- gcc-5.3.1 (devtoolset-4)  
(최소 gcc-4.9.x 이상 필요)
- CMake 3.10 이상
- Python 3.x 이상  
([Linux의 2.x 버전과 충돌 없는 환경구성 필요](#))



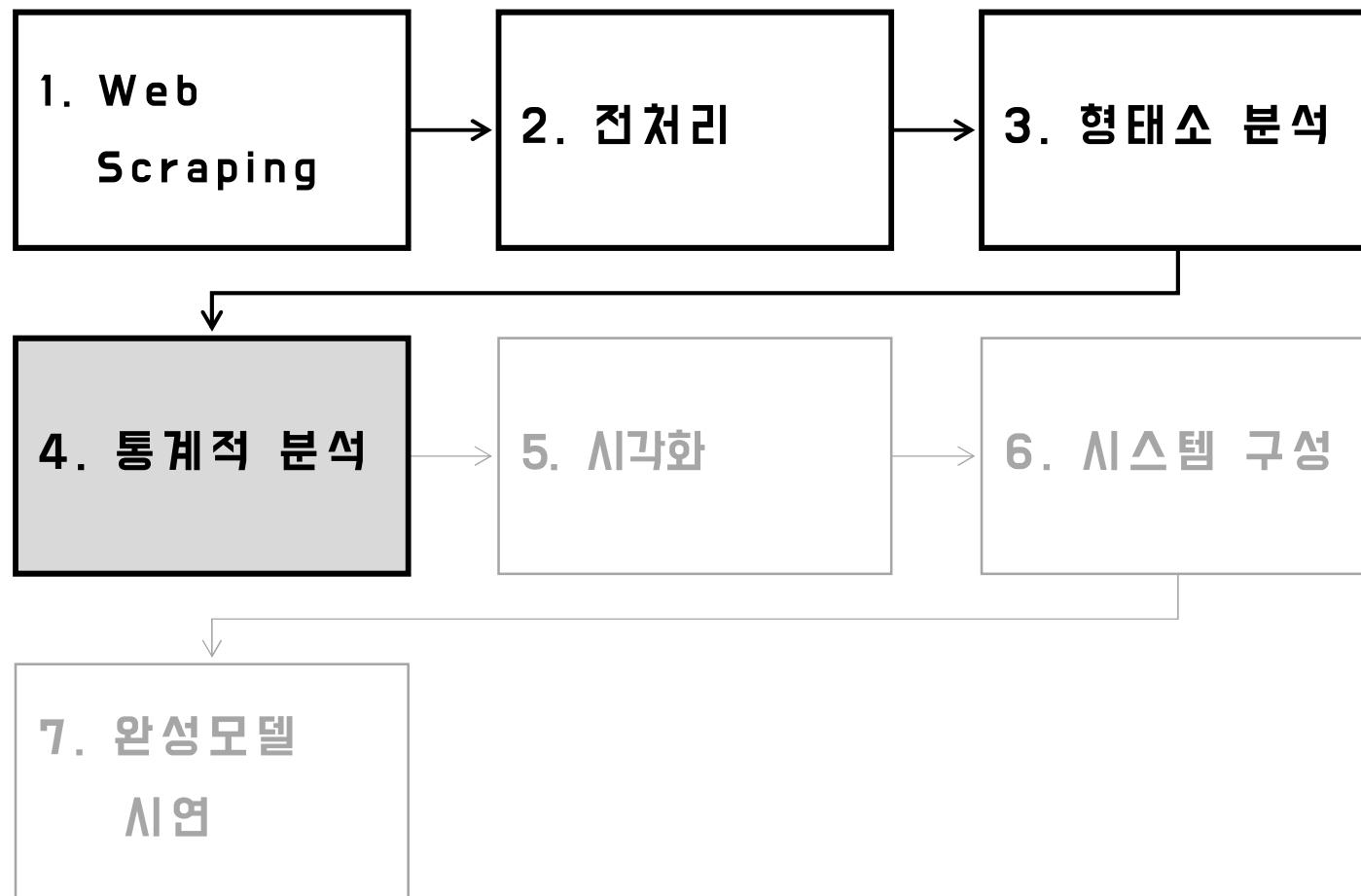
### 3. 형태소 분석

- 분석기 모델 설정 및 이유 : 분석기들 처리 속도 비교 그래프



# 프로젝트 과정

---



# 4. 통계적 분석

## - 통계적 분석방법 선정

<p>Word2Vec을 활용한 문서의 차이기반 분석</p> <p>지도 및 우수교수</p> <p>이 논문을 작성 학위논문으로 제출함</p> <p>2018년 3월 20일</p> <p>한국대학교 교육원 정보인증부처 장 희</p>	<p>Word2Vec을 이용한 한국어 단어 군집화 기법 Korean Language Clustering using Word2Vec method</p> <p>2018년 3월 20일</p> <p>Author: Author and his supervisor: Professor Dr. Hyeon-Jeong Kang, Korea University Abstract: Recent years have witnessed a remarkable increase in various machine learning applications in many fields such as computer vision, natural language processing, and robotics. Among them, deep learning has shown great promise in solving complex problems. In particular, word embeddings have been widely used in various NLP tasks. In this paper, we propose a clustering method based on word embeddings for Korean words. Our proposed method is able to cluster words into groups that share similar semantic meaning. We also evaluate our proposed method by comparing it with other clustering methods. The experimental results show that our proposed method is effective and efficient.</p> <p>Keywords: Machine Learning, Deep Learning, Natural Language Processing, Word Embedding, Clustering, Korean Language, Semantic Meaning</p>
<p>백트 푸시 기반 연관 벌행 감색 방법에 관한 연구</p> <p>지도 및 우수교수</p> <p>이 논문을 작성 학위논문으로 제출함</p> <p>2018년 12월 20일</p> <p>한국대학교 교육원 정보인증부처 이 은나</p>	<p>JDCS</p> <p>한국어 단어 벌행을 위한 Word2Vec 모델화 확장 Optimization of Word2Vec Models for Korean Word Embeddings Author: Author and his supervisor: Professor Dr. Hyeon-Jeong Kang, Korea University Abstract: In this paper, we propose a new optimization method for Word2Vec models for Korean words. The proposed method is based on the backpropagation algorithm, which is a well-known optimization method for neural networks. The proposed method is able to find the optimal parameters of the Word2Vec model by iteratively updating the weights of the model. The experimental results show that the proposed method is effective and efficient.</p> <p>Keywords: Machine Learning, Deep Learning, Natural Language Processing, Word Embedding, Clustering, Korean Language, Semantic Meaning</p>

통계적 분석에 관한 여러 논문들을 참고하여  
공통적으로 사용한 분석 방법을 검토하였음.  
검토결과,

- 연관도 분석을 위해 **Word2Vec** 과

- 중요도 분석을 위해 **TF-IDF** 를

채택하였음.

## 4. 통계적 분석

### - 분석 기법 소개 및 선정 : TF - IDF

TF-IDF는 단어가 문서군 내에서 얼마나 중요한지 반영하기 위한 수치 통계. 오랫동안 정보수집 및 텍스트 마이닝 분야에서 사용되어져 왔음. 대표적인 예로, Google의 공동 창업자인 레리 페이지와 세르게이 브린이 만든 PageRank 알고리즘 (Hypertext)와 TF-IDF를 같이 사용할 시. 웹 문서 키워드의 중요도 측정에 독보적인 성능을 보임.



The Anatomy of a Large-Scale Hypertextual Web Search Engine

Sergey Brin and Lawrence Page

(sergey, page)@cs.stanford.edu

Computer Science Department, Stanford University, Stanford, CA 94305

$$w_{i,j} = tf_{i,j} \times \log \left( \frac{N}{df_i} \right)$$

N : 총 문서의 수

TF(term frequency) : 단어 빈도

DF(document frequency) : 문서 빈도

IDF(inverse document frequency) : 문서 빈도의 역수

# 4. 통계적 분석

## - 분석 기법 소개 및 설정 : TF - IDF

### 제인 오스汀의 저서들을 이용한 예제

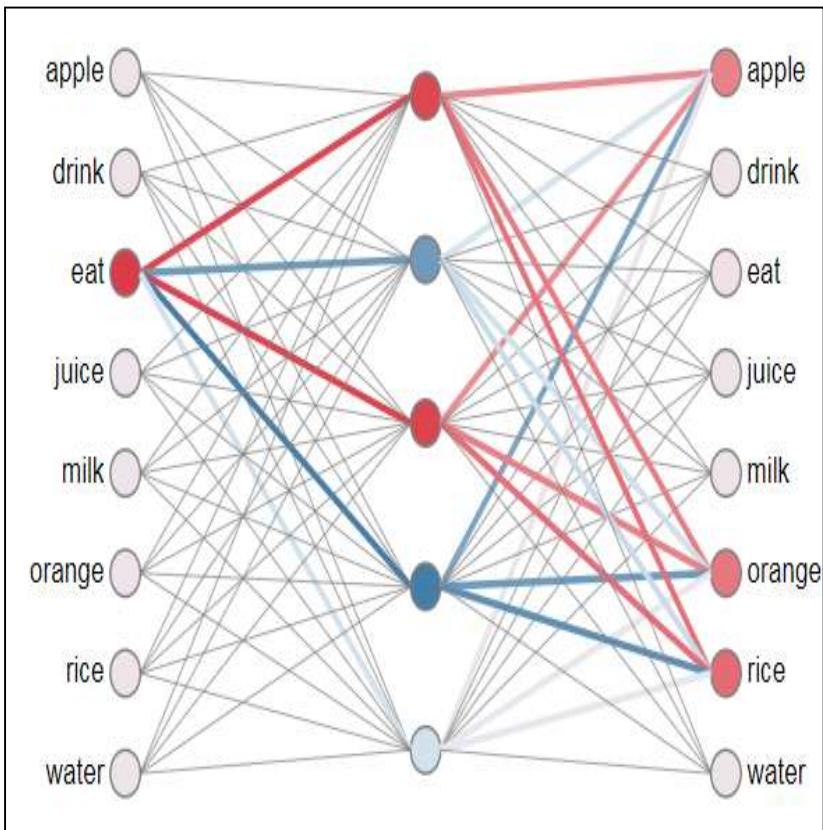
book	word	n	total	rank	term frequency`
		<chr>	<int>	<int>	<dbl>
1 Mansfield Park	the	6206	160460	1	0.0387
2 Mansfield Park	to	5475	160460	2	0.0341
3 Mansfield Park	and	5438	160460	3	0.0339
4 Emma	to	5239	160996	1	0.0325
5 Emma	the	5201	160996	2	0.0323
6 Emma	and	4896	160996	3	0.0304
7 Mansfield Park	of	4778	160460	4	0.0298
8 Pride & Prejudice	the	4331	122204	1	0.0354
9 Emma	of	4291	160996	4	0.0267
10 Pride & Prejudice	to	4162	122204	2	0.0341

book	word	n	tf	idf	tf_idf
		<int>	<dbl>	<dbl>	<dbl>
1 Sense & Sensibility	elinor	623	0.00519	1.79	0.00931
2 Sense & Sensibility	marianne	492	0.00410	1.79	0.00735
3 Mansfield Park	crawford	493	0.00307	1.79	0.00551
4 Pride & Prejudice	darcy	373	0.00305	1.79	0.00547
5 Persuasion	elliot	254	0.00304	1.79	0.00544
6 Emma	emma	786	0.00488	1.10	0.00536
7 Northanger Abbey	tilney	196	0.00252	1.79	0.00452
8 Emma	weston	389	0.00242	1.79	0.00433
9 Pride & Prejudice	bennet	294	0.00241	1.79	0.00431
10 Persuasion	wentworth	191	0.00228	1.79	0.00409



## 4. 통계적 분석

### - 분석 기법 소개 및 선정 : Word2Vec



검색어의 연관도 분석을 위해 사용한 것으로  
2013년 구글에서 개발한 방법론이다.  
이를 그대로 단어(word)를 벡터(vector)로  
바꿔주는 것으로 이로 인해 단어를 수치화 하여  
계산 할 수 있게 된다.

Word2Vec분석이 되었다면 [ eat ]을 넣었을 때  
다음 단어로 [ apple ], [ orange ], [ rice ]가  
올 것을 예상 할 수 있다.

Word2Vec을 이용하여 연관도를 찾게 된다.

# 4. 통계적 분석

## - 분석 기법 적용 : Word2Vec

$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w=1}^W \exp(u_w^T v_c)}$$

중심단어(c)가 주어졌을 때 주변단어(o)가 나타날 확률

	경고	노동조합	파업	난색	인원	요구사항	기간	난항	정도	계속	금요일	총파업	교대	인상	시작	임금피크자	입장	총원	돌입	지속
경고	0	0.64982	0.46396	0.47723	0.7036	0.41837	0.50492	0.68269	0.67533	0.44711	0.34285	0.51754	0.4128	0.42273	0.51295	0.57996	0.69853	0.47359	0.59845	0.54157
노동조합	0.64982	0	0.617	0.57603	0.50822	0.67741	0.69101	0.5784	0.6689	0.433	0.5145	0.60635	0.41156	0.63789	0.75305	0.38746	0.78303	0.57655	0.75265	0.68296
파업	0.46396	0.617	5.96E-08	0.73135	0.74071	0.76447	0.6444	0.77665	0.8206	0.50592	0.43533	0.60961	0.70027	0.79388	0.66177	0.72362	0.88671	0.77784	0.68428	0.70983
난색	0.47723	0.57603	0.73135	0	0.51777	0.17791	0.65019	0.1747	0.54362	0.65654	0.66008	0.35853	0.34807	0.54638	0.72764	0.24136	0.26489	0.29102	0.35145	0.48057
인원	0.7036	0.50822	0.74071	0.51777	0	0.58802	0.38182	0.59711	0.06709	0.59193	0.73124	0.80518	0.64779	0.69527	0.65947	0.48613	0.54256	0.56182	0.67642	0.81768
요구사항	0.41837	0.67741	0.76447	0.17791	0.58802	5.96E-08	0.70764	0.31753	0.56373	0.75417	0.81362	0.41073	0.30212	0.30224	0.75901	0.39201	0.3112	0.29987	0.39034	0.53947
기간	0.50492	0.69101	0.6444	0.65019	0.38182	0.70764	5.96E-08	0.76402	0.40354	0.47188	0.6679	0.753	0.79164	0.77225	0.65636	0.67515	0.6654	0.69745	0.75823	0.84103
난항	0.68269	0.5784	0.77665	0.1747	0.59711	0.31753	0.76402	0	0.64332	0.71547	0.8017	0.44851	0.46923	0.68297	0.75465	0.34285	0.16142	0.48211	0.3709	0.23965
정도	0.67533	0.6689	0.8206	0.54362	0.06709	0.56373	0.40354	0.64332	0	0.76872	0.80023	0.86539	0.66258	0.66083	0.60782	0.57599	0.50119	0.52622	0.69194	0.87041
계속	0.44711	0.433	0.50592	0.65654	0.59193	0.75417	0.47188	0.71547	0.76872	0	0.41006	0.65201	0.68676	0.7964	0.77878	0.64913	0.84778	0.82156	0.80022	0.60931
금요일	0.34285	0.5145	0.43533	0.66008	0.73124	0.81362	0.6679	0.8017	0.80023	0.41006	5.96E-08	0.5516	0.562	0.72268	0.35185	0.55109	0.99257	0.63575	0.7251	0.63988
총파업	0.51754	0.60635	0.60961	0.35853	0.80518	0.41073	0.753	0.44851	0.86539	0.65201	0.5516	0	0.40977	0.62151	0.64359	0.39256	0.70129	0.44724	0.20705	0.57274
교대	0.4128	0.41156	0.70027	0.34807	0.64779	0.30212	0.79164	0.46923	0.66258	0.68676	0.562	0.40977	0	0.22832	0.6354	0.15919	0.59678	0.11729	0.49562	0.53076
인상	0.42273	0.63789	0.79388	0.54638	0.69527	0.30224	0.77225	0.68297	0.66083	0.7964	0.72268	0.62151	0.22832	5.96E-08	0.72995	0.47105	0.6359	0.24068	0.71767	0.65927
시작	0.51295	0.75305	0.66177	0.72764	0.65947	0.75901	0.65636	0.75465	0.60782	0.77878	0.35185	0.64359	0.6354	0.72995	#####	0.62839	0.79993	0.62272	0.53138	0.72298
임금피크자	0.57996	0.38746	0.72362	0.24136	0.48613	0.39201	0.67515	0.34285	0.57599	0.64913	0.55109	0.39256	0.15919	0.47105	0.62839	0	0.53501	0.17164	0.43029	0.53854
입장	0.69853	0.78303	0.88671	0.26489	0.54256	0.3112	0.6654	0.16142	0.50119	0.84778	0.99257	0.70129	0.59678	0.6359	0.79993	0.53501	0	0.51742	0.54543	0.35633
총원	0.47359	0.57655	0.77784	0.29102	0.56182	0.29987	0.69745	0.48211	0.52622	0.82156	0.63575	0.44724	0.11729	0.24068	0.62272	0.17164	0.51742	0	0.47814	0.65478
돌입	0.59845	0.75265	0.68428	0.35145	0.67642	0.39034	0.75823	0.3709	0.69194	0.80022	0.7251	0.20705	0.49562	0.71767	0.53138	0.43029	0.54543	0.47814	0	0.64234
지속	0.54157	0.68296	0.70983	0.48057	0.81768	0.53947	0.84103	0.23965	0.87041	0.60931	0.63898	0.57274	0.53076	0.65927	0.72298	0.53854	0.35633	0.65478	0.64234	5.96E-08

# 4. 통계적 분석

## - 분석 기법 적용 : TF - IDF

```
unitList = []
for i, line in enumerate(strList):
    morpParcellist = line.split("\t")[1].replace(
        "", "").replace("", "").split(" + ")

    morpMetaList = []
    for elem in morpParcellist:
        splited = elem.split("/")
        if \
            splited[0] != "" and \
            splited[1] != "NP" and \
            splited[1] != "NR" and \
            (
                splited[1][0] == "N" or \
                splited[1][0] == "S" or \
                splited[1] == "XSN"
            ):
            morpMetaList.append(splited)
```

```
C:\Users\student>C:/Users/student/AppData/Local/Programs/Python/Python37-32/python.exe "c:/Users/student/Documents/jinahan/통계분석/Ptython/20191016 파이썬을 이용한 단어 빈도수 계산.py"
[{'tag': '파업\n', 'count': 360}, {'tag': '운행\n', 'count': 237}, {'tag': '지하철\n', 'count': 149}, {'tag': '노조\n', 'count': 126}, {'tag': '서울\n', 'count': 115}, {'tag': '16\n', 'count': 93}, {'tag': '열차\n', 'count': 82}, {'tag': '요구\n', 'count': 78}, {'tag': '14\n', 'count': 69}, {'tag': '오전\n', 'count': 68}, {'tag': '서울시\n', 'count': 68}, {"tag": "노동조합\n", "count": 61}, {"tag": "1~8선\n", "count": 62}, {"tag": "준법투쟁\n", "count": 61}, {"tag": "철도노조\n", "count": 59}]

C:\Users\student>
```

IDF까지 진행하지 않음.

형태소 분석단계에서 불용어와 의미가 없는 단어들을 제거를 해주기 때문에,

TF-IDF값이 의미가 없어짐.

## 4. 통계적 분석

---

### - 분석 기법 적용 : Reduced Term Frequency

1. 바나나
2. 바나나, 키위, 키위, 포도, 포도
3. 바나나, 키위, 키위, 키위, 포도, 포도

#### RTF로 알 수 있는 것

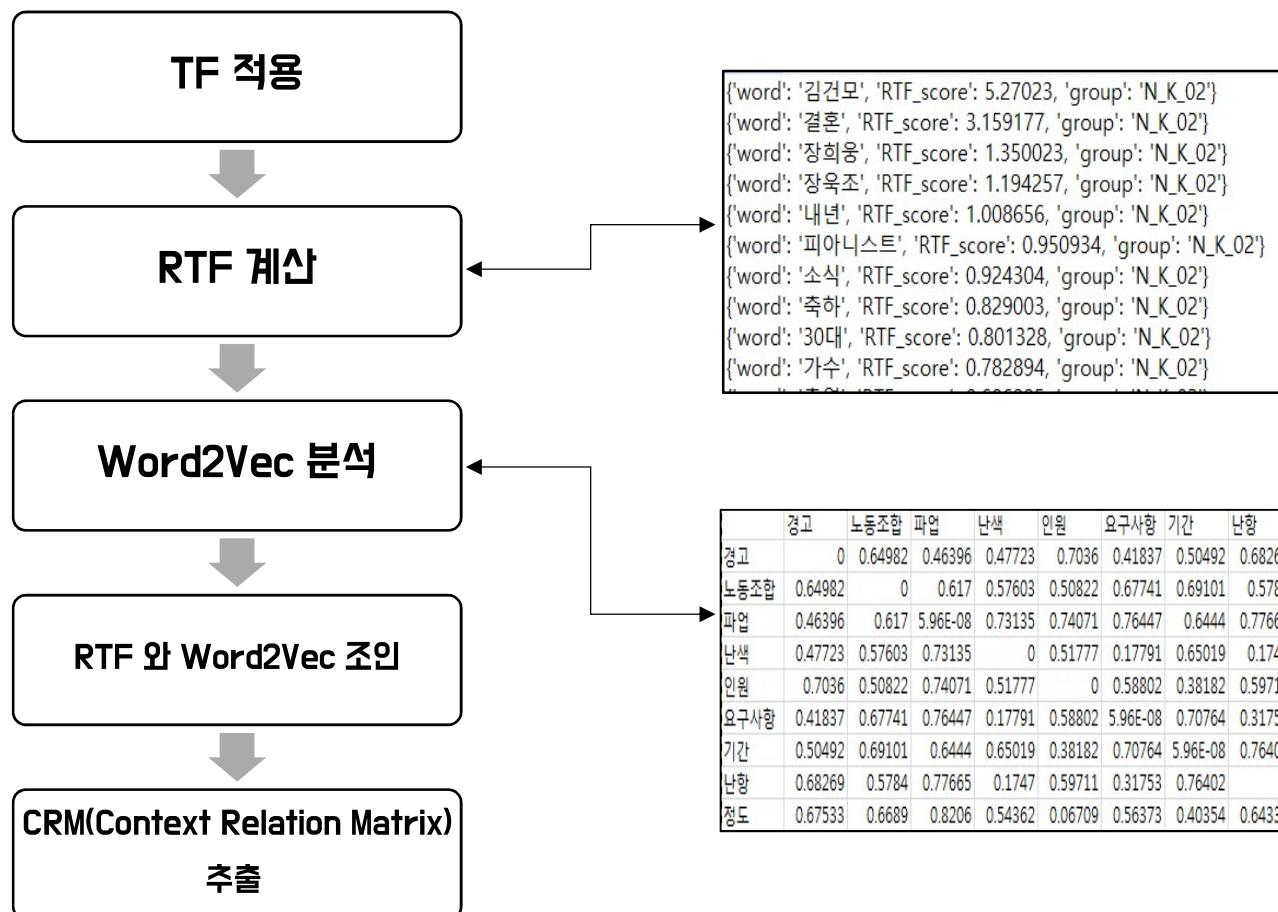
- Word Count에만 의존하지 않는 키워드 빈도수
- 문서 여러 개를 통틀어서 중요한 키워드 추출이 가능하다.

문장	단 어 빈도 수 (TF)		
1번	바나나(1)		
2번	바나나(0.2)	키위(0.4)	포도(0.4)
3번	바나나(0.17)	키위(0.5)	포도(0.33)
합계(RTF)	1.37	0.9	0.73

KeyWord	RTF	Word Count
바나나	1.37	3
키위	0.9	5
포도	0.73	4

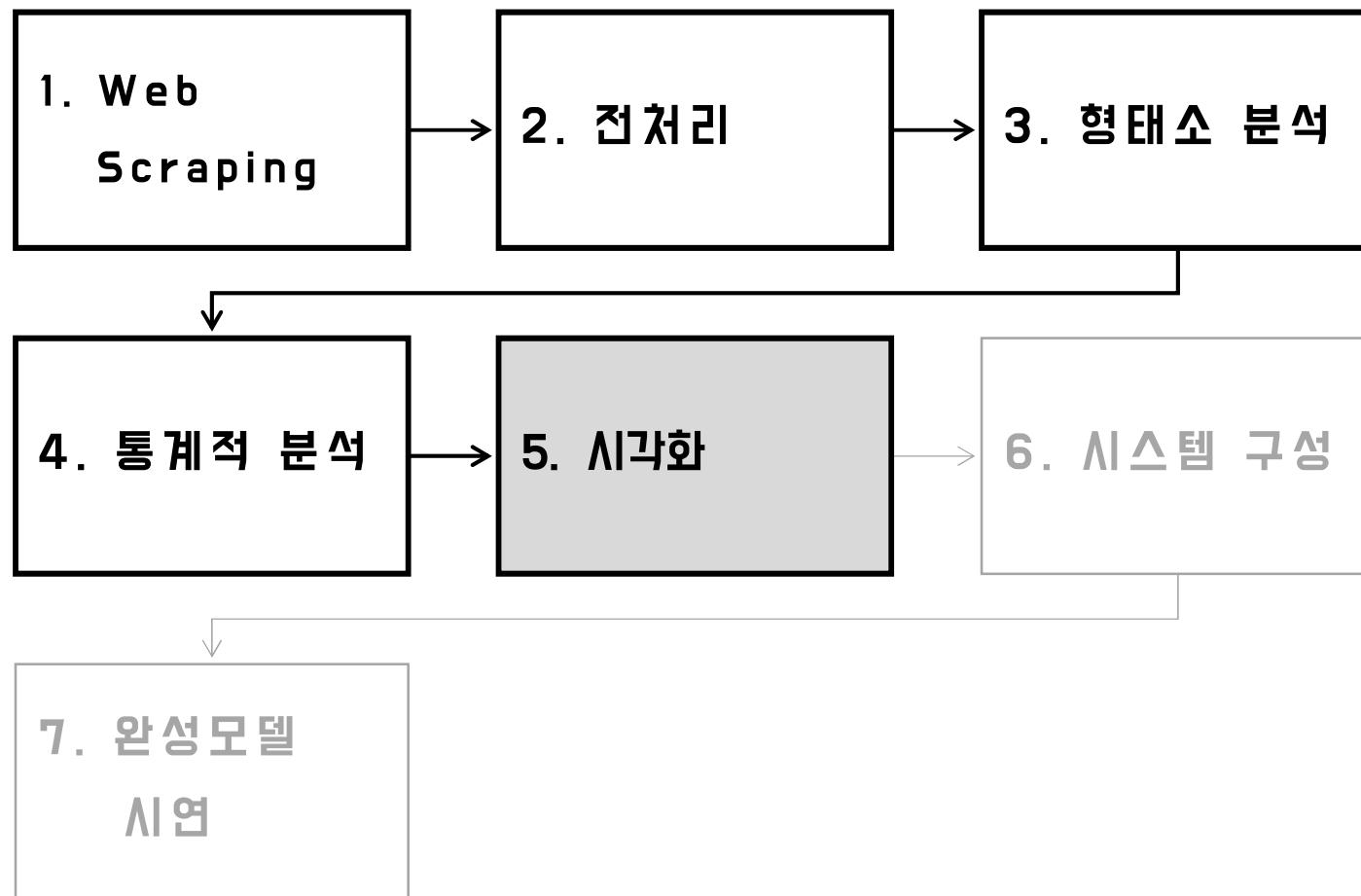
## 4. 통계적 분석

- 분석 기법 적용 : RTF + Word2Vec → Context Relation Matrix



# 프로젝트 과정

---

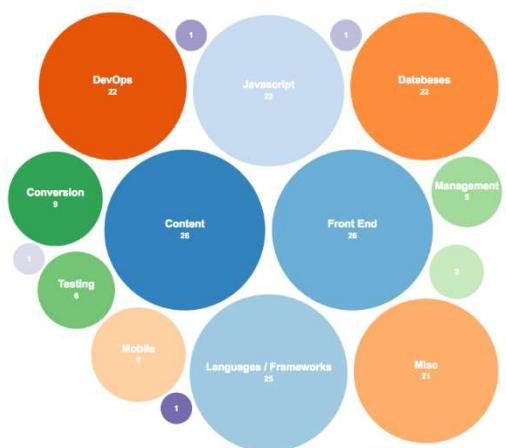


## 5. 시각화

#### - 시각화 모델 분석 : 제외한 모델

## 1. Bubble chart

Bubble chart는 각 데이터들 간의 크기를 한눈에 비교해서 보여주는 기법으로, 뉴스에서 단어들의 수량 표현 외의 한계가 있었다.



## 2. Word cloud

Word cloud는 핵심단어를 시각화하는 기법으로, 뉴스에서 많이 나온 단어를 특정하는 표현 외의 학계가 있었다.



### 3. Tree map

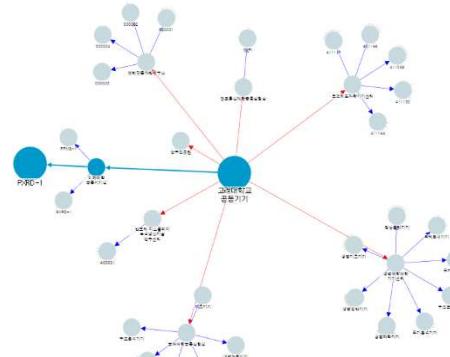
Tree map은 많은 계층 구조(트리 구조) 데이터를 표현하는데 적합한 기법으로.  
뉴스 기사에서 단어들 간의 관계를  
표현하는데 합계가 있었다.

## 5. 시각화

#### - 시각화 모델 선정 및 적용 : 키워드 네트워크

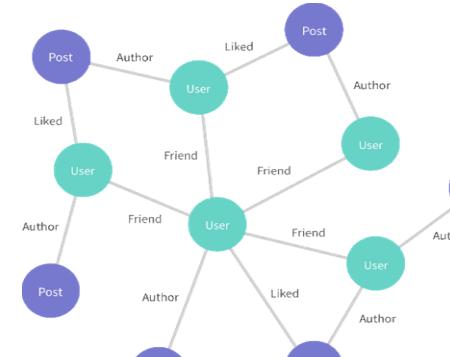
## 키워드 네트워크 1단계.

먼저 키워드(노드)를 표현하고,  
다음으로 키워드 간의 관계(링크)를  
표현한다. 그러나 1단계 모델로는  
일차원적인 관계밖에 표현할 수 없어  
한계가 있었다.



## 키워드 네트워크 2단계.

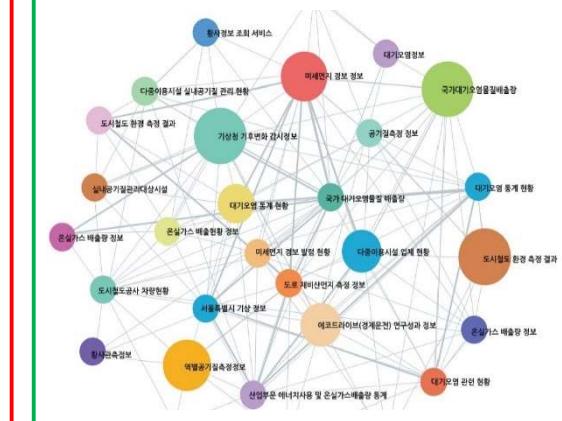
키워드(노드)와 키워드가  
한 개 이상의 관계(링크)로 연결되었다.  
그러나 2단계 모델은 뉴스에서  
**핵심 키워드**들 간의 **모든 관계를**  
표현하는데 한계가 있었다.



키워드 네트워크란?  
키워드는 노드로 표현하고  
키워드 간의 관계는 링크로 표현하는 기법

## 키워드 네트워크 3단계.

노드의 크기로써 키워드의 중요도를  
표현하고, 링크의 연결로써 서로의  
연관성을 표현하고, 링크의 길이로써  
키워드 간의 연관된 정도까지,  
모두 표현하게 되었다.

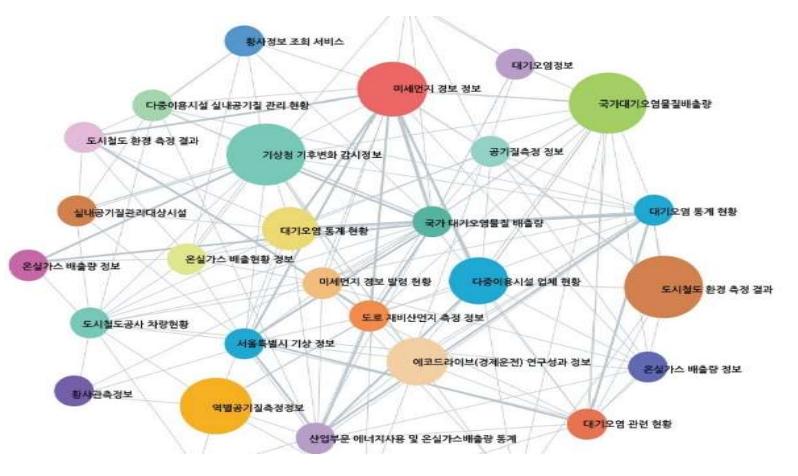


# 5. 시각화

## - 시각화 모델 선정 및 적용 : 키워드 네트워크

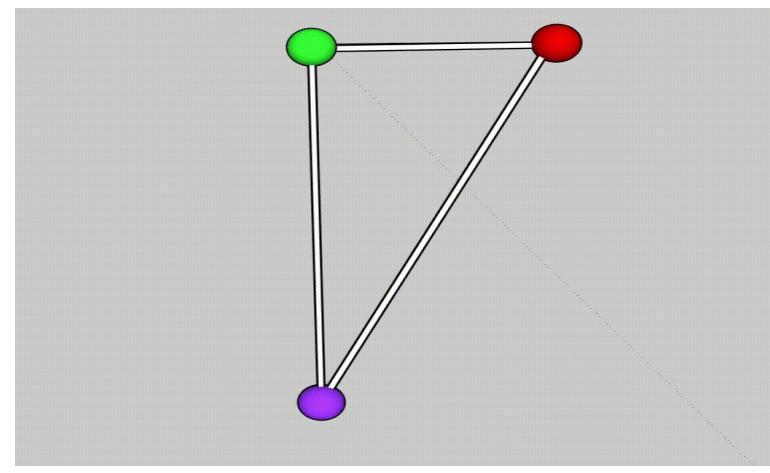
### 2D 표현의 한계.

키워드(노드)와 키워드 간의 관계(링크)가 많을 경우.  
그 관계들을 보기 가 어려웠다. 또한 이러한 데이터들의 집단이  
여러 군집을 이루고 있고, 그 군집들끼리도 서로 관계가 있다면.  
그것을 표현하는데 한계가 있었다.



### 3D 표현의 장점.

키워드(노드)와 키워드 간의 관계(링크)를  
다양한 각도에서 볼 수 있으므로  
많고 복잡한 관계의 데이터들 속에서  
키워드 간의 관계를 정확히 표현할 수 있다.

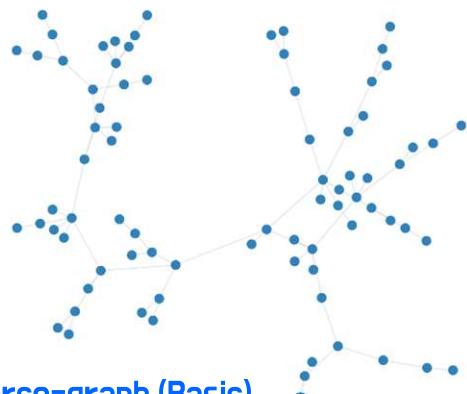


# 5. 시각화

## - 시각화 모델 설정 및 적용 : 키워드 네트워크 3D 시각화

### 1. javascript 라이브러리 사용

D3.js는 훌륭한 시각화 라이브러리이긴 하지만 3D 표현이 부족하여 Three.js(웹브라우저로 3D를 표현하는 라이브러리)를 활용하고자 하였다. 덧붙여 네트워크 시각화에 좋은 3d-force-graph(Three.js기반)를 활용하고자 하였다.



3d-force-graph (Basic)

### 2. 라이브러리의 한계로 wrapping작업

그러나 작업을 진행하면서 주어진 라이브러리만으로는 표현에 한계가 있었다. 그래서 직접 베이스 소스를 만들고, 그 위에 일부의 라이브러리 소스를 Wrapping하는 방식으로 작업하였다.

```
<body>
  <!-- GNB -->
  <!-- %@ include file="global/gnb.jsp" %-->

  <!-- Main Contents [START]-->
  <div class="body">

    <!-- 페이지 넘기시 이미지 -->
    <div id="warppDiv"></div>

    <!-- 매인 3D 그래프 -->
    <div class="graph-wrap">
      <div id="3d-graph"></div>
    </div>

    <!-- 사이트 로고 -->
    <div id="spacehorseDiv"></div>

    <!-- 시계 -->
    <div class="clockcontainer"></div>

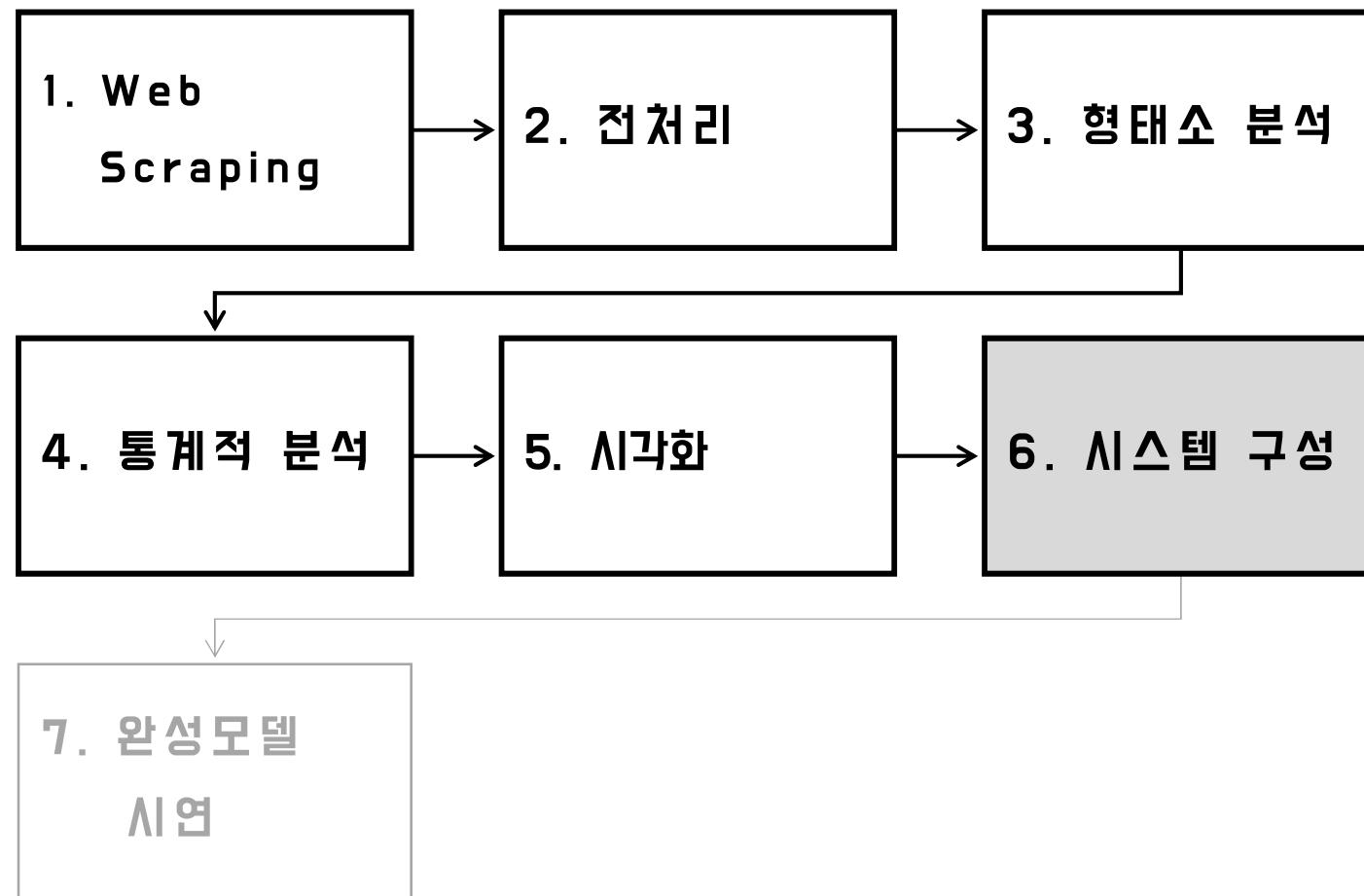
    <!-- Timeline 이동 -->
    <div class="timeline"></div>

    <!-- 주요키워드 이동 -->
    <div id="sigWordsWrap"></div>
  </div>

  /* 실제 3d-force-graph를 그리는 함수
   */
  function drawGalaxy(gData){
    // 그래프 그리기
    g_graph = ForceGraph3D()(document.getElementById('3d-graph'))
      .graphData(gData)
      .nodeAutoColorBy('group')
      .nodeThreeObjectEndo => {
        const obj = new THREE.Mesh(
          new THREE.SphereGeometry(10),
          new THREE.MeshBasicMaterial({
            depthWrite: false, transparent: true, opacity: 0
          })
        );
        const sprite = new SpriteText(node.word);
        sprite.color = node.color;
        sprite.textHeight = node.val;
        obj.add(sprite);
        return obj;
      }
    // linkDirectionalParticles(3)
    .linkOpacity(0.88)
    .nodeVisibility( node => node.val >= settings.NodeThreshold);
    //노드에 물体质하면 음색!
    g_graph.onNodeClick(node => searchingnode(node , gData));
    //g_graph.d3Force('charge').strength(-500);
  }
}
```

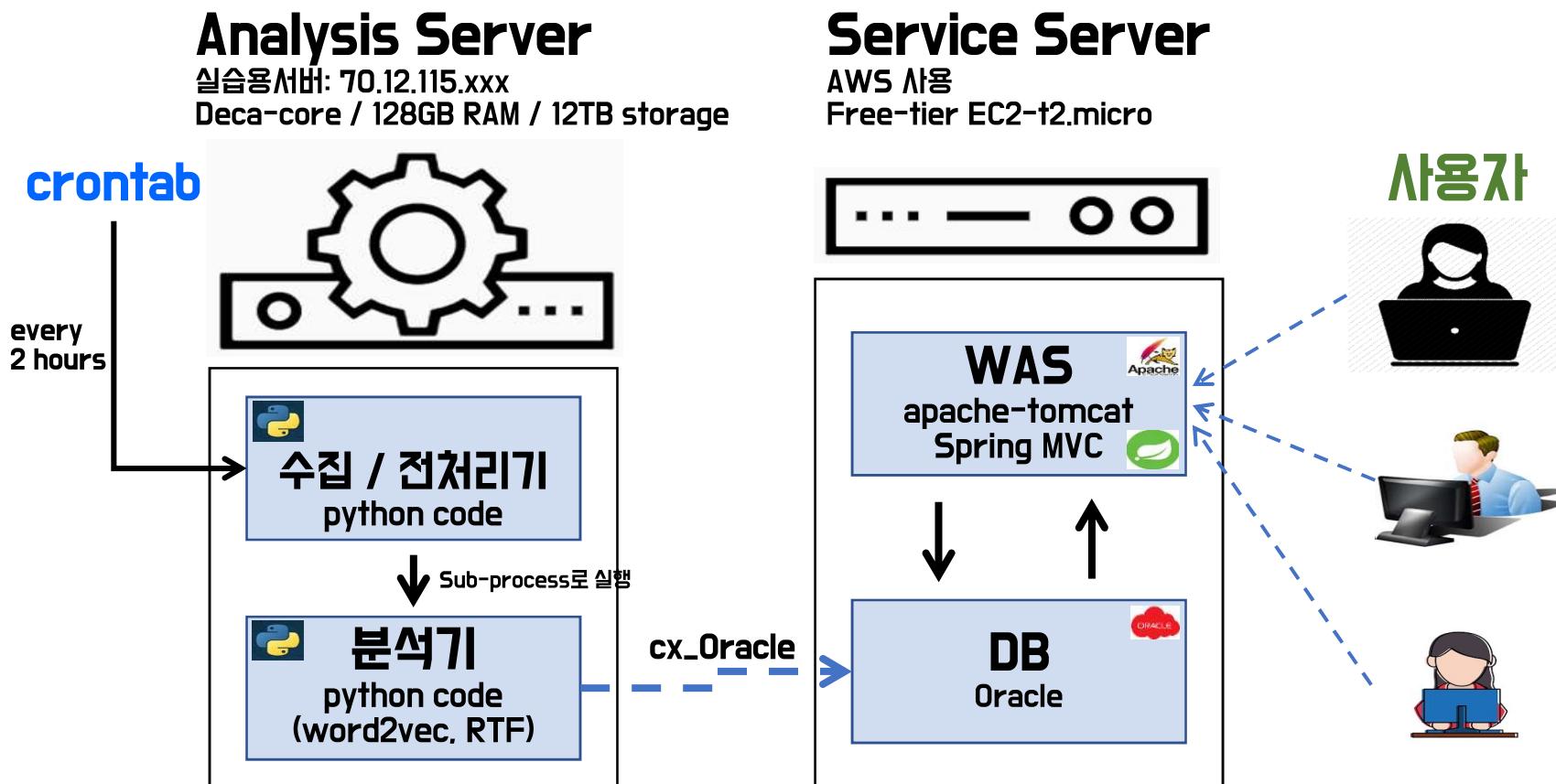
# 프로젝트 과정

---



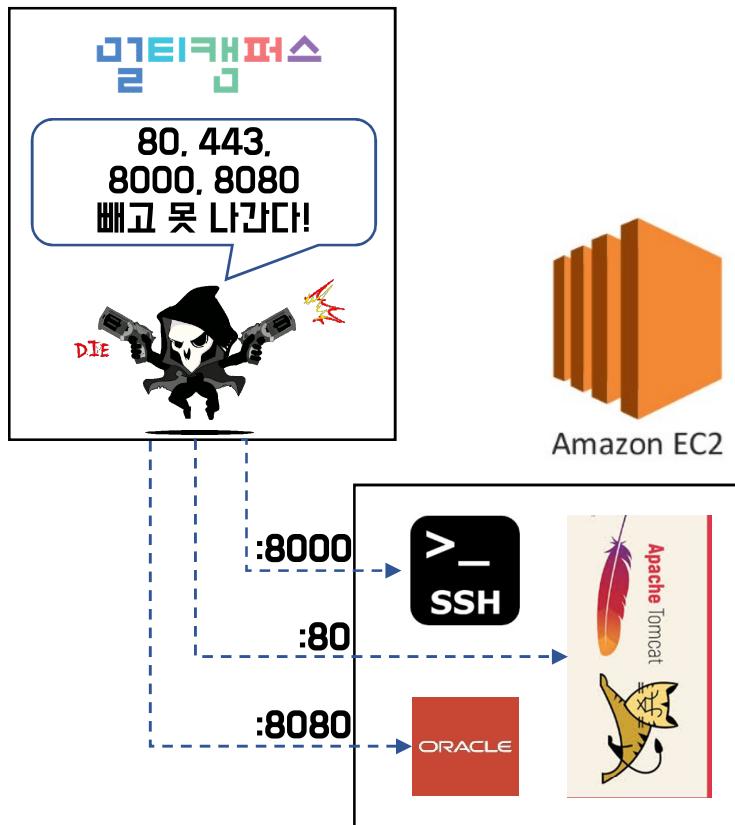
# 6. 시스템 구성

## - 시스템 구성 및 전체 프로세스 모식도



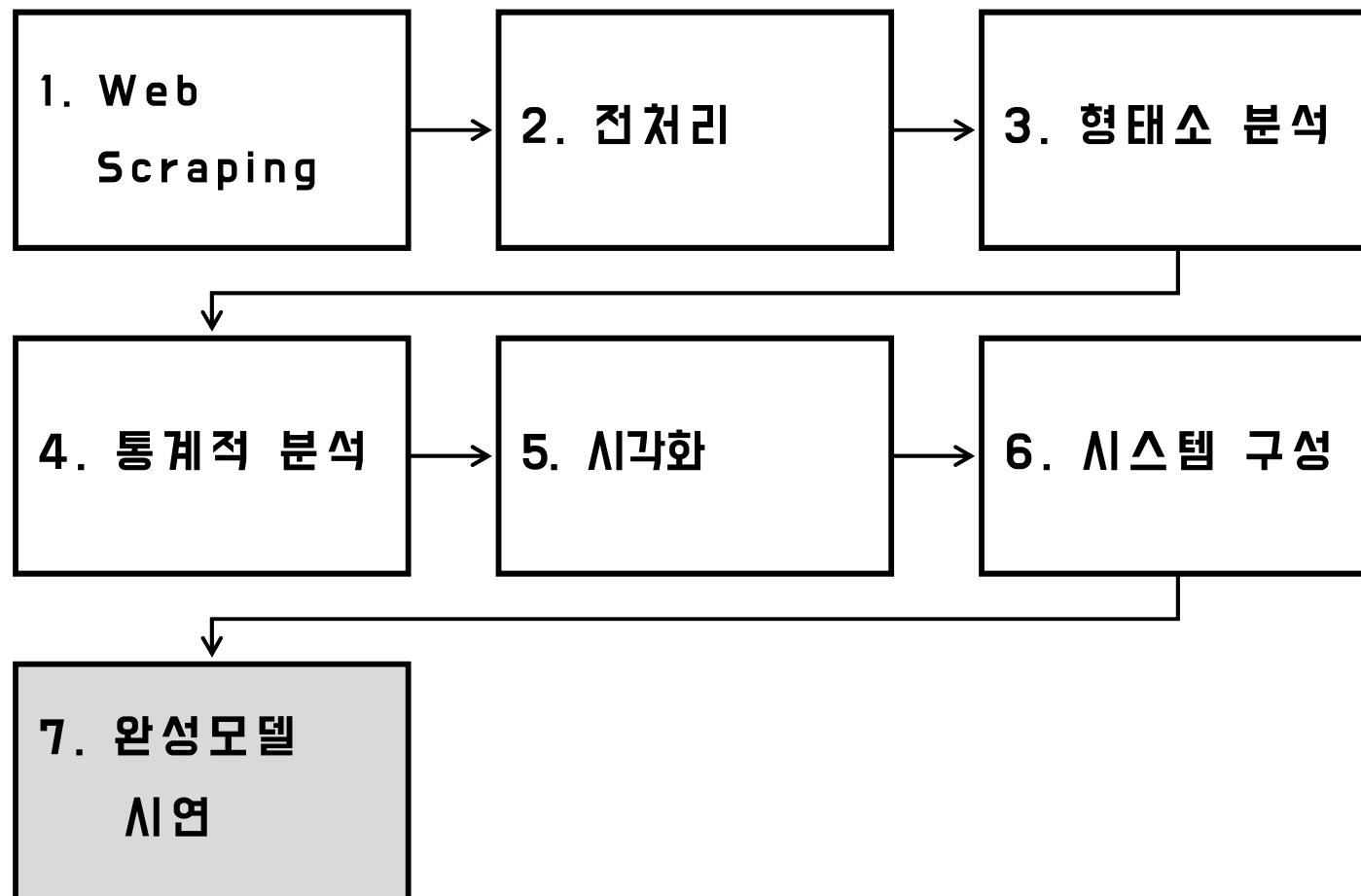
# 6. 시스템 구성

## - AWS 개발 및 배포환경 설정 / 도메인 네임 설정



# 프로젝트 과정

---



## 7. 완성모델 시연

---

**www.issue**what**show.com**

**www.issue**what**show.com**

**www.issue**what**show.com**

---

---

## □ 추후 개발예정 사항

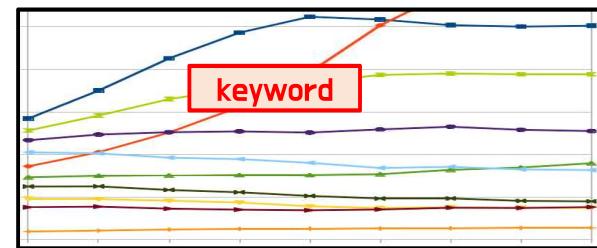
### 1. 사이트별 키워드로 뉴스 보기

ex) 같은 뉴스가 사이트들마다 차이가 있나?

**Daum vs NAVER ?**

### 2. 특정 키워드의 시각별 중요도 보기

ex) 그 뉴스는 언제부터 언제까지 떠 있었을까?



### 3. 월별, 일별, 계절 등으로 변화 보여주기

ex) 긴 기간 유지되는 키워드 분석을 통하여 시국의 변화 양상을 확인하는 서비스를 제공하려 한다.

### 4. 다양한 기기(모바일, 패드 등) 환경에서 최적화된 서비스 구현하기

ex) 창 사이즈만 바뀌는 것이 아니라 기능들도 사용할 수 있도록..

---

# ▢ 만든 사람들

나 때는 말이야 ~

라떼는 말이야 ~

Latte is horse ~

We are Latte Is Horse !



박종선 (조장)

"팀의 분위기 메이커"

- Web Scraping
- 형태소 분석
- 시각화
- PPT 준비



김재현

"팀의 빛과 소금같은 존재"

- Web Scraping
- 형태소 분석
- 전처리
- 통계분석
- DB 및 서버 시스템 구성



이희수

"분석이면 분석, 코드면 코드.  
팀의 멀티플레이어"

- Web Scraping
- 형태소 분석
- 통계분석(Word2Vec)
- 시각화



문진한

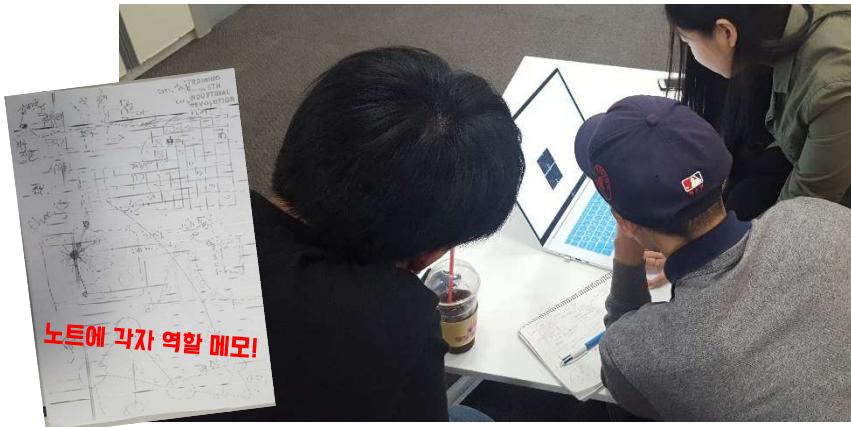
"팀의 날카로운 분석가"

- Web Scraping
- 형태소 분석
- 통계분석(TF-IDF)
- PPT 준비 및 발표

# ㅁ 만든 사람들



제 1장. 하루에 한번은 회의. 그리고 분업화!



제 3장. 치열한 코드 전쟁... (with Github)

제 2장. 작업은 각자. 코드는 모듈화하여 정리.



사이트별 검색어  
가져오는 모듈...

사이트별 뉴스  
검색하는 모듈...

1차 전처리 모듈.  
2차 전처리 모듈...

형태소 분석기 모듈...

통계적 분석기 모듈...

각자에 자리에서  
일하는 코드들 !

제 4장. 쉬는 날은 밖에서 작업... 그리고 기분전환!



## ▣ 사용기술 및 참고문헌

---



# ▣ 사용기술 및 참고문헌

---

## 형태소 분석기

Khaiii(Kakao Hangul Analyzer 3), Komoran(Shineware)

## 통계적 분석기법(TF-IDF, Word2Vec)

장희원, 2015년, Word2Vec을 활용한 문서의 의미기반 탐색, 연세대학교 대학원

이유나, 2016년, 텍스트 문서 기반 연관 법령 검색 방법에 관한 연구, 연세대학교 대학원

강형석, 양장훈, 2019년, 한국어 단어 임베딩을 위한 Word2vec 모델의 최적화, 서울미디어대학원대학교

허지욱, 2018년, Word2vec를 이용한 한국어 단어 군집화 기법, 한국인터넷방송통신학회 논문지 제 18권 제5호

Sergey Brin and Lawrence Page, The Anatomy of a Large-Scale Hypertextual Web Search Engine

## 시각화 라이브러리

3d-force-graph Javascript library (with three.js)

## 그외 API, 라이브러리, 컨텐츠 딜리버 등...

Naver 뉴스 API, Daum 뉴스, Twitter API, cloudflare, Google CDN, BeautifulSoup4, pandas, yweather, numpy, konlpy, gensim and etc.

---

□ 마치며.. 느낀 점 & Comment..

박종선 - 이번 프로젝트는 실무에서 하듯 분업화를 경험하였고 또한 서비스를 실제로 상용화하며 평소 잘 몰랐던 서버구성의 전체적인 흐름을 배웠습니다.  
그리고 가장 중요한 것은, 교육수료 후 내가 무엇을 해야 좋을지 좋은 가이드가 되었습니다.  
앞으로는 React나 Vue.js도 공부해볼까 합니다.

**김재현** - “화 무 십 일 흥” “사  
“권 면 뮬레방아소리 들린다... “고  
불 “우물쭈물 하다 내 이럴 줄 알았지 ” 메기... ” “전 화 위 복 ” “와  
십 초 “내일은 내일의 태양이... ” 진 “권 신  
년 ” “Low man’s Lyric ” 가 “ “Every dog has his day ” 감 토 삼  
“BUT!” “일 체 유 심 조 ” “남 중 지 추 ” 래 “—————” 중 “  
“봉 왁 직 염 ” “아 미 건 아 닌가... ” “참 랑 지 수 탁 혜 가 이 탁 오 족 ”...

이번 프로젝트 기간은, 기억도 나지 않는 '나'. 그리고 잊고 살았던 도전과 열정이라는 멋진 단어들을 팀원들로부터 다시 선물 받는 값진 경험이었습니다. 모두! 고맙습니다.

## ▣ 마치며.. 느낀 점 & Comment..

---

- 문진한 - 이번 프로젝트를 통해서 데이터 분석가를 조금이나마 경험을 해 본 것 같아 더욱 뜻 깊은 것 같습니다. 데이터 분석가는 데이터를 다루기 위한 기술이 필요하고, 전처리를 한 데이터를 가지고 통계적인 접근으로 인사이트를 주는 직업입니다. 마치 모든 틀이 정해져 있고, 데이터만 잘 처리해서 넣으면 나올 수 있다는 생각은 버리게 되었습니다. 학원에서 진행한 프로젝트였지만, 어떤 통계 방식을 선정하고, 어떻게 데이터를 수집하는 방식에 대해서 깊이 생각을 하면서 앞으로 컴퓨터 언어와 통계 분석에 초점을 두고 공부를 해야겠다는 생각이 들었습니다.
- 이희수 - 다양한 분야의 사람들이 모여 한 프로젝트를 하며 배운점이 많았습니다. 같은 주제여도 각자가 생각하는 가 달라 '저런 생각을?!' 이런 생각이 드는 다양한 의견이 있어 이를 추합하는 과정이 너무 재미있었고. 협동하는 일이나 만큼 깃과 깃플로우를 이용해 코드와 일정을 정리하고 공유하는 과정에서 정말 한 줄 한 줄 내가 이 팀과 일하고 있구나 실감했습니다. 교육받은 과정을 거의 다 넣은 활동이라 이를 활용할 수 있었던 주제를 뽑아 만족스러웠고, 각각의 장점이 달라 이를 서로서로 배우며 지내는 과정에 이것이 협업이구나. 코딩은 진짜 협업을 해야 하는구나를 깨우치는 과정이었습니다.
-

## Q n A

---

질문 있으세요?

---

---

**고맙습니다.**

---