고급BA FINAL PROJECT

KAIST 경영공학부

20203986 남덕우

Contents

분석배경 및 데이터 설명

1. 데이터 소개

가짜 뉴스, 사회적 문제로 대두

진짜 뉴스 판별 정확도 평균

사람 60%

AI 95%

“가짜뉴스 판별에 AI가 사람의 능력보다 우수한 시대”

- 수많은 정보의 흐름 속에서 신뢰할 수 있는 뉴스에 대한 논쟁이 점차 가열되고 있음
- **NH금융과 데이콘 주관 ‘진짜뉴스 찾기’ 대회에 참가해 텍스트마이닝 모델 알고리즘에 대해 배워보고자함**

1. 데이터 소개

데이터 예시

뉴스 데이터 (‘20년 1월 - 6월) 의 진짜 유무를 판별하는 이진 분류 문제

Index번호 | 발행날짜 | 뉴스 제목 | 컨텐츠 | 정보유무

Id	date	title	content	info
NEWS02580	20200605	[마감]코스닥 기관 678억 순매도	[이데일리 MARKETPOINT]15:32 현재 코스닥 기관 678억 순매도	0
NEWS02580	20200605	[마감]코스닥 기관 678억 순매도	실적기반 저가에 매집해야 할 8월 급등유망주 TOP 5 전격공개	1
NEWS02580	20200605	[마감]코스닥 기관 678억 순매도	하이스탁론, 선취수수료 없는 월 0.4% 최저금리 상품 출시	1
NEWS02580	20200605	[마감]코스닥 기관 678억 순매도	종합 경제정보 미디어 이데일리 - 무단전재 & 재배포 금지	0

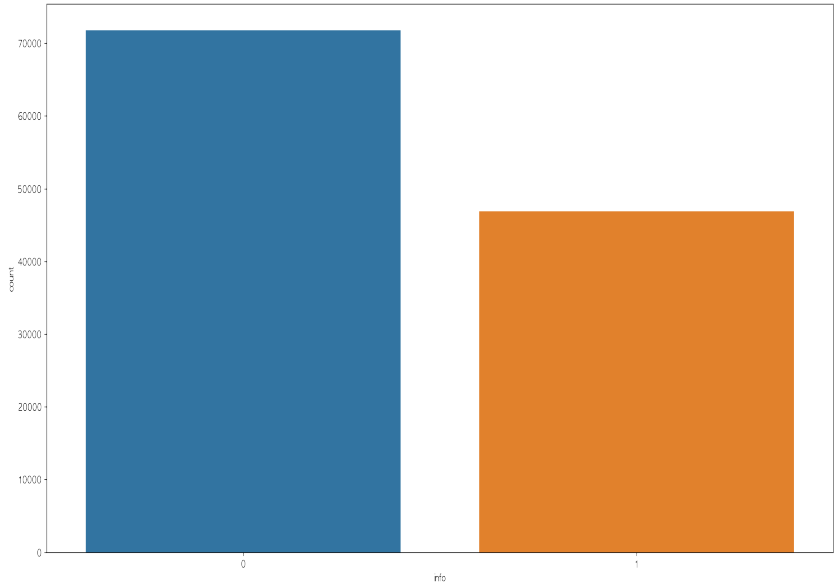
핵심변수로는 제목, 컨텐츠가 있으며, 목표변수는 info임

- Train 118,745건
- Test 142565 건
- Train, Test 기한동일
- 결측치/중복 없음

EDA

2. EDA

목표 변수

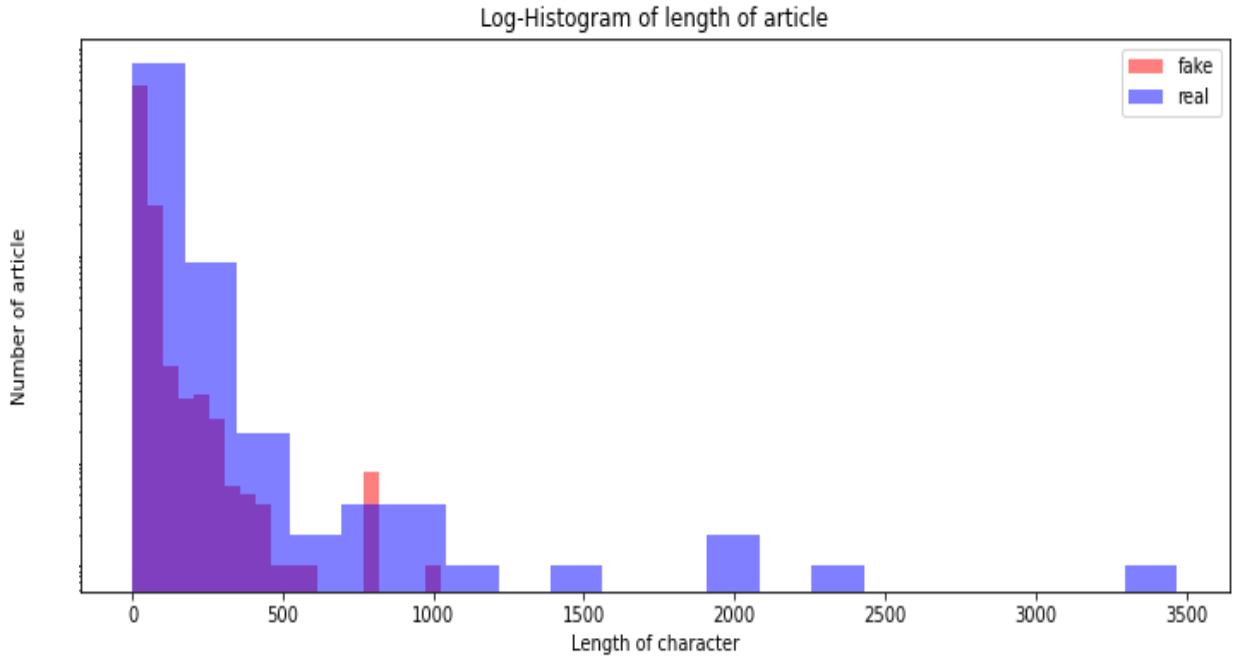


진짜, 가짜뉴스 비율 6:4이며, 월별 분포는 비교적 균등한편

- 진짜뉴스 71813 개
- 가짜뉴스 46932 개
- 1월 가짜뉴스 비율이 다소 낮으나 대체로 균등

2. EDA

뉴스기사 길이 분석



진짜 뉴스가 2배 정도로 긴 경향

- 전체 평균길이 50글자
- 진짜 뉴스 길이가 가짜뉴스 대비 2배, 표준편차도 2배 정도 높음

	가짜뉴스	진짜뉴스
최댓값	1022자	3469자
최소값	2 자	2자
평균값	34.47자	62.34자
표준편차	20.38 자	43.6자

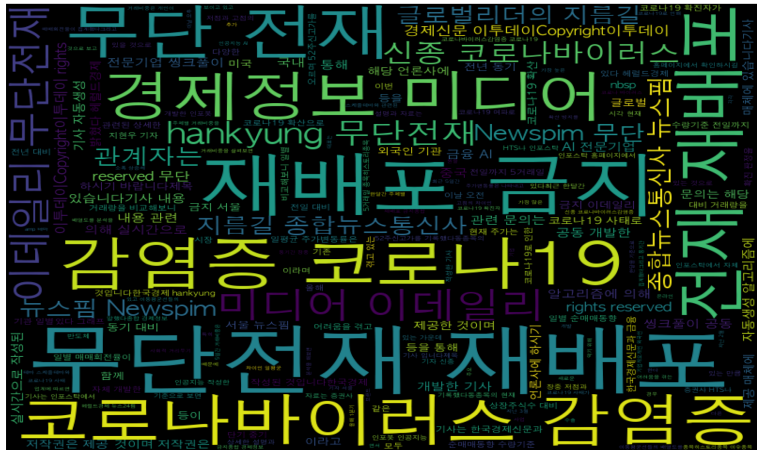
2. EDA

WORD CLOUD

기사에 단어 빈도수를 활용하여 워드클라우드 시각화를 수행하였음

가짜뉴스

진짜뉴스



금전 관련 키워드 중심

- 주식, 수익률, 대박주 등 단어성향이 뚜렷하고, 의도가 담겨있음

신문사, 저작권 키워드 다수

- Copyright 마크가 진짜판별에 주요 단서역할
- 코로나 등 경제나 사회 관련 정보전달 목적 기사 다수

2. EDA

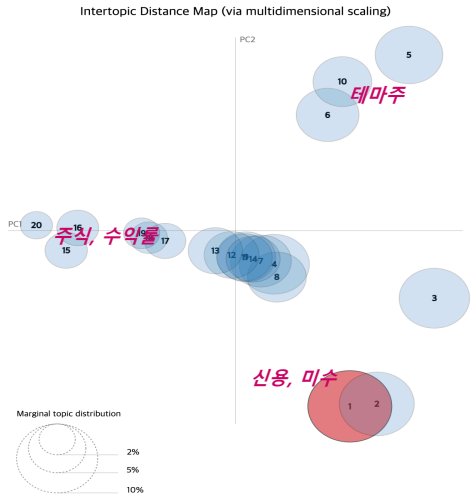
잠재 디리클레 할당(LDA 분석)

진짜, 가짜 뉴스를 구분하여 LDA 토픽 분석을 수행

* LDA는 단어가 특정 토픽에 존재할 확률과 문서에 특정 토픽이 존재할 확률을 결합확률로 추정하여 토픽을 추출함

* 각 원과의 거리는 각 토픽들이 서로 얼마나 다른지를 보여줍니다. 만약 두 개의 원이 겹친다면, 이 두 개의 토픽은 유사한 토픽이라는 의미

가짜뉴스

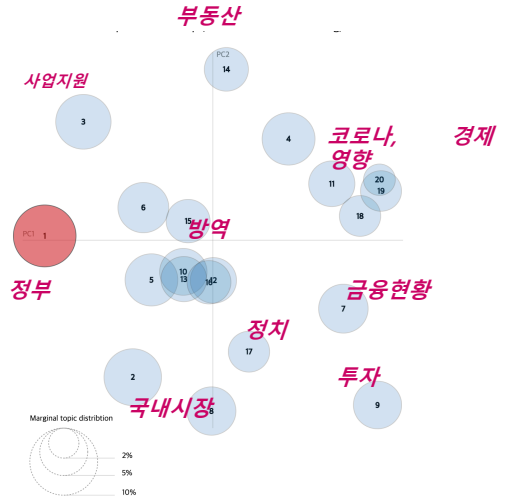


토픽이 획일화

주식, 테마주, 신용/미수 크게

3가지 분류 가능

진짜뉴스



토픽의 다양성

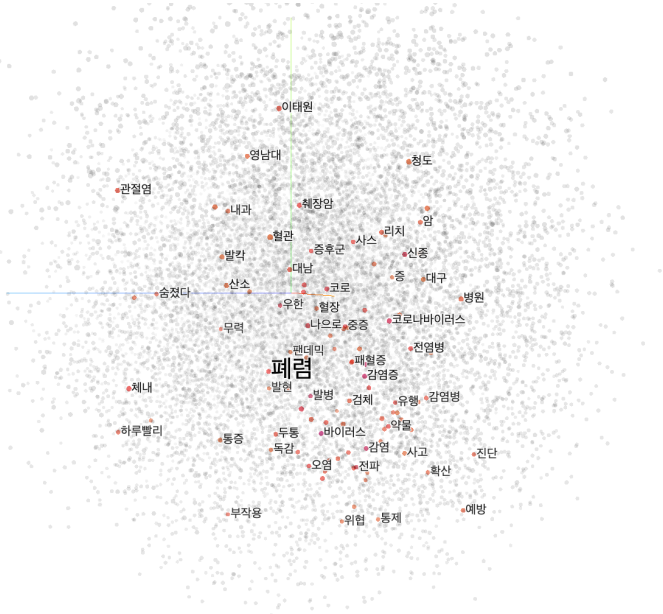
정부, 사업지원, 부동산, 방역, 정치, 국내시장, 투자, 금융, 코로나경제 등 주제 다변화

2. EDA

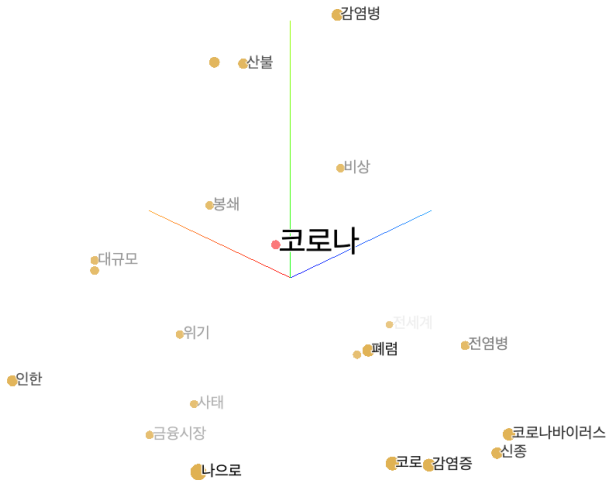
핵심단어 유사도 분석

주요 단어간 코사인 유사도를 분석하여 3차원공간에 맵핑을 수행

전체단어 분포



주요 단어 유사도



- 코로나와 가장 유사한 단어는 코로나바이러스, 감염증, 봉쇄이며, 신용은 미수, 금리, 취급 등으로 나타남

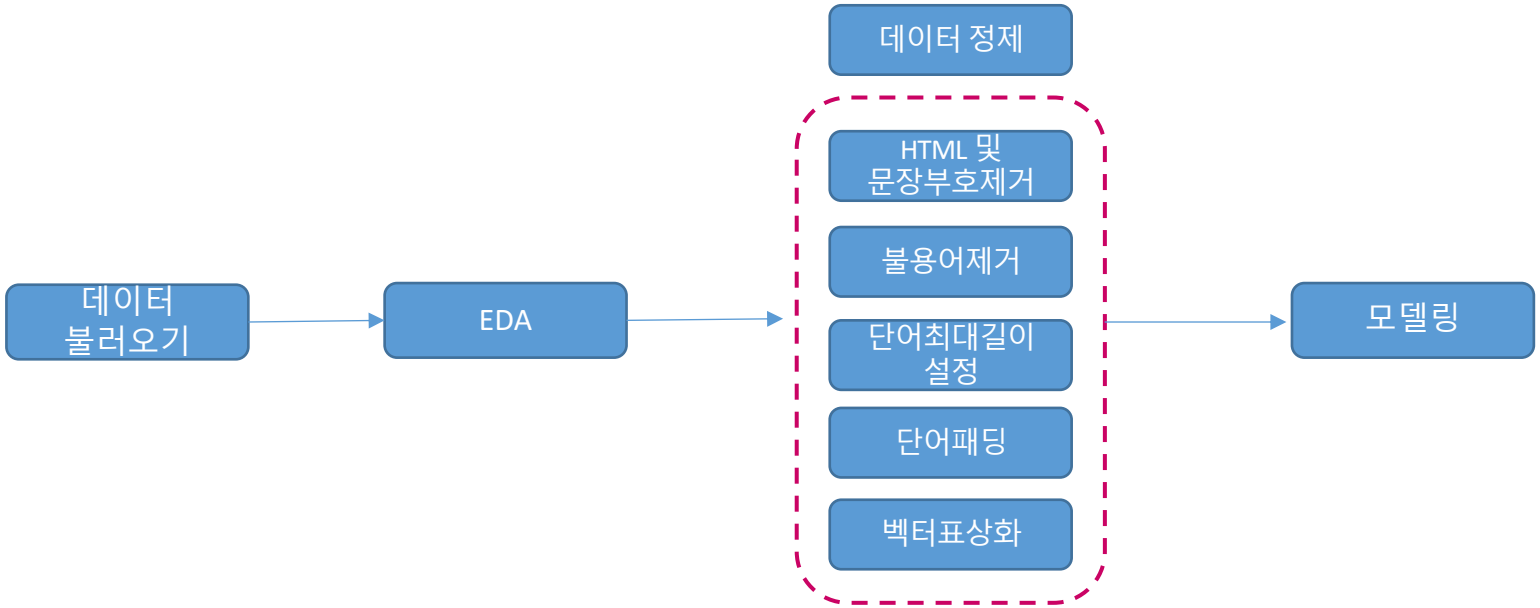
코로나		신용		테마주	
코로나바이러스	0.287	미수	0.117	단타	0.076
감염증	0.328	금리	0.289	공략	0.199
봉쇄	0.407	취급	0.344	대박	0.247
경제활동	0.412	최저	0.366	반사	0.279
감염병	0.414	수수료	0.428	수혜주	0.290

전처리

3. 전처리

텍스트 전처리 과정

머신러닝은 카운트 기반의 TF-IDF 방식을, 딥러닝은 Keras embedding layer 를 사용하여 전처리를 수행하였음



머신러닝

TF-IDF 방식사용

딥러닝

Kesras embedding layer

3. 전처리

PYHON CODE 예시

```
from tqdm import tqdm_notebook

for i in tqdm_notebook(range(0, len(train_df))):
    train_df["content"][i] = list(kkma.morphs(train_df["content"][i]))

from tqdm import tqdm_notebook

for i in tqdm_notebook(range(0, len(test_df))):
    test_df["content"][i] = list(kkma.morphs(test_df["content"][i]))

tk = Tokenizer(filters = "!#$%&()*+,-./:;<=>?@[\\]^_`{|}~\t\n, '")

tk.fit_on_texts(list(train_df["content"]) + list(test_df["content"]))

train_text = tk.texts_to_sequences(train_df["content"])
test_text = tk.texts_to_sequences(test_df["content"])

from keras.preprocessing.sequence import pad_sequences

max_len = 50

padded_train = pad_sequences(train_text, maxlen = max_len)
padded_test = pad_sequences(test_text, maxlen = max_len)
```

1. 형태소 분석(토큰화)
2. 특수문자 제거
3. 정수 인코딩
4. 최대단어길이 설정
5. 패딩

입력 데이터

```
array([[     0,      0,      0, ...,  40427,    490,     30],
       [     0,      0,      0, ...,     82,    160,    930],
       [     0,      0,      0, ...,    109,    405,    122],
       ...,
       [     0,      0,      0, ...,    165,    185,     99],
       [     0,      0,      0, ...,    147,    124,     41],
       [     0,      0,      0, ...,    147,    124,     41]],
```

형태소분석기	최대 단어길이	워드임베딩
Soynlp	*40개*	*사후학습*

모델링

4. 모델링

머신러닝/딥러닝 예측모델

전반적으로 대부분이 98% 이상의 우수한 성능을 보였으며, 딥러닝이 약간 더 우수하였음

	트레인 Accuracy	검증셋 Accuracy
로지스틱 회귀	0.9838	0.98
랜덤포레스트	0.999	0.984
LGBM	0.999	0.9914
Keras Squential	1.0	0.993
LSTM	0.999	0.987

*테스트 사이즈 0.2

- 머신러닝 모델로는 로지스틱회귀, 랜덤포레스트, LGBM을 수행하였으며, 이중에서 LGBM이 가장 우수
- 딥러닝 모델로는 기본 Sequential 모델이 LSTM보다 우수하였으며, 전체적으로도 가장 퍼포먼스가 뛰어남
- TF-IDF 방식의 머신러닝 모델과, 딥러닝 방식의 앙상블 가능

5. Conclusion

Result…

- 실제 테스트셋에서는 98%대의 정확도를 보임
- Overfitting문제 측면에서 아직 모델에 개선여지 있음

How to further improve…

- Micro 전처리 : 가짜 뉴스 오분류 체크 등 데이터 추가검증, STOPWORDS 커스터마이징 등
- Pretrained Word Embedding 사용 (FastText, 엘모 등)
- TFIDF and RNN into one ensemble : 앙상블 모델 시험
- 머신러닝/딥러닝 앙상블 실험
- 트랜스포머, Bert Model 등 상위 모델 시험

APPENDIX_딥러닝 모델링

사용된 Deep Learning Model 구조

기본 시퀀셜 모델

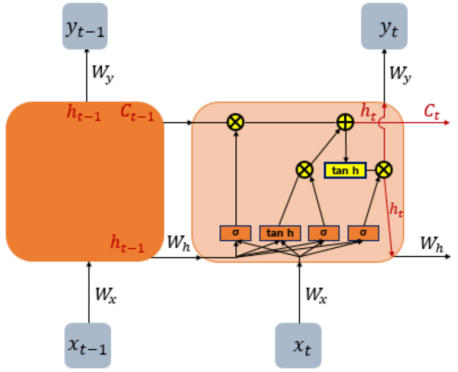
embedding_input: InputLayer	input:	[(?, 40)]
	output:	[(?, 40)]
embedding: Embedding	input:	(?, 40)
	output:	(?, 40, 30)
flatten: Flatten	input:	(?, 40, 30)
	output:	(?, 1200)
dense: Dense	input:	(?, 1200)
	output:	(?, 2)

** 최대단어길이 40 설정*

LSTM 모델

embedding_3_input: InputLayer	input:	[(?, 40)]
	output:	[(?, 40)]
embedding_3: Embedding	input:	(?, 40)
	output:	(?, 40, 32)
lstm_2: LSTM	input:	(?, 40, 32)
	output:	(?, 40, 30)
flatten_2: Flatten	input:	(?, 40, 30)
	output:	(?, 1200)
dense_3: Dense	input:	(?, 1200)
	output:	(?, 2)

LSTM 원리



기본개요

- 단어 최대길이 : 40
- 배치사이즈 : 128
- 에폭 : 20 (Patience = 5)
- Activation : Softmax 함수
- Loss Function : sparse_categorical_crossentropy
- Metric : 정확도 (Accuracy)

E.O.D