



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원 저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리와 책임은 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

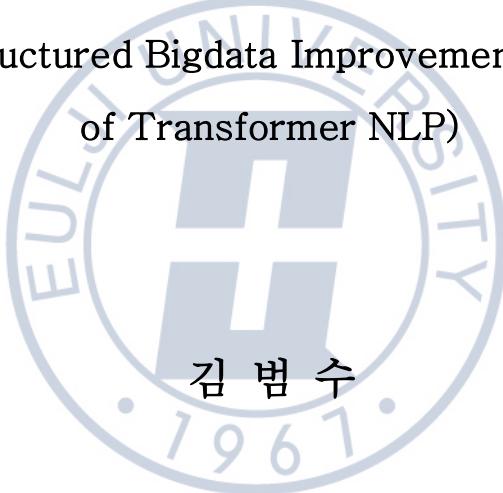
[Disclaimer](#)



박사학위논문

비정형 빅데이터 기반의 주가 추이
예측을 위한 트랜스포머 NLP 의 개선모델

(Stock Trend Prediction based on
Unstructured Bigdata Improvement Model
of Transformer NLP)



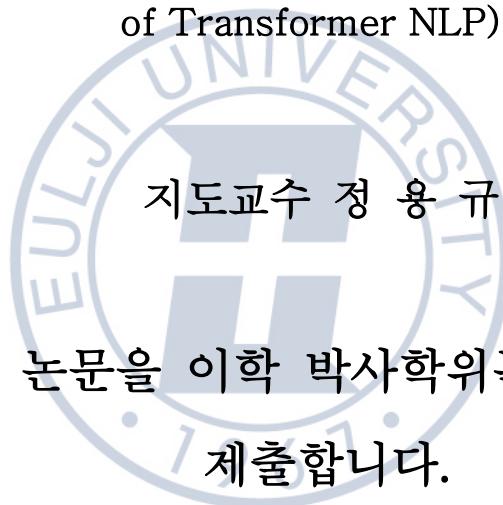
을지대학교 대학원

2021년 8월

박사학위논문

비정형 빅데이터 기반의 주가 추이
예측을 위한 트랜스포머 NLP 의 개선모델

(Stock Trend Prediction based on
Unstructured Bigdata Improvement Model
of Transformer NLP)



지도교수 정 용 규
이 논문을 이학 박사학위논문으로
제출합니다.

2021년 8월

을지대학교 대학원

의료 IT 마케팅학과

김 범 수

이 논문을 김범수의 박사학위 논문으로
인준함

2021년 6월

심사위원장 권영만 (인)

심사위원 강문식 (인)

심사위원 이규대 (인)

심사위원 정동근 (인)

심사위원 정용규 (인)

을지대학교 대학원

국문요지

비정형 빅데이터 기반의 주가 추이 예측을 위한 트랜스포머 NLP 개선 모델

주식 투자자의 연령대가 40~50대에서 인터넷 정보에 민감한 20~30대 MZ 세대로 연령층이 확대되고 있다. 이에 따라 투자 심리가 중요한 주식시장에서 종목뉴스, 주식 카페등 소셜미디어의 중요성이 부각되고 있다. 결국, 소셜미디어에 등장하는 텍스트가 주가에 주는 영향력이 더욱 커진 것이다. 최근에는 소셜미디어의 비정형 데이터를 활용하여 주가를 예측할 수 방법이 연구되고, 소셜미디어와 빅데이터가 주요 키워드로 자리잡고 있다. 이에 따라 NLP(Natural Language Process) 기반으로 각종 소셜미디어의 비정형 텍스트 감성 분석을 통해 주가의 향방을 예측하는 연구들이 등장하기 시작되었다. 이전 연구에서는 종목 관련 뉴스들을 크롤링한 후, 감성어휘사전을 가지고 긍정단어, 부정단어 언급 수를 활용해 뉴스가 긍정(1)인지 부정(0)인지로 인덱싱하고, 긍정 뉴스와 부정 뉴스의 수량을 가지고 해당 날짜, 해당 종목의 감성지수를 산출하는 연구가 있었다. 이 감성지수에 따른 해당 종목의 주가추이(상승 또는 하락/보합)와의 관련성을 머신러닝인 SVM, KNN, DT등으로 모델링하는 형태로 연구들이 진행되었다. 연구결과로 뉴스의 감성점수를 가지고 상승 또는 하락을 예측하는 정확도는 평균 66%로 나타났다.

그러나 이러한 연구들은 비정형데이터를 정형데이터로 변환하는 감성지수화라는 Feature Engineering 단계를 거치고 있다. 신속한 예측이 필요한 상황에서 추가적인 계산의 시간과 비용이 소요되며, 궁정, 부정 어휘들을 분류해 놓은 감성사전의 품질에 따라 결과가 좌우되기도 한다. 본 연구에서는 Transformer 기술의 딥러닝 알고리즘을 적용한 Pre-Trained Model인 KoBERT를 활용해 번거로운 Feature Extraction 없이 뉴스 텍스트 기반의 비정형데이터를 그대로 활용해 주가 추이를 예측하는 모델을 제시한다. 성능 비교 결과, 기존 뉴스 감성지수 기반의 주가 추이 예측 모델 대비, 평균 10% 이상(평균 정확도 77%), p-value 0.05이하로 통계적으로 유의한 개선 효과가 검증되었다. 즉, 번거롭고, 감성사전의 품질에 의존하는 감성지수로의 정형화라는 Feature Engineering 없이도 비정형데이터를 활용한 모델이 유의함을 확인할 수 있다. 또한 카테고리 형태의 Feature들을 텍스트에 함께 기재하는 방법을 제안하여 종전의 연구를 개선하였다. 비정형데이터와 정형데이터를 함께 학습을 시키려면 전처리 과정에서 비정형데이터를 정형데이터로 변환이 필요하다. 그러나 이러한 과정을 역공학적 모델로 제시하여 정형데이터를 비정형데이터에 포함시키는 방법을 모델링하여 개선모델을 제시하고 실험하였다. 비정형데이터에는 ‘뉴스기사’ 등을 들 수 있으며, 정형데이터는 ‘언론사’, ‘기자’ 등을 들 수 있는데, 정형데이터인 ‘언론사’, ‘기자’를 그대로 모든 뉴스기사 안에 포함시켜 모델링하였고,

실험한 결과, 통계적으로 유의한 차이($p\text{-value} < 0.05$)가 도출되었다. 또한 레이블의 분류 클래스를 3가지로 구분한 모델을 제시하였다. 기존 연구에서는 ‘상승’ 또는 ‘하락/보합’ 형태의 2가지 클래스 레이블을 정하는 방법을 개선하였다. 본 연구에서는 ‘지속 하락’, ‘하락 후 원복’, ‘보합/상승’ 형태의 3가지 클래스로 레이블을 구성하여 모델링하였다. 개선된 모델을 실험한 결과 평균 71%의 정확도로 기존 연구 대비 $p\text{-value}$ 값이 0.05미만으로 유의한 차이로 성능이 나오고 있음을 검증하였다. 본 연구의 개선모델은 NLP 비정형+정형 데이터를 그대로 활용해 학습함으로써 다양한 NLP 기반 예측에 활용할 수 있다는 데 기여할 것으로 기대된다. 코인추이를 예측하는 방법으로 예를 들어 <트윗내용 + 트윗인물 + 트윗시간대>을 결합하여 활용할 수 있고, 문학분야에서도 인문공학으로 발전할 수 있는 토대가 될 수 있다. 즉 <웹소설 1회 전문 + 소설 작가 + 장르 + 문장 길이>을 통한 웹소설의 흥행 여부를 예측할 수 있는 모델이 될 수 있다. 의학분야에서도 질병의 예측모델로 활용할 수 있다. 예를 들어 <문진내용 + 의사명 + 의사 전공 + 병원명> 등을 결합한 모델로 활용할 수 있다. 또한 <드라마 시나리오 + 제작사명 + 감독 + 주연 + 방송사>를 통한 시청률 추이 예측등 다양한 곳에 활용이 가능할 것으로 기대된다.

주요어: NLP, Transformer, BERT, 주가예측, 비정형데이터, 인공지능

차 례

국문요지	i
1. 서 론	1
1.1. 연구의 필요성	1
1.2. 연구목적과 연구 범위	6
2. 이론적 배경	12
2.1. NLP(Natural language Process)을 활용한 주가 예측에 대한 연구	12
2.2. 자연어처리 기술에 대한 연구	23
2.2.1. 자연어 처리에 대한 이해	23
2.2.2. Vector Representation	26
2.2.3. 언어모델 (Language Model)	31
2.2.4. Neural Language Model	34
2.2.5. Attention 과 트랜스포머	39
2.2.6. BERT(Bi-Directional Encoder Representations from Transformer)	44
3. 제안된 NLP 모델 설계 및 검증체계	47
3.1. 연구문제 및 NLP 모델 제안	47
3.2. 모델 성능 검증방법	55
3.2.1. 검증 프로세스 방법에 대한 개요	55
3.3. 검증을 위한 실험자료 및 실험설계	58
3.3.1. 실험 자료	58

3.3.2. 실험 방법	6 3
4. 실험 및 결과분석	7 4
4.1. 실험 데이터셋.....	7 4
4.2. 실험환경.....	7 6
4.3. 실험 결과.....	7 9
5. 고 찰	8 2
5.1. 가설 (가)에 대한 실험결과 분석.....	8 2
5.2. 가설 (나)에 대한 실험결과 분석.....	8 7
5.3. 가설 (다)에 대한 실험결과 분석.....	9 1
5.4. 가설 (라)에 대한 실험결과 분석.....	9 6
5.5. 성능향상을 위한 파라미터 튜닝에 대한 고찰.....	1 0 1
6. 결 론	1 0 4
참고 문헌.....	1 0 8
ABSTRACT	1 1 6

표차례

[표 1] 연구 범위 및 목적.....	1 0
[표 2] NLP 기반 주식 예측 연구 분야 및 연구 논문 예시	1 3
[표 3] NLP 기반 주가 예측 관련 연구 단계.....	1 4
[표 4] 기존 연구들에 대한 특징 요약 및 연구 방향	2 2
[표 5] 각 가설에 대한 검증 방법 및 검증에 대한 판단 기준 요약... ..	5 6
[표 6] 실험을 위해 수집한 데이터셋	5 8
[표 7] Label 유형 – 2가지 Type	6 1
[표 8] 실험을 위한 데이터셋 유형	6 2
[표 9] 실험 환경	6 3
[표 10] 단계별 실험 설계	6 4
[표 11] 모델별 학습 소요시간.....	7 6
[표 12] Validation Accuracy – 결과	7 9
[표 13] Test Accuracy – 결과	7 9
[표 14] 실험결과–연구가설별 검증 방법 및 판단 기준	8 0
[표 15] KoBERT Confusion Matrix와 Classification Scores.....	8 2
[표 16] 이전 연구들의 결과 정확도 비교표	8 3
[표 17] 하락으로 예측된 경우 공통적으로 나타나는 단어들	8 6
[표 18] 언어모델별 학습 시간 및 Accuracy.....	8 8
[표 19] Title만 활용 학습 결과 (Confusion Matrix)	9 1
[표 20] Title+언론사 기반 학습 결과 (Confusion Matrix)	9 1
[표 21] Title+언론사+기사 기반 학습결과(Confusion Matrix).....	9 2
[표 22] 3–Classes 학습결과 – Confusion Matrix	9 7
[표 23] 지속하락/하락 후 원복 연관 단어와 기사 예시	1 0 0
[표 24] 성능 향상을 위한 추가 실험 계획.....	1 0 1
[표 25] 모델에 대한 성능 관련 가설 검증 결과 요약	1 0 3

그림차례

[그림 1] 의도적인 뉴스와 주가 하락 후 반등 모습 – 제약사 사례	3
[그림 2] 악재성 뉴스 사례와 주가 변화.....	5
[그림 3] 자연어 처리 시작의 역사와 기존 연구 대상	8
[그림 4] 자연어 처리 임베딩 기법의 종류	9
[그림 5] 주식시장 감성사전 예시	15
[그림 6] 자연어처리 프로세스 비교.....	25
[그림 7] Mecab으로 토큰화한 결과 예시	26
[그림 8] Term Frequency로 Vector Representation 사례.....	27
[그림 9] Word2Vec를 활용한 임베딩 및 적용사례	29
[그림 10] RNN를 활용한 NLM(Neural Language Model)	35
[그림 11] RNN를 활용한 텍스트 기반 분류	36
[그림 12] Seq2Seq모델	38
[그림 13] attention 언어모델 구조.....	40
[그림 14] 임베딩 및 언어모델의 발전 요약.....	44
[그림 15] 기존 연구들에 대한 개요 및 고찰 필요 사항.....	47
[그림 16] 본 연구의 방향 개요	48
[그림 17] 제안된 주가추의 예측 NLP 개선 모델.....	50
[그림 18] 악의적 기사와 악재성 기사 예시	54
[그림 19] 연구 프로세스	55
[그림 20] 크롤링 예시(크롤링 코드, 종목주가, 기사)	59
[그림 21] 주가와 기사 통합 (merge활용)	60
[그림 22] 데이터셋 1번타입(기사전문포함) 예시	62
[그림 23] 데이터셋 2번 타입(타이틀만) 예시	63
[그림 24] 실험환경 (구글 코랩 파이썬 코드 및 Training 환경)	63
[그림 25] 전처리 코드 예시.....	65
[그림 26] 전처리 결과 예시.....	65

[그림 27] 학습데이터셋과 Validation셋 분리	6	5
[그림 28] Word2Vec 기존 학습된 모델 활용 임베딩 코드	6	6
[그림 29] WordPiece Tokenize 및 임베딩 코드	6	6
[그림 30] BERT 모델생성 및 학습을 위한 코드.....	6	7
[그림 31] Word2Vec 기반 LSTM 모델생성 및 학습	6	7
[그림 32] Word2Vec기반 LSTM 테스트셋 평가 코드	6	8
[그림 33] Transformer 기반 BERT 테스트셋 평가 코드	6	8
[그림 34] KoBERT 실험 모델 Summary(Class 2개)	6	9
[그림 35] 실험에 활용된 Word2Vec기반 LSTM 모델 Summary....	7	0
[그림 36] KoBERT 실험 모델 (Class 3개)	7	2
[그림 37] Multi-Class로 할 경우 변경된 코드 (LSTM Case)	7	3
[그림 38] Multi-Class로 할 경우 변경된 코드 (KoBERT Case)	7	3
[그림 39] 실험을 위한 데이터셋.....	7	4
[그림 40] 파이썬 코드 파일들	7	5
[그림 41] 학습 모습 (KoBERT Full데이터 활용시)	7	7
[그림 42] 모델별 평균 학습시간.....	7	8
[그림 43] 모델 및 데이터 유형 별 Testing 결과 정확도	8	0
[그림 44] Epoch별 Training/Validation Accuracy의 변화.....	8	2
[그림 45] 각 모델별 정확도.....	8	3
[그림 46] KoBERT Accuracy에 대한 Cross Validation 통계(30회)		
.....	8	4
[그림 47] KoBERT Accuracy에 대한 QQ-Plot 과 정규성검정 결과		
.....	8	5
[그림 48] KoBERT의 Accuracy에 대한 1-sample t-test 결과.....	8	5
[그림 49] 언어모델별 실험결과 성능 비교.....	8	7
[그림 50] LSTM 및 KoBERT 정확도 데이터셋에 대한 정규성 검증	8	9
[그림 51] LSTM과 KoBERT모델 간 2-Sample t-Test결과	9	0

[그림 52] Epoch 수에 따른 Validation 정확도(좌부터 타이틀만, 타이틀+언론사, F타이틀+언론사+기사)	9 2
[그림 53] KoBERT의 데이터셋의 유형에 따른 Accuracy의 변화..	9 3
[그림 54] 각 데이터셋에 대한 학습후의 정규성 검정 결과	9 3
[그림 55] 데이터셋에 따른 ANOVA 분석 결과	9 4
[그림 56] Feature 추가 방법 및 예시	9 5
[그림 57] 하락후 원복, 지속하락에 대한 Test Accuracy.....	9 6
[그림 58] KoBERT – 3 Classes 모델 Epoch 별 Accuracy.....	9 7
[그림 59] 3 Class KoBERT Accuracy 분포 – 정규성검정.....	9 8
[그림 60] KoBERT 3가지 클래스에 대한 유의성 검정 결과	9 9
[그림 61] 정규식 실행시에 Accuracy의 변화.....	1 0 2
[그림 62] Epoch에 따른 Accuracy의 변화.....	1 0 2
[그림 63] Batch Size 변화에 따른 Accuracy의 변화.....	1 0 3

1. 서 론

1.1. 연구의 필요성

2021년은 국내 주식시장 역사상 매우 의미 있는 해가 되었다. 바로 코스피 3,000, 코스닥도 1,000 이상을 넘었고, 개인 투자자와 개인 투자 규모 역시 역대 최대가 되었다. 지금까지 40대~50대 주류였던 주식 투자 연령대가 20~30대까지 확대되어 젊은 층의 투자 비율이 급격히 증가한 것도 의미있는 일이 되었다. MZ 세대로 불리우는 이 세대의 특징은 인터넷을 통해 정보를 획득한다는 점이다. 주식 투자 역시, 각종 SNS와 인터넷 뉴스 등 실시간으로 확보된 정보를 기반으로 주식투자를 하고 있다는 점에 주목해야 한다. 이들 MZ세대 뿐 아니라, 이제는 모든 연령층에서 모바일 트레이딩을 통한 투자가 일반화되었다. 차트분석이나 기업재무성과 같은 기술적인 분석과 애널리스트의 추천 분석보고서보다는 유튜브, 카페, 주식게시판, 실시간 종목 뉴스 같은 소셜미디어를 기반으로 Opinion Mining 같은 과학기술적 분석을 통해 투자하는 경향이 뚜렷해졌다[1].

또한 2020년 공매도 규제 이후, 다시 공매도가 부활하는 2021년에는 악재에 대해 더욱 민감해지고 있다. 공매도는 실제 주식 없이 주식을 파는 행위로써 주식을 실제 가지고 있지 않아도 공매도 시점의 가격과 하락된 이후의 차익을 가지고 이득을 얻는 거래 방법이다. 결국, 주식이

하락하면 이득을 보는 형태이다[2].

무엇보다 주가 하락은 악재성 뉴스에 매우 민감하다. 악재성 뉴스로 인해, 당일 또는 익일 주가하락에 영향을 줄 수 있다. 그러나 이러한 악재성 뉴스에도 몇 가지 패턴이 존재한다. 주가 하락을 유도하기 위해 퍼블리싱하는 의도적인 뉴스와 실제 악재성 공시와 연결된 뉴스가 대표적이다.

보통 의도적인 뉴스의 경우, 확인되지 않는 정보로써 주가가 떨어졌다가 다시 원복하는 경우가 많고, 악재성 공시 뉴스는 지속 하락하는 경우가 많다. 일반 투자자들은 의도적인 뉴스와 사실인 악재성 뉴스를 구분하기 어렵다. 임의적인 악의적 뉴스는 실제 공매도 세력들과 결탁하는 경우도 많다. 이러한 악의적 뉴스로 발생하는 피해 즉 주가 변동에 의한 개인투자자의 피해는 사회적 문제로 큰 이슈로 등장하고 있다.

[단독] 에이치엘비, FDA 임상 결과 허위공시 혐의…지트리비앤티 檢 수사

5 | 2021.02.16 09:30

글꼴 - +

2019년 FDA의 3상 결과 판정 자의적 해석

지난해 11월 금융위 자조심 심의…증선위 결정 남아

금융당국이 국내 바이오·제약 회사의 미국 임상시험 진행 공시에 대해 칼을 빼들었다. 에이치엘비(028300)가 지난 2019년 자사 항암 치료제의 미국 내 3상 시험 결과를 자의적으로 해석해 허위공시한 혐의에 대해 금융위원회 자본시장조사심의위원회(자조심) 심의를 마치고, 증권선물위원회(증선위) 조치를 앞두고 있다. 지트리비앤티(115450)의 경우 이미 지난해 상반기 증선위에서 검찰 고발키로 의결하고 현재 서울 남부지검에서 수사가 진행되고 있다.

일별 시세

날짜	종가	전일비	시가	고가	저가	거래량
2021.03.02	71,900	▲ 3,300	70,500	72,700	69,500	1,334,456
2021.02.26	68,600	▲ 5,500	75,100	75,400	66,400	3,930,082
2021.02.25	63,100	▲ 400	63,700	64,000	61,100	949,642
2021.02.24	62,700	▼ 2,400	64,900	66,700	62,700	1,122,927
2021.02.23	65,100	▼ 4,800	67,600	68,800	64,900	1,295,178
2021.02.22	69,900	▲ 1,100	71,900	73,400	69,000	2,502,335
2021.02.19	68,800	▲ 8,200	62,400	70,800	61,000	6,635,589
2021.02.18	60,600	▼ 1,900	62,500	64,500	59,800	4,271,216
2021.02.17	62,500	▼ 4,000	65,100	71,000	61,200	7,992,413
2021.02.16	66,500	▼ 24,900	91,400	91,700	64,000	30,900,837

[그림 1] 의도적인 뉴스와 주가 하락 후 반등 모습 – 제약사 사례

이러한 피해를 줄일려면 SNS 상에 올라온 뉴스가 악의적 뉴스인지, 실제 악재성 공시로 이어지는 뉴스인지 파악할 수만 있다면 가능한 것이다. 공매도가 개인까지 확대되는 시점에 공매도 세력들의 악의적 뉴스 전파는 이전보다 증가할 것이고 이로 인한 개인투자자의 피해도 증가될 것으로 예상된다. 이전 연구에서는 공매도를 통한 수익을 얻기

위한 유형이 3 가지이며, 그 중 하나가 악의적인 뉴스를 배포하는 것으로 제시하고 있다[2]. 특정 종목에 대한 공매도 이후 기업의 이익감소 예상 리포트, 증권사의 투자의견하향 보고서, 채권등급 하락 또는 기업등급 가능성, 그리고 대규모 일반 투자자 대상 유상증자 가능성, 대표이사의 배임, 상폐 가능성 등 경영상의 문제들을 제시하고 있다. 만약 확인되지 않은 이러한 뉴스들은 악재성 정보로써 공매도 이후 주가 하락을 조장할 수 있을 것이다.

결국, 악의적 뉴스를 접하게 되면 향후 주가에 영향을 준다. 보통 공매도 후 뉴스를 내고, 이에 따라 이득을 얻는 과정인데, 공매도 세력과 특정 언론사, 특정 기자와도 관련되어 있다고 가정하기도 한다. 본 연구에서는 뉴스 기사와 주가하락과의 관련성을 Transformer 기반의 BERT 알고리즘을 통해 악재성 뉴스가 나왔을 때, 과연 악의적인 의도를 가지고 쓴 뉴스인지, 공시성, 즉, 단순 악재 뉴스인지 구분하는 모델을 개발하고자 한다. 가설로는 악의적 의도를 가지고 쓴 뉴스의 경우에는 단기적으로는 하락했다가 다시 제자리로 돌아오는 경우가 있지만, 악재성 뉴스의 경우에는 지속적으로 하락하는 경우가 있을 것이라는 가정으로 연구를 진행하였다.

헬릭스미스, 2817억원 규모 유상증자 결정

이데일리 | 2020.09.17 18:15 글꼴 - +

[이데일리 권효중 기자] 헬릭스미스(084990)는 시설자금, 운영자금, 채무상환자금 등을 위해 약 2817억원 규모의 주주배정 후 실권주 일반공모 방식 유상증자를 결정했다고 17일 공시했다.

신주의 예정 발행가는 3만8150원이며, 주당 약 0.28주를 배정하는 방식으로 이뤄진다. 이에 증자 전 발행주식 총수 대비 약 28%에 해당하는 750만주가 새로 발행된다.

일별 시세						
날짜	종가	전일비	시가	고가	저가	거래량
2020.09.29	35,450	▲ 2,600	33,700	35,450	33,500	956,053
2020.09.28	32,850	▲ 1,450	32,000	34,700	31,500	970,121
2020.09.25	31,400	▲ 350	31,800	33,000	30,750	782,174
2020.09.24	31,050	▼ 2,450	32,350	35,450	30,200	3,031,905
2020.09.23	33,500	▼ 2,500	36,600	37,000	33,200	1,517,706
2020.09.22	36,000	▼ 900	36,950	38,100	35,800	1,649,521
2020.09.21	36,900	▼ 4,900	39,950	40,400	36,800	3,391,881
2020.09.18	41,800	▼ 10,400	42,000	44,450	41,450	4,640,031
2020.09.17	52,200	▼ 2,900	54,400	54,700	52,200	647,930
2020.09.16	55,100	▲ 3,900	52,200	60,200	51,100	4,757,429

[그림 2] 악재성 뉴스 사례와 주가 변화

본 연구에서는 공매도 거래량이 있는 611 개의 종목에 대해 1 년간의 종목별 뉴스를 가지고 Neural Language Model 를 적용하여 예측 모델을 구현하였다. 고의적인 악성 뉴스와 실제 공시성 뉴스를 사전에 구분할 수 있다면, 보유한 종목에 대해 매도할지 판단할 때 다시 한번 생각하여, 주식시장의 신뢰성과 안정성에 기여하는데 도움이 될 수 있으리라 판단된다.

1.2. 연구목적과 연구 범위

최근 주식시장에서 개인 투자자들은 주식 투자 시 정형 데이터 보다는 비정형 데이터 기반으로 투자하는 경향이 뚜렷하다. 다수의 기존 연구들은 인터넷상의 비정형 빅데이터를 활용하여 주가의 방향을 예측하는 머신러닝 중심의 모델들을 연구하였다. SNS 와 뉴스 기반으로 감성분석, 즉, 해당 종목의 뉴스 또는 글들이 긍정인지, 부정인지 분석하여, 단기적으로 주가의 방향이 상승할 것인지, 하락할 것인지 예측하는 것이다.

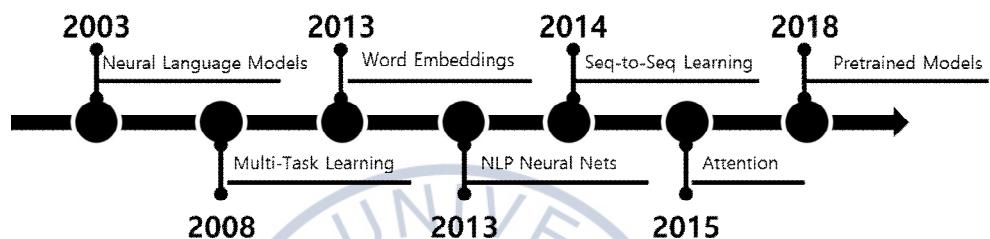
이밖에도, 검색 포털의 해당 기업 검색량 기반으로 주가의 변동성을 예측하는 연구를 통해 비정상적인 검색량은 주가 변동성에 큰 영향을 줄 수 있다는 결과를 밝히고 있다. 그리고 경제신문에 자주 언급되는 단어에 대한 검색량과 주가지수의 관계를 분석한 연구를 통해 개인투자자의 관심이 검색량으로 표출됨을 시사하고 있다[3][4]. 기업명으로 직접 검색한다는 의미는 이미 개인투자자가 그 업체명을 알고 있을 때이다. 즉 급증하는 비정상적인 검색량은 호재 또는 악재가 노출된 순간이다.

뉴스의 감성 분석 역시, 이미 노출된 시그널이다. 주가에 이미 선반영 되거나, 즉각 반영이 된다. 다수의 연구에서는 뉴스 다음날 또는 장중 뉴스 나온 직후의 주가와 비교하고 있다[5][6]. 장마감 이후 나온

뉴스는 다음날 장이 열리자 마자 반영될 것이며, 장 중에 나온 뉴스는 실시간 주가에 반영되므로 분명히 투자 타이밍 선정에 어려움이 있다. 초단기 투자자(데이트레이더) 이외에는 투자자 입장에서 기존의 연구 방법으로 종목을 선정하여 투자하기에는 즉각적 대응이 어렵다. 특히 급등하는 검색량 이후 반영되는 주가의 변동율은 수 주일 후 다시 제자리로 찾아가는 주가 반전현상도 있음을 감안해야 한다. 투자 종목을 탐색하기 위해, 뉴스 키워드 관련 종목을 찾거나 개별 기업과 유사한 기업을 찾기 위한 연구로써 주요 키워드 및 키워드 관련 종목을 탐색하여 함께 제시하는 모델도 연구되었다[11]. 즉, 뉴스에 의해 A라는 업체의 주가가 변동하면, B라는 업체의 주가도 크게 변동할 것이라는 가설을 검증한 것이다.

기존 대다수의 연구에서는 뉴스 또는 SNS를 통해 대중들의 반응을 크롤링한 후, 각 문서의 감성이 긍정적인지, 부정적인지를 통계적 분석 기반인 TF-IDF 분석 등의 통계적 NLP(자연어처리) 방법을 사용한 연구를 기반으로 하고 있다. 주식의 방향성을 예측하기 위해, 뉴스 별 감성을 분석하고, 그 값을 수치화 한 후 수치화한 내용을 기반으로 다시 향후 주식이 어떤 변화가 있을 것인지에 대한 예측하는 로직이다. 즉, Feature로써 뉴스들의 감성 점수를 활용하고 Label로써 당일 또는 익일 주식 상승 여부로 모델링한다. 이때 적용되는 기법은 머신러닝의 경우, Logistic Regression, Support Vector Machine 등이 적용되었으며,

딥러닝을 경우에는 MLP(Multi-layer Perceptron)의 Sigmoid 를 통한 Binary Classification 으로 다소 일반적인 모델로 분석하고 있다. 지금까지 주식분야의 NLP 적용과 관련된 연구에서는 NLP Neural Nets 기술 이전의 언어모델 중심으로 연구가 진행되었다.



[그림 3] 자연어 처리 시작의 역사와 기존 연구 대상

그리고, 대부분 상승여부에 중점을 두고 모델링하였으며, Feature 는 뉴스의 텍스트를 이용해 감성지수화 하는 Feature Engineering 을 실시하였다. 직접 텍스트를 Feature 로 사용하지 않고 수치화하는 복잡한 단계를 거친 것이다. 이렇게 Feature Engineering 하는 이유는 머신러닝의 경우, Feature 에 대한 전처리가 중요하였고, 텍스트를 그대로 임베딩하여 의미있는 문장으로 해석하는데 한계가 있었기 때문이다. 딥러닝 기술이 발전하면서, Text Mining 관점에서도 뉴럴 네트워크를 활용하기 시작했다. 대표적인 기술이 RNN 과 CNN 을 적용한 자연어 처리 기술이다. 이후 딥러닝이 더욱 연구에 박차를 가하면서, 단어 수준의 임베딩이 기본였던 Word2vec 이나 FastText 등에서 문맥을

이해할 수 있는 셀프 어텐션 기반의 트랜스포머 네트워크를 활용한 BERT, GPT 까지, 다양한 기술을 적용하여, 자연어처리 기술은 급속도로 발전하였다. 이러한 딥러닝 방식은 기존의 전처리 중심이었던 머신러닝에서, 전처리 없이 바로 Label 를 예측할 수 있는 방안을 마련할 수 있게 되었다. 그리고 무엇보다도 Pre-Trained Model 로써 이미 대량의 언어를 학습하여 Transfer Learning 관점에서 활용하기 때문에 임베딩 벡터의 정확도가 매우 높다는 평가를 받고 있다. 기존의 Word2Vec 이나 FastText 는 별도의 대량의 말뭉치들을 모아서 학습하여 만들지만, Pre Trained Model 의 경우, 대량의 말뭉치로 이미 학습된 모델을 가지고 추가 Fine Tuning 함으로써 자연어에 대한 임베딩이 더욱 정밀하게 되었다.

임베딩기술	분류	시기
TF-IDF	통계기반 임베딩, 단어수준임베딩	
LSA	통계기반임베딩, 단어수준임베딩	2000년대
LDA	통계기반 임베딩, 문장수준임베딩	
Word2Vec	머신러닝기반, 단어수준임베딩	2013
Doc2Vec	Neural Net기반, 문장수준임베딩	2014
FastText	머신러닝기반, 단어수준임베딩	2017
ELMO	Neural Net기반, 문장수준임베딩	2018
BERT, GPT	Neural Net 기반, Pre-Trained, 문장수준	2018

[그림 4] 자연어 처리 임베딩 기법의 종류

본 연구에서는 기존에 뉴스 기사의 감성분류 후 감성 분류된 지표를 기반으로 주가의 방향을 예측하는 수치적인 방식(Feature=감성점수)이 아닌, 주가에 영향을 주는 텍스트를 직접 입력(Feature=뉴스기사)하여, 감성 분석 없이 바로 기사의 문맥을 이해할 수 있는 딥러닝 방식을 통해 관련 종목 기사의 문맥을 보고, 주가의 향방, 특히 악재로써 작용할 만한 기사인지를 판단하는 모델을 개발하고 LSTM 방식의 단점을 보완한 Transformer 방식의 BERT, 그리고 BERT 의 한국어 모델인 KoBERT 를 사용했을 때 유의성 성능 개선여부를 확인하기 위해 실험을 실시하였다. 그리고 상승여부에 대한 예측보다는 하락에 대한 예측을 통해 공매도가 확대되는 시점에 투자자들의 의사결정에 도움을 주고자 하였다. 악재 기사가 나왔을 때 지속적으로 보유해야 할지, 아니면 매도해야 할지에 대한 의사결정을 위한 모델로써 활용 가능한지도 함께 검증하고, 새로운 Feature 를 추가할 때, 텍스트에 포함시켜 학습해도 반영이 되는지를 검증함으로써, 향후 텍스트 기반의 딥러닝시, 수월한 Feature Engineering 방식도 함께 검증한다.

[표 1] 연구 범위 및 목적

	기존연구	본 연구
연구 목적	-뉴스 감성점수를 기반으로 주가의 상승여부 예측으로 매수 판단에 도움주고자 함	-뉴스 텍스트 기반으로 주가의 하락여부 사전예측, 하락 후 원복 가능성 사전 예측으로 매도 판단에 도움주고자 함
연구	-분석대상: 뉴스, SNS글	-분석대상: 뉴스

범위	-Features: 감성점수 -Label: 상승여부(1) -사용기술: TF-IDF, 머신러닝(SVM, Logistic Regression 등)	-Feature: 뉴스글, 기자, 언론사 -Label: 하락여부, 하락후원복 -사용기술: BERT, KoBERT등 트랜스포머기반 딥러닝 기술, Word2Vec기반 LSTM 기술
연구 방법	-데이터 수집: 크롤링 -모델링: R, 파이썬등	-데이터 수집: 크롤링 -모델링: 파이썬, 텐서플로우, 파이토치 활용 코딩
최종 산출물	-상승 예측 모델	-지속하락 및 하락 후 원복 예측모델 -텍스트에 Feature 추가 방법

요컨데, 뉴스 기사 중에 주가의 하락에 영향을 주는 기사들을 BERT 와 RNN 등의 딥러닝을 통해 학습시키기 위해, 기사를 있는 그대로 입력함으로써 감정분석 단계없이 딥러닝에서 스스로 Feature 를 생성하도록 함으로써 지속 하락할지, 하락 후 원복할지를 예측하는 모델을 파이썬을 통해 구축하고, 추가적인 Feature(특히, Categorical Data)을 쉽게 추가할 수 있는 방법을 제안한다.

2. 이론적 배경

2.1. NLP(Natural language Process)을 활용한 주가 예측에 대한 연구

일반적인 주가예측 모델로는 주가 히스토리 데이터 활용을 통한 시계열 차트 분석과 거시적 경제 상황, 환율, 금리 등 경제지수 관련 수치데이터를 활용해 주가를 예측하는 모델이 대표적이다. 그러나 2010년 이후 SNS와 빅데이터 기술이 급속도로 발전하면서 주식은 대중의 심리에 따라 변동성을 갖는다는 가설을 가지고, 소셜미디어에 기술(記述)된 텍스트를 기반으로 주가를 예측하고자 하는 연구가 시작되었다. 증권사, 투자사도 이제 수치데이터 뿐 아니라, 소셜미디어의 글들을 재료로 투자에 활용하고 있는 것으로 알려져 있다. 대표적으로 골드만 삭스는 주식 투자를 위해 각종 금융 사건들과 수치적 자료, 그리고 여기에 온라인에 떠돌고 있는 텍스트를 분석하여 5분만에 의사결정이 가능한 인공지능 Kensho를 개발하여 실전에 활용하는 것으로 알려졌다[12].

소셜미디어 텍스트 분석을 통한 주가예측 관련 연구로는 크게 2가지 축으로 연구가 선행 진행되었다. <소재의 종류> 축과 <예측 방법> 축으로 구분된다. 예측 소재로는 뉴스와 카페 같은 소셜미디어로 구분된다. 그리고 예측방법으로는 뉴스/미디어 텍스트의 감성분석을 통한

수치를 가지고 머신러닝기반 예측하는 방법과 자연어처리를 통해 딥러닝하는 방법으로 구분될 수 있다.

[표 2] NLP 기반 주식 예측 연구 분야 및 연구 논문 예시

	뉴스	SNS
머신러닝	산업군내 동질성을 고려한 온라인 뉴스 기반 주가예측[13]	SNS 와 뉴스기사의 감성분석과 기계학습을 이용한 주가예측 모형 비교 연구[5]
딥러닝	BERT 를 활용한 뉴스 감성분석과 거시경제지표 조합을 이용한 주가지수 예측[14]	SNS 감성 분석을 이용한 주가 방향성 예측: 네이버 주식 토론방 데이터를 이용하여[10]

표에서 보듯이, 최신의 Neural Language Model 인 BERT 를 적용한 연구가 있었으나, 이 역시, 뉴스를 기반으로 감성점수를 도출하는 데 BERT 를 사용하였고 직접적으로 텍스트를 활용하지는 않았다. 전반적으로 Neural Language Model 기반 주가 예측 연구 논문은 위에 언급된 형태의 Feature Engineering 중심(감성점수화), 상승여부 중심, 기사 중심으로 연구가 진행되었다. NLP 기반 주가 예측 관련 연구들의 흐름을 정리하자면 다음과 같다.

[표 3] NLP 기반 주가 예측 관련 연구 단계

단계	연구주제	주요 적용 기술
1 단계	주가예측을 위한 뉴스의 감성사전화 연구	회귀분석, TF-IDF
2 단계	뉴스기사 감성지수화 및 분석 통한 주가 예측 연구	SVM, CNN, RNN
3 단계	뉴스기사 외 연관 종목 기사, SNS 등의 추가 Feature에 대한 연구	코사인유사도, 군집분석, Word2Vec

초기의 연구에서는 뉴스를 기반으로, 그리고 개별 종목이 아닌, 종합주가지수를 대상으로 연구하였다. 뉴스와 주가 빅데이터 감성분석을 통한 지능형 투자 의사결정 모형 연구에 의하면 주식시장 개장 전에 각종 시황 및 전망, 그리고 해외뉴스 등의 감성은 주가하락시 70%, 상승시에는 78.8%의 예측 정확도를 가지는 것으로 연구되었다[6]. 이 때 활용된 Feature로는 역시 뉴스 텍스트의 감성으로, NP Dictionary 기반으로 긍정/부정 단어의 비율을 분석한 후 긍정 단어수의 비율이 52%이면 긍정 뉴스, 48~52% 이면 중립, 48% 이하이면 부정으로 분류하였다. 그리고 이를 활용하여 하루 동안 수집된 뉴스들의 긍정 비율을 가지고 지표화한 후 당일 종합 주가 상승 여부를 Label로 설정하여 통계적 회귀분석을 통해 모델링하고 예측하였다.

이후에는 감성분석을 개별 종목단위로 예측모델을 구축하였으며, 무엇보다 뉴스의 긍정/부정을 판단하는 단어들의 집합인 감성사전이 예측에 중요하다고 판단되어, 이를 구축하는 연구들이 본격화되었다.

주가지수 방향성 예측을 위한 도메인 맞춤형 감성사전 구축방안 연구에서는 팟스넷의 게시글을 활용하여 소셜미디어가 표출하고 있는 부정, 긍정 등의 주관적인 감성을 파악하기 위해 증권 도메인에 특화된 감성 사전을 구축하였다[7]. 감성사전에는 일반적으로 범용적인 감성사전과 도메인에 특화된 감성사전으로 구분된다. 텍스트를 통해 단순한 '긍정', '부정'의 감성뿐 아니라 사건에 대한 긍정/부정도 구분하기 위해 맞춤형 감성사전이 필요하다는 것을 이전부터 주장 되어졌다[8][9]. 해당 연구는 증권 뉴스를 기반으로 형태소 분리 후 단어별로 긍정/부정의 극성을 모두 라벨링하여 감성사전을 구축한 연구로 아래 그림과 같은 형태의 감성사전을 구축하였다.

감성사전	감성
소비 기대 못미치	부정
소비 증가 안	부정
소비 개선 안	부정
소비 회복 안	부정
소비 저조	부정
소비 나빠지	부정
소비 좋아지	긍정
...	...

[그림 5] 주식시장 감성사전 예시

이러한 증권 특화된 감성 사전이 필요했던 이유는 뉴스나 게시글의 감성이 긍정인지, 부정인지를 수치화 하여 주가와의 관련성을 보기 위함이었다. 일종의 Feature Extraction 의 일환인 것이다. 그러나 최근에는 딥러닝이 적용되고, Pre-trained Model 신기술들이 등장하면서 Feature 로써 감성사전을 활용할 필요성이 있는지는 한번 고려해볼 필요가 있다. 주가의 감성사전 구축 관련 연구 이후에는 감성사전을 활용하여 뉴스기사, SNS 글의 감성을 분석한 후 주가의 향방을 예측하는 연구가 본격적으로 진행되었다. 김동영외 2인(2014) 연구에서는 일정기간 동안 특정 기업의 뉴스 기사를 수집하여 뉴스 기사기반의 증권 도메인에 맞는 감성 사전을 구축하고, 감성분석을 통해 확보한 수치데이터를 기반으로 기계학습으로 주가를 예측하는 연구로써 대표적이다[5]. 데이터 수집을 위해 특정 검색 사이트에서 종목 키워드 서치 후 결과물에 대해 크롤링하여 수집하였고, 각각의 기사에 대해서는 꼬꼬마 형태소 분석기를 통해 토큰나이즈 후, 명사만 추출하여 이를 가지고 분석에 활용하였다. 뉴스와 당일의 주가를 확인한 후 주가와 관련성 높은 단어를 찾아, 빈도수와 그 단어가 긍정인지 부정인지 직접 판단하면서 인덱싱하여 감성사전을 구축했다. 그만큼 예측을 위한 사전작업이 많다는 의미이다. 이때 단어의 긍정지수는 기사가 게재된 후 익일 주가가 상승할 경우를 감안하여 지수를 개발했다. 해당 종목의 일별 감성분석을 위해 수많은 기사의 긍정지수(긍정단어 빈도)를 구해

산술평균으로 일별 긍정지수를 수치화하고, 이 수치를 활용하여, 익일 주가가 상승했는지에 대한 여부를 Label 값으로 Binary Classification 형태로 logistic Regression, SVM 등의 Machine Learning 을 실시하였다. 7 개 코스피 기업 중심으로 뉴스데이터와 SNS(트윗) 데이터를 모두 활용한 결과, 0.8 정도의 정확도를 보였다. 본 연구 결과는 텍스트 데이터, 즉, 비정형 데이터를 활용하여 수치화하고, 이를 머신러닝을 활용하여 분석한 모델로써 의미가 있다.

이후, 뉴스기사 뿐 아니라, 일반적인 투자자들이 많이 참조하는 네이버 주식 토론방의 게시글들과 딥러닝 모델을 활용해 주가를 예측하는 연구들이 진행되었다. 뉴스의 경우에는 전문가 중심의 정형화되고 다듬어진 텍스트였지만 이러한 SNS 게시판에는 다양한 비전문가들의 소리와 “ㅋㅋㅋ”, “ㅠㅠ” 같은 이모티콘 및 감성을 표현한 다양한 문자들이 포함되어 있어, 분석을 위해 전처리가 매우 중요하였으며, 뉴스보다 엄청 많은 양이 생성되어 빠르고 전파되기 때문에, Label 에 대한 정의도 익일이 아닌 초단기성으로 고려되어야 한다.

김명진 외 3인(2020) 연구에 의하면 KOSPI 200 에 포함되어 있는 20 개 종목의 네이버 주식토론방 글들을 수집하여 주가의 움직임을 예측하는 연구를 실행하였다[10]. 이전 연구와 다른 점은 자연어처리의 딥러닝 분야인 RNN 모델의 LSTM 과 CNN 모델 2 가지 알고리즘을 사용하여 비교 분석했다는 점이다. 레이블로는 기존의 뉴스 기사 발생일

대비 익일 종사의 변동여부가 아닌, 1 시간 단위의 변동성으로 정의하여 조사하였다. 게시판 글에 대한 전처리 과정으로, Ranks NL 을 통한 불용어 제거와 Okt(Open Korean Text) 형태소 분석을 통해 진행하였으며, Label 값은 시간단위로 <증가> 또는 <보합/하락>과 <변동폭> 2 가지로 측정하였다. LSTM으로 학습 시 80%는 학습, 20%는 테스트로 활용하여, 결과적으로 50% 이상의 예측 정확도를 보인 것으로 조사되었다. 그리고 최종적으로 LSTM 알고리즘의 성능이 높은 것으로 파악되었다. 본 연구는 이제 감성지수를 산출한후 수치화 된 Feature 가 아닌 SNS 의 말뭉치 그대로 활용해 주가의 방향을 예측한 것에 대해 의미가 있다. 이후 Feature 를 단지 해당 종목의 기사뿐 아니라 다른 추가적인 Feature 들을 추가하여 주가의 방향을 예측하는 연구들이 시도되었다. 꽝란외 3 인(2018) 연구는 뉴스기사를 분석하여 주가의 방향과 관련 높은 키워드와 관련기업들을 탐색할 수 있는 키워드 및 연관 종목 탐색 시스템에 대한 연구이다[11]. 뉴스 기사에서 종목별 유의미한 키워드와 관련종목, 유사 키워드들을 도출한 후, 이를 기반으로 주가변이 추이와 키워드 발생 빈도 등을 종합적으로 분석하여 종목별 상승과 하락 관련 키워드, 그리고 키워드별 상승과 하락 연관 종목들을 도출하고 관리하는 지능적 이슈 트래킹 시스템이다. 종목 뉴스 8 만건을 활용하여 TF-IDF, Word2Vec, 코사인 유사도와 같은 전통적인 NLP 기법을 적용하여 결과를 도출하였다. 관련 키워드 도출에 매우 적합한

Word2Vec 을 통해 유사도 높은 종목 도출을 시도하였으며 TF-IDF 와 코사인유사도 분석을 통해 종목별 뉴스에서 키워드를 추출하는데 활용했다. 그리고 LDA(Latent Dirichlet Allocation)를 통한 주가의 움직임과 관련성 높은 토픽 도출 및 단어 도출에 활용했다. 이 연구는 해당 종목의 뉴스와 SNS 글만으로 주가의 향방을 예측하는 것이 아니라, 관련된 종목을 탐색하고 관련 종목의 글도 함께 Feature 로 추가하여 주가 예측에 NLP 를 적용한 것에 의의를 가지고 있다. 이러한 방향에 입각하여 최근의 연구는 주가의 향방을 예측하는데 있어 단일 종목의 기사뿐 아니라, 관련된 종목을 찾고, 해당 종목의 기사까지도 크롤링하여, 이를 학습하고 주가의 방향을 예측하는 모델 개발로 확대되고 있다. 성노윤 외 1인(2019) 연구에 의하면, 해당종목의 기사뿐 아니라 관련된 종목의 기사도 주가에 영향을 준다는 결과가 도출되었다[13]. 이 연구에서는 k 평균 군집 분석을 통해 관련성 높은 기업들을 군집하였고, 특정 기업의 기사와 더불어 관련 종목의 기사까지도 크롤링하여 인공지능 주가 예측 모델에 활용하였다. 주가 예측을 위해, 딥러닝보다는 주가에 영향을 주는 단어들이 많은 뉴스에 가치를 줄 수 있도록 단어주머니 방식(Bag of Words)과 TF-IDF 를 활용하여 Feature 화 하였다. Label 은 역시 주가의 변동 여부로 레이블링하였고, 학습 방법으로 기계학습을 통해 예측하였다. 추가적으로 하이퍼파라미터를 위해 그리드 서치를 적용하여 최종 결과를 도출한 연구이다. 본 연구결과로 30 여개

종목 대상 3년간 기사 기반으로 모델링 결과 Accuracy는 0.658로 이전 유사 실험보다 다소 향상된 것으로 조사되었다.

가장 최신의 자연어 처리 기술인 BERT를 활용하는 연구도 진행되었다. 장은하와 2인(2020) 연구에서는 BERT와 NLTK VADER로 감성 분석하여 기사의 감성점수를 도출하고, 환율, 금값, 오일 등의 거시적 지표를 통합하여 LSTM을 활용하여 시계열 딥러닝을 통한 다음존스 지수를 예측하는 연구였다[14]. 그러나, 한국어의 경우, BERT나 NLTK VADER로써 성능을 발휘하기에는 다소 부족할 수 있지만, 가장 최신의 언어모델인 Transformer 기반 BERT를 적용하여 NLP 추가예측 모델로 의미가 있다. 지금까지 기사 또는 소셜미디어의 텍스트 정보를 통해 주식 예측 관련 연구들을 살펴보면 다음과 같은 공통적인 특징이 있다.

첫번째 특징은 텍스트 긍정/부정 감성 분석을 통한 감성점수를 입력 데이터로 활용하고 있다. 즉, 텍스트를 감성점수화 하여 수치화한 값을 활용하고 있다. 이유는 머신러닝이나 딥러닝 활용 시, 텍스트 데이터를 직접 임베딩할 수 있는 기술과 성능이 한계가 있었기 때문으로 풀이된다.

두번째 특징은 감성 분석을 위해, 규칙기반(TF-IDF, Bag of word 등)과 SVM(Support Vector Machine)등의 머신러닝 중심으로 연구가 진행되었다. RNN 및 Transformer 등의 딥러닝 기반 연구와 Pre-Trained 된 BERT 등의 모델을 활용하는 연구가 부족하다. 이유는

해당 기술이 등장한지 얼마 안되고, 특히 영어 중심으로 API 제공도 하고 있으나, 한국어 중심으로 개발된 API는 아직 미비하기 때문으로 파악된다.

세번째 특징은 주가 상승에 초점을 두고 있다. 즉, 감성 방향이 긍정에 맞춰져 있다. 지금까지 주가는 상승을 통한 이득을 취하는 것이 초점을 맞추고 있다. 지금까지 연구는 종합주가 지수 1000에서 2000 사이에 이루어 졌기 때문에 매수 타이밍에 대한 의사결정이 매우 중요한 포지션임에 분명하다. 그러나 지수 3000이 넘은 이 시점, 그리고 공매도가 개인까지 확대되는 경우, 하락에 초점을 맞출 필요가 있다. 그리고 많은 사람들이 주식을 보유하기 때문에 매도하는 시점과 매도에 대한 의사결정이 중요해졌기 때문이다.

네번째 특징은 Feature로 활용되는 소재가 뉴스 기사와 SNS 텍스트에 주안점을 두고 있다. 이를 통한 감성 분석을 하기 위함이었다.

[표 4] 기존 연구들에 대한 특징 요약 및 연구 방향

특징	연구방향
텍스트 긍정/부정 감성 분석을 통한 감성점수를 입력 데이터로 활용한 연구	감성점수화 단계 없이 직접 주가 예측할 수 있는 방법 모색
규칙기반과 머신러닝 중심의 연구	다양한 최신의 Neural Language Model 적용 방법 모색
주가 상승 가능성에 대한 연구	하락 가능성과 하락 후 반등에 예측 방법 모색
기사 텍스트만을 활용한 연구	다양한 Feature 들을 임베딩할 수 있는 방법 모색

2.2. 자연어처리 기술에 대한 연구

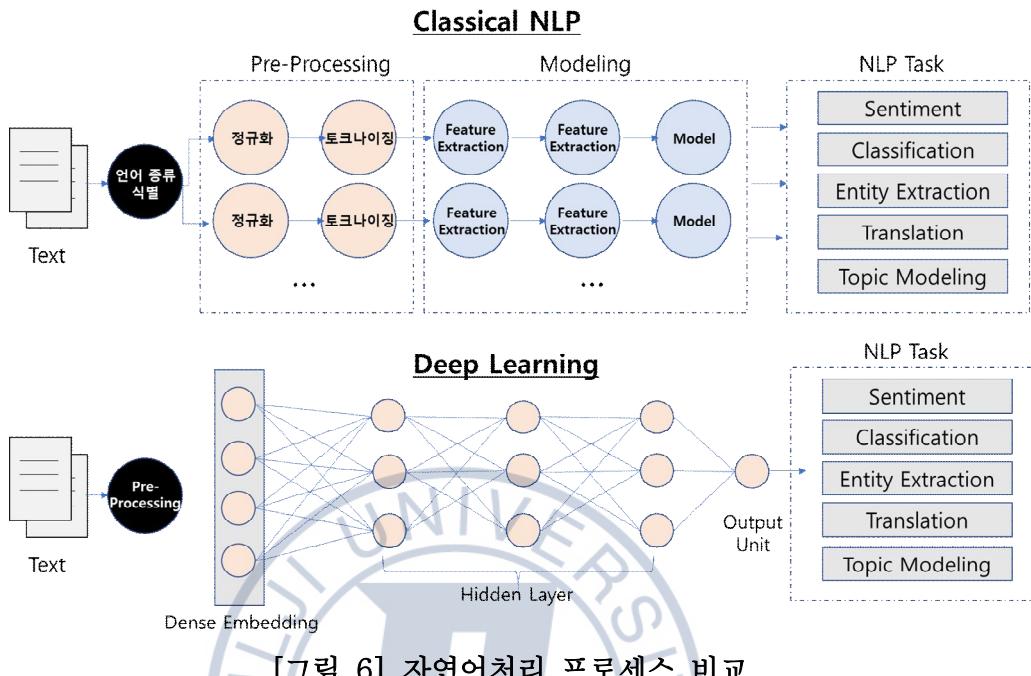
2.2.1. 자연어 처리에 대한 이해

위키백과에 의하면 자연어 처리(Natural Language Process: NLP)란 ‘인간의 언어 현상을 컴퓨터와 같은 기계를 이용하여 묘사할 수 있도록 연구하고 이를 구현하는 인공지능 기술’로 정의되어 진다. 한마디로 사람의 말을 기계가 알아듣고, 다양한 Task 를 기계가 스스로 수행하도록 처리하는 기술이다. 예를 들어 챗봇, 기사나 소설을 스스로 창작하는 인공지능, 특정 웹소설이나 영화의 시놉을 보고 흥행 가능성을 예측하는 인공지능, 보이스봇 등이 NLP 영역에 해당된다. NLP 를 위해서는 가장 중요한 요소가 기계가 사람말을 알아듣도록 언어를 벡터화하는 기술이다. 이 기술을 임베딩(Embedding) 기술이라고 하며 이를 위해 구글과 페이스북등 대기업 중심으로 기술들을 지속 개발하고 있다.

임베딩 기술들은 이전에는 엄청난 양의 단어들을 DB 화하고 그 단어들이 입력되면 어떠한 결과를 출력해야 할지를 미리 룰(Rule)화하는 형태로 개발되어졌다. 이후 TF-IDF 같은 확률/통계적 기반의 임베딩 기술을 거쳐, 인공지능 기술이 발전하면서, Word2Vec, ELMO, BERT 같은 Neural Network 기반의 접근법으로 진화되었다.

언어의 임베딩을 통해 기계가 이해하기 시작하면서 다양한 예측, 분류, 제너레이션등의 다양한 Task 들이 기계를 통해 자동화가 가능해졌다.

인간이 텍스트를 보면서 특정 Task 를 수행하려면 그 텍스트의 의미를 파악해야 한다. 텍스트의 의미를 부여하는 것이 바로 Feature 인데, 이를 Machine Learning 으로 기계가 분석하게 하기 위해서는 Feature Engineering 기술이 필요하다. 그러나, 문장을 그대로 입력하면 기계 스스로 Feature 를 도출해주는 딥러닝 기술이 발전하면서, NLP 영역이 급속도로 발전하게 되었다. 즉, 언어 임베딩 수준이 높아지고, 딥러닝 기술이 발전하면서, 그제서야, 기계가 인간의 말을 알아듣고, 인간이 쓴 글을 이해하여, 예측까지 가능한 시대가 도래한 것이다. 이제는 컴퓨터 스스로 각 단어들의 의미를 이해하고 문장의 문맥을 파악할 수 있는 다양한 기술들이 개발되어지고 있다. 이제 특정 Task 수행을 위한 문장의 특징(Feature)을 찾기 위해 컴퓨터 스스로 Feature Extraction 를 수행하고 스스로 분류할 수 있는 인공지능 기반 언어모델들이 개발되고 있다.



[그림 6] 자연어처리 프로세스 비교

그러나 아직은 기계가 한국어를 이해하기 위해 입력 문장을 클린징하는 전처리 활동이 필요하다. 영어와 달리, 교착어인 한국어는 띄어쓰기, 조사, 특수문자 등에 따라 의미가 달라지므로 전처리가 매우 중요한 요소이다. 전처리란 자연어를 기계가 이해하도록 문장을 정제하는 활동이라 보면 된다. 예를 들어, 개행문자/특수문자/공백/중복표현 등은 컴퓨터가 문맥을 이해하는데 방해가 되므로 이를 제거해주거나, 수정하는 활동이다. 그리고 문장을 기계가 이해할 수 있도록 의미 있는 단위로 분리하는 토큰화라는 단계를 거친다. 특히, 한국어의 경우, 조사, 어미 등을 분리하는 형태소 분석을 통해 토큰화하여, 기계가 좀 더 분명히 이해할 수

있도록 벡터화 직전에 수행한다. 추가적으로 띄어쓰기에 따른 문제도 함께 해결 가능하다.

```
text = "아버지가방에들어가신다"  
  
tokenize("mecab", text)  
  
'아버지 가 방 에 들어가 신다'  
  
tokenizer = get_tokenizer("mecab")  
tokenizer.nouns(text)  
  
['아버지', '방']
```

[그림 7] Mecab으로 토큰화한 결과 예시

이후, 기계가 이해할 수 있도록 수치화, 즉, Term Vector Representation 단계를 거쳐 비로소 딥러닝(RNN, CNN 등)과 머신러닝(SVM, Decision Tree, KNN 등)의 각종 알고리즘이 적용 가능하게 된다. 즉, 텍스트 학습을 통해 감성분류, 구문 분석, 의미 분석, 토크나이징 등이 NLP Task 수행이 가능하게 된다.

2.2.2. Vector Representation

NLP에서 가장 중요한 단계는 기계가 텍스트를 알아듣게 하기 위해 문장을 수치로 바꾸는 Vector Representation이다. 이를 임베딩이라고 불리우며, 다양한 방법들이 존재한다. 문장을 Vector화하는 방법으로

가장 고전적인 방법은 Bag of Words model 이 있다. 이 방법은 모든 입력 문장에서 언급되는 단어 또는 형태소들을 인덱싱한 후, 각 문서에 해당 인덱스(단어/형태소)가 얼마나 발생하는지 빈도수로 채운 것이다. 아래 그림을 보면, Doc1 의 문장에는 “주식”이라는 단어가 4 개, “을”이 1 개, “바이오”가 3 개 포함된 문장이고 Doc2 는 “공매도”가 3 개, “하락”이 2 개 포함된 문장이다. 만약 전체가 5 개의 단어로 이루어 졌다면 5 개의 Dimension 을 가지고 있고, 없는 Dimension 에 대해서는 “0”을 채워 전체를 동일한 차원으로 만든다. 바로 이것이 Vector Representation 이다.

BoW	주식	을	바이오	공매도	하락
Doc1	4	1	3	0	0
Doc2	0	0	0	3	2
Doc1 = [4, 1, 3, 0, 0]					Doc2 = [0,0,0,3,2]

[그림 8] Term Frequency로 Vector Representation 사례

대표적인 방법으로 TF-IDF 라는 방식이 있다. 이 방식은 TF(Term Frequency), 즉, 특정 단어 w 가 문서 d 에 등장한 빈도수를 DF(Document Frequency)를 나눈 값이다. 이 값의 의미는 문서 전체에는 흔하게 등장하지 않고 특정 문서에서만 자주 등장하는 단어의 영향력이라고 정의하고 있다. 즉, 특정 문서에는 각 단어들의 TF-IDF 값이 들어가면서 문장을 Vector 화 하는 것이다. 각 값의 의미는 그

문서에 대표적인 단어를 의미하는 것이다. 위의 그림에서 각 Doc의 벡터는 빈도수 대신에 TF-IDF 값이 들어간다고 보면 된다.

그러나 이러한 단어 빈도수 기반(Count based model)의 벡터화는 매우 Sparse(하나의 문장을 나타내기 위해, 전체 문장에서 나오는 모든 단어들의 값이 포함되어야 한다. 예를 들어, 전체가 100 개의 단어인데, A 문장에 4 개 단어만 들어간다면 96 개는 0 으로 수치로 채워야 한다.)하고, 의미가 비슷한 단어도 다른 단어로 보기 때문에 유사한 의미를 표현하기 어려운 단점이 있다. 그리고 추출된 학습 문서(코퍼스)에 등장하지 않은 단어가 Test 샘플에 있다면 Out of Vocabulary 문제로 성능을 저하시키는 문제를 가지고 있다.

그래서 이를 보완하여 등장한 Vector Representation 방법이 Word2Vec 이다[15]. Word2Vec 방식은 함께 등장하는 단어들과의 분포를 가지고 표현함으로써 특정 단어와 함께 등장하는 단어들을 파악 가능하다. 즉, 기존의 Count 기반의 Representation 보다 좀 더 단어 하나하나가 의미를 가질 수 있도록 Vector화 한 것이다. Word2Vec의 사상은 Skip Gram 이라는 중심 단어를 가지고 주변단어를 맞출 수 있도록 학습된 방법이라, 단어가 주어지면 해당 단어와 유사한 단어를 추출하기 용의하다. 만약, 경제 관련 기사들을 Word2Vec 으로 학습시켜, 문장들을 벡터화 시키면, 그 벡터들(즉, 문장안에서 함께 등장했던 단어들)은 같은 방향, 비슷한 벡터를 가지고 임베딩 될 것이다. 예를

들어, 다음과 같이 “청년”과 유사한 Vector 를 찾게 되면 기사에서 항상 함께 언급된 단어들이 도출된다.

```
w2v_model.wv.most_similar(positive=["청년"])

2021-03-25 01:44:37,151 : INFO : precomputing L2-norms of word weight vectors
[('일자리', 0.7216043472290039),
 ('기업인', 0.6847594976425171),
 ('희망', 0.6828774213790894),
 ('창업', 0.6622642278671265),
 ('취업', 0.6309168934822083),
 ('구직자', 0.6183404326438904),
 ('장학', 0.6096115708351135),
 ('공동체', 0.6079512238502502),
 ('중소기업', 0.5930776000022888),
 ('창출', 0.5890467166900635)]
```

[그림 9] Word2Vec를 활용한 임베딩 및 적용사례

그래서 King - man + woman = Queen 이라는 일반적으로 사람이 생각하는 단어의 연산이 수치적으로 가능하게 된 것이다. 그러나 Word2vec 은 단어 간의 유사도, 관계파악, 추론 등은 가능하나, 문서에 한번도 등장하지 않는 단어의 Sub 단어에 대해서는 여전히 예측할 수 없다. 예를 들어 학습데이터에 ‘서울’만 있었고 여기서 파생된 ‘서울시’가 학습 문서에 한번도 없었다면 “서울”과 “서울시”는 다른 것으로 인식하는 것이다. 즉 OoV(out of vocabulary) 문제는 여전히 존재하는 것이다.

그래서 나온 임베딩, 즉 수치화 방법이 페이스북의 FastText 방법이다[16]. FastText 방법의 경우에는 문자단위(이를 n-gram 이라

불리움)로 세분하여 학습하기 때문에, ‘서울시’만 학습했더라도, [‘서’, ‘울’, ‘시’], [‘서울’], [‘울시’], [‘서울시’]를 따로 따로 학습함으로써 ‘서울’도 ‘서울시’와 동일한 단어로 유추 가능하게 된 것이다. 즉, 하나의 단어에서 파생된 단어의 유사도를 유추할 수 있는 Sub word Representation 문제를 해결한 것이다. 그러나 여전히 이 두가지 임베딩 방식에도 한계가 있다. Word2Vec이나, FastText나 동형어, 다의어 등 즉, 같은 단어인데 문맥에 따라 다른 의미를 사용하는 경우 성능이 좋지 못하다는 것이다. 즉, 주변의 단어만을 가지고 학습하기 때문에 문장의 문맥을 고려하지 못한다는 한계가 있다. 예를 들어 ‘배가 아프다’와 ‘배를 타고 간다’를 학습한 후 ‘배를 타고 가는데 배가 아프다’라는 문장이 입력되었을 때 각각의 ‘배’가 어떤 의미인지를 기계가 분석할 수 없다는 단점이 존재한다. 즉, ‘배’를 같은 벡터로 표시한다는 점이다.

2018년 등장한 ELMO의 경우에는 이러한 Word2Vec과 FastText의 단어수준의 임베딩 한계를 극복하고자 노력했다. ELMO 역시 단어 수준의 임베딩이지만, Pre-Trained 이면서, 양방향(BiLM: Bi-directional Language Model)으로 학습하여 문맥을 반영한 모델이라는 점이다. ELMO(Embedding from Language Model)로 기반으로 단어 임베딩에서 문장 수준의 임베딩으로 확대된 임베딩 방식등이 개발되었다. 바로 GPT, BERT((Bidirectional Encoder

Representations from Transformer)는 셀프어텐션 기반의 트랜스포머 네트워크를 활용하면서 문장수준의 임베딩이 가능하게 되었다. BERT 의 Full name 을 보더라도 ELMO 로부터 계승한 것으로 유추할 수 있다. ELMO 와 동일한 년도에 발표된 BERT 역시 Pre Trained Model 이다[17]. 특히 어마어마한 다양한 언어의 말뭉치로 사전 학습되어 있기 때문에, 어떠한 문장이 들어오더라고 기존에 학습된 모델을 기반으로 임베딩을 실시한다. 그러므로 그냥 입력으로 텍스트(Feature)와 Label(Classification, Generation etc)만 지정하여 학습을 하면 원하는 Task 수행이 가능해진다. 즉, 기 학습된 모델을 활용해 Task 에 적합하도록 인풋 Text 를 최적으로 벡터화 해주는 것이다. 바로, 이때부터 사람과 유사하게 문맥을 이해하고 단어들의 시퀀스를 고려하여 어떤 단어가 문장안에서 나올지, 이때 쓰인 단어의 의미가 무엇인지를 기계가 예측할 수 있는 언어모델의 근간이 되었다고 봐도 과언이 아니다.

2.2.3. 언어모델 (Language Model)

언어모델이란, 인간이 말하는 언어의 법칙을 컴퓨터로 모사한 모델로, 단어가 주어졌을 때 다음 등장할 단어의 확률을 예측하는 방식으로 학습함으로써 사람과 유사한 문장을 발휘할 수 있는 능력이다. 바로 언어모델이 형성되었다는 의미는 단어를 이해하고 문맥을 이해할 수

있다는 것을 의미한다. 예를 들어, 편의점에서 물건을 사고 점원이 “영”이라고 말하는 순간, 고객은 영수증으로 바로 인지하여 “영수증 필요없어요”라고 말하는 것과 같이 단어가 주어졌을 때 맥락을 바로 이해하는 것을 의미한다. 이러한 언어모델은 단어와 단어사이에 확률을 학습하면서 나온 것이다. 수많은 문장(말뭉치, Corpus)을 수집하여, 단어와 단어사이의 출현빈도를 카운트하고 이를 확률로 계산하여 단어별 확률 분포를 정의할 수 있다. 이렇게 함으로써 궁극적인 언어모델의 목표는 우리가 일상 생활에서 사용하는 언어의 문장과 단어의 분포를 정확하게 모델링할 수 있게 되는 것이다.

언어모델링을 할 수 있는 방법은 다음과 같이 문장을 샘플링한 후 문장안의 단어 분포들을 분석하여, X_1 부터 X_n 까지의 단어가 주어졌을 때 특정 단어 $X_{(n+1)}$ 가 나타날 확률로 계산된다. 즉, A, B, C, D 로 구성된 문장들을 학습하여, 이전 단어들이 주어졌을 때, 각 단어 별 출현 확률을 계산하는 방식이다. 이때 적용하는 것이 체인 룰이고 체인 룰에 적용되는 공식은 다음 수식(1)과 같다

$$\begin{aligned} P(A,B,C,D) &= P(D|A,B,C) P(A,B,C) = P(D|A,B,C)P(C|A,B)P(A,B) \\ &= P(D|A,B,C)P(C|A,B)P(B|A)P(A) \quad --- \quad (1) \end{aligned}$$

그러나 이때 문제는 계산할 때, 중간에 조합이 없으면 계산이 안된다는 것이다. 어마어마한 문자들의 조합을 모아야 하는데, 그렇지 못할 경우, 계산이 불가능하다. 예를 들어 아래와 같이 “나는 당신을 정말 사랑합니다.”라는 문장의 발생확률을 계산한다면 다음 수식(2)와 같다.

$$P(\text{나는}, \text{ 당신을}, \text{ 정말}, \text{ 사랑합니다.}) = P(\text{사랑합니다}|\text{나는}, \text{당신을}, \text{정말}) * P(\text{정말}|\text{나는}, \text{당신을}) * P(\text{당신을}|\text{나는}) * P(\text{나는}) \quad \dots \quad (2)$$

이때, ‘나는 당신을 정말 사랑합니다.’라는 문장이 없다면, 특정 단어가 주어졌을 때의 확률 계산이 불가능하다. 그래서 등장한 방법이 n-gram 기반 Markov Assumption 이다. 단어들을 모두 참조하지 않고, k 개만 보고, 근사 시킨다는 개념이다. 앞의 모든 단어를 보는게 아니라, 앞의 단어의 일부만 보면서 특정 단어가 등장할 확률을 구한다는 의미이다. 예를 들어 직전 등장한 2 개 까지만 본다면 다음 수식(3)과 같이 계산이 가능하다.

$$P(X_n|X_{<n}) = P(X_n|X_{n-1}, X_{n-2}) = \text{Count}(X_{n-2}, X_{n-1}, X_n) / \text{Count}(X_{n-2}, X_{n-1}) \quad \dots \quad (3)$$

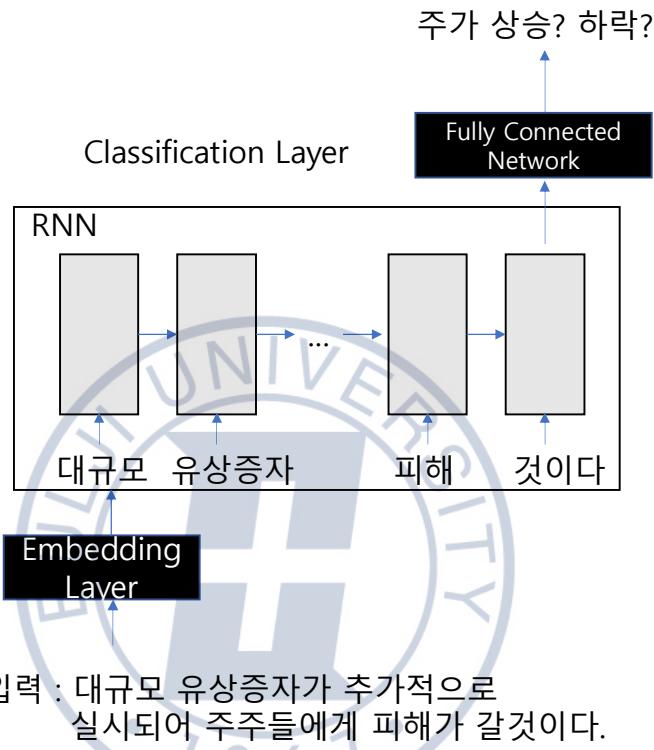
(※ $\text{Count}(X_{n-2}, X_{n-1}, X_n)$: 단어 X_{n-2}, X_{n-1}, X_n 이 코퍼스에 언급된 수)

즉, ‘당신을’ ‘정말’이라는 단어만 있어도 ‘사랑합니다’가 나올 확률을 계산할 수 있다는 의미이다. 정리하자면, 전체 문장이 아닌, 일부를 활용하여 근사한다는 개념으로, 참조할 단어들인 n 이 커지면 확률이 떨어지고, 너무 작으면, Markov Assumption, 즉, 전체 문장을 보지 않고 일부만 보고서 근사 시킨다는 가정을 왜곡시킬 수 있다. 그래서 보통 3-gram, 4-gram 정도를 활용한다. 그래도 중간에 관련 문장이 없게 되면 역시, 학습이 안되기 때문에, smoothing이라는 기법, interpolation이라는 기법, Back-off라는 기법을 통해 n -gram의 성능을 올릴 수 있는 방법들이 등장하게 되었다. 그러나 n -gram 기법은 말뭉치를 통해 단어들을 카운트하여 일반적으로 나올 확률을 구하기 때문에 쉽고, 간편하지만, 등장하지 않는 단어 조합에 대처하기 힘들다는 점과 n -gram 특성상 주변의 단어에만 포커싱하기 때문에 멀리 떨어진 단어에 대해 대처가 불가능하다는 점이다. 이를 Unseen Word Sequence 문제로 정의되는데, 이를 해결하기 위한 방법이 바로 Neural Language Model이다.

2.2.4. Neural Language Model

대표적인 NLM은 바로 시계열에 특화된 RNN(Recurrent Neural Network) 모델이다. 이 방법은 문장을 embedding 한 input layer와 sequence를 학습하는 RNN layer, 그리고 각각의 다음 단어들의 출현

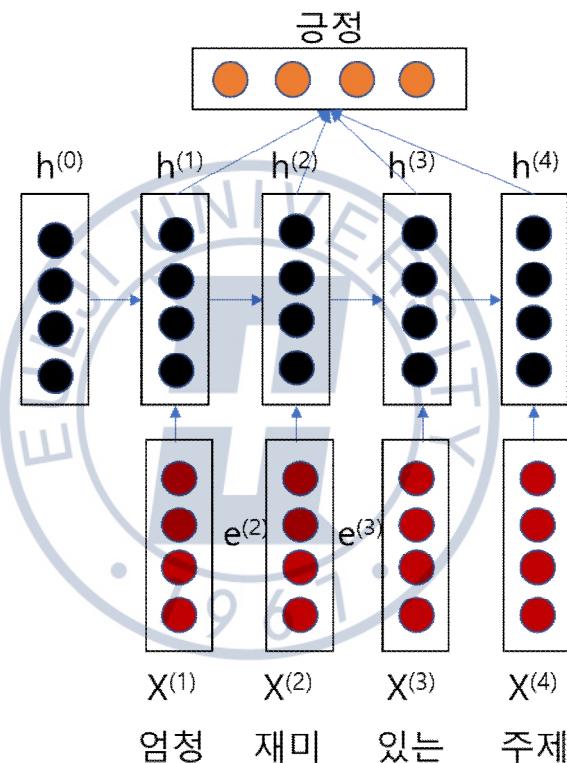
확률값을 계산하는 SoftMax layer로 이루어진다. 즉, 다음 그림과 같이 각 단어 다음에 등장할 최적의 단어를 예측하는 모델이다.



[그림 10] RNN를 활용한 NLM(Neural Language Model)

RNN을 활용하면, Unseen word Sequence, 즉, 중간에 단어가 없을 때의 문제를 해결하게 된다. RNN은 각 레이어마다, 입력벡터와 이전 출력벡터를 모두 고려하기 때문에 전체 시퀀스 정보를 가지고 판단 가능하기 때문이다. 그래서 문장 전체의 문맥을 기계가 이해하여 문장에 대해 최종 판단도 가능해진 것이다. 만약 어떤 문장의 긍정/부정

판단하는 Task 라면, <문장>과 그 문장의 <긍정/부정 레이블>만 가지고 학습시키면 전혀 새로운 문장을 넣어도 결과가 출력될 수 있다는 의미이다. 사람은 어떤 문장을 보고, “아! 이것은 부정적인 글이네” 바로 판단할 수 있는 것처럼, 기계도 가능하게 된다.



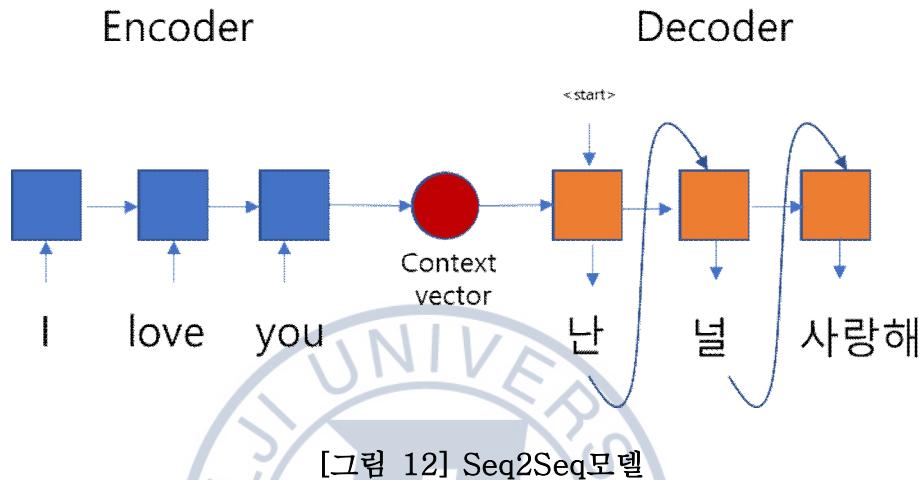
[그림 11] RNN를 활용한 텍스트 기반 분류

그러나 RNN 은 구조적으로 입력 문장의 길이가 긴 경우, 초반부에 나온 단어의 정보의 가중치가 매우 작아지면서, 긴 Sequence 중 처음 정보가 희석될 수 있고, 최근의 정보에 대해 가중치가 높아져, 정확한

문장의 문맥을 제공하지 못한다는 문제를 가지고 있다. 이를 Long Term Dependency Problem 이라고 불리운다. 그리고 최근 정보, 즉 문장의 마지막 단어에 좀더 집중되면서 불필요한 정보가 학습될 수 있다는 단점이 있다. 이를 해결하기 위해 등장한 Neural Language Model 이 바로 GRU(Gated Recurrent Unit)와 LSTM(Long Short Term Memory)이다. 이 두가지 방법은 각 단계별 Hidden State에서 필요한 정보만 저장하고 나머지 불필요한 정보를 잊어버리는 로직을 넣어 해당 단어의 정보를 계속 가지고 갈지 아니면 버릴지 판단 후 웨이트를 부여하기 때문에 초반의 정보에 대해 주요한 정보의 경우, 손실없이 끝까지 유지함으로써, 마지막 최종 Context Vector에 어느 정도 웨이트를 가지고 저장될 수 있다. 이를 통해 RNN의 단점인 입력문장이 길어질수록 앞의 정보가 뒤로 충분히 전달되지 못하는 Long Term Dependency problem을 다소 완화시킬 수 있게 되었다.

여기에 역방향으로도 학습함으로써 좀더 다음 등장할 단어의 추론 정확도를 높이는 Bi directional LSTM도 등장했다. 일종의 2 Layered LSTM 모델이다. 바로 이를 기반으로 나온 모델이 ELMO와 BERT이다. 그리고 이후 RNN을 기반으로 문장을 인코딩하여 만든 매우 Dense한 Context Vector를 기반으로 단지, 직접 문장을 가지고 학습하여 Regression이나 Classification 문제를 해결하는 것뿐 아니라, 이를

활용해 디코더를 추가하여 다양한 출력, 즉, 문장을 출력하는 Seq2Seq 구조가 탄생했다.



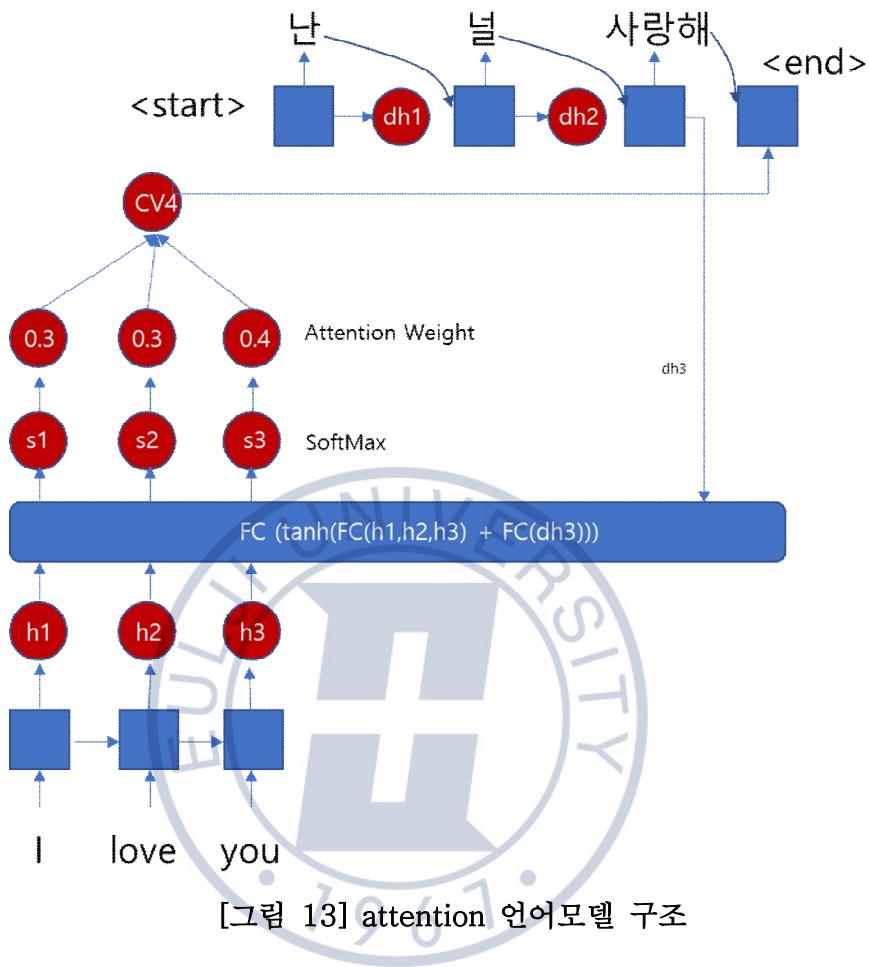
Seq2Seq 모델인 인코더와 디코더가 쌍으로 존재하는 구조로써 음성을 입력으로 넣으면, 텍스트로 출력되는 STT(Speech to Text) 솔루션, 영어를 넣으면 한국어로 번역되어서 나오는 번역 솔루션, 이미지를 넣으면 이미지에 대한 설명이 나오는 image 해석솔루션 등, 각종 실시간 Natural Language Generation 이 가능하게 되었다.

RNN이나, LSTM, GRU 는 Long term Dependency 문제를 다소 해결한 언어모델이지만, 아직, 사람이 가지고 있는 언어모델까지는 한계가 있다. 역시 엄청 긴 문장에서는 Long term Dependency 가 여전히 남아 있다는 점과 문장에서 핵심적인 키워드를 캐치하지 못한다는 점이다. 예를 들어, 사람들은 “난 널 사랑해”라는 말에서,

“사랑해”에만 집중해도 전체 문장을 이해할 수 있다. 즉, 특정 키워드에 집중함으로써 전체를 이해하는 행태처럼, 기계에도 이러한 ‘집중’ 개념을 넣는 것이 바로 Attention 모델이다. 요약하자면 인간이 정보 처리시, 모든 문장의 Sequence 를 이해하면서 정보 처리하는 것이 아니라, 중요한 단어, 즉 Feature 를 고려해서 전체를 이해하는 것이 Attention 언어 모델이다. 앞서 RNN 기반의 Seq2Seq 모델은 RNN 인코더의 최종 출력 값인 Context Vector 만을 활용했지만, Attention 모델에서는 Encoder 의 각 RNN 레이어의 아웃풋을 매번 모두 활용함으로써 전체를 효율적으로 파악하는 모델이라 할 수 있다[18]. Attention 의 개념이 결국, 본 연구에서 다루는 핵심이기 때문에, 자세히 살펴보고자 한다.

2.2.5. Attention 과 트랜스포머

Attention 의 개념은 RNN 에서 시작한다. RNN 을 통해 임베딩된 문장의 각 단어를 인풋으로 받아 Feed Forward Fully Connected Layer 를 거쳐 Attention Score 를 각 단어가 부여받게 된다. 여기에 SoftMax 를 취해 줌으로써 가장 집중해야 할 단어에 Attention Weight 를 주는 것이다. Attention Weight 를 최적화하기 위해 decoder 의 정답과 비교하면서 보정하여 최적의 attention weight 를 구하는 구조이다.



예를 들어 “I love you” 라는 문장이 인코더에 들어가면, “I”, “love”, “you” 각각에 attention weight 가 주어지고, 이를 가지고, 학습을 위한 출력 정답이 “난” 다음에는 뭐가 나오지?”라고 질의했을 때 인코더에서 각 단어의 attention weight 를 통해 구해진 Context vector 로 “사랑해” 가 바로 나오면, 다시 Attention weight 를

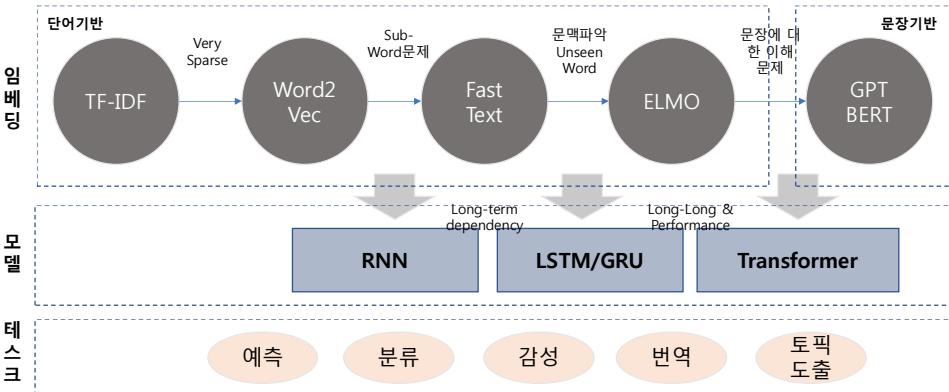
보정하여, 정답인 “난 너를”이 나올 때까지 Attention weight 를 최적화한다.

이렇게 진행함으로써 엄청 긴 문장이라도, 핵심 단어의 정보를 유지하면서 학습이 가능하다. 즉, Attention 은 기존 RNN, LSTM, GRU 에서 아직 미비한 엄청 긴 문장에서의 성능 저하와, 키워드 집중에 대한 문제를 해결한 모델이라 볼 수 있다. 그리고 무엇보다 Seq2Seq 의 성능을 극적으로 올린 모델이라 할 수 있다. 그러나, 역시 Attention 모델은 RNN 를 적용하고 있기 때문에 순차적이다. 순차적인 적용은 연산 역시 순차적으로 이뤄지기 때문에 속도가 느린 단점이 있다. 그래서 등장한 개념이 Self-Attention 개념이다. 기존에 RNN 기반의 Attention 모델은 순차적이면서 Attention 웨이트를 최적화하기 위해서는 Decoder 이후 나온 결과와 실제 정답과 비교하여 Attention weight 를 조정하여 최적화하였다. 이에 비해 Self-Attention 모델은 병렬적으로 Attention Weight 를 계산할수 있도록, RNN 을 없애 버리고 디코더의 출력값과 비교하지 않고, Encoder 의 입력값 자체와 비교하여 Attention weight 를 구하는 방식을 취하였다. 입력값과 인코더를 통해 나온 입력값의 예측값과 일치하는지를 보면서 Attention weight 를 구하기 때문에 이를 Self Attention 이라 불리운다. 즉, 인풋으로 들어오는 문장을 가장 잘 예측할수 있는, 자기 자신을 가장 잘 표현하는 Attention 을 구하는 것이다. 이를 위해, Self Attention 모델의 각

단어(토큰)은 Query, Key, Value 라는 벡터를 가지고 있다. 예를 들어 “나는 너를 사랑해”에서 “나는”과 “나는”, “나는”과 “너를”, “나는”과 “사랑해”의 상관관계를 구한 값이다. (이때 “나는”은 Query, 각각의 대상 단어들은 Key, 상관계수는 Value 라 불리운다.) 그리고 이러한 Q, K, V 를 동시에 수 개를 만들어 가장 자기 자신을 잘 표현하는 벡터를 구함으로써 최적화된 언어모델을 생성하는 것이다. 이때 수개의 QKV 벡터를 병렬을 생성하여 학습한기 때문에 Multi head self-attention 이라 불리운다. 이런 Multi head self-attention 개념을 통합하여 encoder-decoder 개념으로 RNN 기반의 seq2seq 를 개혁한 모델이 바로 구글에서 만든 트랜스포머(Transformer) 언어모델이다[18].

지금까지의 언어모델 관련된 연구 내용을 간단히 정리해보면 다음과 같다. 자연어를 기계가 이해하기 위해 문장을 수치화 하는 과정이 바로 임베딩이다. 이전에는 TF-IDF 같은 통계적 벡터로, 즉, One hot vector 로 표시하여 기계가 이해하게 했으며, 이후 Word2Vec 등이 등장하면서, skip gram 로직을 통해 문장안에서 인근에 함께 등장한 단어들을 학습하여, 단어들을 벡터화 하던지, 단어와 유사한, 연관된 단어를 찾는 등에 활용하였다. 이를 가지고 특정 단어가 많이 나오면, 해당 글이 어떤 분류인지도 알 수 있는 분류의 문제 해결에도 도움이 되었다. 그러나 Word2vec 의 경우, sub-word representation 문제는 해결할 수 없었는데, 페이스북의 FastText 를 통해 이를 해결할 수

있었다. 그러나 여전히 문맥을 이해하는 데는 단어기반은 턱없이 부족하였다. 이후, 문장의 문맥을 이해하는 언어모델이 등장하였다. 우선은 n-gram 를 통해 전체 문장을 훑으면서 단어들과 해당 단어가 나왔을 때 다음 단어가 무엇이 등장할지 확률적으로 계산하는 방식이 도입되었지만, Unseen word problem 에 의해 정확도가 현저히 떨어지는 문제가 발생하여, 이를 근본적으로 해결할 수 있는 Neural language model 이 개발되었다. 그 첫번째가 바로 RNN 모델이다. RNN 를 통해 문장의 각 단어를 Forward, Backward 로 학습하여, 전체 문맥을 Seq 있게 벡터화할 수 있었고, 최종 산출물인 각종 Label 과 연결 가능하게 되었다. 그러나 문제는 Long term dependency 문제가 있어 이 해결책으로 LSTM, GRU 모델이 나왔지만, 직렬적으로 처리함으로 발생하는 시간문제와, 긴 문장의 경우, LSTM, GRU 에서 처리하기에 역부족이라는 단점이 있었다. 이를 해결한 방법으로 Attention 이라는 개념이 나왔고, 병렬적으로 처리할 수 있는 멀티헤드 어텐션 개념과, 디코더 없이 입력만을 가지고 스스로 학습할 수 있는 self-Attention 개념이 접목되면서, 인코더 디코더의 성능을 월등히 향상시킬 수 있었다. 이를 활용해 만든 언어모델이 바로 Transformer 이다. 이를 그림으로 표현하면 다음과 같다. 이후 구글의 방대한 언어 데이터 셋을 가지고 Transformer 로 학습하여 만든 Pre Trained Model 이 바로 BERT 이다.



[그림 14] 임베딩 및 언어모델의 발전 요약

2.2.6. BERT(Bi-Directional Encoder Representations from Transformer)

BERT는 bi-directional RNN 같이 Bi Directional Transformer 언어 모델(임베딩과 모델이 모두 포함되어 있음)이다. 단, 가장 큰 특징 중 하나는 Transformer의 Encoder만 가지고 웬만한 테스크들을 모두 구현가능하다는 점이다. 이미 학습된 Pre-Trained Model로써 각종 테스크를 위한 레이어(예:classification layer)만 부착하면 다양한 NLP 테스크들이 수행 가능하다. 구글의 방대한 언어 데이터로 8 억개의 책에 있는 단어들과 25 억개의 위키피디아 단어, 그리고 30,000 개의 토큰화된 말뭉치로 학습한 방대한 모델이다. 방대한 문장을 self-attention 방식으로 학습되었기 때문에, 특정 단어 다음에 어떤 단어가 오는지, 특정 문장 다음에 어떤 문장이 오는지 예측이 가능하다[17].

이러한 방대한 자료를 가지고 학습한 방법은 Masking 기법과 Next Sentence Prediction 기법을 활용하였다. Masking 기법이란, 문장이

주어지면, 문장의 한 단어를 Masking(가리거나)하거나, 다른 단어로 대체하거나 하여, 정답을 찾는 방식이다. 역시 Self-Attention 이 적용되어 입력 문장을 그대로 활용하여 맞출 때까지 학습한다. 이런식으로 방대한 문장(다양한 언어들 포함)들을 사전에 학습되었기 때문에, 그대로 활용해도 어느정도 성능은 나오게 된다. 추가 성능 향상을 위해서는 학습데이터를 추가로 모아서 학습하는 파인튜닝이라는 단계가 포함되어 기존의 방대한 Pre-trained 모델의 업그레이드 모델을 만들어 성능을 추가로 업그레이드 가능하다.

요컨데 BERT 는 방대한 텍스트 자료를 기반으로 텍스트들을 이미 학습한 후 임베딩하여, 기계가 사람들과 유사한 언어모델을 가질 수 있게 하였다. 지금까지 나온 언어모델들은 별도의 학습이 필요했지만, BERT 는 별도의 학습 필요 없이도 Pre-Trained 된 모델을 기반으로 사람처럼 어떤 단어가 나오면 그 다음 나올 단어와 문장이 있으면 그 다음 나올 문장을 예측할 수 있게 되었다. 사람은 많은 경험을 통해 어떤 글을 보면, 그 글에 의해 무엇이 발생할지, 그 글이 어떤 의미인지, 그 글이 어떤 다른 글과 유사한지, 그 글이 어떤 부류 인지 알 수 있다. 마찬가지로, BERT 도 이러한 학습을 미리 시킨 것이다. 특히, 글들에 대해 어떤 의미인지 알 수 있게 미리 학습한 것이다. 그러면 여기에 추가로 그 글을 읽고 무엇이 예측될 것인지 어마 어마한 양으로 학습시키면 그 글로 인한 예측이 가능해 질것이다. 바로 이러한

언어모델을 활용해 글을 통한 주가 예측이 가능하다는 것이 바로 본 연구의 주제이다.

본 연구에서는 종목 뉴스에 따른 주가의 향방을 예측하는데 다양한 텍스트 임베딩 방법들을 적용하여 비교할 예정이다. 특히 BERT 의 인간과 유사한 임베딩 성능으로 뉴스와 주가 하락여부에 대한 많은 양의 학습은, 그 안에서 패턴을 찾고, 그 패턴에 따라 예측이 가능할 것이라는 가설로 진행해 보고자 한다.

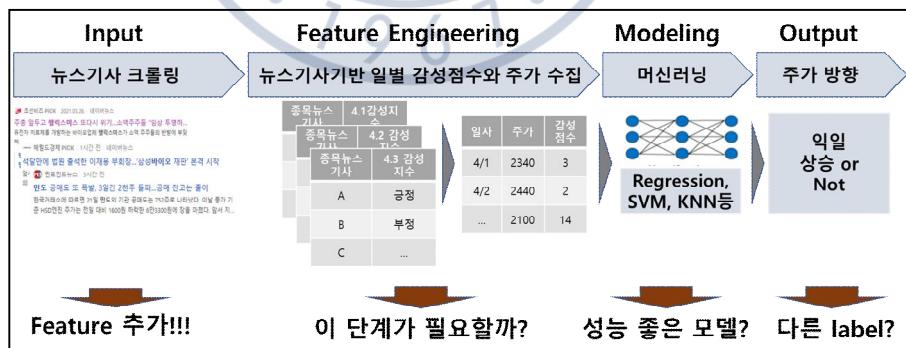


3. 제안된 NLP모델 설계 및 검증체계

3.1. 연구문제 및 NLP 모델 제안

지금까지 인공지능 기반 자연어 처리를 통한 주가 방향에 대한 연구들에 대해 요약하면 다음과 같다.

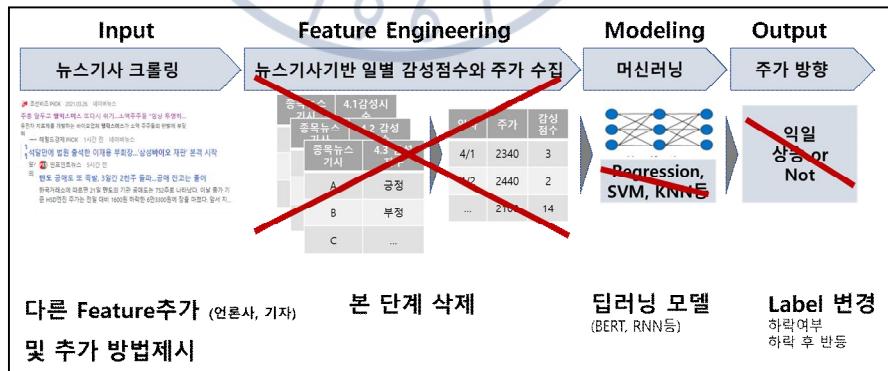
- 뉴스기사 또는 SNS 의 감성분석을 통한 익일 종목 또는 종합 주가 상승여부에 대한 예측으로 매수에 초점
- TF-IDF, Word2vec 등 최신의 Neural Language Model이 아닌 통계적인 임베딩 모델 및 모델링을 위해 머신러닝 다수 적용
- BERT 를 활용한 최신 연구는 한국어가 아닌 영어 기사활용
- Feature 는 뉴스 기사 기반 감성 지수화 후, 해당 종목의 해당 일에 대한 감성 점수화 하는 Feature Engineering 실시



[그림 15] 기존 연구들에 대한 개요 및 고찰 필요 사항

기존 연구 대비 본 연구는 다음과 같이 기존 연구를 보완한다는 관점에서 진행하였다.

- ① 가장 최신의 Neural Language Model 인 Transformer 기반의 BERT 와 한국어 BERT 인 KoBERT 를 적용하여 한국어 뉴스기반 주가 방향을 예측하는 데 있어 성능향상을 위한 모델을 실험한다.
- ② 공매도의 범위 확대와 주가 상승에 따른 투자자들의 매도 시기 포착에 대한 니즈가 증가하는 시점에서 보유하고 있는 종목을 매도 또는 지속 보유 관련 의사결정을 하기 위한 모델의 가능성을 확인한다. 즉, 개별 종목의 하락을 유도하기 위해 악의적인 뉴스(하락 후 반전)와 그렇지 않는 뉴스(지속 하락)를 검출할 수 있는 모델을 개발한다.
- ④ Feature Engineering 없이 바로 딥러닝으로 성능이 나오는지 확인하고, 새로운 Feature 들을 Text 에 추가할 수 있는 방안을 검증 및 제시한다.



[그림 16] 본 연구의 방향 개요

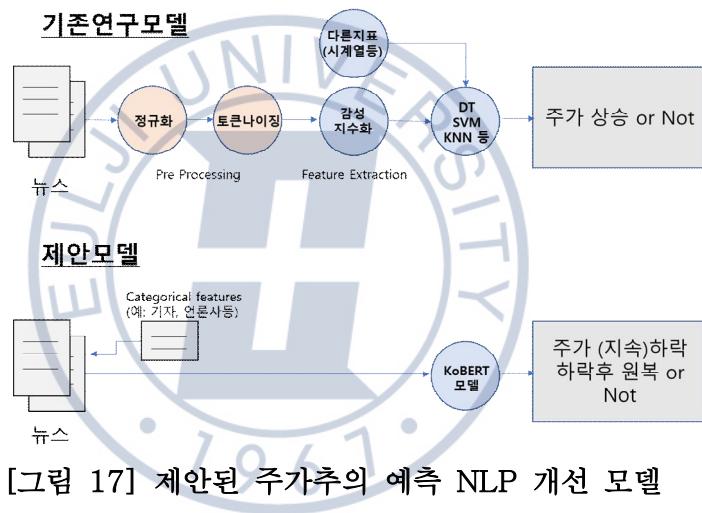
요컨데, 본 연구에서는 NLM(Neural language Model)을 활용하여 직접적으로 뉴스와 주가의 하락 가능성을 예측하고, 공매도가 확대되는 시점에 일시적 하락을 노리는 악의적인 기사인지, 지속적인 하락의 전조로써의 악재인지를 구분할 수 있는 모델을 검증하였다.

공매도 전략의 투자성과 분석 연구에 의하면 특정 종목에 공매도가 발생한 이후 실제로 기업의 이익감소, 애널리스트의 투자의견하향 보고서, 채권등급하락, 기업등급하락, 유상증자등 악재성 이유도 있지만, 주가하락을 임의로 조장하기 위해 악재성 정보를 흘리는 경우 등 다양한 공매도 이후의 주가 하락을 위한 다양한 패턴을 제시하였다[2]. 2020년 9월 수소전기차로써 제2의 테슬라로 급등한 수소 트럭 개발사 ‘니콜라’ 대상으로 부정적인 기업 리포트를 퍼블리싱하여 니콜라 주식을 폭락시켜 공매도를 통한 거액의 돈을 번 사례도 있었듯, 공매도로 돈을 벌 수 있는 상황에서 임의적 악재성 뉴스기사는 개별종목의 주가 하락을 조장한다. 그러나 Fact 가 없는 악의적인 기사는 투자자의 이성적이고 공정한 투자에 방해가 되고, 오히려 시장 질서를 무너뜨릴 수 있다. 앞으로 공매도가 활성화되는 시기에 분명히 공매도를 통한 이득 창출을 위해 이러한 행태들이 증가될 것으로 예상된다.

본 연구에서는 기사들과 주가하락과의 관계를 모델링하여, 종목 뉴스가 나왔을 때, 매도해야 할지, 또는 지속 보유하고 있어야 할지 의사결정하기 위한 모델을 개발, 검증함으로써 투자자의 의사결정에

도움을 주는 방향으로 진행하였다. 본 연구에서 제안하는 추가의 추이를 예측하는 NLP 모델은 다음과 같다.

- 1) 감성점수화 없이 텍스트 그대로 처리할 수 있는 텍스트 임베딩 활용
- 2) Pre-Trained 모델 활용(KoBERT 활용)
- 3) Text에 Categorical Feature 추가(언론사, 기자, 타이틀)방법 적용
- 4) 3 Classes Label로 하락이후의 방향도 함께 예측할 수 있는 모델



[그림 17] 제안된 주가추이 예측 NLP 개선 모델

본 연구에서 제안하는 모델에 대해 다음과 같은 가설을 통해 성능향상에 대해 검증하고자 한다.

[가설 1]

종목 뉴스로 주가추이를 예측할 때, 감성점수화 하는 Feature Engineering 없이도 뉴스 기사를 그대로 활용한 BERT(KoBERT)모델을 통해 유의미한 모델이 가능하다.

기존의 NLP 기반 주가 예측 방식들은 뉴스들의 감성점수를 구한 후 이를 가지고 다시 종합하여 긍정/부정의 뉴스의 수들을 이용해 해당 종목, 해당 날짜의 종합 감성점수화 하고 이를 활용해 예측하는 번거로움이 있다. 즉, 어마어마한 뉴스 크롤링과 점수화를 위한 연산작업 등의 사전 작업에 들어가는 시간이 많이 소요된다. 바로 머신러닝의 전형적인 Feature Engineering 단계이다. 그러나 최근 개발된 Attention 언어모델의 강점인 맥락이해 메카니즘을 적용한다면 직접 주가와 연계하여, 방향을 결정할 수 있을 것이라는 기대를 가지고 가능성에 대해 검증해 보았다. 예를 들어, 하나의 뉴스만으로도 뉴스의 재료에 따라 영향도를 판단할 수 있다는 기대를 가지고 빠르고 쉽게 주가의 방향을 예측할 수 있다는 측면에서 의미를 가질 가설이라고 판단된다.

[가설 2]

종목 뉴스와 주가하락과의 관계모델 중에서 정확도는 한국어 기반의 Pretrained BERT 모델이 기존 LSTM 모델 대비 성능이 좋을 것이다.

지금까지 연구된 언어모델들의 비교를 통해, 기사를 통한 주가의 방향을 가장 잘 예측할 수 있는 모델을 검토하였다. Word2Vec 임베딩 기반의 LSTM은 긴 문장에 대해서는 성능이 오히려 좋지 않을 것으로 예상되며, 긴 문장의 문맥을 파악하고, 수많은 문장을 사전에 학습된 BERT가 성능이 좋을 것으로 가정하여 실험하였다. 그리고 무엇보다 한국어를 특화로 만든 KoBERT는 한국어 기사에서는 BERT 보다도 성능이 좋을 것으로 가정하고 실험을 실시하였다.

[가설 3]

뉴스기사 내용뿐 아니라, ‘언론사’와 ‘기자’를 Feature로 포함되었을 때 더 높은 성능을 보일 것이다.
그리고 NLP에서 Feature 추가는 메인 Text에 병합하는 방식으로 수월하게 추가될 수 있을 것이다.

뉴스기사를 살펴 보면, 특정 언론사와 특정 기자들의 글이 매우 악의적인 경우가 종종 보인다. 특히, 한 종목에 대해 지속적으로 악의적인 글을 내비치는 언론사와 기자도 있을 것이다. 이러한 패턴을 보았을 때 언론사와 기자까지도 함께 Feature로 고려하여 분석이 된다면 주가 향방의 높은 예측력을 예상하였다. 그리고 또 한가지 더

검증하고자 했던 것은 NLP 에서는 Feature 를 추가하는 방법은 기존에는 문장 임베딩벡터에 추가적인 Feature 벡터를 추가하는 번거로운 작업을 통해 Feature 를 추가했지만, 별도의 Vector화 없이 단지, Text 항목에 그대로 추가하여 한번에 임베딩 벡터화 함으로써 수월하게 Feature 추가가 된다는 것도 함께 검증을 시도하였다.

[가설 4]

기사를 통해 단기적 원복 현상을 통해 악의적으로 쓴 기사와 악재성 기사는 구분할 수 있을 것이다.
또는 하락 후 지속할지, 반등할지 예측할 수 있을 것이다.

그랜빌의 법칙에 의하면, 3:3 법칙이라는 것이 있다. 20 일 이동평균 밑으로 떨어졌지만, 3 일 이내, 그리고 3%이내의 조정이 마무리된다면 기존의 상승 추세는 유효하다는 의미이다. 이 법칙을 활용한다면, 진정한 악재가 아닌 악의성 기사의 경우, 3 일 안에 다시 원복 될 수 있다고 가정하고 과연 모델링을 통해 구분할 수 있을지 검증하였다. 다음 글들을 살펴보면, 하나는 공시로는 발표되지 않았던 추측성 기사이고, 다른 하나는 공시로 나온 악재성 기사이다. 이후의 주가를 보면 추측성 기사가 나온 경우, 하락했다가 3 일 후 원복함을 보였고, 악재성 기사 이후에는 지속 하락함을 볼 수 있었다.

[단독] 금감원, 장밋빛 홍보자료 쓸어낸 헬릭스미스에 '경고'

이데일리 | 2020.11.30 04:03 글꼴 - +

헬릭스미스, 부실펀드 투자 논란 후 홍보자료 밀어내기

금감원 제출서류에 기재한 투자 위험은 쑥 빼[이데일리 박종오 기자] 최근 부실 사모펀드 투자 사실이 드러나 논란이 된 코스닥 상장 바이오 기업 헬릭스미스(084990)가 금융 감독 당국으로부터 경고를 받았다. 회사가 장밋빛 전망을 담은 홍보 자료를 밀어내기식으로 시장에 배포해 투자자의 혼란을 초래할 수 있다는 이유에서다. 헬릭스미스는 누적 적자로 인해 관리 종목 지정 위기가 불거져 연내 1000억원대 신규 투자금을 모집하기 위한 유상증자를 추진 중이다.

2020.12.03	27,700	▲ 1,450	26,150	28,250	24,850	1,153,168
2020.12.02	26,250	▼ 700	26,550	26,900	25,500	703,771
2020.12.01	26,950	▼ 250	27,550	27,850	26,500	368,526
2020.11.30	27,200	▼ 1,900	28,500	28,700	27,200	544,441

헬릭스미스, 부실 사모펀드 등에 489억 투자…유증 악재

머니투데이 | 2020.10.18 18:47 글꼴 - +

[머니투데이 김근희 기자] ["유상증자 실패 시 관리종목 지정될 수 있어"]

HELIXMITH

바이오 기업 헬릭스미스가 부실 사모펀드 등에 투자해 손실이 커질 가능성이 있는 것으로 드러났다. 올 연말 추진할 계획이었던 2800억원대 유상증자에 악재가 생기면서 관리종목으로 지정될 위기에 몰렸다.

2020.10.22	20,450	▼ 50	19,450	21,750	19,200	2,095,204
2020.10.21	20,500	▲ 900	19,800	22,250	19,450	3,532,438
2020.10.20	19,600	▼ 1,950	20,450	20,900	18,200	5,304,930
2020.10.19	21,550	↓ 9,200	26,250	27,150	21,550	6,990,165
2020.10.16	30,750	▼ 500	31,200	31,600	30,200	446,124

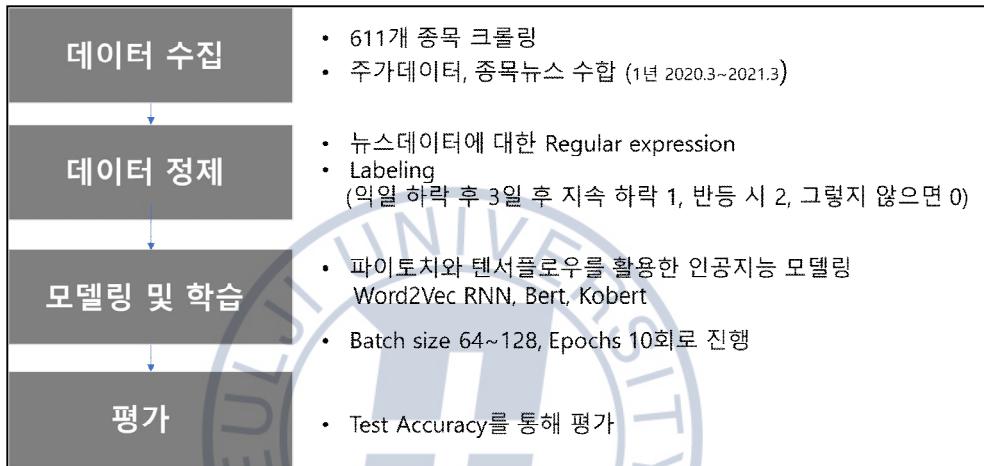
[그림 18] 악의적 기사와 악재성 기사 예시

위의 가설의 검증 결과에 따라 얻고자 하는 의미는 뉴스기사를 가지고 미리 주가 하락을 예견할 수 있는 지와, 다시 원복 할 기사인지, 지속적으로 하락할 기사인지를 구분함으로써, 투자자의 손실을 막기 위한 인공지능 모델을 만드는데 의의를 가진다고 할 수 있다.

3.2. 모델 성능 검증방법

3.2.1. 검증 프로세스 방법에 대한 개요

본 연구는 다음과 같이 4 단계로 진행하였다.



[그림 19] 연구 프로세스

첫번째 단계인 데이터 수집단계에서는 파이썬 프로그래밍을 통해 611 개 종목에 대해 2020.3 월부터 2021.3 일까지의 주가데이터와 종목뉴스를 포털 증권 사이트를 통해 크롤링하였다.

두번째 단계인 데이터 정제 단계에서는 텍스트 데이터의 노이즈 문자들을 제거하는 Regular expression 을 실시하였고, Word2Vec 기반 RNN 의 경우에는 별도의 토크나이징을 실시하였다. Label 은 주가 데이터를 가지고 뉴스 발표 후 익일 주가 하락시 (또는 지속하락시) ‘1’ 로 레이블링하고, 하락 후 반등시에는 ‘2’ , 그렇지 않으면

'0' 으로 부여하였다. 세번째 단계인 모델링 및 학습단계에서는 학습데이터와 테스트 데이터를 미리 분류(9:1 로 분리함)하여 모든 모델에 공통적으로 활용하였다. 파이토치와 텐서플로우를 활용하여 인공지능 모델인 RNN, BERT, KoBERT 로 각각 모델링 하고 마지막으로 4 단계에서는 평가 및 가설에 대해 검증하여 가설이 맞는지, 평가 결과가 어떨지 확인하였다. 그리고 앞서 제시한 4 가지 가설에 대해 유의성을 검증하기위해 다음과 같은 방법과 기준을 가지고 실시하였다.

[표 5] 각 가설에 대한 검증 방법 및 검증에 대한 판단 기준 요약

연구 가설	평가 방법	판단 기준
(가) 종목 뉴스기사 만으로 감성지수화 없이 주가 추이를 예측할 수 있다.	뉴스기사의 감성지수 없이 주가하락여부(Label) 와 종목뉴스(Feature) 사이의 학습을 통해 테스트 예측 정확도로 판단	기존 연구들의 평균인 66%보다 평균적으로 높은지 검증(1-sample t-Test)
(나) KoBERT 가 성능이 가장 좋을 것이다.	주가 하락 여부(Label) 와 종목 뉴스(Feature) 사이의 학습모델을 바꿔가면서 테스트 정확도 확인 (동일한 테스트/학습 데이터 셋)	LSTM 대비 KoBERT 의 평균이 높은지 검증 (2-sample t-Test)
(다) 주가하락은 기사뿐 아니라, 기자와 언론사에 따라 좀더 정확히 예측할 것이다.	3 가지 유형의 데이터 셋 변경하면서 비교(<기사제목> <언론사/기사제목> <기사/언론사/기사제목/기자>)	데이터셋에 따른 평균 Accuracy 비교 (ANOVA test)

(라) 원복하는 기사와 지속 하락하는 기사를 구분할 수 있을 것이다.	단기 복구 여부, 지속하락 여부, 상승/보합으로 3 개의 Class 로 구분하여 기사들을 학습하고 모델링한후, 기존 연구보다 높은 Accuracy 가 나오는지 확인(66%)	기존 연구들의 평균인 66%보다 평균적으로 높은지 검증(1-sample t 테스트)
--	---	--



3.3 검증을 위한 실험자료 및 실험설계

3.3.1. 실험 자료

4 가지 가설들을 검증하기 위해, 다음과 같이 Label 데이터인 종목주가데이터와 종목별 뉴스기사 데이터는 포털 증권에서 1년동안의 데이터들을 파이썬을 통해 크롤링하여 수집하였다. 수집한 데이터의 개요와 주요 내용은 다음과 같다.

[표 6] 실험을 위해 수집한 데이터셋

데이터	컬럼	건수	수집 방법	비고
종목코드	종목코드, 종목명	611	종목 크롤링	<21.3.30> 기준 공매도 거래 있는 종목
종목별 종목 기사	날짜, 언론사, 기사제목, 기사원문, 기자, 종목명, 종목코드	82,616	포털 증권 종목별 뉴스 크롤링	<2020.3 ~2021.3>
종목별 주가	종목명, 날짜, 시작가, 최고가, 최저가, 종가, 거래량, 코드	305,497	포털 증권 종목별 주가 크롤링	<2019.3 ~2021.3>

```
#뉴스 기사 전체 크롤링

articles = []

for article in link_result:
    headers = {"user-agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 11_1_0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.149 Safari/537.36"}
    response = requests.post(article, headers=headers)
    dom = BeautifulSoup(response.content, "html.parser")
    articles.append(dom.select_one("div#news_read").get_text())

# 변수를 합쳐서 해당 디렉토리에 csv파일로 저장하기

result= {"날짜": date_result, "언론사": source_result, "기사제목": title_result, "링크": link_result, "기사원문": articles}

#df_result_temp = pd.DataFrame(result)

#print("다운 받고 있습니다-----")
```

df_stock.info[50:80]

	날짜	시작가	최고가	최저가	증가	거래량	증목코드	label	label12	shift1	shift3	label13	label14	
0	2019-03-25	53400	53400	51700	51700	15765	006840	False	0	NAN	50300	False	True	
1	2019-03-26	51700	52800	51200	51200	15493	006840	True	1	51700	50900	True	True	
2	2019-03-27	51300	52300	51300	51300	14667	006840	False	0	51200	52400	False	False	
3	2019-03-28	50400	53000	51600	50200	60300	18926	006840	True	1	51500	52000	True	False
4	2019-03-29	50400	53000	50400	50900	40999	006840	False	0	50300	53500	False	False	
5	2019-04-01	51100	53100	51100	52400	20690	006840	False	0	50900	54200	False	False	
6	2019-04-02	52400	52800	51600	52000	9778	006840	True	1	52400	53000	True	False	
7	2019-04-03	51500	53800	51200	53300	24453	006840	False	0	52000	55400	False	False	
8	2019-04-04	51200	54300	53200	54200	14729	006840	False	0	53500	55700	False	False	
9	2019-04-05	54700	55300	53600	55000	16541	006840	False	0	54200	55900	False	False	

company_articles.head()

	날짜	언론사	기사제목	링크	기사원문	기자	총목	총목	시간
0	2021-02-15	이데일리	AK홀딩스, 첫년 영업손실 2221억... 쪽지전환	https://finance.naver.com/item/news_read.nhn?code=A006840&id=...&category=...&date=2021-02-15	winketwet (이데일리) 정동희 기자 AK홀딩스(006840)는 지난해 연말기준... ₩(2020년 주요사업 실적)보다 10% 이상 증가한 70%↑... 그쳤다/김... 1,398억원으로 전년... 2	AK홀... 등... 006840	17:25		
1	2021-02-15	서울경... 계	AK홀딩스 매출 50%↓... 2,200억원 영업赤자	https://finance.naver.com/item/news_read.nhn?code=A006840&id=...&category=...&date=2021-02-15	winketwet (서울경기) 김보나 기자 AK홀딩스(006840)는 지난해 영업... 매출 대폭 감소...◇AK홀딩... 3	AK홀... 등... 006840	16:37		
2	2021-02-04	매일경... 계	(주)증권회사장 공시) 한일불광 / AK홀딩스... 절대... 증권업	https://finance.naver.com/item/news_read.nhn?code=A006840&id=...&category=...&date=2021-02-04	winketwet ◇한일불광(코스피1990) 관계자는 회사에... 4	AK홀... 등... 006840	17:30		
3	2020-11-20	아시아... 경제	AK홀딩스, 캐주얼등총 300억 규모 자본보유 결정	https://finance.naver.com/item/news_read.nhn?code=A006840&id=...&category=...&date=2020-11-20	winketwet (아시아경제 박지환 기자) AK홀딩스(006840)는 재정적... 자본보유 결정에... 5	AK홀... 등... 006840	16:01		
4	2020-09-11	서울경... 계	(시그널) 아모 차들석 부회장 신임 임명... 부회장 신임 임명	https://finance.naver.com/item/news_read.nhn?code=A006840&id=...&category=...&date=2020-09-11	winketwet (서울경기) AK홀딩스(006840)는 재정적... 자본보유 결정에... 6	AK홀... 등... 006840	10:15		

[그림 20] 크롤링 예시(크롤링 코드, 종목주가, 기사)

종목별 주가와 종목 뉴스를 크롤링 후에는 날짜와 종목코드를 기반으로

Merge 하고, 중복된 기사는 제거하였다.

[그림 21] 주가와 기사 통합 (merge 활용)

실험 데이터셋에서 레이블은 다음과 같이 정의하였다. 뉴스가 장전이나, 장중(3 시 40 분이전)에 나온 경우에는 전일 주가와 비교하였으며, 장마감후 나온 경우에는 익일 주가와 비교하여 뉴스의 영향도를 확인하였다. 2 가지 타입으로 Label 유형을 구분하였다. 첫번째 타입은 뉴스가 나온 이후 주가에 즉각적인 영향을 보기 위해, 0(뉴스 나온 후 상승/보합), 1(뉴스 나온 후 하락)로 Binary Data(2 개 클래스)로 정의한 타입이다. 그리고 두번째 타입은 뉴스로 인한 지속적 하락(1)인지, 일시적 하락(2)인지, 그렇지 않은지(0)로 Multi Categorical Data로 정의하였다. (3 개 클래스로 구분), 타입 1 를 활용하여 가설(가), 가설(나), 가설(다)을 확인하고, 타입 2 를 활용하여 가설(라)를 검증하는데 활용했다. 그리고 주가 종합주가 지수에 따른 영향을 본 연구에서는 고려하지 않았다. 예를 들어, 종합주가지수 전체가 떨어지는

경우에 좋은 뉴스가 나오더라도 종합주가지수에 따라 하락하는 경우가 발생할수 있어, 일종의 노이즈가 될 수 있지만, 본 연구에서는 다른 연구 대비하여 대량 종목 데이터를 이용하고 우선적으로 기존 연구보다 좋은 성능이 나오는지를 검증하는 것이므로, 일단 종합주가지수에 따른 변동성은 고려하지 않기로 하고, 향후 추가 성능향상을 위한 방향으로 제시하였다.

[표 7] Label 유형 – 2가지 Type

레이블	0	1	2	비고
타입 1	익일/당일 주가 상승/보합	익일/당일 주가 하락	–	Binary Classification
타입 2	익일/당일주 가 상승/보합	익일/당일 주가 하락 후 3일후 추가 하락	익일/당일 주가 하락 후 3일후 원복	Multi Classification

우선 상기의 Label 과 Feature 데이터를 기반으로 4 가지 유형의 데이터셋으로 만들었다. 구축된 데이터셋은 다음과 같다. 일관성을 가지고 학습과 테스트를 위해 동일한 셋으로 구분하였으며, 테스트 셋은 10% 로 추출하였다.

[표 8] 실험을 위한 데이터셋 유형

데이터셋	내용	활용 용도(가설)
Train_full.txt Test_full.txt	-레이블: 익일하락(1), 보합/상승(0), -Features: <기사제목> + <언론사> + <기자> + <기사 원문(3 문장)>	가설 (가), 가설(나) 검증
Train_small1.txt Test_small1.txt	-레이블: 익일하락(1) 보합/상승(0) -Features: <기사제목> 만	가설 (다) 검증
Train_small2.txt Test_small2.txt	-레이블: 익일하락(1) 보합/상승(0) -Features: <기사제목>+<언론사>	가설 (다) 검증
Train_multi.txt Test_multi.txt	-레이블: 보합/상승(0), 하락 후 상승(2), 지속하락(1) -Features: 기사제목+언론사+기자 + 기사원문 (3 문장)	가설(라) 검증

text	label
대출제약, 400억 규모 자사주 처분...미래먹거리 투자. (머니투데이) [머니투데이 김근희 기자] 「대출, 대출제약 주식취득 결의...지분 47.7%로 증가! 대출제약 전경/사장-대출제약대대출제약은 18일 개최된 이사회에서 자사주 30만513주를 처분하기로 결의하고 미래성장동력을 확보에 나선다고 밝혔다. 또 대놓은 자회사 대출제약의 주식 취득을 결의했다. 대출제약은 지주회사인 대중에 전체 지분의 약 2.6%를 처분하고 400억원 규모의 현금 유동성을 확보했다.」	1
롯데칠성음료, 페트병 자체 생산한다. (파이낸셜뉴스) 롯데칠성음료 안선풍장(파이낸셜뉴스) 롯데칠성음료는 주요 음료포장용기인 페트(PET) 공병의 자체생산률을 높여 생산능률을 증대 및 원가경쟁력을 강화해 나선다. 롯데칠성음료는 지난 5일 이사회를 열고 롯데알미늄의 페트사업 일부에 대한 영업양수 인건을 실의·의결했다. 페트사업에 대한 영업양수대상은 페트 자가생산을 위한 롯데알미늄의 인적 및 물적자산이며 양수대금은 68억5000만원이다. 1	1
아이에스통신, 폐기물 사업 더 키운다.. '코엔텍' 인수(상보). (머니투데이) [머니투데이 구경민 기자] 「우선협상대상자로 선정; 국내 중견건설업체 아이에스통신(도풀서)가 산업폐기물을 처리업체인 코엔텍의 우선협상대상자로 선정됐다. 지난해 건설폐기물 1위업체 인선이엔티에 이은 폐기물 업체 인수로 폐기물 사업의 보폭을 넓힐 수 있게 됐다. 아이에스통신은 4일 코엔텍 및 새한환경의 매도자인 맥쿼리코리아오퍼튜니티즈운용(맥쿼리오)과 도풀서-E&F, 블라이빗애워틴(BE) 컨소시엄이 주식매매 계약을 체결했다고 공시했다.」	0
종로구립도서관, 원스토어 굿파트너 2020 선정. (머니투데이) [머니투데이 김건우 기자] 종로구립도서관이 지난해 출시한 모바일 MMORPG '단성: 별을 살린 자 (이하 단성)' 등의 타이틀로 모바일 앱마켓인 원스토어가 수여하는 '원스토어 굿파트너 2020(GOOD PARTNER 2020)'에 선정됐다고 4일 밝혔다. 원스토어가 선정하는 '원스토어 굿파트너 2020'은 2020년 한 해를 빛낸 개인사를 선정해 개인 사업의 동반자로서 수여하는 상이다. 원스토어는 지난해 '단성' 등 총 3종의 신작 타이틀을 출시한 종로구립도서관 2020년을 빛낸 개인사로 선정했다. 0	0
대우건설 지난해 해외수주 6조원 육박...실적 턴어리운드 기대. (서울경제) [서울경제] 대우건설(047040)이 2020년 해외에서 6조 원에 육박하는 규모의 신규 수주를 기록했다. 지난해 목표로 삼았던 누적 수주 5조 696억 원을 초과 달성한 성적이다. 대우건설은 지난 12월 31일 이 랙크 일 포 신향만 사업(조감도) 후속공사로 5건, 2조9,000억 원 규모의 신규 수주를 수의로 계약하며 2020년 총 11건, 5조 8,624억 원의 신규 수주를 기록했다고 6일 밝혔다. 0	0

[그림 22] 데이터셋 1변타입(기사전문포함) 예시

레이블	기사요약
1	대웅제약, 400억 규모 자사주 처분...미래먹거리 투자
1	롯데칠성음료, 페트병 자체 생산한다
0	아이에스풀서, 폐기물 사업 더 키운다..‘코엔텍’ 인수(상보)
0	풀무코리아, 원스토어 오피파트너 2020 선정
0	대우건설 지난해 해외수주 6조원 유행...실적 텁여라운드 기대
0	[ENRASEI] 대상풀디스(084690), 52주 신고가 경신...10...
0	LG 아예 비건 앤티비에징 ‘얼티밋’ 라인 신규 출시
0	[단독] ‘C-쇼크’ 하니투어, 희망퇴직 이어 ‘복직 보장없는’ 안식년 ...
0	포스코-행복열라이언스...행복도시락‘배달’
1	CJ? 신세계·네이버 진짜 신무기는 ‘SME 챕봇’
0	[ENRASEI] 부산산업, 5.28% 오르며 거래량 증가
1	나노엔텍, 지난해 영업이 34억...전년 비 11.5% ↑
-	최근 나스닥 전기차 스타트업 투자가 시세

[그림 23] 데이터셋 2번 타입(타이틀만) 예시

3.3.2. 실험 방법

수집된 데이터를 기반으로 각각의 가설들을 검증하기 위해 다음과 같은 환경을 구축하였다.

[표 9] 실험 환경

구분	실험 환경
HW	구글 Colab pro 활용(GPU, cuda 및 고용량 RAM)
SW	코랩 제공 파이썬, 텐서플로우, 파이토치 활용

The screenshot shows a Google Colab interface with several code cells and a configuration dialog. The code cells contain Python code related to LSTM and sequence processing. A configuration dialog titled '노트 설정' (Notebook Settings) is open, showing options for GPU usage and other notebook configurations. The browser tab bar includes links to Google Drive, Colab, and various external websites.

[그림 24] 실험환경 (구글 코랩 파이썬 코드 및 Training 환경)

모델링은 텐서플로우와 파이토치를 활용하고, Github 의 오픈 소스들을 최대한 활용하여 모델들을 구현하고 검증하였다. 각 가설을 검증하기 위해 구현한 코드와 세부 내용은 다음과 같다.

가설 (가) ‘Feature Engineering 없이 종목 뉴스기사를 통해 직접적으로 하락을 예측할 수 있다.’에 대한 실험 설계

이를 검증하기 위해 다음과 연구 단계별로 구체적인 계획을 수립하였다. Feature 가 되는 뉴스는 공매도 거래가 있는 종목들을 추려내어, 포털 종목뉴스 1년치를 크롤링하였으며, 주가는 뉴스가 나온 날과 익일 변동으로 데이터를 수집하여 분석하였다. “주가 경향 예측 모델의 공정한 성능 평가 방법” 연구에 의하면, 주가 자체의 예측보다는 미래 주가 경향 예측이 수익에 도움을 준다고 하여, 주가 보다는 경향, 즉 하락(레이블: 1) 할지 상승 또는 보합 할지(레이블:0)로 결정하였다[19]. 세부적인 단계별 실험 설계 내용은 다음 표에 자세히 설명하였다.

[표 10] 단계별 실험 설계

단계	세부계획 및 프로그래밍 내용
데이터 수집	<ul style="list-style-type: none">-Feature(텍스트 데이터): 파이썬 코딩을 통해 크롤링한 기사 활용-Label(분류데이터): 종목별 뉴스 출현 날짜의 주가 기반 하락 또는 상승/보합 여부<ul style="list-style-type: none">- 중복 및 Null 값 제거: 611 개 중 22 개 종목은 뉴스데이터가 없어 제거 (총 45,431 개 뉴스 데이터 활용)- Label 분포:

	<p>하락(1), 17,662 개, 보합내지 상승(0) 27,769 개</p> <p>-<언론사> 분포</p> <p>이데일리 (8,957), 파이낸셜뉴스(6,662), 한국경제 (6,439), 아시아경제(6,120), 머니투데이(4,785), 매일경제(3,771), 서울경제(3,402), 헤럴드경제(3,226), 조선비즈(1,338), 조세일보(731)</p> <table border="1"> <thead> <tr> <th>언론사</th><th>0</th><th>1</th><th>하락비율</th></tr> </thead> <tbody> <tr><td>이데일리</td><td>5213</td><td>3744</td><td>42%</td></tr> <tr><td>한국경제</td><td>4610</td><td>1829</td><td>28%</td></tr> <tr><td>파이낸셜뉴스</td><td>4213</td><td>2449</td><td>37%</td></tr> <tr><td>아시아경제</td><td>3650</td><td>2470</td><td>40%</td></tr> <tr><td>머니투데이</td><td>2785</td><td>2000</td><td>42%</td></tr> <tr><td>매일경제</td><td>2238</td><td>1533</td><td>41%</td></tr> <tr><td>서울경제</td><td>1995</td><td>1407</td><td>41%</td></tr> <tr><td>헤럴드경제</td><td>1883</td><td>1343</td><td>42%</td></tr> <tr><td>조선비즈</td><td>782</td><td>556</td><td>42%</td></tr> <tr><td>조세일보</td><td>400</td><td>331</td><td>45%</td></tr> </tbody> </table>	언론사	0	1	하락비율	이데일리	5213	3744	42%	한국경제	4610	1829	28%	파이낸셜뉴스	4213	2449	37%	아시아경제	3650	2470	40%	머니투데이	2785	2000	42%	매일경제	2238	1533	41%	서울경제	1995	1407	41%	헤럴드경제	1883	1343	42%	조선비즈	782	556	42%	조세일보	400	331	45%
언론사	0	1	하락비율																																										
이데일리	5213	3744	42%																																										
한국경제	4610	1829	28%																																										
파이낸셜뉴스	4213	2449	37%																																										
아시아경제	3650	2470	40%																																										
머니투데이	2785	2000	42%																																										
매일경제	2238	1533	41%																																										
서울경제	1995	1407	41%																																										
헤럴드경제	1883	1343	42%																																										
조선비즈	782	556	42%																																										
조세일보	400	331	45%																																										
입력 데이터 전처리	<p>-기사의 경우, Word2Vec 사용시 Regular Expression 실시</p> <pre>test['text'] = test['text'].str.replace("[^ㄱ-ㅎㅏ-ㅣ가-힣A-Za-z0-9]","") train['text'] = train['text'].str.replace("[^ㄱ-ㅎㅏ-ㅣ가-힣A-Za-z0-9]","")</pre> <p style="text-align: center;">[그림 25] 전처리 코드 예시</p> <table border="1"> <thead> <tr> <th>id</th> <th>text</th> <th>label</th> </tr> </thead> <tbody> <tr><td>0 159</td><td>BGF리테일 3분기 영업익 637억원 전년 17 머니투데이 머니투데이 이재은 기자 ...</td><td>0</td></tr> <tr><td>1 7788</td><td>금호석화 금호리조트 부가가치 창출활 신사업인사 단행 한국경제 대표이사에 김성일 금호...</td><td>0</td></tr> <tr><td>2 15392</td><td>대웅제약 에볼루스가 메디톡스애브비에 풀 합의금로열티 일부 부담기로 매일경제 미국 국...</td><td>0</td></tr> <tr><td>3 9557</td><td>GC녹십자엠에스에 진단키트 2900억원 규모 수출 아시아경제 아시아경제 조현의 기...</td><td>0</td></tr> <tr><td>4 43045</td><td>자사주 소각무상증자 혐대열리베이터 상한가 매일경제 악재 터지자 주주 보호 나서 남북...</td><td>0</td></tr> </tbody> </table> <p style="text-align: center;">[그림 26] 전처리 결과 예시</p>	id	text	label	0 159	BGF리테일 3분기 영업익 637억원 전년 17 머니투데이 머니투데이 이재은 기자 ...	0	1 7788	금호석화 금호리조트 부가가치 창출활 신사업인사 단행 한국경제 대표이사에 김성일 금호...	0	2 15392	대웅제약 에볼루스가 메디톡스애브비에 풀 합의금로열티 일부 부담기로 매일경제 미국 국...	0	3 9557	GC녹십자엠에스에 진단키트 2900억원 규모 수출 아시아경제 아시아경제 조현의 기...	0	4 43045	자사주 소각무상증자 혐대열리베이터 상한가 매일경제 악재 터지자 주주 보호 나서 남북...	0																										
id	text	label																																											
0 159	BGF리테일 3분기 영업익 637억원 전년 17 머니투데이 머니투데이 이재은 기자 ...	0																																											
1 7788	금호석화 금호리조트 부가가치 창출활 신사업인사 단행 한국경제 대표이사에 김성일 금호...	0																																											
2 15392	대웅제약 에볼루스가 메디톡스애브비에 풀 합의금로열티 일부 부담기로 매일경제 미국 국...	0																																											
3 9557	GC녹십자엠에스에 진단키트 2900억원 규모 수출 아시아경제 아시아경제 조현의 기...	0																																											
4 43045	자사주 소각무상증자 혐대열리베이터 상한가 매일경제 악재 터지자 주주 보호 나서 남북...	0																																											
학습 데이터 구분	<p>-학습 데이터 및 테스트데이터 10%로 구분 (학습데이터는 모든 모델에서 동일)</p> <p>-Validation set 도 학습데이터 중에서 10% 설정</p> <pre># 훈련셋과 검증셋으로 분리 train_inputs, validation_inputs, train_labels, validation_labels = train_test_split(input_ids, labels, random_state=2018, test_size=0.1)</pre> <p style="text-align: center;">[그림 27] 학습데이터셋과 Validation셋 분리</p>																																												

데이터 임베딩	<p>-Word2Vec 기반 LSTM 의 경우, Word2Vec 으로 기 학습된 ko.bin 을 활용하여 임베딩 실시 (입력데이터들은 Mecab 을 토크나이징 실시)</p> <pre>path = '/content/gdrive/MyDrive/주식분석_NLP/ko.bin' ko_vec = Word2Vec.load(path) embedding_matrix = np.random.rand(vocab_size_ko_vec, word_vector_dim_ko_vec) for i in range(4,vocab_size_ko_vec): if index_to_word[i] in ko_vec: embedding_matrix[i] = ko_vec[index_to_word[i]]</pre> <p>[그림 28] Word2Vec 기준 학습된 모델 활용 임베딩 코드</p> <p>-BERT 와 KoBERT 는 자체 제공하는 WordPiece Tokenize 로 토큰화된 것으로 임베딩함</p> <pre># BERT의 토크나이저로 문장을 토큰으로 분리 tokenizer = BertTokenizer.from_pretrained('bert-base-multilingual-cased', do_lower_case=False) tokenized_texts = [tokenizer.tokenize(sent) for sent in sentences] # 토큰을 숫자 인덱스로 변환 input_ids = [tokenizer.convert_tokens_to_ids(x) for x in tokenized_texts] # 문장을 MAX_LEN 길이에 맞게 자르고, 모자란 부분을 패딩 0으로 채움 input_ids = pad_sequences(input_ids, maxlen=MAX_LEN, dtype="long", truncating="post", padding="post")</pre> <p>[그림 29] WordPiece Tokenize 및 임베딩 코드</p>
모델링 및 학습	<p>-파이토치와 텐서플로우를 통해 LSTM, BERT, KoBERT 구현 적용 (10 번의 epoch, 256 이하의 batch 사이즈로 학습)</p> <pre># 분류를 위한 BERT 모델 생성 model = BertForSequenceClassification.from_pretrained("bert-base-multilingual-cased", num_labels=2) # GPU가 있다면 다음의 cuda 함수를 불러서 사용한다 model.cuda()</pre>

```

for epoch_i in range(0, epochs):
    total_loss = 0
    model.train()
    for step, batch in enumerate(train_dataloader):
        if step % 500 == 0 and not step == 0:
            elapsed = format_time(time.time() - t0)
            batch = tuple(t.to(device) for t in batch)
            b_input_ids, b_input_mask, b_labels = batch
            outputs = model(b_input_ids,
                            token_type_ids=None,
                            attention_mask=b_input_mask,
                            labels=b_labels)
            loss = outputs[0]
            total_loss += loss.item()
            loss.backward()
            torch.nn.utils.clip_grad_norm_(model.parameters(), 1.0)
            optimizer.step()
            scheduler.step()
            model.zero_grad()
    avg_train_loss = total_loss / len(train_dataloader)

```

[그림 30] BERT 모델생성 및 학습을 위한 코드

```

# LSTM
lstm = keras.Sequential()
lstm.add(keras.layers.Embedding(vocab_size,
                                word_vector_dim,
                                embeddings_initializer=Constant(embedding_matrix),
                                input_length=maxlen,
                                trainable=True))
lstm.add(keras.layers.LSTM(128))
lstm.add(keras.layers.Dense(128, activation='relu'))
lstm.add(keras.layers.Dense(1, activation='sigmoid'))

epochs = 10

lstm.compile(optimizer='adam',
             loss='binary_crossentropy',
             metrics=['accuracy'])

history_lstm = lstm.fit(partial_X_train,
                        partial_y_train,
                        epochs=epochs,
                        batch_size=512,
                        validation_data=(X_val, y_val),
                        verbose=1)
cnn.compile(optimizer='adam',
            loss='binary_crossentropy',
            metrics=['accuracy'])

```

[그림 31] Word2Vec 기반 LSTM 모델생성 및 학습

평가	-10 epoch 후 Test Accuracy(전체의 10%)로 평가 (최종)
----	---

비교는 Test Accuracy = (예측값과 실제값이 동일한셋)/(전체 테스트셋))

```
results_lstm = lstm.evaluate(X_test, y_test, verbose=2)
print(results_lstm)
```

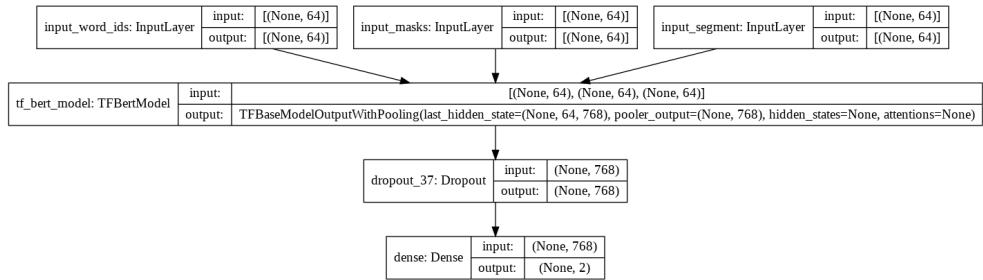
[그림 32] Word2Vec기반 LSTM 테스트셋 평가 코드

```
# 평가모드로 변경
model.eval()
eval_loss, eval_accuracy = 0, 0
nb_eval_steps, nb_eval_examples = 0, 0

for step, batch in enumerate(test_dataloader):
    if step % 100 == 0 and not step == 0:
        elapsed = format_time(time.time() - t0)
        batch = tuple(t.to(device) for t in batch)
        b_input_ids, b_input_mask, b_labels = batch
        with torch.no_grad():
            outputs = model(b_input_ids,
                            token_type_ids=None,
                            attention_mask=b_input_mask)
        logits = outputs[0]
        logits = logits.detach().cpu().numpy()
        label_ids = b_labels.to('cpu').numpy()
        tmp_eval_accuracy = flat_accuracy(logits, label_ids)
        eval_accuracy += tmp_eval_accuracy
        nb_eval_steps += 1
    print("Accuracy: {:.2f}%".format(eval_accuracy/nb_eval_steps))
```

[그림 33] Transformer 기반 BERT 테스트셋 평가 코드

KoBERT 모델은 다음과 같이 64 개의 Dimension 을 가진 입력을 받고, Pre-Trained Layer 를 가진 형태로 모델링된다. BERT 모델 역시, KoBERT 에서 KoBERT 대신 BERT 를 사용한 모델로 차이는 임베딩 Layer 로써 나머지는 동일하게 실험에 활용하였다.



Model: "model"

```

Layer (type) Output Shape Param # Connected to
=====
input_word_ids (InputLayer) [(None, 64)] 0
input_masks (InputLayer) [(None, 64)] 0
input_segment (InputLayer) [(None, 64)] 0
tf_bert_model (TFBertModel) TFBASEMODELOUTPUTWITHPOOLING 92186880 input_word_ids[0][0] input_masks[0][0]
input_segment[0][0]
dropout_37 (Dropout) (None, 768) 0 tf_bert_model[0][1]
dense (Dense) (None, 2) 1538 dropout_37[0][0]
=====
Total params: 92,188,418 Trainable params: 92,188,418 Non-trainable params: 0

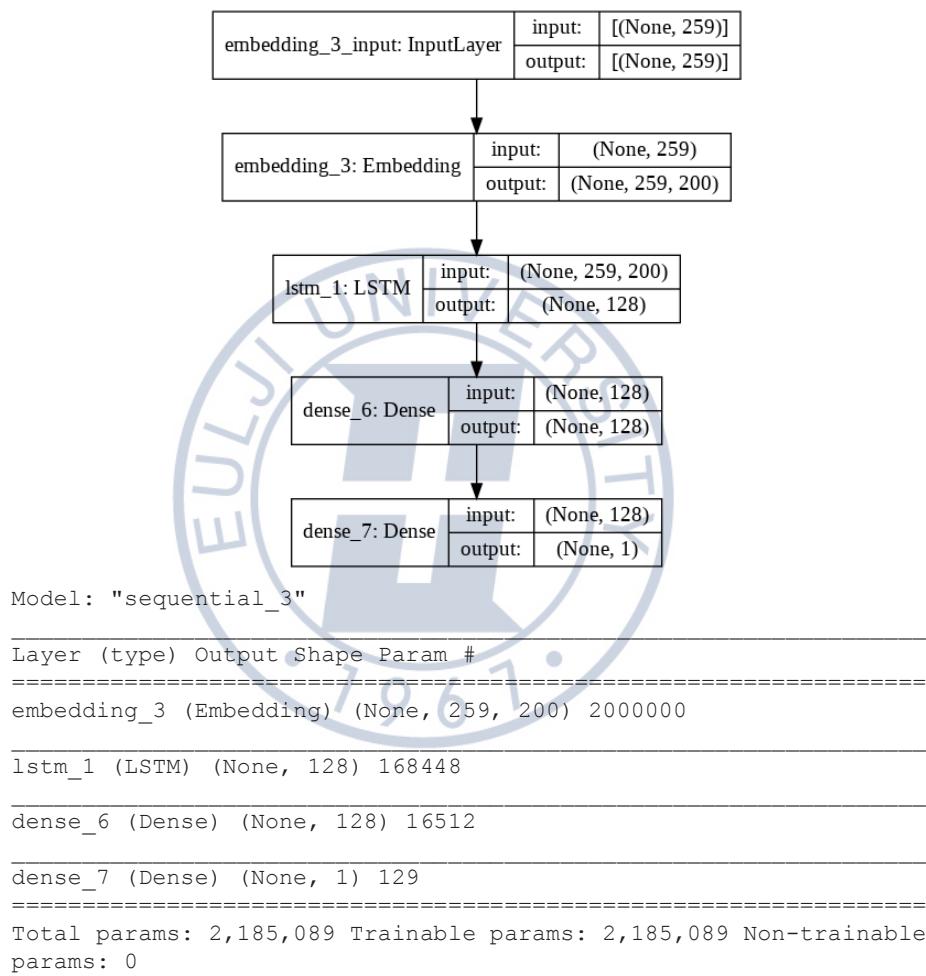
```

[그림 34] KoBERT 실험 모델 Summary (Class 2개)

가설 (나) ‘한국어 특화 KoBERT 가 가장 성능이 좋을 것이다.’에 대한
실험 설계

본 가설은 (가)의 가설 검증 계획과 동일한 데이터 셋과 동일한 코드를 이용하며, 동일한 조건하에서 모델링하고 학습하여, Accuracy로 비교 분석하였다. Attention 기반의 Transformer는 LSTM 대비 병렬적 처리 및 긴 문장에 대한 성능이 좋아, 기사 원문을 추가할 경우, LSTM 대비 Transformer의 성능이 좋을 것으로 예상되었다. 단, 짧은 문장(제목)의 경우에는 Word2vec 기반의 LSTM이 attention 기반의 BERT 보다는

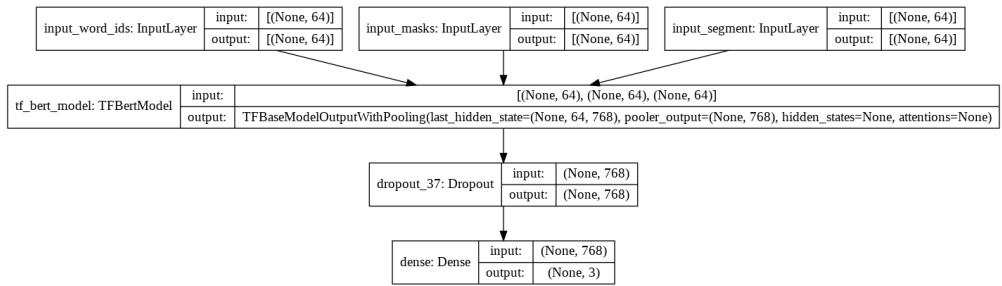
높은 성능이 예상되었으며, 국내 통신사에서 개발한 KoBERT는 전세계 언어를 사전 학습한 BERT 보다 좀 더 한국어에 특화되어 있어, 긴 문장에서는 BERT 보다 훨씬 좋은 성능이 가정하고 실험을 진행하였다.



가설 (다) ‘[기자]와 [언론사] Feature 추가 시 성능향상’에 대한 실험 설계

앞의 가설 2 가지([가], [나])는 뉴럴언어모델들을 가지고 최적화시키는 것이라면, 이번 가설은 Feature 추가에 따른 성능 향상 여부를 확인하는 과정이다. 앞서 활용한 Feature 는 <기자명>, <언론사명>, <제목>, <기사 본문 3 줄> 이 포함된 데이터셋을 이용하였다. 검증하고자 하는 것은 <기사> 만으로 텍스트를 구성하여 학습한 경우와, 이외의 Feature 를 텍스트에 포함시켜 학습할 경우, 학습 결과가 다르고, 더욱 성능이 좋아질 것이라는 것을 확인하였다.

Feature 를 추가하는 방법은 <기사>안에 <언론사>와 <기자> 단어를 추가하여 합쳐서 학습시키는 것이다. 즉, 추가 Feature 들을 본문 텍스트와 함께 임베딩하여 학습시키는 형태이다. 만약 언론사와 기자가 주요한 Feature 로 학습시에 확인이 된다면 자연스럽게 Feature vector 로 만들어질 것이기 때문에 향후 텍스트 기반으로 추가적인 Feature, 예를 들어 남/녀, 지역 등의 Feature 를 추가하여 문제가 정의된 경우, 수월하게 text 에 추가함으로써 모델링이 가능함으로 시사할 수 있을 것이다. 각각의 테스트 정확도를 구하고, 이에 따라, 통계적으로 유의한 차이가 있는지 확인하였다.



[그림 36] KoBERT 실험 모델 (Class 3개)

가설 (라) ‘단기적 복구 패턴을 통해 의도적으로 쓴 기사와 악재성 기사에 대한 구분’에 대한 실험 설계

그랜빌의 법칙에 의하면, 3:3 법칙이라는 것이 있다. 20 일 이동평균 밀으로 떨어졌지만, 3 일 이내, 그리고 3%이내의 조정이 마무리된다면 기존의 상승 추세는 유효하다는 의미이다. 이 법칙을 활용한다면, 진정한 악재가 아닌 악의성 기사의 경우, 3 일안에 다시 원복 될 수 있다고 가정하고자 한다. 학습을 위해, 3 일 이후에도 지속적으로 떨어질 경우는 진정한 기업의 악재라고 판단하고자 하며, 그렇지 않고 3 일 이내에 원복한다면 악의적인 기사유형으로 시장에 지속적으로 반영하지 않는 소문 정도로 취급할 수 있을 것으로 간주한다. 결국, 상승 흐름에서 기업의 큰 악재가 없음에도 단지 언론사의 추측성(일종의 찌라시) 기사로 인해, 손실보고 매도함으로써 바로 원복하게 됨으로 투자자 및 피해 기업의 손실을 최소화하기 위함이다.

이를 위해 레이블은 <타입 2>의 3 개의 클래스를 구성된 데이터셋을 활용하였고, Binary 에서 Multi 클래스로 변경됨으로 각각의 모델을 Binary_Cross_Entropy 대신에 Sparse_Categorical_Crossentropy 를 활용했다. 그리고 아웃풋레이어에서 Sigmoid 함수 대신에 Softmax 함수로써 다중 클래스의 함수를 구하는 코드로 변경하여 활용했다.

```

lstm.add(keras.layers.LSTM(128))
lstm.add(keras.layers.Dense(128, activation='relu'))
lstm.add(keras.layers.Dense(3, activation='softmax'))

lstm.compile(optimizer='adam',
              loss=keras.losses.SparseCategoricalCrossentropy(),
              metrics=['accuracy'])

```

[그림 37] Multi-Class로 할 경우 변경된 코드 (LSTM Case)

```

stock_drop = tf.keras.layers.Dropout(0.1)(bert_outputs)

stock_first = tf.keras.layers.Dense(train_df['label'].nunique(), activation='softmax',
                                    kernel_initializer=tf.keras.initializers.TruncatedNormal(stddev=0.02))(stock_drop)
#stock_first = tf.keras.layers.Dense(train_df['label'].nunique(), activation='sigmoid',
#                                    kernel_initializer=tf.keras.initializers.TruncatedNormal(stddev=0.02))(stock_drop)
stock_model = tf.keras.Model([token_inputs, mask_inputs, segment_inputs], stock_first)

stock_model.compile(optimizer=opt, loss="sparse_categorical_crossentropy", metrics=["accuracy"])
#stock_model.compile(optimizer=opt, loss="binary_crossentropy", metrics=["accuracy"])

```

[그림 38] Multi-Class로 할 경우 변경된 코드 (KoBERT Case)

각각의 모델에 대해 Full data 를 적용한 Test Accuracy 통해 성능이 나오는지 확인하고, 기존 연구들의 테스트 정확도 평균인 66%보다도 통계적으로 유의한지를 확인하였다.

4. 실험 및 결과분석

4.1. 실험 데이터셋

실험방법에 따라, 다음과 같은 데이터셋을 파일 코딩을 통해 준비하였다. 총 4 종류로 Test 셋과 Train 셋으로 구분하여 text 파일로 저장하였다. 특히 Full Set 의 경우 기사 원문까지 포함되어 있어 수십 메가의 용량을 차지하고 있음을 확인할 수 있다.

이름	소유자	마지막으로...	↑	파일 크기
train_full.txt	나	오후 10:45	20MB	
test_full.txt	나	오후 10:45	2MB	
test_small1.txt	나	오후 10:45	331KB	
train_small1.txt	나	오후 10:46	3MB	
test_small2.txt	나	오후 10:46	415KB	
train_small2.txt	나	오후 10:46	4MB	
train_multi.txt	나	오후 10:47	20MB	
test_multi.txt	나	오후 10:47	2MB	

[그림 39] 실험을 위한 데이터셋

그리고 다음과 같은 4 가지 모델에 대해 평가 결과를 코드에 함께 저장하기 위해 각각 다른 파일로 코딩하여 저장하였다. 총 3 개 모델, 4 가지 데이터 타입으로 총 12 개의 코드가 있으며, 추가로

Regular Expression 을 적용했을 경우에도 성능이 개선 여부를 확인하기

위한 실험을 위해 2 개의 RE 적용 파일을 추가하였다.



이름	소유자	마지막으로 수정한 날짜	파일 크기
01_word2vec_LSTM_3class_4_13.ipynb	나	2021. 4. 13. 나	72KB
01_word2vec_LSTM_full_4_13.ipynb	나	2021. 4. 13. 나	75KB
01_word2vec_LSTM_title_newspaper_4_12.ipynb	나	2021. 4. 13. 나	79KB
01_word2vec_LSTM_title_only_4_12.ipynb	나	2021. 4. 13. 나	89KB
02_bert_3class_4_13.ipynb	나	2021. 4. 13. 나	164KB
02_bert_full_4_13.RE없이진행.ipynb	나	2021. 4. 13. 나	146KB
02_bert_full_4_13.RE로_진행.ipynb	나	오후 9:40 나	98KB
02_bert_title_newspaper_4_13.ipynb	나	2021. 4. 13. 나	131KB
02_bert_title_only_4_13.ipynb	나	2021. 4. 13. 나	130KB
03kobert_3class_tf활용_4_13.ipynb	나	오후 10:33 나	108KB
03kobert_full_tf활용_4_13.ipynb	나	오후 7:24 나	70KB
03kobert_full_tf활용_4_13.ipynb	나	오후 11:13 나	102KB
03kobert_title_newspaper_tf활용_4_13.ipynb	나	오후 8:29 나	95KB
03kobert_title_only_tf활용_4_13.ipynb	나	오후 8:34 나	115KB

[그림 40] 파이썬 코드 파일들

4.2. 실험환경

파이썬 프로그램들은 구글 코랩을 통해 코딩하고 실행하여 테스트 Accuracy 를 도출하였다. 각 프로그램별 학습시간은 다음이 같이 총 446 분 소요되었다. (10 epoch 기준) 이때 적용된 환경은 클라우드 환경인 구글 코랩에서 그들이 제공하는 GPU 와 고용량 RAM 환경을 활용하였다.

[표 11] 모델별 학습 소요시간

모델명	학습데이터 및 레이블	학습 소요시간
Word2Vec 기반 LSTM 모델	Feature (타이틀+기자+언론사+기사원문), Label (0/1)	2 분 6 초
Word2Vec 기반 LSTM 모델	Feature (타이틀+언론사), Label (0/1)	37 초
Word2Vec 기반 LSTM 모델	Feature (타이틀), Label (0/1)	37 초
Word2Vec 기반 LSTM 모델	Feature(타이틀+기자+언론사+기사원 문), Label(0/1/2)	3 분 47 초
Transformer 기반 BERT	Feature (타이틀+기자+언론사+기사원문), Label(0/1)	45 분 50 초
Transformer 기반 BERT	Feature (타이틀+언론사), Label (0/1)	77 분 40 초
Transformer 기반 BERT	Feature (타이틀), Label (0/1)	43 분 50 초
Transformer 기반 BERT	Feature (타이틀+기자+언론사+기사원문), Label (0/1/2)	77 분 40 초
한국어 특화 KoBERT	Feature (타이틀+기자+언론사+기사원문),	51 분 43 초

	Label (0/1)	
한국어 특화 KoBERT	Feature (타이틀+언론사), Label (0/1)	49 분 15 초
한국어 특화 KoBERT	Feature (타이틀), Label (0/1)	52 분 55 초
한국어 특화 KoBERT	Feature (타이틀+기자+언론사+기사원문), Label (0/1/2)	48 분 56 초
총소요시간		446 분 36 초

참고로 모델링하기전에 크롤링 시간은 주가와 611 여개의 관련종목별 기사 82,000 여개 수집하는데, 거의 18 시간 소요되어, 모델링 시간보다 훨씬 많은 시간이 소요되었다.

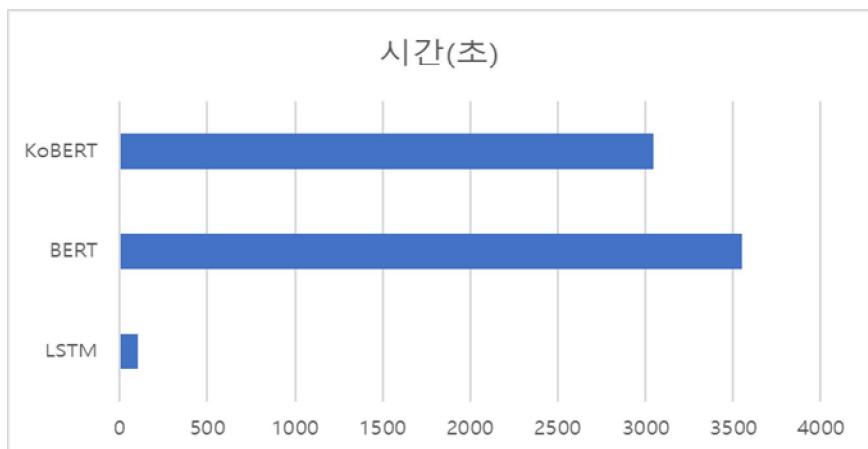
```

Epoch 1/10
WARNING:tensorflow:The parameters 'output_attentions', 'output_hidden_states' and 'use_cache' cannot be updated when calling a model. They
WARNING:tensorflow:The parameter 'return_dict' cannot be set in graph mode and will always be set to 'True'.
WARNING:tensorflow:The parameters 'output_attentions', 'output_hidden_states' and 'use_cache' cannot be updated when calling a model. They
WARNING:tensorflow:The parameter 'return_dict' cannot be set in graph mode and will always be set to 'True'.
639/639 [=====] - ETA: 0s - loss: 0.6737 - accuracy: 0.5644WARNING:tensorflow:The parameters 'output_attentions'
WARNING:tensorflow:The parameter 'return_dict' cannot be set in graph mode and will always be set to 'True'.
639/639 [=====] - 3185s 5s/step - loss: 0.6737 - accuracy: 0.5644 - val_loss: 0.6462 - val_accuracy: 0.6128
Epoch 2/10
639/639 [=====] - 3141s 5s/step - loss: 0.6298 - accuracy: 0.6212 - val_loss: 0.6371 - val_accuracy: 0.6201
Epoch 3/10
639/639 [=====] - 3172s 5s/step - loss: 0.6110 - accuracy: 0.6425 - val_loss: 0.6010 - val_accuracy: 0.6672
Epoch 4/10
639/639 [=====] - 3212s 5s/step - loss: 0.5544 - accuracy: 0.7112 - val_loss: 0.5663 - val_accuracy: 0.6995
Epoch 5/10
639/639 [=====] - 3172s 5s/step - loss: 0.4668 - accuracy: 0.7814 - val_loss: 0.5550 - val_accuracy: 0.7297
Epoch 6/10
639/639 [=====] - 3181s 5s/step - loss: 0.3630 - accuracy: 0.8453 - val_loss: 0.6126 - val_accuracy: 0.7455
Epoch 7/10
639/639 [=====] - 3182s 5s/step - loss: 0.2658 - accuracy: 0.8911 - val_loss: 0.6300 - val_accuracy: 0.7576
Epoch 8/10
639/639 [=====] - 3172s 5s/step - loss: 0.2006 - accuracy: 0.9196 - val_loss: 0.7115 - val_accuracy: 0.7660
Epoch 9/10
639/639 [=====] - 3244s 5s/step - loss: 0.1520 - accuracy: 0.9416 - val_loss: 0.7923 - val_accuracy: 0.7728
Epoch 10/10
639/639 [=====] - 3210s 5s/step - loss: 0.1241 - accuracy: 0.9537 - val_loss: 0.8472 - val_accuracy: 0.7744
CPU times: user 8d 15h 26min 42s, sys: 5h 37min 6s, total: 8d 21h 3min 49s
Wall time: 8h 51min 11s
<tensorflow.python.keras.callbacks.History at 0x7fa106d0a910>

```

[그림 41] 학습 모습 (KoBERT Full데이터 활용시)

각 모델별 평균 학습 시간은 다음과 같아], LSTM 이 2 분, BERT 가 60 분, KoBERT 가 51 분 정도로 속도면에서는 확실히 LSTM 이 빠른 것으로 확인 되었다.



[그림 42] 모델별 평균 학습시간



4.3. 실험 결과

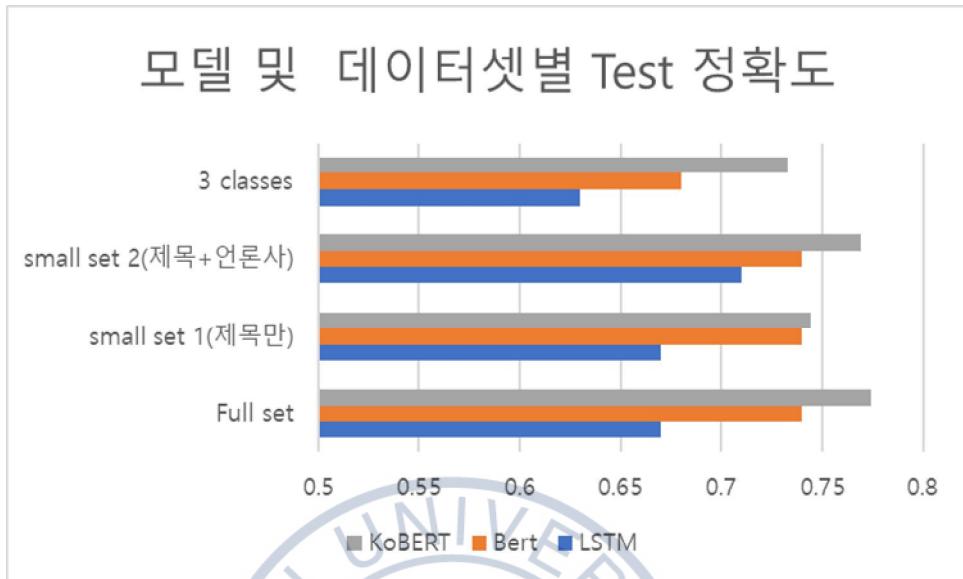
실험결과는 12 개의 실험케이스에 대해 Validation Accuracy 와 Test Accuracy 로 정리될 수 있다. 대부분, Validation Accuracy 보다 Test Accuracy 가 높거나 같은 결과로 봐서 Overfitting 은 없는 것으로 판단하였다. 결과는 다음과 같다. 세부 분석 내용은 다음 세션에서 고찰하고자 한다.

[표 12] Validation Accuracy – 결과

Validation Accuracy	LSTM	BERT	KoBERT
Full set	0.67	0.74	0.774
small set 1(제목만)	0.66	0.74	0.746
small set 2(제목+언론사)	0.7	0.74	0.769
3 classes	0.61	0.69	0.733

[표 13] Test Accuracy – 결과

Test Accuracy	LSTM	BERT	KoBERT
Full set	0.67	0.74	0.774
small set 1(제목만)	0.67	0.74	0.744
small set 2(제목+언론사)	0.71	0.74	0.769
3 classes	0.63	0.68	0.733



[그림 43] 모델 및 데이터 유형 별 Testing 결과 정확도

실험 결과, 각 가설에 대한 검증 결과는 다음과 같다. 각 연구 가설에 대한 통계적으로 유의성 검정을 위해 같은 모델이지만 다른 테스트셋을 가지고 Cross Validation(30 회 실시)를 통해 가설에 대해 검증을 실시하였다.

[표 14] 실험결과-연구가설별 검증 방법 및 판단 기준

연구 가설	판단 기준
가설(가) 소수의 종목 뉴스기사만으로 사전처리 없이 하락을 예측할 수 있다.	-통계적 가설(1-sample t-Test 실행) H0: KoBERT 평균 Accuracy = 0.66 H1: KoBERT 평균 Accuracy > 0.66 (Data Set: Binary Classification Set) - 비교 대상 : 이전 연구 결과의 평균 = 0.66
가설(나) KoBERT 의 성능이	-통계적 가설(2-sample t-test 실행) H0: KoBERT 평균 = LSTM 평균

기존 방식인 LSTM 보다 높다.	H0: KoBERT 평균 > LSTM 평균
가설 (다)주가추이 예측은 기자와 언론사등의 Feature 가 추가될수록 정확하다.	<ul style="list-style-type: none"> -통계적 가설 (ANOVA 분석 실시) H0: Data 1 평균=Data 2 평균=Data 3 평균 H1: 하나라도 평균이 같지 않다. - Data 1: 기사+언론사+기자 Data 2: 기사제목+언론사 Data 3: 기사제목
가설(라) 단기적 복구 패턴을 통해 악의적으로 쓴 기사와 악재성 기사는 예측가능 할 것이다.	<ul style="list-style-type: none"> -통계적 가설(1-sample t-Test 실행) - KoBERT 의 평균 정확도 > 66% H0: KoBERT 평균 Accuracy = 0.66 H1: KoBERT 평균 Accuracy > 0.66 (Data Set: 3 Classification Set) - 비교 대상 : 이전 연구 결과의 평균 = 0.66

5. 고 찰

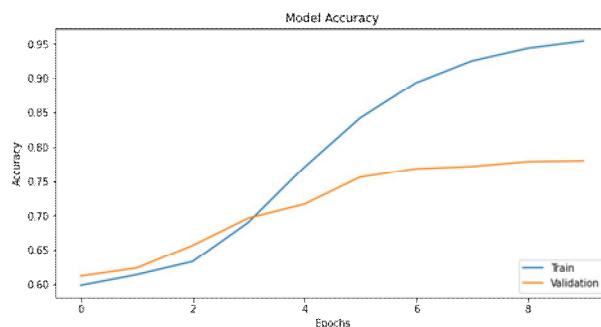
5.1. 가설 (가)에 대한 실험결과 분석

모델별 Accuracy 는 KoBERT 적용결과, 정확도가 78%로 상당히 양호한 수치를 보이고 있다. F1 Score 와 정밀도 모두, 72% 수준이다.

[표 15] KoBERT Confusion Matrix와 Classification Scores

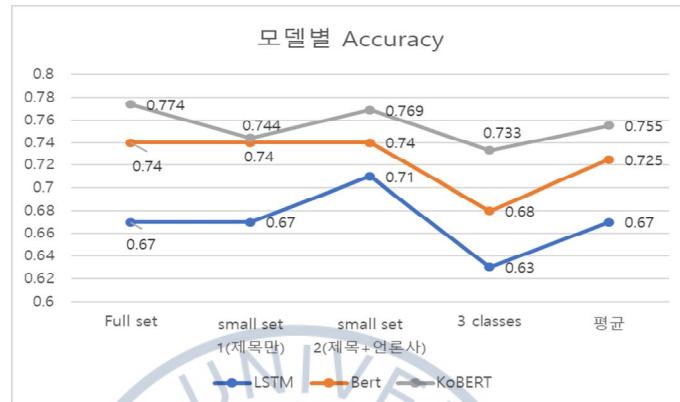
Confusion Matrix		예측	
실제	0	0	1
	1	496	1290
		precision	recall
0	0.82	0.82	0.82
1	0.72	0.72	0.72
accuracy			0.78
macro	0.77	0.77	0.77
weighted	0.78	0.78	0.78

그리고 10 번의 Epoch 으로 실험한 결과, 더 이상 Validation Accuracy 의 향상이 보이지 않음을 확인하였다.



[그림 44] Epoch별 Training/Validation Accuracy의 변화

이는 감성 지수 변환 없이, 텍스트로 이루어진 기사 만으로 임베딩을 통해서도 예측이 가능함을 명확히 보여지고 있는 것이다.



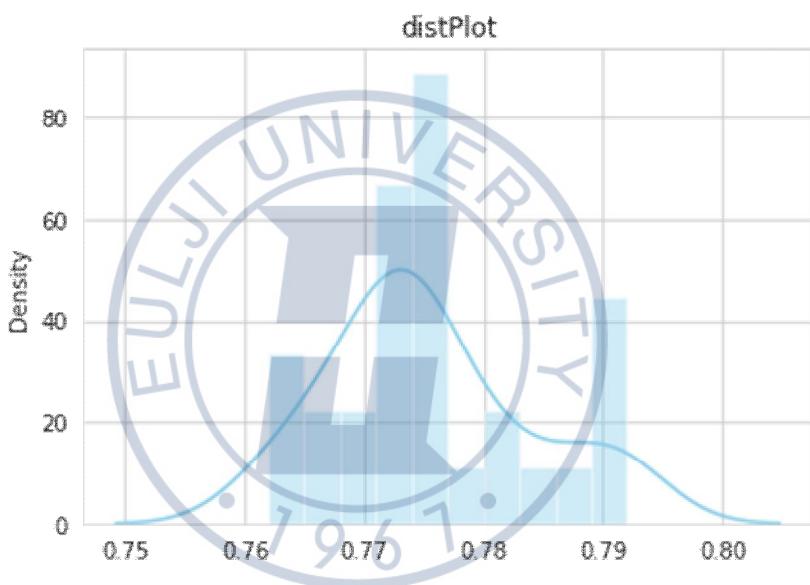
[그림 45] 각 모델별 정확도

그리고 감성지수화를 통한 기존 연구의 정확도와 딥러닝을 활용한 기존 연구의 정확도 대비 향상된 결과를 보이고 있다. 기존 연구 중에는 80%의 정확도 연구 결과도 있었지만, 이는 개별 종목 예측이 아닌 종합주가 지수 예측이기 때문에 기준이 다소 다르다.

[표 16] 이전 연구들의 결과 정확도 비교표

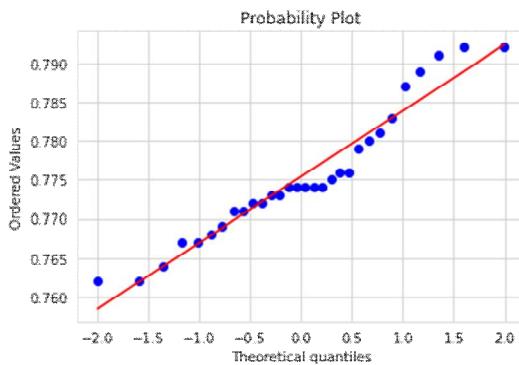
연구명	정확도	비고
뉴스와 주가: 빅데이터 감성분석을 통한 지능형 투자의 사결정모형	70%	Label:종합주가지수 Feature:뉴스감성지수
SNS 와 뉴스기사의 감성분석과 기계학습을 이용한 주가예측 모형 비교 연구	80%	Label:7 개 기업 주가 Feature:뉴스+SNS 감성지수
SNS 감성 분석을 이용한 주가 방향성 예측	50%	Label:20 개 기업 Feature:기사텍스트
시스템적인 군집 확인 과 뉴스를 이용한 주가 예측,	65%	Label:30 개 기업 Feature:관련기사감성지수
기존 연구들의 평균 Accuracy	66%	

무엇보다도 기존연구들의 평균인 66% 대비하여 본 연구 모델 중 가장 높은 Accuracy 를 보이고 있는 KoBERT 모델(데이터셋은 제목+언론사+기자+기사원문)에 대해 테스트셋을 달리하면서 30 번 Cross Validation 를 실시하였다. 그 결과, 다음과 같이 평균 0.775 의 Accuracy 를 가지고 있는 분포를 얻을 수 있었다.



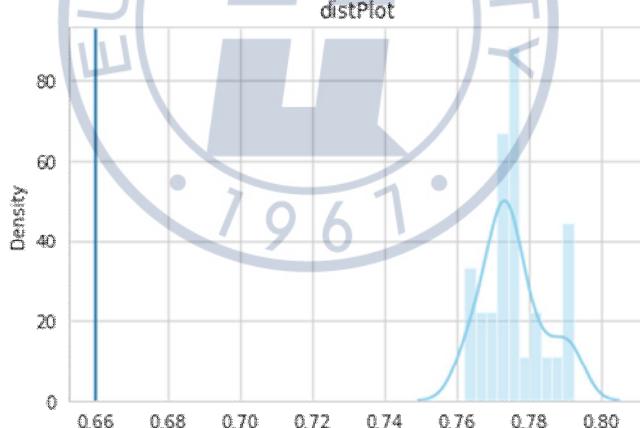
[그림 46] KoBERT Accuracy에 대한 Cross Validation 통계(30회)

통계적으로 기존 연구들의 평균치인 0.66 보다 큰지를 1-Sample t test 를 통해 검증한 결과, 기본 가정인 정규성은 만족($p\text{-value} = 0.059$)하고 있었으며, $p\text{-value}$ 가 $2.10\text{e-}11$ 로 0.05 보다 훨씬 작기에, 평균이 0.66 과 같다는 귀무가설을 기각하기에 충분하였다(즉, 기존 연구들의 Accuracy 보다는 유의미하게 크다고 할수 있다.).



- H0: 표본의 모집단이 정규분포를 이루고 있다.
- H1: 표본의 모집단이 정규분포를 이루고 있지않다.
- 통계량: 0.933
- P-Value: 0.059
- P-Value값이 0.05보다 크므로 귀무가설 채택 (정규성이다.)

[그림 47] KoBERT Accuracy에 대한 QQ-Plot 과 정규성검정 결과



- H0: 평균 = 0.66 (기존 연구들의 평균 Accuracy)
- H1: 평균은 0.66 이 아니다.
- 통계량: -13.89
- P-Value: 2.10e-11
- P-Value값이 0.05 이하이므로 귀무가설 기각
(0.66 보다 크다고 할수 있다.)

[그림 48] KoBERT의 Accuracy에 대한 1-sample t-test 결과

딥러닝의 특성상, 학습을 통한 새로운 Feature 들을 스스로 찾기 때문에 구체적인 원인을 추출하기에는 한계가 있었으나, 하락에 영향을 주는 공통적인 요소들을 추론해 본 결과 다음과 같은 몇 가지 기사들의 공통적인 특징을 발견할 수 있었다.

모델 중, 테스트셋의 실제 값과 예측 값이 동일한 “1”(하락으로 예측)인 사례들에 대해 분석 결과, 다음과 같은 단어들이 공통적으로 나타나고 있었다.

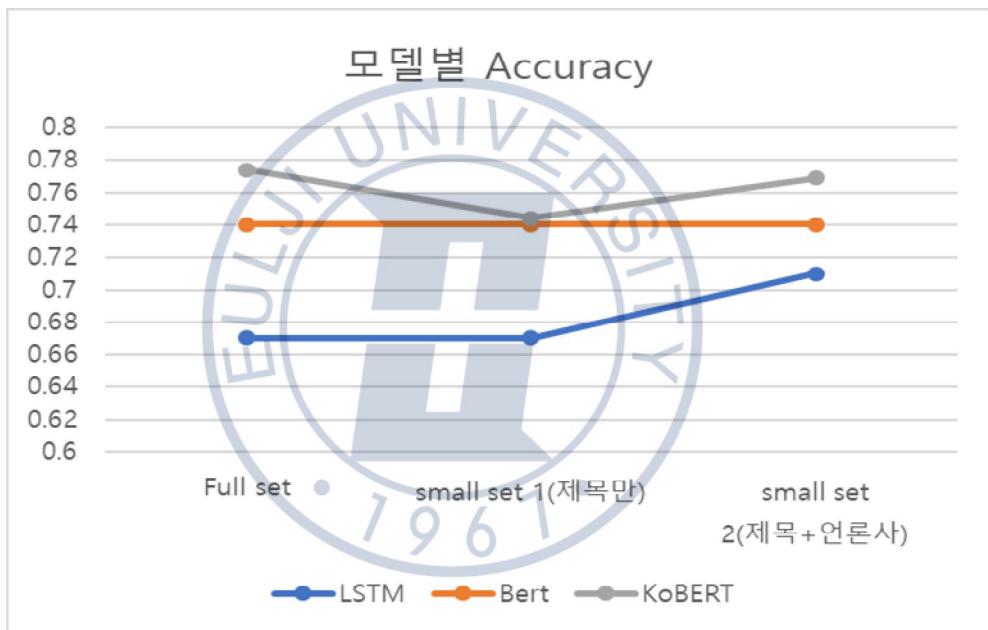
[표 17] 하락으로 예측된 경우 공통적으로 나타나는 단어들

단어	하락	상승 /보합	예시
판결	51	26	한국테크놀로지그룹 경영권 분쟁…조현범사장 9월 2심 판결이 고비. (헤럴드경제)
중지	21	11	메디톡스 "대전식약청, 이노톡스 잠정 제조중지 및 판매중지 명령". (이데일리)
처분	112	72	'과열 논란' 신풍제약, 자사주 2153억 처분. (한국경제)
배임	6	3	최신원 횡령·배임 'SKC·SK 네트웍스' 거래정지…"전화위복 기회될 (머니투데이)
허위	23	7	지트리비앤티 “허위공시 혐의 고발 사실무근” . (이데일리)
실패	23	10	오스코텍, SYK 저해제 유효성 입증 실패…목표가 ↓ -하나. (이데일리)

5.2. 가설 (나)에 대한 실험결과 분석

두번째 가설은 Word2Vec 기반 LSTM 보다는 Transformer 기반 BERT 모델이, 그리고 한국어 기반의 KoBERT 순으로 정확도가 높을 것이라는 가설이다.

아래 그림에서 보듯이 KoBERT의 성능이 모든 면에서 앞서고 있다.



[그림 49] 언어모델별 실험결과 성능 비교

특이 사항은 LSTM의 경우, Small set2(기사+언론사)에서 성능이 70%이상 나온다는 점이다. 아래 표와 같이 학습 시간까지 고려한다면, 학습시간이 절대적으로 짧은 LSTM 도 적용 검토 대상으로 봐야 할 것이다.

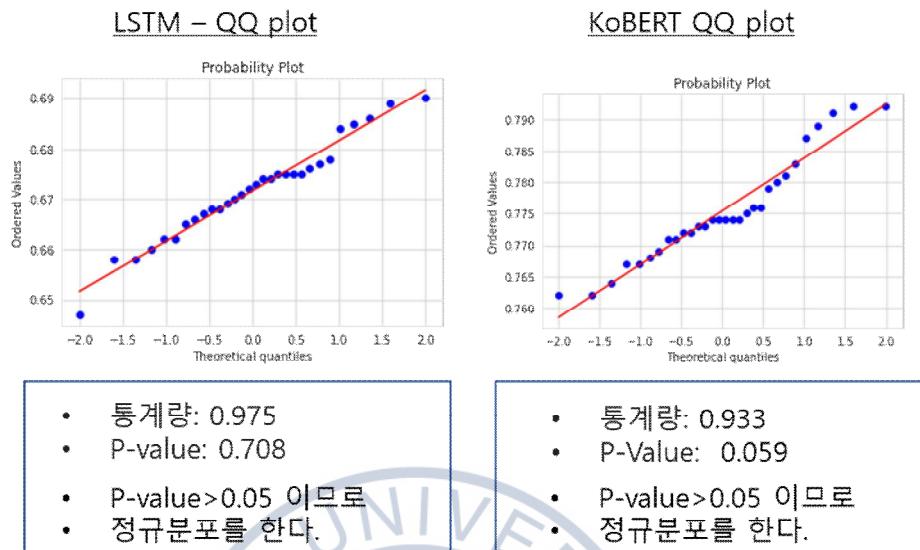
[표 18] 언어모델별 학습 시간 및 Accuracy

	학습시간(초)	Accuracy (small set2 기준)
LSTM	37	0.71
BERT	4620	0.74
KoBERT	2940	0.769

LSTM 의 경우, Full Set 이나, 기사제목만 포함된 Small Set 1 보다 언론사까지 포함되었던 Small Set2 데이터에서 가장 좋은 성능을 보였다.

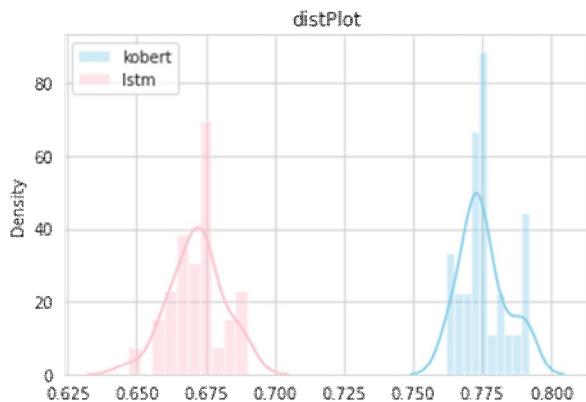
이유는 긴 문장일 경우, ‘기사제목 + 언론사 + 기사 본문’으로 구성된 데이터셋에서는 앞 부문의 정보를 기억하는데 한계가 있어, 이를 적당히 반영하지 못한 것으로 추정된다. 그만큼 긴 문장에서 중요 키워드들이 앞에 있다면 LSTM 의 경우 Attention 기반의 Transformer 모델보다는 정확도가 떨어지는 것을 보여주는 사례라 볼 수 있다.

이 중에서 기존 LSTM 모델과 KoBERT 를 동일한 테스트셋을 가지고 30 회 반복 실험(동일하게 epoch 은 10 회)하여 Accuracy 를 구하고, KoBERT 의 Accuracy 가 정말 LSTM 보다 통계적으로 유의한 차이를 가지고 높은지 확인하였다. 2-Sample t-Test 로 진행하였으며 검증하기 전에 LSTM 의 Accuracy 데이터들과, KoBERT 의 Accuracy 데이터들의 정규성이 있는지 확인하였다.



[그림 50] LSTM 및 KoBERT 정확도 데이터셋에 대한 정규성 검증

파이썬 `scipy` 모듈을 활용하여 2-Sample t-테스트를 실행한 결과, p-value 값이 2.42×10^{-46} 로, 0.05 부터 상당히 작은 수치로써 귀무가설은 LSTM의 평균과 KoBERT의 평균이 같다는 가설을 기각할 수 있었다. 결과적으로 통계량을 보더라도, KoBERT의 평균(0.775)이 LSTM의 평균(0.67) 대비 10%정도 높음을 확인할 수 있었다.



- H0: LSTM의 평균 = KoBERT의 평균
- H1: 평균이 같지 않다.
- 통계량: 44.15
- P-Value: 2.42 e-46
- P-Value값이 0.05 이하이므로 귀무가설 기각 즉, 평균이 같지 않다. 즉, KoBERT가 높다.

[그림 51] LSTM과 KoBERT모델 간 2-Sample t-Test결과

5.3. 가설 (다)에 대한 실험결과 분석

세번째 가설은 주가에 대한 예측 성능이 기사만 있을 때 보다 기자와 언론사가 추가될 경우 성능이 좋을 것이라는 가설이다. KoBERT 을 활용하여, 단지 타이틀만 활용하였을 때, 그리고 타이틀과 언론사만을 활용하였을 때, 전체기사와 기자, 언론사를 모두 Feature 로 활용하였을 때 각각의 실험 결과는 다음과 같다.

[표 19] Title만 활용 학습 결과 (Confusion Matrix)

	precision	recall	f1score	Confusion Matrix	예측	
0	0.78	0.82	0.80		0	1
1	0.68	0.63	0.65			
accuracy			0.74		0	2292
macro	0.73	0.72	0.73		1	506
weighted	0.74	0.74	0.74			1092

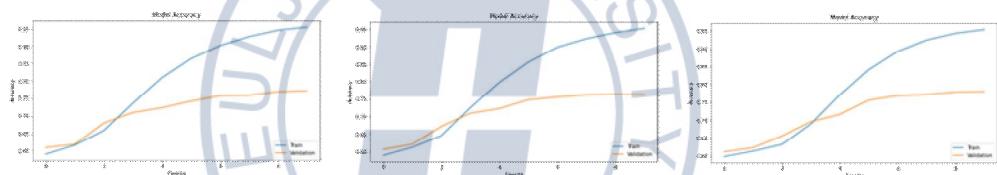
[표 20] Title+언론사 기반 학습 결과 (Confusion Matrix)

	precision	recall	f1score	Confusion Matrix	예측	
0	0.81	0.79	0.80		0	1
1	0.69	0.72	0.70			
accuracy			0.76		0	2171
macro	0.75	0.75	0.75		1	586
weighted	0.76	0.76	0.76			1290

[표 21] Title+언론사+기사 기반 학습결과(Confusion Matrix)

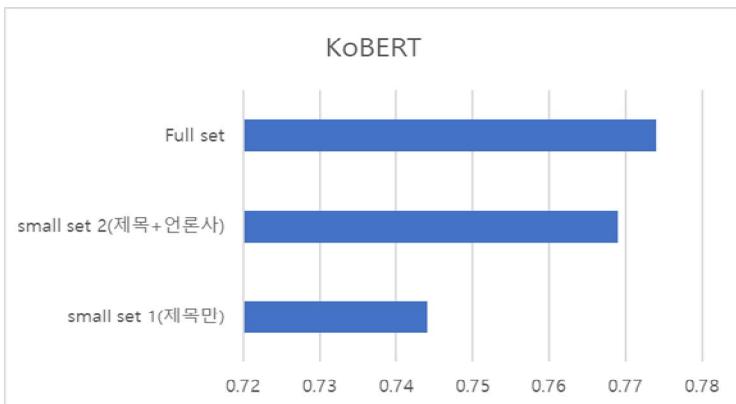
	precision	recall	f1score	Confusion Matrix		예측	
0	0.82	0.82	0.82			0	1
1	0.72	0.72	0.72	실제	0	2256	501
accuracy			0.78		1	496	1290
macro	0.77	0.77	0.77				
weighted	0.78	0.78	0.78				

3 가지 학습 모델은 10 회 Epoch 으로 Accuracy 가 더 이상 향상되지 않음을 확인할 수 있었다.



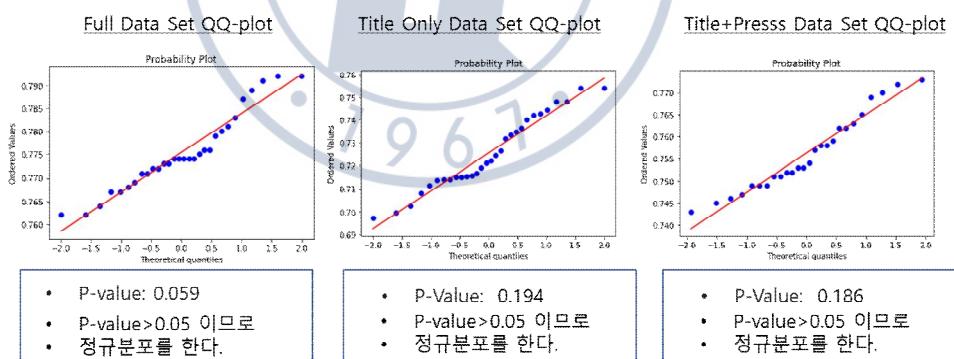
[그림 52] Epoch 수에 따른 Validation 정확도(좌부터 타이틀만, 타이틀+언론사, F타이틀+언론사+기사)

가장 평균적으로 성능이 좋은, KoBERT 의 경우, 기자와 언론사까지 포함되었을 때 가장 성능이 좋은 것을 볼 수 있다.



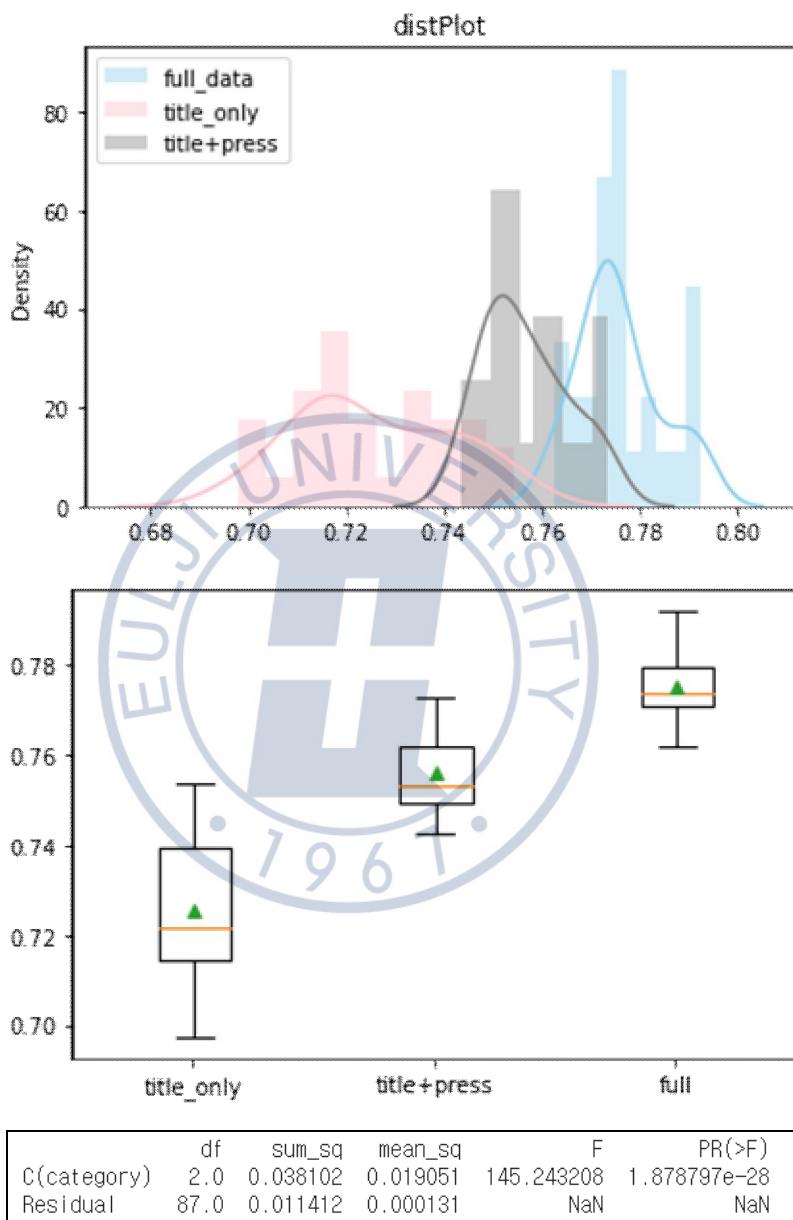
[그림 53] KoBERT의 데이터 셋의 유형에 따른 Accuracy의 변화

좀 더 유의한 차이가 있는지 확인하기 위해, 각각의 데이터셋에서 test셋을 30번 추출하여 ANOVA 분석을 통해 차이가 있는지 통계적 검증을 실시하였다. 우선 3가지 데이터 타입이 정규성이 있는지 검증한 결과 정규성이 모두 있음을 확인할 수 있었다.



[그림 54] 각 데이터셋에 대한 학습후의 정규성 검정 결과

30회씩 학습한 결과에 따라, ANOVA 분석 결과, 다음과 같이 P-Value 값이 0.05 보다 작기에 유의한 차이가 있음을 확인할 수 있었으며, 무엇보다 기사부터, 기자, 언론사까지 모두 포함시켰을 때 가장 좋은 정확도를 보임을 확인할 수 있었다.



[그림 55] 데이터셋에 따른 ANOVA 분석 결과

이를 통해 Text Data 에 추가적인 Feature 를 추가하는 방법도 제안이 가능하다. 즉, 비정형 데이터에 정형데이터를 추가하는 방법이다. 바로 메인인 되는 Text Data 에 정형데이터를 단어 그대로 병합하는 방법이다. NLP 에서는 문장을 임베딩하면서 단어들 모두를 벡터화하기 때문에, 입력 문장에 정형데이터를 있는 그대로 글자로 추가함으로써 새로운 Feature 가 추가될 수 있다.

본 연구에서는 다음과 그림과 같이 Feature(언론사)를 추가하였다.

```
preprocessed_data["기사요약"] = company_stock_articles_shuffled["기사제목"] + " " + "(" +  
company_stock_articles_shuffled["언론사"] + ")" " + preprocessed_data["기사요약"] |
```

세이브존 I&C, 자회사 투어캐빈 흡수합병 결정. (이데일리) [이데일리
송승현 기자] 세이브존 I&C;(067830)는 자회사 투어캐빈을
흡수합병하기로 결정했다고 23 일 공시했다. 존속회사는
세이브존 I&C;이고, 소멸회사는 투어캐빈이다.

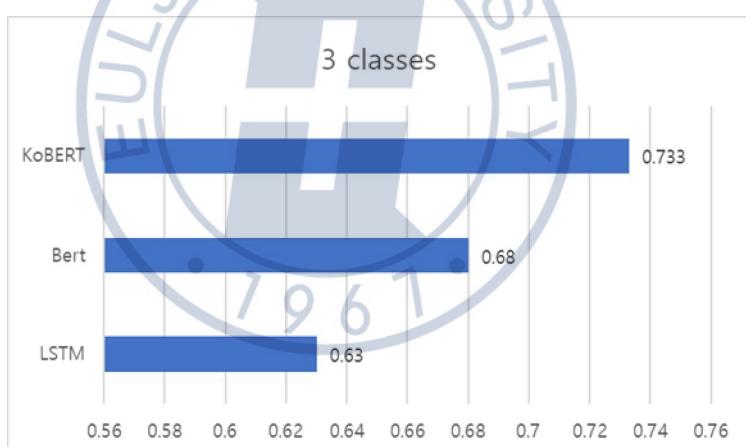
[그림 56] Feature 추가 방법 및 예시

코드에서 보듯이, 기사제목과 언론사와 기자명이 포함된 기사요약문을 모두 합쳐, 기사요약에 추가하였다. 무엇보다도 <기사제목>만으로 학습했을 때, <제목 + 언론사>로 학습했을 때 평균 차이가 급격히 나는 것을 봐도 이 방법이 유효하다는 것이 검증되었다. 만약, 언론사뿐 아니라 Feature 가 Categorical Data 인 경우, 텍스트 형태로 결합하면 어떤 정형 데이터이고 추가가 가능하다. 예를 들어, 주가의 경우, 언론사명, 기자성명, 코스닥/코스피 여부, 종목명, “좋아요/싫어요 개수”, 뉴스 댓글들을 추가하기 위해, 기사 이후에 텍스트형태로 결합하는 것이다.

5.4. 가설 (라)에 대한 실험결과 분석

네번째 가설은 단기적 복구 패턴을 통해 의도적으로 쓴 기사와 악재성 기사는 예측가능한지 확인하는 것이다. 이는 지속 하락할지, 하락 후 반등할지에 대한 케이스들을 예측할 수 있는지로 검증하였다.

지속 하락시 “1”, 하락 후 3 일안에 원복” 2”. 그렇지 않은 경우 ”0”의 클래스를 할당하여 학습하였다. 모든 모델이 60%이상의 Accuracy 를 보였고, 특히 KoBERT 는 73%의 상대적으로 높은 Accuracy 가 도출되었다.



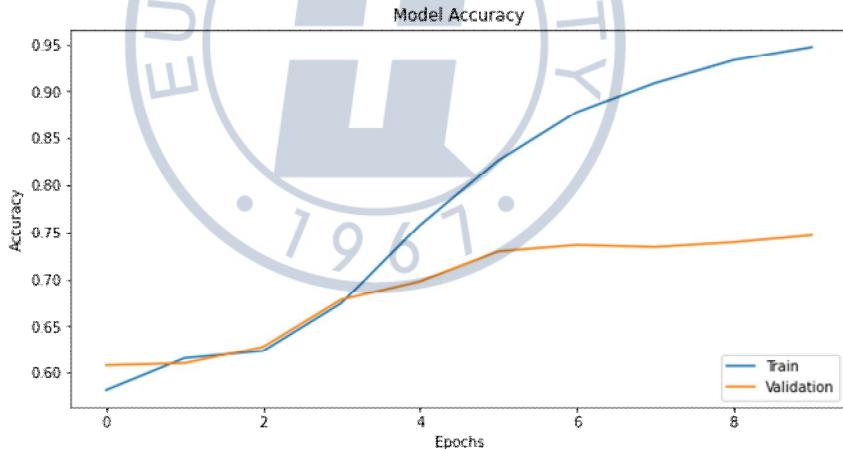
[그림 57] 하락후 원복, 지속하락에 대한 Test Accuracy

KoBERT 모델에 대한 세부적인 학습 결과는 다음과 같다.

[표 22] 3–Classes 학습결과 – Confusion Matrix

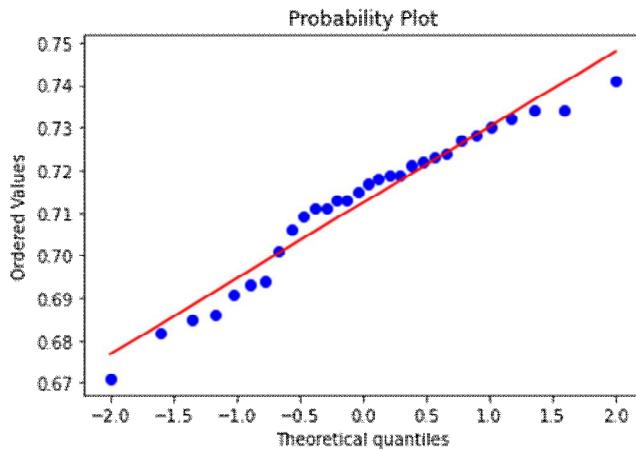
	precision	recall	f1score	Confusion Matrix	예측			
					0	1	2	
0	0.76	0.88	0.82					
1	0.71	0.53	0.61					
2	0.63	0.51	0.56					
accuracy			0.73					
macro	0.70	0.64	0.66					
weighted	0.73	0.74	0.73					
				실제	0	2382	145	188
				1	336	481	90	
				2	399	54	468	

이 모델 역시, Epoch 10 이후 validation Accuracy 가 더 이상 향상되지 않음을 확인할 수 있었다.



[그림 58] KoBERT – 3 Classes 모델 Epoch 별 Accuracy

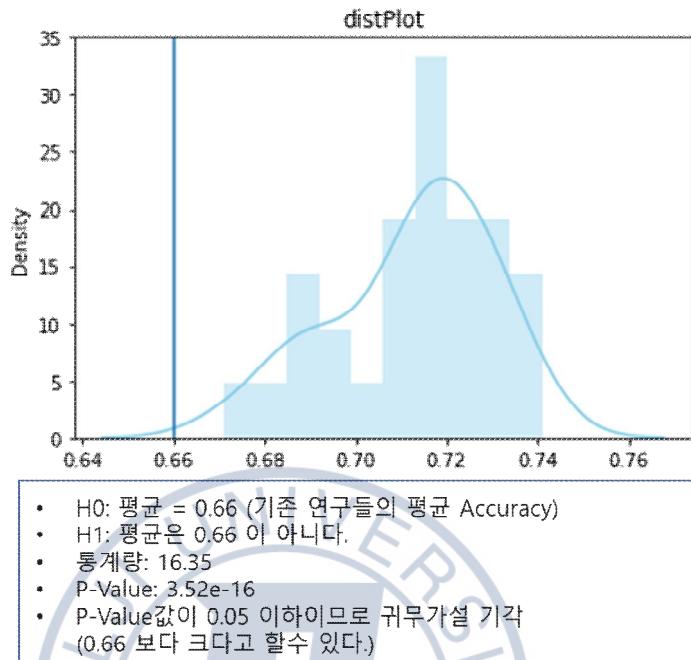
통계적인 유의차 검정을 위해, KoBERT 만 활용하여 다른 테스트셋을 가지고 30 회 반복하여 Accuracy 를 확인한 결과는 다음과 같다.



- H0: 표본의 모집단이 정규분포를 이루고 있다.
- H1: 표본의 모집단이 정규분포를 이루고 있지 않다.
- 통계량: 0.951
- P-Value: 0.189
- P-Value값이 0.05보다 크므로 귀무가설 채택 (정규성이다.)

[그림 59] 3 Class KoBERT Accuracy 분포-정규성검정

정규성 검정 결과 정규성을 띠고 있었으며, 이를 기반으로 기존 연구들의 Accuracy 평균인 0.66 과 비교하여 1-Sample t-Test 를 실시한 결과, p-value 값이 0.05 보다 작음으로써 유의한 차이가 있어, 0.66 보다 크다는 것을 확인할 수 있었다.



[그림 60] KoBERT 3가지 클래스에 대한 유의성 검정 결과

예측된 값과 실제 값이 동일한 테스트셋을 통해 몇 가지 분석해 본 결과 다음과 같은 패턴을 예상할 수 있었다.

[표 23] 지속하락/하락 후 원복 연관 단어와 기사 예시

단어	지속하락	하락 후 원복	예시
철회	1	9	[특징주] 더블유게임즈, 美 상장 철회 소식에 12%대 하락. (서울경제)
부실	8	2	[특징주] 헬릭스미스, 부실 펀드 손실에 '급락'. (이데일리)
불확실	17	9	엘앤씨바이오, 중국 진출 불확실성 겉혀야-한국. (이데일리)
과정금	9	16	공정위, 한온시스템 '단가 후려치기' 역대급 과정금 '115 억' 부과. (이데일리)
판결	48	3	메디톡스, ITC 항소..."판결 전문에 대웅 균주 도용 명시". (이데일리)
중지	3	18	한올바이오파마 자가면역질환 신약 후보 HL161 의 美임상 2상 중지. (매일경제)
↓	208	163	SM, 자회사 부진에 영업적자 지속…목표가 ↓ -이베스트. (이데일리)

그러나, 위의 사례에서 보듯이, 악의적으로 쓴 글인지에 대해서는 구분하기 어려웠다. 결론적으로, 가설 4의 경우, 악의성인지, 악재성인지 구분하기는 불가능하지만, 특징적인 문맥을 통해, 지속적으로 하락할지, 단기적 하락 후 원복 할지에 대해서는 어느정도 예측이 가능한 것으로 판단되었다.

5.5. 성능향상을 위한 파라미터 튜닝에 대한 고찰

KoBERT 에 Full Data set 를 활용했을 때 평균 77.5%의 정확도를 보였다. 이때 조건은 <Batch Size: 64>, <Epochs: 10 회>, <Regular Expression: None>으로 학습한 결과였다.

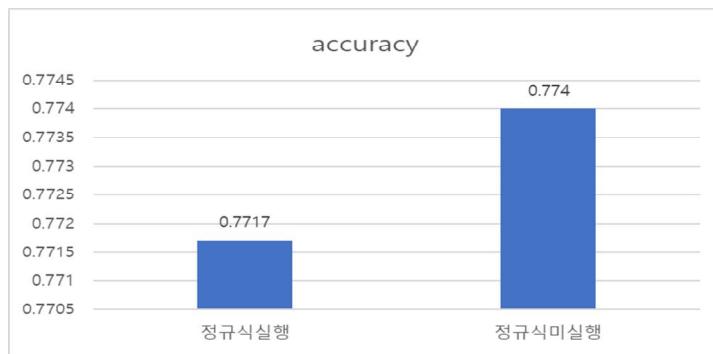
성능을 높이기 위해, 다음과 같은 2 가지 실험을 추가로 진행하였다.

[표 24] 성능 향상을 위한 추가 실험 계획

안	성능향상을 위한 추가 실험
1 번 (정규식 적용)	<p><정규식을 적용한 Text 적용></p> <ul style="list-style-type: none">- 정규식을 적용할 경우, 전각문자, 기호, 무의미한 숫자, 잘못된 띄어쓰기 등의 노이즈를 제거하기 때문에 학습 효과가 좋아질 것으로 예상됨.- 본 연구에서는 최소한의 RE 로써 기호와 전각문자만 제거하는 것으로 다음의 코드 추가 후 학습 및 테스팅 실시 <pre>train['text'] = train['text'].str.replace("[^ㄱ-ㅎㅏ-ㅣ가-힣 A-Za-z0-9]","")</pre>
2 번 (배치 및 에폭수 증가)	<ul style="list-style-type: none">- Epochs 수와 Batch Size 를 늘려 학습량 증가. Backward Propagation 을 통한 파라미터 업데이트로 Loss 최소화에 기여할 것으로 예상- Batch Size 를 64 에서 128, Epoch 을 10 에서 50 회로 늘려 학습량 증가 실험 실시

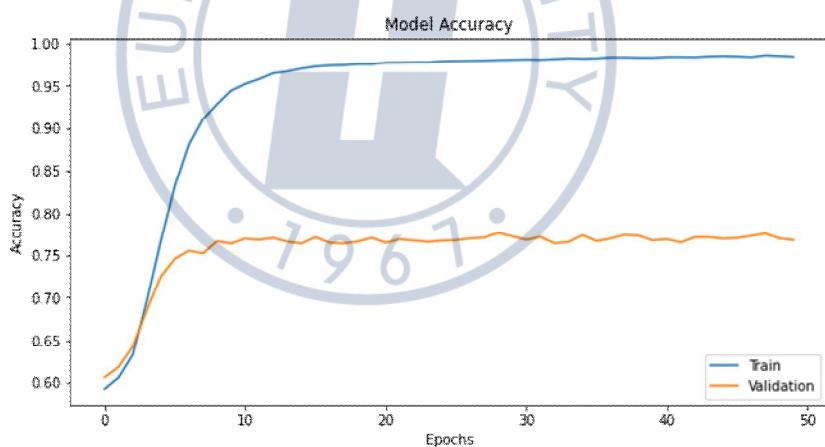
실험 결과, 다음과 같은 결과가 도출되었다. 1 번, 2 번 모두 큰 성능 향상이 있음을 확인하지 못하였다.

우선 1 번 추가 실험의 경우에는 Regular Expression 을 한 경우와 하지 않은 경우의 성능 차이가 거의 없음을 확인할 수 있었다. 오히려 정규식을 실행했을 경우가 다소 낮은 Accuracy 를 보이고 있다.

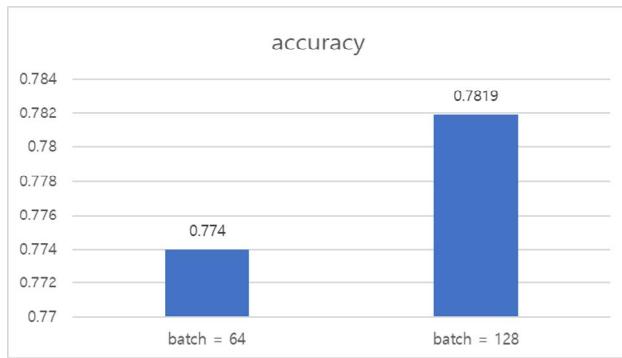


[그림 61] 정규식 실행시에 Accuracy의 변화

2 번째 실험에서는 epoch 과 batch 사이즈를 증가시켜 성능향상을 확인해 본 결과, 1%정도의 향상만 있을 뿐 더 이상 학습을 통한 성능 향상을 볼 수 없었다.



[그림 62] Epoch에 따른 Accuracy의 변화



[그림 63] Batch Size 변화에 따른 Accuracy의 변화

지금까지의 실험 결과를 요약하면 다음과 같다.

[표 25] 모델에 대한 성능 관련 가설 검증 결과 요약

연구 가설	판단 기준	검증결과
가설(가) 소수의 종목 뉴스기사만으로 사전처리 없이 하락을 예측할 수 있다.	-통계적 가설(1-sample t-Test 실행) H0: KoBERT 평균 Accuracy = 0.66 H1: KoBERT 평균 Accuracy > 0.66 (Data Set: Binary Classification Set)	p-value = 8.42e-35 “신규모델의 정확도가 이전 연구모델보다 높다고 할수 있다.”
가설(나) KoBERT의 성능이 기존 방식인 LSTM 보다 높다.	-통계적 가설(2-sample t-test 실행) H0: KoBERT 평균 = LSTM 평균 H0: KoBERT 평균 > LSTM 평균	p-value = 22.4e-46 “BERT 모델의 성능이 기존모델보다 성능이 높다고 할수 있다”
가설 (다) 추가Feature 예측은 기자와 언론사등의 Feature 가 추가될수록 정확하다.	-통계적 가설 (ANOVA 분석 실시) H0: Data 1 평균=Data 2 평균=Data 3 평균 H1: 하나라도 평균이 같지 않다. - Data 1: 기사+언론사+기자 Data 2: 기사제목+언론사 Data 3: 기사제목	p-value = 1.8e-28 “추가 Feature 를 기사와 함께 병합하여 임베딩하여 추가할수록 성능이 높다.”
가설 (라) 단기적 복구 패턴을 통해 악의적으로 쓴 기사와 악재성 기사는 예측가능 할 것이다.	-통계적 가설(1-sample t-Test 실행) - KoBERT 의 평균 정확도> 66% H0: KoBERT 평균 Accuracy = 0.66 H1: KoBERT 평균 Accuracy > 0.66 (Data Set: 3 Classification Set)	p-value = 3.52e-16 “주가 방향에 대해 상승/하락뿐 아니라, 하락후 반등까지도 예측이 가능하다.”

6. 결 론

본 연구에서는 Transformer라는 최신의 딥러닝 알고리즘을 적용한 Pre-Trained Model인 KoBERT를 활용해 번거로운 Feature Extraction 없이 뉴스 텍스트 기반의 비정형데이터를 그대로 활용해 주가 추이를 예측하는 모델을 개발하였다. 성능 비교 결과, 기존 뉴스 감성지수 기반의 주가 추이 예측 모델 대비, 평균 10% 이상(평균 정확도 77%), p-value 0.05 이하로 통계적으로 유의한 개선 효과가 검증되었다. 즉, 번거롭고, 감성사전의 품질에 의존하는 감성지수로의 정형화라는 Feature Engineering 없이도 비정형데이터를 활용한 모델이 유의함을 확인할 수 있었다. 또한 카테고리 형태의 Feature들을 텍스트에 함께 기재하는 방법을 제안하였다. 비정형데이터와 정형데이터를 함께 학습을 시키려면 비정형데이터를 정형데이터화 해야 한다. 그러나 반대로 정형데이터를 비정형데이터에 그대로 포함시키는 방법을 모델링하여 실험하였다. 본 연구 결과를 토대로 다음과 같은 4 가지 결론을 도출하였다.

- 뉴스기사 기반 주가 예측시, 감성수치화라는 Feature Engineering 없이 바로, Text 입력으로 예측 가능하다.

- Transformer 기반의 BERT 언어모델, 특히 한국어 기반의 KoBERT 언어모델의 성능이 기존 LSTM 보다 성능이 우수하다.
- Text 기반 분류 Task 실행 시, 추가 Feature, 특히 카테고리컬 데이터의 입력 방법은 Text에 있는 그대로 병합하여 포함시키면 반영된다.
- 악의적 기사와 악재기사를 구분하기는 어려우나, 기사에 따라 지속 하락, 또는 하락 후 원복 할지 판단할 수 있는 모델링이 가능하다.

이러한 결과를 가지고 활용 가능한 영역으로는 주식 뿐만 아니라, 최근 암호화 화폐의 급등락 예측을 위해 기사를 기반으로 모델링이 가능하리라 판단된다. 그리고 이외에도 주가 예측뿐 아니라 텍스트를 기반으로 다양한 분류 문제에도 충분히 활용 가능하다. 예를 들어 웹소설 1화만 가지고도, 과거 천만구독자 여부로 학습한다면, 대박 여부를 사전에 예측할 수 있다. 그러면 이를 상위 노출시킴으로써 더욱 구독자들을 확보하고, IP 화하는데 도움을 줄 수 있을 것이다. 그리고 드라마 시나리오와, 감독, 작가 등을 하나의 텍스트에 포함시킨 후 10% 시청율 여부로 학습하면, 새로운 드라마의 흥행 여부도 예측이 가능할 것이다. 이밖에도 의사와의 상담내용을 가지고 특정질병의 유무를 판단하든지, 누군가에게 전화가 왔을 때, 피싱여부를 학습한다면, 피싱을 사전에 예방도 가능할 것이다.

모델 성능을 향상시키는데는 몇 가지 한계가 있을 것으로 판단된다. 가장 큰 원인 중 하나는 주가가 하락하는 이유가 기사 하나만이 원인이 아니기 때문이다. 예를 들어 전체 주가지수가 하락하든지, 아니면 해당 섹터의 주가들이 특정 악재에 의해 전체가 하락하는 하는 경우도 있기 때문이다. 이를 해결할 수 있는 방법 중 하나는 종합주가 지수와 비교하여, 종합주가지수 하락 시 이를 보정하면 좀더 하락에 대한 인덱스 정확도를 높힐 수 있을 것으로 판단된다.

또 다른 한계는 전체 종목을 가지고 학습할 것이 아니라, 특정 종목군 별로 분석하면 더욱 정확도가 올라가는 종목군이 있을 것으로 예상된다. 특히, 바이오 분야의 경우, 각 종목 간의 동조현상이 있기 때문에 다른 종목의 뉴스라도 그 영향이 전체 관련 종목으로 확대될 수 있기 때문이다. 뿐만 아니라, 기사의 전반적인 양에도 한계가 있었다. 이번 연구에 활용한 기사데이터는 포털 증권의 종목뉴스만 가지고 왔지만, 활용에는 한계가 있었다. 실제 대중들이 접하는 것은 <뉴스 세션>에 상위로 노출되는 종목의 뉴스와 검색 시 상위 노출되는 종목 뉴스이기 때문이다. 그리고 검색 시 하나의 포털뿐만 아니라, 다양한 포털까지도 확대가 필요할 것이며, 더욱이 각종 종목의 카페 글도 함께 분석이 되어야 좀더 정확한 모델링이 가능하리라 판단된다.

네번째 결론인 <의도적 기사>와 <악재 기사>의 판정을 주가가 단기간 원복한다면 의도적 기사일 것이라는 주가의 패턴을 가지고 대용특성으로

판단해보긴 했지만, 기사 내용들을 세부적으로 들여다 보면 의도적 기사인지 아닌지 알 수가 없었다. 악의적 가짜 뉴스를 밝히기 위해서는 수많은 뉴스 중 하나 하나를 보면서 가짜임을 레이블링 하여 학습한다면 가짜 뉴스를 밝히고, 이를 기반으로 주가와 함께 재 학습한다면 가능성은 있으리라 기대된다.

본 연구의 개선모델은 NLP 비정형+정형 데이터를 그대로 활용해 학습함으로써 다양한 NLP 기반 예측에 활용할 수 있다는 데 기여할 것으로 기대된다. 코인추이를 예측하는 방법으로 예를 들어 <트윗내용 + 트윗인물 + 트윗시간대>을 결합하여 활용할 수 있고, 문학분야에서도 인문공학으로 발전할 수 있는 토대가 될 수 있다. 즉 <웹소설 1 회 전문 + 소설 작가 + 장르 + 문장 길이>을 통한 웹소설의 흥행 여부를 예측할 수 있는 모델이 될 수 있다. 의학분야에서도 질병의 예측모델로 활용할 수 있다. 예를 들어 <문진내용 + 의사명 + 의사 전공 + 병원명> 등을 결합한 모델로 활용할 수 있다. 또한 <드라마 시나리오 + 제작사명 + 감독 + 주연 + 방송사>를 통한 시청률 추이 예측등 다양한 곳에 활용이 가능할 것으로 기대된다.

참고 문헌

- [1] 옥기율, & 이민규. (2019). 코스닥 시장에서의 주가 예측 모형에 관한 연구. “한국 사회과학연구”, 38(3), pp. 141–162.
- [2] 우민철, & 김명애. (2019). 공매도 전략의 투자성과 분석. “한국 증권학회지”, 48(3), pp. 371–391.
- [3] 전새미, 정여진, & 이동엽. (2016). 개별 기업에 대한 인터넷 검색량과 주가 변동성의 관계. “지능정보연구”, 22(2), pp. 81–96.
- [4] Koo, P., & M. Kim. (2015). A study on the Relationship between Internet Search Trends and Company's Stock Price and Trading Volume. “The Journal of Society for e-Business Studies”, Vol 20, No.2, pp. 1~14.
- [5] 김동영, 박제원, & 최재현. (2014). SNS와 뉴스기사의 감성분석과 기계학습을 이용한 주가예측 모형 비교 연구. “한국 IT서비스 학회지”, pp. 221–233.
- [6] 김유신, 김남규, & 정승렬. (2012). 뉴스와 주가 빅데이터 감성분석을 통한 지능형 투자의사결정모형. “지능정보연구”, 18(2), pp. 143–156.
- [7] 김재봉, & 김형중. (2017). 주가지수 방향성 예측을 위한 도메인 맞춤형 감성사전 구축방안. “한국디지털콘텐츠학회”, pp. 585–592.
- [8] B Liu. (2012). Sentiment Analysis and Opinion Mining. “Morgan & Claypool.”
- [9] Y.Kim, N.Kim., & S.R. Jeong. (2014). Stock index invest model using news big data opinion mining. “KIIS Journal of Intelligence and Information Systems”, vol.18, pp. 143–156.
- [10] 김명진, 류지혜, 차동호, & 심민규. (2020). SNS감성 분석을 이용한 주가 방향성 예측:네이버 주식 토론방 데이터를 이용하여. “한국전자거래학회지”, 25(4), pp. 61–75.
- [11] 곽란, 이성우, 서봉원, & 김성현. (2018). 지능적 이슈트래킹 시스템: 주가와 종목별 키워드 관계탐색. “한국 HCI학회 학술대회”, pp. 351–356.
- [12] TimesNew YorkThe. (2016). “The Robots Are Coming for Wall Street ” . The New York Times Magazine: <https://www.nytimes.com/2016/02/28/magazine/the-robots-are-coming-for-wall-street.html>에서 검색됨

- [13] 성노윤, & 남기환. (2019). 시스템적인 군집 확인 과 뉴스를 이용한 주가 예측. “지능정보연구”, 25(3), pp. 1–17.
- [14] 장은하, 최회련, & 이홍철. (2020). BERT를 활용한 뉴스 감성분석과 거시경제지표 조합을 이용한 주가지수 예측. “한국컴퓨터정보학회논문지”, 25(5), pp. 47–56.
- [15] MikolovTomas. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781.
- [16] MannesJohn. (2018년 January월 12일). “Facebook's fastText library is now optimized for mobile ” . TechCrunch: <https://techcrunch.com/2017/05/02/facebook-fasttext-library-is-now-optimized-for-mobile/>에서 검색됨
- [17] Devlin, J., Chang, M.-W., Lee, K., & Toutanova. (2018). K. BERT: Pre-training of deep bidirectional transformers for language understanding.. arXiv preprint arXiv:1810.04805.
- [18] Polosukhin, Illia, Kaiser, Lukasz, & Gomez, Aidan N. (2017). Attention Is All You Need. 1706.03762. arXiv.
- [19] 성노윤, & 남기환. (2018). 산업군내 동질성을 고려한 온라인 뉴스 기반 주가예측. 24(2). 지능정보연구.
- [20] 정지선, 김동성, & 김종우. (2015). 온라인상의 뉴스 감성 분석을 활용한 개별 주가 예측에 관한 연구. “한국지능정보시스템학회 학술대회논문집”, pp. 45–58.
- [21] 조성민, & 김우생. (2020). RNN 과 강화학습을 이용한 자동문서 제목 생성. “Journal of information Technology Applications & Management,” , 27(1), pp. 49–58.
- [22] 지규빈, 나요셉, 곽경민, & 최태영. (2020). LSTM 기반 Sequence-toSequence을 활용한 한국어제목 생성. “한국정보기술학회 종합학술대회” , pp. 308–311.
- [23] 한윤지, & 김남규. (2018). 뉴스와 소셜데이터를 활용한 텍스트 기반 가짜 뉴스 탐지 방법론. “한국 전자 거래학회지” , 23(4), pp. 19–39.
- [24] 홍서영, 심미단, & 이대호. (2020). Transformer 기반 한국어 채팅 체 문장 교정 모델. “한국정보과학회 학술발표 논문집” , pp. 1409–1411.
- [25] 홍태석, 강상우, & 서정연. (2018). 심층 인공신경망을 활용한 가짜 뉴스 판별. “한국정보과학회 학술발표논문집” , pp. 617–619.
- [26] 김민수, 혀몽하, & 권혁준. (2020). 한국 포털 사이트 검색강도가 주가 동조성 및 위험에 미치는 영향. “한국전자거래학회지” , 25(4), pp. 125–141.
- [27] 김연군, & 이정우. (2020). Transformer 기반 한국어 음성인식 모

- 델. “한국통신학회 학술대회논문집”, pp. 647–648.
- [28] 김유미. (2021). 소셜미디어의 가짜뉴스(Fake News)에 대한 제3자 효과:감염병 관련 허위 정보를 중심으로. “한국방송학보”, 35(1), pp. 5–32.
- [29] 박윤보, 조국한, & 송영준. (2020). Transformer 모델을 이용한 채팅사용자 식별. “대한전자공학회 학술대회”, pp. 2056–2058.
- [30] 서현태, 한요섭, 정혜동, & 고상기. (2019). Sequence-toSequence모델 기반 문맥 자유 언어 이해 및 자동 오류 수정 학습. “한국정보과학회 학술발표논문집”, pp. 907–909.
- [31] 손병찬, 이준엽, 천성준, 최병진, & 김남수. (2020). 지도Attention을 이용한 한국어 Transformer 음성 합성에 관한 연구. “한국통신학회 학술대회논문집”, pp. 638–639.
- [32] 안수현, & 조정현. (2018). 텍스트마이닝을 활용한 웹사이트FAQ개선방안. “한국콘텐츠학회 종합학술대회 논문집”, pp. 361–362).
- [33] 안혁주, 최맹식, & 김학수. (2015). 음성기반 FAQ 검색의 성능향상. “한국정보과학회 학술발표논문집”, pp. 678–680.
- [34] 임정수. (2020). 주가 경향 예측 모델의 공정한 성능 평가 방법. “한국콘텐츠학회”, 20(10), pp. 702–714.
- [35] Ahn, S., (2012). “An Empirical Study on the Volatility Decomposition In Korea Stock Market – The analysis of the Return Effect and the Volatility Decomposition in the Industry and the Period”, Dongguk University,
- [36] Aghabozorgi, S., and Y. W. Teh.,(2014). "Stock Market Co-Movement Assessment Using a Three-Phase Clustering Method," Expert Systems with Applications, Vol.11, No.4, 1301~1314.
- [37] Aiolfi, F., and M. Donini,. (2015). "Easymkl: A Scalable Multiple Kernel Learning Algorithm," Neurocomputing, Vol.169, No.1, 215~224.
- [38] Barber, B. M., and T. Odean, (2008). “All that Glitters: The Effect of Attention and News on the Buying Behavior of Individual and Institutional Investors,” Review of Financial Studies, Vol.21, No.2, 787~818.
- [39] Berkman, H., P. D. Koch, L. Tuttle, and Y. J. Zhang., (2012). “Paying Attention: Overnight Returns and the Hidden Cost of Buying at the Open,” Journal of Financial and Quantitative Analysis Vol.47, No.4, 715~741.
- [40] Bordino, I., S. Battiston, G. Caldarelli, M. Cristelli, A. Ukkonen, and L. Weber, (2012). “Web search queries can predict stock

- market volumes," PLOS One, Vol.7, No.7, 1~17.
- [41] Bun, J., R. Allez, J.-P. Bouchaud, and M. Potters, . (2016). "Rotational Invariant Estimator for General Noisy Matrices," IEEE Transactions on Information Theory, Vol.62, No.12,7475~7490.
- [42] Bun, J., J.-P. Bouchaud, and M. Potters,. (2017). "Cleaning Large Correlation Matrices: Tools from Random Matrix Theory," Physics Reports, Vol.666, No.1, 1~109.
- [43] Chang, Y. B., Y. Kwon., and W. Cho, (2015) . "Attention to the Internet : The Impact of Active Information Search on Investment Decisions" , Journal of Intelligence and Information Systems, Vol.21, No.3, 117~129.
- [44] Chemmanur, T. and A. Yan,(2009). "Advertizing, Attention, and Stock Returns," Technical Report, Boston College and Fordham University.
- [45] Cho, C. H., and T. Mooney,. (2015). "Stock Return Comovement and Korean Business Groups," Review of Development Finance, Vol.5, No.2, 71~81.
- [46] Cooper, C., K. Mallon, S. Leadbetter, L. Pollack, and L. Peipins, (2005) . "Cancer Internet Search Activity on a Major Search Engine, United States 2001~2003," Journal of Medical Internet Research, Vol.7, No.3, e36.
- [47] Da, Z., J. Engelberg, and P. Gao, (2011). "In Search of Attention," Journal of Finance, Vol.66, No.5, 1461~1499.
- [48] Engelberg, J. E., and C. A. Parsons. (2011). "The causal impact of media in financial markets." The Journal of Finance, Vol.66, No.1, 67~97.
- [49] Ettredge, M., J. Gerdes, and G. Karuga, (2005). "Using Web-based search data to predict macroeconomic statistics," Communications of the ACM, Vol.48, No.11, 87~92.
- [50] Jeong, J.S., Kim, D.S., & Kim, J.W.,. (2015). "Influence analysis of Internet buzz to corporate performance Individual stock price prediction using sentiment analysis of online news. "Korea intellingent information Systems Society" , Vol.21, No.4, pp. 37~51.
- [51] Lee, M.S., & Ahn,H.C., (2018). A Time Series Graph based Convolutional Neural Network Model for Effective Input Variable Pattern Learning: Application to the Prediction of Stock Market. " Korean Intelligent information systems

- society" , Vol.24, No.1, pp. 167–181.
- [52] Choi,H. (2014). Investor attention and stock return reversals : evidence from the KOSDAQ market. Yonsei University.
- [53] García, A., (2016). "Global Financial Indices and Twitter Sentiment: A Random Matrix Theory Approach," Physica A: Statistical Mechanics and its Applications, Vol.461, No.1, 509~522.
- [54] Gervais, S., R. Kaniel, and D. H. Mingelgrin, (2001). "The high-volume return premium," Journal of Finance, Vol.56, No.3, 877~919.
- [55] Ginsberg, J., M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant, (2009). "Detecting influenza epidemics using search engine query data," Nature, Vol.457, No.7232, 1012~1014.
- [56] GillNav. (2017).
<https://www.datasciencecentral.com/profiles/blogs/overview-of-artificial-intelligence-and-role-of-natural-language>에서
검색됨
- [57] Groth, S. S., and J. Muntermann, (2011). "An Intraday Market Risk Management Approach Based on Textual Analysis," Decision Support Systems, Vol.50, No.4, 680~691.
- [58] Gu, Y., C. Wang, D. You, Y. Zhang, S. Wang, and Y. Zhang, (2012). "Representative Multiple Kernel Learning for Classification in Hyperspectral Imagery," IEEE Transactions on Geoscience and Remote Sensing, Vol.50, No.7, 2852~2865.
- [59] Hagenau, M., M. Liebmann, and D. Neumann, (2013). "Automated News Reading: Stock Price Prediction Based on Financial News Using Context-Capturing Features," Decision Support Systems, Vol.55, No.3, 685~697.
- [60] Hong,S.H. (2020). A study on stock price prediction system based on text mining method using LSTM and stock market news. "Journal of Digital Convergence" , Vol.18, No. 7, pp. 223–228.
- [61] Hsu, C., C. Chang, and C. Lin, (2010). "A Practical Guide to Support Vector Classification," Department of Computer Science National Taiwan University.
- [62] Jeong, J. S., D. S. Kim, and J. W. Kim, (2015). "Influence analysis of Internet buzz to corporate performance : Individual stock price prediction using sentiment analysis of online

- news" , Journal of Intelligence and Information Systems, Vol.21, No.4,37~51.
- [63] Kang, I., (2013). "study on the relationship between volatility of industrial stock market and volatility of exchange rate," Korean Journal of Business Administration, Vo.25, No.3, 1703~1724.
- [64] Keerthi, S. S., and C.-J. Lin, (2003). "Asymptotic Behaviors of Support Vector Machines with Gaussian Kernel," Neural computation, Vol.15, No.7, 1667~1689.
- [65] Kang,Y.J. , & Jang, W.W. (2016). The Five Factor Asset Pricing Model: Applications to the Korean Stock Market. "Eurasian Studies" , Vol.12, No.2, pp. 155–180.
- [66] Kim, B. C., (2015) . "Using Internet Search Trends Analysis to Monitor and Predict Suicide Risk in Korea" , Korean Journal of Communication Studies, Vol.23, No.2, 99~120.
- [67] Kim, D.-H., and H. Jeong, (2005). "Systematic Analysis of Group Identification in Stock Markets," Physical Review E, Vol.72, No.4, 046133.
- [68] Kim, D. and J. S. Yu, (2014). "A Dynamic Relationship Between Internet Search Activity, Housing Price, and Trading Volume" , Korean Appraisal Review, Vol.24, No.2, 125~140.
- [69] Kim, D.Y., & Lee,Y.I. (2018). News based Stock Market Sentiment Lexicon Acquisition Using Word2Vec. "The Korea Journal of Bigdata" , Vol. 3, No.1, pp. 13–20.
- [70] Kim, D.Y., Park,J.W., & Choi, J.H. (2014). "A Comparative Study between Stock Price Prediction Models Using Sentiment Analysis and Machine Learning Based on SNS and Mews Articles. " Journal of Information Technology Services" , Vol.12, No.3, pp. 221–233.
- [71] Kim, Y., N. Kim, and S. R. Jeong, (2012). "Stock–Index Invest Model Using News Big Data Opinion Mining" , Journal of Intelligence and Information Systems, Vol.18, No.2, 143~156.
- [72] Kim,Y. S., Kim,N.G., & Jeong, S. R. (2012). Stock Index Invest Model using News Big data Opinion Mining. " Journal of Intelligence and Information Systems" , Vol. 18, No.2, pp. 143–156.
- [73] Laloux, L., P. Cizeau, M. Potters, and J.-P. Bouchaud, (2000). "Random Matrix Theory and Financial Correlations," International Journal of Theoretical and Applied

- Finance, Vol.3, No.3, 391~397.
- [74] Lee.H.J. (2019). Analysis of News Big Data for Deriving Social Issues in Korea. "The Journal of Society for e-Business Studies" , Vol.24, No32, pp. 163~182.
- [75] Loh, L., (2013). "Co-Movement of Asia-Pacific with European and Us Stock Market Returns: A Cross-Time-Frequency Analysis," Research in International Business and Finance, Vol.29, No.1, 1~13.
- [76] "medium.com" . <https://medium.com/@antoine.louis/a-brief-history-of-natural-language-processing-part-2-f5e575e8e37>에서 검색됨
- [77] Morck, R., B. Yeung, and W. Yu, (2000)."The Information Content of Stock Markets: Why Do Emerging Markets Have Synchronous Stock Price Movements?," Journal of financial economics, Vol.58, No.1~2, 215~260.
- [78] Nam, K., and N. Seong, (2019). "Financial News-Based Stock Movement Prediction Using Causality Analysis of Influence in the Korean Stock Market," Decision Support Systems, Vol.117, No.1, 100~112.
- [79] Park, E. L., and S. Cho, (2014). "Konlpy: Korean Natural Language Processing in Python," Proceedings of the 26th Annual Conference on Human & Cognitive Language Technology
- [80] Preis, T., H. S. Moat, and H. E. Stanley, (2013) . "Quantifying trading behavior in financial markets using Google Trends" , Scientific reports, Vol.3, No.1684.
- [81] Rua, A., and L. C. Nunes, (2009). "International Comovement of Stock Market Returns: A Wavelet Analysis," Journal of Empirical Finance, Vol.16, No.4, 632~639..
- [82] Shynkevich, Y., T. M. McGinnity, S. A. Coleman, and A. Belatreche, (2016). "Forecasting Movements of Health-Care Stock Prices Based on Different Categories of News Articles Using Multiple Kernel Learning," Decision Support Systems, Vol.85, No.1, 74~83.
- [83] Vui, C. S., G. K. Soon, C. K. On, R. Alfred, and P. Anthony, (2013). "A Review of Stock Market Prediction with Artificial Neural Network (Ann), "IEEE International Conference on Control System, Computing and Engineering: IEEE, 477~482.
- [84] Yun, J. S., S. G. Yoon, and C. H. Hong, (2008). "Momentum

and Contrarian Strategies and Behavior of Foreign Investors in
Korean Stock Market," International Area Studies Review,
Vol.12, No.3, 195~216.



ABSTRACT

Stock Trend Prediction based on Unstructured Bigdata Improvement Model of Transformer NLP

Bumsu Kim

Department of Medical IT Marketing, Graduate School,

Eulji University

(Supervised by Professor Yong Gyu Jung, Ph. D.)

Abstract

The social media such as Internet news becomes to be important source in the stock trading market, which is influenced by opinion leader and investment sentiment. Since 2012, there have been studied on predicting stock price trends through sentiment analysis of various social media article based on NLP.

The common method of previous studies are as follows. First of all, after crawling the news related to the companies, which indexes whether the news is positive or negative using the number of positive and negative words in emotional vocabulary dictionary. Then, with the quantity of positive news and negative news, the sentiment index of the company for that date is calculated. Studies were conducted in the form of modeling the relationship between the stock price trend (up and down) of the stock according to this sentiment index using machine learning such as Support Vector Machine (SVM), K–Nearest Neighbor (KNN), and Decision Tree (DT). In conclusion, the accuracy of predicting the rise or fall with the sentiment score of the news was about 66% on average. However, these studies are undergoing a feature engineering stage called sentiment indexing that converts unstructured data into structured data. It takes a bit of effort, and the result depends on the quality of the sentiment dictionary, which categorizes positive and negative vocabularies. In this study, using KoBERT, a pre-trained model applying the latest deep learning algorithm called Transformer, developed a model that predicts stock price trends by using unstructured data based on news text as it is without cumbersome

Feature Extraction. As a result of the performance comparison, the statistically significant improvement effect was verified with an average of 10% or more (average accuracy of 77%) and a p-value of 0.05 or less compared to the existing news sentiment index-based stock price trend prediction model. In other words, it was confirmed that the model using unstructured data is significant without feature engineering that changes to the cumbersome structured data. In order to learn unstructured data and structured data together, it is necessary to convert unstructured data into structured data. However, on the contrary, I proposed a convenient method of including structured data in unstructured data as it is.

Unstructured data is 'news article', and structured data is 'press' and 'reporter'. A statistically significant difference ($p\text{-value} < 0.05$) was derived by modeling unstructured data news articles including structured data such as "press" and "reporter" as they are. Finally, I developed a model that divides the classification classes of labels into three. In the previous study, two class labels were set in the form of 'rising' or 'falling'. In this study, labels were constructed and modeled in three classes: 'continuous decline', 'fall and rise again', and 'rise'. As a result of the verification, it was verified that the p-

value value was less than 0.05 compared to the previous study with an average accuracy of 71%, showing a significant difference in performance.

This research model is expected to contribute to being able to use it for various NLP-based predictions by learning by using NLP unstructured + structured data as it is.

For example, you can predict the trend of the coin value through <Tweet content + Tweet character + Tweet time zone>. In addition, it is possible to predict the success of a web novel through <the first story of a web novel + novel writer + genre + sentence length>. It can also be used to predict disease through <consultation content + doctor name + doctor major + hospital name>, and is expected to be applicable to prediction of viewership trends through <drama scenario + production company name + director + main actor + broadcaster>.

Keywords : NLP, BERT, Transformer, Stock Estimation, Unstructured Data, AI