creative commons
COMMONS DEED

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 원저작자를 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 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

박사학위논문

소규모 학습 데이터와 전이 학습을 활용한 경제정책 불확실성 지수

HANSUNG UNIVERSITY

2025년

한 성 대 학 교 대 학 원

경 제 부 동 산 학 과

경 제 학 전 공

이 천 우

HANSUNG
UNIVERSITY

박사학위논문
지도교수 김상봉

소규모 학습 데이터와 전이 학습을 활용한 경제정책 불확실성 지수

Economic Policy Uncertainty Index Using
Small-Scale Learning Data and Transfer Learning

HANSUNG UNIVERSITY

2025년 6월 일

한 성 대 학 교 대 학 원

경 제 부 동 산 학 과

경 제 학 전 공

이 천 우

박사학위논문
지도교수 김상봉

소규모 학습 데이터와 전이 학습을 활용한 경제정책 불확실성 지수

Economic Policy Uncertainty Index Using Small-Scale Learning Data and Transfer Learning

위 논문을 경제학 박사학위 논문으로 제출함

2025년 6월 일

한 성 대 학 교 대 학 원

경 제 부 동 산 학 과

경 제 학 전 공

이 천 우

HANSUNG UNIVERSITY

이천우의 경제학 박사학위 논문을 인준함

2025년 6월 일

심사위원장 김 정 렬 (인)

심 사 위 원 유 연 우 (인)

심 사 위 원 김 상 봉 (인)

심 사 위 원 여 효 성 (인)

심 사 위 원 전 우 소 (인)

HANSUNG UNIVERSITY

국 문 초 록

소규모 학습 데이터와 전이 학습을 활용한 경제정책 불확실성 지수

한 성 대 학 교 대 학 원
경 제 부 동 산 학 과
경 제 학 전 공
이 천 우

불확실성은 경제주체의 기대 형성과 의사결정, 그리고 실물 경제에 영향을 미치는 주요 요소임에도, 그 자체가 관측되지 않아 이를 측정하려는 다양한 노력이 진행되어왔다. 특히, Baker et al.(2016)이 제안한 경제정책 불확실성 지수(Economic Policy uncertainty Index, EPU)는 빅데이터를 활용하여 불확실성을 측정한 대표적인 사례로, 3개의 단어군(경제 단어군, 정책 단어군, 불확실성 단어군)이 동시에 포함된 뉴스기사의 빈도를 기반으로 불확실성을 측정한다. 그러나 이처럼 특정 키워드의 출현 빈도만으로 불확실성을 측정하게 되면 불확실성을 과대 측정하는 문제가 발생할 수 있으며, 뉴스기사가 내포한 다양한 문맥적 정보를 고려할 수도 없다. 이러한 기존 EPU 지수의 한계를 보완하기 위해 본 연구에서는 기계학습 기반의 감성 분석을 활용한 새로운 EPU 지수를 제안하였다.

기계학습 기반의 감성 분석을 수행하기 위해서는 감정 분류 모형을 구축

하는 과정이 필요하다. 그러나 이를 구축하기 위한 대규모 라벨링 데이터의 확보와 최신 자연어 처리 기법의 활용은 개인 연구자나 소규모 연구팀이 감당하기 어려운 막대한 시간적·자원적 부담을 요구한다. 본 연구는 이러한 현실적 제약을 완화하기 위해 소규모 학습 데이터(약 20,000개의 라벨링 데이터)와 전이 학습을 활용한 감정 분류 모형을 제안하였다. 특히, BERT와 같은 사전 훈련된 언어 모형(Pre-trained Language Model, PLM)의 높은 컴퓨팅 비용과 추론 지연 문제를 고려하여 단어 임베딩 모형을 전이 학습에 활용하여도 실용성 있는 모형을 구축하는데 주안점을 두었다.

한편, 본 연구는 본 연구의 다운스트림 태스크 난이도가 비교적 쉬운 편에 속하기 때문에 고성능 감정 분류 모형으로 작성한 EPU 지수와 본 연구 모형으로 작성한 EPU 지수 간의 차이가 크지 않을 것이라 주장한다. 본 연구에서는 이를 검증하기 위해 BERT 계열의 PLM을 활용하여 감정 분류 모형을 구축하는 과정을 별도로 다루고, 해당 모형으로 작성한 EPU 지수를 본 연구의 비교지표로 활용하였다. 그리고 두 EPU 지수로 다양한 유용성 평가를 수행하여 본 연구의 주장을 실증적으로 뒷받침하였다.

연구 과정은 ① 뉴스기사 수집, ② 텍스트 전처리, ③ 임베딩 모형 구축, ④ 감정 분류 모형 구축, ⑤ 비교 대상 모형 구축, ⑥ 유용성 평가로 나누어 수행하였다. ① 뉴스기사 수집은 이긍희 외(2020)의 단어군을 활용하여 기사 제목을 수집한 뒤, 중복 및 예외 기사를 제거한 후, 해당 기사 제목을 포털사이트에 검색하는 방식으로 수행되었다.

② 텍스트 전처리는 사전 정제, 말뭉치 정의, 문장 토큰화, 사용자 사전 구축, 형태소 분석의 5단계로 구분하여 수행하였다. 사전 정제 과정에서는 불필요한 문자열 제거, 특수 문자 변환, 소괄호 내의 단어 제거 등의 작업을 수행하였고, 말뭉치는 문장으로 정의하였다. 문장 토큰화 과정에서는 본 연구에 적합한 문장 분리기를 선정하기 위해 다양한 문장 분리기들의 성능을 평가하였다. 평가는 처리 속도, 정답률, Dice Coefficient를 기준으로 수행되었으며, 평가 결과에서는 NLTK가 문장 분리 정확도와 처리 속도 모두에서 우수한 성능을 보였다. 사용자 사전 구축은 soynlp의 명사 추출기를 활용하였으며, 그 결과 일반명사 71,126개와 고유명사 16,787개로 구성된 사용자 사전이 구

축되었다. 형태소 분석 과정에서도 본 연구에 적합한 분석기를 선정하기 위해 다양한 형태소 분석기들의 성능을 평가하였다. 평가는 품사 목록 비교, 모호성 평가, 처리 속도 비교, 사용자 사전 활용 가능성 검토로 구분하여 수행하였으며, 그 결과 Kiwi가 본 연구의 형태소 분석기로 선정되었다.

③ 단어 임베딩 모형은 Word2Vec, FastText, GloVe의 세 가지 방법론을 비교하여 구축하였으며, 각 임베딩 모형을 구축한 후에는 성능 평가를 통해 본 연구의 최적 모형을 선정하였다. 평가는 단어 유추 평가, 단어 유사도 평가, 외재적 평가로 나누어 수행하였으며, 외재적 평가는 감정 분류 태스크에서의 성능 향상을 측정하였다. 평가 결과에서는 Word2Vec가 세 가지 평가 모두에서 가장 우수한 성능을 나타내었다.

④ 감정 분류 모형은 RNN, CNN, 트랜스포머의 세 가지 딥러닝 아키텍처를 기반으로 구축하였으며, 다양한 신경망 구조와 하이퍼라미터 설정을 검토하여 최적의 성능을 도출하였다. 이때, RNN은 LSTM, BiLSTM, GRU, BiGRU에 어텐션 메커니즘을 적용한 구조를 고려하였고, CNN은 Kim(2014)의 연구에서 소개된 CNN-non-static과 CNN-multichannel을 기반으로 설계하였다. 모형 평가에서는 트랜스포머 모형이 F1 Score를 기준으로 가장 높은 성능을 기록하였으나, BiGRU+Attention 모형과의 성능 차이가 0.11%P밖에 나타나지 않았다. 또한, 추론 시간은 BiGRU+Attention이 트랜스포머 모형보다 2배 빠른 속도를 나타내, 본 연구에서는 BiGRU+Attention이 본 연구의 최적 모형이라 판단하였다.

⑤ 비교 대상 모형은 BERT 계열의 PLM을 전이 학습에 활용하여 구축하였으며, PLM은 범용 도메인에서 학습된 KLUE-BERT와 뉴스기사 도메인에 특화된 KPF-BERT, 그리고 금융 도메인에 특화된 KF-DeBERTa를 활용하였다. 모형 평가에서는 F1 Score를 기준으로 KF-DeBERTa가 가장 높은 성능을 기록하였으나, KPF-BERT와의 성능 차이는 약 0.15%P밖에 나타나지 않았다. 또한, 추론 속도는 KPF-BERT가 KF-DeBERTa보다 더 빠른 것으로 분석되어 본 연구에서는 KPF-BERT가 본 연구의 최적 모형이라 판단하였다.

⑥ 유용성 평가는 경제통계와의 상관관계 분석, 불확실성이 거시경제에 미치는 영향 분석, 당기예측 모형을 활용한 예측력 평가로 나누어 수행하였다.

EPU 지수는 본 연구 모형으로 작성한 EPU 지수(EPU_small), 비교 대상 모형으로 작성한 EPU 지수(EPU_BERT), Baker et al.(2016)이 작성한 EPU 지수(EPU), Cho and Kim(2023)이 작성한 EPU 지수(EPU_KOREA)를 각각 사용하여 EPU 지수의 적합성과 강건성을 확인하였다.

경제통계와의 상관관계 분석에서는 EPU 지수의 선행성과 경제지표와의 연관성을 평가하기 위해 교차상관관계 분석을 수행하였다. 분석 결과, EPU_small은 VIX 지수를 제외한 모든 지표와 가장 높은 상관관계를 나타냈으며, 월별 경제통계와는 1~5개월, 분기별 경제통계와는 0~1분기 선행하는 것으로 분석되었다. EPU는 경기 선행지수와 양(+), 환율 변동성과 음(-), 재화 수출입과 양(+)의 상관관계를 나타냈으며, EPU_KOREA는 민간소비, 설비투자, 재화 수출과 양(+)의 상관관계를 나타내어 경제이론과 부합하지 않는 결과를 보였다. 또한, EPU_BERT는 EPU_small과 최대상관시차가 동일하고 상관계수 값도 매우 유사하게 나타났다.

불확실성이 거시경제에 미치는 영향 분석에서는 EPU 지수의 외생성을 확인하고, 거시경제변수와의 관계에 있어서 정책적 의미를 지니는지를 확인하기 위해 그랜저 인과관계 검정과 VAR 모형의 충격반응분석을 수행하였다. 동 분석은 선행연구들을 참고하여 월별 경제변수를 이용한 분석과 분기별 경제변수를 이용한 분석으로 나누어 수행되었다. 월별 경제변수를 이용한 그랜저 인과관계 검정에서는 EPU_small과 EPU_BERT가 산업생산증가율, 주가수익률, 취업자 수 증가율에 대해 가장 뚜렷한 외생성을 나타냈으며, 원/달러 환율은 EPU_KOREA가 가장 뚜렷한 외생성을 나타냈다. 분기별 경제변수를 이용한 그랜저 인과관계 검정에서는 EPU_small과 EPU_BERT가 모든 변수에 대해 가장 뚜렷한 외생성을 나타냈다. 월별 경제변수를 이용한 충격반응분석에서는 EPU_small과 EPU_BERT가 산업생산증가율, 주가수익률, 취업자 수 증가율을 유의하게 감소시키고, 원/달러 환율과 물가상승률을 유의하게 상승시키는 것으로 나타났다. 반면, EPU와 EPU_KOREA는 주가수익률과 원/달러 환율을 제외한 나머지 변수가 전체 기간에 걸쳐 통계적으로 유의하지 않았다. 분기별 경제변수를 이용한 충격반응분석에서는 EPU_small과 EPU_BERT가 소비 및 투자 심리, 대기업 대출태도지수, 가계(일반) 대출태도지수를 유의하

게 하락시키고, 신용스프레드는 유의하게 상승시키는 것으로 분석되었다. 반면, EPU와 EPU_KOREA는 소비심리지표와 신용스프레드만 통계적으로 유의하고, 나머지 변수는 전체 기간이 통계적으로 유의하지 않았다.

당기예측 모형을 활용한 예측력 평가에서는 DFM을 기반으로 당기예측모형을 구축하고, 각 EPU 지수가 GDP 당기예측 향상에 도움이 되는지를 살펴보았다. 표본외 예측 수행 결과, EPU 지수를 반영하지 않은 기준 모형(baseline)은 RMSE 0.9107, MAE 0.6346을 기록하였으며, EPU 지수가 반영된 4개의 모형은 모두 RMSE 기준에서 예측 성능의 개선을 보였다. 특히, EPU_small이 반영된 모형은 RMSE 0.8914, MAE 0.6273을 기록하여 두 지표 모두에서 가장 우수한 예측력을 나타냈으며, EPU_BERT가 반영된 모형 또한 RMSE 0.8925, MAE 0.6294를 기록하여 이와 매우 유사한 예측력을 나타냈다. 반면, EPU와 EPU_KOREA가 반영된 모형은 RMSE를 기준으로는 Baseline보다 소폭 낮은 값을 기록하였으나, MAE를 기준으로는 오히려 예측오차가 증가하였다. 이러한 결과는 본 연구에서 작성한 EPU 지수가 기존 EPU 지수들보다 높은 현실설명력을 제공하며, 당기예측 모형의 유용한 정보변수로 활용될 수 있음을 시사한다.

본 연구의 의의는 다음과 같다. 첫째, 문어체로 작성되는 뉴스기사를 기반으로 다양한 텍스트 전처리 도구들의 성능을 정량적으로 비교·평가하였다는 점이다. 둘째, 경제정책 도메인에 특화된 단어 임베딩 모형을 구축하고, 단어 임베딩 기법을 비교·평가하였다는 점이다. 셋째, 딥러닝 아키텍처를 비교·평가하여 최적의 성능을 도출하고, 유의미한 시사점을 제시하였다는 점이다. 넷째, 한글 뉴스기사를 기반으로 직접 라벨링 데이터를 구축하고, 소규모 학습 데이터와 전이 학습을 활용하여 감정 분류 모형을 구축한 최초의 사례라는 점이다. 다섯째, 처음으로 PLM과 단어 임베딩 모형의 실용성을 실무적으로 비교하였다는 점이다. 여섯째, 기존 EPU 지수의 방법론적 한계를 극복하고, 보다 현실설명력이 높은 불확실성 지수를 개발하였다는 점이다. 일곱째, 소규모 학습 데이터를 사용함으로써 하위 부문별 EPU 지수 구축을 위한 실증적 기반을 마련하고, 나아가 뉴스기사를 활용한 다양한 정보변수 개발의 토대를 제공하였다는 점이다.

【주요어】 경제정책 불확실성 지수, 소규모 학습 데이터, 전이 학습, 빅데이터, 자연어 처리, 인공 신경망, 딥러닝

HANSUNG UNIVERSITY

목 차

표 목 차

HANSUNG UNIVERSITY

그 림 목 차

HANSUNG UNIVERSITY

Ⅰ. 서 론

1.1 연구의 배경 및 목적

1.1.1 불확실성(Uncertainty) 지수의 작성

경제학에서 불확실성은 미래에 전개될 상황을 정확히 예측할 수 없거나 어떤 상황이 발생할 가능성을 명확히 측정할 수 없는 상태로 정의된다(장민·황인도, 2004).[1] 시장경제에서 불확실성은 완전히 제거할 수 없는 불가피한 현상이지만, 불확실성의 과도한 확대는 국가 경제에 불필요한 부작용을 발생시킨다. 특히, 자신의 정보집합을 토대로 합리적 의사결정을 내리는 경제주체들은 불확실성으로 인해 미래에 대한 기대 형성이 어려워지고, 이로 인해 가계와 기업은 위험을 회피하고 관망(wait and see)하려는 유인이 제공되어 소비와 투자를 크게 위축시킨다. 또한, 불확실성이 높은 시기에는 경제전망 및 정책분석의 실효성도 크게 떨어지기 때문에 정책당국 입장에서는 국가 경제의 변동요인을 신속히 파악하고 이에 대한 대응책을 마련하기 위해 불확실성을 적시에 측정할 필요가 있다.

불확실성이 국가 경제에 부정적인 영향을 미친다는 사실은 적어도 이론적으로 다양하게 제시되어왔다. 그러나 이러한 이론이 설득력을 가질 수 있는 실증적 연구는 상대적으로 미미한 편인데, 이는 주로 불확실성이 비관측 변수(unobservable variable)인 것에서 기인한다. 즉, 실증분석을 수행하기 위해서는 불확실성을 대변하는 대리변수(proxy variable)를 이용하여 간접적으로 측정해야 하는 데다, 대리변수의 선정 역시도 쉽지 않기 때문이다. 이에 따라 경제학 문헌에서는 불확실성을 측정하기 위한 다양한 방법론이 제시되어왔는데, 이는 불확실성 측정에 사용된 자료를 기준으로 크게 서베이(survey) 기반, 경제변수 기반, 텍스트 데이터 기반으로 구분된다.

1) 사실 경제학에서 불확실성 의미에 대한 합의된 논의는 없다. Knight(1921)는 흔히 분산으로 측정되는 위험(risk)과 확률분포로 나타낼 수 없는 불확실성을 구분하여 정의하였으나, 현실 경제에서는 이를 구분하기가 어려워 위험을 불확실성에 포함하거나 서로 혼용하여 정의하는 경우가 많으며, 또는 단순히 확률분포 상의 위험을 불확실성이라 표현하기도 한다.

첫 번째 서베이 기반은 불확실성 관련 설문조사를 전문가 또는 일반인에게 시행하여 설문조사에 나타난 이산(dispersion) 정도로 불확실성을 측정하는 방법이다. 이와 관련하여 Lahiri and Sheng(2010), Clements(2014)는 전문가 전망의 분산을 활용하여 불확실성을 측정한 바 있으며, Bachman et al.(2013)은 독일 기업을 대상으로 경영 여건 전망 등에 대한 설문조사를 시행하여 불확실성을 측정한 바 있다. 그러나 서베이 기반은 단순히 응답자의 견해 차이로 나타난 결과일 수 있으며, 군집행동(herding behavior)이 적절히 제어되지 않는다면 측정에 편의(bias)가 생길 수 있다는 단점이 있다. 게다가 설문조사는 일반적으로 자주 시행되는 것이 아니므로 불확실성의 월별 또는 분기별 자료를 구축하는데 제한이 따를 수도 있다.

두 번째 경제변수 기반은 경제통계 자료와 계량모형을 활용하여 불확실성을 측정하는 방법이다. 과거 금융위기 이전에는 동 방법으로 지역(local) 불확실성을 측정하는 시도가 많았으나,[2] 금융위기를 계기로 거시경제 전체에 대한 불확실성 측정의 필요성이 제기되면서 최근에는 전역(global) 불확실성을 측정하려는 시도가 늘어나고 있다. IMF, St. Louis Fed 등은 VIX, 환율 변동성, 신용스프레드 등의 다양한 변동성 지표들을 이용하여 금융시장 전반의 불확실성을 나타내는 금융스트레스지수(Financial Stress Index)를 편제하고 있으며, Haddow et al.(2013)은 영국 경제를 대상으로 불확실성과 관련된 다양한 대리변수를 선정한 뒤 이들에 대해 주성분 분석(principal component analysis)을 수행하는 방식으로 거시경제 불확실성 지수를 작성하였다. 또한, Jurado et al.(2015)은 경제변수 전망 시 예측할 수 없는 요소(unpredictable component)가 많을수록 불확실성이 커진다는 관점하에 전망모형의 예측오차(forecast error)의 분산을 이용하여 개별 경제변수들의 불확실성을 측정하고, 이들을 종합하여 거시경제 불확실성 지수를 작성하였다.

국내에서도 경제변수 기반의 방법론들을 원용하여 우리나라의 거시경제 불확실성을 측정한 연구들이 있다. 윤용준·이경태(2013)는 IMF의 금융스트레스지수에 Bloom et al.(2011)의 뉴스 기반 불확실성(news-based uncertainty)

2) 주식시장의 내재 변동성을 측정한 VIX, VKOSPI 등이 이에 해당하며, 이외에도 환율의 내재 변동성이나 소득 불확실성을 측정하기 위해 ARCH, GARCH 등의 계량모형을 활용한 연구도 상당수 존재한다.

지수[3]를 추가하여 거시경제 불확실성 지수를 작성하였으며, 이현창·정원석(2016)과 김광민(2021)은 Haddow et al.(2013) 방법론을, 이항용·이진(2017)은 Jurado et al.(2015)의 방법론을 원용하여 거시경제 불확실성 지수를 작성하였다. 다만, 김광민(2021)의 연구에서는 Haddow et al.(2013)과 Jurado et al.(2015)의 두 가지 방법으로 불확실성을 측정하여 비교하였을 때, Haddow et al.(2013)의 방법론으로 작성한 불확실성 지수가 우리나라의 불확실성 이벤트를 더 잘 설명하는 것으로 나타났다.

세 번째 텍스트 데이터 기반은 텍스트 마이닝(Text Mining)[4] 및 자연어 처리(Natural Language Process, NLP)[5] 기법을 활용하여 텍스트 데이터가 지닌 특정 정보를 추출하고 이를 이용해 불확실성을 측정하는 방법이다. 이는 다시 텍스트 데이터의 종류에 따라 중앙은행 커뮤니케이션을 이용한 방법과 뉴스기사를 이용한 방법으로 나누어진다.

중앙은행 커뮤니케이션을 이용한 방법은 중앙은행 보고서, 의결문, 기자간담회 등의 텍스트 데이터를 이용하여 중앙은행이 평가하는 불확실성의 정도를 측정한다. 이와 관련하여 Lucca and Trebbi(2011)는 FOMC 의결문에 사용된 단어의 극성(polarity)을 분석하여 심리지수를 편제한 뒤 이를 통해 향후 통화정책 방향에 대한 상당한 정보가 의결문에 포함되어 있음을 확인한 바 있으며, ECB는 기자회견문에 담긴 어조(tone)가 ECB의 통화정책 방향에 유의한 설명력을 지니고 있음을 보고하였다. 또한, Evans et al.(2015)은 통화정책 의사록을 통해 연준 FOMC 위원들이 평가하는 불확실성을 지수화하였으며, Castelnuovo and Tran(2017)은 Fed의 경제동향보고서(Beige Book)와 호주 중앙은행의 경제상황보고서(Statement on Monetary Policy)에서 불확실성

3) 언론에서 경제정책 변화가 언급되는 빈도수 등을 이용하여 측정한 지수이다.

4) 마이닝(mining)이란 데이터로부터 통계적 의미가 있는 특성과 정보 등을 추출하는 일련의 과정을 말한다. 이때 해당 데이터가 정형 데이터(structured data)인 경우를 데이터 마이닝(Data Mining), 비정형 데이터(unstructured data)인 경우를 텍스트 마이닝(Text Mining)이라 한다. 데이터 마이닝은 기계학습(Machine Learning)만으로도 처리되지만, 텍스트 마이닝은 기계학습 이외에도 자연어 처리 및 문서 처리 기술 등이 추가로 요구된다.

5) 자연어 처리는 인간의 언어를 기계가 묘사할 수 있도록 연구 및 구현하는 인공지능(AI)의 주요 분야이다. 즉, 비정형 데이터를 기계가 이해할 수 있도록 정형 데이터로 변환·정제하는 일련의 과정을 말한다. 텍스트 마이닝은 이러한 자연어 처리에 많이 의존하고 있지만, 한국어는 영어보다 기술적 진척이 부족한 편이며, 의역, 띄어쓰기, 높임법, 주어 및 목적어의 빈번한 생략 등의 이유로 분석도 상당히 어려운 편에 속한다.

관련 어휘를 선정한 뒤 이들 어휘에 대한 검색 빈도를 구글 트랜드에 검색하는 방식으로 불확실성을 측정한 바 있다.

중앙은행 커뮤니케이션을 이용한 국내 연구로는 Lee et al.(2019)과 이상우·주동헌(2021)이 있다. Lee et al.(2019)은 한국은행 금융통화위원회 의사록을 활용하여 논조 지수를 구축한 뒤 동 지수가 기준 금리에 대한 설명력과 예측력을 가진다는 연구 결과를 제시하였으며, 이상우·주동헌(2021)은 한국은행 통화신용정책 보고서에서 불확실성 관련 어휘를 선정한 뒤 동 어휘에 대한 검색 빈도를 구글 트랜드에 검색하여 불확실성 지수를 작성하였다.

뉴스기사를 이용한 방법은 뉴스기사가 내포하고 있는 불확실성 관련 정보들을 추출하여 이를 통해 불확실성의 정도를 측정하는 방법이다. 대표적인 사례로 Baker et al.(2016)이 작성한 경제정책 불확실성(Economic Policy Uncertainty, EPU) 지수가 있다. 동 지수의 작성 방법을 요약하면 다음과 같다. 첫째, 주요 언론사와 E(economy), P(policy), U(uncertainty)의 단어군을 선정한다. 둘째, 세 단어군이 동시에 한 개 이상 포함된 뉴스기사를 추출하고 이를 언론사별 월별로 합한다. 셋째, 추출된 기사 건수를 언론사별 총 월별 기사 건수로 나누어 언론사별 상대 빈도수를 구한다. 넷째, 언론사별 표준편차와 평균을 이용하여 표준화를 수행하고 불확실성 지수를 작성한다. 이렇게 작성된 EPU 지수는 현재 경제정책 불확실성 웹사이트[6]를 통해 매월 편제되고 있으며, 미국 EPU 지수뿐만 아니라 우리나라를 포함한 세계 각국의 EPU 지수도 함께 작성, 제공되고 있다. 그러나 해당 웹사이트에서 제공되고 있는 EPU 지수들은 미국 EPU 지수의 단어군을 그대로 번역해 사용하다 보니 각 나라의 언론사와 용어가 충분히 고려되지 못한다는 단점이 있다. 이에 따라 일본, 중국, 홍콩 등에서는 언론사 및 검색 단어군을 달리하여 EPU 지수를 작성하고 있으며, 우리나라 역시도 한국개발연구원, 이긍희 외(2020), Cho and Kim(2023)이 우리나라 사정에 맞는 언론사와 단어군을 선정하여 EPU 지수를 작성한 바 있다. 이중 한국개발연구원과 Cho and Kim(2023)이 작성한 EPU 지수는 현재 한국개발연구원 웹사이트와 경제정책 불확실성 웹사이트를 통해 매월 편제되고 있다.

6) https://www.policyuncertainty.com/

본 연구에서는 이상의 다양한 불확실성 측정 방법 중 텍스트 데이터를 활용한 접근법에 주목하였다. 앞서 살펴본 중앙은행 커뮤니케이션을 활용한 기존 연구들은 분석 방법에 따라 단어의 출현 빈도를 활용하는 방식과 감성 분석(Sentiment Analysis)[7]을 수행하는 방식으로 구분될 수 있다. EPU 지수는 이중 전자의 방법을 채택하고 있다. 즉, EPU 지수는 불확실성과 관련된 특정 키워드(예: 불확실성, 걱정, 우려 등)가 뉴스기사 본문에 포함된 경우, 해당 기사를 불확실성이 높은 것으로 해석한다. 그러나 이러한 방식은 불확실성 해소와 관련된 뉴스기사까지도 불확실성이 높은 것으로 해석하기 때문에[8] 불확실성을 과대 측정하는 경향이 있다. 게다가 EPU 지수의 U 단어군만으로는 뉴스기사에서 사용되는 불확실성의 다양한 표현을 고려하기가 힘들며,[9] 뉴스기사가 내포하고 있는 다양한 문맥적 정보와 문장의 구조 등도 고려할 수가 없다.[10] 본 연구에서는 이러한 기존 EPU 지수의 한계를 보완하기 위해 EPU 지수의 경제정책(E, P) 단어군으로 뉴스기사를 수집하고, 해당 뉴스기사에 감성 분석을 수행하는 방식으로 EPU 지수를 작성하고자 한다. 즉, 본 연구의 주된 목적 중 하나는 기존 EPU 지수의 단점을 보완한 새로운 EPU 지수의 작성 방법을 제시하는 것이다.

본 연구에서 제시하는 불확실성 지수가 기존 연구들과 구분되는 또 다른 차별점은 감성 분석을 수행하는 방식에 있다. 감성 분석의 수행방식은 크게 사전 기반 접근법(lexicon based approach)과 기계학습 기반 접근법(machine learning based approach)으로 구분된다. 여기서 사전 기반 접근법은 미리 구축된 감성사전(sentiment lexicons)을 활용하여 텍스트 내 단어들의 극성을 매칭하고(예: 매우 긍정, 긍정, 중립, 부정, 매우 부정), 이를 통해 감정 분류(sentiment classification) 태스크(task)를 수행하는 방식이다. 앞서 언급하였던

---

7) 텍스트 데이터를 분석하여 해당 텍스트가 긍정적, 부정적, 또는 중립적 감정을 표현하는지를 분류하는 분석 방법이다.

8) 예를 들어 “우려 섞인 목소리가 줄어들었다.”, “걱정이 줄었다.” 등의 긍정적인 문장이 포함된 뉴스기사도 ‘우려’, ‘걱정’이란 단어가 포함되어 있기 때문에 불확실성이 높은 것으로 해석한다.

9) 뉴스기사에서는 ‘불투명한 미래’, ‘VIX의 상승’, ‘예측 불가능’ 등과 같이 불확실성을 다양하게 표현할 수 있지만, EPU 지수의 U 단어군만으로는 이러한 표현을 모두 고려하기가 힘들다.

10) “불확실성이 어제는 증가하였지만, 오늘은 감소하였다.”, “취업자 수는 감소보다 증가가 더 컸다.” 등과 같은 문장이 긍정적인지 부정적인지를 구분할 수가 없다.

Lee et al.(2019)의 연구가 이에 해당하며, 홍성욱 외(2020)도 사전 기반 접근법을 활용하여 제조업 주요 업종들의 불확실성 지수를 작성한 바 있다. 특히, 사전 기반 접근법은 분석 대상 도메인(domain)에 특화된 감성사전을 구축하는 것이 중요한데, Lee et al.(2019)은 이를 위해 n-gram과 나이브 베이즈 분류기(Naive Bayes Classifier)를 활용하여 경제·금융 분야에 특화된 감성 사전을 구축하였다는 점에서 의미가 있다. 그러나 사전 기반 접근법은 중립적 또는 복합적 감정을 처리하기가 어려우며, 텍스트 데이터의 문맥적 정보, 문장의 구조 및 단어 간의 관계를 충분히 반영하지 못한다는 단점이 있다. 즉, 기존 EPU 지수의 작성 방법과 비슷한 한계점을 지닌다.[11] 이를 고려하여 본 연구에서는 기존 선행연구들과 달리 기계학습 기반 접근법을 활용하여 불확실성 지수를 작성하고자 한다. 즉, 본 연구의 또 다른 목적은 감정 분류 태스크를 수행하기 위한 경제정책 도메인에 특화된 감정 분류 모형을 구축하는 것이다. 그리고 이를 위해 논문 본문에서는 다양한 임베딩(Embedding) 기법과 딥러닝(Deep Learning) 모형들을 비교·평가하는 과정이 진행되며, 텍스트 전처리 과정에서는 경제정책 도메인에 특화된 사용자 사전을 구축하는 과정 등이 진행된다.[12]

1.1.2 감정 분류(Sentiment Classification) 모형 구축

경제분석에서 빅데이터를 활용하는 방안은 수년 전부터 많은 논의가 이루어져 왔다. 특히 한국은행, 통계청 등의 주요 통계·경제 기관들은 빅데이터의 중요성을 인식하여 관련 연구용역과제를 외부에 공모해 왔으며, 빅데이터 전담 부서를 신설·운영하는 등 다각적인 노력을 기울여 왔다.

이러한 노력에 힘입어 경제학 문헌에서도 빅데이터를 활용한 사례가 점차 증가하고 있는데, 그중에서도 가장 활발히 사용되고 있는 빅데이터가 텍스트 데이터이다. 텍스트 데이터는 경제학 분야에서 유용한 정보를 제공하는 경우

11) EPU 지수는 단어의 극성을 분류하는 태스크를 수행하진 않지만, 단어군이라는 특정 단어 사전을 이용하여 뉴스기사를 분류한다는 점에 있어서 사전 기반 접근법으로 분류할 수 있다.

12) 사용자 사전과 이를 구축하는 작업은 II장에서 다루어지며, III장에서는 이를 이용하여 경제정책 도메인에 특화된 임베딩 모형을 구축한다.

가 많다. 특히 개별 경제주체들의 합리적 선택의 기반이 되는 정보집합은 개인의 지식과 경험 또는 인적 네트워크를 통해 형성되기도 하지만, 웹 검색이나 소셜 미디어와 같은 텍스트 데이터를 통해 형성되는 경우가 많아 경제주체들의 심리를 파악하는 데 유용하게 활용될 수 있다. 또한, 텍스트 데이터는 크롤링(crawling)이나 API 등을 통해 실시간 수집이 가능하여 당기예측과 같은 분석에서도 유용한 정보변수로 활용될 수 있으며, 거의 모든 주제를 포괄할 만큼 데이터의 다양성과 범용성이 크기 때문에 경제, 산업, 지역 등 다양한 부문의 정보변수를 구축하는 것도 가능하다.

이러한 장점들 덕분에 경제학 분야에서도 텍스트 데이터를 활용한 사례가 점차 증가하고 있지만, 국내 경제학 문헌에서 사용되고 있는 텍스트 마이닝 방법론은 여전히 제한적이다. 대부분의 연구가 관련 어휘의 출현 빈도를 활용하는 데 그치는 경우가 많으며, 특히 본 연구와 같이 감정 분류 모형을 구축하여 지수를 개발한 사례는 한국은행의 뉴스심리지수(News Sentiment Index, NSI)[13]가 유일한 것으로 보인다.

이처럼 국내 경제학 문헌에서 텍스트 마이닝 방법론이 다양하지 않은 이유는 최신 자연어 처리 기법의 활용과 모형 구축 과정에서 발생하는 여러 현실적 제약이 크기 때문으로 사료 된다. NSI를 구축한 서범석 외(2022)의 연구에 따르면, NSI는 트랜스포머(Transformer) 모형과 16명의 통계조사원이 작성한 약 44만 개의 라벨링 데이터를 기반으로 학습된 덕분에 감정 분류 모형의 성능이 96%에 달할 정도로 매우 우수한 성능을 보유하고 있다. 또한, NSI는 경제 뉴스기사에 특화된 감정 분류 모형을 구축한 독보적 사례인 만큼 경제학 분야에 활용되는 텍스트 마이닝의 방법론을 확장하는 데 중요한 벤치마크로 활용될 가능성이 있다. 따라서 본 연구 역시 이를 벤치마크 삼아 모형을 구축할 수도 있지만, 개인 연구자가 이와 같은 고성능 모형을 구축하고 활용하는 데에는 여러 현실적인 어려움이 따른다. 특히 대규모 라벨링 데이터 구축과 최신 자연어 처리 기법의 활용은 막대한 시간적, 자원적 투자를 요구하며, 이는 개인 연구자나 소규모 연구팀이 감당하기 어려운 경우가 많다.

13) 한국은행 경제통계시스템(ecos.bok.or.kr)에서 2022년 2월부터 실험적 통계(2022-001호)로 공개하고 있는 지수이다. 해당 데이터는 새로운 유형의 데이터 또는 새로운 방식을 활용하여 실험적으로 작성되는 통계로 국가통계는 아니다.

이에 본 연구는 이러한 현실적 제약을 완화하고자, 감정 분류 모형을 구축하는 데 있어 다음과 같은 두 가지 사항을 연구 목적으로 설정하였다.

첫 번째는 NSI의 약 5% 수준인 2만 개의 학습 데이터만 사용하여 실효성 있는 감정 분류 모형을 구축하는 것이다. 일반적으로 감정 분류 모형의 성능을 높이는 가장 효과적인 방법은 대규모 라벨링 데이터를 확보하여 모형 학습에 활용하는 것이다. 그러나 라벨링 작업은 노동집약적 수작업으로 이루어지기 때문에 NSI와 같이 대규모 라벨링 데이터를 구축하는 작업은 현실적으로 매우 큰 부담으로 작용한다.[14] 게다가 경제 분야의 라벨링 작업은 전문적인 지식이 요구되는 만큼 일반인에게 쉽사리 의뢰하기도 어려우며, 다수의 인원이 작업을 하였더라도 라벨링 기준이 일관되지 않으면 모형의 설명력이 크게 저하되기 때문에 철저한 검수 작업도 필요하다. 본 연구에서는 이러한 라벨링 작업의 부담을 줄일 수 있도록 소규모 학습 데이터로도 실효성 있는 모형을 설계하는 데 목적을 두었다.

두 번째는 컴퓨팅 비용이 낮은 경량화된 감정 분류 모형을 구축하는 것이다. 본 연구와 같이 소규모 학습 데이터로 감정 분류 모형의 성능을 높이기 위해서는 전이 학습(Transfer Learning)의 활용이 필수적이다. 특히, 현재 자연어 처리 분야에서는 BERT(Biderectional Encoder Representation from Transformers)나 GPT(Generative Pre-trained Transformer)와 같은 사전 훈련된 언어 모형(Pre-trained Language Model, PLM)을 활용한 전이 학습이 뛰어난 성능을 입증하고 있어, 본 연구에서도 이를 활용한다면 고성능 모형을 구축하는 것이 가능하다. 그러나 문제는 이들 PLM이 사전 훈련과 추론 과정에서 높은 컴퓨팅 비용을 요구한다는 것이다. 이제는 소형 PLM이 된 BERT의 경량화된 버전조차 사전 훈련에 최소 $1,700(한화 약 220만 원)의 비용이 들어가기 때문에(Devlin et al., 2018), 모형의 성능 향상과 평가 과정까지 고려한다면 실제 서비스할 수 있는 모형을 구축하기까지 적지 않은 비용이 발생한다. 또한, PLM은 대규모 학습 파라미터에 기인한 막대한 연산 비용과 추론 지연(latency) 문제로 인해 실시간 분석이 필요한 환경에서 실용적이지 못한 상황을 발생시킨다. 본 연구의 경우에도 매일 대량의 뉴스기사를 수집하

14) 저자의 경험을 기반으로 계산하였을 때, NSI와 같은 44만 개의 라벨링 데이터를 구축하는데 필요한 시간적 비용은 약 1,470시간(10시간씩 147일)이다.

여 분석을 수행해야 하는데, 이러한 상황에서 추론 지연 문제까지 발생한다면 분석 자료의 수를 줄이거나 모형의 크기를 축소해야 하는 선택을 강요받게 된다.[15] 이러한 현실적 제약을 고려하여 본 연구에서는 BERT 계열의 PLM 대신 사전 훈련된 단어 임베딩(Pre-trained Word Embedding) 모형을 활용한 감정 분류 모형을 제안한다. 단어 임베딩 기반 전이 학습은 성능 면에서는 BERT보다 열위일 수 있으나, 컴퓨팅 자원이 적게 소요되고 추론 속도가 빠르다는 장점이 있다. 따라서 불확실성을 신속히 측정해야 하는 본 연구에 더 적합하며, 나아가 본 연구 모형의 실용성이 입증된다면 향후 주가 불확실성이나 시장 예측과 같은 실시간 분석이 요구되는 환경에서도 유용하게 활용될 수 있을 것이다.

한편, 본 연구의 EPU 지수는 모형의 성능으로 인해 한 가지 의구심이 제기될 수 있다. 본 연구에서 활용할 단어 임베딩 기법은 다의어 및 동음이의어를 구별하지 못한다는 한계가 있어 BERT보다 문맥을 이해하는 능력이 부족한 편이다. 이로 인해 본 연구의 감정 분류 모형은 BERT를 활용한 경우보다 모형의 성능이 상대적으로 낮을 수밖에 없으며, 이는 곧 본 연구 모형으로 작성한 EPU 지수가 BERT를 활용하여 작성한 EPU 지수보다 신뢰성이 다소 떨어질 수 있다는 의구심을 제기하게 된다.

그러나 본 연구에서는 본 연구 모형으로 작성한 EPU 지수와 BERT를 활용하여 작성한 EPU 지수 간의 차이가 크지 않을 것이라 주장한다. 이러한 주장이 가능한 이유는 본 연구의 다운스트림 태스크가 비교적 쉬운 난이도에 속하기 때문이다. 예를 들어 본 연구 모형과 BERT를 활용한 모형 간의 성능 차이가 5%라고 가정해 보자. 이는 BERT가 정답으로 분류한 문장 중 5%는 본 연구 모형이 정확히 분류하지 못했음을 의미한다. 여기서 본 연구 모형이 정답으로 분류하지 못한, 즉 분류 난이도가 높은 5%에 해당하는 문장들은 전체 뉴스기사에서 차지하는 비중이 그리 크지 않을 것이라는 게 본 연구의 주장이다. 만약 문장이 아닌 문단으로 감정 분류를 수행하거나, 하나의 문장 내에서도 특정 단어의 감정이 복합적으로 표현되어 문장이 길고 구조적으로 복잡한 경우라면 BERT와 같은 고성능 모형이 필요할 수도 있다. 그러나 문어

15) 실제로 NSI도 컴퓨팅 비용을 낮추기 위해 입수한 뉴스기사 중 일부의 표본만 추출하여 작성되고 있다.

체로 작성되는 뉴스기사는 문장으로 분리하기가 매우 쉬운 편이며, 문장의 길이도 짧아 그러한 사례는 드물 것으로 판단된다. 다만, 경제 뉴스기사는 전문용어가 다수 사용되는 만큼 태스크의 난이도가 상승할 가능성도 존재한다. 하지만 이는 형태소 분석기의 사용자 사전을 구축함으로써 해결할 수 있으며, 단어 임베딩 모형의 단점인 다의어 및 동음이의어 문제는 형태소 분석과 품사 표식(PoS Tagging: Part-of Speech Tagging) 과정을 통해 어느 정도 보완할 수 있을 것으로 판단된다.

이를 검증하기 위해 본 연구에서는 BERT 계열의 PLM을 활용하여 감정 분류 모형을 구축하고, 이를 통해 작성된 EPU 지수를 본 연구의 비교지표로 활용하고자 한다. 따라서 논문 본문에서는 본 연구 모형을 구축하는 과정뿐만 아니라 BERT를 활용한 감정 분류 모형의 구축 과정 또한 별도로 다룰 것이다.

HANSUNG UNIVERSITY

1.2 연구 범위 및 방법

1.2.1 연구 범위

본 연구의 목적은 감정 분류 모형의 구축과 이를 활용한 새로운 EPU 지수의 작성으로 구분된다. 감정 분류 모형 구축의 주된 목표는 경제정책 뉴스 기사에 특화된 모형을 개발하고, 소규모 학습 데이터와 전이 학습을 활용하여 높은 효율성과 성능을 동시에 갖춘, 즉 가성비 좋은 모형을 개발하는 것이다. EPU 지수는 기존 EPU 지수들보다 높은 현실설명력을 제공하고, 속보성을 갖는 고빈도 경제지표로 작성되는 것을 목표로 한다.

본 연구의 목적을 달성하기 위해 다루어야 할 사항은 다음과 같다.

첫 번째는 뉴스기사의 수집 방법이다. 뉴스기사는 특정 주제에 국한되지 않고 경제, 정치, 사회, 문화 등 다양한 분야를 다루기 때문에 텍스트 데이터 중에서도 주제의 다양성과 범용성이 매우 큰 데이터이다. 따라서 주제와 무관한 뉴스기사가 분석에 포함되지 않으려면 연구 주제와 밀접한 관련이 있는 키워드를 선정해야 하며, 키워드를 선정한 후에는 이를 중심으로 유효한 정보의 뉴스기사를 필터링할 수 있는 수집 방안을 마련해야 한다.

두 번째는 텍스트 데이터의 전처리 방법이다. 텍스트 마이닝과 자연어 처리의 성능을 높이기 위해서는 텍스트 데이터의 품질을 향상시키는 텍스트 전처리(text preprocessing) 과정을 수행해야 한다. 텍스트 전처리의 효과는 전처리 도구와 텍스트 데이터의 종류에 따라 상이하게 나타나므로 연구에 적합한 전처리 도구를 선정하기 위해서는 각 전처리 도구들의 성능을 비교·평가하는 과정이 필요하다. 특히, 본 연구의 EPU 지수는 속보성이 중요하므로 전처리 도구들의 성능이 유사하게 나타난다면 처리 속도가 더 빠른 도구를 선택하는 것이 중요하다.[16]

세 번째는 단어 임베딩 모형의 구축 방법이다. 본 연구에서는 컴퓨팅 비용을 절감하기 위해 단어 임베딩 모형을 전이 학습에 활용한다. 그러나 단어 임베딩은 사용되는 방법론에 따라 단어를 학습하고 표현하는 방식이 다르므로

16) 반대로 처리 속도가 유사하다면 성능이 높은 전처리 도구를 선택하는 것이 합리적일 것이다.

언어의 구조, 감정 분류 태스크의 특성 등에 따라 연구에 적합한 임베딩 기법이 달라진다.[17] 게다가 같은 임베딩 기법도 하이퍼파라미터(hyperparameter) 설정에 따라 성능에 큰 차이가 나타나므로,[18] 연구에 적합한 임베딩 모형을 선정하고 성능을 극대화하기 위해서는 다양한 임베딩 기법을 비교·평가하고, 하이퍼파라미터 설정을 검토하는 과정이 필요하다. 또한, 단어 임베딩은 다의어 및 동음이의어를 구별하지 못하는 한계가 있으므로 이를 보완할 방안이 필요하며, 경제정책 도메인에 특화된 임베딩 모형을 구축하기 위해서는 경제정책과 관련된 전문 용어도 적절한 벡터값이 할당되도록 형태소 분석기의 사용자 사전을 구축하는 과정이 필요하다.

네 번째는 감정 분류 모형의 구축 방법이다. 감정 분류 모형은 구축된 임베딩 모형을 전이(transfer) 받아 텍스트의 감정을 분류하는 역할을 한다. 감정 분류 모형의 성능 역시 적용된 학습 방법, 하이퍼파라미터 설정, 신경망 구조 등에 따라 크게 달라질 수 있다. 따라서 성능과 효율성 사이에서 최적의 균형을 찾기 위해서는 다양한 모형 아키텍처를 비교·평가하고, 최적의 하이퍼파라미터를 탐색하는 과정이 필요하다. 특히, 최근 개발된 신경망 모형들은 시퀀스(sequence) 데이터의 장기 의존성(long-term dependency) 문제를 해결하는 데 중점을 두지만, 본 연구에서 다루는 텍스트 데이터는 비교적 짧은 시퀀스 길이를 가진데다 스트레이트 기사[19]도 다수 포함하고 있어 상대적으로 경량화된 모형도 높은 성능을 나타낼 가능성이 있다. 다만, 이를 위해서는 텍스트 전처리 과정에서 문장의 계층적 구조를 적절히 분리할 필요가 있다.

다섯 번째는 작성된 EPU 지수의 평가와 비교 대상 모형의 구축 방법이다.

---

17) 이와 관련하여 Park et al.(2018), 강형석·양장훈(2020) 등의 연구에서는 한국어 텍스트 데이터와 다양한 임베딩 방법론을 활용하여 임베딩 모형의 성능을 평가하였다. 특히, 감정 분류 태스크에서 Park et al.(2018)은 FastText의 성능이 높게 나타난 반면, 강형석·양장훈(2020)은 FastText보다 Word2Vec의 성능이 높게 나타났다.

18) 이와 관련하여 최정명·김유섭(2019)은 Word2Vec와 FastText의 하이퍼파라미터를 조절하면서 감정 분류 태스크의 정확도를 측정하고 비교하였다.

19) 뉴스기사는 작성 방식에 따라 스트레이트 기사와 피처 기사로 구분된다. 스트레이트 기사는 육하원칙에 따라 사실 위주로 작성되는 기사로, 핵심 사실을 빠르게 전달하기 위해 중요한 정보를 서두에 배치하는 역피라미드 구조로 작성된다. 반면, 피처 기사는 스트레이트 기사 이외에 기사를 일컬으며, 스트레이트 기사보다 사건을 심층적으로 다루고, 수용자의 관심을 끌기 위해 사례를 들어 작성하거나, 창의적인 스타일로 작성되는 경우가 많다. 따라서 피처 기사가 스트레이트 기사보다 뉴스 길이도 길고, 감정 분류 태스크의 난이도도 높다.

EPU 지수의 실용성과 신뢰성을 검증하기 위해서는 기존 선행연구들을 참고하여 다양한 유용성 평가 방법을 적용하고, 평가 결과를 면밀히 검토해야 한다. 또한 본 연구는 뉴스기사의 특성과 다운스트림 태스크의 난이도를 고려했을 때, 본 연구 모형으로 작성한 EPU 지수와 BERT를 활용하여 작성한 EPU 지수의 차이가 크지 않을 것이라 주장한다. 따라서 이를 실증적으로 검증하기 위해서는 다양한 BERT 계열의 PLM을 비교하여 본 연구에 적합한 감정 분류 모형을 구축하고, 해당 모형을 통해 작성된 EPU 지수를 본 연구의 비교지표로 활용할 필요가 있다. 즉, 작성된 EPU 지수를 평가하는 과정에서는 강건성 검증(robustness check)의 하나로 본 연구 모형으로 작성한 EPU 지수와 BERT를 활용하여 작성한 EPU 지수를 비교하는 과정이 포함되어야 한다.

1.2.2 연구 방법

본 연구의 주요 내용과 수행 흐름도는 아래 [그림 1-1]과 같다:

- **경제정책 뉴스기사 수집.** 뉴스기사 수집은 ① 단어군 선정, ② 기사 제목 수집, ③ 중복 및 예외 기사 제거, ④ 기사 본문 수집으로 나누어 수행한다:

① 단어군 선정: 단어군 선정은 경제정책과 관련된 뉴스기사를 선별하기 위해 EPU 지수의 E, P 단어군을 선정하는 단계이다. 본 연구에서는 이를 위해 이긍희 외(2020)의 단어군을 활용한다.

② 기사 제목 수집: 기사 제목 수집은 검색된 뉴스기사의 제목을 수집하는 단계이다. 본 연구에서는 빅카인즈 웹사이트[20]를 활용하여 E, P 단어군으로 검색된 뉴스기사의 제목을 수집한다.

③ 중복 및 예외 기사 제거: 뉴스기사의 제목을 수집한 후에는 중복 및 예외 기사들을 제거하는 작업을 수행한다. 여기서 예외 기사는 인사, 부고, 동정, 포토 등의 내용을 담은 기사를 말하며, 이외에도 광고성 기사나 영어, 한자 등의 다국어로 작성된 기사도 제거 대상에 포함된다.

④ 기사 본문 수집: 중복 및 예외 기사들이 제거된 후에는 기사 제목을 포털

20) 한국언론진흥재단이 제공하는 언론사로부터 뉴스기사를 수집해 다양한 분석 서비스를 제공하는 사이트이다.(http://www.kinds.or.kr)

사이트에 검색하여 기사 본문을 수집한다. 본 연구에서는 이를 위해 URL 인코딩과 웹 스크래핑(web scraping)을 활용한다. 즉, 빅카인즈에서 수집한 기사 제목을 URL 인코딩 형식으로 변경하여 포털사이트에 검색한 뒤, 검색 결과의 기사 제목과 빅카인즈의 기사 제목이 정확히 일치하는 경우에만 기사 본문이 수집되도록 코드를 설계한다.

• **텍스트 전처리.** 텍스트 전처리 과정은 ① 사전 정제, ② 말뭉치(corpus) 정의, ③ 문장 토큰화(sentence tokenization), ④ 사용자 사전 구축, ⑤ 형태소 분석으로 구분하여 수행한다:

① 사전 정제: 사전 정제는 토큰화(tokenization) 작업에 방해가 되는 불필요한 부분을 제거하기 위해 수행된다. 본 연구에서는 파이썬에서 지원하는 정규표현식(Regular Expression) 모듈 re를 활용하여 불필요한 문자열 제거, 특수문자 변환, 소괄호 내의 단어 제거 등의 작업을 수행한다.

② 말뭉치 정의: 말뭉치 정의는 모델의 입력 구조를 결정하는 과정이다. 본 연구에서는 의사 전달의 최소 단위인 문장을 말뭉치로 정의한다.

③ 문장 토큰화: 문장 토큰화는 텍스트 데이터를 문장 단위로 분리하는 과정이다. 본 연구에서는 경제정책 뉴스기사에 적합한 문장 분리기를 선정하기 위해 NLTK(Natural Language Toolkit), KSS(Korean Sentence Splitter), 그리고 문장 분리 기능을 제공하는 여섯 개의 형태소 분석기(Kiwi, OKT, 한나눔, 코모란, 은전한잎, 꼬꼬마)를 대상으로 문장 분리 성능을 평가한다. 평가는 처리 속도, 정답률, Dice Coefficient를 기준으로 수행하며, 테스트셋은 수집된 경제정책 뉴스기사를 기반으로 직접 구축한다. 또한 앞서 언급하였듯이 태스크의 난이도를 낮추려면 문장의 계층적 구조, 특히 인용문이 포함된 문장을 분리하는 과정이 필요하다. 이를 위해 본 연구에서는 인용문을 추출하는 추가적인 전처리 작업을 수행한다. 다만, KSS와 Kiwi는 인용문 분절 기능이 제공되므로 해당 기능을 활용한다.

④ 사용자 사전 구축: 사용자 사전 구축은 형태소 분석기에 미등록된 단어, 특히 경제정책 뉴스에 등장하는 전문 용어들을 정확하게 인식하고 처리할 수 있도록 형태소 분석기용 단어 사전을 구축하는 작업이다. 이를 위해 본 연구에서는 비지도 학습 기반 단어 추출기인 soynlp[21]를 활용한다. 즉, 문장 토큰

화가 완료된 텍스트 데이터를 대상으로 soynlp 알고리즘을 학습한 후, 형태소 분석기에 등록되지 않은 단어를 선별하여 사용자 사전을 구축한다.
⑤ 형태소 분석: 형태소 분석은 텍스트 데이터의 의미와 구조를 파악하기 위해 단어의 구성 요소들을 최소 의미로 분해하여 인식하는 과정을 말한다. 형태소 분석에서도 본 연구에 적합한 분석기를 선정하기 위해 Kiwi, OKT, 한나눔, 코모란, 은전한잎, 꼬꼬마를 대상으로 성능 비교를 수행한다. 평가는 품사 목록 비교, 모호성 평가, 처리 속도 비교, 사용자 사전 활용 가능성 검토를 기준으로 수행하며, 모호성 평가의 테스트셋은 바른(bareun)에서 구축한 데이터셋을 활용한다. 또한 형태소에 품사 정보를 부착하여 단어 임베딩 모형의 다의어 및 동음이의어 문제를 완화한다.

• **감정 분류 모형 구축.** 감정 분류 모형의 구축 과정은 ① 임베딩 모형 구축, ② 감정 분류 모형 구축으로 구분되며, ③ 비교 대상 모형 구축 과정이 별도로 진행된다. 각 과정의 세부 내용은 다음과 같다:
① 임베딩 모형 구축: 단어 임베딩 모형은 Word2Vec, FastText, GloVe의 세 가지 방법론을 활용하여 구축한다. 학습에 사용될 말뭉치 데이터는 텍스트 전처리 과정을 거친 문장 데이터이며, 각 문장은 품사 정보가 부착된 형태소로 구성한다. 각 임베딩 모형을 구축한 후에는 성능 평가를 통해 본 연구에 적합한 모형을 선정한다. 평가는 내재적 평가(intrinsic evaluation)와 외재적 평가(extrinsic evaluation)로 구분하며, 내재적 평가에서는 단어 유추(word analogy) 및 단어 유사도(word similarity) 평가를 수행하고, 외재적 평가에서는 감정 분류 태스크에서의 성능 향상을 측정한다. 테스트셋은 유추 평가의 경우 KATS(Korean Analogy Test Set), 유사도 평가의 경우 WordSim353을 활용한다. 임베딩 모형을 선정한 후에는 이를 감정 분류 모형의 임베딩 층으로 전이한 뒤 미세 조정(fine-tuning)을 수행한다.
② 감정 분류 모형 구축: 감정 분류 모형은 순환 신경망(Recurrent Neural Networks, RNN), 합성곱 신경망(Convolutional Neural Networks, CNN), 트랜스포머의 세 가지 딥러닝 아키텍처를 기반으로 구축한다. 이때 RNN은 LSTM, BiLSTM, GRU, BiGRU에 어텐션 메커니즘(Attention Mechanism)을

21) https://github.com/lovit/soynlp

적용한 구조를 고려하며, CNN은 Kim(2014)이 제시한 CNN-non-static과 CNN-multichannel을 기반으로 설계한다. 학습 데이터셋은 수집된 뉴스기사에서 약 2만 개의 문장을 추출해 긍정, 부정, 중립으로 분류한 뒤, 80%를 학습 데이터(training data)로, 20%를 검증 데이터(validation data)로 활용한다. 학습 데이터셋이 부족한 점을 고려해 별도의 테스트 데이터는 분리하지 않는 대신, 계층적 k-fold CV(Stratified k-fold cross-validation)를 모형 평가에 활용하여 모형의 안정성과 신뢰성을 높인다. 또한, 레이블 불균형 문제가 발생할 경우, 레이블 가중치를 모형 학습에 반영하여 불균형 데이터를 조정한다. 모형 구축 과정에서는 다양한 하이퍼파라미터 설정과 신경망 구조 등을 검토해 최적의 성능을 도출하며, 모든 모형을 구축한 후에는 분류 정확도와 추론 시간을 비교·평가하여 본 연구에 가장 적합한 감정 분류 모형을 선정한다.

③ 비교 대상 모형 구축: 비교 대상 모형은 공개된 BERT 계열의 PLM을 전이 학습에 활용하여 구축한다. 한국어 말뭉치를 기반으로 학습된 BERT 계열의 PLM으로는 KoBERT, KLUE-BERT 등이 대표적으로 활용되지만, 이들 모형은 범용적인 도메인에서 학습되었기 때문에 특정 도메인에서는 취약한 면모를 보일 수 있다. 특히, 고유명사와 전문 용어가 많이 사용되는 경제·금융과 같은 도메인에서는 OOV(Out-Of-Vocabulary) 문제로 인한 언어 이해 부족, 도메인 특화 단어들의 유의어 관계에 대한 이해 부족, 문장 내 수치에 대한 이해 부족 등의 문제가 발생할 수 있다. 본 연구에서는 이를 고려하여 본 연구와 유사한 도메인에서 학습된 도메인 특화 PLM들을 탐색하고, 해당 모형들로 전이 학습을 수행한다. 또한, 모형 구축 과정에서는 도메인 특화의 필요성을 확인하기 위해 KLUE-BERT도 함께 전이 학습을 수행하며, 모든 모형을 구축한 후에는 분류 정확도와 추론 시간을 비교·평가하여 본 연구에 적합한 모형을 선정한다.

• **EPU 지수 작성.** EPU 지수는 구축된 감정 분류 모형을 활용하여 뉴스기사의 각 문장을 긍정, 부정, 중립으로 분류한 뒤, 분류된 긍정, 부정 문장의 비율을 계산하여 작성한다. 표준화는 지수의 평균과 분산을 이용하여 평균이 100, 표준편차가 10이 되도록 조정하며, 표준화 구간과 지수의 작성 기간은

수집된 뉴스기사의 연도별 집계 결과를 바탕으로 결정한다. 또한 본 연구 모형으로 작성한 EPU 지수 이외에도 BERT 기반의 감정 분류 모형을 활용한 EPU 지수도 추가로 작성하여 본 연구의 비교지표로 활용한다.

• **유용성 평가.** 유용성 평가에서는 본 연구에서 작성한 EPU 지수가 기존 EPU 지수들보다 높은 현실설명력을 갖추었는지 검증하기 위해 기존 EPU 지수들과의 비교·분석을 수행한다. 비교 대상 지표는 웹사이트에 공개된 Baker et al.(2016)과 Cho and Kim(2023)의 EPU 지수를 포함하며, BERT 기반의 감정 분류 모형으로 작성한 EPU 지수도 함께 활용한다. 평가는 ① 경제통계와의 상관관계 분석, ② 불확실성이 거시경제에 미치는 영향 분석, ③ 당기예측 모형을 활용한 예측력 평가로 구분되며, 각 평가 방법의 세부 내용은 다음과 같다:

① 경제통계와의 상관관계 분석: 본 연구에서 작성한 EPU 지수가 새로운 정보변수로서 유용성을 가지기 위해서는 기존 지표보다 선행성이 우수하고, 거시경제변수와도 밀접한 관계가 있어야 한다. 이를 확인하기 위해 해당 평가에서는 EPU 지수와 경제통계 간의 교차상관(Cross-correlation) 분석을 수행한다.

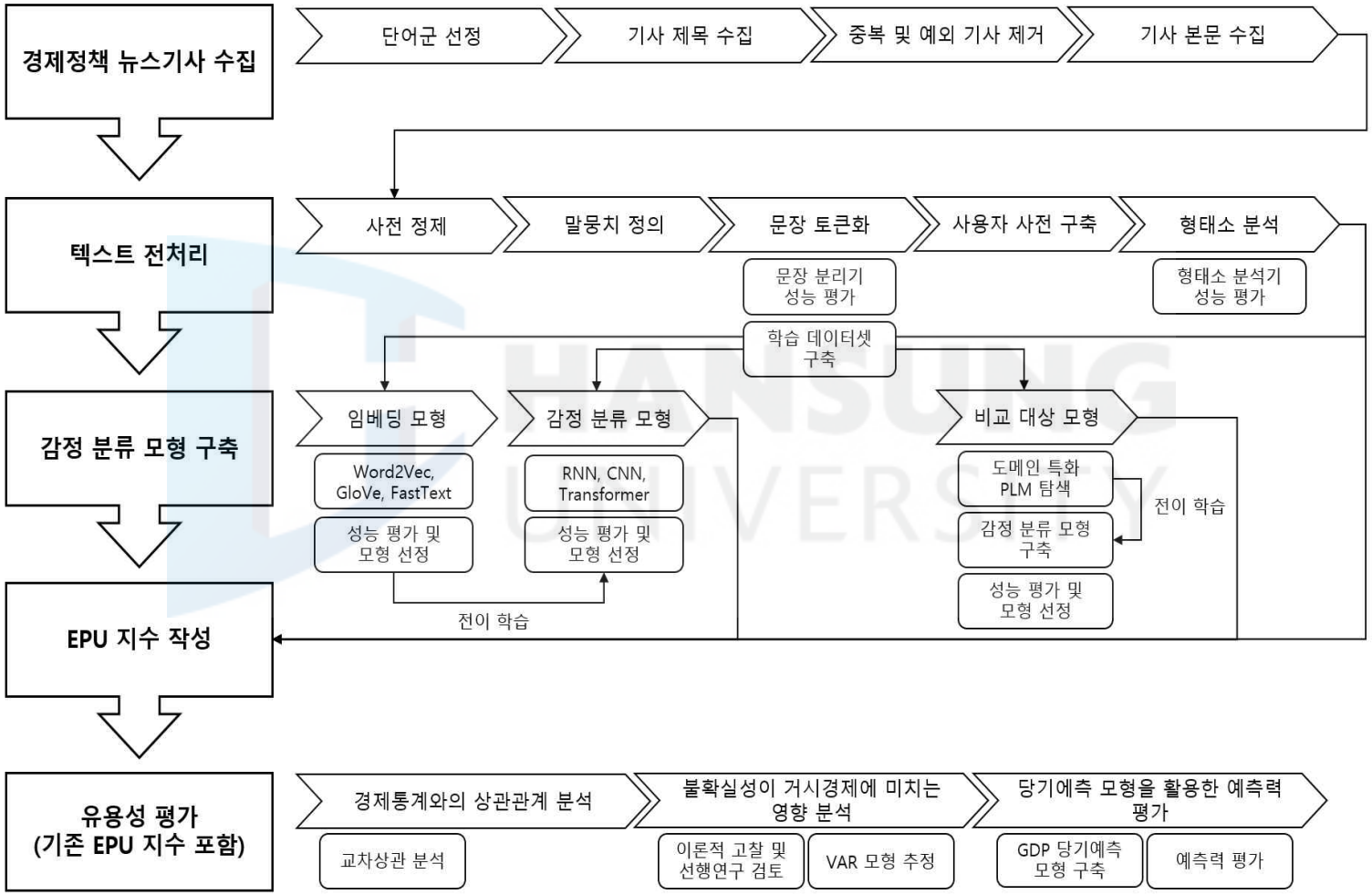
② 불확실성이 거시경제에 미치는 영향 분석: EPU 지수의 유용성을 평가하는 또 다른 방법은 해당 지수의 외생성(endogeneity)을 확인하고, 거시경제변수와의 관계에 있어서 정책적 의미를 지니는지 확인하는 것이다. 이를 위해 해당 평가에서는 EPU 지수를 불확실성의 대리변수로 사용하여 불확실성이 거시경제변수에 미치는 영향을 분석하고, 분석 결과가 이론적 직관에 부합하는지 확인한다. 주요 분석 방법은 그랜저 인과관계 검정(Granger Causality Test)과 충격반응분석(Impulse Response Analysis)이며, 벡터자기회귀(Vector AutoRegression, VAR) 모형의 설계 및 결과 해석의 타당성을 확보하기 위해 불확실성에 대한 이론적 고찰과 국내 실증연구들을 검토하는 과정도 함께 진행한다.

③ 당기예측 모형을 활용한 예측력 평가: 뉴스기사로 작성되는 EPU 지수는 속보성을 갖는 고빈도 경제지표로서 당기예측에 매우 유용하게 활용될 수 있으며, 특히 불확실성과 GDP 변동이 이론적으로 밀접하게 연관되어 있다는

점을 고려하면 GDP 당기예측 모형의 예측력 향상을 기대해 볼 수 있다. 이를 확인하기 위해 해당 평가에서는 GDP 당기예측 모형을 구축하여 본 연구의 EPU 지수가 GDP 당기예측 향상에 도움이 되는지를 살펴본다. GDP 당기예측 모형은 동태요인모형(Dynamic Factor Model, DFM)을 기반으로 구축하며, 모형 추정은 EM(Expectation Maximization) 알고리즘 기반의 최대우도추정법(Maximum Likelihood Estimate)을 활용한다.

HANSUNG UNIVERSITY

[그림 1-1] 연구의 주요 내용 및 수행 흐름도



1.3 논문의 구성

본 논문의 구성은 아래와 같다:

• Ⅰ장 서론에서는 연구의 배경과 목적, 연구의 범위 및 방법에 대해 알아본다.

• Ⅱ장 연구 자료에서는 경제정책 뉴스기사 수집, 텍스트 전처리, 학습 데이터셋에 대하여 기술한다. 경제정책 뉴스기사 수집에서는 단어군 선정 결과와 뉴스기사의 수집 방법을 살펴보고, 뉴스기사의 수집 결과를 바탕으로 EPU 지수의 작성 기간을 결정한다. 텍스트 전처리에서는 전처리 방법과 주요 전처리 도구들의 성능 평가 결과를 논의하고, 텍스트 전처리 결과를 기술한다. 학습 데이터셋에서는 구축 결과를 바탕으로 감정 분류 모형의 평가 방법과 학습 방법을 논의한다.

• Ⅲ장 연구 모형에서는 단어 임베딩 모형, 감정 분류 모형, 비교 대상 모형의 선정 과정을 기술한다. 단어 임베딩 모형의 선정 과정에서는 Word2Vec, FastText, GloVe의 방법론을 상세히 살펴보고, 세 임베딩 모형의 구축 방법과 성능 평가 결과를 논의하여 본 연구에 적합한 임베딩 모형을 선정한다. 감정 분류 모형의 선정 과정에서는 RNN, CNN, 트랜스포머의 방법론을 상세히 살펴보고, 다양한 하이퍼파라미터 설정과 신경망 구조, 학습 방법 등을 논의하여 최적의 감정 분류 모형을 제시한다. 비교 대상 모형 선정 과정에서는 전이 학습과 BERT의 방법론을 상세히 살펴보고, 한국어 말뭉치로 학습된 다양한 BERT 계열의 PLM을 검토한다. 또한, 도메인 특화 PLM을 활용한 전이 학습 수행 결과를 논의하여 본 연구에 적합한 비교 대상 모형을 선정한다.

• Ⅳ장 지수 작성 및 유용성 평가에서는 EPU 지수의 작성 결과와 유용성 평가 결과를 제시한다. EPU 지수 작성 결과에서는 본 연구 모형을 기반으로 작성한 EPU 지수와 기존 EPU 지수들을 시각적으로 비교함으로써 각 지수가 국내 불확실성 이벤트를 얼마나 적절히 반영하는지 살펴본다. 또한, 본 연구

모형을 활용하여 작성한 EPU 지수와 BERT 기반의 감정 분류 모형으로 작성한 EPU 지수를 비교하여 두 지수 간의 차이를 확인하고, 본 연구 모형의 신뢰성과 강건성을 검토한다. 유용성 평가 결과는 경제통계와의 상관관계 분석, 불확실성이 거시경제에 미치는 영향 분석, 당기예측 모형을 활용한 예측력 평가로 나누어 제시한다. 경제통계와의 상관관계 분석에서는 각 EPU 지수와 월·분기별 경제통계 자료 간의 교차상관관계를 분석하여 각 EPU 지수의 선행성과 경제지표와의 연관성을 평가한다. 불확실성이 거시경제에 미치는 영향 분석에서는 그랜저 인과관계 검정 및 충격반응분석 등을 수행하여 각 EPU 지수의 외생성과 경제이론과의 부합 여부를 평가한다. 이 과정에서는 VAR 모형 설계를 위해 변수 선정 및 시차 결정 등의 VAR 분석 절차를 상세히 기술하고, 분석 결과의 타당성과 이론적 해석의 적절성을 확보하기 위해 불확실성에 대한 이론적 고찰과 국내 선행 실증연구를 검토한다. 또한, 평가 결과는 월별 경제변수를 이용한 분석과 분기별 경제변수를 이용한 분석으로 나누어 제시하며, 특히 분기별 경제변수를 이용한 분석에서는 불확실성의 다양한 파급경로를 구분하여 각 경로별 영향을 분석하고 그 결과를 제시한다. 마지막으로 당기예측 모형을 활용한 예측력 평가에서는 GDP 당기예측 모형의 구축 및 분석과정을 설명하고, 해당 모형을 활용하여 각 EPU 지수가 GDP 예측력 향상에 기여하는지를 평가한다.

- Ⅴ장 결론에서는 본 연구의 주요 결과를 요약하고, 연구 의의를 기술한다.

Ⅱ. 연구 자료

2.1 경제정책 뉴스기사 수집

뉴스기사를 연구 자료로 활용하기 위해서는 분석 대상에 적합한 뉴스기사를 선별하는 과정이 필요하다. 이를 위해 본 연구에서는 불린(boolean) 검색[22]과 이긍희 외(2020)의 단어군을 활용하여 경제정책과 관련된 뉴스기사를 선별하였다. 불린 검색에 사용된 단어군은 [표 2-1]과 같이 경제 단어군 4개와 정책 단어군 45개로 구성되며, 언론사의 경우에는 이긍희 외(2020)보다 9개를 더 추가한 19개로 구성된다.[23] 즉, 수집 대상이 되는 뉴스기사는 경제 카테고리에 있는 19개의 언론사 기사 중 경제 단어군과 정책 단어군이 동시에 1개 이상 포함된 기사들이다.

[표 2-1] 검색 단어군

경제 단어군	정책 단어군	언론사
경제, 경기, 무역, 금융	정부, 청와대, 국무회의, 국회, 당국, 한은, 한국은행, 중앙은행, 기재부, 기획재정부, 재무부, 경제기획원, 통화, 개혁, 재경원, 재정경제원, 재경부, 재정경제부, 기획예산처, 기재부, 기획재정부, 금융위, 금융위원회, 금통위, 금통운위, 금융통화위원회, 금융통화운영위원회, 국제통화기금, IMF, WTO, 세계무역기구, 정책, 제정, 입법, 법안, 법률, 규정, 예산, 세금, 재정, 규제, 통제, 적자, 부족, 부채	경향신문, 국민일보, 내일신문, 동아일보, 문화일보, 서울신문, 세계일보, 조선일보, 중앙일보, 한국일보, 매일경제, 서울경제, 아주경제, 한국경제, 한겨레, 머니투데이, 아시아경제, 헤럴드경제, 파이낸셜뉴스

22) 불린 검색은 AND, OR, NOT과 같은 연산자를 활용하여 특정 검색어의 포함 여부를 나타내는 검색 방법이다. 예를 들어 서울의 부동산 관련 뉴스기사를 검색하기 위해 '서울'이란 단어는 반드시 포함하면서 'LTV'나 'DTI'와 같은 부동산 관련 단어도 하나 이상 포함하고 싶다면 검색어는 '(서울) AND (LTV OR DTI)'가 된다.

23) 본 연구는 뉴스기사의 빈도수로 불확실성을 측정하는 것이 아니므로 불확실성(U) 단어군은 고려하지 않는다.

단어군을 선정한 후에는 해당 단어군으로 검색된 뉴스기사의 제목을 수집한다. 검색된 기사 본문을 수집하는 것이 이상적이지만, 국내 뉴스 소비가 가장 많이 이루어지는 네이버(NAVER)나 다음(DAUM)과 같은 포털사이트는 검색어 길이를 50~75자로 제한되고 있어 본 연구와 같은 대규모 단어군을 사용할 수가 없다. 이를 해결하기 위해 본 연구에서는 빅카인즈에서 기사 제목을 수집한 후, 해당 기사 제목을 네이버에 검색하는 방식으로 뉴스기사를 수집하였다. 빅카인즈에서 기사 본문을 수집할 수도 있으나, 이 경우 본문 내용이 200글자 이내로 제한된다는 단점이 있으며, 소규모 웹사이트 특성상 크롤링과 같은 방식을 활용하면 서버 부하 방지를 위해 IP가 차단되는 문제가 발생할 수 있다.

빅카인즈에서 기사 제목을 수집한 후에는 중복 및 예외 기사들을 제거하는 작업을 수행한다. 빅카인즈에서도 중복 및 예외 기사를 제거하는 기능이 제공되고 있으나, 이를 이용하여도 여전히 중복 및 예외 기사들이 남아있기 때문에 본 연구에서는 이를 처리하기 위한 추가적인 전처리 작업을 병행하였다. 구체적으로 2005년부터 2022년까지 수집된 2,003,684건의 기사 중 빅카인즈에서 제공하는 중복 및 예외 기사[24] 92,078건을 제거하였으며, 이 외에도 제목만 다르고 본문 내용이 똑같거나 본문 내용을 조금만 수정하여 게재한 기사들 96,674건도 추가로 제거하였다. 또한, 우리나라의 뉴스 길이는 보통 300자 이상으로 작성되므로(정연구 외, 2016) 200자 미만인 기사는 광고성 기사일 가능성이 크며, 200자를 넘는 기사 중에서도 영어나 한자로 작성된 기사는 분석 대상으로 활용될 수가 없는 기사들이다. 따라서 뉴스 길이가 200글자 미만이거나 한글이 90글자 미만인 기사들 25,932건도 추가로 제거하였으며, 그 결과 빅카인즈에서 수집된 2,003,684건의 기사 중 1,799,000건의 기사가 유효표본으로 남았다.

중복 및 예외 기사들이 제거된 후에는 빅카인즈에서 수집한 기사 제목을 네이버에 검색하여 기사 본문을 수집한다. 이때 빅카인즈는 기사 원문의 특수문자를 일부 변형하므로 이를 네이버에 검색하기 위해서는 변형된 특수문자를 원래 형태로 복구해야 한다.[25] 또한, 불린 검색은 특수문자가 검색 연산자

24) 인사, 부고, 동정, 포토 등의 내용을 담은 기사를 말한다.

로 활용되므로 검색어에 특수문자가 포함되어 있으면 검색 결과가 왜곡될 수 있으며, 검색할 기사 제목이 짧은 경우에는 검색어와 무관한 기사들이 함께 수집되는 문제[26]도 발생한다. 이를 방지하기 위해 본 연구에서는 URL 인코딩[27]과 웹 스크래핑(web scraping)[28]을 활용하여 뉴스기사를 수집하였다. 즉, 빅카인즈에서 수집된 기사 제목을 URL 인코딩 형식으로 변경하여 네이버에 검색한 뒤, 검색 결과의 기사 제목과 빅카인즈의 기사 제목이 정확히 일치하는 경우에만 기사 본문이 수집되도록 코드를 설계하였다.

[표 2-2]는 네이버에서 수집한 뉴스기사를 연도별로 집계한 결과이다. 빅카인즈에서 수집한 1,799,000건의 기사 제목을 네이버에 검색하였을 때, 언론사에서 게재를 취소하거나 네이버에 게재되지 않은 640,101건의 기사는 검색되지 않았으며, 그 결과 전체 1,158,899건의 뉴스기사가 수집되었다. 또한 연도별 기사 건수는 2005년 7,416건에서 2022년 111,216건으로 크게 증가하였는데, 이는 인터넷 뉴스기사를 게재하는 언론사의 수가 증가한 것과 함께 오래된 뉴스기사는 언론사에서 게재를 취소하는 경우가 발생한 것으로 해석된다. 특히 2005년부터 2007년의 뉴스기사는 일 평균 20~22건밖에 되지 않아 해당 기간은 불확실성 지수를 작성하여도 유의미한 결과를 도출하기가 힘들 것으로 판단된다. 따라서 이후에 진행되는 EPU 지수의 작성 기간은 2008년부터로 한정한다.

25) 특히 기사 제목에 자주 사용되는 '…'과 같은 특수기호도 '...'와 같이 변형되기 때문에 빅카인즈에서 수집한 기사 제목을 그대로 사용하면 네이버에 검색되지 않는 경우가 종종 발생한다.

26) 예를 들어 네이버에 '경제성장률 1.5%'를 검색하면 '한국은행 경제성장률 1.5%로 발표'라는 제목의 기사도 함께 수집되는 문제가 발생한다.

27) HTTP 프로토콜을 통해 서버에 요청을 보내는 방식 중 하나로 GET 방식이 있다. GET 방식은 URL에 ASCII(아스키코드) 문자를 사용하여 서버에 요청을 보내는 방식으로 ASCII 이외에 다른 문자는 ASCII 형식으로 변환시키는 작업이 필요하다. URL 인코딩은 이러한 ASCII 문자를 %와 함께 16진수의 숫자로 대체시켜주는 방법이다. 예를 들어 URL에 포함될 수 없는 띄어쓰기는 %20과 같은 형태로 나타낸다.

28) 웹 스크래핑은 웹 크롤링(web crawling)과는 달리 웹 페이지 내에서 필요한 특정 정보를 추출하는 방식으로 HTML(Hypertext Markup Language)이나, CSS(Cascading Style Sheet) 등의 이해가 필요하다.

[표 2-2] 연도별 뉴스기사 수

연도	뉴스기사 수
2005	7,416
2006	7,847
2007	8,262
2008	25,694
2009	33,191
2010	33,787
2011	52,739
2012	51,383
2013	95,803
2014	87,431
2015	95,459
2016	89,830
2017	78,595
2018	82,170
2019	96,164
2020	105,396
2021	96,517
2022	111,216
합계	1,158,899

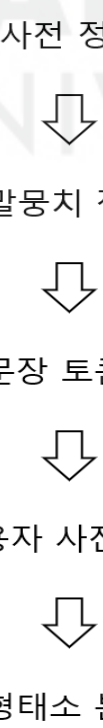
HANSUNG UNIVERSITY

2.2 텍스트 전처리(Text Preprocessing)

웹 스크래핑을 통해 수집한 텍스트 데이터는 우리가 오랜 세월 동안 일상적으로 사용해온 자연어로 이루어져 있다. 자연어는 프로그래밍 언어와 같이 인공적으로 만들어진 언어가 아니기 때문에 이를 바로 컴퓨터 알고리즘에 활용하기에는 많은 어려움이 따른다. 따라서 자연어를 다루기 위해서는 이를 분석에 활용할 수 있도록 정제하고 변환하는 텍스트 전처리 과정이 필요하며, 이러한 전처리 과정은 모형의 성능을 크게 좌우하기 때문에 자연어 처리 분야에서는 특히 더 중요하게 다루어지는 작업이라 할 수 있다.

텍스트 전처리 과정은 연구자나 연구 목적에 따라 다양하게 제시될 수 있으나, 본 연구에서는 다음 [그림 2-1]과 같이 다섯 단계로 구분하여 전처리 과정을 수행하였다. 본 절에서는 이에 대해 상세히 살펴보기로 한다.

[그림 2-1] 텍스트 전처리 과정



2.2.1 사전 정제

사전 정제는 이후에 수행되는 토큰화 작업이 원활하게 이루어지도록 분석에 방해가 되는 노이즈 데이터를 제거하는 작업이다. 본 연구에서는 이를 수행하기 위해 파이썬에서 지원하는 정규 표현식[29] 모듈 re를 활용하였다. 구체적으로는 한자, 이모티콘, 바이라인(by-line), URL 주소, 뉴스 게재 시간 등과 같이 분석에 불필요한 문자열을 제거하는 작업이 수행되었으며, 뉴스기사에서 사용되는 다양한 형태의 특수문자도[30] 정규 표현식을 이용해 표준화된 형태로 변환하였다. 또한 본 연구에서는 소괄호 안에 있는 문자열도 제거 대상에 포함하였다. 일반적으로 소괄호는 본문의 내용을 보충하거나 부연 설명하는 용도로 사용되지만, 경제 분야의 뉴스기사에서는 주로 특정 단어의 별칭, 자료출처, 주식 종목 코드 등과 같이 분석에 필수적이지 않거나 중복된 정보를 담고 있는 경우가 많아 오히려 노이즈로 작용할 가능성이 있다. 더욱이 이후의 텍스트 전처리 과정에서는 괄호와 같은 특수문자들이 대부분 제거되기 때문에[31] 데이터의 명확성을 확보하기 위해서는 사전 정제 시 소괄호 내부의 내용을 제거할 필요가 있다.[32]

사전 정제 과정을 거치고 난 후에는 중복 및 예외 기사들을 한 번 더 제거하였다. 그 결과, 수집된 1,158,899건의 기사 중 중복 기사 4,726건과 예외 기사 1,644건이 추가로 제거되었으며, 이에 따라 1,152,529건의 기사가 분석 대상으로 남았다.

---

29) 정규 표현식은 문자열에 나타나 있는 패턴을 파악하여 특정 조건의 문자열들을 검색하거나 치환하는 과정을 간편하게 처리하는 방법이다. 예를 들어 이메일 주소의 경우 영어 문자와 @, 그리고 인터넷 주소의 형태로 구성되어 있으므로 이를 정규 표현식으로 표현하면 다음과 같다.

'[a-zA-Z0-9₩._+]+@[A-Za-z]+.+(com|org|edu|net|co.kr)'

30) 뉴스기사는 특수문자가 다양한 방법으로 표현되는 경우가 많다. 예를 들어 ❝, ！, 【, 】와 같이 키보드 자판에 없는 특수문자들은 텍스트 전처리 과정에서 인식되지 못하는 경우가 종종 발생하므로 이를 적절히 변환해 줘야 한다.

31) 사실 특수문자는 텍스트 전처리 과정에서 대부분 제거되지만, 이후에 진행되는 문장 토큰화에서는 특수문자가 중요한 역할을 하므로 사전 정제 과정에서는 이를 제거하는 데 주의해야 한다.

32) 이 외에도 영어의 대소문자 통합, 반복되는 특수기호 변환, 불필요한 공백 제거 등의 다양한 전처리 작업이 사전 정제 과정에서 수행되었다.

2.2.2 말뭉치(Corpus) 정의

텍스트 분석은 연구 목적에 따라 문자, 단어, 문장 등 다양한 수준에서 분석이 이루어질 수 있다. 따라서 텍스트 데이터를 수집한 후에는 이를 분석 수준에 맞게 적합한 형태로 바꾸는 작업이 필요한데, 이를 위해서는 먼저 말뭉치를 정의해야 한다. 말뭉치는 보통 '문서들의 집합'으로 정의되며, 여기서 문서는 분석하고자 하는 데이터의 단위를 말한다. 예를 들어 감정 분류 모형으로 뉴스기사 문장의 긍·부정을 예측하고자 한다면 모형의 입력으로 사용되는 각 문장이 문서에 해당하고, 말뭉치는 이 문장들의 집합으로 정의된다. 즉, 말뭉치를 정의한다는 것은 결국 모형의 입력 구조를 결정하는 작업이라 할 수 있다.

본 연구에서는 뉴스기사로 감정 분류를 수행하는 것이 목적이다. 만약 분석 대상이 댓글 데이터라면 댓글 데이터는 작성자의 주관적인 감정이 비교적 명확하게 반영되어있어 작성자별로 감정 분류를 수행하는 것이 효과적일 것이다. 그러나 객관적으로 작성되는 뉴스기사는 긍정적인 문장과 부정적인 문장을 동시에 전달하는 경우가 많아 기사 한 편마다 감정을 분류하는 것은 그리 효과적이지 않다. 이를 고려해 본 연구에서는 의사 전달의 최소 단위인 문장을 문서로 하여 말뭉치를 정의하였다. 다시 말해 감정 분류 모형의 입력 데이터는 뉴스기사를 구성하는 각 문장이 된다.

2.2.3 문장 토큰화(Sentence Tokenization)

말뭉치를 정의한 후에는 텍스트 데이터의 구조를 변환하기 위해 토큰화 작업을 수행한다. 토큰화는 '토큰(token)'이라 불리는 유용한 의미적 단위로 텍스트 데이터를 분리하는 작업이다. 일반적으로 토큰은 문자, 단어, 문장 단위로 구분되는데, 특히 영어권에서는 단어 토큰으로 신경망 언어 모델을 학습시키는 경우가 많아 단어 토큰화(word tokenization)가 거의 필수적으로 수행된다. 그러나 단어 토큰화를 수행하더라도 본 연구와 같이 말뭉치가 문장으로 정의되었다면 텍스트 데이터를 문장 단위로 변환하는 작업이 선행되어야 하

며, 대부분의 텍스트 데이터가 문장 단위로 구분되어 있지 않다는 점을 생각하면 이를 처리하기 위한 문장 토큰화 역시 단어 토큰화만큼 매우 중요하게 다루어지는 작업이라 할 수 있다.

문장 토큰화는 문장이 시작하는 위치와 끝나는 위치를 인식하여 토큰화 작업이 수행된다. 한국어는 문법상 서술어와 종결 어미의 결합으로 문장이 종결되므로 문어체 문장의 경우에는 구두점(punctuation)과 종결 어미만으로도 문장 경계(sentence boundary)를 비교적 쉽게 인식할 수 있다. 그러나 구어체 문장은 서술어를 생략하거나 명사형 어미 또는 관형사형 어미로 문장을 마치기도 하며, 심지어는 새로운 어미를 만들어 문장을 마치기도 한다. 이로 인해 구어체 문장은 문장 경계를 인식하기가 상당히 어려운 편이며, 대부분의 한국어 문장 분리기들도 이러한 구어체 문장 경계를 인식하는 데 초점을 맞추고 있다. 그러나 구어체 문장에 최적화된 문장 분리기가 반드시 문어체 문장에서도 높은 성능을 나타낸다고 단정할 수는 없다. 게다가 같은 구어체라도 텍스트 데이터의 특성에 따라 문장 분리 성능이 다르게 나타나기 때문에[33] 분석 데이터에 적합한 문장 분리기를 선정하기 위해서는 해당 데이터의 유형과 특성을 고려하여 문장 분리기의 성능을 비교·평가하는 과정이 필요하다.

본 연구에서 분석하는 경제 분야의 뉴스기사는 정확한 정보 전달 및 전문적인 독자층의 요구를 충족하기 위해 공식적이고 정형화된 표현을 주로 사용한다. 이러한 특성으로 인해 문법적 정확성이 상대적으로 높은 편이며, 문장 분리 난이도 역시 상당히 쉬운 편에 속한다. 따라서 구어체 문장 분리에서 낮은 성능을 보이는 문장 분리기도 본 연구의 텍스트 데이터에서는 높은 성능을 보일 가능성이 크다. 게다가 본 연구에서는 EPU 지수는 속보성이 중요하므로 문장 분리기들이 모두 높은 성능을 보인다면 처리 속도가 주요 고려 사항이 된다. 이를 실증적으로 검토하기 위해 본 연구에서는 처리 속도, 정답률, Dice Coefficient를 기준으로 주요 문장 분리기들의 성능을 평가하였다. 평가에 사용된 테스트셋은 수집된 뉴스기사에서 추출한 약 2,500개의 문장 말뭉치이며,[34] 평가 대상이 되는 문장 분리기는 KSS[35]와 파이썬 자연어 처리 패

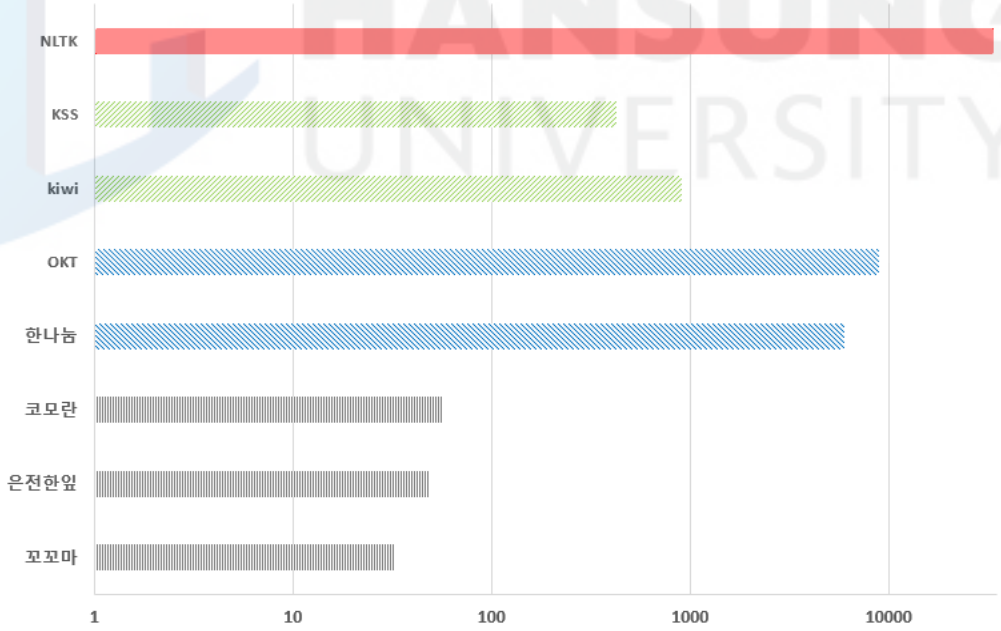
33) 예를 들어 트위터에서 수집한 텍스트 데이터는 OKT 문장 분리기의 성능이 높게 나타날 수도 있고, 뉴스 댓글 데이터는 KSS가 문장 분리기의 성능이 높게 나타날 수 있다.

34) 2013년부터 2022년까지 수집된 뉴스기사에서 월별로 1개의 뉴스기사를 무작위로 추출하였다.

키지인 NLTK, 그리고 문장 분리 기능을 제공하고 있는 형태소 분석기 6개(Kiwi, OKT, 한나눔, 코모란, 은전한잎, 꼬꼬마)이다.

먼저 [그림 2-2]는 문장 분리기의 초당 처리 가능한 문장 개수를 계산한 결과이다. 평가 대상이 되는 문장 분리기 중 KSS, Kiwi, 코모란, 은전한잎, 꼬꼬마는 형태소 분석과 문장 분리를 함께 수행하는 반면, NLTK, OKT, 한나눔은 별도의 형태소 분석 없이 자체적으로 문장 분리 기능을 수행한다. 이로 인해 평가 결과에서도 NLTK가 초당 33,605개의 문장을 분리하여 가장 빠른 처리 속도를 나타냈으며, OKT와 한나눔도 각각 8,896개와 5,981개의 문장을 분리하여 우수한 처리 속도를 나타냈다. 반면 형태소 분석과 문장 분리를 함께 수행하는 문장 분리기 중에서는 Kiwi가 903개로 가장 빨랐으며, KSS는 426개[36], 코모란, 은전한잎, 꼬꼬마는 각각 56개, 48개, 32개로 상당히 느린 처리 속도를 나타냈다.

[그림 2-2] 초당 처리 가능한 평균 문장 개수



35) 카카오 문장 분리기에 착안하여 개발한 패턴 기반의 한국어 문장 분리기이다.

36) KSS는 pecab, mecab, punct의 형태소 분석기를 사용할 수 있다. 이 중에서 처리 속도가 가장 우수하다고 알려진 mecab을 사용하여 문장 분리를 수행하였다.

문장 분리 성능은 정답률과 Dice Coefficient를 기준으로 평가하였다. 정답률은 전체 문장 중 정답 문장이 차지하는 비중으로 계산되는데, 이는 문장 분리 결과가 오답인 경우, 분리된 문장과 정답 문장 사이의 유사성을 평가하기 어렵다는 단점이 있다. 이를 보완하기 위해 본 연구에서는 Dice Coefficient를 보조 지표로 활용한다. Dice Coefficient는 아래 식 (2-1)과 같이 계산되며, 여기서 A는 정답 문장의 글자 집합을, B는 문장 분리기가 분리한 문장의 글자 집합을, $n(A \cap B)$는 두 집합에 공통으로 나타난 글자 수를 의미한다.

$$(2\text{-}1)\quad Dice(A,B) = \frac{2n(A \cap B)}{n(A)+n(B)}$$

[그림 2-3]은 Dice Coefficient의 계산 사례를 보여준다. 여기서 Case 1은 문장을 전혀 문장 분리를 하지 않은 경우이고, Case 2는 정답보다 과도하게 분리한 경우이다. 이를 살펴보면 Case 2가 Case 1보다 더 높은 점수를 받고 있는데, 이는 Dice Coefficient가 문장을 과도하게 분리할수록 더 높은 점수를 부여하는 경향이 있기 때문이다. 따라서 문장 분리 성능을 종합적으로 판단하기 위해서는 Dice Coefficient뿐만 아니라 정답률도 함께 고려할 필요가 있다.

[그림 2-3] Dice Coefficient 계산 사례

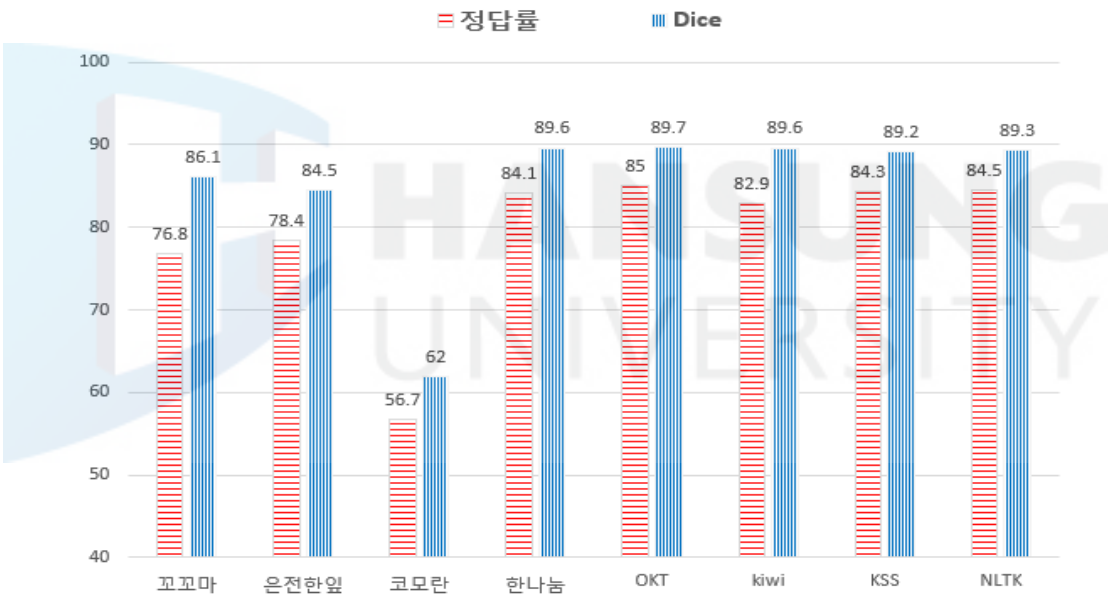
한 금융권 관계자는 "규제 완화가 이뤄졌어도 은행권이 여전히 우량 고객 유치에만 힘을 쏟고 있다"고 지적했다. 또 "은행권이 담보나 신용평가 뿐 아니라 기업의사업 전망이나 거래 신뢰도 등을 토대로 대출에 나서면 중금리대출상품이 자리를 잡지 못할 이유가 없다"고 말했다.

⇩

정답	Case 1		Case 2	
	예측	점수	예측	점수
한 금융권 관계자는 "규제 완화가 이뤄졌어도 은행권이 여전히 우량 고객 유치에만 힘을 쏟고 있다"고 지적했다.	-	(2*0)/(47+0) = 0	한 금융권 관계자는 "규제 완화가 이뤄졌어도 은행권이 여전히 우량 고객 유치에만 힘을 쏟고 있다"	(2*41)/(47+41) = 0.93
또 "은행권이 담보나 신용평가 뿐 아니라 기업의사업 전망이나 거래 신뢰도 등을 토대로 대출에 나서면 중금리대출상품이 자리를 잡지 못할 이유가 없다"고 말했다.	한 금융권 관계자는 "규제 완화가 이뤄졌어도 은행권이 여전히 우량 고객 유치에만 힘을 쏟고 있다"고 지적했다. 또 "은행권이 담보나 신용평가 뿐 아니라 기업의사업 전망이나 거래 신뢰도 등을 토대로 대출에 나서면 중금리대출상품이 자리를 잡지 못할 이유가 없다"고 말했다.	(2*68)/(68+115) = 0.74	고 지적했다. 또 "은행권이 담보나 신용평가 뿐 아니라 기업의사업 전망이나 거래 신뢰도 등을 토대로 대출에 나서면 중금리대출상품이 자리를 잡지 못할 이유가 없다"	(2*63)/(68+69) = 0.92
-	-	-	고 말했다.	-
평균	0.37		0.925	

정답률과 Dice Coefficient의 계산 결과는 [그림 2-4]에 제시되어있다. 평가 결과, 한나눔, OKT, Kiwi, KSS, NLTK는 정답률과 Dice 점수 모두에서 우수한 성능을 나타냈으며, 이들 중 Kiwi는 상대적으로 문장을 과도하게 분리하는 것으로 나타났다. 꼬꼬마와 은전한잎은 중간 수준의 성능을 기록하였고, 코모란은 정답률과 Dice 점수 모두에서 현저히 낮은 성능을 나타냈다. 특히, 꼬꼬마는 모든 문장 분리기 중에서도 문장을 가장 과도하게 분리하는 것으로 분석되었다. 따라서 본 연구에서 수집한 경제정책 뉴스기사에 대해서는 한나눔, OKT, KSS, NLTK가 가장 적합한 문장 분리기로 판단된다.[37)]

[그림 2-4] 문장 분리 정확도



이상의 평가 결과를 종합적으로 고려하여 본 연구에서는 문장 분리 정확도와 처리 속도 모두에서 우수한 성능을 보인 NLTK를 문장 분리기로 채택하였다. 그러나 NLTK는 문장이 계층적 구조를 가지는 경우, 이를 분절할 수

37) 이처럼 문장 분리 성능이 우수하다고 평가받는 KSS가 NLTK와 유사한 성능을 보인다는 점이 다소 의외일 수도 있으나, 이는 뉴스 데이터에 국한된 결과임에 유의해야 한다. 특히, NLTK와 OKT는 구두점 뒤 띄어쓰기 여부에 따라 문장을 분리하는 경향이 있어, 구어체로 작성된 텍스트 데이터에서는 낮은 성능을 보일 가능성이 크다. 그러나 본 연구와 같이 문어체로 작성되는 뉴스기사에 대해서는 NLTK 역시 우수한 성능을 보이므로 굳이 처리 속도가 느린 문장 분리기를 사용할 필요는 없다고 판단된다.

있는 기능이 제공되지 않는다. 특히 경제정책 뉴스기사는 인용구가 자주 활용되는 특성이 있어 이를 분리하지 않을 경우, 문장이 과도하게 길어지고 상반된 감정 표현이 하나의 문장 내에 혼재되는 문제[38]가 발생할 수 있다. 이를 보완하기 위해 본 연구에서는 NLTK로 문장 분리를 수행한 이후, 인용문을 추출하는 전처리 작업을 추가로 수행하였다. 아울러 인용문 처리 이후에는 광고 문구와 같은 분석에 불필요한 문장들을 제거하였으며, 그 결과 1,152,529건의 뉴스기사는 약 1,500만 개의 문장으로 분리되었다.

2.2.4 사용자 사전 구축

다음 항에서 수행되는 형태소 분석은 일반적으로 사전에 기반하여 수행되기 때문에 사전에 등록되지 않은 단어, 예컨대 신조어, 복합어, 축약어와 같은 단어는 하나의 단어로 인식하지 못하고 분절되어 처리되는 단점이 있다. 게다가 금융, 법률, 의학과 같은 특수 도메인에서 사용되는 전문 용어 역시 형태소 분석기에 등록되지 않은 경우가 많아, 본 연구와 같은 경제 분야의 텍스트 데이터를 대상으로 형태소 분석을 수행하면, 문장 내 핵심 정보가 손실되는 상황이 발생할 수 있다.[39] 본 항에서는 이러한 문제를 보완하기 위해 형태소 분석기에 미등록된 단어, 특히 경제정책 뉴스에 등장하는 전문 용어들을 중심으로 형태소 분석기용 단어사전을 구축하는 작업을 수행한다.

형태소 분석기의 사용자 사전을 구축하는 작업은 보통 형태소 분석을 수행한 뒤 잘못 처리된 단어를 고르는 방식으로 처리된다. 그러나 이러한 방식은 노동집약적인 수작업으로 이루어지기 때문에 시간적 비용이 많이 소모될 뿐만 아니라 전문 지식을 가진 인력도 필요하다. 본 연구에서는 이러한 수작업을 줄이기 위해 비지도 학습 기반 단어 추출기인 soynlp를 활용하여 사용자 사전을 구축하였다.

---

38) 예를 들어 〈A씨는 "가게 음식이 너무 맛있어서 또 오고 싶다"면서도 "가게 음식이 너무 비싸서 또 오긴 힘들 것 같다"고 말했다.〉라는 문장은 긍정과 부정의 문장이 혼재되어 있으므로 이를 적절히 분리해줘야 하는데 NLTK는 이를 하나의 문장으로 처리해 버린다.

39) 예를 들어 '개발제한구역'과 같은 단어는 '개발', '제한', '구역'으로 분리된 결과를 보여주기 때문에 본래의 '개발제한구역'이란 단어의 정보가 사라져버리는 문제가 발생한다.

soynlp는 서로 다른 알고리즘을 사용하는 두 가지 종류의 단어 추출기 모듈을 제공한다. 하나는 품사에 상관없이 의미를 지닌 모든 단어를 추출하는 단어 추출기(WordExtractor)이고, 다른 하나는 명사만을 대상으로 추출하는 명사 추출기(NounExtractor)이다. 본 연구에서는 이 중 명사 추출기를 활용하여 사용자 사전을 구축하였다. 명사 추출기를 활용한 이유는 명사가 한국어 어절에서 가장 빈번하게 사용되는 품사이면서도, 형태소 분석기의 미등록 단어 문제가 특히 심하게 나타나기 때문이다. 더욱이 용언(동사, 형용사)은 체언(명사, 대명사, 수사)보다 신조어가 쉽게 만들어지지 않으며, 만들어지더라도 경제정책 뉴스기사의 공식적이고 정형화된 문체에서는 사용될 가능성이 크지 않다고 판단하였다.

soynlp의 명사 추출기는 주어진 문장 말뭉치에서 어절들의 구조를 학습하여 명사를 추출한다.[40] 본 연구에서는 앞서 문장 토큰화가 완료된 약 1,500만 개의 문장을 대상으로 학습을 수행하였으며, 그 결과 약 200만 개의 명사가 도출되었다. 이 중 등장 빈도가 10회 이상이고, 3음절 이상으로 구성된 명사 251,156개를 1차로 선별하였으며, 이 가운데 형태소 분석기에 등록된 125,432개의 명사는 제외하였다. 이후 남은 125,724개 명사에 대해 최종 검수 작업을 거쳐, 일반명사 71,126개와 고유명사 16,787개로 구성된 사용자 사전을 구축하였다.

2.2.5 형태소 분석(Morpheme Analysis)

앞서 단어 토큰화는 영어권 NLP에서 거의 필수적으로 수행되는 전처리 단계임을 언급한 바 있다. 그러나 한국어는 영어와 달리 단어 토큰화를 수행하기가 상당히 까다로운 편인데, 이는 크게 두 가지 이유에서 비롯된다. 첫

40) soynlp는 한글 어절의 구조적 특성을 이용하여 단어를 추출한다. 여기서 한글 어절의 구조적 특성이란, 어절의 왼쪽(L part)은 의미를 지니는 단어(명사, 동사, 형용사 등)가 위치하고, 어절의 오른쪽(R part)은 문법적 기능을 하는 단어(조사, 어미 등)가 위치한다는 것을 말한다. soynlp의 단어 추출기는 이러한 한글 어절의 경계(단어의 경계)를 학습하여 단어를 추출하며, 이를 위해 Accessor Variety(Feng et al., 2004), Branching Entropy(Jin and Tanaka-Ishii, 2006), Cohesion score라는 3가지 알고리즘을 활용한다. 또한, soynlp의 명사 추출기는 한글 어절의 오른쪽, 즉 명사 우측에 등장하는 글자 분포를 활용하여 명사를 추출한다. 이에 대한 보다 자세한 설명은 soynlp 깃허브에서 확인할 수 있다.

번째는 한국어가 영어에 비해 띄어쓰기 규칙이 복잡하고, 이를 일관되게 적용하지 않는 경우가 많기 때문이다. 즉, 한국어는 띄어쓰기가 잘 지켜지지 않는 경우가 많아, 영어와 같이 띄어쓰기만으로 토큰화를 수행하여도 단어의 경계(word boundary)가 명확히 드러나지 않는다.[41] 두 번째는 한국어가 교착어인 것에서 기인한다. 영어와 같은 굴절어는 단어 자체가 독립적으로 구분되고, 문법적 역할에 따라 단어의 형태가 바뀐다. 따라서 영어는 대부분이 단어 단위로 띄어쓰기가 이루어져 합성어나 축약어와 같은 예외 처리만 적용한다면 띄어쓰기 토큰화와 단어 토큰화가 거의 같아지게 된다. 그러나 교착어인 한국어는 낱말에 첨가되는 조사, 접사 등에 의해 문법적 기능이 달라지며, 같은 단어임에도 어떤 조사가 교착되느냐에 따라 서로 다른 단어로 인식된다. 이로 인해 한국어 NLP에서는 어절 토큰화가 거의 지양되고 있으며, 유의미한 단어를 정확히 인식하기 위해서는 조사와 같은 단어들을 모두 분리해주는 과정도 필요하다.

이러한 단어 토큰화의 어려움으로 인해 한국어 NLP에서는 단어 토큰화 대신 형태소 분석을 통해 토큰화 작업이 수행된다. 형태소 분석은 뜻을 가진 말의 최소 단위인 형태소(morpheme)를 발굴하여 인식하는 과정을 말한다. 즉, 단어를 더 세부적으로 분해하여 단어에 붙은 조사나 어미 등도 토큰으로 인식하는 과정이다. 또한 형태소 분석에서는 토큰화 과정 외에도 각 형태소의 품사 정보를 부착하는 품사 표식 과정도 수행된다. 예를 들어 '나는 밥을 먹는다'라는 문장은 '나(대명사)/는(보조사)/밥(일반명사)/을(목적격 조사)/먹(동사)/는(선어말 어미)/다(종결 어미)'와 같은 형태로 품사 정보가 부착되며, 이를 이용하면 필요한 품사 정보를 필터링하는 것이 가능해진다. 게다가 품사 표식은 텍스트 데이터의 중의성과 모호성을 해결하기 위해 수행되는 주요 작업이기도 하다. 예컨대, '못'이라는 단어는 명사와 부사로 쓰이느냐에 따라 그 의미가 달라지므로 해당 단어의 의미를 정확히 인식하기 위해서는 품사 표식이 중요한 역할을 하게 된다.

41) 한국어의 띄어쓰기가 잘 지켜지지 않는 이유는 여러 견해가 있으나, 가장 기본적인 견해는 한국어는 띄어쓰기가 지켜지지 않아도 글을 쉽게 이해할 수 있다는 것이다. 예를 들어 영어는 "hellopython"과 같이 띄어쓰기가 지켜지지 않으면 그 의미를 파악하기 힘들지만, 한국어는 "안녕파이썬"과 같이 띄어쓰기가 없어도 이를 쉽게 이해할 수 있다.

형태소 분석 역시 전처리 도구에 따라 분석 결과가 상이하게 나타나므로 연구에 적합한 도구를 선정하는 과정이 필요하다. 이를 위해 본 연구에서는 5가지 형태소 분석기(한나눔, 꼬꼬마, 코모란, 은전한잎, Okt)[42]를 사용할 수 있는 KoNLPy(박은정·조성준, 2014)와 최근 오픈 소스로 개발된 Kiwi[43]를 대상으로 성능 평가를 수행하였다. 평가는 품사 목록 비교, 모호성 평가, 처리 속도 비교, 사용자 사전 활용 가능성 검토를 기준으로 수행하였으며, 모호성 평가의 경우에는 바른[44]에서 구축한 모호성 평가 데이터셋[45]을 활용하였다.

먼저 분석 가능한 품사 종류를 확인하기 위해 각 형태소 분석기의 품사 목록을 [표 2-3]에 정리하였다. 이를 살펴보면 Kiwi, 은전한잎, 코모란은 거의 비슷한 품사 목록을 갖고 있으며, 꼬꼬마는 이들보다 용언, 어미 등에서 좀 더 세분된 품사 목록을 갖추고 있음을 확인할 수 있다. 또한, 한나눔은 형태소 분석기 중 가장 많은 품사 목록을 제공하고 있는데, 이처럼 분석 가능한 품사 종류가 많다고 해서 반드시 형태소 분석기의 성능이 높게 나타나는 것은 아니다. 특히, 품사 체계가 지나치게 세분되면 단어의 본래 의미가 상실되는 상황이 발생할 수 있으며,[46] 상세한 품사 목록이 불필요한 경우에는 오히려 품사 목록이 가장 적은 OKT가 높은 성능을 발휘할 수도 있다. 다만, 본 연구의 경우에는 고유명사와 대명사의 사용 비중이 높아 이를 모두 일반명사로 처리하는 OKT의 품사 체계는 본 연구에 적합하지 않을 가능성도 존재한다. 따라서 품사 목록만을 기준으로 판단하였을 때는 꼬꼬마, Kiwi, 은전한잎, 코모란이 본 연구에 더 적합한 형태소 분석기로 평가된다.

---

42) 한나눔, 꼬꼬마, 코모란은 Kaist Semantic Web Research Center, 서울대학교 IDS(Intelligent Data Systems) 연구실, Shineware에서 개발된 형태소 분석기이며, 은전한잎은 일본어용 형태소 분석기를 한국어에 맞게 수정한 버전이다. 또한, OKT는 기존 Twiiter 형태소 분석기를 의미한다(0.5.0 버전부터 이름이 변경되었다.).

43) https://github.com/bab2min/kiwi

44) 2023년에 바이칼에이아이와 한국언론진흥재단에서 공동으로 구축한 형태소 분석기이다.

45) 국내에서 구축된 모호성 평가 데이터셋 중 가장 큰 규모에 해당하며, 전체 35,396개의 문장으로 구성되어 있다.

46) 예를 들어 '허위자료'란 단어를 '허(감탄사)/위자료(고유명사)'와 같이 분절시키는 상황이 발생할 수 있다.

[표 2-3] 형태소 분석기 품사 목록 비교

한나눔 (69개)	꼬꼬마 (56개)	Kiwi (52개)	은전한잎 (43개)	코모란 (42개)	OKT (19개)
동작성 명사	일반명사	일반명사	일반명사	일반명사	명사
상태성 명사					
비서술성 명사					
비서술성 직위명사					
성	고유명사	고유명사	고유명사	고유명사	
이름					
성+이름					
기타-일반					
비단위성 의존명사	일반 의존명사	의존명사	일반 의존명사	의존명사	
비단위성 의존명사 (--하다 붙는 것)					
단위성 의존명사	단위 의존명사		단위 의존명사		
양수사	수사	수사	수사	수사	
서수사					
인칭 대명사	대명사	대명사	대명사	대명사	
지시 대명사					
지시 동사	동사	동사	동사	동사	동사
일반 동사					
지시 형용사	형용사	형용사	형용사	형용사	형용사
성상 형용사					
보조용언	보조 동사	보조용언	보조용언	보조용언	-
	보조 형용사				-
-	긍정 지정사	긍정 지정사	긍정 지정사	긍정 지정사	-

–	부정 지정사	부정 지정사	부정 지정사	부정 지정사	–
–	수 관형사	관형사	관형사	관형사	관형사
지시 관형사	일반 관형사				
성상 관형사					
일반 부사	일반 부사	일반 부사	일반 부사	일반 부사	일반 부사
접속 부사	접속 부사	접속 부사	접속 부사	접속 부사	접속사
지시 부사	–	–	–	–	–
감탄사	감탄사	감탄사	감탄사	감탄사	감탄사
주격 조사	주격 조사	주격 조사	주격 조사	주격 조사	조사
보격 조사	보격 조사	보격 조사	보격 조사	보격 조사	
관형격 조사	관형격 조사	관형격 조사	관형격 조사	관형격 조사	
목적격 조사	목적격 조사	목적격 조사	목적격 조사	목적격 조사	
부사격 조사	부사격 조사	부사격 조사	부사격 조사	부사격 조사	
호격 조사	호격 조사	호격 조사	호격 조사	호격 조사	
인용격 조사	인용격 조사	인용격 조사	인용격 조사	인용격 조사	
접속격 조사	접속 조사	접속 조사	접속 조사	접속 조사	
공동격 조사	–	–	–	–	
통용 보조사	보조사	보조사	보조사	보조사	
종결 보조사					
서술격 조사	–	–	–	–	
선어말 어미	존칭 선어말 어미	선어말 어미	선어말 어미	선어말 어미	선어말 어미
	시제 선어말 어미				
	공손 선어말 어미				
종결 어미	평서형 종결 어미	종결 어미	종결 어미	종결 어미	어미
	의문형 종결 어미				
	명령형 종결 어미				
	청유형 종결 어미				

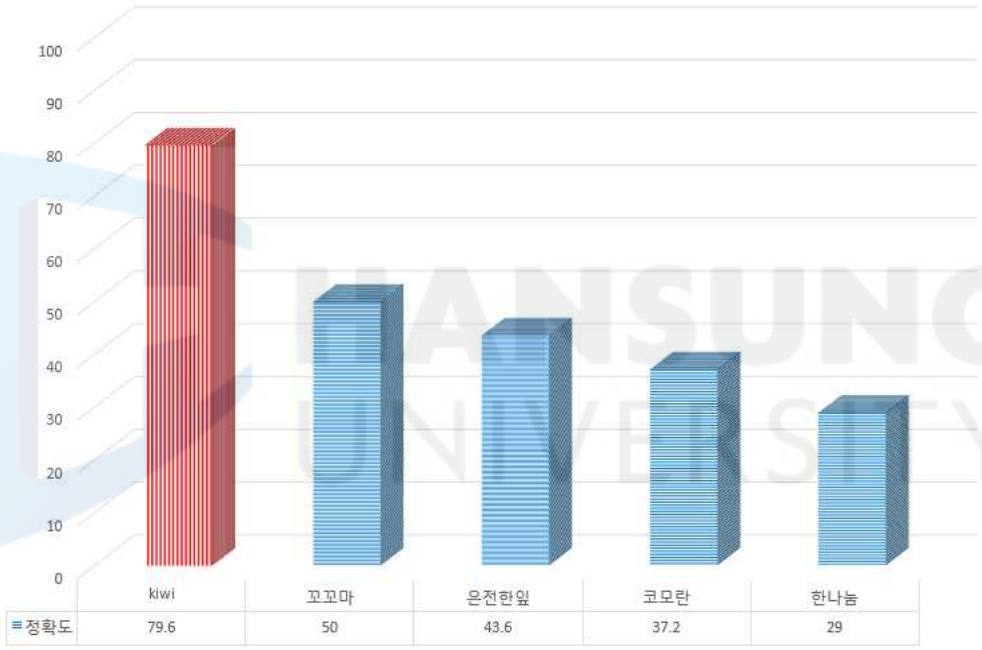
	감탄형 종결 어미				
	존칭형 종결 어미				
대등적 연결 어미	대등적 연결 어미	연결 어미	연결 어미	연결 어미	
보조적 연결 어미	보조적 연결 어미				
종속적 연결 어미	종속적 연결 어미				
명사형 전성 어미	명사형 전성 어미	명사형 전성 어미	명사형 전성 어미	명사형 전성 어미	
관형형 전성 어미	관형형 전성 어미	관형형 전성 어미	관형형 전성 어미	관형형 전성 어미	
접두사	체언 접두사	체언 접두사	체언 접두사	체언 접두사	-
	용언 접두사	-	-	-	-
단위 뒤	명사 파생 접미사	명사 파생 접미사	명사 파생 접미사	명사 파생 접미사	접미사
일반명사 뒤					
일반명사 뒤					
동작성 뒤					
상태성 뒤					
인명 1,3 뒤					
모든 명사 뒤					
동사 뒤	동사 파생 접미사	동사 파생 접미사	동사 파생 접미사	동사 파생 접미사	-
일반명사 뒤					-
동작명사 뒤					-
상태명사 뒤	형용사 파생 접미사	형용사 파생 접미사	형용사 파생 접미사	형용사 파생 접미사	-
일반명사 뒤					-
형용사 뒤	-	부사파생접미사	-	-	-
상태명사 뒤	-		-	-	-
-	어근	어근	어근	어근	-
마침표	종결 부호(. ! ?)	종결 부호(. ! ?)	종결 부호(. ! ?)	종결 부호(. ! ?)	구두점
줄임표	줄임표	줄임표	줄임표	줄임표	
여는 따옴표, 묶음표	괄호, 따옴표, 줄표	인용 부호 및 괄호	여는 괄호	괄호, 따옴표, 줄표	

닫는 따옴표, 묶음표			닫는 괄호		
쉼표	구분 부호(,·/:;)	구분 부호(,·/:;)	구분 부호(,·/:;)	구분 부호(,·/:;)	
기타 기호	붙임표(- ~)	붙임표(- ~)	기타 기호	붙임표(- ~)	
단위 기호	기타 기호 (논리 수학 기호, 화폐 기호)	기타 기호 (논리 수학 기호, 화폐 기호)		기타 기호 (논리 수학 기호, 화폐 기호)	외국어, 한자, 기타 기호
이음표					
–	한자	한자	한자	한자	
외국어	외국어	알파벳	외국어	외국어	알파벳
–	숫자	숫자	숫자	숫자	숫자
–	명사추정범주	분석 불능	–	명사추정범주	미등록어
–	–		–	용언추정범주	
–	–		–	분석불능범주	
–	–	해시태그(#abcd)	–	–	해쉬태그
–	–	멘션(@abcd)	–	–	멘션(@abcd)
–	–	이메일 주소	–	–	이메일 주소
–	–	URL 주소	–	–	URL 주소
–	–	일련번호 (전화번호, 통장번호, IP주소 등)	–	–	(ex: ㅋㅋ)
–	–	덧붙은 받침	–	–	–

자료: 1) https://konlpy.org/ko/latest/morph/#id2
2) https://github.com/bab2min/kiwi

다음으로 [그림 2-5]는 각 형태소 분석기가 중의성 및 모호성 문제를 얼마나 잘 해소하는지를 평가한 결과이다. 여기서 정확도는 평가 문장 대비 정답으로 분석한 문장의 비율을 의미한다.[47] 평가 결과에서는 n-gram 기반의 Kneser-ney 언어 모형을 사용하는 Kiwi가 HMM이나 CRF 등의 확률 모형을 사용하는 다른 형태소 분석기보다 더 높은 정확도를 보였으며, 품사 목록이 가장 많았던 한나눔의 경우에는 오히려 가장 낮은 성능을 나타냈다.

[그림 2-5] 형태소 분석기 모호성 평가[1)]



주: 1) OKT는 품사 표식 목록이 너무 적어 해당 평가 데이터로는 정확도를 측정하지 못하였으나, 평가 데이터[48]를 따로 구축하여 평가하였을 때 약 44.1%의 정확도를 나타냄

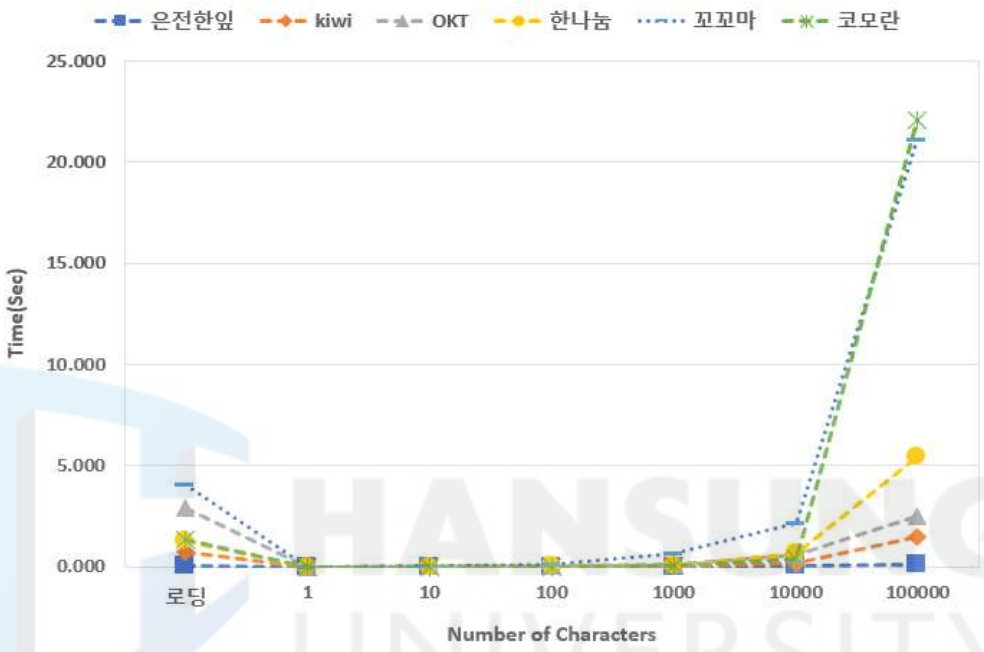
[그림 2-6]은 약 10만 개의 문자를 대상으로 각 형태소 분석기들의 처리 속도를 비교한 결과이다. 10만 개의 문자를 처리하였을 때 은전한잎은 0.13초, Kiwi는 1.505초, OKT는 2.476초, 한나눔은 5.405초, 꼬꼬마는 21.078초, 코모란은 22.067초의 시간이 소요되었다. 즉, 처리 속도 측면에서는 은전한잎

47) 예를 들어 '하늘을 나는 자동차'라는 문장에서 '나는'을 '나(명사)+는(조사)'로 분석하면 오답으로 분리하고, '날(동사)+는(어미)'로 분석하면 정답으로 분리한다.

48) 규칙 및 불규칙 활용 동사를 구분하는 문장, 형태가 동일한 동사, 형용사, 명사를 구분하는 문장 등으로 구성되어 있다.

이 가장 뛰어난 성능을 보였으며, Kiwi와 OKT도 꽤 우수한 성능을 보이는 것으로 나타났다.

[그림 2-6] 형태소 분석기 처리 속도 비교



본 연구에서는 사용자 사전이 매우 중요한 역할을 하므로 형태소 분석기별로 사용자 사전의 활용 가능성을 검토할 필요가 있다. 이를 위해 본 연구에서는 미등록된 단어를 사용자 사전에 등록한 후, 해당 단어에 대한 품사 표식이 정확히 수행되었는지를 확인하는 방식으로 실험을 설계하였다. 실험 대상은 모호성 평가에서 우수한 성능을 보였던 Kiwi, 꼬꼬마, 은전한잎으로 한정하였으며, 그 결과는 아래 [표 2-4]에 제시하였다. 실험 결과, 은전한잎은 세 가지 단어 모두에서 올바른 품사 표식 결과를 나타낸 반면, Kiwi와 꼬꼬마는 '국민의 힘'이란 단어를 올바르게 인식하지 못하는 것으로 나타났다. 이는 두 분석기의 사용자 사전이 공백을 포함하는 다어절 단어를 지원하지 않기 때문이다. 또한, 꼬꼬마의 경우에는 '개발제한구역'과 같은 단일 어절조차도 올바르게 인식하지 못하였는데, 이는 사용자 사전에 있는 기존 단어가 새로 등록된 단어보다 높은 우선순위를 가지고 있기 때문이다. 즉, 기존 단어가 새로

등록된 단어보다 더 낮은 단어비용을 가져 기존 단어를 더 우선으로 반영한 것이다. 반면, 은전한잎과 Kiwi는 이러한 단어비용 값을 조정할 수 있는 기능이 제공되어 '개발제한구역' 단어에 대해서도 올바른 품사 표식 결과를 얻을 수 있었다.

[표 2-4] 형태소 분석기 사용자 사전 비교

단어	꼬꼬마	Kiwi	은전한잎
이아무개	이아무개 (고유명사)	이아무개 (고유명사)	이아무개 (고유명사)
개발제한구역	개발(일반명사)+ 제한(일반명사)+ 구역(일반명사)	개발제한구역 (일반명사)	개발제한구역 (일반명사)
국민의 힘	국민(일반명사)+ 의(관형격 조사)+ 힘(일반명사)	국민(일반명사)+ 의(관형격 조사)+ 힘(일반명사)	국민의 힘 (일반명사)

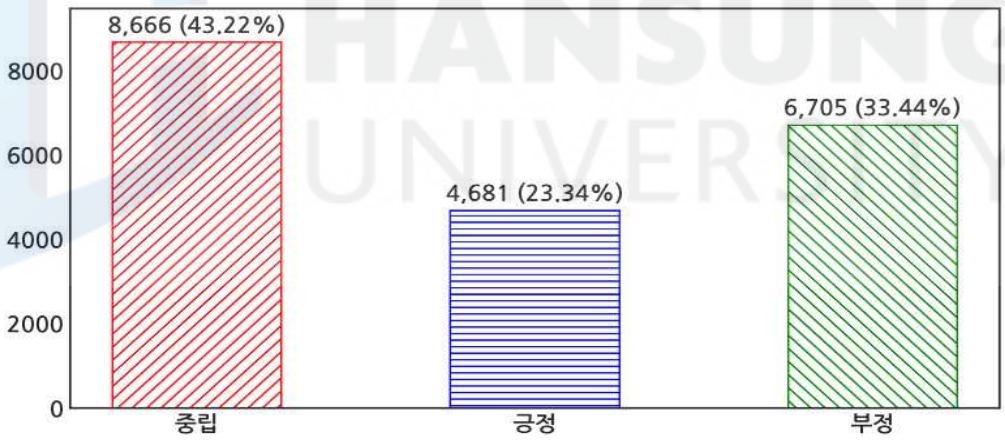
이상의 평가 결과를 종합하면, 처리 속도 및 사용자 사전 활용 가능성 측면에서는 은전한잎이, 분석 품질 측면에서는 Kiwi의 성능이 우수한 것으로 요약할 수 있다. 본 연구에서는 은전한잎과 Kiwi의 처리 속도 차이가 크지 않은 것을 고려하여 Kiwi를 본 연구의 최종 형태소 분석기로 선정하였다. 다만, Kiwi는 다어절 단어에 대한 사용자 사전 적용에 한계가 있으므로 이를 보완하기 위해 띄어쓰기 정규화 과정을 추가로 수행하였다.[49]

49) 형태소 분석기는 '개발제한구역', '개발 제한 구역'과 같이 띄어쓰기가 다른 단어를 서로 다른 단어로 인식한다. 이에 본 연구에서는 사용자 사전에 등록된 단어들의 표기 형식을 기준으로 텍스트 내 해당 단어들의 띄어쓰기를 통일하는 정규화 작업을 수행하였다.

2.3 학습 데이터셋

학습 데이터셋은 2절에서 구축한 문장 데이터 중 약 2만개를 추출하여 80%를 학습 데이터로, 나머지 20%를 검증 데이터로 구성하였다. 데이터셋이 부족한 점을 고려해 테스트 데이터는 구축하지 않는 대신 계층적 k-fold CV[50]를 모형 평가에 활용하여 모형의 안정성과 신뢰성을 높였다. 또한 해당 데이터셋은 저자가 직접 감정 레이블을 작성하여 다음 [그림 2-7]과 같이 23%가 긍정, 33%가 부정, 44%가 중립으로 분류되었다. 그러나 이처럼 레이블이 불균형한 경우 다수 레이블만 과도하게 학습하고 소수 레이블은 잘 분류하지 못하는 문제가 발생할 수 있으므로 모형 학습 시에는 레이블 가중치[51]를 반영하여 불균형 데이터를 조정하였다.

[그림 2-7] 레이블 분포

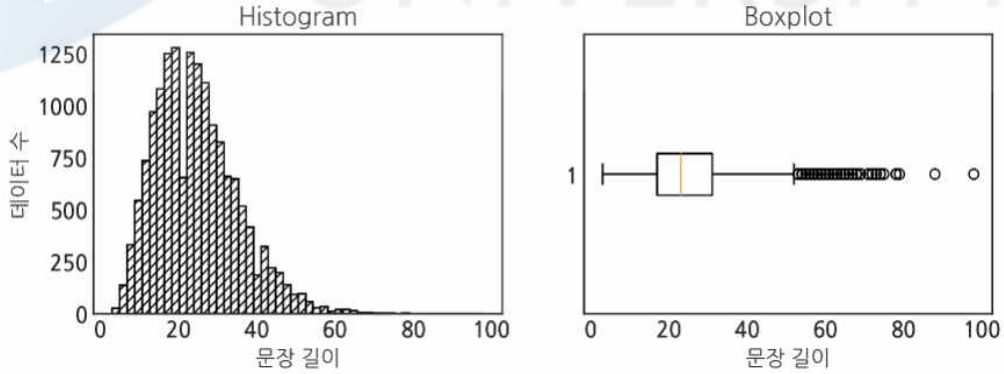


50) k-fold 교차 검증은 모형의 일반화 능력을 평가하기 위해 사용되는 방법으로 데이터를 k개의 fold로 나누어 k-1개를 학습 데이터로, 나머지 1개를 검증 데이터로 사용해 모델을 k번 학습하고 평가를 수행하는 방법이다. 이는 모델을 학습할 때마다 다른 검증 데이터를 사용하기 때문에 모델의 성능을 정확히 평가할 수 있으며, 특히 본 연구처럼 학습 데이터가 적은 상황에서는 더욱 유용한 평가 방법이 된다. 또한 계층적 k-fold 교차 검증은 fold를 나눌 때 fold 간의 레이블 비율을 전체 데이터셋의 레이블 비율과 일치하게 나누는 방법을 말한다.

51) 소수 레이블에 더 높은 가중치를 부여하여 모델이 소수 레이블을 더 잘 인식하고 예측하게 도와주는 기법이다. 이때 가중치는 다양한 방법으로 산출될 수 있는데, 본 연구에서는 모든 레이블이 모델 학습에 동일하게 기여하도록 균형 가중치(Balanced Weighting) 방식을 사용하였다. 이는 (전체 데이터 수) / (해당 클래스의 데이터 수 × 레이블 수)로 산출된다.

한편 학습 데이터를 모형의 입력으로 사용하기 위해서는 정수 인코딩과 패딩(padding) 과정을 거쳐야 한다. 정수 인코딩은 단어를 빈도순으로 정렬하여 빈도수에 따라 각 단어에 고유한 정수를 할당하는 과정이다. 즉, 모형의 입력값인 임베딩[52] 벡터로 변환하기 위해 텍스트 데이터를 컴퓨터가 해석할 수 있는 숫자 데이터로 변환해 주는 작업이다. 또한 학습 데이터를 모형의 입력으로 사용하기 위해서는 입력 데이터의 크기(shape)가 일정할 필요가 있는데, 패딩은 이를 위해 모든 문장의 길이를 같은 길이로 맞춰주는 작업이다. 패딩 길이는 가장 긴 문장의 길이로 설정할 수도 있지만, 매우 긴 문장이 몇 개만 존재하는 경우 다수의 짧은 문장에 과도한 패딩이 추가되므로 메모리 낭비와 계산 효율성이 저하될 수 있다. 따라서 패딩 길이는 연구자 주관에 따라 모형 학습에 불필요한 계산 부담을 최소화하면서 입력 데이터의 정보를 최대한 보존할 수 있는 길이로 설정하게 된다. 본 연구에서는 학습 데이터셋이 다음 [그림 2-8]과 같이 최대 길이 98, 평균 길이 25로 약 99.5%의 데이터가 60자 이내에 들어가 있다는 것을 고려해 60으로 패딩 길이를 설정하였다.

[그림 2-8] 문장 길이 분포



52) 임베딩에 대한 내용은 다음 장에서 자세히 다루고 있다.

Ⅲ. 연구 모형

3.1 단어 임베딩 모형(Word Embedding Model)

인간이 사용하는 자연어를 컴퓨터가 처리하기 위해서는 이를 컴퓨터가 이해할 수 있는 형식으로 변환하는 과정이 필요하다. 즉, 인간이 사용하는 자연어를 기계가 이해할 수 있도록 표현(representation)하는 단계를 거쳐야 한다. 임베딩은 이를 처리하기 위한 단계로 자연어를 벡터 공간에 맵핑 시킨 결과, 혹은 그 일련의 과정 전체를 의미한다(이기창, 2019).

임베딩이 중요한 이유는 자연어의 표현뿐만 아니라 모형의 효율성과 정확성을 향상하는 데 도움을 주기 때문이다. 임베딩은 다양한 유형의 고차원 데이터를 저차원의 밀집 벡터(dense vector)로 변환하여 모형의 계산 복잡성을 줄이고 학습 속도를 향상하는 데 도움을 준다. 또한 임베딩 과정을 거치면 비슷한 의미의 단어들을 유사한 벡터로 표현하는 것이 가능해진다. 이는 모형이 유사한 감정을 가진 단어나 표현을 더 쉽게 인식하고 일반화하는 데 도움을 주어 특히 감정 분류와 같은 태스크에 도움을 주는 요소로 작용한다.

임베딩이 중요한 또 다른 이유는 전이 학습 때문이다. 본 연구와 같이 학습 데이터가 적은 상황에서는 해당 문제를 풀기에 최적화된 임베딩 벡터값을 얻기가 쉽지 않다. 이 경우 학습 데이터보다 많은 데이터로 학습된 임베딩 모형을 사용하면 딥러닝 모형의 성능을 더욱 높일 수가 있다. 다시 말해 품질 좋은 임베딩을 딥러닝 모형의 입력값으로 사용하여 모형의 정확도와 학습 속도를 높일 수 있다. 게다가 임베딩 모형의 학습 데이터는 레이블 작업이 필요 없어 특정 도메인에 적합한 텍스트 데이터가 많이 구축되어 있다면 손쉽게 고품질의 임베딩 모형을 구축하는 것이 가능하다.

본 절에서는 앞서 구축한 약 1,500만 개의 문장 말뭉치 데이터를 활용하여 3가지 단어 임베딩 모형(Word2Vec, FastText, GloVe)을 구축한다. 이를 위해 이하에서는 각 임베딩 모형의 방법론을 살펴보고, 구축된 임베딩 모형들의 성능 평가를 수행하여 본 연구에 적합한 임베딩 모형을 선정한다.

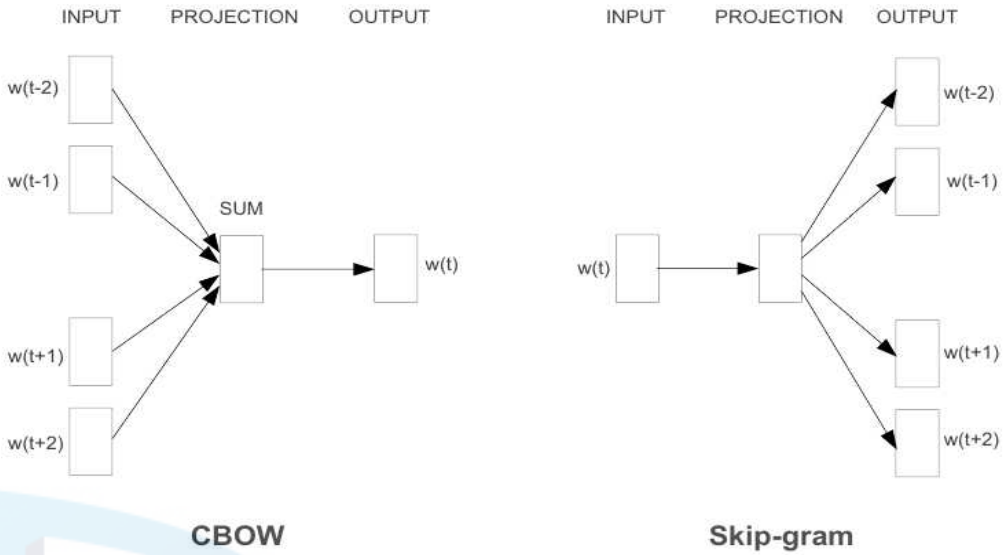
3.1.1 단어 임베딩 방법론 검토

3.1.1.1 Word2Vec

Word2Vec(Mikolov et al., 2013)는 구글 연구팀이 개발한 임베딩 기법으로 단어 임베딩 모형 중 가장 널리 쓰이고 있는 모형이다. Word2Vec의 학습 방식은 CBOW(continuous bag of words)와 Skip-gram 두 가지가 있다. CBOW는 주변에 있는 문맥 단어(context word)를 가지고 중심에 있는 타깃 단어(target word)를 예측하는 과정에서 학습되는 방식이고, Skip-gram은 타깃 단어를 가지고 문맥 단어를 예측하는 과정에서 학습되는 방식이다.

Mikolov et al.(2013)이 제안한 CBOW와 Skip-gram의 기본 구조는 다음 [그림 3-1]과 같다. 여기서 CBOW는 앞, 뒤 4개의 문맥 단어를 가지고 타깃 단어를 예측하는 모습을 보여주며, Skip-gram은 타깃 단어를 가지고 앞, 뒤 4개의 문맥 단어를 예측하는 모습을 보여준다. 이때 타깃 단어 앞뒤에 위치하는 문맥 단어의 범위를 윈도우(window)라 한다. 예를 들어 윈도우 크기가 n이면 문맥 단어의 수는 2n이 되며, 타깃 단어가 문장 맨 앞에 위치하면 문맥 단어는 타깃 단어 뒤에 있는 2개의 단어가 된다. 또한, Skip-gram은 같은 말뭉치 데이터에 대해 CBOW보다 더 많은 학습 예제를 생성한다는 특징이 있다. 즉, 윈도우가 2인 경우, CBOW는 입력 데이터 4개($w(t-1)$, $w(t-2)$, $w(t-3)$, $w(t-4)$)와 출력 데이터 1개($w(t)$)로 하나의 학습 데이터 쌍(pair)을 구성하는 반면, Skip-gram은 $\{w(t), w(t-1)\}$, $\{w(t), w(t-2)\}$, $\{w(t), w(t-3)\}$, $\{w(t), w(t-4)\}$와 같은 4개의 학습 데이터 쌍을 구성한다. 따라서 윈도우 크기에 상관없이 한 번의 학습만 수행하는 CBOW보다 윈도우 크기에 따라 반복적 학습이 가능한 Skip-gram의 임베딩 품질이 더 좋은 경향이 있다.

[그림 3-1] CBOW와 Skip-gram 구조도(Mikolov et al., 2013)



CBOW의 구체적인 학습 과정은 아래 [그림 3-2]와 같다. 먼저 CBOW의 투사층(projection layer)[53]은 문맥 단어들의 원-핫 벡터(one-hot vector)[54]를 입력으로 받는다. 이는 식 (3-1)과 같이 정의되며, 여기서 |V|는 단어 집합 V의 크기(원-핫 벡터의 차원)를, m은 $\mathbf{x}_c$가 타깃 단어일 때에 윈도우 크기를 의미한다. 투사층에 입력이 들어오면 각 문맥 단어들은 가중치 행렬 $\mathbf{W} \in \mathbb{R}^{|V| \times d}$와 내적(inner product)을 계산하여 단어 벡터를 생성한다. 이때 원-핫 벡터와 $\mathbf{W}$의 내적은 (원-핫 벡터가 특정 인덱스에만 1의 값을 가지므로) $\mathbf{W}$의 행 벡터 $\mathbf{w}$를 그대로 읽어오는 것과(look up) 동일하다. 이를 룩업 테이블(lookup table)[55]이라 하며, 임베딩 벡터의 차원이 d일 때 CBOW의

53) 투사층은 활성화 함수가 사용되지 않는 층을 말하며, 활성화 함수는 은닉층과 출력층의 출력값을 결정하는 함수를 말한다. 활성화 함수는 보통 여러 개의 층을 쌓거나 모형이 좀 더 복잡한 패턴을 학습할 수 있도록 비선형 함수를 사용한다는 특징을 갖는데, 투사층은 이러한 비선형 함수 없이 가중치 행렬과의 곱셈만 이루어져 선형층(linear layer)이라 부르기도 한다.

54) 원-핫 벡터 또는 원-핫 인코딩(one-hot encoding)은 단어 집합(vocabulary)의 크기를 벡터의 차원으로 하여 해당 단어의 인덱스에는 1의 값을, 그리고 다른 인덱스에는 0을 부여하는 단어의 벡터 표현 방법이다.

55) 자연어 처리에서 사용되는 일반적인 임베딩층은 이 룩업 테이블을 말하며, 이를 통해 원-핫 벡터는 고차원의 희소 벡터(sparse vector)에서 저차원의 밀집 벡터(dense vector)로 변환되어 진다. 그리고 임베딩층을 통과한 이 밀집 벡터를 임베딩 벡터라 부른다.

룩업 테이블은 식 (3-2)와 같이 정의된다. 이렇게 산출된 단어 벡터들은 출력층으로 보내지기 위해 식 (3-3)과 같이 평균 벡터 $\overline{\mathbf{w}} \in \mathbb{R}^{\mathrm{d}}$로 변환되고, 출력층에서는 이 평균 벡터와 또 다른 가중치 행렬 $\mathbf{W}' \in \mathbb{R}^{d \times |\mathrm{V}|}$ 56)의 내적이 계산되어 스코어 벡터(score vector) $\mathbf{z}(=\overline{\mathbf{w}}\mathbf{W}') \in \mathbb{R}^{|\mathrm{V}|}$를 산출한다. 스코어 벡터는 다시 출력층의 활성화 함수인 소프트맥스(softmax)[57]를 지나면서 식 (3-4)와 같이 벡터의 각 인덱스가 0~1 사이의 실수로 정규화된다. 여기서 $\mathbf{z}_j$는 $\mathbf{z}$의 j번째 인덱스를 말하며 이는 $\overline{\mathbf{w}}$와 $\mathbf{w}'_j$($\mathbf{W}'$의 j번째 열 벡터)의 내적으로 산출된다. CBOW는 예측된 타깃 단어 $\hat{\mathbf{y}}(=P(\mathbf{x}_c|\mathbf{x}_{c-m},\cdots,\mathbf{x}_{c-1},\mathbf{x}_{c+1},\cdots,\mathbf{x}_{c+m})) \in \mathbb{R}^{|\mathrm{V}|}$이 실제 타깃 단어 $\mathbf{y}(=\mathbf{x}_c) \in \mathbb{R}^{|\mathrm{V}|}$와 가까운 값을 갖는 것이 목적이다. 그런데 이때 $\mathbf{y}$ 역시도 하나의 인덱스만 1의 값을 갖는 원-핫 벡터이므로 $\mathbf{y}$의 j^* 인덱스가 1의 값을 갖는다면 결국 CBOW의 목표는 $\hat{\mathbf{y}}_{j^*}$를 극대화하는 것이 된다. 즉, 식 (3-5)에 정의된 CBOW의 cross-entropy 손실 함수를 최소화하기 위해서는 식 (3-4)에 의해 산출되는 $\hat{\mathbf{y}}_{j^*}$를 극대화해야 한다. 그리고 $\hat{\mathbf{y}}_{j^*}$를 극대화하기 위해서는 $\overline{\mathbf{w}}$와 $\mathbf{w}'_{j^*}$의 내적 값을 키워야 하는데, 두 벡터의 내적 값이 크다는 것은 곧 두 벡터의 코사인 유사도(cosine similarity)[58]가 1에 가까운 값을 갖는다고 해석할 수 있다. 다시 말해 내적 값의 상향은 두 단어가 벡터 공간상 가깝게 위치한다는 의미이고 이는 두 단어가 유사한 의미, 좀 더 정확하게는 두 단어가 비슷한 문맥에서 자주 표현된다고 이해할 수 있다. 따라서 모형 학습을 위한 역전파(back propagation)[59] 과정은 $\overline{\mathbf{w}}$와 $\mathbf{w}'_j$

56) $\mathbf{W}'$은 $\mathbf{W}$의 전치(transpose)나 역행렬이 아니라 서로 독립적인 행렬임에 주의해야 한다.

57) 소프트맥스는 정규화 지수함수(normalized exponential function)라고도 하며 이는 벡터 내의 원소들의 합을 1로 표준화하여 벡터의 원소들을 확률분포로 변환하는 역할을 한다.

58) 코사인 유사도는 벡터 간 유사도 측정 기법의 일종으로 아래 식과 같이 두 벡터의 내적을 두 벡터의 크기로 나눠준 값을 말한다. 코사인 유사도는 -1~1사이의 값을 가지며, 값이 1에 가까울수록 두 벡터의 방향이 매우 유사함을 의미하고, 0에 가까우면 두 벡터는 서로 독립적이거나 각도가 90도인 것으로 간주한다. 그리고 값이 -1에 가까우면 두 벡터가 반대 방향을 가리키고 있음을 의미한다.

$$\cos(\theta) = \frac{\mathbf{x}\mathbf{y}}{\|\mathbf{x}\| \, \|\mathbf{y}\|} = \frac{\Sigma_{i=1}^{n} \mathbf{x}_i \mathbf{y}_i}{\sqrt{\Sigma_{\mathrm{i}=1}^{\mathrm{n}} (\mathbf{x}_i)^2} \sqrt{\Sigma_{i=1}^{n} (\mathbf{y}_i)^2}}$$

의 내적을 극대화하는 $\mathbf{W}$와 $\mathbf{W}'$을 찾는 과정이 되며, 학습이 완료되면 $\mathbf{W}$만 d차원의 임베딩 벡터로 사용할 수도 있고, $\mathbf{W}$와 $\mathbf{W}'^{\top}$를 더하거나 $\mathbf{W}$와 $\mathbf{W}'^{\top}$를 연결(concatenate)하여 사용할 수도 있다.

$$(3\text{-}1)\quad \mathbf{x}_{c-m}, \cdots, \mathbf{x}_{c-1}, \mathbf{x}_{c+1}, \cdots, \mathbf{x}_{c+m} \in \mathbb{R}^{|\mathrm{V}|}$$

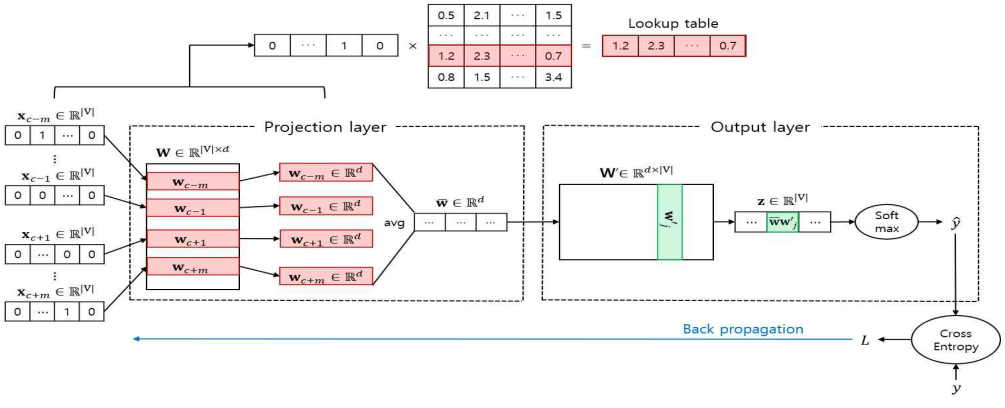
$$(3\text{-}2)\quad \mathbf{x}_{c-m}\mathbf{W}, \cdots, \mathbf{x}_{c-1}\mathbf{W}, \cdots, \mathbf{x}_{c+m}\mathbf{W} = \mathbf{w}_{c-m}, \cdots, \mathbf{w}_{c-1}, \cdots, \mathbf{w}_{c+m} \in \mathbb{R}^{d}$$

$$(3\text{-}3)\quad \overline{\mathbf{w}} = \frac{\mathbf{w}_{c-m}, \cdots, \mathbf{w}_{c-1}, \mathbf{w}_{c+1}, \cdots, \mathbf{w}_{c+m}}{2m} \in \mathbb{R}^{d}$$

$$(3\text{-}4)\quad softmax(\mathbf{z}_j) = \frac{\exp(\mathbf{z}_j)}{\sum_{k=1}^{|\mathrm{V}|}\exp(\mathbf{z}_k)} = \frac{\exp(\overline{\mathbf{w}}\mathbf{w}'_j)}{\sum_{k=1}^{|\mathrm{V}|}\exp(\overline{\mathbf{w}}\mathbf{w}'_k)} = \hat{\mathbf{y}}_j$$

$$\begin{aligned}(3\text{-}5)\quad L &= H(\hat{\mathbf{y}}, \mathbf{y}) \\ &= -\sum_{j=1}^{|\mathrm{V}|}\mathbf{y}_j \log(\hat{\mathbf{y}}_j) \\ &= -\log(\hat{\mathbf{y}}_{j^*}) \\ &= -\log\left(\frac{\exp(\overline{\mathbf{w}}\mathbf{w}'_{j^*})}{\sum_{k=1}^{|\mathrm{V}|}\exp(\overline{\mathbf{w}}\mathbf{w}'_k)}\right) \\ &= -\overline{\mathbf{w}}\mathbf{w}'_{j^*} + \log\sum_{k=1}^{|\mathrm{V}|}\exp(\overline{\mathbf{w}}\mathbf{w}'_k)\end{aligned}$$

[그림 3-2] CBOW 학습 과정



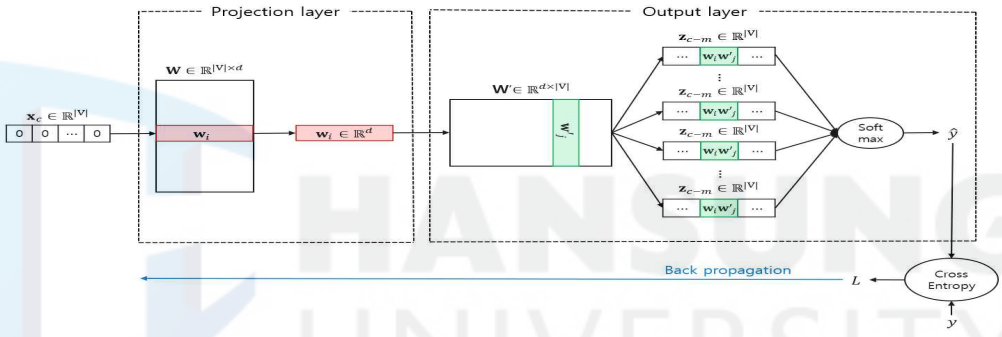
59) 역전파는 손실 함수에 대한 매개변수의 기울기(gradient)를 계산하고 이 기울기를 통해 각 매개변수를 업데이트하는 과정이다. 이때 매개변수의 업데이트는 경사 하강법(gradient descent) 또는 그 변형인 SGD, Adam 등이 사용된다.

아래 [그림 3-3]은 Skip-gram의 학습 과정을 보여준다. CBOW가 문맥 단어들을 통해 타깃 단어를 예측했다면 Skip-gram은 타깃 단어를 통해 문맥 단어들을 예측한다. Skip-gram의 연산 과정은 CBOW와 대체로 같지만 두 가지의 주요 차이점이 있다. 첫 번째로 Skip-gram은 투사층에 입력되는 원-핫 벡터(타깃 단어 $\mathbf{x}_c \in \mathbb{R}^{|V|}$)가 하나밖에 없어 평균 벡터를 구하는 과정이 없다. 따라서 출력층의 입력은 $\mathbf{W}' \in \mathbb{R}^{d \times |V|}$의 행 벡터 $\mathbf{w}_i$가 된다. 두 번째로 Skip-gram은 하나의 타깃 단어에 대해 다수의 문맥 단어를 예측하며, 각 문맥 단어는 서로 독립적으로 예측된다. 이를 위해 Skip-gram의 출력층은 입력으로 들어온 $\mathbf{w}_i$에 대해 $2m$개의 스코어 벡터를 생성한다. 이때 스코어 벡터는 CBOW와 마찬가지로 $\mathbf{w}_i$와 $\mathbf{W}'$의 내적으로 산출되며, 각 스코어 벡터는 모두 동일한 값을 갖는다. 예를 들어 $c-m$번째 스코어 벡터와 $c+m$번째 스코어 벡터의 j번째 인덱스 $\mathbf{z}_{c-m,j}$, $\mathbf{z}_{c+m,j}$는 모두 $\mathbf{w}_i$와 $\mathbf{w}'_j$의 내적 값이다. 그리고 각 스코어 벡터는 소프트맥스 함수를 통해 $2m$개의 $\hat{\mathbf{y}}(= P(\mathbf{x}_{c-m}, \cdots, \mathbf{x}_{c-1}, \mathbf{x}_{c+1}, \cdots, \mathbf{x}_{c+m} | \mathbf{x}_c)) \in \mathbb{R}^{|V|}$을 생성하며, 이에 따라 예측된 타깃 단어 $\hat{\mathbf{y}}_{c-m}, \cdots, \hat{\mathbf{y}}_{c-1}, \hat{\mathbf{y}}_{c+1}, \cdots, \hat{\mathbf{y}}_{c+m} \in \mathbb{R}^{|V|}$들의 j번째 인덱스는 식 (3-6)과 정의된다. 또한, CBOW는 하나의 인덱스 값만 극대화하면 되었으나 Skip-gram은 $2m$개의 인덱스 값을 극대화해야 한다. 즉, 실제 타깃 단어 $\mathbf{y}_{c-m}, \cdots, \mathbf{y}_{c-1}, \mathbf{y}_{c+1}, \cdots, \mathbf{y}_{c+m} \in \mathbb{R}^{|V|}$들은 모두 다른 인덱스에서 1의 값을 가지므로 극대화해야 하는 $\hat{\mathbf{y}}_{c-m}, \cdots, \hat{\mathbf{y}}_{c-1}, \hat{\mathbf{y}}_{c+1}, \cdots, \hat{\mathbf{y}}_{c+m} \in \mathbb{R}^{|V|}$들의 인덱스 위치도 모두 다르다. 따라서 Skip-gram의 손실 함수는 식 (3-7)과 같이 정의되며, 해당 식에서 j^*_{c-m+l}가 각 문맥 단어들이 1의 값을 갖는 인덱스 위치를 의미한다.

$$
\begin{aligned}
(3\text{-}6)\ softmax(\mathbf{z}_{c-m+l,j}) &= \left(\frac{\exp(\mathbf{z}_{c-m+l,j})}{\Sigma_{k=1}^{|V|}\exp(\mathbf{z}_k)}\right) \\
&= \left(\frac{\exp(\mathbf{w}_i \mathbf{w}'_{c-m+l,j})}{\Sigma_{k=1}^{|V|}\exp(\mathbf{w}_i \mathbf{w}'_k)}\right) \\
&= \hat{\mathbf{y}}_{c-m+l,j},\ \ where\ \ l = 0 \sim 2m, l \neq 1
\end{aligned}
$$

$$
\begin{aligned}
(3\text{-}7)\quad L &= H(\hat{\mathbf{y}},\mathbf{y}) \\
&= -\prod_{l=0,\,l\neq 1}^{2m}\sum_{j=1}^{|\mathrm{V}|}\mathbf{y}_{c-m+l,\,j}\log(\hat{\mathbf{y}}_{c-m+l,\,j}) \\
&= -\log\prod_{l=0,\,l\neq 1}^{2m}(\hat{\mathbf{y}}_{c-m+l,\,j^{*}_{c-m+l}}) \\
&= -\log\prod_{l=0,\,l\neq 1}^{2m}\left(\frac{\exp(\mathbf{w}_i\mathbf{w}'_{c-m+l,\,j^{*}_{c-m+l}})}{\sum_{k=1}^{|\mathrm{V}|}\exp(\mathbf{w}_i\mathbf{w}'_k)}\right) \\
&= -\sum_{l=0,\,l\neq 1}^{2m}\mathbf{w}_i\mathbf{w}'_{c-m+l,\,j^{*}_{c-m+l}} + 2m\log\sum_{k=1}^{|\mathrm{V}|}\exp(\mathbf{w}_i\mathbf{w}'_k)
\end{aligned}
$$

[그림 3-3] Skip-gram 학습 과정



3.1.1.2 FastText

FastText(Bojanowski et al., 2017)는 페이스북에서 개발한 Word2Vec 기반의 단어 임베딩 모형이다. FastText의 핵심 골자는 Word2Vec 학습 시 내부단어(subword)를 활용한 것이다. 즉, FastTex는 하나의 단어를 n-gram의 문자(charater) 단위로 쪼개어 내부단어 벡터들을 생성하고, 이 생성된 내부단어 벡터들을 모두 더하여 해당 단어의 단어 벡터를 표현한다. 예를 들어 n을 3으로 잡은 tri-gram의 경우 '하모니카'란 단어는 다음 식 (3-8)과 같이 5개 내부단어 벡터의 합으로 표현된다.

$$
(3\text{-}8)\quad \tilde{\mathbf{w}}_{\text{하모니카}} = \mathbf{z}_{<\text{하모}} + \mathbf{z}_{\text{하모니}} + \mathbf{z}_{\text{모니카}} + \mathbf{z}_{\text{니카}>} + \mathbf{z}_{<\text{하모니카}>}
$$

여기서 〈, 〉는 접두사와 접미사를 구분하기 위해 단어의 시작과 끝의 경계를 나타내는 FastText의 특수기호이며, 〈하모니카〉는 n-gram 이외에 단어 자체를 학습시키기 위해 추가되는 특별 토큰이다. 또한 실제 모형 학습에서는 n의 최솟값과 최댓값의 범위를 설정하므로 FastText의 단어 벡터는 다음 식 (3-9)와 같이 정의된다.

$$(3\text{-}9)\quad \tilde{\mathbf{w}} = \sum_{g \in G_t} \mathbf{z}_g$$

여기서 G_t는 타깃 단어 t에 속한 문자 단위 n-gram 들의 집합을 의미하며, $\mathbf{z}_g$는 각 n-gram과 가중치 행렬의 내적으로 산출된 내부단어 벡터들이다. 그리고 FastText는 이를 이용하여 Word2Vec과 동일한 방식으로 학습을 수행한다. 차이점은 $\tilde{\mathbf{w}}$에 속한 모든 문자 단위 n-gram 벡터($\mathbf{z}$)들을 업데이트하는 것이다. 따라서 FastText의 손실 함수는 식 (3-10)과 같이 정의되며, 이를 식 (3-7)과 비교해 보면 $\mathbf{w}_i$가 $\tilde{\mathbf{w}}$로 대체되었음을 알 수 있다.

$$(3\text{-}10)\quad L = -\sum_{l=0, l\neq 1}^{2m} \tilde{\mathbf{w}}\mathbf{w}'_{c-m+l, j^*_{c-m+l}} + 2m\log\sum_{k=1}^{|V|}\exp(\tilde{\mathbf{w}}\mathbf{w}'_k)$$
$$= -\sum_{l=0, l\neq 1}^{2m}\sum_{g\in G_t} \mathbf{z}_g^{\top}\mathbf{w}'_{c-m+l, j^*_{c-m+l}} + 2m\log\sum_{k=1}^{|V|}\exp(\sum_{g\in G_t}\mathbf{z}_g^{\top}\mathbf{w}'_k)$$

FastText의 이러한 학습 방법은 Word2Vec와 비교했을 때, 다음과 같은 세 가지 강점을 갖는다. 첫째, FastText는 단어의 형태론적 구조를 최대한 살려서 단어를 벡터화한다. Word2Vec는 단어의 내부 구조를 무시하고 하나의 단어에 고유한 벡터를 할당하기 때문에 단어의 형태론적 특성을 반영하지 못한다는 단점이 있다. 예를 들어 go, goes, going이란 단어를 Word2Vec로 학습하면 세 단어는 (형태소 분석을 통해 정규화 과정을 거치지 않을 때) 서로 다른 벡터가 할당되고 이 벡터는 학습을 통해 서로 유사한 값을 갖게 된다. 그런데 만약 goes가 드물게 나타나는 형태의 단어라면 Word2Vec는 학습 데이터의 부족으로 이를 올바르게 학습하기가 힘들다. FastText는 이를 위해 형태가 유사한 단어들은 유사한 벡터값을 가질 수 있도록 문자 수준 정보를 이

용하여 단어 벡터를 학습시킨다. 즉, 하나의 단어에 여러 개의 내부단어 벡터를 할당하고 이를 학습시켜 단어의 형태론적 특성을 반영한다. 이는 대부분의 동사가 40개 이상의 변화형을 가지는 프랑스어나 스페인어처럼 형태론적으로 풍부한 언어들에게 큰 장점이 된다.

둘째, FastText는 희귀 단어(rare word)에 대한 대응이 가능하다. 단어 집합 내에 출현 빈도가 적은 단어는 참고할 수 있는 경우의 수가 적다 보니 정확하게 임베딩 되지 않을 가능성이 크다. 하지만 FastText는 희귀 단어도 해당 단어의 n-gram이 다른 단어의 n-gram과 겹치는 경우라면 비교적 정확한 임베딩 벡터를 얻을 수 있다. 즉, 오타(typo)나 맞춤법이 틀린 단어에 대해서도 FastText는 일정 수준의 성능을 보일 수 있다.

마지막으로 FastText는 OOV 문제에 대한 대응이 가능하다. Word2Vec는 신조어와 같이 단어 집합 내에 없는 단어는 임베딩 벡터를 얻을 수가 없다. 하지만 FastText는 모든 단어의 n-gram에 대해서도 임베딩 벡터를 생성하므로 이를 이용하면 학습되지 않은 단어에 대해서도 유사도를 계산할 수가 있다. 예를 들어 birthday, place, birthplace란 단어가 있을 때, birthplace가 학습되지 않은 단어라면 FastText는 birth와 place라는 내부단어를 이용하여 birthplace의 단어 임베딩 벡터를 얻는다. 이는 OOV 문제에 대처하지 못하는 다른 단어 임베딩 모형들과 구분되는 FastText의 가장 큰 장점이 된다.

한편, 한국어도 Park et al.(2018)과 같이 FastText를 적용하기 위한 시도가 있었다. 영어는 단어 수준 이하의 정보를 추출하기 위해 알파벳 단위로 n-gram을 추출하는 반면, 한국어는 음절 단위와 자모 단위(초성, 중성, 종성 단위)로 n-gram을 추출할 수 있다. 예를 들어 '먹었다'라는 단어에 대해 tri-gram을 추출하면 다음과 같이 음절 단위는 한 개의 tri-gram을 추출하고 자모 단위는 여섯 개의 tri-gram을 추출한다.

음절 단위: {ㅁ,ㅓ,ㄱ,ㅇ,ㅓ,ㅆ,ㄷ,ㅏ,e}
자모 단위: {<,ㅁ,ㅓ}, {ㅓ,ㄱ,ㅇ}, {ㄱ,ㅇ,ㅓ}, {ㅆ,ㄷ,ㅏ}, {ㅓ,ㅆ,ㄷ}, {ㅏ,e,>}
(여기서 e는 종성이 없는 음절을 위해 추가된 임의에 문자열이다.)

따라서 자모 단위는 음절 단위보다 더 세분화된 n-gram 벡터를 학습할 수

있다는 장점이 있으며, 특히 한국어는 한 음절 단어가 독립된 의미를 갖는 경우가 많고 일부 형태소는 음절보다 작은 단위를 갖기 때문에 자모 단위로 n-gram을 추출하는 것이 오타나 OOV 측면에서 더 나은 성능을 기대할 수 있다.

3.1.1.3 GloVe(Global Word Vector)

GloVe(Pennington et al., 2014)는 미국 스탠포드 대학교 연구팀에서 개발한 단어 임베딩 기법이다. GloVe는 잠재의미분석(Latent Semantic Analysis, LSA)[60]과 Word2Vec의 단점을 극복하고자 나왔다. LSA는 말뭉치의 전체적인 통계량을 활용할 수 있으나, 단어 간 유추 작업(analogy task)에는 성능이 떨어진다는 단점이 있다. 반면 Word2Vec는 LSA보다 유추 작업 성능이 더 뛰어나지만, 윈도우 크기 내에 로컬 문맥(local context)만 학습하기 때문에 말뭉치의 전체적인 통계 정보를 반영하지 못한다는 단점이 있다. GloVe는 이러한 기존 방법론들의 한계를 보완하기 위해 LSA와 Word2Vec 두 가지 방법론을 모두 사용한다.

GloVe의 학습은 동시 발생(co-occurrence) 행렬을 만드는 것부터 시작한다. 동시 발생 행렬은 단어 집합에 있는 단어들을 행과 열에 넣고 윈도우 크기에 따라 동시에 등장한 횟수를 기재한 행렬을 말한다. 예를 들어 “I like deep learning.”, “I like NLP.”, “I enjoy flying.”이라는 3개의 문장이 있을 때 윈도우 크기가 1인 동시 발생 행렬 $\mathbf{X} \in \mathbb{R}^{|V| \times |V|}$(|V|: 단어 집합의 크기)는 다음 [그림 3-4]와 같이 대칭행렬(symmetric matrix)로 생성된다.

60) LSA 또는 LSI(Latent Semantic Indexing)는 단어-문서 행렬(Document-Term Matrix, DTM)이나 TF-IDF(Term Frequency-Inverse Document Frequency) 행렬에 절단된(truncated) 특이값 분해(Singular Value Decomposition, SVD)라는 기법을 사용하여 차원을 축소하고, 이를 통해 행(row)간의 숨어 있는 단어들의 잠재적(latent) 의미를 도출하는 방법론이다.

[그림 3-4] 동시 발생 행렬(Manning, et al., 2017)

$$X = \begin{array}{c} \\ I \\ like \\ enjoy \\ deep \\ learning \\ NLP \\ flying \\ . \end{array} \begin{array}{c} \begin{array}{cccccccc} I & like & enjoy & deep & learning & NLP & flying & . \end{array} \\ \begin{bmatrix} 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

이렇게 생성된 동시 발생 행렬은 모형 학습 시 동시 발생 확률을 계산하기 위해 사용된다. 동시 발생 확률은 동시 발생 행렬의 i번째 행의 총합을 분모로 하고, i행 k열의 값을 분자로 한 값이다. 즉, i가 타깃 단어, k가 문맥 단어라면 타깃 단어 i가 등장했을 때 문맥 단어 k가 등장할 조건부 확률 $P(k|i)$을 의미한다. 아래 [표 3-1]은 Pennington et al.(2014)에서 제시된 동시 발생 확률의 예시를 보여준다. 여기서 $P(k|ice)$와 $P(k|steam)$는 타깃 단어 얼음(ice), 증기(steam)와 문맥 단어 k의 동시 발생 확률을 의미한다. 이 값은 타깃 단어와 문맥 단어가 관련성이 높을 때 큰 값을 갖고, 관련성이 없을 때 작은 값을 갖는다. 그리고 $P(k|ice)/P(k|steam)$는 두 동시 발생 확률의 비율이다. 이 값은 문맥 단어 k에 따라 1보다 매우 크거나 작은 값을 가질 수도 있고 1에 가까운 값을 가질 수도 있다. 즉, k가 얼음과 관련된 단어(solid)라면 1보다 큰 값을 갖게 되고, k가 증기와 관련된 단어라면 1보다 작은 값을 갖게 되며, k가 두 타깃 단어 모두와 관련이 있는 경우(water)와 없는 경우(fashion)에는 1에 가까운 값을 갖는다. 따라서 동시 발생 확률의 비율은 동시 발생 확률과 비교했을 때 두 단어와 관련된 단어(solid, gas)와 관련 없는 단어(water, fashion) 사이를 구별하는 데 있어 더 나은 능력을 보여줄 뿐만 아니라 관련된 단어 사이의 연관성을 파악하는 데에도 유용함을 보여준다.

[표 3-1] 동시 발생 확률과 그 비율(Pennington et al., 2014)

	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4} (large)	6.6×10^{-5} (small)	3.0×10^{-3} (large)	1.7×10^{-5} (small)
$P(k\|steam)$	2.2×10^{-5} (small)	7.8×10^{-4} (large)	2.2×10^{-3} (large)	1.8×10^{-5} (small)
$\frac{P(k\|ice)}{P(k\|steam)}$	8.9 (large)	8.5×10^{-2} (small)	1.36 (close to one)	0.96 (close to one)

GloVe는 단어 벡터의 학습이 동시 발생 확률이 아닌 그 비율이 되어야 함을 주장했다. 즉 GloVe의 목적은 동시 발생 확률 비(ratio)를 산출하는 함수를 찾고, 이 함수를 이용해 타깃 단어 벡터와 문맥 단어 벡터의 내적을 계산하여 그 값이 동시 발생 확률이 되도록 임베딩 벡터를 만드는 것이다. 이를 위해 GloVe는 다음 식 (3-11)과 같이 동시 발생 확률 비가 세 단어(i, j, k)에 의존한다는 것부터 논의를 시작한다.

$$(3\text{-}11)\quad F(\mathbf{w}_i, \mathbf{w}_j, \tilde{\mathbf{w}}_k) = P_{ik}/P_{jk}$$

여기서 좌변에 있는 $\mathbf{w}_i \in \mathbb{R}^d$, $\mathbf{w}_j \in \mathbb{R}^d$는 d차원에 임베딩된 타깃 단어 벡터이고, $\tilde{\mathbf{w}}_k \in \mathbb{R}^d$는 d차원에 임베딩된 문맥 단어 벡터이다. 그리고 우변에 있는 P_{ik}와 P_{jk}는 동시 등장 확률 $P(k|i)$와 $P(k|j)$이다. 해당 식을 만족하는 함수 F를 찾기 위해서는 먼저 $P(k|i)$와 $P(k|j)$의 비율 정보가 벡터 공간에 인코딩되어야 한다. 그리고 벡터 공간은 본질적으로 선형 구조를 가지므로 이를 수행하는 가장 자연스러운 방법은 벡터의 차이를 이용하는 것이다. 즉, $\mathbf{w}_i$와 $\mathbf{w}_j$의 차이를 F의 입력으로 사용하면 식 (3-11)은 다음과 같이 변경된다.

$$(3\text{-}12)\quad F(\mathbf{w}_i - \mathbf{w}_j, \tilde{\mathbf{w}}_k) = P_{ik}/P_{jk}$$

여기서 F의 인자는 벡터지만 우변은 스칼라값을 가진다. 이를 성립시키기 위한 다양한 방법이 있지만 GloVe는 선형 구조를 유지하기 위해 다음 식

(3-13)과 같이 F 인자에 내적을 수행한다.

$$(3\text{-}13)\quad F((\mathbf{w}_i - \mathbf{w}_j)^\top \tilde{\mathbf{w}}_k) = P_{ik}/P_{jk}$$

또한 실제 학습에서는 타깃 단어와 문맥 단어가 무작위로 선택되므로 F는 $\mathbf{w}_i$와 $\tilde{\mathbf{w}}_k$의 위치가 바뀌어도 성립되는 함수여야 한다. 이를 만족하기 위해서는 F가 준동형사상(homomorphism)[61]을 만족하는 함수여야 하는데, 이를 위해 GloVe는 P_{ik}/P_{jk}를 $F(\mathbf{w}_i^\top \tilde{\mathbf{w}}_k / \mathbf{w}_j^\top \tilde{\mathbf{w}}_k)$로 정의하여 식 (3-13)을 다음과 같이 변경한다.

$$(3\text{-}14)\quad F(\mathbf{w}_i^\top \tilde{\mathbf{w}}_k - \mathbf{w}_j^\top \tilde{\mathbf{w}}_k) = F(\mathbf{w}_i^\top \tilde{\mathbf{w}}_k)/F(\mathbf{w}_j^\top \tilde{\mathbf{w}}_k)$$

이때 식 (3-14)를 정확하게 만족시키는 준동형사상 F는 지수함수가 된다. 따라서 F를 지수함수 exp라 표기하면 식 (3-14)는 다음과 같이 정의된다.

$$(3\text{-}15)\quad \exp(\mathbf{w}_i^\top \tilde{\mathbf{w}}_k - \mathbf{w}_j^\top \tilde{\mathbf{w}}_k) = \exp(\mathbf{w}_i^\top \tilde{\mathbf{w}}_k)/\exp(\mathbf{w}_j^\top \tilde{\mathbf{w}}_k)$$

함수 F를 찾았으므로 이제 이를 이용하여 타깃 단어 벡터와 문맥 단어 벡터의 내적이 동시 발생 확률이 되도록 모형을 학습해야 한다. 다시 말해 $\exp(\mathbf{w}_i^\top \tilde{\mathbf{w}}_k) = P_{ik}$가 되는 임베딩 벡터 $\mathbf{w}_i$와 $\tilde{\mathbf{w}}_k$를 찾아야 한다. 이때 레이블에 해당하는 P_{ik}는 동시 등장 행렬 $\mathbf{X}$에 따라 $P_{ik} = P(k|i) = \mathbf{X}_{ik}/\mathbf{X}_i$로 산출되므로 학습 대상이 되는 $\exp(\mathbf{w}_i^\top \tilde{\mathbf{w}}_k) = P_{ik}$는 양변에 로그를 취하면서 다음과 같이 정리된다.

$$(3\text{-}16)\quad \mathbf{w}_i^\top \tilde{\mathbf{w}}_k = \log P_{ik} = \log(\mathbf{X}_{ik}/\mathbf{X}_i) = \log\mathbf{X}_{ik} - \log\mathbf{X}_i$$

---

61) 준동형사상은 대수학에서 사용되는 용어로 동일한 대수적 구조 X(연산: $\bullet$)와 Y(연산: $\circ$)가 있을 때 모든 $x_1, x_2 \in X$에 대해 $f(x_1 \bullet x_2) = f(x_1) \circ f(x_2)$를 만족하는 함수 $f: X \rightarrow Y$를 의미한다. 가령 지수함수(의 덧셈법칙)는 실수의 덧셈($\mathbb{R}$, +)에서 양수의 곱셈($\mathbb{R}^+$, ×)으로 가는 준동형사상이다.

그런데 식 (3-16)은 앞서 언급하였듯이 $\mathbf{w}_i$와 $\tilde{\mathbf{w}}_k$의 위치가 바뀌어도 된다는 조건이 필요하다. 즉, $\log P_{ik} = \log P_{ki} \rightarrow \log \mathbf{X}_{ik} - \log \mathbf{X}_i = \log \mathbf{X}_{ki} - \log \mathbf{X}_k$가 성립해야 한다. 이때 $\log \mathbf{X}_{ik}$와 $\log \mathbf{X}_{ki}$는 동시 등장 행렬 $\mathbf{X}$가 대칭행렬이므로 문제 되지 않으나 문제는 $\log \mathbf{X}_i \neq \log \mathbf{X}_k$가 된다. 이로 인해 GloVe는 $\log \mathbf{X}_i$를 다른 값으로 대체하면서 식을 한 번 더 변경한다. 즉, $\log \mathbf{X}_i$는 k와 독립적이므로 이를 $\mathbf{w}_i$에 대한 편향 $\mathbf{b}_i$로 대체하는 한편, $\tilde{\mathbf{w}}_k$에 대한 편향 $\tilde{\mathbf{b}}_k$도 추가하여 양변의 대칭성이 만족하도록 식을 변경한다.

$$\mathbf{w}_i^{\top} \tilde{\mathbf{w}}_k + \mathbf{b}_i + \tilde{\mathbf{b}}_k = \log \mathbf{X}_{ik} \quad (3\text{-}17)$$

마지막으로 식 (3-17)를 이용해 GloVe의 손실 함수를 정의해야 한다. 해당 식에서 우변은 동시 등장 행렬을 통해 이미 알고 있는 값, 즉 레이블을 의미하며, 좌변에 있는 4개의 항은 학습을 통해 업데이트되는 매개변수를 의미한다. 따라서 최소화해야 하는 GloVe의 손실 함수는 좌변과 우변의 차이로 정의할 수 있는데, 문제는 로그의 인자 $\mathbf{X}_{ik}$가 [그림 3-4]에서처럼 0의 값을 갖는 경우가 많다는 것이다. 이를 위한 대안으로 $\log \mathbf{X}_{ik}$를 $\log(1+\mathbf{X}_{ik})$로 변경하는 방법이 있으나, 이를 사용하여도 $\mathbf{X}_{ik}$가 작은 값을 갖거나 0인 경우에 동등한 가중치가 부여된다는 문제가 남는다. $\mathbf{X}_{ik}$, 즉 동시 등장 빈도가 작은 값을 갖는다는 것은 그만큼 적은 정보를 내포함을 의미한다. 게다가 동시 등장 행렬에는 0뿐만 아니라 동시 등장 빈도가 작은 값을 갖는 경우도 많으므로 동시 등장 빈도가 작은 단어 쌍에는 동시 등장 빈도가 큰 단어 쌍보다 작은 가중치를 부여할 필요가 있다. GloVe는 이를 해결하기 위해 $\mathbf{X}_{ik}$에 의해 영향을 받는 가중치 함수 $f(\mathbf{X}_{ik})$를 손실 함수에 도입하였으며, 이는 다음 식 (3-18)과 같이 정의된다.

$$f(x) = \begin{cases} (x/x_{\max})^{\alpha} & \text{if } x < x_{\max} \\ 1 & otherwise \end{cases} \quad (3\text{-}18)$$

여기서 α와 $x_{\max}$는 모형 학습 시 사용되는 하이퍼파라미터이며 기본값은 각각 0.75와 100이다. α는 단어 쌍의 동시 등장 빈도에 따라 가중치를 조절하는 역할을 하며, Pennington et al.(2014)에서는 α=1인 선형 버전보다 α=3/4인 경우에 더 나은 성능을 보인다고 제시하였다. $x_{\max}$는 동시 등장 빈도의 최댓값으로 이를 넘으면 $f(x)$는 1을 반환한다. 즉, 지나치게 높은 빈도를 갖는 단어 쌍의 영향을 줄이기 위한 장치이다. 그리고 이렇게 정의된 $f(x)$는 GloVe의 손실 함수에 곱해지면서 가중치 역할을 하게 된다. 따라서 식 (3-18)을 최소 제곱 문제로 설정하고, 여기에 $f(\mathbf{X}_{ik})$를 도입한 GloVe의 최종적인 손실 함수는 다음 식 (3-19)와 같이 일반화된 식으로 정의된다.

$$L = \sum_{i,j=1}^{|V|} f(\mathbf{X}_{i,j})(\mathbf{w}_i^{\top}\tilde{\mathbf{w}}_j + \mathbf{b}_i + \tilde{\mathbf{b}}_j - \log\mathbf{X}_{ij})^2 \quad (3\text{-}19)$$

(여기서 |V|는 단어 집합의 크기를 의미한다.)

3.1.2 임베딩 모형 구축 및 평가

3.1.2.1 임베딩 모형 구축

임베딩 모형 학습에 사용된 말뭉치 데이터는 2.2에서 구축한 경제정책 뉴스기사이다. 해당 데이터는 약 1,500만 개의 문장으로 이루어져 있으며, 각 문장은 품사가 부착된 형태소로 구성되어 있다. 품사가 부착된 형태소를 학습 데이터로 사용한 이유는 동음이의어를 구별하여 임베딩 벡터를 생성하기 위함이다. 예를 들어 '이'라는 단어는 관형사(this), 일반명사(tooth), 수사(two) 등 다양하게 구분될 수 있지만, 품사가 부착되지 않은 채 임베딩 벡터를 학습하면 이들 모두에게 같은 임베딩 벡터를 할당하는 문제가 발생한다. 이를 보완하기 위해 본 연구에서는 '이(MM)', '이(NNG)', '이(NR)'와 같은 형태로 품사 표식 작업을 수행하였으며, 이렇게 품사가 부착된 형태소 중 한자(SH), 덧붙은 받침(Z_CODA), 해시태그(#abcd), 일련번호(W_SERIAL) 등의 품사

가 붙은 형태소는 학습에서 제외하였다.[62] 그 결과, 전체 말뭉치에 등장한 형태소의 수는 약 3억 7천 8백만 개로 나타났으며, 중복을 제거한 형태소의 수(단어 집합의 크기)는 529,589개로 나타났다.

모형 학습에 사용된 주요 하이퍼파라미터는 [표 3-2]와 같다. Word2Vec와 FastText의 모형 아키텍처는 Skip-gram이고, 임베딩 차원과 윈도우 크기는 최정명·김유섭(2019)의 연구를 참고하여 300과 6으로 설정하였다. 또한, 모형 학습에 포함할 단어의 최소 빈도수는 기본값인 5로 설정하였으며, FasText의 자모 단위 n-gram은 부착된 품사를 고려하여 13~29로 설정하였다. GloVe의 경우에는 임베딩 차원은 300으로, 동시 발생 행렬 생성 시 참고하는 윈도우 크기는 6으로 설정하였으며, 가중치 함수의 하이퍼파라미터인 α와 $x_{\max}$는 기본값인 0.75와 100으로 설정하였다.

[표 3-2] 임베딩 모형 하이퍼파라미터

Model	Hyperparameter	Value
Word2Vec (Skip-gram)	vector dimension	300
	window size	6
	minimum word count	5
FastText (Skip-gram)	vector dimension	300
	window size	6
	minimum word count	5
	n-gram	13-29
GloVe	vector dimension	300
	window size	6
	alpha	0.75
	maximum co-occurrence count	100

단어 임베딩 모형의 학습 결과는 [표 3-3]과 같다. Word2Vec의 경우, 가장 빠른 학습 시간과 적은 메모리 크기를 나타낸 반면, 같은 말뭉치 크기로 학습된 FastText는 가장 느린 학습 시간과 큰 메모리 크기를 나타냈다. 이는

62) 형태소 분석기는 2.2.5에서 언급하였듯이 Kiwi를 사용하였으며, Kiwi에서 사용되는 품사 표식 목록은 앞에 [표 2-3]과 Kiwi 깃허브(https://github.com/bab2min/Kiwi)에서 확인할 수 있다.

FastText가 단어의 임베딩 벡터 외에도 단어를 구성하는 n-gram 벡터까지 생성하는 것에서 기인하며, 실제로 3.5GB 중 2.2GB가 n-gram 벡터로 인한 파일 크기를 나타낸다. 또한 GloVe는 두 모형과 달리, 전체 말뭉치 데이터로 학습되었기 때문에 FastText보다 큰 파일 크기를 나타냈지만, 학습 시간이 FastText와 비슷한 것으로 보아 학습 속도 측면에서는 GloVe가 가장 우수한 것으로 판단된다. 마지막으로 Word2Vec와 FastText는 동음이의어 10,421개가 포함된 218,370개의 임베딩 벡터가 생성되었으며, GloVe는 동음이의어 40,091개가 포함된 529,589개의 임베딩 벡터가 생성되었다.

[표 3-3] 임베딩 모형 구축 결과

구분	Word2Vec	FastText	GloVe
학습 소요 시간	약 1시간 10분	약 1시간 30분	약 1시간 25분
파일 크기	1.3GB	3.5GB	4.5GB
단어 집합 크기	218,370	218,370	529,589

3.1.2.2 임베딩 모형 평가

임베딩 모형의 성능을 평가하는 방법은 크게 내재적 평가와 외재적 평가가 있다(Schnabel et al., 2015). 내재적 평가는 학습된 임베딩 벡터를 직접적으로 평가하는 방식으로 임베딩에 의미론적(semantic), 구문론적(syntactic) 관계가 얼마나 잘 녹아 있는지를 평가하는 방법이다. 대표적으로 단어 유추 평가와 단어 유사도 평가가 수행된다. 외재적 평가는 학습된 단어 임베딩 모형을 특정 다운스트림 태스크에 적용하여 태스크의 성능을 비교하는 방식이다. 본 연구에서는 감정 분류가 이에 해당한다.

1) 단어 유추 평가

단어 유추 평가는 Word2Vec 모형을 개발한 구글 연구진이 제시한 평가 방법이다. 이는 ‘a는 b에 있어서 c는 무엇인가?’와 같은 유추(analogy) 문제

를 사용하여 단어 간 의미적, 문법적 관계가 임베딩에 얼마나 잘 녹아 있는지를 정량적으로 평가하는 방법이다. 예를 들어 'man과 woman의 관계는 king과 queen의 관계와 같다'라는 문장은 'man - woman + king = queen'과 같은 사칙연산으로 표현할 수 있으므로, 이를 이용하면 임베딩 모형이 유추 문제를 푸는 것이 가능하다. 즉, 'man - woman + king'의 임베딩 벡터값을 계산한 후, 해당 벡터값의 코사인 유사도가 가장 높은 단어를 찾아 'queen'과 일치하는지를 확인하는 방식이다.

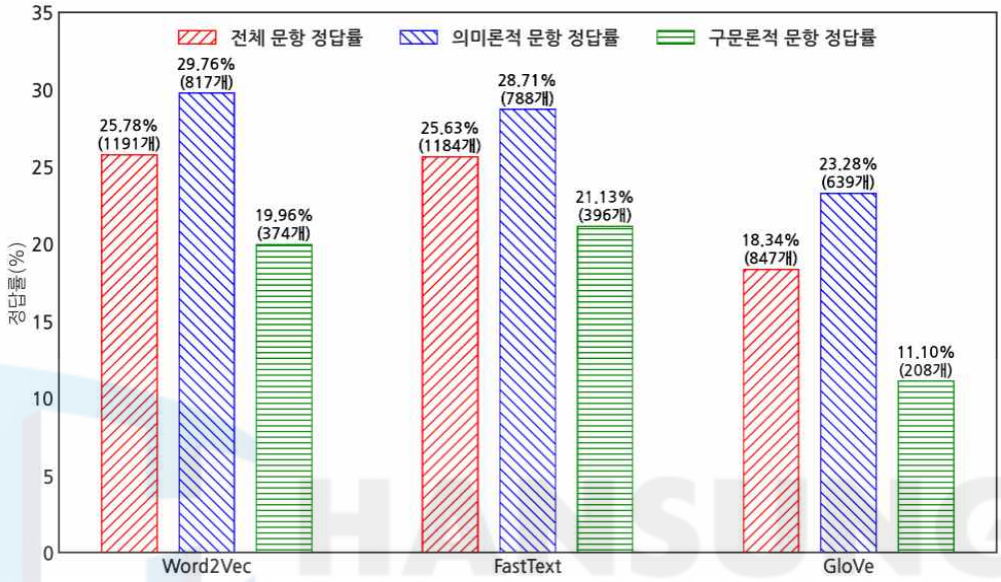
단어 유추 평가를 위한 테스트셋은 구글 연구진이 제시한 GATS(Google Analogy Test Set)를 번역해서 사용하는 경우가 많다. GATS는 의미론적 검사 세트와 구문론적 검사 세트로 구성되어 있으며, 이 중 의미론적 검사 세트는 대부분 한국어에도 무리 없이 적용할 수 있다. 그러나 구문론적 검사 세트는 한국어에 그대로 적용하기 어려운 경우가 많다. 예를 들어 영어의 단수형과 복수형의 관계를 한국어에 적용하려면 '-들'이라는 명사파생접미사를 분리하지 않고 형태소 분석을 수행해야 하는 문제가 발생한다. 이를 고려하여 본 연구에서는 강형석·양장훈(2018)이 구축한 KATS를 활용하여 단어 유추 평가를 수행하였다. KATS는 BATS(Bigger Analogy Test Set)[63]를 한국어 특성에 맞게 재구성한 한국어 유추 평가 테스트셋으로 의미론적 질의 2,746개와 구문론적 질의 1,874개로 구성되어 있다.

유추 평가 결과는 아래 [그림 3-5]와 같다. 여기서 정답률은 처리 문항 수 대비 정답으로 판별된 문항의 비율을 의미하며, 빨간색 막대는 전체 문항의 정답률을, 파란색과 초록색 막대는 각각 의미론적 문항과 구문론적 문항의 정답률을 의미한다. 평가 결과에서는 예측 기반으로 단어 벡터를 학습하는 Word2Vec와 FastText가 행렬 기반으로 단어 벡터를 학습하는 GloVe보다 모든 문항에서 더 높은 정답률을 나타냈다. 또한, Word2Vec와 FastText를 비교하였을 때는 Word2Vec가 FastText보다 의미론적 문항에서는 높은 정답률을, 구문론적 문항에서는 낮은 정답률을 나타냈다. 다만, 전체 문항을 기준으로 살펴봤을 때는 Word2Vec가 FastText보다 근소하게 높은 정답률을 나타

63) Gladkova et al.(2016)이 GATS(Google Analogy Test Set)를 확장하여 개발한 단어 유추 평가 테스트셋이다.

냈으며, 정답 개수 차이는 7개에 불과하여 두 모형 간의 성능 차이는 미미한 수준으로 판단된다.

[그림 3-5] 유추 평가 결과



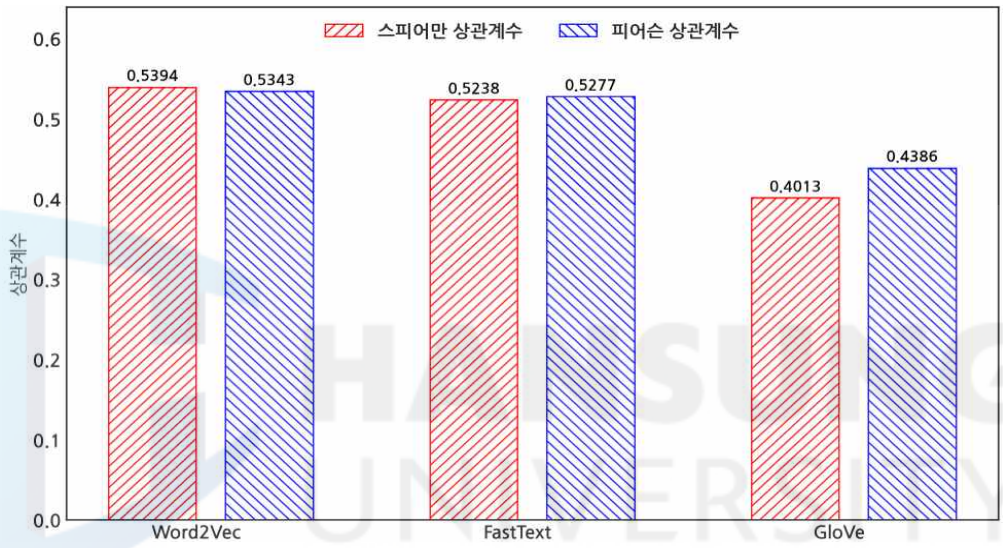
2) 단어 유사도 평가

단어 유사도 평가는 일련의 단어 쌍에 대해 사람이 직접 평가한 유사성 점수와 임베딩 벡터로 계산한 유사성 점수 사이의 상관관계(스피어만 순위 상관계수, 피어슨 상관계수)를 계산하여 단어 임베딩의 품질을 평가하는 방법이다. 이때 사람이 평가한 유사성 점수는 0~10점 척도로 작성되며, 임베딩 벡터의 유사성 점수는 단어 벡터 사이의 코사인 유사도를 통해 계산된다.

단어 유사도 평가를 위해 사용되는 테스트셋은 WordSim353(Finkelstein et al.,2002)이 주로 사용된다. 이 테스트셋은 총 353개의 영어 단어 쌍으로 구성되며, 대부분이 명사 중심의 단어 쌍으로 이루어져 있다. 본 연구에서도 이를 한국어로 번역하여 평가에 활용하였으며, 번역 시 원문의 의미가 보존되지 못하거나 한국의 지리적·시대적 상황과 부합하지 않는 단어 쌍 42개는 평가 단어에서 제외하였다.

유사도 평가 결과는 아래 [그림 3-6]과 같다. 여기서 빨간색 막대는 스피어만(Spearman) 상관계수를, 파란색 막대는 피어슨(Pearson) 상관계수를 의미한다. 평가 결과는 유추 평가와 유사하게 나타났다. GloVe는 세 모형 중 가장 낮은 성능을 보였으며, Word2Vec는 FastText보다 상대적으로 강한 상관관계를 나타냈으나, 그 차이는 매우 근소하게 나타났다.

[그림 3-6] 유사도 평가 결과



3) 외재적 평가: 감정 분류

외재적 평가는 사전 훈련된 임베딩 모형을 다양한 자연어 처리 태스크에 적용하여 임베딩의 품질을 간접적으로 평가하는 방법이다. 본 연구는 감정 분류 모형을 구축하는 것이 목적이므로 여기서는 간단한 감정 분류 모형을 구축하여 각 임베딩 모형을 전이하였을 때 얼마나 성능 향상을 가져오는지를 측정해보기로 한다.

평가에 활용된 감정 분류 모형은 다음과 같이 설계되었다.[64] 먼저, 입력층(input layer)[65]은 구축된 임베딩 모형으로 초기화된 임베딩 행렬을 입력으로

64) 딥러닝 모형의 방법론 및 학습 방법에 대한 자세한 내용은 다음 절에서 다룬다.

65) 외부 데이터를 신경망에 제공하여 다음 계층인 은닉층으로 전달하는 역할을 한다.

받아 미세 조정 과정을 거친다. 이때 임베딩 초기화 시 발생하는 OOV 단어 문제는 모든 단어 벡터의 평균으로 처리하였으며, FastText의 경우에는 n-gram 벡터로 OOV 단어를 처리할 수 있으므로 단어 벡터의 평균을 활용한 경우와 n-gram 벡터를 활용한 경우를 구분하여 모형을 구축하였다. 다음으로 은닉층(hidden layer)[66]은 1개의 BiLSTM층으로만 설계하였으며, 은닉 상태의 크기는 32, 64, 128을 비교하여 분류 정확도가 높은 64로 설정하였다. 마지막 출력층(output layer)[67]은 다중 분류 문제를 풀기 위해 소프트맥스(softmax) 함수를 사용하였으며, 이에 따라 손실 함수는 categorical cross entropy가 사용되었다. 또한 레이블 불균형 문제에 대응하기 위해 레이블 가중치를 손실 함수에 반영하였으며, 최적화(optimizer) 알고리즘은 초기 학습 단계의 불안정성을 해결하기 위해 Adam의 변형인 RAdam(Rectified Adam)을 사용하였다.

모형 구축 결과는 [표 3-4]와 같다. 여기서 정확도는 계층적 5-fold CV의 수행 결과를 나타내며,[68] FastText와 FastText_n은 각각 n-gram 벡터를 활용하지 않은 경우와 활용한 경우의 평가 결과를 나타낸다. 평가 결과는 내재적 평가와 마찬가지로 Word2Vec가 가장 높은 성능을 기록하였고, GloVe는 세 모형 중 가장 낮은 성능을 나타냈다. 구체적으로 Word2Vec는 전이 학습을 수행하지 않았을 때의 정확도(74%)보다 약 8.5%P 향상된 82.5%의 정확도를 기록하였으며, GloVe는 약 5.1%P 향상된 79.1%의 정확도를 기록하였다. 한편, FastText는 전이 학습을 수행하지 않은 정확도보다 약 7.2%P가 향상된 81.2%의 정확도를 기록하였고, n-gram 벡터를 활용한 FastText_n은 약 7.7%P가 향상된 81.7%의 정확도를 기록하였다. FastText의 이러한 결과는 앞서 언급한 FastText의 장점이 부각된 결과로 해석될 수 있다. 즉, 경제 정책 뉴스기사에 나타난 OOV 단어에 대해서도 FastText는 n-gram 벡터를 활용하여 적절한 단어 벡터를 생성한 것으로 평가된다.

66) 입력 데이터의 패턴이나 특징을 학습하는 계층이다. 은닉층은 여러 개의 뉴런으로 구성되며, 은닉층이 두 개 이상인 경우를 심층 신경망(Deep Neural Network, DNN)이라 한다.

67) 은닉층에서 받은 정보를 바탕으로 신경망의 최종 예측이나 분류 결과를 제공하는 계층이다.

68) 평가 방법에 대한 자세한 설명은 3.2.2.1에 제시되어있다.

[표 3-4] 외재적 평가 결과

모형	정확도	성능 향상
Word2Vec	82.5%	8.5%P
FastText	81.2%	7.2%P
FastText_n	81.7%	7.7%P
GloVe	79.1%	5.1%P

3.1.3 임베딩 모형 선정

이상의 단어 임베딩 평가 결과를 바탕으로 본 연구에서는 세 가지 평가 모두에서 가장 우수한 성능을 나타낸 Word2Vec를 본 연구에 최종 임베딩 모형으로 선정하였다. 다만, FastText는 내재적 평가에서 Word2Vec와의 성능 차이가 매우 근소하였고, 외재적 평가에서도 정확도 차이가 0.5%P에 불과하여 이 역시도 충분히 좋은 선택지로 평가된다. 특히, 외재적 평가에 활용된 본 연구의 학습 데이터셋에는 OOV 단어가 290개로 극히 제한적이었기 때문에 보다 방대한 규모의 학습 데이터셋을 활용하였다면 평가 결과가 달라졌을 가능성도 존재한다. 따라서 향후 더 많은 양의 뉴스기사가 수집되어 신조어와 같은 OOV 단어의 출현 빈도가 증가하는 상황이 온다면 FastText의 활용 여부도 적극적으로 검토할 필요가 있다고 판단된다.

3.2 감정 분류 모형(Classifier)

감정 분류 모형은 RNN, CNN, 트랜스포머의 세 가지 딥러닝 아키텍처를 고려하여 구축한다. 본 절에서는 각 아키텍처의 이론적 배경과 구조적 특성을 상세히 고찰하고, 이를 적용하여 설계한 감정 분류 모형들의 성능을 비교·평가하는 과정을 진행한다.

3.2.1 인공 신경망(Artificial Neural Network) 방법론 검토

3.2.1.1 순환 신경망(Recurrent Neural Network)

1) 순환 신경망의 기본 구조와 연산 과정

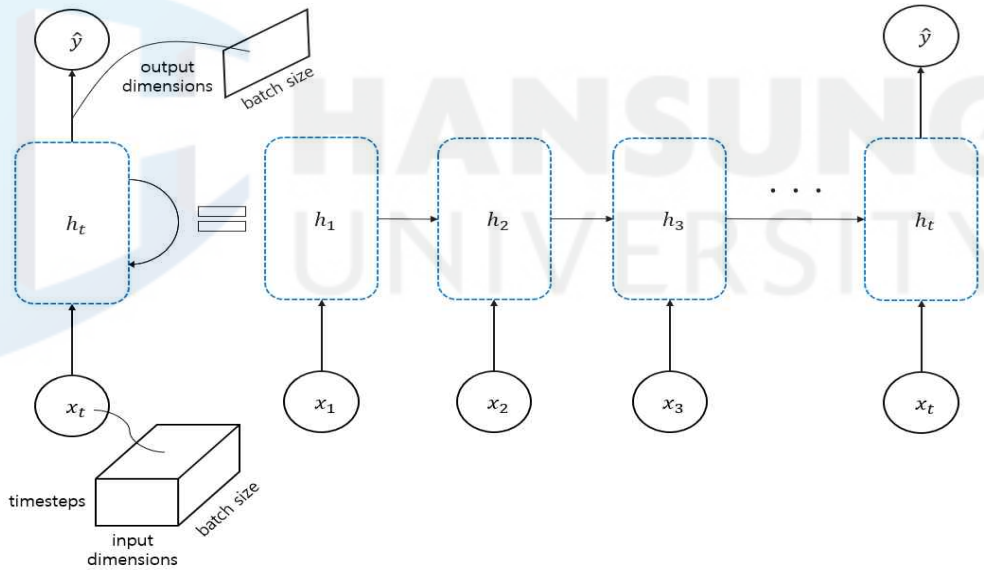
과거 RNN 등장 이전에는 주로 입력층, 은닉층, 출력층 순으로만 연산이 전개되는 순방향 신경망(Feed-Forward Neural Network, FFNN)을 활용하여 순차 데이터(sequential data)를 처리하였다. 그러나 FFNN은 입력의 길이가 고정되어 있어 시퀀스 길이가 가변적인 데이터를 처리할 수 없었고,[69] 단방향 구조로 인해 현재 입력값과 과거 입력값의 관계를 파악할 수도 없었다. 이로 인해 시퀀스 길이의 가변성과 시간에 따라 변화하는 데이터의 패턴을 학습할 필요성이 제기되었고, 이러한 요구에 대응하기 위해 제안된 인공 신경망이 RNN이다.

RNN의 핵심 아이디어는 모형 내에 순환 구조를 도입한 것이다. 즉, 은닉층의 출력값이 다시 은닉층의 입력값으로 재사용되는 구조다. 이를 이해하기 위해 아래 [그림 3-7]에는 RNN의 기본 구조를 도식화하였다. 하나씩 살펴보면, 먼저 RNN은 (배치 크기, 시점의 수, 입력 차원의 수)의 3D 텐서를 입력으로 받는다. 여기서 배치 크기(batch size)는 한 번에 학습되는 샘플의 수, 시점의 수(timesteps)는 시퀀스의 길이, 입력 차원의 수(input dimensions)는

69) 시퀀스는 순서대로 나열된 일련의 객체들을, 시퀀스의 길이는 이 객체들의 개수를 의미한다. 예컨대, 하나의 문장이 시퀀스라면 시퀀스의 길이는 각 문장을 구성하는 단어의 수가 된다.

임베딩 벡터의 차원을 의미한다. 입력이 들어온 후에는 메모리 셀(파란색 점선의 사각형)에서 이전 시점의 메모리 셀에서 나온 값을 입력으로 재사용하는 재귀적 활동이 수행되는데, 이때 각 메모리 셀의 출력값을 은닉 상태(hidden state), 메모리 셀 안에 있는 뉴런의 수를 은닉 상태의 크기(utits)[70]라 한다. 즉, 아래 그림은 첫 번째 입력(x_1)이 들어오면 첫 번째 은닉 상태(h_1)가 산출되고, 두 번째 입력(x_2)이 들어오면 첫 번째 은닉 상태와 두 번째 입력이 계산되어 두 번째 은닉 상태(h_2)가 산출되는 구조를 보여준다. 그리고 메모리 셀의 마지막 시점에서는 (출력 차원의 수[71], 배치 크기)의 은닉 상태(h_t)가 출력되고, 이를 출력층으로 보내 최종 예측치 $\hat{y}$을 산출한다.[72]

[그림 3-7] RNN 구조도



70) 각각의 시점마다 유지되는 은닉 상태 벡터의 차원을 말하며, 이를 통해 모형이 기억할 수 있는 정보의 양, 즉 모형의 용량(capacity)이 결정된다.

71) 이때 출력 차원의 수(output dimensions)는 은닉 상태의 크기가 된다.

72) 다만, RNN의 입력과 출력 형태는 풀고자 하는 문제에 따라 달라진다. 예컨대, [그림 3-7]은 감정 분류 문제에 사용되는 다대일(many-to-one) 구조의 RNN을 나타내었지만, 번역기, 품사 표식 작업, 또는 RNN을 여러 개 쌓는 깊은 순환 신경망(Deep Recurrent Neural Network)에서는 다대다(many-to-many) 구조의 RNN을 설계해야 한다. 그리고 다대다 구조에서는 메모리 셀의 각 시점이 (배치 크기, 시점의 수, 출력 차원의 수)의 은닉 상태를 출력층으로 내보낸다.

RNN의 구체적인 연산 과정은 식 (3-20), (3-21)과 같다([그림 3-8]). 해당 식에서 $\mathbf{x}_t$는 t시점의 단어 임베딩 벡터, f와 g는 활성화 함수, $\mathbf{W}_x$, $\mathbf{W}_h$, $\mathbf{W}_y$는 가중치 행렬, $\mathbf{b}_h$와 $\mathbf{b}_y$는 편향(bias) 벡터를 의미한다. $\mathbf{W}_x$, $\mathbf{W}_h$, $\mathbf{W}_y$는 모든 시점마다 같은 값을 사용하며, RNN 층이 2개 이상일 때는 층마다 다른 값을 사용한다. 메모리 셀의 활성화 함수는 일반적으로 하이퍼볼릭 탄젠트(hyperbolic tangent, tanh)[73] 함수가 사용되며, 출력층은 분류 문제에 따라 이진 분류에서는 시그모이드, 다중 분류에서는 소프트맥스[74] 함수가 사용된다. 또한, 최종 예측치가 산출되면 손실 함수를 통해 실제값과 예측치 간의 차이를 측정하는데, 이때 손실 함수 역시도 분류 문제에 따라 이진 분류에서는 binary cross-entropy, 다중 분류에서는 categorical cross-entropy[75] 손실 함수가 사용된다. 마지막으로 매개변수(parameter) 학습을 위해 수행되는 역전파(Backpropagation) 과정은 메모리 셀 시점마다 역전파를 수행하는 BPTT(Backpropagation Through Time) 방식이 수행되며, 이때 학습되는 매개변수는 가중치 $\mathbf{W}_x$, $\mathbf{W}_h$, $\mathbf{W}_y$와 편향 $\mathbf{b}_h$, $\mathbf{b}_y$이다.

---

73) tanh는 원점을 기준으로 대칭성을 갖는 S자 형태의 비선형 함수로서 RNN에 비선형성을 부여하여 복잡한 패턴과 시간적 의존성을 학습할 수 있도록 도와주는 역할을 한다. tanh의 출력값은 −1~1 사이이며, 수식적 정의는 다음과 같다.

$$\tanh(x) = (e^{x} - e^{-x})/(e^{x} + e^{-x})$$

74) 두 함수의 수식은 다음과 같다.

$$sigmoid(x) = 1/(1+e^{-z_t}) = \hat{y_t}$$

$$softmax(z_{t,i}) = e^{z_{t,i}} / \sum_{j=1}^{K} e^{z_{t,j}} = \hat{y}_{t,i}$$

(여기서 $z_{t,i}$는 레이블 i에 대한 출력 로짓(logit)을 의미한다.)

75) 두 함수의 수식은 다음과 같다.

$$L_t = -\left[y_t \log(\hat{y_t}) + (1-y_t)\log(1-\hat{y_t})\right]$$

(여기서 y_t는 0 또는 1의 값을 갖는 실제 레이블이며, $\hat{y_t}$는 0~1 사이의 값을 갖는 예측확률이다.)

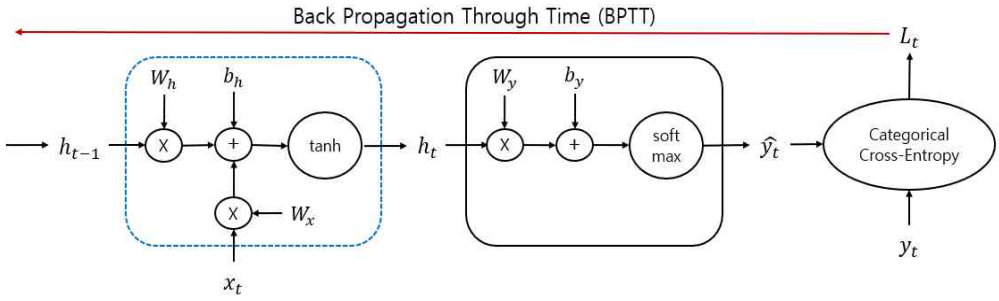
$$L_t = -\sum_{c=1}^{C} y_{t,c} \log(\hat{y}_{t,c})$$

(여기서 C는 레이블의 개수, $\hat{y}_{t,c}$는 레이블 c에 대한 모델의 예측확률, $y_{t,c}$는 레이블 c에 해당하는 원-핫 인코딩(One-Hot Encoding) 벡터의 요소이다.)

(3-20) $\mathbf{h}_t = f(\mathbf{W}_h\mathbf{h}_{t-1} + \mathbf{W}_x\mathbf{x}_t + \mathbf{b}_h)$

(3-21) $\hat{\mathbf{y}}_t = g(\mathbf{W}_y\mathbf{h}_t + \mathbf{b}_y)$

[그림 3-8] RNN 연산 과정



RNN의 역전파 연산 과정은 [그림 3-9]와 같다. 역전파 연산에서 가장 먼저 등장하는 것은 손실 함수에 대한 $\hat{\mathbf{y}}_t$의 기울기(gradient) $\partial L/\partial\hat{\mathbf{y}}_t$이다. 이는 소프트맥스 함수($\hat{y}_{t,i} = e^{z_{t,i}}/\Sigma_{j=1}^{K} e^{z_{t,j}}$)와 Categorical Cross-Entropy 손실 함수($L_t = -\Sigma_{c=1}^{C} y_{t,c}\log(\hat{y}_{t,c})$)를 사용하는 경우, 예측치와 실제값의 차이로 계산되며[76] 그림에서는 이를 편의상 $\partial\hat{\mathbf{y}}_t$로 표현하였다. $\partial\hat{\mathbf{y}}_t$는 덧셈 노드를 따라 양방향에 그대로 전달되어 출력층의 편향 $\mathbf{b}_y$의 기울기($\partial L/\partial\mathbf{b}_y$)를 산출한다. 그리고 곱셈 노드를 지나면 출력층의 또 다른 매개변수 $\mathbf{W}_y$의 기울기가 $\partial\hat{\mathbf{y}}_t$와 은닉 상태 $\mathbf{h}_t$의 외적(outer product)으로 산출되며, $\partial\hat{\mathbf{y}}_t$와 $\mathbf{W}_y$의 외적이 t 시점의 메모리 셀로 전달된다. 메모리 셀에서는 전달받은 $\partial\mathbf{h}_t$가 tanh 노드를 지나면서 tanh 함수의 미분 값($(1-\tanh^2(\mathbf{h}_{raw}))$)과 내적이 수행되는데, 여기서

76) 연쇄 법칙(Chain Rule)을 활용하여 손실 함수 L에 대한 소프트 맥스 함수의 로짓(logit) $z_{t,i}$의 편미분을 취하면 다음과 같이 계산된다.

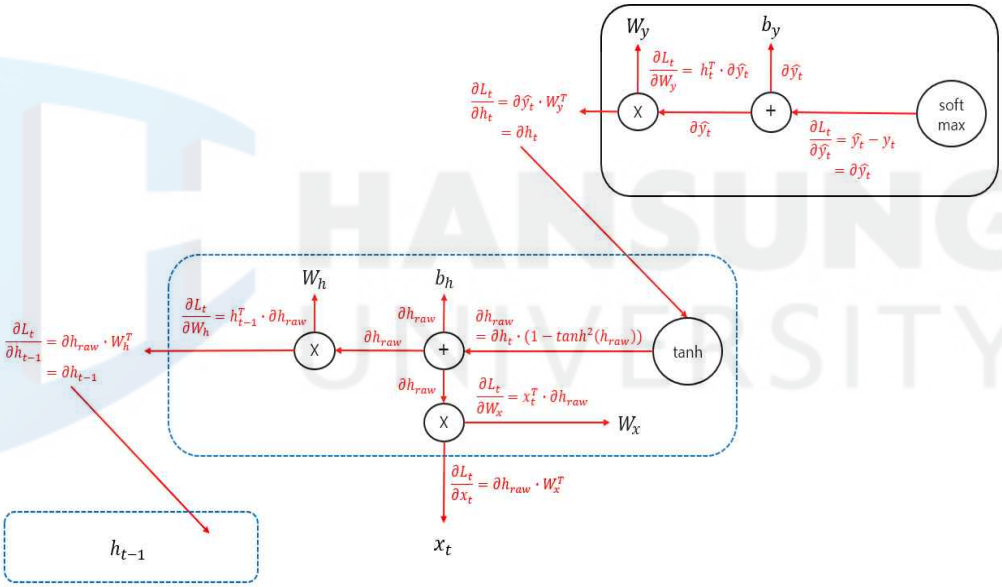
$$\frac{\partial L}{\partial z_{t,i}} = \frac{\partial L}{\partial \hat{y}_{t,i}} \times \frac{\partial \hat{y}_{t,i}}{\partial z_{t,i}} \text{ , where } \frac{\partial L}{\partial \hat{y}_{t,i}} = -\frac{y_{t,i}}{\hat{y}_{t,i}}, \frac{\partial \hat{y}_{t,i}}{\partial z_{t,i}} = \hat{y}_{t,i}(1-\hat{y}_{t,i})$$

$$\frac{\partial L}{\partial z_{t,i}} = -\frac{y_{t,i}}{\hat{y}_{t,i}} \times \hat{y}_{t,i}(1-\hat{y}_{t,i})$$

$$\frac{\partial L}{\partial z_{t,i}} = \hat{y}_{t,i} - y_{t,i}$$

$\mathbf{h}_{raw}$는 tanh 함수를 적용하기 전의 선형 결합 값, 즉 $\mathbf{W}_h\mathbf{h}_{t-1}+\mathbf{W}_x\mathbf{x}_t+\mathbf{b}_h$ 이다. 그리고 산출된 $\partial\mathbf{h}_{raw}$는 덧셈 노드를 따라 세 방향으로 전달되어 평향 $\mathbf{b}_h$의 기울기($\partial L/\partial\mathbf{b}_h$)를 산출하고, 아래쪽 곱셈 노드를 지나 $\partial\mathbf{h}_{raw}$와 입력 벡터 $\mathbf{x}_t$의 외적으로 매개변수 $\mathbf{W}_x$의 기울기를 산출한다. 마지막으로 $\partial\mathbf{h}_{raw}$가 왼쪽 곱셈 노드로 넘어가면 전 시점의 은닉 상태 $\mathbf{h}_{t-1}$과 외적을 수행하여 $\mathbf{W}_h$의 기울기를 산출하고, $\partial\mathbf{h}_{raw}$와 $\mathbf{W}_h$의 외적을 이전 시점의 메모리 셀로 전달하여 위 과정을 반복하게 된다.

[그림 3-9] RNN 역전파 연산 과정



위 그림에도 나와 있듯이, RNN의 역전파 과정은 tanh와 2번의 행렬 곱(dot)이 연속으로 수행되기 때문에 시점이 길어질수록 기울기가 0에 가까워지는 기울기 소실(Gradient Vanishing) 문제가 발생한다. 특히, tanh는 출력값이 −1~1 사이에 값을 가지므로 입력값이 매우 크거나 작은 경우에는 역전파 시점이 몇 단계만 지나도 미분 값이 0에 가까워져 버린다. 따라서 RNN 모형을 설계할 때는 tanh 대신 ReLU(Rectified Linear Unit)[77]로 활성화 함

77) ReLU 함수는 음수 값을 0으로 만들지만, 양수 값에 대해서는 미분 값을 1로 유지하여 기울기 소실 문제를 줄여준다.

수를 변경하거나, 메모리 셀의 연결을 적당한 길이로 나누어 역전파를 수행하는 Truncated-BPTT(Truncated-BackPropagation Through Time)를 활용해 기울기 소실 문제에 대응하게 된다. 그러나 이 역시도 완전한 해결책은 되지 못하기 때문에 기울기 소실 문제는 RNN의 주요 한계로 여겨지고 있다.

기울기 소실의 가장 큰 문제는 기울기가 작아지면서 과거에 학습했던 정보도 함께 사라진다는 것이다. 즉, 앞서 학습했던 정보가 뒤로 충분히 전달되지 못해 모형이 장기 의존성(Long-Term Dependency)을 학습하지 못하는 문제가 발생한다.[78] 이로 인해 RNN은 비교적 짧은 길이의 시퀀스에 대해서만 효과를 본다는 한계가 지적되었고, 결국 이를 해결하기 위한 RNN의 다양한 변형들이 나오게 되었다.

2) LSTM(Long Short-Term Memory)

LSTM(Hochreiter and Schmidhuber, 1997)은 RNN의 장기 의존성 문제를 해결하기 위해 제안된 특수한 구조의 RNN이다. LSTM의 핵심은 기존의 은닉 상태 외에 셀 상태(cell state)라는 장기기억 장치를 추가한 데 있다. 셀 상태는 일부 선형적인 상호작용(linear interaction)만을 수행하여 정보가 특별한 변화 없이 그대로 흘러가도록 하는 컨베이어 벨트와 같은 역할을 한다. 즉, 기울기 소실의 주된 원인이었던 활성화 함수를 거치지 않아 역전파 과정 시 정보 손실이 거의 발생하지 않는 것이 특징이다. 하지만 이처럼 모든 시점의 정보를 그대로 누적하기만 한다면 불필요한 정보도 함께 축적되어 핵심 정보의 전달이 약화될 수 있다. 이를 보완하기 위해 LSTM은 셀 상태의 불필요하거나 중요한 정보를 조절할 수 있도록 망각 게이트(forget gate), 입력 게이트(input gate), 출력 게이트(output gate)라는 3가지 메커니즘을 사용한다.

먼저, 망각 게이트는 셀 상태의 과거 정보를 삭제하는 역할을 한다([그림 3-10]). 이는 아래 식 (3-22)를 따라 현재 시점의 입력 $\mathbf{x}_t$와 이전 시점의 은닉 상태 $\mathbf{h}_{t-1}$를 시그모이드 함수에 통과시켜 출력값 $\mathbf{f}_t$를 생성한 뒤, 이를 이

78) 예를 들어 문장 처음과 끝에 등장한 두 단어가 서로 연관되어 있다면 모형은 이 둘 사이의 의존성을 학습해야 하는데, 기울기 소실은 이 의존성 학습에 필요한 정보를 잃어버리게 만든다.

전 시점의 셀 상태 $\mathbf{c}_{t-1}$과 곱하여 수행된다. 이때 $\mathbf{f}_t$는 시그모이드 함수로 인해 0~1 사이의 값을 가지며, 0에 가까울수록 이전 시점의 입력 정보가 더 많이 제거됨을 의미한다.

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \quad (3\text{-}22)$$

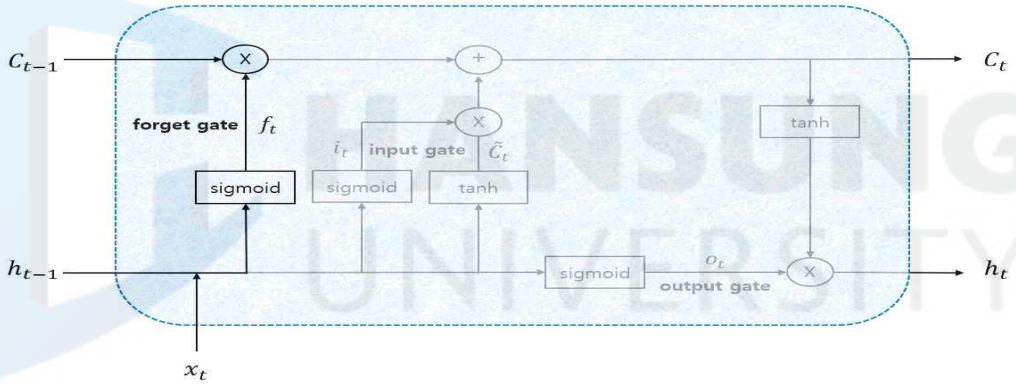
$\mathbf{h}_{t-1}$, $\mathbf{x}_t$: 이전 시점의 은닉 상태, 단어 임베딩 벡터

$\mathbf{W}_f$, $\mathbf{b}_f$: 망각 게이트의 가중치와 편향

σ: 시그모이드 함수

$\mathbf{f}_t$: 망각 게이트의 출력

[그림 3-10] 망각 게이트



다음으로, 입력 게이트는 현재 시점의 입력 정보를 조절하는 역할을 한다([그림 3-11]). 이는 식 (3-23)과 (3-24)에 의해 수행되며, 여기서 $\mathbf{i}_t$는 입력 게이트의 출력을, $\tilde{\mathbf{c}}_t$는 셀 상태의 후보 값을 의미한다. $\mathbf{i}_t$와 $\tilde{\mathbf{c}}_t$는 모두 현재 시점의 입력 $\mathbf{x}_t$와 이전 시점의 은닉 상태 $\mathbf{h}_{t-1}$를 기반으로 계산되며, $\mathbf{i}_t$는 시그모이드 함수에 의해 0~1 사이의 값을, $\tilde{\mathbf{c}}_t$는 tanh 함수에 의해 −1~1 사이의 값을 출력한다. 그리고 두 값이 출력되면 이를 곱해 현재 시점의 입력 정보를 조절한다. 따라서 $\mathbf{i}_t$는 1에 가까운 값을 가질수록 현재 시점의 입력 정보를 많이 반영한다는 의미를 지닌다. 또한, 입력 게이트의 결과값이 산출된 후에는 이를 삭제 게이트의 결과값과 더해 현재 시점의 셀 상태 $\mathbf{c}_t$를 산출한

다. 이는 식 (3-25)와 같이 정의되며, 여기서 ∘는 요소별 곱(element-wise product)을 의미한다.

$$(3\text{-}23)\ \mathbf{i}_t = \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i)$$

$$(3\text{-}24)\ \tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_c)$$

$$(3\text{-}25)\ \mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tilde{\mathbf{c}}_t$$

$\mathbf{h}_{t-1}$, $\mathbf{x}_t$: 이전 시점의 은닉 상태, 단어 임베딩 벡터

$\mathbf{W}_i$, $\mathbf{b}_i$: 입력 게이트의 가중치와 편향

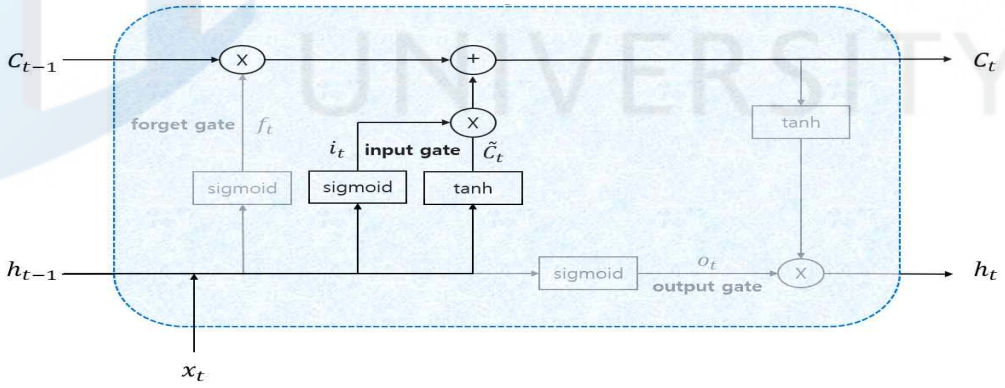
$\mathbf{W}_c$, $\mathbf{b}_c$: 셀 상태 후보 값의 가중치와 편향

σ, tanh: 시그모이드 함수, tanh 함수

$\mathbf{i}_t$, $\mathbf{f}_t$: 입력 게이트의 출력, 망각 게이트의 출력

$\tilde{\mathbf{c}}_t$, $\mathbf{c}_t$: 셀 상태의 후보 값, 셀 상태

[그림 3-11] 입력 게이트



마지막으로 출력 게이트는 은닉 상태의 출력을 위해 셀 상태의 정보를 조절하는 역할을 한다([그림 3-12]). 출력 게이트는 아래 식 (3-26)과 같이 정의되며, 이 역시 현재 시점의 입력 $\mathbf{x}_t$와 이전 시점의 은닉 상태 $\mathbf{h}_{t-1}$를 시그모이드 함수에 통과시켜 출력값을 생성한다. 망각 게이트와 입력 게이트를 통해 산출된 셀 상태 $\mathbf{c}_t$는 다음 시점으로 넘어가는 동시에 은닉 상태 산출을 위한 입력값으로 전달된다. 전달된 $\mathbf{c}_t$는 식 (3-27)과 같이 tanh 함수를 통해

−1~1 사이의 값으로 정규화되고, 이 값이 출력 게이트의 출력과 요소별 곱을 수행하여 은닉 상태가 된다. 그리고 이렇게 산출된 은닉 상태는 다음 시점의 메모리 셀로 전달되며, 다대다 구조에서는 출력층으로도 향하게 된다.[79]

$$(3\text{-}26)\ \mathbf{o}_t = \sigma(\mathbf{W}_o \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o)$$

$$(3\text{-}27)\ \mathbf{h}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}_t)$$

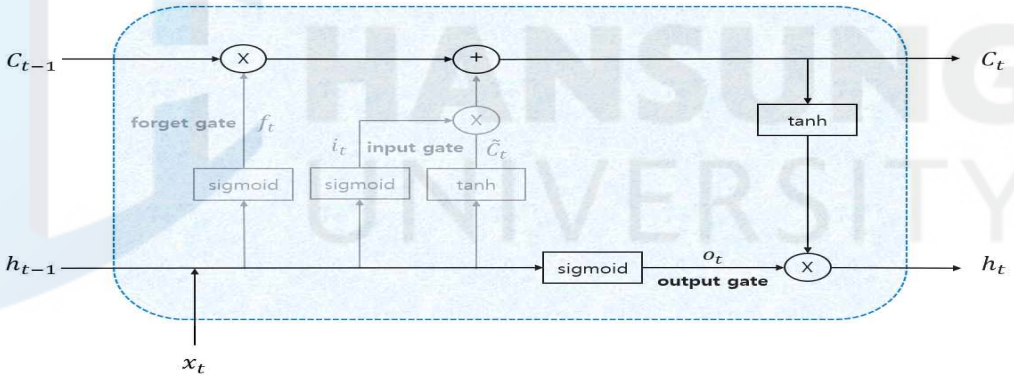
$\mathbf{h}_{t-1}$, $\mathbf{x}_t$: 이전 시점의 은닉 상태, 단어 임베딩 벡터

$\mathbf{W}_o$, $\mathbf{b}_o$: 출력 게이트의 가중치와 편향

σ, tanh: 시그모이드 함수, tanh 함수

$\mathbf{o}_t$, $\mathbf{c}_t$, $\mathbf{h}_t$: 출력 게이트의 출력, 셀 상태, 은닉 상태

[그림 3-12] 출력 게이트



3) GRU(Gated Recurrent Unit)

GRU는 Cho et al.(2014)에서 제안된 인공 신경망으로, LSTM의 장기 의존성 학습 능력은 유지하면서 계산 효율성은 높이기 위해 개발되었다. GRU의 핵심은 기존 LSTM에서 사용되던 셀 상태를 제거하고, 3가지 게이트 구조를 리셋 게이트(reset gate)와 업데이트 게이트(update gate)로 축소하여 은닉 상태에 대한 계산 과정을 좀 더 효율적으로 간소화시킨 것이다.

79) 주석 73)에서도 언급하였듯이 다대다 구조에서는 은닉 상태가 출력층으로도 향한다.

먼저, 리셋 게이트는 은닉 상태의 과거 정보를 제거하기 위한 게이트로, 아래 식 (3-28)과 (3-29)에 의해 작동된다([그림 3-13]). 구체적으로, 리셋 게이트의 출력은 현재 시점의 입력 $\mathbf{x}_t$와 이전 시점의 은닉 상태 $\mathbf{h}_{t-1}$를 시그모이드 함수에 통과시켜 산출되며, 출력값 $\mathbf{r}_t$는 $\mathbf{h}_{t-1}$과 요소별 곱을 수행하여 $\tilde{\mathbf{h}}_{t-1}$를 산출한다. 여기서는 $\tilde{\mathbf{h}}_{t-1}$를 리셋 값이라 표현하였으며, 이 $\tilde{\mathbf{h}}_{t-1}$은 업데이트 게이트로 전달되어 은닉 상태의 후보 값을 계산하는 데 사용된다.

$$(3\text{-}28)\ \mathbf{r}_t = \sigma(\mathbf{W}_r \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_r)$$

$$(3\text{-}29)\ \tilde{\mathbf{h}}_{t-1} = \mathbf{r}_t \circ \mathbf{h}_{t-1}$$

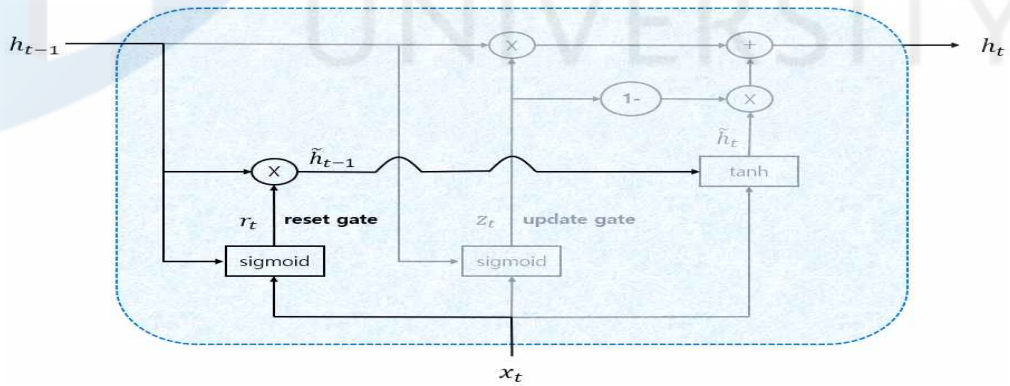
$\mathbf{h}_{t-1}$, $\mathbf{x}_t$: 이전 시점의 은닉 상태, 단어 임베딩 벡터

$\mathbf{W}_r$, $\mathbf{b}_r$: 리셋 게이트의 가중치와 편향

σ: 시그모이드 함수

$\mathbf{r}_t$, $\tilde{\mathbf{h}}_{t-1}$: 리셋 게이트의 출력, 리셋 값

[그림 3-13] 리셋 게이트



리셋 게이트가 과거 정보를 얼마나 반영할지를 조절하였다면, 업데이트 게이트는 과거 정보와 현재 정보의 조합 비율을 제어한다([그림3-14]). 이에 대한 구체적인 연산 과정은 아래 식 (3-30), (3-31), (3-32)와 같다. 우선, 식 (3-30)을 살펴보면, 업데이트 게이트 역시 다른 게이트들과 마찬가지로 현재 시점의 입력 $\mathbf{x}_t$와 이전 시점의 은닉 상태 $\mathbf{h}_{t-1}$를 시그모이드 함수에 통과시

켜 출력값 $\mathbf{z}_t$를 생성한다. 그리고 이 $\mathbf{z}_t$와 요소별 곱을 수행하는 값은 이전 시점의 은닉 상태 $\mathbf{h}_{t-1}$과 은닉 상태의 후보 값 $\tilde{\mathbf{h}}_t$이다. $\tilde{\mathbf{h}}_t$는 식 (3-31)와 같이 리셋 값 $\tilde{\mathbf{h}}_{t-1}$과 $\mathbf{x}_t$를 tanh 함수에 통과시켜 산출되며, 이 과정에서 GRU는 과거 정보와 현재 입력을 결합한 새로운 정보를 생성한다. 또한, 은닉 상태 $\mathbf{h}_t$는 식 (3-32)를 따라 $\mathbf{h}_{t-1}$는 $\mathbf{z}_t$와 $\tilde{\mathbf{h}}_t$는 $(1-\mathbf{z}_t)$와 요소별 곱을 수행하여 산출된다. 이는 업데이트 게이트가 1에 가까운 값을 갖는다면 새로운 정보보다 과거 정보를 중요시하고, 반대로 0에 가까운 값을 갖는다면 과거 정보보다 새로운 정보를 더 중요시한다는 것을 의미한다.

$$\mathbf{z}_t = \sigma(\mathbf{W}_z \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_z) \quad (3-30)$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_h \cdot [\tilde{\mathbf{h}}_{t-1}, \mathbf{x}_t] + \mathbf{b}_h) \quad (3-31)$$

$$\mathbf{h}_t = \mathbf{z}_t \circ \mathbf{h}_{t-1} + (1-\mathbf{z}_t) \circ \tilde{\mathbf{h}}_t \quad (3-32)$$

$\mathbf{h}_{t-1}$, $\mathbf{x}_t$, $\tilde{\mathbf{h}}_{t-1}$: 이전 시점의 은닉 상태, 단어 임베딩 벡터, 리셋 값

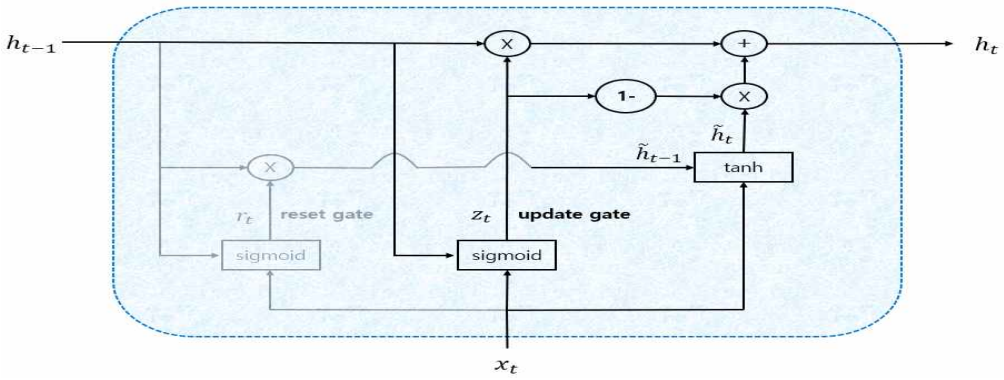
$\mathbf{W}_z$, $\mathbf{b}_z$: 업데이트 게이트의 가중치와 편향

$\mathbf{W}_h$, $\mathbf{b}_h$: 은닉 상태 후보 값의 가중치와 편향

σ, tanh: 시그모이드 함수, tanh 함수

$\mathbf{z}_t$, $\tilde{\mathbf{h}}_t$, $\mathbf{h}_t$: 업데이트 게이트의 출력, 은닉 상태의 후보 값, 은닉 상태

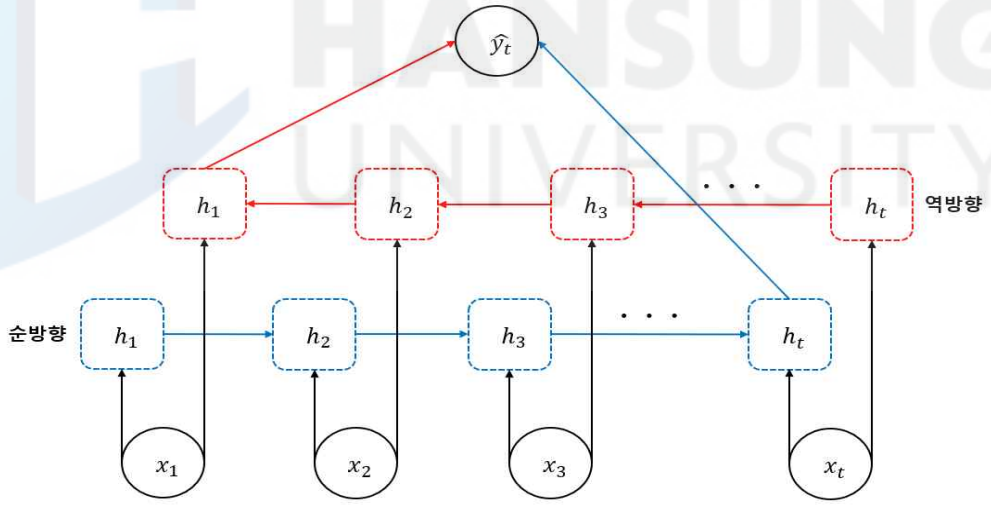
[그림 3-14] 업데이트 게이트



4) BiRNN(Bidirectional Recurrent Neural Network)

BiRNN(Schuster & Paliwal, 1997)은 순차 데이터의 양방향 문맥 정보를 반영하기 위해 제안된 구조로, 과거 시점의 입력뿐만 아니라 미래 시점의 입력까지 예측에 기여할 수 있다는 아이디어에 기반한다. BiRNN의 핵심은 아래 [그림 3-15]와 같이 같은 입력에 대해 두 개의 메모리 셀을 사용한다는 것이다. 즉, 첫 번째 메모리 셀은 앞 시점의 은닉 상태(Forward States)를 전달받아 현재의 은닉 상태를 계산하고, 두 번째 메모리 셀은 뒤 시점의 은닉 상태(Backward States)를 전달받아 현재의 은닉 상태를 계산하는 구조다. 이때 두 방향의 은닉 상태는 각각 독립적으로 학습되며, 두 방향의 은닉 상태가 산출된 후에는 이를 모두 출력층으로 보내 최종 예측에 사용한다.

[그림 3-15] BiRNN 구조도[1)]



주: 1) 감정 분류 문제에 사용되는 다대일 구조의 BiRNN을 나타냄

BiRNN의 양방향 구조는 LSTM과 GRU에도 동일하게 적용할 수 있다. 이 경우, 전자를 BiLSTM(Bidirectional Long Short-Term Memory), 후자를 BiGRU(Bidirectional Gated Recurrent Unit)라 부른다. BiLSTM과 BiGRU는 장기 의존성 문제를 처리함과 동시에 과거 정보와 미래 정보까지 함께 고려할 수 있다는 장점이 있다. 특히, 텍스트 데이터의 전체 맥락을 이해하는 것

이 중요한 경우에는 앞뒤 문맥을 파악할 수 있는 양방향 구조가 유리하게 작용한다. 그러나 반대로 단방향 구조가 양방향 구조보다 더 나은 선택이 되는 경우도 존재한다. 예컨대, 입력이 실시간으로 주어지거나 미래 정보가 존재하지 않는 경우(예: 주가 뉴스, 시장 예측 등), 또는 정보가 순차적으로 전개되거나 핵심 정보가 문장 앞에 위치하는 경우[80]에는 초기 정보에 높은 가중치를 부여하는 단방향 구조가 더 나은 성능을 나타낼 수도 있다. 따라서 양방향 구조인 BiLSTM과 BiGRU가 단방향 구조인 LSTM과 GRU보다 항상 우수한 성능을 보인다고 일반화할 수는 없으며, 분석 데이터의 특성과 연구 목적에 적합한 모형을 설계하기 위해서는 각 아키텍처의 성능을 비교·평가하는 과정이 필요하다.

5) 어텐션 메커니즘(Attention Mechanism)

앞서 소개된 RNN 모형들로 감정 분류 모형을 구축할 수도 있지만, 여기에 어텐션 메커니즘을 추가하면 해당 모형의 성능을 더 높일 수 있다. 어텐션은 Bahdanau et al.(2014)에서 처음 등장한 후, Luong et al.(2015)에 의해 정립된 아이디어로, 최근 대세 모듈로 사용되고 있는 트랜스포머의 기반이 되는 아이디어다.

어텐션의 본래 목적은 RNN 기반의 Seq2Seq(Sequence to Sequence) 모형[81]의 성능을 높이는 것이었다. Seq2Seq 모형은 입력을 받아 처리하는 인코더와 출력을 생성하는 디코더로 구성되는데, RNN 기반의 Seq2Seq 모형은 인코더(RNN 셀)에서 전체 입력 시퀀스를 하나의 벡터로 압축하여 표현하고, 디코더(RNN 셀)는 이를 입력으로 받아 출력 시퀀스를 생성하는 구조를 가졌다. 즉, 인코더 RNN 셀의 마지막 시점의 은닉 상태가 디코더 RNN 셀의 첫 번째 시점의 은닉 상태로 사용되는 구조다. 그러나 이처럼 전체 입력 시퀀스

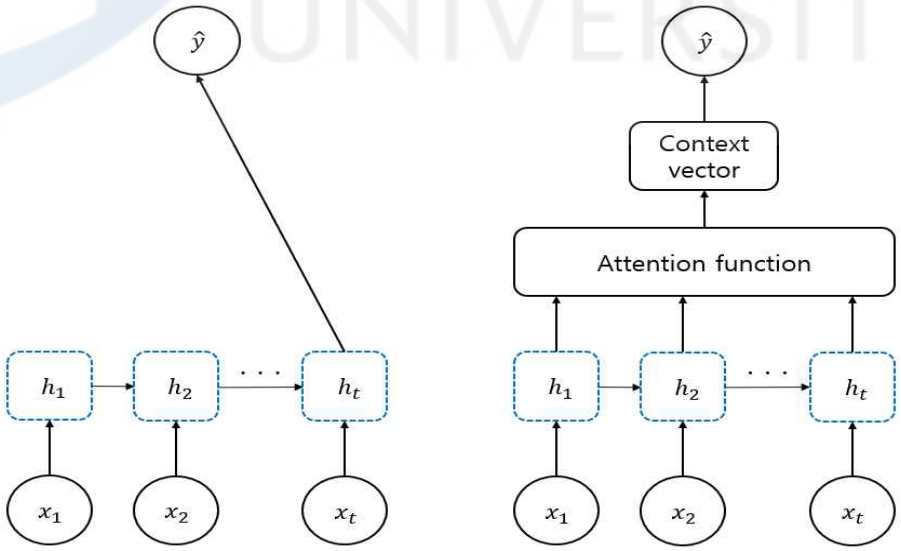
80) 예를 들어 “한국경제는 지난 분기에 예상보다 강한 성장을 기록했으며, 이는 소비자 지출 증가와 수출 호조에 기인한다.”와 같은 문장은 경제 성장이라는 주요 사실을 먼저 인지한 후, 소비자 지출과 수출 증가라는 세부 정보를 순차적으로 나타낸다.

81) Seq2Seq 모형은 기계 번역(machine translation)이나 챗봇(chatbot)과 같이 입력 시퀀스로부터 다른 도메인의 시퀀스를 출력하는 분야에서 사용되는 모형이다.

를 하나의 고정된 길이의 벡터로 압축하면 일부 입력 시퀀스의 정보가 사라지기 때문에 시퀀스 길이가 길어질수록 모형의 성능이 저하되는 단점이 있다. 어텐션은 이를 보완하기 위해 디코더가 출력을 예측하는 시점마다 인코더의 전체 시퀀스를 참고하는 동시에 해당 시점에서 예측해야 할 단어와 연관이 있는 입력 단어는 좀 더 집중(attention)해서 참고한다는 아이디어를 갖는다.

아래 [그림 3-16]은 기존 RNN 모형(좌측)과 어텐션 기반 RNN 모형(우측)의 차이를 도식화한 것이다. 그림에서 보이듯이, 기존 RNN 모형은 마지막 시점의 은닉 상태로 예측을 수행하는 반면, 어텐션 기반 RNN 모형은 어텐션 함수(attention function)로 산출된 문맥 벡터(context vector)를 활용하여 예측을 수행한다. 특히 주목할 점은 어텐션 함수가 마지막 시점의 은닉 상태가 아닌, 모든 시점의 은닉 상태를 입력으로 받는다는 것이다. 즉, 마지막 시점의 은닉 상태는 시점을 지나면서 일부 정보가 손실된 상태이고 이 손실된 정보 중에는 유용한 정보들이 포함될 수 있으니 모든 시점의 은닉 상태를 다시 한번 참고하여 예측을 수행하자는 것이 어텐션의 아이디어다.[82)]

[그림 3-16] 어텐션 기반 RNN 모형



82) 참고로 여기서는 감정 분류 모형을 가정하여 디코더 부분은 생략했지만, Seq2Seq 모형에서는 문맥 벡터가 디코더에 전송되고 디코더는 이 문맥 벡터를 통해 해당 시점의 출력을 예측한다.

어텐션 함수는 Vaswani et al.(2017)를 따라 아래 식 (3-33)과 같이 표현될 수 있다. 여기서 어텐션 함수의 입력인 쿼리(query), 키(key), 밸류(value)는 모형 설계에 따라 다른 의미를 가질 수 있으나, 일반적인 Seq2Seq 모형에서는 쿼리가 디코더 셀 t시점의 은닉 상태를, 키와 밸류가 인코더 셀 모든 시점의 은닉 상태를 의미한다. 보다 일반적으로 정의한다면 쿼리는 현재 처리하고 있는 타깃을 표현하는 벡터를, 키는 비교 대상이 되는 입력 시퀀스의 각 요소를 표현하는 벡터를 의미하며, 밸류는 키와 연관된 데이터라 정의할 수 있다. 그리고 어텐션 함수를 통해 산출된 어텐션 값(attention value)은 [그림 3-16]의 문맥 벡터를 의미한다.

$$(3\text{-}33)\quad Attention(\mathbf{query},\mathbf{key},\mathbf{value}) = \mathbf{attention\ \ value}$$

어텐션 함수의 구체적인 연산 과정은 다음과 같은 세 단계로 나눌 수 있다. 먼저, 첫 번째 단계에서는 어텐션의 핵심이라 할 수 있는 어텐션 스코어를 계산한다. 어텐션 스코어는 주어진 쿼리에 대해 각 키가 얼마나 관련 있는지를 평가한 점수를 의미한다. 즉, 쿼리와 각 키의 유사도를 말하며, 이 유사도를 측정하는 방식은 다음 식 (3-34)에 제시된 스코어 함수의 종류에 따라 달라진다.

$$(3\text{-}34)\quad Score(\mathbf{query},\mathbf{key}) = \begin{cases} \mathbf{query}^{\top}\mathbf{key} & dot \\ \dfrac{\mathbf{query}^{\top}\mathbf{key}}{\sqrt{n}} & scaled\,dot \\ \mathbf{query}^{\top}\mathbf{W}\mathbf{key} & general \\ \mathbf{v}^{\top}\tanh(\mathbf{W}[\mathbf{query};\mathbf{key}]) & concat1 \\ \mathbf{v}^{\top}\tanh(\mathbf{W}_a\mathbf{query}+\mathbf{W}_b\mathbf{key}) & concat2 \end{cases}$$

식 (3-34)에서 *dot*은 단순히 쿼리와 키의 내적으로 유사도를 측정하는 방식이고, *scaled dot*은 *dot*의 내적 값이 커지는 것을 방지하기 위해 키 벡터의 차원 수인 n의 제곱근을 나누어주는 방식이다. 또한, *general*은 쿼리와 키 이외에 학습 가능한 가중치 행렬 $\mathbf{W}$를 함께 사용하는 방식이며, *concat*1은 가중치 행렬 $\mathbf{W}$ 이외에 학습 가능한 가중치 벡터 $\mathbf{v}$를 함께 사용하는 방식이다. 마지

막으로 *concat*2는 *concat*1을 약간 변형한 버전으로, *concat*1이 쿼리와 키를 연결(concatenate)하여 가중치 행렬을 학습한 것과 달리, *concat*2는 쿼리와 키에 따라 다른 가중치 행렬을 학습한다는 것에 차이가 있다. 본 연구에서는 이 중 *concat*2를 활용하여 어텐션 함수를 구현하였으며, 이 경우 스코어 값의 구체적인 산출 방법은 다음 식 (3-35)와 같다.

$$(3\text{-}35)\ e_t = \mathbf{v}^\top \tanh(\mathbf{W}_a\mathbf{s} + \mathbf{W}_b\mathbf{h}_t)$$

여기서 $\mathbf{s}$는 쿼리에 해당하는 마지막 시점의 은닉 상태를 의미하며, $\mathbf{h}_t$는 키에 해당하는 t시점의 은닉 상태를 의미한다. 만약 GRU로 모형을 설계하였다면 $\mathbf{s}$와 $\mathbf{h}_t$는 앞서 제시된 식 (3-32)를 따라 결정된다. 그리고 여기서는 감정 분류 모형을 가정하였지만 Seq2Seq 모형에서는 쿼리에 해당하는 $\mathbf{s}$가 디코더의 $t-1$ 시점의 은닉 상태를 의미한다는 것에 주의해야 한다.

다음으로 두 번째 단계에서는 식 (3-35)를 통해 계산된 모든 은닉 상태의 어텐션 스코어에 대해 소프트맥스를 적용하여 어텐션 분포를 구한다. 이는 식 (3-36)과 같이 정의되며, 여기서 산출된 어텐션 분포의 각각의 값을 어텐션 가중치라 한다.

$$(3\text{-}36)\ \boldsymbol{\alpha} = softmax(\mathbf{e}),\ (\mathbf{e} = e_1, e_2, \cdots, e_t)$$

마지막으로 세 번째 단계에서는 쿼리와 키의 유사도를 나타내는 $\boldsymbol{\alpha}$를 키와 맵핑되어 있는 각각의 밸류에 반영한 후, 이를 모두 더하여 어텐션 값을 계산한다. 즉, 세 번째 단계는 다음 식 (3-37)과 같이 각각의 밸류($\mathbf{h}_t$)에 유사도가 반영되도록 가중합(weighted sum)을 진행하는 과정을 말하며, 이렇게 산출된 어텐션 값을 인코더의 전체 문맥을 포함하고 있다고 하여 문맥 벡터라 부른다.

$$(3\text{-}37)\ \mathbf{c} = \sum_t \boldsymbol{\alpha}_t \mathbf{h}_t$$

3.2.1.2 합성곱 신경망(Convolution Neural Network)

1) 합성곱 신경망 개요

CNN은 이미지 데이터의 처리, 즉 컴퓨터 비전 분야를 위해 설계된 다층 신경망이다. CNN은 초기 LaNet(LeCun et al., 1989)이 제시되었을 당시만 해도 계산 비용과 데이터 부족 등의 이유로 널리 사용되지는 않았으나, AlexNet(Krizhevsky et al., 2012)이라는 CNN 모형이 ILSVRC(ImageNet Large Scale Visual Recognition Challenge)[83]에서 압도적인 성능을 보여주면서 급속도로 발전되기 시작하였다. AlexNet의 성공은 VGGNet(Simonya and Zisserman, 2014), GoogLeNet(Szegedy et al., 2015), ResNet(He, 2016) 등 다양한 CNN 아키텍처들의 개발을 이끌었을 뿐만 아니라, CNN이 이미지 인식 이외에도 음성 인식, 자연어 처리, 의료 영상 분석 등 다양한 분야에서도 활용될 수 있는 계기를 만들어주었다. 특히, 텍스트 데이터에도 이미지 데이터와 같은 공간적, 시각적 정보가 있음이 발견되면서[84], Kim(2014), Zhang and Lecun(2015)와 같은 텍스트 처리를 위한 다양한 연구들이 시도되었고, 그 결과 텍스트 데이터에 대해서도 CNN이 유용한 특징을 추출할 수 있음이 증명되어 자연어 처리 분야에서도 널리 채택되어 사용되게 되었다.

CNN이 다양한 분야에 활용되면서 CNN은 처리하는 데이터의 차원에 따라 1D·2D·3D CNN으로 구분되어 사용되고 있다. 이때 데이터의 차원은 데이터 배열의 차원을 의미하는 것이 아니라 데이터가 가지는 공간적·시간적 차원을 말한다. 즉, 의료 영상과 같이 높이, 너비, 깊이(시간)의 3차원 형태를 가지는 데이터는 3D CNN을, 이미지 데이터와 같이 높이, 너비의 2차원 형태를 가지는 데이터는 2D CNN을 사용하고, 텍스트 데이터나 시계열 데이터처럼 1차원 특징을 갖는 데이터는 1D CNN을 사용한다. 따라서 자연어 처리

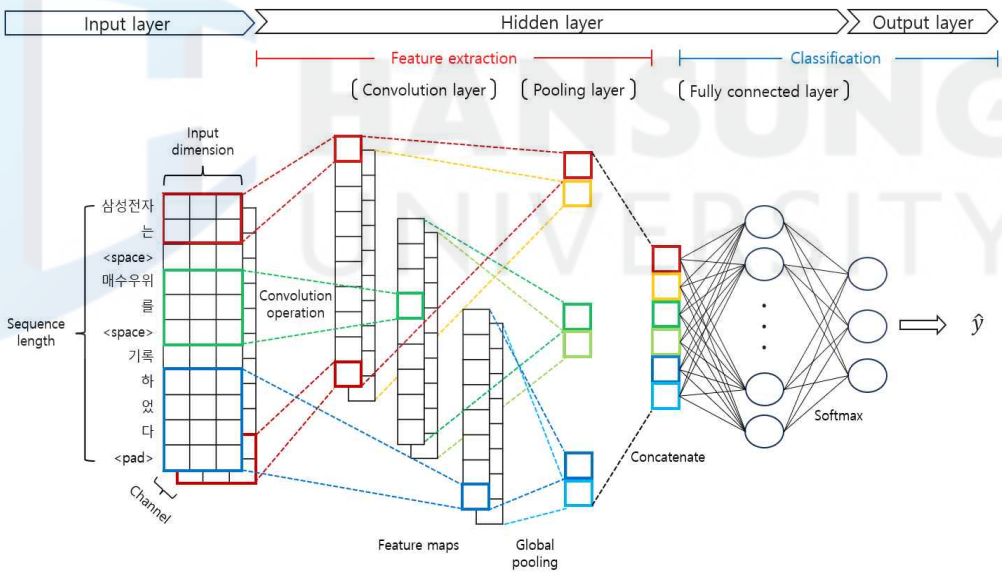
83) 2010년부터 시작된 컴퓨터 비전 분야에서 가장 영향력 있는 연례 경진 대회 중 하나이다.

84) 예를 들어 문장에서 단어들의 순서와 배열이 특정한 문맥을 형성하는 것은 이미지 데이터의 픽셀 배열과 유사한 공간적 구성을 이룬다. 또한 단어 임베딩 기술이 발전하면서 단어 사이의 의미적 관계를 수치화하는 것이 가능해졌고, 이는 CNN이 텍스트 데이터의 패턴을 인식하는 데 필요한 시각적 정보를 제공하게 하였다.

분야에서는 1D CNN을 활용하여 특정 단어의 조합이나 문맥적 관계를 학습하고, 이를 통해 감정 분류와 같은 작업을 수행하게 된다.

아래 [그림 3-17]은 1D CNN의 구조도를 보여준다. 해당 그림은 입력층, 은닉층, 출력층으로 구성되며, 은닉층은 합성곱층(convolution layer) 3개, 풀링층(pooling layer) 3개, 그리고 완전연결층(fully connected layer)이 1개로 설계된 구조이다. 여기서 합성곱층과 풀링층은 입력된 텍스트 데이터의 문맥적 특징이나 의미론적 패턴을 추출하는 특징 추출(feature extraction)의 역할을 하며, 완전연결층과 출력층은 추출된 특징을 분류(classification)하는 역할을 한다. 이하에서는 이러한 1D CNN 구조를 기준으로 각 계층이 어떻게 텍스트 데이터를 처리하는지 상세히 살펴보기로 한다.

[그림 3-17] 1D CNN 구조도



2) 입력층(Input Layer)

1D CNN은 (시퀀스의 길이, 임베딩 벡터의 차원)을 입력으로 받는다[85]. 이는 앞서 살펴본 RNN의 입력과 같은 형태이지만, 여기에 채널(channel)이

85) 여기서부터는 편의상 배치 크기는 1로 가정한다.

란 차원이 하나 더 추가될 수 있다는 점에서 차이가 있다. 채널은 본래 이미지 처리에서 이미지의 색상 정보를 나타내는 차원으로,[86] 일반적으로 텍스트 처리에서는 고려되지 않는 요소이다. 그러나 Kim(2014)은 위 [그림 3-17]과 같이 두 개의 채널이 있는 1D CNN을 활용하여 감정 분류 모형을 구축한 바 있다. Kim(2014)의 연구에서 활용된 멀티 채널 CNN은 각 채널에 서로 다른 임베딩 벡터를 사용하여 두 임베딩 벡터를 사전 훈련된 임베딩 벡터로 초기화한 후, 한 채널만 역전파로 학습하는 방식이다. 이러한 방식은 하나의 임베딩 벡터는 일반적인 의미를 유지하면서 다른 임베딩 벡터는 특정 태스크에 대해 미세 조정할 수 있다는 이점이 있다. 다만, Kim(2014)의 연구 결과에서는 싱글 채널 구조와 멀티 채널 구조의 성능이 서로 혼합되어 나왔으므로[87], 텍스트 데이터에 대해서는 멀티 채널 구조가 싱글 채널 구조보다 항상 성능이 뛰어난 것은 아님에 유의할 필요가 있다.

3) 합성곱층(Convolution Layer)

합성곱층에서는 CNN의 핵심이라 할 수 있는 합성곱 연산(convolution operation)이 수행된다. 합성곱 연산은 커널(kernel) 또는 필터(filter)라 불리는 작은 가중치 행렬을 이용하여, 입력된 데이터를 정해진 스트라이드(stride)에 따라 슬라이딩 윈도우 방식으로 국소적 특징(local feature)을 추출하고, 이를 특징맵(feature map)에 저장하는 과정이라 정의할 수 있다. 이를 좀 더 구체적으로 설명하기 위해 아래에는 합성곱 연산이 수행되는 4가지 예시를 [그림 3-18]~[그림 3-21]에 걸쳐 제시하였다.

첫 번째 예시인 [그림 3-18]은 4×3 크기의 입력 데이터와 3×3 크기의 커널이 1개씩 있는 경우의 합성곱 연산을 2단계로 나누어 보여준다. 해당 그림과 같이 커널은 입력된 데이터의 특징을 추출하기 위해 사용되는 가중치

86) 예를 들어 흑백 이미지의 채널 수가 1이면 컬러 이미지는 적색(red), 녹색(green), 청색(blue)의 삼원색 조합으로 이루어지므로 채널 수가 3이 된다.

87) Kim(2014)에서는 이러한 결과를 개선하기 위해 멀티 채널 대신 싱글 채널을 사용하고, 그 대신 초기화된 임베딩은 유지하면서 일부 임베딩에 대해서만 특정 작업에 미세 조정되도록 추가적인 연구가 필요하다고 언급하였다.

행렬을 의미하며, 특징맵은 이 커널이 입력 데이터 위를 이동하면서 계산한 내적들의 결과물을 의미한다. 그리고 빨간색 네모로 표현된 커널의 크기, 즉 윈도우가 일정 간격으로 미끄러지듯 이동하면서 연산을 진행하는 방식을 슬라이딩 윈도우라 표현하며, 이때 윈도우가 이동하는 간격을 스트라이드라 한다. 또한, 특징맵의 각 요소가 커널을 통해 '보고 있는' 빨간색 음영 부분을 수용 영역(receptive field)이라 표현하는데, 특징맵은 이 수용 영역을 커널의 크기로 조정해 국소적 특징을 저장하게 된다. 따라서 특징맵은 입력 크기, 커널 크기, 스트라이드에 의해 그 크기가 결정되며, 만약 스트라이드가 1이고, 입력 데이터와 필터의 크기가 각각 $n \times l$, $h \times l$ 이라면 특징맵은 아래 식 (3-38)[88]을 따라 $n-h+1$ 크기의 벡터가 된다. 마지막으로 여기서는 1D CNN을 가정하여 커널이 한 방향으로만 움직인다고 표현하였지만, 커널은 [그림 3-18] 하단에 나와 있는 것처럼 1D·2D·3D CNN에 따라 움직이는 방향이 달라질 수 있다. 즉, 1D·2D·3D에서 D는 커널이 움직이는 방향의 차원 수를 의미하며, 이로 인해 2D CNN과 3D CNN에서는 특징맵이 2차원 행렬과 3차원 볼륨 형태를 취하게 된다.[89] 따라서 1D CNN에서는 커널이 한 방향으로만 움직이므로 커널의 길이는 항상 입력 데이터의 길이와 똑같이 설정되며, 커널의 크기는 곧 커널의 높이를 의미하게 된다. 그리고 1D CNN은 이 커널의 높이를 조정해 n-gram의 국소적 특징을 추출하게 된다.[90]

$$(3\text{-}38)\quad O_h = \lfloor (I_h - K_h)/S \rfloor + 1, \quad O_l = \lfloor (I_l - K_l)/S \rfloor + 1$$

88) 참고로 식 (3-38)은 패딩(padding)을 고려하지 않은 식이다. 합성곱층에서 패딩은 특징맵의 크기를 조정하는 역할을 한다. 즉, 합성곱 연산이 수행되면 입력의 크기가 줄어들게 되므로 이 크기를 보존하기 위해 특징맵에 테두리를 추가하는 작업이다. 이때 특징맵의 테두리는 주로 제로 패딩(zerop padding)이 수행되어 0으로 값을 채우게 되며, 패딩이 수행된 합성곱을 넓은 합성곱(wide convolution), 패딩이 수행되지 않은 합성곱을 좁은 합성곱(narrow convolution)이라 부르게 된다. 패딩(P)을 고려한 특징맵의 크기는 아래 식과 같이 산출된다.

$$O_h = \left\lfloor \frac{(I_h - K_h + 2P)}{S} \right\rfloor + 1, \quad O_l = \left\lfloor \frac{(I_l - K_l + 2P)}{S} \right\rfloor + 1$$

89) 따라서 1D·2D·3D CNN은 엄밀히 말하면 입력 데이터의 형태에 따라 구분된다기보다는 커널의 진행 방향의 차원 수에 따라 구분된다고 할 수 있다.

90) 예를 들어 커널의 크기가 2이면 각 연산에서 참고하는 단어 묶음은 bigram이 되고, 커널의 크기가 3이면 각 연산에서 참고하는 단어 묶음은 trigram이 된다.

O_h, O_l: 특징맵의 높이와 길이

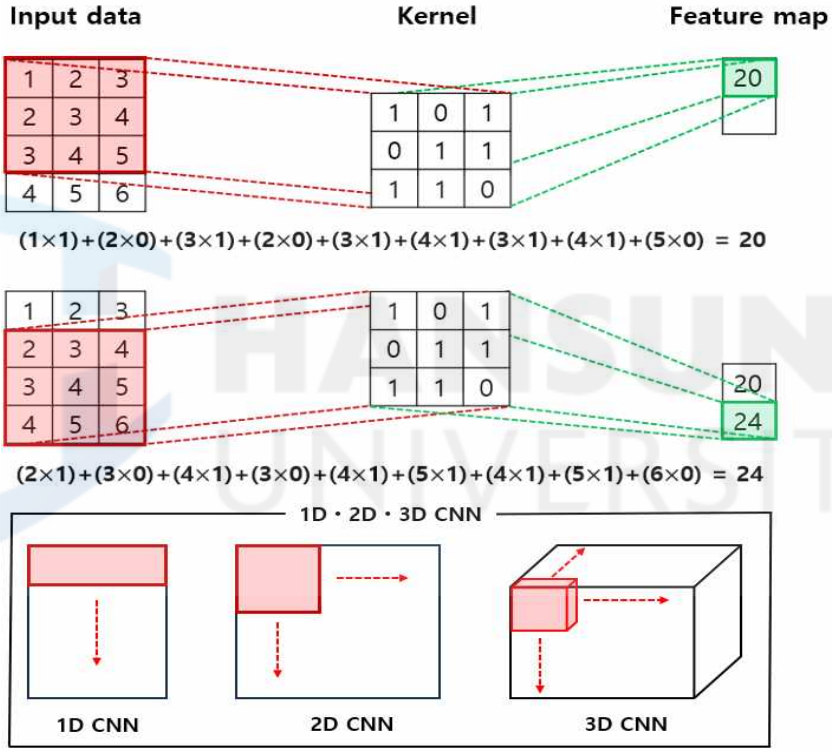
I_h, I_l: 입력의 높이와 길이

K_h, K_l: 커널의 높이와 길이

S: 스트라이드

$\lfloor\ \rfloor$: 바닥 함수(floor function)

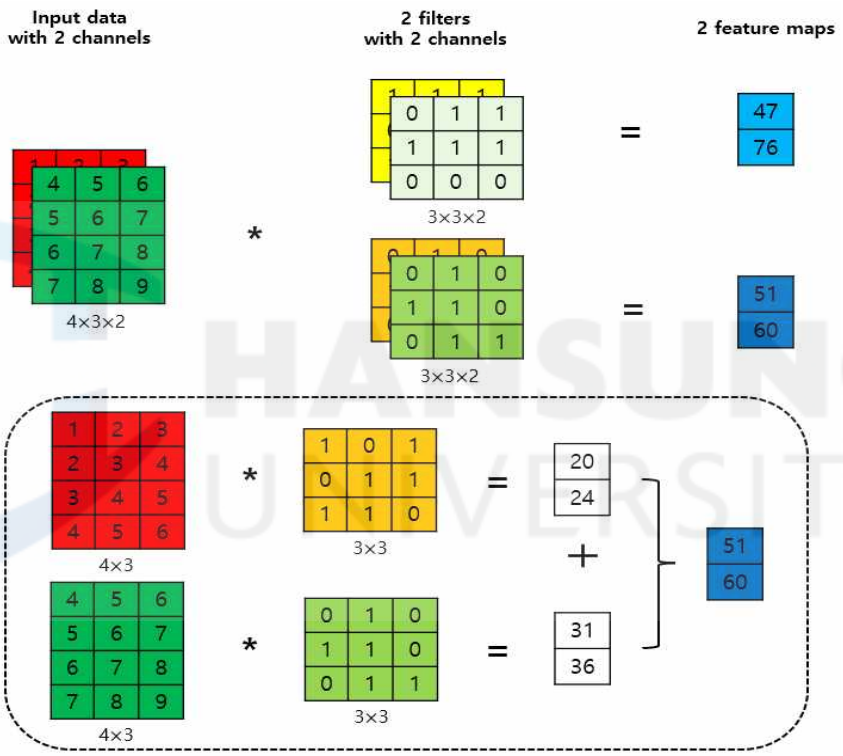
[그림 3-18] 합성곱 연산 예시1



두 번째 예시인 [그림 3-19]는 입력 데이터의 채널이 2개인 경우의 합성곱 연산을 보여준다. 입력 데이터가 다채널 구조를 가진 경우에는 커널의 개수도 입력 데이터의 채널 수와 똑같이 설정된다. 보통 커널과 필터는 혼용되어 사용되는 경우가 많으나, 엄밀히 말하면 필터는 커널이 2개 이상 사용된 커널들의 집합을 말하며 커널의 개수는 필터의 채널을 말한다. 즉, 아래 그림은 2개의 채널을 가진 3×3×2 크기의 필터가 2개 사용된 경우를 보여주며, 이때 특징맵은 입력 데이터와 필터의 각 채널끼리 합성곱 연산을 수행한 결

과를 요소별로 더하여 필터마다 하나의 특징맵을 생성한다. 따라서 필터의 채널(커널의 수)은 특징맵의 산출 방식에만 영향을 줄 뿐 특징맵의 수와 크기에는 영향을 주지 않으며, 특징맵의 수는 필터의 수에 의해 결정된다. 다시 말해 입력 데이터의 채널 수는 필터의 채널 수가 되며, 필터의 개수는 특징맵의 채널 수가 된다.

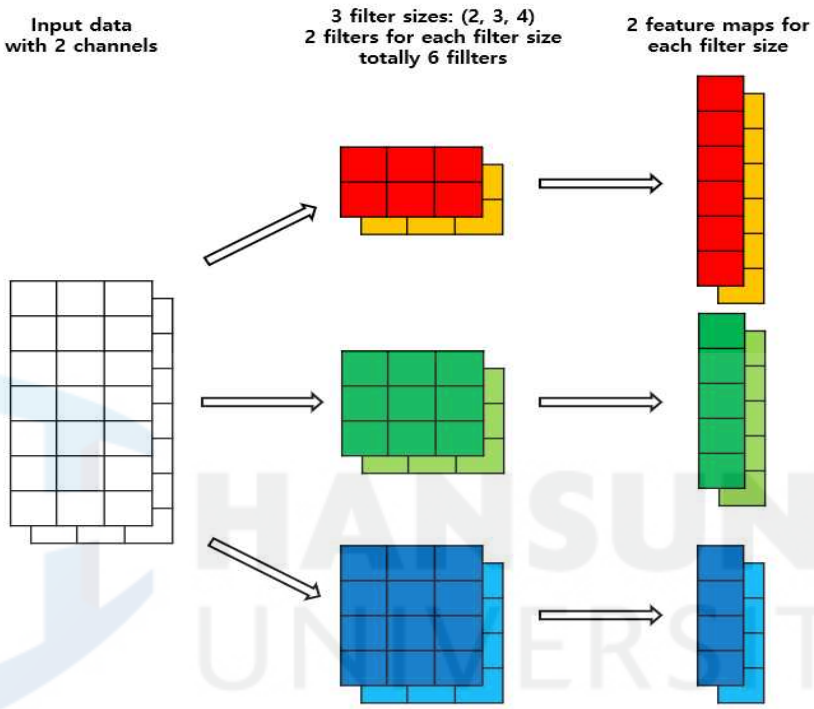
[그림 3-19] 합성곱 연산 예시2



세 번째 예시인 [그림 3-20]은 앞서 제시된 [그림 3-17]과 같이 크기가 다른 3개의 필터를 각각 2개씩 사용한 경우의 합성곱 연산을 보여준다. 즉, 크기가 2인 필터 2개, 크기가 3인 필터 2개, 크기가 4인 필터 2개를 사용하여 전체 6개의 특징맵을 생성한 구조이다. 이 경우 특징맵은 필터 크기마다 서로 다른 크기를 갖게 되고, 이를 통해 모형은 다양한 관점에서 국소적 특징을 학습하게 된다. 또한 두 그림은 1개의 합성곱층이 사용된 구조가 아니라 3개의 합성곱층을 사용한 구조임에 유의해야 한다. 즉, 모형 설계 시 크기가

다른 필터마다 합성곱층을 생성한 구조이며 각 합성곱층에서 생성된 특징맵들은 풀링층 이후에 서로 연결(concatenate)되는 과정을 거치게 된다.

[그림 3-20] 합성곱 연산 예시3

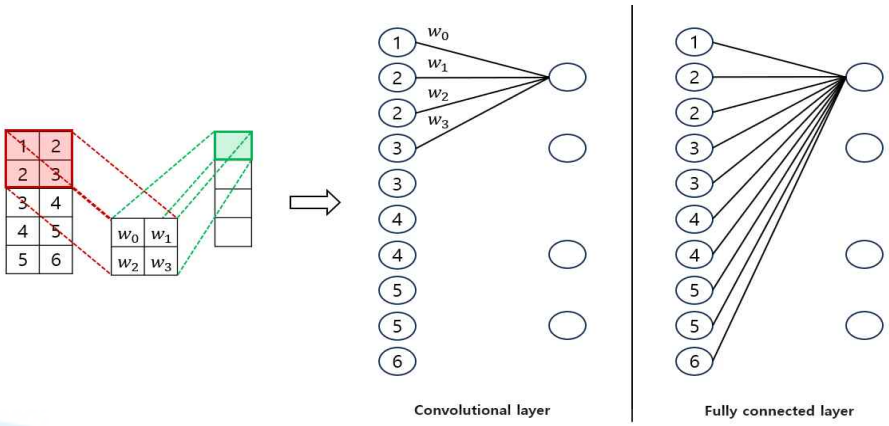


네 번째 예시인 [그림 3-21]은 합성곱 연산을 인공 신경망 형태로 표현한 그림이다. 합성곱층에서 가중치는 필터의 각 원소를 의미하므로 해당 그림과 같이 필터의 크기가 2×2×1인 경우에는 사용되는 가중치가 4개뿐이다. 또한 합성곱 연산은 국소적 연결성(local connectivity)으로 인해 필터와 연결되는 입력층의 뉴런만 은닉층의 뉴런과 연결되므로 모든 입력층의 뉴런이 은닉층의 뉴런과 연결되는 완전연결층보다 훨씬 적은 수의 가중치를 사용하게 된다. 따라서 합성곱층은 적은 매개변수의 양으로 국소적 정보를 보존한다는 특징이 있으며, 이때 편향을 포함한 합성곱층의 전체 매개변수의 수는 다음 식 (3-39)[91]와 같이 산출된다.

91) 해당 식은 편향이 필터의 채널이 아니라 필터의 수마다 생성된다는 것을 보여준다.

(3-39) 입력 채널의 수×필터 높이×필터 길이×필터의 수 + 편향의 수

[그림 3-21] 합성곱 연산 예시4



한편 앞서 살펴본 RNN 모형들은 가중치 연산을 마치고 비선형성을 추가하기 위해 활성화 함수를 적용하였는데, 이는 합성곱층도 마찬가지다. 합성곱층에서도 합성곱 연산을 통해 나온 특징맵에 활성화 함수를 적용하여 비선형성을 추가하게 되며, 이때 활성화 함수는 주로 ReLU나 ReLU의 변형들을 사용하게 된다. 즉, 합성곱층은 합성곱 연산을 통해 얻은 특징맵을 활성화 함수에 적용하는 것까지 포함한 개념을 말한다.

이상의 1D CNN의 합성곱층 연산 과정을 수리적으로 정의하면 다음과 같다. 먼저 입력 데이터가 $n \times l$ 크기의 문장 시퀀스이고, $\mathbf{x}_i \in \mathbb{R}^l$가 i번째 단어에 해당하는 l-차원의 단어 벡터를 의미한다면 전체 입력 데이터는 다음 식 (3-39)와 같다.

$$\mathbf{x}_{1:n} = \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \cdots \oplus \mathbf{x}_n \quad (3\text{-}39)$$

여기서 $\oplus$는 접합(concatenation) 연산자를 의미한다. 예를 들어 $\mathbf{x}_{i:i+j}$는 단어 i부터 단어 j까지의 접합($\mathbf{x}_i, \mathbf{x}_{i+1}, \cdots, \mathbf{x}_{i+j}$)을 말한다. 특징맵은 앞서 살펴보았듯이 입력 데이터의 수용 영역과 필터의 내적으로 생성되므로 $\mathbf{W} \in \mathbb{R}^{hl}$가 $h \times l$ 크기의 필터, 즉 가중치 행렬을 의미한다면 특징맵의 i번째 요소 c_i

는 다음 식 (3-40)과 같이 산출된다[92].

$$(3\text{-}40)\ c_i = f(\mathbf{W} \cdot \mathbf{x}_{i:i+h-1} + b)$$

여기서 f는 ReLU와 같은 비선형 함수를 말하며, b는 편향이다. 그리고 특징맵은 식 (3-38)을 따라 $n-h+1$의 크기를 가지므로 전체 특징맵은 다음 식 (3-41)과 같이 $\mathbf{c} \in \mathbb{R}^{n-h+1}$의 벡터가 된다.

$$(3\text{-}41)\ \mathbf{c} = [c_1, c_2, \cdots, c_{n-h+1}]$$

4) 풀링층(Pooling Layer)

합성곱층은 필터의 개수에 따라 다수의 특징맵을 생성하므로, 모형의 성능을 높이기 위해 필터의 수를 늘리거나 여러 개의 합성곱층을 사용하게 되면은 하나의 입력 데이터로부터 수많은 특징맵을 생성하게 된다. 그러나 이는 매개변수의 수를 기하급수적으로 증가하게 만들어 과적합(overfitting)의 원인으로 작용할 수 있으며, 이로 인한 과도한 연산량 증가는 더 많은 훈련 시간과 계산 자원을 필요하게 만든다. CNN은 이러한 문제를 방지하기 위해 합성곱층 다음에는 주기적으로 풀링층을 삽입하여 특징맵의 크기를 줄이고 특정 특징을 강조하게 도와주는 풀링 연산을 수행하게 된다.

풀링 연산은 합성곱 연산과 마찬가지로 슬라이딩 윈도우 방식으로 연산이 이루어지며, 일반적으로는 [그림 3-22]와 같이 각 풀링 연산이 서로 겹치지 않는 영역에서 수행되도록 필터와 스트라이드를 설정하게 된다.[93] 그리고 이

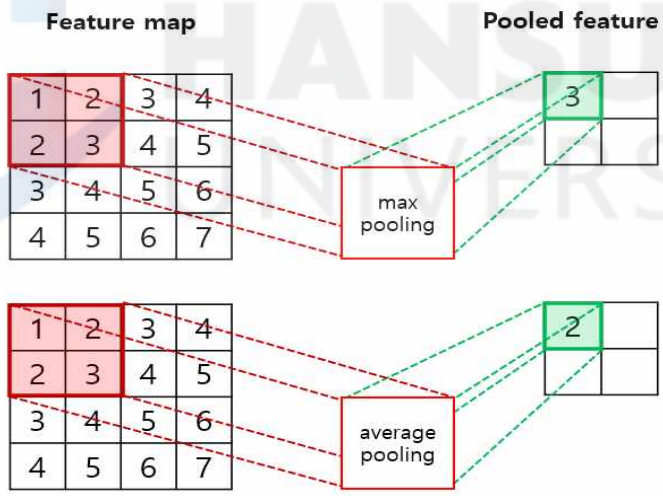
92) 만약 입력 데이터가 2개의 채널을 가졌다면 c_i는 다음과 같이 산출된다.

$$c_i = f(\sum_{k=1}^{2} W_k \cdot X_{k,i:i+h-1} + b)$$

93) [그림 3-22]는 4×4 특징맵에 2×2 크기의 필터(또는 풀링 윈도우)를 스트라이드 2로 적용한 모습을 보여준다. 이처럼 특징맵의 각 원소가 한 번씩만 풀링 연산에 사용되는 것을 겹치지 않는 풀링(non-overlapping pooling)이라 하며, 반대로 스트라이드를 필터의 크기보다 작게 설정하여 특징맵의 각 원소가 여러 번 연산에 사용되는 방식을 겹치는 풀링(overlapping pooling)이라 한다. 앞서 언급하였던 AlexNet이 겹치는 풀링을 사용하였다.

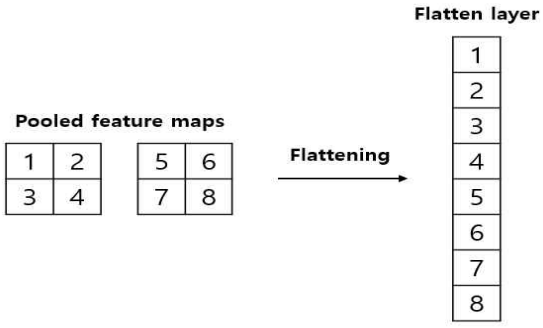
때 필터와 겹치는 부분에서 최댓값을 추출하는 풀링 연산을 최대 풀링(max pooling)이라 하며, 필터와 겹치는 부분에서 평균값을 추출하는 풀링 연산을 평균 풀링(average pooling)이라 한다. 최대 풀링은 특정 위치에서 가장 중요한 특징을 추출하게 되고, 평균 풀링은 특정 위치 전체를 대변하는 특징을 추출하게 된다. 이는 입력 데이터의 중요한 특징은 유지하면서 불필요한 정보나 잡음은 제거하게 도와주는 역할을 하며, 이를 통해 모형은 좀 더 핵심적인 특징에 집중할 수 있게 되고 입력 데이터의 작은 위치 변화에도 크게 영향을 받지 않는 강건한 특징을 얻게 된다. 따라서 풀링 연산은 필터와 스트라이드의 개념이 있다는 점에서는 합성곱 연산과 유사하지만, 최댓값이나 평균을 취하는 연산만 수행하기 때문에 학습해야 할 가중치가 없고 특징맵의 채널 수가 변하지 않는다는 차이가 있다.

[그림 3-22] 풀링 연산



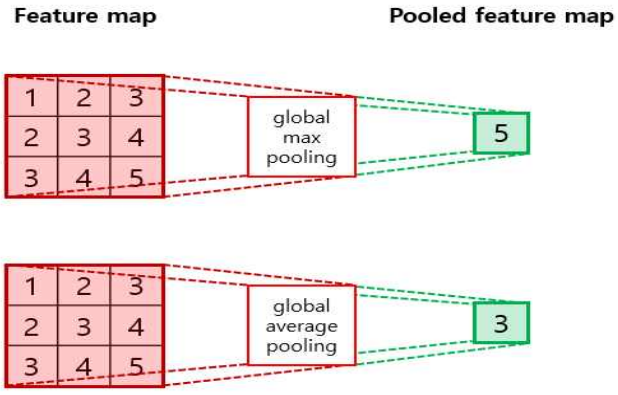
풀링 연산이 끝난 특징맵은 분류 작업을 수행하기 위해 완전연결층으로 보내지게 된다. 그러나 완전연결층은 1차원 데이터를 입력으로 받기 때문에 특징맵을 완전연결층에 보내기 위해서는 이를 1차원 형태로 바꿔주는 작업이 필요하다. 평탄화층(flatten layer)은 이를 위해 다음 [그림 3-23]과 같이 풀링층을 거친 모든 특징맵을 일렬로 펴서 1차원 벡터로 변환하는 역할을 한다.

[그림 3-23] 평탄화층



그러나 평탄화 과정은 특징맵이 가지고 있던 공간적 위치 정보를 사라지게 만드는 문제가 있다. 즉, 다차원 특징맵이 1차원 벡터로 퍼지면서 각 특징이 위치했던 정보가 소실되어 버리는 것이다. 전역 풀링(global pooling)은 이러한 문제를 해결할 수 있는 또 다른 풀링 기법이다. 이는 다음 [그림 3-24]와 같이 특징맵의 특정 영역이 아닌 전체 영역을 하나의 스칼라(scalar) 값으로 변환하는 연산 방법으로, 전체 영역에서 가장 큰 값을 사용하는 경우를 전역 최대 풀링(Global Max Pooling, GMP), 전체 영역의 평균값을 사용하는 경우를 전역 평균 풀링(Global Max Pooling, GAP)이라 한다. 따라서 전역 풀링은 풀링 연산 후에도 평탄화 과정 없이 1차원 형태를 가질 수 있다는 장점이 있으며, 풀링 후에도 정보 손실이 적어 완전연결층 없이 바로 분류 작업을 수행해 모형의 경량화와 과적합 방지에 도움을 줄 수도 있다.

[그림 3-24] 전역 풀링



5) 완전연결층(Fully Connected Layer), 출력층(Output Layer)

합성곱층과 풀링층을 통해 특징이 추출되면 그 이후 과정은 다른 인공 신경망들과 같다. 즉, 추출된 특징을 입력으로 사용한다는 것만 다를 뿐 다층퍼셉트론(Multilayer Perceptreon, MLP)과 같이 완전연결층과 출력층을 사용하여 분류나 회귀 문제를 풀게 된다. 이때 완전연결층은 이전 계층의 모든 뉴런과 연결되어 상당수의 매개변수를 가지며, 출력층은 완전연결층의 출력을 입력으로 받아 소프트맥스와 같은 활성화 함수를 적용해 예측치를 산출한다. 그리고 출력층의 예측치는 손실 함수를 통해 오차가 계산되고, 계산된 오차는 역전파 과정에서 전체 매개변수를 학습하는 데 사용한다.

3.2.1.3 트랜스포머(Transformer)

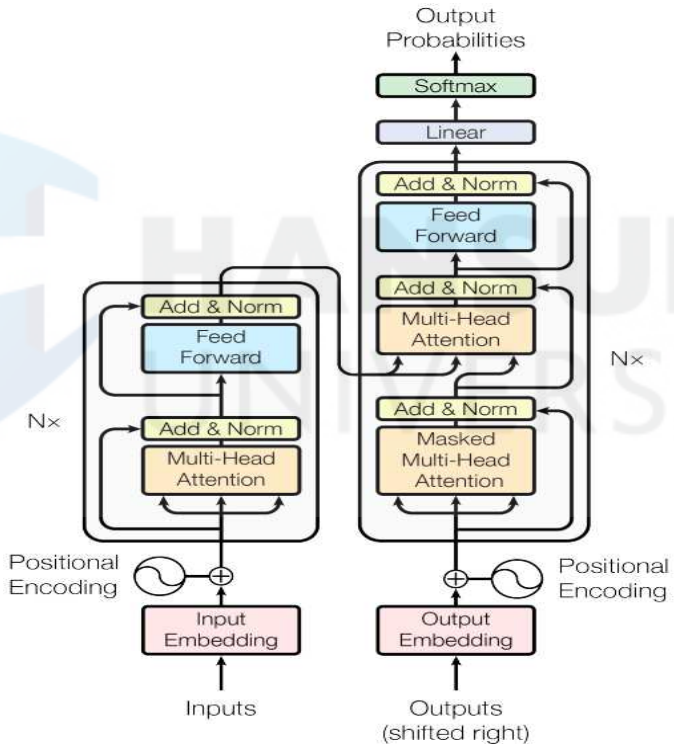
1) 트랜스포머 개요

트랜스포머(Vaswani et al., 2017)는 2017년 구글 연구진이 제시한 모형으로 RNN 없이 어텐션만으로 Seq2Seq의 인코더-디코더 구조를 구현한 모형이다. 트랜스포머는 어텐션만으로 모형을 설계하였음에도 자연어 처리 태스크에서 뛰어난 성능을 나타냈으며, 최근 주목받은 BERT와 GPT 역시 트랜스포머의 인코더와 디코더를 활용한 언어 모형들이다.

Vaswani et al.(2017)이 제안한 트래스포머의 구조도는 아래 [그림 3-25]와 같다. 트랜스포머의 인코더와 디코더는 N 개의 동일한 레이어로 이루어져 있으며, 임베딩 벡터와 위치 인코딩(positional encoding)이 더해진 값을 입력으로 사용한다. 인코더의 각 레이어는 멀티 헤드 셀프 어텐션(multi-head self-attention)층과 위치별 완전 연결 FFNN(position-wise fully connected FFNN)층으로 구성되며, 이 두 개의 서브층(sublayer)은 잔차 연결(residual connection)과 층 정규화(layer normalization)가 적용된다. 즉, 두 서브층의 출력은 $LayerNorm(x + Sublayer(x))$이다. 디코더의 경우에는 각 레이어가 3개

의 서브층으로 구성된다. 첫 번째 서브층은 인코더의 멀티 헤드 셀프 어텐션 층과 동일한 연산을 수행하지만, 어텐션 스코어에 룩-어헤드 마스크(look-ahead mask)가 적용된다는 차이가 있으며, 두 번째 서브층은 셀프 어텐션이 아니라는 차이가 있다. 즉, 두 번째 서브층은 어텐션의 쿼리가 디코더의 첫 번째 서브층의 출력이 되고, 키와 밸류는 인코더의 출력이 되는 구조다. 그리고 세 번째 서브층은 인코더의 두 번째 서브층과 같은 위치별 완전 연결 FFNN층을 말한다.

[그림 3-25] 트랜스포머 구조도(Vaswani et al., 2017)



2) 위치 인코딩(Positional Encoding), 위치 임베딩(Position Embedding)

트랜스포머는 RNN처럼 단어 위치에 따라 순차적으로 입력을 받는 방식이 아니기 때문에 임베딩 벡터를 그대로 모형의 입력으로 사용하면 단어의 위치 정보(position information)가 반영되지 않는다. 따라서 트랜스포머는 단

어의 위치 정보를 별도로 제공해야 하는데, Vaswani et al.(2017)은 이를 위해 위치 인코딩을 활용한다.

위치 인코딩은 아래 식 (3-42)와 같이 사인 함수와 코사인 함수를 통해 구현된다. 해당 식에서 pos는 임베딩 벡터의 위치를 의미하며, d_{model}은 임베딩 벡터의 차원을, i는 임베딩 백터 차원의 인덱스를 의미한다. 즉, 짝수 인덱스에는 사인값을 배치하고, 홀수 인덱스에는 코사인값을 배치하는 방식이다. 그리고 이렇게 계산된 위치 인코딩 값을 각 임베딩 벡터에 더하면 해당 임베딩 벡터는 같은 단어라도 문장 내에 위치에 따라 다른 값을 갖게 된다.

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}}) \quad (3\text{-}42)$$
$$PE_{(pos,2i+1)} = cos(pos/100000^{2i/d_{model}})$$

위치 인코딩이 사전에 정의된 함수를 통해 위치 정보를 구현하였다면 위치 임베딩(position embedding)은 학습을 통해 위치 정보를 구현하는 방식이다. 즉, 입력을 위한 임베딩층 이외에 위치 정보를 위한 임베딩층을 하나 더 사용하여 각 위치가 어떻게 표현되어야 할지를 스스로 학습하게 한다. 이러한 방식은 BERT에서 사용되는 방법이기도 하며, 본 연구에서도 위치 인코딩 대신 위치 임베딩을 이용해 트랜스포머 모형을 설계하였다.

3) 멀티 헤드 셀프 어텐션(Multi-head Self-attention)

멀티 헤드 셀프 어텐션에서 사용되는 셀프 어텐션은 앞서 살펴본 어텐션과는 달리 쿼리, 키, 밸류가 모두 같은 입력 시퀀스에서 파생된다는 특징을 갖는다. 즉, 입력 시퀀스가 문장이라면 해당 문장을 구성하는 단어들이 쿼리, 키, 밸류가 되어 각 단어 사이의 유사도(어텐션 스코어)를 측정하는 방식이다. 그리고 트랜스포머는 이러한 어텐션 방식을 통해 장기 의존성 문제를 매우 효과적으로 해결하게 된다. 특히 LSTM은 시점이 길어질수록 셀 상태에 들어가 있는 정보가 손실되거나 왜곡될 수 있는데, 셀프 어텐션은 각 쿼리에 대한 모든 키의 유사도가 직접적으로 계산되므로 이러한 문제를 방지할 수 있다.

게다가 LSTM은 문장 끝에 위치한 단어의 정보를 파악하기 위해 각 시점에 걸쳐 셀 상태의 정보를 전달해야 하지만, 셀프 어텐션은 아무리 멀리 떨어진 두 단어의 관계도 한 번의 계산으로 파악할 수 있어 계산 비용 측면에서도 LSTM보다 효율적이다.

셀프 어텐션을 수행하기 위해 가장 먼저 처리하는 작업은 Q(쿼리), K(키), V(밸류) 벡터를 변환하는 것이다. 다시 말해 입력 시퀀스의 각 단어 벡터가 Q, K, V에 할당되면 Q, K, V는 가중치 행렬 $\mathbf{W}^{Q} \in \mathbb{R}^{d_{model} \times d_k}$, $\mathbf{W}^{K} \in \mathbb{R}^{d_{model} \times d_k}$, $\mathbf{W}^{V} \in \mathbb{R}^{d_{model} \times d_v}$와 곱해지는 과정이 수행된다. 이때 각 가중치 행렬의 크기는 $d_{model} \times (d_{model}/h)$가 되며, 여기서 d_{model}은 임베딩 벡터의 차원을, h는 어텐션을 수행할 때 사용되는 병렬의 개수(어텐션의 헤드 수)를 의미하며, $d_{model}/h = d_k = d_v$가 된다. 예를 들어 임베딩 벡터의 차원(d_{model})이 512이고, h가 8이라면 Q, K, V는 64의 크기를 갖는 벡터로 변환된다.

Q·K·V가 변환되면 앞서 살펴본 어텐션 메커니즘과 마찬가지로 Q에 대한 모든 K의 어텐션 스코어를 구하고, 이를 V에 반영해 어텐션 값(문맥 벡터)을 계산한다. 그리고 이때 사용되는 스코어 함수는 식 (3-34)의 *scaled dot*이며, 스케일링 값($\sqrt{n}$)은 $\sqrt{d_k}$가 된다. 이에 따라 어텐션 함수는 식 (3-43)과 같이 정의되는데, 여기서 한 가지 주의할 점은 $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$가 벡터가 아닌 행렬을 의미한다는 것이다. 즉, 실제 멀티 헤드 셀프 어텐션의 연산 과정은 각 단어에 대한 어텐션 값을 하나씩 계산하는 것이 아니라, 행렬 연산을 통해 일괄적으로 처리된다. 다시 말해 문장 내에 단어를 하나씩 Q·K·V에 할당하는 것이 아니라, 문장 길이(seq_len)×임베딩 차원(d_{model}) 크기의 문장 행렬이 곧 Q·K·V가 되고, 여기에 가중치 행렬 $\mathbf{W}^{Q}$, $\mathbf{W}^{K}$, $\mathbf{W}^{V}$를 곱하여 $\mathbf{Q} \in \mathbb{R}^{seq\_len \times d_k}$, $\mathbf{K} \in \mathbb{R}^{seq\_len \times d_k}$, $\mathbf{V} \in \mathbb{R}^{seq_len \times d_v}$를 생성한다. 따라서 $\mathbf{QK}^{\top}/\sqrt{d_k}$로 산출되는 행렬은 각 행과 열이 스코어 값을 갖게 되고, 최종적으로 산출되는 어텐션 값은 $seq\_len \times d_v$ 크기의 행렬이 된다.

$$(3\text{-}43)\quad Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\mathbf{QK}^{\top}/\sqrt{d_k})\mathbf{V}$$

마지막으로 멀티 헤드 셀프 어텐션층은 이상의 어텐션을 한 번만 수행하는 것이 아닌 어텐션의 헤드 수만큼 병렬로 수행한다는 특징을 갖는다. 구체적으로 설명하면, 어텐션 헤드마다 다른 가중치 행렬($\mathbf{W}^Q$, $\mathbf{W}^K$, $\mathbf{W}^V$)을 사용하여 $seq_len \times d_{model}/h$의 크기를 갖는 $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ 행렬을 생성하고, 이렇게 생성된 h 개의 $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$에 대해 어텐션을 수행한다. 그리고 병렬 어텐션이 수행된 모든 어텐션 값을 연결(concatenate)하는 과정을 거치면 $seq\_len \times d_{model}$ 크기의 행렬이 생성되고, 여기에 또 다른 가중치 행렬 $\mathbf{W}^O \in \mathbb{R}^{hd_v \times d_{model}}$를 곱하여 멀티 헤드 셀프 어텐션층의 최종 출력을 결정한다. 이를 수식으로 나타내면 식 (3-44)와 같으며, 이에 따라 멀티 헤드 셀프 어텐션층의 출력은 인코더의 입력이었던 문장 행렬의 크기($seq\_len \times d_{model}$)와 같은 크기를 갖는다.

$$(3\text{-}44)\quad MultiHead(\mathbf{Q},\mathbf{K},\mathbf{V}) = Concat(\mathbf{head}_1,\cdots,\mathbf{head}_h)\mathbf{W}^O$$
$$\text{,where } \mathbf{head}_i = Attention(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V)$$

이러한 병렬 어텐션은 모형이 단어 간의 다양한 관계를 학습할 수 있도록 도와주는 역할을 한다. 예를 들어 “그 여자는 가게에 들어가서 그녀에게 어울리는 꽃을 샀다”라는 문장이 있으면 첫 번째 어텐션 헤드는 의미적 관계에 집중하여 ‘여자’와 ‘그녀’의 연관도를 높게 판단할 수도 있고, 두 번째 어텐션 헤드는 문법적 관계에 집중하여 주어인 ‘그녀’와 동사인 ‘샀다’의 연관도를 높게 판단할 수도 있다. 그리고 이렇게 어텐션 헤드들이 각기 다른 측면에 집중하게 되면 모형은 다양한 문맥적 정보들을 수집하여 학습할 수 있게 된다.

4) 위치별 완전 연결 FFNN(Position-wise Fully Connected FFNN)

트랜스포머 인코더의 두 번째 서브층인 위치별 완전 연결 FFNN은 다음 식 (3-45)와 같이 2개의 완전연결층과 그 사이에 활성화 함수(ReLU)로 설계된 특수한 형태의 FFNN 중 하나이다. 해당 식에서 $\mathbf{x}$는 $seq\_len \times d_{model}$ 크기의 입력 행렬을 의미하지만, 이를 행 벡터 단위로 생각한다면 해당 식은 각

행 벡터의 위치마다 적용된다고 볼 수 있다. 즉, $\mathbf{x}$의 각 단어 벡터(위치)는 식 (3-45)를 통해 두 번의 선형 변환과 한 번의 비선형성이 적용된다. 또한, $\mathbf{W}_1$와 $\mathbf{W}_2$는 각각 $d_{model} \times d_{ff}$와 $d_{ff} \times d_{model}$의 크기를 가지는 가중치 행렬을 의미하며, 이때 d_{ff}는 첫 번째 완전연결층의 뉴런 수를 의미하는 하이퍼파라미터이다. 따라서 위치별 완전 연결 FFNN 역시 연산 후에는 인코더의 입력 크기 $seq_len \times d_{model}$가 보존되며, 이렇게 출력된 행렬은 잔차 연결과 층 정규화를 거쳐 다음 인코더의 입력으로 사용되거나 최종 분류 작업에 활용된다.

$$(3\text{-}45)\quad FFNN(\mathbf{x}) = Max(0, \mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2$$

5) 잔차 연결(Residual Connection)과 층 정규화(Layer Normalization)

트랜스포머는 학습 안정성을 위해 서브층마다 잔차 연결과 층 정규화라는 기법이 추가로 적용된다. 잔차 연결은 입력을 서브층의 출력에 직접 더하는 방식으로 구현된다. 예를 들어 서브층의 출력이 $Sublayer(x)$이면 잔차 연결은 $x + Sublayer(x)$와 같이 수행된다. 이는 앞서 언급한 CNN의 ResNet에서 사용되던 방법으로 신경망의 층이 깊어질수록 발생하는 기울기 소실과 정보 소실 문제를 방지하기 위해 사용되는 방법이다.

잔차 연결 이후에는 바로 층 정규화가 수행된다. 층 정규화는 Ba et al.(2016)이 제안한 정규화 기법으로 특히 배치 정규화 사용에 문제가 발생하는 RNN에서 주로 사용되는 방법이다. 트랜스포머의 층 정규화 과정은 다음 세 가지 식으로 설명될 수 있다. 먼저 층 정규화의 평균과 분산은 다음 식 (3-46)에 의해 산출된다.

$$(3\text{-}46)\quad \mu_i = \frac{1}{d_{model}}\sum_{j=1}^{d_{model}} \mathbf{x}_{i,j} \qquad \sigma_i^2 = \frac{1}{d_{model}}\sum_{j=1}^{d_{model}} (\mathbf{x}_{i,j} - \mu_i)^2$$
$$(where\ \ i = 1, 2, \cdots, seq\_len)$$

여기서 $\mathbf{x}_i$는 문장 행렬 내에 있는 각각의 단어 벡터를 의미하며, $\mathbf{x}_{i,j}$는 각

단어 벡터의 원소를 의미한다. 따라서 μ_i와 σ_i^2는 각 단어 벡터의 평균과 분산을 말하며, 이렇게 산출된 μ_i와 σ_i^2는 식 (3-47)를 따라 $\mathbf{x}_i$를 정규화하는 데 사용된다.

$$\hat{\mathbf{x}}_{i,j} = \frac{\mathbf{x}_{i,j} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}} \quad (3\text{-}47)$$

여기서 ϵ은 분모가 0이 되는 것을 방지하기 위해 사용되는 상수(예: 10^{-5})이다. 마지막으로 식 (3-47)를 통해 산출된 $\hat{\mathbf{x}}_i$는 식 (3-48)과 같이 학습 가능한 매개변수 $\boldsymbol{\gamma} \in \mathbb{R}^{d_{model}}$와 $\boldsymbol{\beta} \in \mathbb{R}^{d_{model}}$를 통해 다시 조정되며, 이때 가중치 역할을 하는 $\boldsymbol{\gamma}$의 초기값은 1이고 편향 역할을 하는 $\boldsymbol{\beta}$의 초기값은 0이 된다.

$$\mathbf{y}_i = \boldsymbol{\gamma}\hat{\mathbf{x}}_i + \boldsymbol{\beta} \quad (3\text{-}48)$$

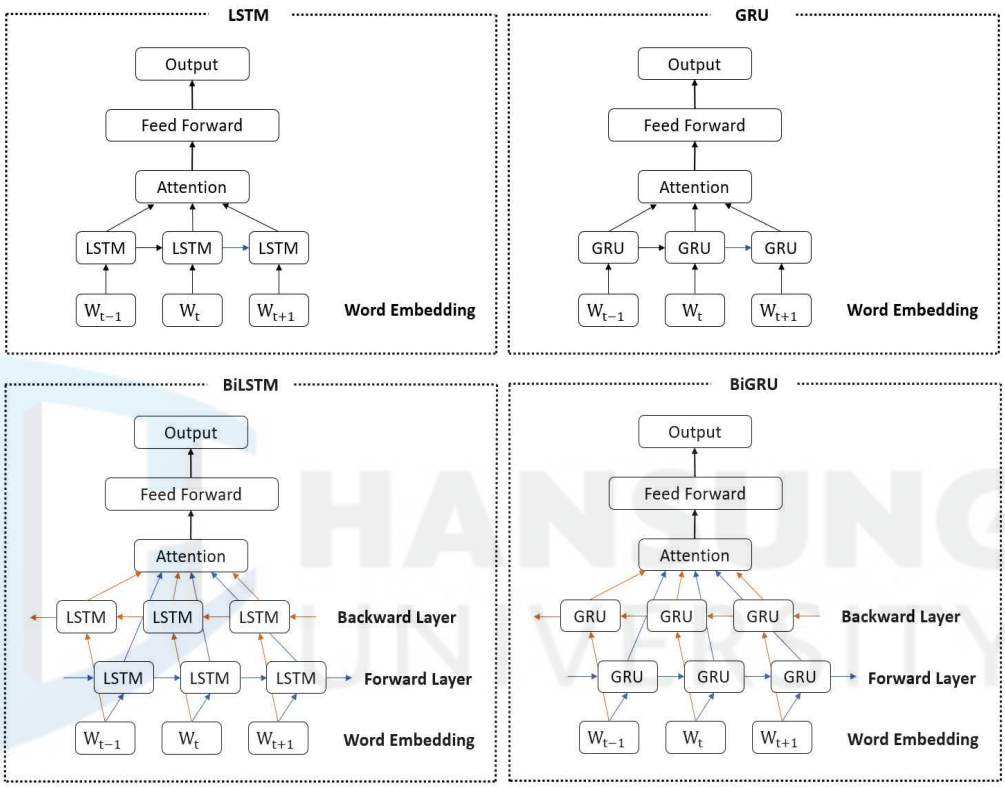
3.2.2 감정 분류 모형 구축 및 평가

3.2.2.1 RNN 모형 구축 및 평가

RNN 기반 모형은 [그림 3-26]과 같이 4개의 모형을 구축하여 LSTM, BiLSTM, GRU, BiGRU의 성능을 비교하였다. 각 모형의 입력층은 앞서 구축한 Word2Vec 모형으로 초기화되며, 학습 과정 시 미세 조정이 수행된다. 은닉층은 RNN층, 어텐션층, 완전연결층이 하나씩 포함되도록 설계하였다. 이때 RNN층은 2개를 쌓은 깊은 순환 신경망보다 1개만 쌓았을 때 더 높은 성능을 나타냈으며, 어텐션층과 완전연결층은 추가되지 않은 경우보다 추가되었을 때 더 높은 성능을 나타냈다. 또한, 어텐션 함수는 식 (3-34)의 *concat*2를 활용하였으며, 이에 따라 어텐션층의 스코어 값, 어텐션 가중치, 문맥 벡터는 식 (3-36)~(3-38)과 같이 구현되었다. 마지막으로 출력층은 다중 분류 문제를 풀기 위해 소프트맥스 함수가 사용되었고, 손실 함수는 categorical cross

entropy가 사용되었다. 그리고 레이블 불균형 문제에 대응하기 위해 손실 함수에 레이블 가중치를 반영하였다.

[그림 3-26] RNN 기반 평가 모형 아키텍처



모형 구축에 사용된 주요 하이퍼파라미터는 [표 3-5]과 같다. Word2Vec의 윈도우 크기와 단어의 최소 빈도수는 각각 6과 5로 설정하였으며, 임베딩 벡터의 차원은 100, 200, 300을 비교하여 분류 정확도가 높은 300으로 설정하였다. 은닉 상태의 크기와 완전연결층의 뉴런 수는 각각 32, 64, 128을 비교한 결과, 은닉 상태는 128, 완전연결층은 64일 때 가장 높은 성능을 보였으며, 배치 크기는 64일 때 학습 속도와 예측 성능 간 균형이 가장 우수하였다. 최적화 알고리즘은 초기 학습 단계의 불안정성을 해결하기 위해 Adam의 변형인 RAdam($\beta_1 = 0.9,\ \beta_2 = 0.999,\ \epsilon = 10^{-9}$)을 사용하였고, 초기 학습률은 0.0005로 설정하였다. 스텝 수는 배치 크기 64, 학습 데이터 16,000개, 에폭

(epoch) 30을 기준으로 7,500이 계산되었으며, 이를 바탕으로 학습률이 스텝 수에 따라 점진적으로 감소하는 지수기반 스케줄링(exponential scheduling)을 적용하였다. 또한, 전체 스텝의 10% 구간을 웜업(warm-up) 단계로 설정하여 학습 초기에 발생할 수 있는 급격한 가중치 변화와 손실 폭등을 완화하고자 하였다.

[표 3-5] RNN 하이퍼파라미터

Category	Item	Value
Word2Vec hyperparameter	Vector dimension	300
	Window size	6
	Minimum word count	5
Model hyperparameter	RNN units	128
	Dense units	128
Training hyperparameter	Batch size	64
	Epochs	30
	Steps	7,500

모형 평가는 학습 데이터가 부족한 점을 고려하여 계층적 5-fold CV를 수행하였으며, 평가 결과는 각 평가 지표에 대한 5회 검증 결과의 평균값으로 측정하였다. 구체적으로는 전체 데이터셋을 레이블 비율이 유지되도록 다섯 개의 하위 그룹으로 분할 한 후, 그중 네 그룹을 학습 데이터로, 나머지 한 그룹을 검증 데이터로 사용하여 검증 데이터에 대한 정확도(accuracy), 정밀도(precision), 재현율(recall), F1 Score를 측정하였다. 그리고 이 과정을 검증용 그룹을 바꿔가며 총 다섯 번 반복하고, 해당 지표들의 평균값을 최종 성능으로 사용하였다. 이때 정밀도, 재현율, F1 Score는 다중 레이블과 레이블 불균형을 고려하여 가중 평균(weighted average)으로 측정하였다. 또한, 검증 데이터의 손실(validation loss)이 4 에폭 동안 개선되지 않을 경우, 학습을 조기 종료(early stopping)하여 모형의 과적합을 방지하였다.

평가 결과는 [표 3-6]에 제시된 바와 같이 BiGRU가 모든 평가 지표에서 가장 우수한 성능을 기록하였다. 구체적으로, BiGRU는 정확도 85.48%, 정밀도 85.59%, 재현율 85.48%, F1 Score 85.48%를 기록하였으며, 그 뒤를 이

어 GRU 84.65%, BiLSTM 84.09%, LSTM 83.45% 순으로 F1 Score가 높게 나타났다. 이러한 결과는 본 연구의 경제정책 뉴스기사에 대해 LSTM 계열보다 GRU 계열이, 단방향 구조보다 양방향 구조가 더 적합하다는 점을 시사한다. 특히, GRU 기반 모형의 상대적 우수성은 GRU가 LSTM보다 파라미터 수가 적고 구조도 간결하여, 소규모 학습 데이터 환경에 더 안정적인 성능을 보일 수 있었던 것으로 해석된다.[94] 또한, log loss 역시 BiGRU가 0.3999로 가장 낮은 값을 기록하였는데, 이는 모형이 정답 레이블에 더 높은 확률을 안정적으로 부여하고, 예측 확률 분포가 실제 정답과 가장 유사함을 의미한다.[95] 따라서 RNN 기반 모형들 중에서는 BiGRU가 본 연구의 최적 모형으로 판단된다.

[표 3-6] RNN 모형 평가 결과

구분[1]	LSTM	BiLSTM	GRU	BiGRU
정확도	83.43%	84.09%	84.66%	85.48%
정밀도	83.66%	84.28%	84.72%	85.59%
재현율	83.43%	84.09%	84.66%	85.48%
F1 Score	83.45%	84.09%	84.65%	85.48%
log loss	0.4380	0.4267	0.4137	0.3999

주: 1) 정밀도, 재현율, F1 Score는 다중 레이블로 인해 평균값으로 측정되며, 이때 평균값은 레이블 불균형을 고려해 가중 평균으로 측정됨

3.2.2.2 CNN 모형 구축 및 평가

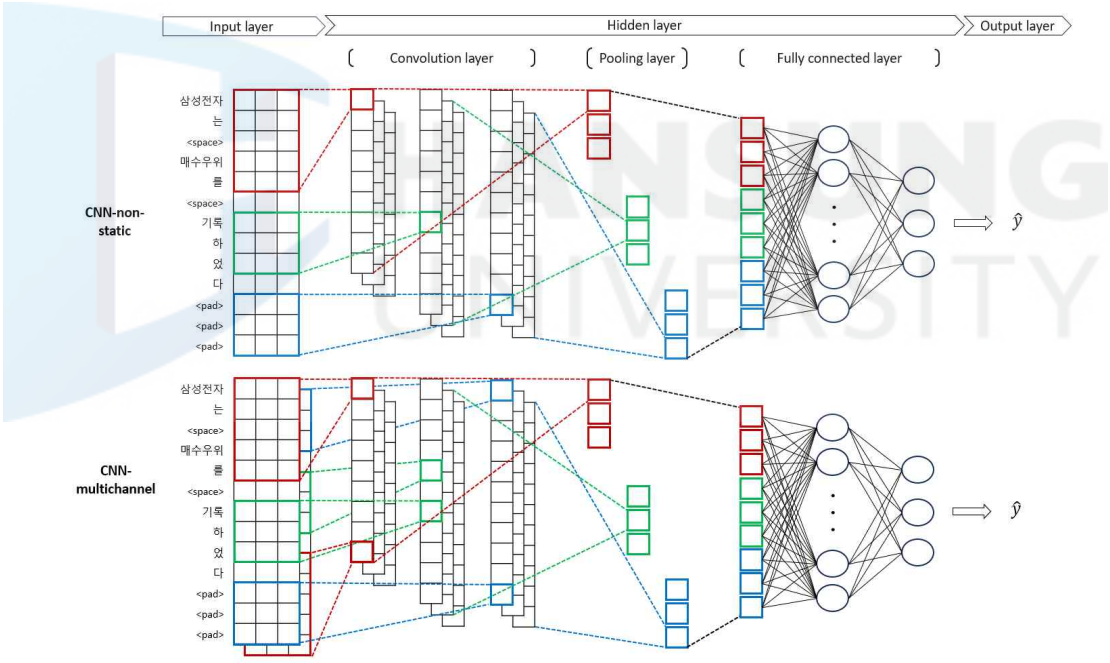
CNN 모형은 Kim(2014)의 연구에서 소개된 모형 중 다양한 데이터셋에 대해 전체적으로 성능이 우수하였던 CNN-non-static과 CNN-multichannel을 기반으로 구축하였다. 구축된 두 모형의 아키텍처는 아래 [그림 3-27]과 같다. 먼저, 입력층은 앞서 구축한 Word2Vec 모형을 초기 가중치로 사용하

94) 경험적으로 학습 데이터의 양이 적을 때는 매개변수의 양이 적은 GRU가 조금 더 낮고 학습 데이터의 양이 많은 경우에는 LSTM이 조금 더 낮다고 알려져 있다(유원준·안상준, 2022).

95) 로그 손실(log loss)은 값이 작을수록 모형의 확률 분포가 실제 정답 분포와 유사함을 의미하며, 값이 클수록 예측 신뢰도가 낮거나 잘못된 레이블에 높은 확률을 부여했음을 의미한다.

였다. 이때 CNN-non-static은 사전 학습된 단어 벡터가 태스크에 맞게 미세 조정되며, CNN-multichannel은 한 채널만 미세 조정되고 다른 채널은 정적(static)으로 유지된다. 다음으로 은닉층은 합성곱층 3개, 풀링층 3개, 완전연결층 1개로 구성하였다. 합성곱층은 크기가 5, 3, 3인 필터가 50개씩 있으며, 풀링층에서는 전역 최대 풀링이 풀링 연산으로 수행된다. 그리고 완전연결층으로 설계된 penultimate layer에서는 정규화(regularization)를 위해 드롭아웃(dropout)과 l2-노름(norms)이 적용되어 가중치 크기가 제한된다. 마지막으로 출력층은 다중 분류 문제를 풀기 위해 소프트맥스 함수가 사용되고, 손실 함수는 레이블 가중치가 반영된 categorical cross entropy가 사용된다.

[그림 3-27] CNN 기반 평가 모형 아키텍처



하이퍼파라미터는 각 구성 요소별 성능 비교를 통해 최적화하였으며, 그 결과는 [표 3-7]에 제시되어있다. Word2Vec의 윈도우 크기와 단어의 최소 빈도수는 6과 5로 고정하였고, 임베딩 벡터의 차원은 100, 200, 300 중 300이 가장 높은 분류 정확도를 보였다. 합성곱층의 커널 크기는 (3, 3, 3)부터 (5, 5, 5)까지 다양한 조합을 실험한 결과, (5, 3, 3)이 가장 높은 성능을 나

타냈으며, 필터 수는 50과 100 중 50에서 더 높은 성능이 관찰되었다. 완전연결층의 뉴런 수는 32, 64, 128 중에서, 드롭아웃 비율은 0.1, 0.2, 0.5 중에서 32와 0.2가 최적값으로 선택되었고, 배치 크기는 64에서 학습 속도와 예측 성능 간 균형이 가장 우수하였다. 스텝 수는 배치 크기 64, 학습 데이터 16,000개, 에폭 30을 기준으로 7,500이 계산되었고, 전체 스텝의 초기 10%는 웜업 단계로 설정하였다. 마지막으로 최적화 알고리즘은 RAdam($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-9}$)을 사용하였으며, 초기 학습률은 0.0005로 설정하였다.

[표 3-7] CNN 하이퍼파라미터

Category	Item	Value
Word2Vec hyperparameter	Vector dimension	300
	Window size	6
	Minimum word count	5
Model hyperparameter	Kernel sizes	[5, 3, 3]
	Number of filters	50
	Dense units	32
	Dropout ratio	0.2
Training hyperparameter	Batch size	64
	Epochs	30
	Steps	7,500

모형 평가는 RNN 때와 마찬가지로 계층적 5-fold CV를 수행하였으며, 그 결과는 [표 3-8]에 제시하였다. 평가 결과에서는 Kim(2014)의 연구에서 기대했던 대로 CNN-multichannel 모형이 CNN-non-static 모형보다 전반적으로 우수한 성능을 나타냈다. 구체적으로는, CNN-multichannel이 정확도 85.23%, 정밀도 85.21%, 재현율 85.23%, F1 Score 85.21%를 기록하였으며, CNN-non-static이 정확도 85.06%, 정밀도 85.09%, 재현율 85.06%, F1 Score 85.04%를 기록하였다. 또한 log loss 지표에서도 CNN-multichannel이 0.4455, CNN-non-static이 0.4547을 기록하여, 확률 기반 예측의 정밀도(confidence) 측면에서도 CNN-multichannel이 더 우월한 것으로 나타났다.

[표 3-8] CNN 모형 평가 결과

구분[1)]	CNN-non-static	CNN-multichannel
정확도	85.06%	85.23%
정밀도	85.09%	85.21%
재현율	85.06%	85.23%
F1 Score	85.04%	85.21%
log loss	0.4547	0.4455

주: 1) 정밀도, 재현율, F1 Score는 다중 레이블로 인해 평균값으로 측정되며, 이때 평균값은 레이블 불균형을 고려해 가중 평균으로 측정됨

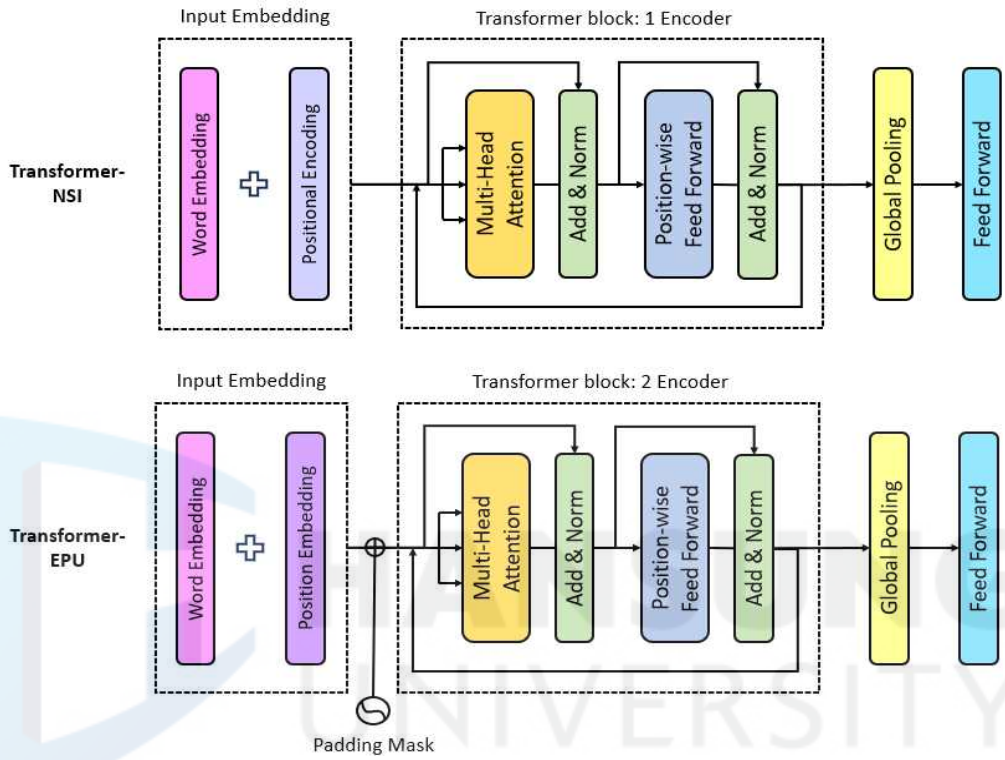
3.2.2.3 트랜스포머 모형 구축 및 평가

트랜스포머 모형은 아래 [그림 3-28]과 같은 2개의 아키텍처를 설계하여 비교하였다. 첫 번째 아키텍처인 Transformer-NSI는 서범석 외(2022)에서 제안된 구조이며, 두 번째 아키텍처인 Transformer-EPU는 본 연구에서 제안한 구조이다. 두 모형의 입력층은 앞서 구축한 Word2Vec 모형으로 초기화되며, 학습 시 미세 조정이 수행된다. 이때, Transformer-NSI는 단어 임베딩에 위치 인코딩을 합산하여 입력 문장 행렬을 생성하지만, Transformer-EPU는 단어 임베딩에 위치 임베딩을 합산하여 입력 문장 행렬을 생성한다는 것에 차이가 있다. 트랜스포머 블록은 Transformer-NSI가 단일 인코더로 구성된 반면, Transformer-EPU는 2개의 인코더를 누적하여 심층적인 문맥 표현을 학습할 수 있도록 설계하였다. 이때, 각 인코더는 멀티 헤드 셀프 어텐션층과 위치별 완전 연결 FFNN층으로 구성되며, Transformer-NSI에는 입력 마스킹을 위한 패딩 마스크(Padding Mask)[96)]가 추가로 적용된다. 트랜스포머 블록 이후에는 두 모형 모두 풀링층과 완전연결층을 거쳐 분류 작업을 수행하며, 이때 수행된 풀링 연산은 전역 평균 풀링이다. 마지막으로, 출력층에서는 다중 분류 문제를 풀기 위해 소프트맥스 함수가 사용되고, 손실 함수는 레이블

96) 패딩 마스크는 입력 시퀀스에 패딩 토큰(<pad>)이 있는 경우, 이를 마스킹하여 어텐션에서 제외하기 위한 작업이다. 이때, 마스킹을 수행하는 방식은 패딩 토큰이 위치한 키 열에 매우 작은 음수 값을 넣어 주는 것이다. 즉, 키 열에 -1e9와 같은 음수 값을 넣어 주면 소프트맥스 함수를 지난 후 밸류 행렬과 곱해진 결과가 0이 되므로 해당 위치는 단어쌍 간 유사도를 계산하는 데 반영되지 않는다.

가중치가 반영된 categorical cross entropy가 사용된다.

[그림 3-28] 트랜스포머 기반 평가 모형 아키텍처



하이퍼파라미터는 아래 [표 3-9]와 같이 설정하였다. Word2Vec의 윈도우 크기와 단어의 최소 빈도수는 각각 6과 5로 설정하였고, 임베딩 벡터의 차원은 128, 256, 300을 비교하여 분류 정확도가 높은 300으로 설정하였다. 인코더는 앞서 언급한 대로 Transformer-NSI는 1개, Transformer-EPU는 2개로 설계하였으며, 어텐션의 헤드 수 h는 1, 2, 3, 4, 5, 6, 8 중 4에서 가장 높은 성능이 관찰되었다. 위치별 완전연결 FFNN의 뉴런 수 d_{ff}와 완전연결층의 뉴런 수는 32, 64, 128, 256 중에서 128이 최적값으로 선택되었고, 배치 크기는 64에서 학습 속도와 예측 성능 간 균형이 가장 우수하였다. 스텝 수는 배치 크기 64, 학습 데이터 16,000개, 에폭 30을 기준으로 7,500이 계산되었고, 전체 스텝의 초기 10%는 웜업 단계로 설정하였다. 마지막으로 최적화 알고리즘은 RAdam($\beta_1 = 0.9,\ \beta_2 = 0.999,\ \epsilon = 10^{-9}$)을 사용하였으며, 초기 학습

률은 0.0005로 설정하였다.

[표 3-9] 트랜스포머 하이퍼파라미터

Category	Item	Value
Word2Vec hyperparameter	Vector dimension(d_{model})	300
	Window size	6
	Minimum word count	5
Model hyperparameter	Encoder layers(N)	1, 2
	Attention heads(h)	4
	Position-wise FFNN units(d_{ff})	128
	Dense units	128
Training hyperparameter	Batch size	64
	Epochs	30
	Steps	7,500

평가 결과는 [표 3-10]과 같이 Transformer-EPU가 Transformer-NSI보다 전반적으로 우수한 성능을 나타냈다. 구체적으로, Transformer-EPU는 정확도 85.23%, 정밀도 85.21%, 재현율 85.23%, F1 Score 85.21%를, Transformer-NSI는 정확도 85.25%, 정밀도 85.44%, 재현율 85.25%, F1 Score 85.25%를 기록하였다. 또한 log loss 역시 Transformer-EPU가 더 낮은 값을 기록하여 예측의 신뢰도 측면에서도 Transformer-EPU가 더 안정적인 것으로 나타났다.

[표 3-10] 트랜스포머 모형 평가 결과

구분[1)]	Transformer-NSI	Transformer-EPU
정확도	85.25%	85.61%
정밀도	85.44%	85.69%
재현율	85.25%	85.61%
F1 Score	85.25%	85.59%
log loss	0.4182	0.3945

주: 1) 정밀도, 재현율, F1 Score는 다중 레이블로 인해 평균값으로 측정되며, 이때 평균값은 레이블 불균형을 고려해 가중 평균으로 측정됨

3.2.3 감정 분류 모형 선정

감정 분류 모형 선정을 위해 [표 3-11]에는 각 아키텍처에서 가장 우수한 성능을 기록한 BiGRU+Attention, CNN-multichannel, Transformer-EPU의 평가 결과를 제시하였다. 표에 나와 있듯이, 모형 성능은 F1 Score를 기준으로 Transformer-EPU, BiGRU+Attention, CNN-multichanne 순으로 높게 나타났으나, 세 모형 간의 차이는 그리 크지 않다. 특히, BiGRU+Attention과 Transformer-EPU의 성능 차이는 불과 0.11%P밖에 나타나지 않았는데, 이는 본 연구의 텍스트 데이터가 비교적 짧은 시퀀스 길이를 가진 데에서 기인한 결과로 해석된다. 즉, 앞서 살펴보았듯이 트랜스포머의 셀프 어텐션은 시퀀스 길이가 긴 경우에 그 이점이 부각될 수 있는데, 본 연구와 같이 짧은 시퀀스 길이를 가진 데이터는 BiGRU에 어텐션 메커니즘을 추가한 것만으로도 충분히 효과적인 처리가 가능한 것으로 보인다. 게다가 본 연구와 같이 학습 데이터셋이 소규모인 경우에는 멀티 헤드 어텐션을 사용하여도 다양한 문맥적 학습 정보가 수집되지 않아, 트랜스포머의 학습 능력이 충분히 발휘되지 못할 가능성도 존재한다. 한편, [표 3-11]에 제시된 추론 시간은 각 모형이 한 개의 문장을 예측하는데 걸린 평균 시간을 나타낸다. 추론 시간 측정 결과에서는 BiGRU+Attention이 Transformer-EPU보다 2배 빠른 처리 속도를 나타냈다. 구체적으로는, 10만 개의 문장을 하나씩 처리할 때 BiGRU+Attention은 약 5분, CNN-multichannel은 약 6분 40초, Transformer-EPU는 약 10분의 시간이 소요되었다. 따라서 감정 분류 성능뿐만 아니라 처리 속도도 함께 고려한다면 BiGRU+Attention 모형이 본 연구의 최적 모형으로 판단된다.

[표 3-11] 분류기 모형 비교

구분	BiGRU+Attention	CNN-multichannel	Transformer-EPU
F1 Score	85.48%	85.21%	85.59%
log loss	0.3999	0.4455	0.3945
추론 시간	0.003초	0.004초	0.006초

3.3 비교 대상 모형: BERT를 활용한 전이 학습

본 절에서는 본 연구의 비교지표 작성을 위해 BERT 계열의 PLM을 활용한 감정 분류 모형의 구축 과정을 다룬다. 구체적으로 3.3.1과 3.3.2에서는 전이 학습과 BERT의 방법론을 살펴보고, 3.3.3에서는 한국어 말뭉치로 학습된 BERT 계열의 PLM들을 검토하여 이들의 전이 학습 수행 결과를 평가한다. 그리고 이후 Ⅳ장에서는 선정된 비교 대상 모형과 본 연구 모형으로 EPU 지수를 작성하고, 그 결과를 비교한다.

3.3.1 전이 학습(Transfer Learning)

3.3.1.1 전이 학습의 정의

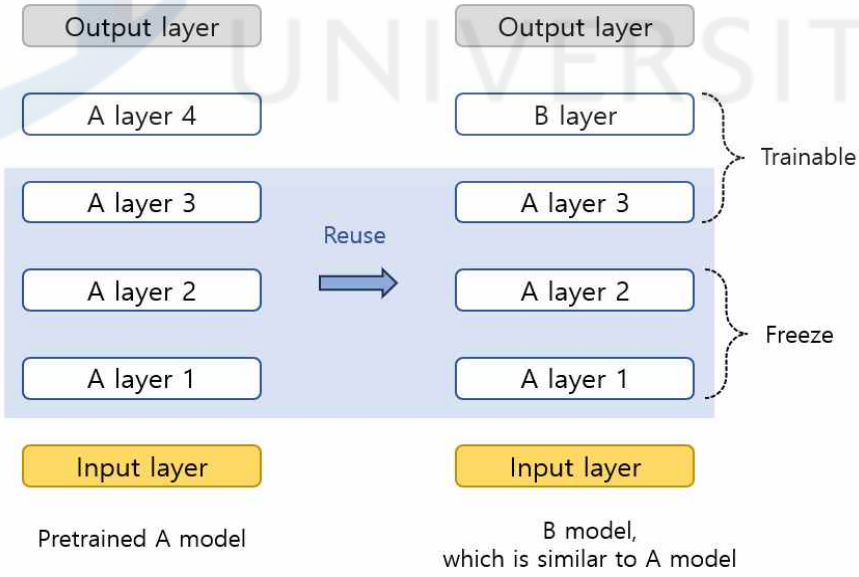
Goodfellow et al.(2016)의 저서 *DEPP LEARNING*에서는 전이 학습을 다음과 같이 정의한다.

"한 환경에서 학습된 것을 다른 환경의 일반화를 개선하기 위해 활용하는 것(the situation where what has been learned in one setting is exploited to improve generalization in another setting)"

이 정의를 이해하기 위해 아래 [그림 3-29]와 같은 전이 학습의 한 예시를 살펴보자. 해당 그림에서 A 모형은 택시 이미지 인식을 위해 사전 학습된 딥러닝 모형이고, B 모형은 버스 이미지 인식을 위해 새로 구축하는 딥러닝 모형이라 하자. 딥러닝 모형의 주요 특징 중 하나는 학습이 계층적으로 이루어진다는 것이다. 즉, A 모형의 전반부 층들은 택시의 창문이나 바퀴의 형태 같은 일반적인(general) 특징들을 추출하는 반면, 후반부 층들은 해당 특징들의 크기나 모양과 같은 구체적인(specific) 특징들을 추출하여 모형을 학습시킨다. 이는 B 모형도 마찬가지다. B 모형 역시 처음부터 학습을 수행한다면 전반부 층들은 버스의 창문이나 바퀴의 형태 같은 일반적인 특징들을 추출하

고, 후반부 층들은 버스 이미지의 구체적인 특징들을 추출하게 된다. 따라서 일반적인 특징을 추출하는 전반부 층들은 두 모형의 학습 데이터가 비슷한 환경을 가질수록 서로 유사한 특징을 추출할 가능성이 크다. 전이 학습은 이를 이용하여 학습 데이터가 부족한 B 모형에게 A 모형의 전반부 층들을 전이하는 학습 방법이다. 이때 B 모형은 전이 받은 층들을 그대로 사용할 수도 있고, 아래 그림과 같이 일부 층의 가중치를 새로운 학습 데이터에 맞게 재학습하여 사용할 수도 있다. 전자의 방법을 특징 추출(feature extraction)이라 하며, 후자의 방법을 미세 조정(fine-tuning)이라 한다. B 모형의 학습 데이터가 A 모형의 학습 데이터와 매우 유사하다면 특징 추출로도 높은 성능을 달성할 수 있지만, 두 학습 데이터의 분포가 다르거나 더 높은 성능이 필요하다면 미세 조정을 통해 B 모형의 성능을 최적화할 수 있다. 다만 두 학습 데이터의 유사성이 매우 낮은 경우에는 전이 받은 층 전체를 재학습할 수도 있으므로 미세 조정은 높은 컴퓨팅 비용이 발생할 수 있음에 유의해야 한다.

[그림 3-29] 전이 학습 예시



전이 학습의 좀 더 공식화된 정의는 Pan and Yang(2009)에서 찾을 수 있다. Pan and Yang(2009)은 도메인과 태스크, 그리고 전이 학습을 주변 확

률(marginal probabilities)과 조건부 확률(conditional probabilities)을 사용하여 정의한다. 먼저 도메인 D는 식 (3-49)와 같이 특징 공간(feature space) $\mathcal{X}$와 특징 공간상에서의 주변 확률 분포 $P(X)$로 구성된다.

$$D = \{\mathcal{X},\ P(X)\} \tag{3-49}$$

여기서 X는 특징 공간 내에 인스턴스 집합, 즉 $X = \{\mathbf{x} | \mathbf{x}_i \in \mathcal{X},\ i = 1, \cdots, n\}$를 의미한다. 예를 들어 문서 분류 문제를 다루는 경우, $\mathcal{X}$는 모든 용어 벡터의 공간이 되며, $\mathbf{x}_i$는 어떤 문서에 해당하는 i번째 용어 벡터, X는 훈련 샘플과 연관된 확률 변수(random variable)가 된다. 다음으로 태스크 T는 식 (3-50)과 같이 레이블 공간(label space) $\mathcal{Y}$와 음함수(implicit function) $f(\bullet)$로 구성된다.

$$T = \{\mathcal{Y},\ f(\bullet)\} \tag{3-50}$$

$f(\bullet)$는 $\mathbf{x}_i \in \mathcal{X}$와 $y_i \in \mathcal{Y}$의 쌍으로 이루어진 학습 데이터에 의해 학습되며, $\mathbf{x}$가 주어졌을 때 $f(\mathbf{x})$는 $\mathbf{x}$에 대응되는 레이블 y를 예측하기 위해 사용된다. 따라서 $f(\bullet)$는 확률론적 관점에서 조건부 확률 분포 $f(Y|X)$로도 나타낼 수 있으며, 이때 확률 변수 Y는 레이블 집합 $Y = \{y | y_i \in \mathcal{Y}, i = 1, \cdots, n\}$가 된다. 즉, 태스크가 이진 문서 분류라면 $\mathcal{Y}$는 모든 레이블 집합 {Treu, False}이고, y_i는 True와 False 중 하나이다. 그리고 이러한 조건부 확률 분포를 이용하면 소스 도메인(source domain), 타깃 도메인(target domain), 소스 태스크(source task), 타깃 태스크(target task)는 다음과 같이 정의될 수 있다.

소스 도메인: $D_S = \{\mathcal{X}_S,\ P(X_S)\}$, 타깃 도메인: $D_T = \{\mathcal{X}_T,\ P(X_T)\}$

소스 태스크: $T_S = \{\mathcal{Y}_S,\ P(Y_S|X_S)\}$, 타깃 태스크: $T_T = \{\mathcal{Y}_T,\ P(Y_T|X_T)\}$

이때 소스 도메인의 학습 데이터 쌍은 $\{(x_{S_1}, y_{S_1}), \cdots, (x_{S_{nS}}, y_{S_{nS}})\}$, 타깃 도메인의 학습 데이터 쌍은 $\{(x_{T_1}, y_{T_1}), \cdots, (x_{T_{nT}}, y_{T_{nT}})\}$이며, x_{S_i}, y_{S_i}, x_{T_i}, y_{T_i}는 각

각 $x_{S_i} \in \mathcal{X}_S$, $y_{S_i} \in \mathcal{Y}_S$, $x_{T_i} \in \mathcal{X}_T$, $y_{T_i} \in \mathcal{Y}_T$이다. 그리고 일반적으로 nT는 nS보다 훨씬 작은 값을 갖는다($nT \ll nS$). 마지막으로 이상의 도메인과 태스크의 정의를 통해 전이 학습은 다음과 같이 정의된다.

전이 학습: 소스 도메인 D_S와 소스 태스크 T_S, 그리고 타깃 도메인 D_T와 타깃 태스크 T_T가 주어졌을 때, 전이 학습은 D_S와 T_S의 전이 가능한 지식을 사용하여 T_T의 예측 함수 $f_T(\bullet)$의 성능을 향상하는 것을 목표로 한다. 이때 $D_S \neq D_T$이거나 $T_S \neq T_T$이다.

전통적인 머신러닝과 딥러닝에서는 도메인과 태스크의 환경이 같지만 ($D_S = D_T$, $T_S = T_T$), 전이 학습은 위 정의와 같이 도메인이 다르거나 태스크가 다른 문제를 다루게 된다($D_S \neq D_T$ 또는 $T_S \neq T_T$). 따라서 전이 학습의 발생 가능한 시나리오는 도메인과 태스크의 구성 요소에 따라 네 가지로 구분될 수 있다. 첫 번째 시나리오는 소스 도메인과 타깃 도메인의 특징 공간이 다른 경우, 즉 $\mathcal{X}_S \neq \mathcal{X}_T$인 상황이다. 예를 들어 문서 분류 태스크에서 이 시나리오는 소스 도메인 문서와 타깃 도메인 문서가 서로 다른 언어로 기술되어 두 문서의 용어 특징이 다른 상황을 의미한다. 두 번째 시나리오는 소스 도메인과 타깃 도메인의 주변부 확률 분포가 다른 경우, 즉 $\mathcal{X}_S = \mathcal{X}_T$이지만 $P(X_S) \neq P(X_T)$인 상황이다. 이 시나리오는 일반적으로 도메인 적응(domain adaption)이라고도 알려져 있으며, 문서 분류 예시에서는 두 문서가 서로 다른 주제를 다루고 있는 상황을 말한다. 세 번째 시나리오는 소스 태스크와 타깃 태스크의 레이블 공간이 다른 경우, 즉 $\mathcal{Y}_S \neq \mathcal{Y}_T$인 상황이다. 문서 분류 예시에서 이 시나리오는 두 문서의 분류 클래스가 서로 다른 상황을 말하며, 보통 레이블 공간이 다르면 사전 및 조건부 확률 분포도 다르기 때문에 세 번째 시나리오는 다음 네 번째 시나리오와 함께 발생하는 경우가 많다. 네 번째 시나리오는 소스 태스크와 타깃 태스크의 조건부 확률 분포가 다른 경우, 즉 $P(Y_S|X_S) \neq P(Y_T|Y_S)$인 상황이다. 문서 분류 예시에서는 두 문서의 클래스가 불균형 경우를 말한다. 이 시나리오는 실제로 매우 흔히 발생하는 상황

으로 오버샘플링(over-sampling), 언더샘플링(under-sampling) 또는 SMOTE (Chawla et al., 2002)와 같은 접근법이 널리 사용된다.

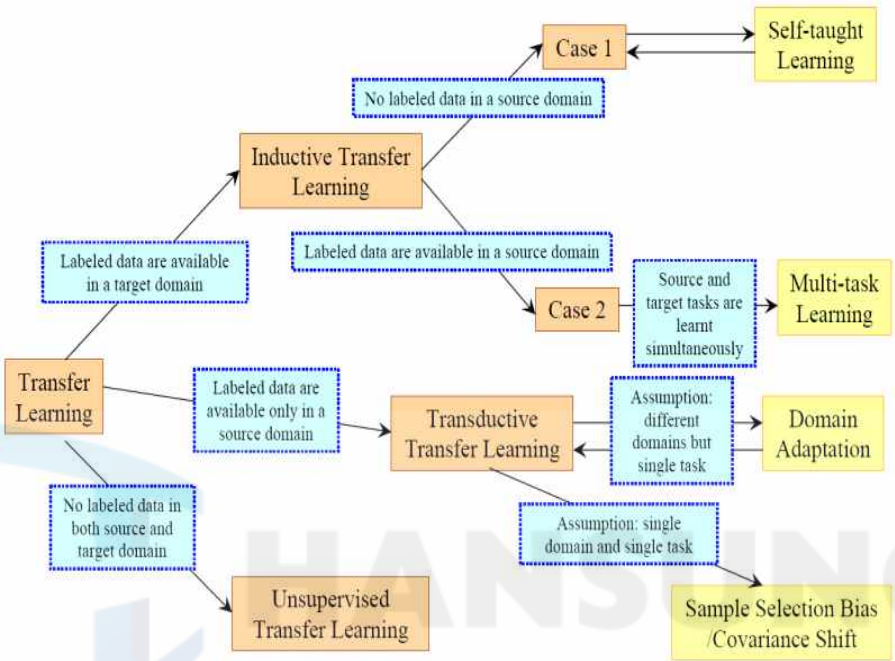
3.3.1.2 전이 학습의 범주화

한편 전이 학습은 여러 가지 범주화 방식이 제시되어왔다. 그중 Pan and Yang(2009)에서는 다음 [그림 3-30]과 같이 귀납적 전이(inductive transfer), 변환적 전이(transductive transfer), 비지도 전이(unsupervised transfer)라는 세 가지 범주로 전이 학습을 분류한다. 먼저 귀납적 전이는 소스 태스크와 타깃 태스크가 다르고($T_S \neq T_T$), 타깃 도메인에 레이블 정보가 있는 상황을 다룬다. 그리고 이는 소스 도메인에 레이블 정보가 있는 경우와 없는 경우로 더 세분화할 수 있으며, 전자의 경우는 다중 태스크 학습(multi-task learning), 후자의 경우는 자율 학습(self-taught learning)으로 분류할 수 있다.[97] 다음으로 변환적 전이는 소스 태스크와 타깃 태스크는 같으나 도메인이 다르고, 타깃 도메인에는 레이블 정보가 없으나 소스 도메인에는 레이블 정보가 있는 상황을 다룬다. 이 경우에는 소스 도메인과 타깃 도메인의 특징 공간이 다른 경우($\mathcal{X}_S \neq \mathcal{X}_T$)와 특징 공간은 같지만 주변 분포가 다른 경우 ($P(X_S) \neq P(X_T)$)로 세분화할 수 있으며, 후자의 경우는 도메인 적응(domain adaption), 샘플 선택 편향/공분산 이동(sample selection bias/covariance shift)의 요구 사항과 유사하다. 마지막으로 비지도 전이는 소스 태스크와 타깃 태스크가 다르고, 소스 도메인과 타깃 도메인 모두 레이블 데이터가 없는 경우, 즉 $T_S \neq T_T$이면서 $\mathcal{Y}_S$와 $\mathcal{Y}_T$가 관찰되지 않는 상황을 다룬다. 이 경우는 앞에 두 범주에 비해 비교적 관련 연구가 적은 편이며, 주로 클러스터링, 차원 축소 또는 생성 모델과 같은 비지도 기법을 사용하여 레이블 정보 없이

97) 타깃 도메인에 레이블 데이터가 없다는 것은 소스 도메인과 타깃 도메인의 레이블 공간이 다르다는 것을 의미한다. 따라서 이는 레이블 공간이 다르다는 전제하에(타깃 도메인에 레이블이 없거나 소스 도메인과 타깃 도메인의 레이블이 다른 경우) 모형을 학습시키는 자율 학습 설정과 매우 유사하다. 하지만 다중 태스크 학습은 소스 태스크와 타깃 태스크를 동시에 학습하여 두 태스크의 성능을 모두 향상하는 것이 주된 목적이므로 소스 태스크에서 학습한 지식을 사용하여 타깃 태스크의 성능을 높이는 것에만 집중하는 귀납적 전이와 약간의 차이가 있다.

타깃 도메인 데이터를 학습하는 것에 중점을 둔다.

[그림 3-30] 전이 학습 분류(Pan & Yang, 2009)



Pan and Yang(2009)이 도메인 간의 유사성과 레이블 환경을 기준으로 전이 학습을 분류하였다면 Weiss et al.(2016)은 레이블 환경과 상관없이 도메인의 유사성을 기준으로 전이 학습을 분류한다. 이 경우 전이 학습은 동종 전이 학습(homogeneous transfer learning)과 이종 전이 학습(heterogeneous transfer learning)이라는 두 가지 범주로 분류된다. 동종 전이 학습은 특징 공간과 레이블 공간이 같은 경우, 즉 $\mathcal{X}_S=\mathcal{X}_T$, $\mathcal{Y}_S=\mathcal{Y}_T$인 상황을 다룬다. 따라서 동종 전이 학습은 주변 확률 분포가 다르거나($P(X_S)\neq P(X_T)$), 주변부 확률 분포가 다른 경우($P(Y_S|X_S)\neq P(Y_T|Y_S)$)를 해결하고자 한다. 반면, 이종 전이 학습은 특징 공간이나 레이블 공간이 다른($\mathcal{X}_S\neq\mathcal{X}_T$ 또는 $\mathcal{Y}_S\neq\mathcal{Y}_T$) 상황을 다룬다. 이 경우에는 특징 공간이나 레이블 공간 간의 차이를 해소한 뒤 분포(주변 또는 조건부)의 차이를 수정해야 하는 동종 전이 학습 문제로 변환하여 문제를 해결하게 된다.

위에서 언급한 두 가지 주요 범주화 방식은 전이 학습을 적용하고 연구할 수 있는 다양한 상황들을 보여준다. 그리고 이러한 범주들은 전이되는 내용에 따라 다음 [표 3-12]와 같이 네 가지 접근법을 적용할 수 있다. 먼저 인스턴스 기반(instance-based) 접근법은 소스 도메인의 인스턴스를 타깃 도메인에 재사용하거나, 소스 도메인의 인스턴스와 타깃 도메인의 인스턴스를 결합하여 학습하는 방식이다. 예를 들어 소스 도메인에서 레이블 된 데이터를 학습한 경우, 해당 데이터에 가중치를 부여하고 이를 재조정하여(re-weight) 타깃 도메인 학습에 사용하게 된다. 그리고 이때 가중치는 주로 도메인 간의 주변 분포 차이를 줄이는 데 중점을 두게 된다. 두 번째로 특징 기반(feature-based) 접근법은 소스 도메인에서 학습된 특징을 타깃 도메인에서도 사용할 수 있도록 변환하여 새로운 특징 표현을 만드는 방식이다. 이는 다른 접근법들과 달리 비지도 전이 학습과 이종 전이 학습에도 적용할 수 있으며, 특히 이종 전이 학습은 특징 공간 간의 차이를 줄이는 데 중점을 두므로 특징 기반이 적절한 접근법으로 활용될 수 있다. 또한 특징 기반은 특징을 변환하는 방식에 따라 비대칭 특징 변환(asymmetric feature transformation)과 대칭 특징 변환(symmetric feature transformation)으로 세분화할 수 있다. 비대칭 특징 변환은 소스 또는 타깃 도메인의 특징을 다른 도메인에 특징에 맞게 변환하는 방식이고, 대칭 특징 변환은 공통된 잠재 특징 공간(latent feature space)을 찾은 다음 소스와 타깃 특징을 모두 새로운 특징 표현으로 변환하는 방식이다. 세 번째로 매개변수 기반(parameter-based) 접근법은 소스 도메인에서 학습된 모형의 매개변수를 타깃 도메인 모델에 초기값으로 사용하는 학습 방법이다. 앞서 [그림 3-30]에 제시된 층 전이(layer transfer) 또는 층 미세 조정(layer fine-tuning)과 같은 전이 학습이 매개변수 기반 접근법에 해당한다. 이는 태스크의 일부 매개변수 또는 모형의 하이퍼 파라미터의 사전 분포가 공유된다는 가정하에 작동된다. 즉, 매개변수 기반의 기본 개념은 소스 도메인에서 훈련된 모형이 잘 정의된 구조를 학습하고, 두 태스크가 관련이 있다면 이 구조를 타깃 모형으로 전이할 수 있다는 것이다. 이처럼 매개변수(가중치)를 공유한다는 개념은 딥러닝 모형에서 널리 사용되고 있는 방법이며, 일반적으로 딥러닝 모형에서 가중치를 공유하는 방법은 소프트 가중치 공유

(soft weight sharing)와 하드 가중치 공유(hard weight sharing)로 나눌 수 있다. 전자는 각 태스크에 대해 별도의 모형을 학습시키면서 모형 간의 가중치가 유사해지도록 손실 함수에 제약(예: L2 정규화 항)을 추가하는 방법이고, 후자는 모형의 일부 층을 여러 태스크에 공유하여 이전에 학습된 가중치가 모형의 초기 가중치로 사용되도록 하는 방법이다. 마지막으로 관계 기반(relational-based) 접근법은 소스 도메인과 타깃 도메인의 공통적 관계나 구조적인 정보를 전이하는 학습 방식이다. 즉, 전이될 지식은 데이터 간의 관계가 되며, 이에 따라 관계 기반은 독립 항등 분포(independent and identically distributed, iid)가 아닌 데이터를 처리하게 된다. 예를 들어 소셜 네트워크 데이터를 처리하기 위해 그래프 신경망(Graph Neural Networks, GNN)과 같은 관계 기반 접근법을 사용할 수 있다.[98]

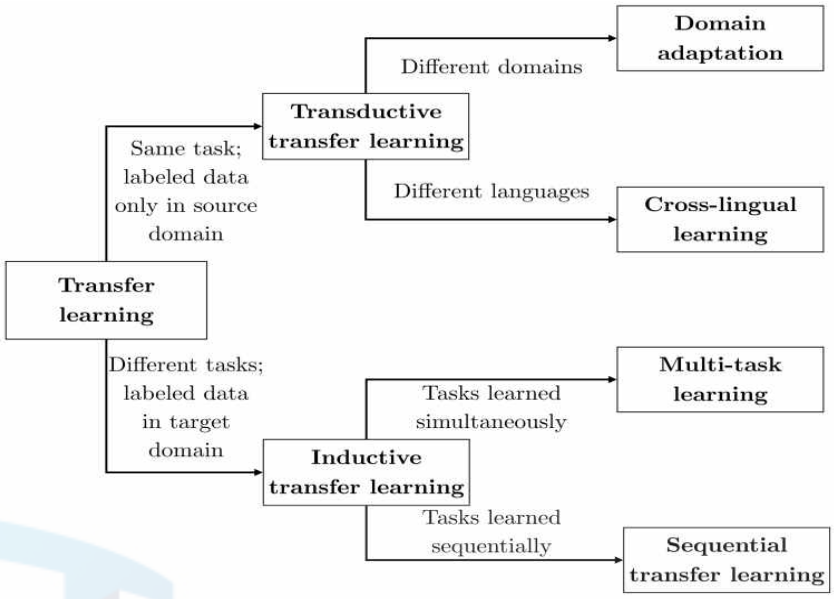
[표 3-12] 전이 학습의 접근법 및 범주화

Problem categorization		Solution categorization			
		Instance based	Feature based	Parameter based	Relational based
Label-setting categorization	Inductive transfer	✓	✓	✓	✓
	Transductive transfer	✓	✓		
	Unsupervised transfer		✓		
Space-setting categorization	Homogeneous transfer	✓	✓	✓	✓
	Heterogeneous transfer		✓		

98) 소셜 네트워크 데이터는 사용자들이 서로 관계를 맺고 있는 형태의 데이터로 각 사용자의 게시물은 독립적이지 않지만, 친구나 팔로워 등의 관계를 통해 상호작용과 영향을 주고 받는다. 또한 그래프 신경망은 그래프 구조 데이터를 처리하고 분석하기 위해 설계된 신경망 모형으로 그래프는 노드(node)와 엣지(edge)로 구성되며, 노드는 개별 데이터 포인트를 엣지는 데이터 포인트 간의 관계를 나타낸다. 예를 들어 소셜 네트워크 데이터로 그래프 신경망을 사용한다면 사용자들은 노드로, 사용자 간의 관계를 엣지로 하는 그래프가 생성되며, 각 노드(사용자)에는 해당 사용자의 게시물을 통해 추출된 임베딩 벡터를 추가할 수 있다. 그리고 그래프 신경망은 이러한 그래프의 구조를 통해 노드 간의 정보를 주고받으며 학습을 수행한다.

이 외에도 Ruder(2019)는 태스크 및 도메인의 특성, 그리고 학습 순서에 따라 자연어 처리를 위한 전이 학습의 분류 체계를 다음 [그림 3-31]과 같이 제시한 바 있다. 해당 분류는 Pan and Yang(2009)의 범주화를 자연어 처리 시나리오에 맞게 수정한 것으로 주요 변경 사항은 다음 세 가지이다. 첫째, 다국어 학습(cross-lingual learning)이라는 자연어 처리의 특정 범주화를 추가하였다. 이는 주로 여러 개의 언어가 하나의 임베딩 공간을 공유하는 유니버셜 임베딩(universal embedding)을 학습하는 것에 중점을 두며, BERT의 다국어 버전인 Multilingual BERT(mBERT)나 XLM-RoBERTa 등이 이 범주에 속한다고 할 수 있다. 둘째, 샘플 선택 편향/공분산 이동 범주를 생략하고 이를 도메인 적응의 특별한 경우로 간주하였다. 예를 들어 영화평 감정 분류 모형을 상품평 감정 분류 모형에 활용한다면 두 모형의 태스크는 같으나 도메인이 다르므로 도메인 적응 범주에 해당하고, 한/영 번역 모형을 한/일 번역 모형에 활용한다며 두 모형의 태스크는 같으나 도메인의 언어가 다르므로 교차 언어 학습 범주에 해당한다. 마지막으로 다중 태스크 학습과의 차이를 강조하기 위해 순차적 전이 학습(sequential transfer learning)이라는 범주를 도입하였다. 순차적 전이 학습은 소스 태스크와 타깃 태스크가 다르면서 학습이 순차적으로 수행되는 상황으로 정의된다. 예를 들어 두 태스크 T_1과 T_2가 각각 $[t_1, t_2]$와 $[t_3, t_4]$ 구간 동안 훈련되고, $t_1 < t_2$, $t_3 < t_4$이고, $t_1 \leq t_3$라면, 다중 태스크 학습은 일반적으로 $t_1 = t_3$, $t_2 = t_4$이다. 다시 말해 두 태스크 모두 동시에 훈련이 시작되고 마친다. 반면 순차적 전이 학습은 첫 번째 태스크의 학습이 종료된 후 두 번째 태스크의 학습이 시작된다. 즉, $t_2 < t_3$이다. 다중 태스크 학습과 비교했을 때 순차적 전이 학습은 주로 세 가지 시나리오에서 유용하다: (a) 두 태스크의 데이터를 동시에 사용할 수 없는 경우; (b) 소스 태스크의 학습 데이터가 타깃 태스크의 학습 데이터보다 훨씬 많은 경우; (c) 여러 타깃 태스크에 대한 전이가 필요할 경우. 즉, 본 연구와 같이 사전 학습된 단어 임베딩 모형을 활용하거나, BERT나 GPT와 같이 사전 학습된 언어 모델을 활용하여 특정 태스크(예: 감정 분류, 기계 번역, 질의 응답 등)에 전이하는 경우가 모두 순차적 전이 학습에 해당한다.

[그림 3-31] 자연어 처리에서의 전이 학습 분류(Ruder, 2019)



순차적 전이 학습은 사전 훈련(pre-training) 단계와 적응(adaption) 단계로 구분된다(Ruder, 2019). 사전 훈련 단계는 대규모 데이터를 통해 초기 가중치를 학습하는 과정으로 최근 컴퓨터 비전 및 자연어 처리 분야에서는 대부분 자기 지도 학습(self-supervised learning)이 이를 위해 사용되고 있다. 자기 지도 학습은 데이터 자체에서 레이블을 생성하여 학습을 수행하는 방법으로 레이블 데이터가 필요 없다는 점에 있어서는 비지도 학습(unsupervised learning)의 하위 범주로 분류될 수 있지만, 지정된 레이블에 대해 성능을 최적화한다는 점에 있어서는 지도 학습(supervised learning)과 밀접한 관련이 있다.[99] 특히 최근 몇 년 동안 뛰어난 결과를 보여주고 있는 구글의 BERT나 Open AI의 GPT, 또는 Meta의 LLAMA(Large Language Model Meta AI) 등이 모두 자기 지도 학습으로 사전 훈련된 모형들이며, 이중 BERT의 사전 훈련 과정은 아래 3.3.2에서 좀 더 구체적으로 살펴보게 된다.

99) 지도 학습과 자기 지도 학습은 둘 다 손실함수를 통해 성능을 최적화하기 위해 기준값(레이블이 지정된 훈련 데이터)이 필요하지만, 자기 지도 학습은 지도 학습과 달리 레이블이 지정되지 않은 데이터에서 기준값을 추론할 수 있도록 설계된다. 또한 비지도 학습과 자기 지도 학습은 둘 다 레이블이 없는 데이터의 내재적 상관관계나 패턴을 파악하여 학습을 수행하지만, 비지도 학습은 자기 지도 학습과 달리 기존에 알려진 기준값과 비교하여 결과를 측정하지 않는다(예: 클러스터링, 이상 탐지 및 차원 축소 등).

적응 단계에서는 사전 훈련된 모형을 특정 태스크에 맞게 조정하고 최적화하는 과정이 이루어진다. 즉, 앞서 언급된 특징 추출과 미세 조정이 적응 단계에 해당한다. 특징 추출은 사전 학습된 표현들이 고정되는 반면, 미세 조정은 재학습 과정을 통해 사전 학습된 표현들이 다운스트림 태스크에 맞게 업데이트된다. 따라서 특징 추출은 컴퓨팅 비용이 낮다는 장점이 있지만, 일반적으로 미세 조정을 수행하는 것이 더 높은 성능을 나타내는 경우가 많다. 다만 학습 데이터가 너무 적거나 테스트 데이터에 많은 OOV 단어가 포함되어 있다면 미세 조정이 오히려 낮은 성능을 보일 수 있으며(Dhingra et al., 2017), Peters et al.(2019)에서는 소스 태스크와 타깃 태스크가 유사할 때는 미세 조정이 더 나은 성능을 보이지만 소스 태스크와 타깃 태스크가 멀리 떨어져 있을 때는 특징 추출이 더 나은 성능을 보인다는 것을 발견한 바 있다.

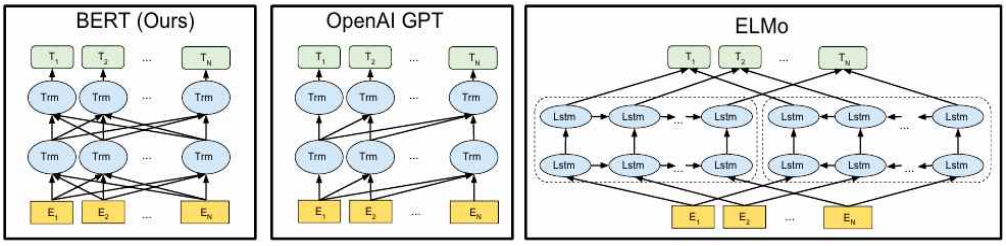
3.3.2 BERT(Bidirectional Encoder Representation from Transformers)

트랜스포머의 등장 이후, 2018년 구글에서는 트랜스포머의 인코더를 기반으로 한 BERT(Devlin et al., 2018)가 제시되었다. BERT는 등장과 함께 수많은 자연어 처리 태스크에서 최고 성능을 보여주었고, 기존 SOTA(state of the art) 모형이던 ELMO(Embeddings from Language Model)(Peters et al., 2018)의 성능을 크게 앞지르는 한편, 미세 조정만으로도 범용적인 태스크에서 높은 성능을 달성할 수 있다는 가능성을 보여주었다. 이에 따라 자연어 처리 분야에서는 PLM을 활용한 전이 학습이 주된 방법론으로 자리 잡게 되었고, 최근까지도 소형 PLM을 확장(모형 및 데이터 크기의 확장)한 대형 PLM(Large Language Model, LLM)들이 계속해서 개발되는 추세이다.

BERT의 성공 비결은 성능이 검증된 트랜스포머 블록을 사용했을 뿐만 아니라 모형의 속성이 양방향을 지향한다는 점에 있다. 아래 [그림 3-32]는 Devlin et al.(2018)이 BERT 이전의 기존 모형인 GPT-1(Radford et al., 2018)과 ELMO의 차이를 시각화한 것이다. GPT-1은 트랜스포머의 인코더-디코더 구조가 LSTM보다 좋은 성능을 얻자, RNN 계열의 신경망에서 탈피하여 트랜스포머의 디코더를 12개 쌓은 구조로 PLM을 구축하였다. 그러나

이는 주어진 시퀀스를 가지고 그다음 단어를 예측하는 방식으로 사전 훈련되기 때문에 문맥의 양방향 성질을 고려할 수가 없다. 다시 말해, 언어 모형은 이전 단어들을 가지고 다음 단어를 맞추는 방식으로 훈련되는데, 이를 양방향 언어 모형(Bidirectional Language Model, BiLM)으로 설계하면 맞춰야 하는 단어를 모형에게 미리 알려주는 꼴이 된다. ELMo는 이를 보완하여 순방향 언어 모형과 역방향 언어 모형을 따로 학습시키는 양방향 언어 모형을 구축하였다. 즉, 아래 그림과 같이 순방향 LSTM과 역방향 LSTM을 따로 학습시킨 후에 이를 결합하는 방식으로 문맥적 임베딩을 생성한다. 이러한 임베딩은 문맥에 따라 임베딩 벡터값이 달라지므로 기존의 단어 임베딩이 해결할 수 없었던 다의어 문제를 해결할 수가 있다. 하지만 이 역시도 결국 단방향 구조로 모형을 학습시키기 때문에 완전한 양방향성을 학습한다고 보기는 어렵다. BERT는 이러한 기존 언어 모형들의 한계를 보완하기 위해 마스크드 언어 모형(Masked Language Model, MLM)이라는 새로운 구조의 언어 모형을 제시하였다. MLM은 기존 언어 모형의 학습 방식에서 벗어나, 일단 문장 전체를 모형에 알려주고, 빈칸(mask)에 해당하는 단어를 예측하는 방식으로 학습을 해보자는 아이디어다. 예를 들어 '한국의 [MASK]가 1.5%P 상승했다'라는 문장을 주고 [MASK]에 들어갈 단어를 맞추게 하는 방식이다.[100] BERT는 이러한 사전 훈련 방법으로 문맥의 양방향성을 얻을 수 있었고, 이를 통해 GPT-1과 ELMo보다 높은 성능을 나타낼 수 있었다.

[그림 3-32] BERT, GPT-1, ELMO의 아키텍처(Devlin et al., 2018)



BERT는 MLM과 다음 문장 예측(Next Sentence Prediction, NSP)이라는 두 가지 방식으로 사전 훈련을 수행한다. 먼저 MLM은 앞서 언급하였듯이

100) 이 경우 정답이 가려져 있으므로 문장 전체를 모형에 알려주어도 정답을 알 수가 없다.

입력으로 들어온 텍스트의 15%를 무작위로 마스킹(masking)한 뒤, 모형에게 이 가려진 단어들을(masked words) 예측하게 하는 방식으로 학습이 수행된다. 좀 더 정확하게는 15%의 단어들을 전부 가리는 것이 아니라, 아래와 같은 세 가지 규칙에 따라 변경되어 학습이 수행된다.

• 마스킹 단어 중 80%는 [MASK]로 변경되고, 모형은 해당 위치의 정답 단어가 무엇인지를 예측한다. 예: 남자가 [MASK]에 들어갔다. → 가게
• 마스킹 단어 중 10%는 무작위로 단어가 변경되고, 모형은 해당 위치의 정답 단어가 무엇인지를 예측한다. 예: 남자가 [컴퓨터]에 들어갔다. → 가게
• 마스킹 단어 중 10%는 그대로 두고, 모형은 해당 위치의 정답 단어가 무엇인지를 예측한다. 예: 남자가 [가게]에 들어갔다. → 가게

그리고 이러한 학습을 통해 모형은 아래와 같은 효과를 기대할 수 있게 된다.

• [MASK] 빈칸을 채우는 과정에서 앞뒤 문맥을 읽을 수 있게 된다.
• '남자가 [가게]에 들어갔다'와 '남자가 [컴퓨터]에 들어갔다'를 비교하면서 주어진 문장의 의미상 또는 문법상 비문인지 아닌지를 가려낼 수 있게 된다.
• 모형은 어떤 단어가 마스킹 될지 모르므로 문장 내 모든 단어 사이의 의미적 또는 문법적 관계를 세밀하게 살피게 된다.

BERT의 또 다른 사전 훈련인 NSP는 두 개의 문장이 이어지는 문장인지 아닌지를 예측하는 방식으로 수행된다. 이때 NSP의 학습 데이터는 정답 예제와 거짓 예제가 각각 50%씩 구성되며, 정답 예제는 실제 이어지는 두 개의 문장을 뽑아 정답 레이블(IsNext)을 부여하고, 거짓 예제는 서로 다른 두 개의 문서에서 문장을 하나씩 뽑아 거짓 레이블(NotNext)을 부여한다. 그리고 모든 예제가 완성되면 정답과 거짓 예제는 동등한 확률로 샘플링된다. 또한 NSP는 학습 태스크가 너무 쉬워지는 것을 막기 위해 max_num_tokens(최대 토큰 수)를 정의한 후, 학습 데이터의 90%는 max_num_tokens와 사용자가 정의한 max_sequence_length(시퀀스의 최대 길이)가 같아지게 설정하고, 나머지 10%는 max_num_tokens가 max_sequence_length보다 작아지도록 설정

한다. 그리고 이전에 뽑은 두 문장의 단어 수가 max_num_tokens를 넘지 못할 때까지 두 문장 중 단어 수가 많은 쪽의 문장 맨 앞이나 맨 뒤 단어를 50%의 확률로 하나씩 제거한다. BERT의 이러한 학습 과정은 아래와 같은 효과를 기대할 수 있게 해주며, 이를 통해 질의 응답(Question Answering, QA)이나 자연어 추론(Natural Language Inference, NLI)과 같이 문장 간의 이해관계가 중요한 다운스트림 태스크의 성능을 높일 수 있다.

• 두 문장이 이어진 문장인지 아닌지를 반복적으로 학습하여 문장 사이의 의미 관계를 이해할 수 있다.
• 문장 맨 앞 또는 맨 뒤의 단어를 일부 삭제한 덕분에 문장 내에 일부 성분이 없어도 전체 의미를 이해할 수 있다.
• 학습 데이터의 10%는 max_sequence_length보다 짧은 길이로 구성되어 있어, 학습 데이터에 짧은 문장이 포함되어도 성능이 크게 떨어지지 않는다.

한편 BERT의 기본 구조는 트랜스포머의 인코더를 여러 개 쌓아 올린 형태이다. Devlin et al.(2018)에서는 BERT_BASE와 BERT_LARGE 두 가지 버전을 제시하였는데, BASE 버전에서는 12개의 인코더를, LARGE 버전에서는 24개의 인코더를 쌓아 올렸다. 이 외에 두 버전의 어텐션의 헤드 수(h), d_{model}의 크기, 총 파라미터 수는 다음 [표 3-13]과 같으며, 두 버전은 모두 약 30,000개의 단어 집합과 33억 개(위키피디아 25억 단어 + BooksCorpus 8억 단어)의 학습 데이터로 사전 훈련되었다. 이때 BASE 버전의 하이퍼파라미터는 GPT-1과 동일한데, 이는 GPT-1과의 성능 비교를 위해 설정된 값이며, BERT가 세운 기록들은 대부분 LARGE 버전을 통해 이루어졌다.

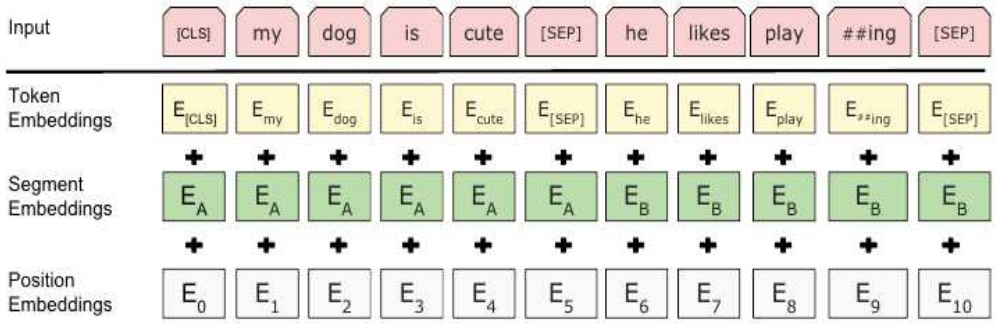
[표 3-13] BERT의 크기

	BERT_BASE	BERT_LARGE
Encoder layers(N)	12	24
Attention heads(h)	12	16
Vector dimension(d_{model})	768	1,024
Total parameters	110M(1억 1천만)	340M(3억 4천만)

[101]BERT의 트랜스포머 블록은 아래 [그림 3-33]과 같이 세 개의 임베딩을 입력으로 받는다. 먼저 첫 번째 임베딩인 토큰 임베딩(token embeddings)에서는 WordPiece를 통해 분할된 각 토큰에 고유한 임베딩 벡터가 할당된다. 이때 WordPiece는 BERT에서 사용되는 토크나이저를 말하며, 이는 단어 집합에 없는 단어를 부분 단어(subword)로 분리한다는 특징을 갖는다. 예를 들어 BERT의 단어 집합에 playing이라는 단어는 없고, play, ##ing라는 부분 단어들이 존재한다면 playing은 play, ##ing로 분리된다. 또한 BERT는 이 외에도 문장의 시작을 나타내는 토큰 [CLS], 문장의 종결을 나타내는 토큰 [SEP], 마스크 토큰 [MASK], 패딩 토큰 [PAD]의 네 가지 스페셜 토큰을 사용한다. 다만 여기서는 [CLS]와 [SEP]를 문장의 시작과 끝을 나타낸다고 설명하였지만, 실제로는 문장이 아니라 문단이나 문서가 될 수도 있다. 두 번째 임베딩인 세그먼트 임베딩(segment embeddings)은 두 개의 문장을 구분하기 위해 사용된다. 즉, 아래 그림과 같이 첫 번째 문장의 모든 토큰에는 A 세그먼트 임베딩이 더해지고, 두 번째 문장의 모든 토큰에는 B 세그먼트 임베딩이 더해지는 방식이며, 두 개의 문장만 고려되므로 사용되는 임베딩 벡터도 두 개가 된다. 다만 이 역시도 다운스트림 태스크에 따라 두 개의 문장은 두 개의 문서 또는 텍스트가 될 수 있으며, 본 연구와 같은 감정 분류 태스크에서는 한 개의 문장만 분류하므로 세그먼트 임베딩은 하나의 임베딩만 더해지게 된다. 세 번째 임베딩인 위치 임베딩(position embedding)은 토큰의 위치 정보를 학습하기 위한 임베딩으로 문장 내에서 각 토큰의 위치에 따라 고유한 임베딩이 할당된다. 이에 대해서는 3.2.1.3에서 설명한 바 있으며, BERT의 경우 최대 512개의 토큰까지 위치 임베딩이 지원된다. 이러한 세 개의 임베딩이 요소별(element-wise)로 더해지고 나면, 각각의 벡터는 층 정규화와 드롭아웃이 적용되어 트랜스포머 블록의 입력 행렬로 사용된다. 예를 들어 아래 그림과 같이 11개의 토큰이 있는 문장이라면 입력 행렬의 크기는 11×1024가 되며, 토큰, 세그먼트, 위치 벡터를 만들 때 참조하는 행렬은 사전 훈련 과정에서 다른 학습 파라미터와 함께 업데이트된다.

101) 여기서부터는 BERT_LARGE의 구조를 기준으로 설명한다.

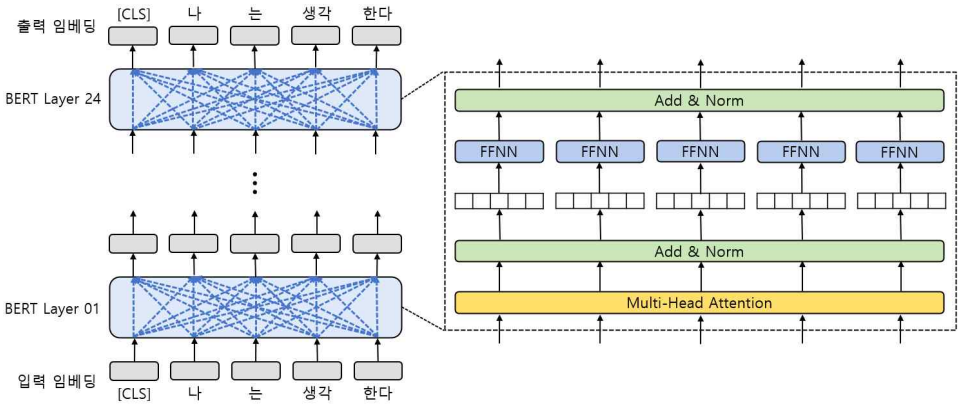
[그림 3-33] BERT의 입력(Devlin et al., 2018)



아래 [그림 3-34]는 BERT의 내부 구조를 보여준다. BERT는 1,024차원의 [CLS], [나], [는], [생각], [한다]라는 다섯 개의 임베딩 벡터를 입력받으면 내부적인 연산을 거쳐 입력 임베딩과 같은 1,024차원의 다섯 개의 임베딩 벡터를 출력한다. 즉, BERT의 초기 입력 당시에는 단순히 임베딩 층을 지난 임베딩 벡터였지만, BERT 층을 지나고 나서는 문장 내에 모든 문맥 정보를 가진 임베딩 벡터로 변환된다. 아래 그림에서는 각 단어가 문장 내에 모든 단어 벡터들을 참고하고 있다는 사실을 파란색 점선의 화살표로 표현하였다. 그리고 이처럼 각 단어가 모든 단어를 참고하기 위해 수행되는 연산이 바로 트랜스포머의 인코더이다. 다시 말해 BERT의 24개 층에서는 전부 멀티 헤드 셀프 어텐션과 위치별 완전 연결 FFNN의 연산이 수행되며, 각 층의 연산이 끝난 출력 임베딩은 바로 다음 층의 입력 임베딩으로 사용된다. 다만 원조 트랜스포머(Vaswani et al., 2017)의 위치별 완전 연결 FFNN에서는 활성화 함수로 ReLU가 사용되었지만, BERT는 GPT-1을 따라 GELU(Gaussian Error Linear Units)(Hendrycks & Gimpel, 2016)[102]를 사용했다는 차이가 있다. 또한 BERT 학습에 사용된 warmup_steps와 최적화 알고리즘은 각각 10,000과 Adam(학습률: 10^{-4}, β_1: 0.9, β_2:0.999 L2 가중치 감소: 0.01)이며, 각 층에 사용된 드롭아웃 비율은 0.1, 학습 손실은 MLM의 평균 손실과 NSP의 평균 손실의 합이다.

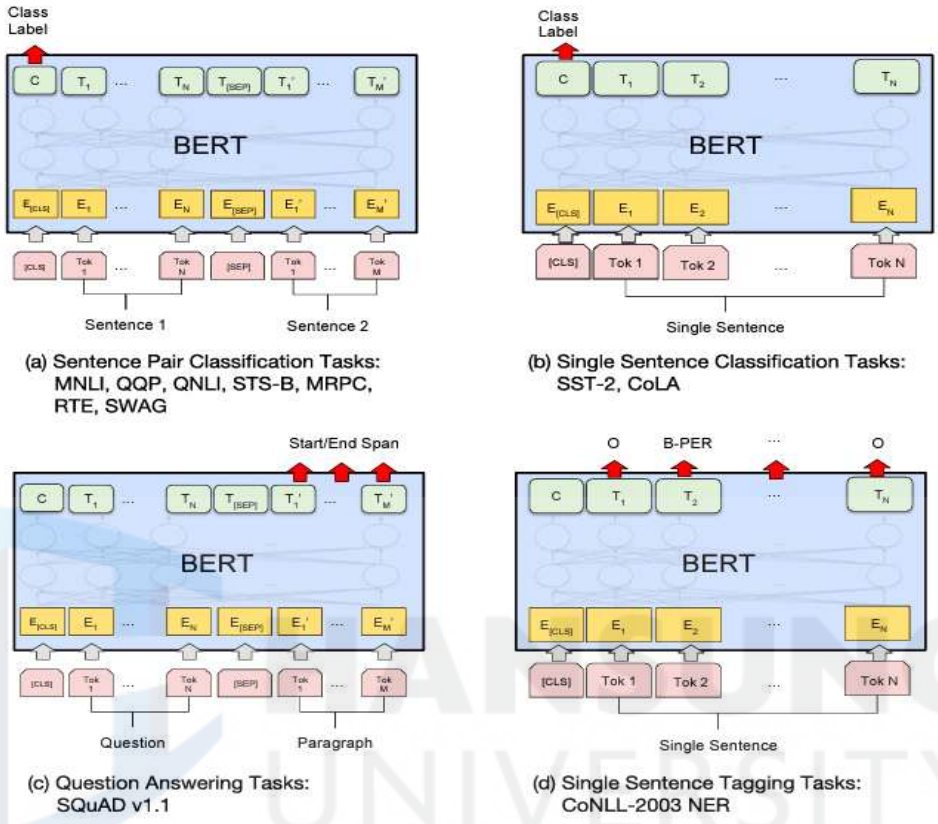
102) GELU는 표준 가우시안 분포의 누적분포함수(Cumulative Distribution Functions, CDF)를 사용하는 비선형 활성화 함수이다. 이는 ReLU와 같이 입력을 0 또는 원래 값으로 자르지 않고, 0 주위의 값을 부드럽게 변환해 주어 ReLU의 경직성을 완화해 준다는 장점이 있다.

[그림 3-34] BERT의 내부 구조



사전 훈련이 완료된 BERT 모형을 우리가 원하는 태스크에 활용하기 위해서는 해당 태크스의 데이터로 추가 학습을 진행하는 미세 조정 과정이 수행되어야 한다. BERT의 미세 조정은 BERT 모형 위에 추가적인 층을 올린 구조로 수행되며, 아래 [그림 3-35]와 같이 태스크 유형에 따라 네 가지 아키텍처로 구분될 수 있다. 첫 번째 유형인 (a) 텍스트 쌍에 대한 분류 태스크는 NLI와 같이 두 텍스트의 논리적 관계나 연관성을 파악하기 위해 수행되는 태스크이다. 이 경우 모형은 [SEP] 토큰으로 분리되는 두 개의 텍스트를 입력으로 받으며, 내부 연산을 통해 모든 토큰의 정보가 내포된 [CLS] 토큰으로 분류 문제를 풀게 된다. 따라서 첫 번째 유형은 출력층이나 분류 예측을 위해 설계되는 층들이 모두 [CLS] 토큰 위에 위치한다. 두 번째 유형인 (b) 하나의 텍스트에 대한 분류 태스크는 본 연구의 감정 분류 태스크처럼 하나의 텍스트로 분류 예측을 수행하는 유형이다. 이 경우 역시 [CLS] 토큰으로 분류 문제를 풀지만, 입력된 텍스트가 두 개가 아닌 하나라는 것에 차이가 있다. 세 번째 유형인 (c) QA 태스크는 질문(question)과 본문(paragraph)의 두 가지 텍스트를 입력으로 받은 뒤 본문의 일부를 추출하여 질문에 대한 답을 구하는 태스크이다. 이 경우에는 본문의 토큰들로 예측을 수행하므로 출력층이 본문 토큰들 위에 위치한다. 마지막 유형인 (d) 하나의 텍스트에 대한 태깅 태스크는 품사 태깅이나 개체명 인식과 같은 태스크들을 말한다. 이 경우에는 입력 텍스트가 하나이며, 출력층은 입력 텍스트의 각 토큰 위에 위치한다.

[그림 3-35] BERT의 미세 조정 아키텍처(Devlin et al., 2018)



3.3.3 비교 대상 모형 구축 결과

BERT의 성공 이후, 한국에서도 대량의 한국어 말뭉치를 학습한 다양한 PLM들이 공개되었다. 그중 대표적으로 사용되는 BERT 계열의 PLM은 아래 [표 3-14]와 같다. KorBERT는 한국전자통신연구원(ETRI)에서 개발한 최초의 한국어 BERT 모형으로 교착어인 한국어의 특성을 반영한 형태소 기반 버전과 WordPiece를 사용한 버전이 있다. 두 버전은 모두 뉴스와 백과사전 등에서 추출한 23GB의 말뭉치(형태소 47억 개)로 학습되었으며, 단어 집합의 크기는 30,349개(형태소)와 30,797개(WordPiece), 파라미터 크기는 100M(1억)이다. KoBERT는 SKT에서 개발한 모형으로 위키피디아에서 수집한 5,000만 개의 문장(54,000만 개의 단어)으로 학습되었다. 동 모형은 데이터(위키피

디아) 기반으로 학습된 토크나이저(SentencePiece)를 사용한다는 특징이 있으며, 단어 집합의 크기는 8,002개, 파라미터의 크기는 92M(9,200만)이다. HanBERT는 투블럭AI에서 공개한 모형으로 일반 및 특허 문서 70GB의 데이터로 학습된 모형이다. 자체 개발한 토크나이저(Moran)를 사용한다는 특징이 있으며, 단어 집합의 크기는 54,000, 파라미터의 크기는 128M(12,800만)이다.[103] KLUE-BERT는 벤치마크 데이터인 KLUE(Korean Language Understanding Evaluation)(Park et al., 2021)[104]의 베이스라인을 위해 개발된 모형으로 모두의 말뭉치[105], 나무위키, 뉴스 크롤링, CC-100-Kor[106], 청와대 국민 청원에서 추출한 63GB의 데이터로 학습되었다. 형태소 기반의 부분 단어 토크나이저(morepheme-based subword tokenizer)[107]를 사용하며, 단어 집합의 크기는 32,000개, 파라미터의 크기는 111M(11,100만)이다.

[표 3-14] BERT 계열 PLM 비교

PLM	학습 데이터	토크나이저	단어 집합 크기	파라미터 크기
KorBERT (ETRI)	뉴스 및 백과사전 (25GB)	Morpheme, WordPiece	30,349 (Morpheme), 30,797 (WordPiece)	110M
KoBERT (SKT)	위키피디아 (50M 문장)	SentencePiece	8,002	92M
HanBERT (투블럭AI)	일반 및 특허 문서 (70GB)	Moran	54,000	128M
KLUE-BERT (KLUE)	5개의 한국어 말뭉치 (63GB)	Morpheme-based subword	32,000	111M

103) 그러나 해당 모형은 현재 공개가 중단된 상태로 보인다.

104) 한국어 PLM의 평가 데이터셋을 구축하기 위해 31명의 공동 연구진이 참여한 오픈 프로젝트이다. 총 8개의 태스크에 대해 평가할 수 있는 데이터셋으로 구성되어 있다.

105) 국립국어원이 배포한 한국어 말뭉치 모음으로 문어체로 된 텍스트(뉴스 및 서적)와 구어체로 된 텍스트 등이 포함되어 있다.

106) CC-Net(Wenzek et al., 2020)을 이용한 대규모 다국어 웹 크롤링 말뭉치로 KLUE-BERT는 이 중 한국어 부분을 사용한다.

107) Mecab 형태소 분석기를 통해 텍스트를 형태소로 분리한 뒤, BPE(Byte Pair Encoding)(Sennrich et al., 2016)를 적용하여 최종 어휘를 얻는 방식이다.

이상의 PLM들은 위키 문서, 일반 뉴스, 웹 페이지 등과 같은 범용적인 말뭉치(general corpus)로 학습되었기 때문에 범용적인 도메인에 대해서는 높은 성능을 보이지만, OOD(Out-of-Distribution)에 해당하는 도메인에는 취약한 면모를 보일 수 있다. 특히 금융, 법률, 의료 등과 같은 도메인은 해당 도메인에서만 사용되는 전문 용어나 고유명사가 많아 OOV 문제가 빈번히 발생하며, 이로 인해 모형은 전문 용어 및 고유명사에 대한 이해 부족, 유사어의 관계나 문서 내 수치에 대한 이해 부족 등의 문제가 발생할 수 있다.

이러한 문제를 방지하기 위해서는 해당 도메인에 특화된 PLM을 구축해야 하지만, 문제는 사전학습에 들어가는 컴퓨팅 비용이다. Devlin et al.(2018)에서는 BASE 버전의 경우 4개의 Cloud TPU(총 16개의 TPU 칩)를, LARGE 버전의 경우 16개의 Cloud TPU(총 64개의 TPU 칩)를 사용하여 4일간 학습시켰으므로, BASE 버전은 $1,712($4.5(TPU v2의 시간당 비용)*4(TPU 수)*24(시간)*4(학습일)), LARGE 버전은 $6,912($4.5(TPU v2의 시간당 비용)*16(TPU 수)*24(시간)*4(학습일))의 사전학습 비용이 발생한다.[108] 더욱이 언어 모형은 매개변수 규모가 일정 수준을 넘을 때 모형의 성능이 크게 향상될 뿐만 아니라 일련의 복잡한 태스크를 수행할 경우 소규모 PLM에서는 볼 수 없었던 새로운 능력(emergent ability)이 나타나기 때문에(Zhao et al., 2023) 최근에는 일반 기업조차 감당하기 힘들 정도의 LLM이 개발되고 있다.

따라서 사전학습 비용이 부담되는 개인 연구자들은 직접 모형을 구축하기보다는 공개된 모형 중 자신의 연구 도메인과 유사한 모형을 찾아 연구를 수행하는 경우가 많은데, 다행히도 최근에는 이러한 수요에 맞춰 특정 도메인에 특화된 언어 모형을 개발하는 시도가 늘어나고 있다. 대표적으로 의료 분야는 Med-PaLM(Singhal et al, 2023), 법률 분야는 Legal-BERT(Chalkidis et al., 2020)가 있으며, 금융 분야는 FinBERT(Araci, 2019)와 최근 블룸버그에서 개발한 BloombergGPT(Wu et al., 2023)가 유명하다. 국내에서도 이처럼 특정 도메인에 특화된 언어 모형을 구축하려는 시도가 있었다. 그중 본 연구와 유사한 도메인을 가진 모형으로는 KB국민은행의 KB-BERT(김동규 외,

108) 참고로 근래 화두가 되었던 Chat GPT(GPT-3)는 1,750억 개의 파라미터를 가지고 있으며, 사전학습 비용만 무려 1,200만 달러에 달한다.

2022), 카카오뱅크의 KF-DeBERTa(전은광 외, 2023), 한국언론진흥재단의 KPF-BERT[109]가 있다. 이중 KB-BERT는 해당 모형 대신 BERT의 경량화 버전인 ALBERT(A Lite BERT)(Lan et al., 2020)로 학습된 모형만 공개되고 있으며, 이마저도 별도의 허가를 받아야 사용할 수 있기 때문에 본 연구에서는 깃허브와 허깅 페이스를 통해 자유롭게 사용할 수 있는 KPF-BERT와 KF-DeBERTa만 사용하여 전이 학습을 수행하였다.

KPF-BERT와 KF-DeBERTa의 주요 특징은 [표 3-15]에 제시되어있다. KPF-BERT는 2000년부터 2021년 8월까지의 빅카인즈 뉴스기사 약 4,000만 건으로 학습된 뉴스 도메인 특화 PLM이다. 해당 모형은 구글의 BERT를 기반으로 설계되었으며, 사용된 토크나이저는 WordPiece, 단어 집합의 크기는 32,000개[110]이다. KF-DeBERTa는 DeBERTa(He et al., 2020)[111]를 기반으로 구축한 금융 도메인 특화 PLM이다. 토크나이저는 형태소 인식 부분 단어(morpheme-aware subword)(Park et al., 2020)[112] 방식을 사용하며, 단어 집합의 크기는 128,000개이다. 또한, 학습에 사용된 말뭉치는 기존 한국어 PLM에서 사용되던 범용 도메인 말뭉치에 금융 관련 말뭉치가 추가된 구성으로, 전체 116.3GB의 중 약 28%(32.4GB)가 금융 관련 문서로 구성되어 있다. 구체적으로는 Common Crawal(35GB), 모두의 말뭉치(14.1GB), AHUB 말뭉치(11.3GB), 뉴스 댓글(12.5GB), 뉴스기사(4.6GB), 나무위키(4.8GB), 위키피디아(1.6GB)가 범용 도메인에 포함되며, 에프엔가이드에서 제공한 금융 리포트(5.9GB)와 금융 관련 뉴스기사(26.5GB)가 금융 도메인에 포함된다.[113]

109) https://github.com/KPF-bigkinds/KPF-BERT

110) 모두의 말뭉치, Common Crawl 등 5개의 말뭉치에서 약 20GB를 샘플링하여 구축하였다.

111) DeBERTa(Decoding-enhanced BERT with disentangled attention)는 마이크로소프트에서 제안된 모형으로 disentangled attention과 enhanced mask decoder라는 두 가지 방법을 사용하여 BERT의 성능을 개선한다. disentangled attention은 기존 BERT가 토큰 임베딩과 위치 임베딩의 합인 단일 벡터를 사용한 것과 달리, 토큰을 content와 position으로 분리하여 두 벡터의 조합(content-to-content, content-to-position, position-to-content)에 따라 어텐션 메커니즘을 적용하는 아이디어고, enhanced mask decoder는 MLM에서 마스킹 토큰을 예측할 때 절대적 위치(absolute position)도 함께 고려하자는 아이디어다.

112) KLUE-BERT와 마찬가지로 텍스트를 형태소로 분리한 뒤 BPE를 적용하는 방식이다.

113) 참고로 KB-BERT의 학습 말뭉치는 위키피디아, 뉴스, 웹 문서 등이 포함된 범용 도메인 말뭉치 50GB와 금융 상품 설명서 및 투자 리포트 등이 포함된 금융 도메인 말뭉치 40GB로 구성되며, 이때 금융 도메인 말뭉치는 검색 기반의 말뭉치 증강 과정을 수행한 용량이다.

[표 3-15] 도메인 특화 PLM 비교

구분	KPF-BERT	KF-DeBERTa
학습 데이터	빅카인즈 뉴스기사 4,000만 건	범용 도메인 말뭉치(83.9GB)[1], 금융 도메인 말뭉치(32.4GB)[2]
모형 아키텍처	BERT	DeBERTa
토크나이저	WordPiece	Morpheme-aware subword
단어 집합 크기	32,000	128,000

주: 1) Common Crawal(35GB), 모두의 말뭉치(14.1GB), AHUB 말뭉치(11.3GB), 뉴스 댓글(12.5GB), 뉴스기사(4.6GB), 나무위키(4.8GB), 위키피디아(1.6GB)
2) 금융 리포트(5.9GB), 금융 뉴스기사(26.5GB)

전이 학습에 사용된 미세 조정 아키텍처는 [그림 3-35]에 제시된 (b) 유형과 같다. 즉, 각 모형은 하나의 문장을 입력받은 뒤 내부 연산을 거쳐 [CLS] 토큰을 출력으로 내보내며, 해당 토큰은 완전연결층의 입력으로 들어가 감정 분류 태스크에 활용된다. 또한 학습에 사용된 최적화 알고리즘은 RAdam이며, 배치 크기, 에폭, 스텝 수는 각각 32, 20, 16,000으로 설정하였다. 모형 평가는 앞에 연구 모형과 마찬가지로 계층적 5-fold CV를 수행하여 다섯 개 평가 지표의 평균값으로 측정하였으며, 도메인 특화의 필요성을 확인하기 위해 KLUE-BERT 모형도 함께 평가를 진행하였다.

KLUE-BERT, KPF-BERT, KF-DeBERTa의 전이 학습 결과는 아래 [표 3-16]에 제시되어있다. 평가 결과에서 세 모형은 모두 90%가 넘는 높은 성능을 보였으며, KF-DeBERTa(92.52%), KPF-BERT(92.37%), KLUE-BERT(91.68%) 순으로 좋은 성능을 나타냈다. 다만 KPF-BERT와 KF-DeBERTa는 F1 스코어를 기준으로 성능 차이가 약 0.15%P밖에 나타나지 않았으며, 이 두 모형과 KLUE-BERT의 성능 차이도 각각 0.69%P와 0.84%P로 나타나 그 차이가 그리 크지 않은 것으로 나타났다. 이처럼 범용 도메인과 도메인 특화 모형의 성능 차이가 크지 않은 것은 사실 Araci(2019)와 전은광 외(2023)에서도 나타난 결과인데,[114] 이는 본 연구의 테스트셋이 소규모인 이유도 있겠지만, Zheng et al.(2021)과 본 연구에서 주장한 바와 같이 본 연구의 태스크 난이도가 생각보다 쉬운 것에서 기인한 결과로 해석된다. 또한, 뉴스기사

114) Araci(2019)에서는 FinBERT가 Vanilla BERT보다 1%의 높은 성능을 보였으며, 전은광 외(2023)에서는 KF-DeBERTa가 KLUE-RoBERTa보다 평균적으로 1.2%의 높은 성능을 보였다.

는 KLUE-BERT도 1,280만 건이 학습되었기 때문에 도메인 특화의 필요성이 비교적 적을 수 있으며, 본 연구에서는 특정 단어군으로 뉴스기사를 수집하므로 금융 부문보다 다른 부문의 뉴스기사들이 더 많이 수집되었을 가능성도 존재한다. 반면 추론 시간에서는 KLUE-BERT와 KPF-BERT가 비슷한 속도를 보였으며, KF-DeBERTa가 가장 느린 속도를 나타냈다. 구체적으로, 10만 개의 문장을 하나씩 처리할 때 KLUE-BERT와 KPF-BERT는 약 3시간 40분이, KF-DeBERTa는 약 4시간 40분이 소요되었다. 앞서 여러 번 언급하였듯이 본 연구에서는 추론 시간도 매우 중요한 고려 요소 중 하나이다. 따라서 KF-DeBERTa가 가장 높은 성능을 나타냈지만, 본 연구에서는 이와 비슷한 성능을 나타내면서 추론 시간도 빠른 KPF-BERT가 본 연구에 최적 모형이라 판단하였다.

[표 3-16] 비교 대상 모형 비교

구분[1)]	KLUE-BERT	KPF-BERT	KF-DeBERTa
정확도	91.7%	92.36%	92.53%
정밀도	91.81%	92.4%	92.58%
재현율	91.7%	92.36%	92.53%
F1 Score	91.68%	92.37%	92.52%
log_loss	0.2412	0.2183	0.2149
추론 시간	0.131초	0.135초	0.166초

주: 1) 정밀도, 재현율, F1 Score는 다중 레이블로 인해 평균값으로 측정되며, 이때 평균값은 레이블 불균형을 고려해 가중 평균으로 측정됨

한편, [표 3-17]은 본 연구 모형과 비교 대상 모형 간의 주요 차이를 정리한 것이다. 비교 대상 모형에 활용된 KPF_BERT는 본 연구 모형이 활용한 Word2Vec보다 문맥을 이해하는 능력이 더 우수하고, 사전 훈련에 사용된 학습 데이터의 규모도 약 35배가량 더 많다. 이러한 차이로 비교 대상 모형은 본 연구 모형보다 약 6.9%P 높은 F1 Score를 기록하고 있지만, 추론 시간은 본 연구 모형이 약 40배 이상 빠른 속도를 나타내고 있다. 실제 EPU 지수를 작성하는 과정에서도 본 연구 모형은 약 3시간 만에 전체 데이터를 처리한 반면, 비교 대상 모형은 배치 크기를 512로 설정하였음에도 약 10일 이상의

시간이 소요되었다. 따라서 컴퓨팅 자원이 부족하거나 실시간 분석이 필요한 환경에서는 본 연구 모형이 더 효율적일 것으로 판단된다. 다만, 본 연구 모형의 실용성을 입증하기 위해서는 본 연구의 주장을 실증적으로 검토하는 과정도 필요하다. 이를 위해 이후 Ⅳ장에서는 본 연구 모형과 비교 대상 모형으로 EPU 지수를 작성하고, 두 지수의 작성 결과를 실증적으로 비교하는 과정을 진행한다.

[표 3-17] 본 연구 모형과 비교 대상 모형 비교

구분	본 연구 모형	비교 대상 모형
학습 데이터	경제정책 뉴스기사 문장 2만 개	
사전 훈련에 사용된 말뭉치	경제정책 관련 뉴스기사 약 116만 건	빅카인즈에서 수집한 뉴스기사 약 4,000만 건
임베딩 모형	Word2Vec	KPF_BERT
F1 Score	85.48%	92.37%
추론 시간	0.003	0.135

HANSUNG UNIVERSITY

Ⅳ. 지수 작성 및 유용성 평가

4.1 경제정책 불확실성 지수의 작성

4.1.1 작성 방법

경제정책 불확실성 지수는 Ⅲ장에서 구축한 감정 분류 모형으로 뉴스기사의 각 문장을 긍정·부정·중립으로 분류한 뒤, 분류된 긍정·부정 문장의 비율을 계산하여 작성한다. 표준화는 지수의 평균과 분산을 이용해 평균이 100, 표준편차가 10이 되도록 산출하며, 표준화 구간과 지수의 작성 기간은 2005년~2007년의 수집 기사가 일 평균 20~22개밖에 되지 않은 것을 고려해 2008년부터로 한정한다.

구체적인 작성 방법은 다음과 같다. 먼저 특정 월 t에 대한 월별 EPU 지수 X_t를 다음 식 (4-1)과 같이 부정 문장의 비율과 긍정 문장의 비율의 차이로 산출한다.

$$(4\text{-}1)\quad X_t = \frac{\sum_{t \in M_t} N_t - \sum_{t \in M_t} P_t}{\sum_{t \in M_t} S_t}$$

여기서 N_t와 P_t는 특정 월 t에 부정과 긍정으로 분류된 문장의 수를 의미하며, S_t는 특정 월 t에 수집된 전체 문장의 수를 의미한다. 또한 M_t는 특정 월 t에 포함되는 모든 일자 집합에 대응되는 날짜 지표의 집합(index set)이다. 즉, 월별 EPU 지수는 일별로 수집된 뉴스기사들의 누적치를 기준으로 산출된다. X_t가 산출된 다음에는 X_t의 평균과 표준편차를 사용하여 표준화를 수행한다. 이는 다음 식 (4-2)와 같이 정의된다.

$$(4\text{-}2)\quad EPU\_small_t = \frac{X_t - \overline{X}}{S} \times 10 + 100$$

$$\text{, where } \overline{X} = \frac{1}{|\ T\ |}\sum_{t \in T} X_t, \quad S = \sqrt{\frac{1}{|\ T\ | - 1}\sum_{t \in T}(X_t - \overline{X})^2}$$

여기서 표준화 구간을 나타내는 T는 2008년 1월부터 해당 월까지의 모든 월을 포함하는 집합에 대응하는 날짜 지표 집합이다. 따라서 해당 월에 나타난 지수 값이 100보다 크다면 이는 불확실성이 과거 평균보다 비관적인 것을 의미하며, 100보다 작은 경우에는 불확실성이 과거 평균보다 낙관적인 것을 의미한다.

분기별 EPU 지수는 월별 EPU 지수와 같은 방식으로 작성되지만, 일별 EPU 지수는 변동성을 고려해 7일 동안 발간된 기사를 기준으로 작성된다. 예를 들어 N_t, P_t, T_τ가 각각 특정일 t에 수집된 긍정, 부정, 전체 문장의 수를 의미한다면 X_t는 다음 식 (4-3)과 같이 특정일 t부터 과거 7일간의 이동 합계(moving sum)로 산출된다.

$$(4\text{-}3)\quad X_t = \frac{\sum_{\tau=1}^{7} P_{t-\tau} - \sum_{\tau=1}^{7} N_{t-\tau}}{\sum_{\tau=1}^{7} S_{t-\tau}}$$

이때 표준화는 표준화 구간 T를 2008년 1월 1일부터 해당일까지의 모든 일자를 포함하는 날짜 지표의 집합으로 바꾸어 산출하며, 분기별 EPU 지수의 표준화는 T를 2008년 1분기부터 해당 분기까지의 모든 분기를 포함하는 날짜 지표의 집합으로 바꾸어 산출한다.

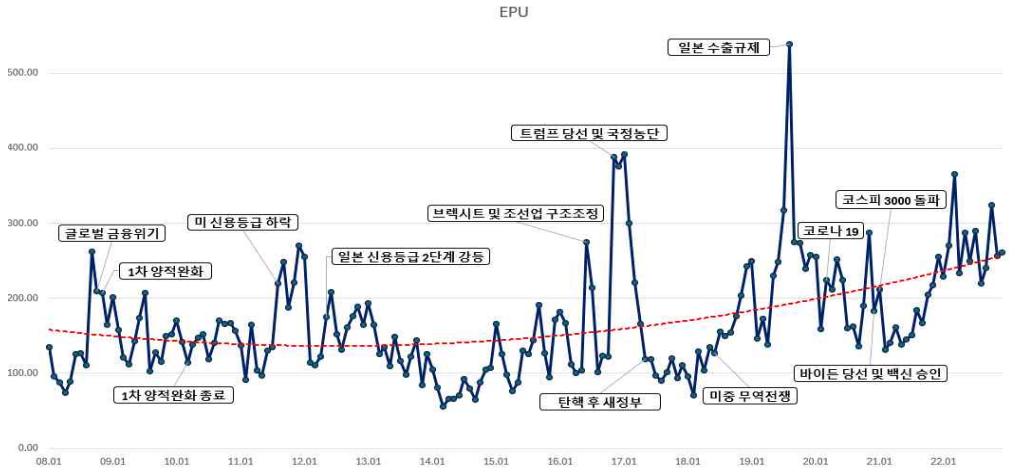
4.1.2 작성 결과

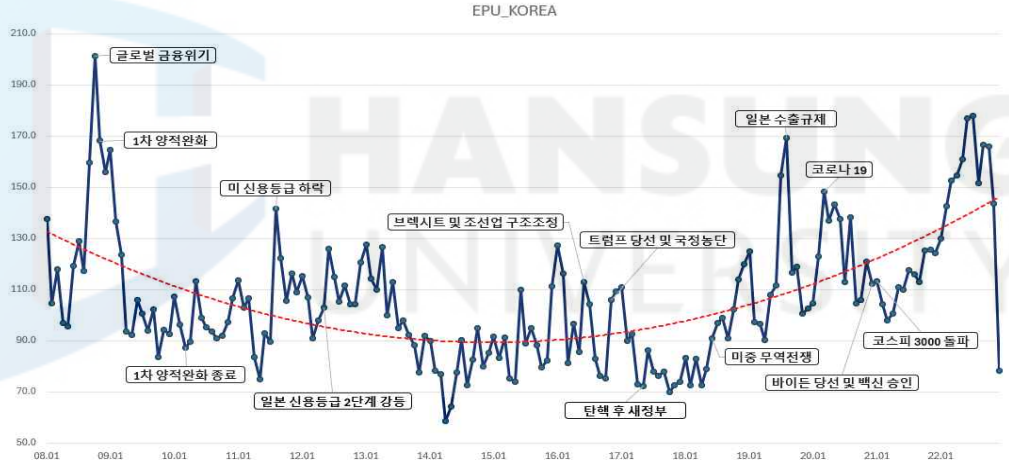
EPU 지수의 작성 결과는 [그림 4-1]과 같다. 여기서 EPU_small이 본 연구에서 작성한 EPU 지수를 보여주며, EPU와 EPU_KOREA는 각각 Baker et

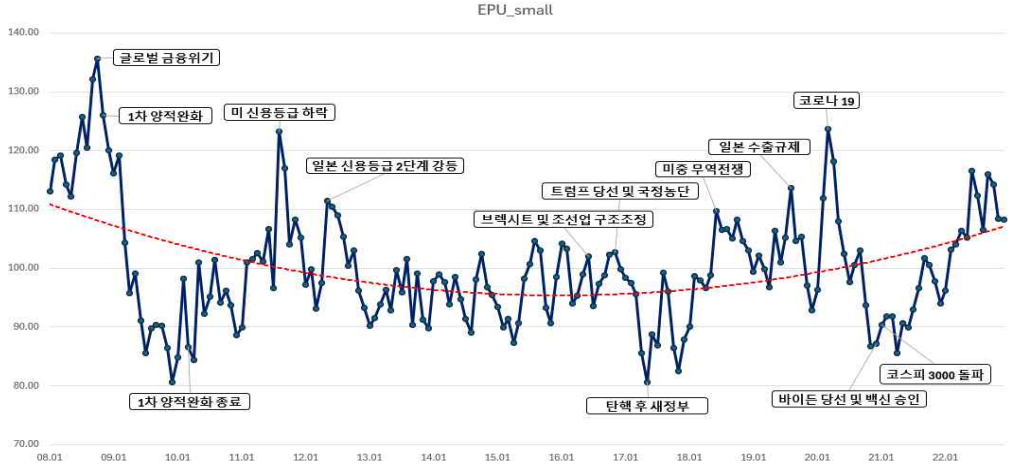
al.(2016)과 Cho and Kim(2023)이 작성한 EPU 지수를 보여준다. 먼저 EPU부터 살펴보면, EPU는 단어군과 언론사가 우리나라의 사정을 충분히 고려하지 못하다 보니 글로벌 금융위기와 같은 불확실성이 높은 시기에도 의미 있는 큰 값을 나타내지 못하고 있다. 게다가 EPU는 미국의 신용등급이 강등되었던 2011년 8월보다 국정농단 의혹 사건이 공론화되었던 2016년 11월의 불확실성을 더 높게 평가하고 있는데, 이 경우 2016년 11월의 불확실성은 경제적 불확실성보다는 정치적 불확실성으로 해석하는 것이 더 타당하다. 실제로 경제 불확실성과 민감하게 연결된 설비투자지수의 추이를 살펴보면, 미국 신용등급 강등 직후인 2011년 9월에는 설비투자지수가 전월대비 7.1% 감소하였지만, 국정농단 사건 직후인 2016년 12월에는 전월대비 2.7% 상승한 것을 확인할 수 있다. 반면, EPU의 단점을 보완한 EPU_KOREA는 글로벌 금융위기 때 매우 큰 값을 나타내고 있으며, 미국 신용등급이 하락하였던 시기에도 불확실성의 증대를 잘 포착하고 있다. 그러나 EPU와 EPU_KOREA는 U 단어군의 출현 빈도만으로 지수를 작성하기 때문에 불확실성의 다양한 표현을 고려하지 못하며, 불확실성 해소와 관련된 뉴스기사도 불확실성이 높은 것으로 해석하는 오류를 발생시킨다. 이로 인해 두 지수는 긍정적인 기사가 많이 발행됐던 시기에 뚜렷한 저점을 나타내지 못하고 있으며, 특히 최근 코로나 팬데믹이 발생했던 시기에는 불확실성 수준을 과소평가하는 모습을 보이고 있다. 이와 달리 감성 분석을 활용한 EPU_small은 코로나 19시기에 유의미한 큰 값을 나타내고 있으며, 코로나 백신 승인이 시작됐던 시기(2020년 12월)와 코스피가 처음으로 3000을 돌파했던 시기(2021년 1월)에는 불확실성 정도를 낮게 측정하고 있다. 또한, EPU_KOREA가 과소평가하였던 일본의 신용등급이 2단계 강등된 시기(2012년 5월)와 미중 무역전쟁이 본격화된 시기(2018년 7월)[115]에도 EPU_small은 뚜렷한 큰 값을 나타내고 있다.

115) 2018년 7월에 미국이 중국산 수입품에 대해 25% 관세를 부과하면서 본격적인 무역전쟁이 시작되었다.

[그림 4-1] EPU 지수 작성 결과

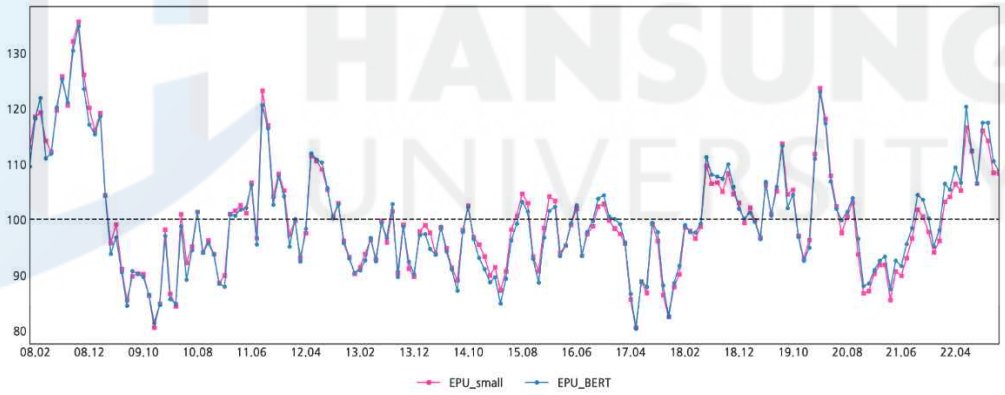






아래 [그림 4-2]는 본 연구 모형으로 작성한 EPU 지수(EPU_small)와 비교 대상 모형으로 작성한 EPU 지수(EPU_BERT)의 시계열 추이를 비교한 것이다. [표 3-17]에서 살펴보았듯이 본 연구 모형은 비교 대상 모형보다 약 6.9%P 낮은 성능을 보유하고 있어, EPU_small의 신뢰성이 상대적으로 떨어질 수 있다는 우려가 제기될 수 있다. 그러나 이러한 우려와 달리 아래 그림은 두 지수 간의 전반적인 추이가 매우 유사하며, 주요 변동 시점에서도 큰 차이를 보이지 않는다는 것을 시각적으로 보여준다. 이러한 결과는 본 연구의 주장을 뒷받침하는 결과로 해석될 수 있다. 다만, 일부 시점에서는 두 지수의 변동폭이 약간 차이가 나는 부분도 관찰되므로, 이러한 차이가 실제 EPU 지수를 활용하는 데 있어 유의미한 영향을 미치는지는 실증적으로 검토할 필요가 있다.

[그림 4-2] EPU_small과 EPU_BERT 비교



4.2 경제통계와의 상관관계 분석

본 연구에서 작성한 EPU 지수가 새로운 정보변수로서 유용성을 가지기 위해서는 기존 지표보다 선행성이 우수하고 거시경제변수와도 밀접한 관계가 있어야 할 것이다. 이를 확인하기 위해 본 절에서는 EPU 지수와 경제통계 간의 교차상관 분석을 수행한다.

교차상관은 한 시계열을 다른 시계열에 대해 시차(lag)를 두고 비교함으로써 두 시계열의 상관성을 평가하는 통계적 방법으로 다음 식 (4-4)와 같은 교차상관 함수에 의해 수행된다.

$$(4-4)\quad CCF(k) = \gamma_k = \frac{\sum_{t=1}^{N-k}(x_t - \bar{x})(y_{t+k} - \bar{y})}{\sqrt{\sum_{t=1}^{N}(x_t - \bar{x})^2 \sum_{t=1}^{N}(y_t - \bar{y})^2}}$$

여기서 $k=0$인 경우의 γ를 교차상관계수라 하며, $k \neq 0$이 아닌 경우의 γ를 시차상관계수라 한다. 예를 들어 교차상관계수 r_0가 0의 값을 가지면 두 변수는 서로 경기 중립적임을 의미하고, $r_0 > 0$인 경우에는 경기 순응을, $r_0 < 0$인 경우에는 경기 역행을 의미한다. 또한 시차상관계수 r_k는 그 값이 최대가 되는 시차 k에 따라 두 변수의 선·후행 관계를 판단한다. 즉, γ_k의 값이 최대가 되는 시차 k가 양수(+)이면 y_t는 x_t의 후행지표를, 음수(−)이면 y_t는 x_t의 선행지표를 의미하고, k가 0인 경우에는 두 변수가 동행지표임을 의미한다.

분석 대상이 되는 월별 경제통계 자료는 경제심리지수(ESI), 소비자심리지수(CCSI), 기업경기실사지수(BSI), 경기선행지수 순환변동치, 그리고 주식시장과 외환시장의 내재 변동성을 나타내는 VKOSPI, VIX, 환율 변동성이다. 이 중 환율 변동성은 월별 환율을 종속변수로 하여 GARCH(1,1) 모형으로 추정하였다. 분기별 경제통계 자료는 국민계정 상의 GDP, 내수, 민간소비, 설비투자, 재화 수출입이며, 모든 자료는 실질, 계절조정, 전기비 자료를 사용하였다. 분석 기간은 월별 통계의 경우 2008년 1월~2022년 12월, 분기 통계의

경우 2008년 1분기~2022년 4분기이며, 소비자심리지수는 작성 시점으로 인해 초기 시점이 2008년 7월부터이다.

월별 경제통계 지표와 EPU 지수 간의 교차상관분석 결과는 [표 4-1]과 같다. 표에서 괄호 안에 숫자는 최대상관시차 k를 의미하며, 각 지표에 대해 가장 높은 상관관계를 보인 EPU 지수는 음영으로 표시하였다. 분석 결과, EPU는 모든 지표에 대해 가장 낮은 상관관계를 나타냈으며, 경기 선행지수와는 양(+)의 상관관계를, 환율 변동성과는 음(-)의 상관관계를 나타내 경제이론과 부합하지 않는 결과를 보였다. EPU_KOREA는 VIX와 가장 높은 상관관계를 나타냈고, VKOSPI와도 가장 높은 상관관계를 나타낸 EPU_small과 비슷한 수준의 상관계수를 나타냈다. 다만, EPU_KOREA는 VIX와 VKOSPI에 대해 선행성을 보이진 않았다. EPU_small은 VIX를 제외한 모든 지표와 가장 높은 상관관계를 나타냈으며, 모든 지표에 대해 1~5개월 선행하는 것으로 분석되었다. 한편 EPU_BERT는 EPU_small과 최대상관시차가 동일하고, 상관계수 값에서도 큰 차이를 보이지 않았다.

[표 4-1] 월별 경제통계와의 교차상관분석 결과[1)]

경제지표	EPU	EPU_KOREA	EPU_small	EPU_BERT
ESI	-0.23(-5)	-0.38(-3)	-0.57(-5)	-0.55(-5)
BSI	-0.2(-1)	-0.35(-2)	-0.51(-5)	-0.5(-5)
CCSI	-0.4(-1)	-0.67(-1)	-0.74(-1)	-0.74(-1)
경기선행지수	0.19(0)	-0.39(-1)	-0.68(-3)	-0.64(-3)
VKOSPI	0.15(0)	0.58(0)	0.61(-1)	0.58(-1)
VIX	0.22(0)	0.63(0)	0.58(-1)	0.55(-1)
환율 변동성	-0.16(10)	0.39(-2)	0.52(-4)	0.5(-4)

주: 1) EPU는 Baker et al.(2016)이 작성한 EPU 지수, EPU_KOREA는 Cho and Kim(2023)이 작성한 EPU 지수, EPU_small은 본 연구 모형으로 작성한 EPU 지수, EPU_BERT는 비교 대상 모형으로 작성한 EPU 지수를 의미

분기별 경제통계와의 교차상관분석 결과는 [표 4-2]에 제시하였다. 여기서도 괄호 안에 숫자는 최대상관시차 k를 의미하며, 음영은 각 지표에 대해 가장 높은 상관관계를 보인 EPU 지수를 의미한다. 분기별 경제통계와의 분석 결과에서도 EPU_small과 EPU_BERT는 큰 차이를 보이지 않았으며, 모든 지표가 EPU_small과 가장 높은 상관관계를 보였다. 또한, EPU는 재화 수출과

재화 수입에 양(+)의 상관관계를, EPU_KOREA는 민간소비, 설비투자, 재화 수출에 양(+)의 상관관계를 나타내 경제이론에 부합하지 않는 결과를 보였다.

[표 4-2] 분기별 경제통계와의 교차상관분석 결과[1)]

경제지표	EPU	EPU_KOREA	EPU_small	EPU_BERT
경제성장률	-0.20(0)	-0.43(0)	-0.55(0)	-0.53(0)
내수	-0.23(-5)	-0.30(0)	-0.51(0)	-0.50(0)
민간소비	-0.23(-5)	0.27(-7)	-0.42(0)	-0.39(0)
설비투자	-0.27(-8)	0.37(-4)	-0.41(0)	-0.41(0)
재화 수출	0.23(-4)	0.21(-4)	-0.24(-1)	-0.24(-1)
재화 수입	0.17(6)	-0.31(0)	-0.42(-1)	-0.42(-1)

주: 1) EPU는 Baker et al.(2016)이 작성한 EPU 지수, EPU_KOREA는 Cho and Kim(2023)이 작성한 EPU 지수, EPU_small은 본 연구 모형으로 작성한 EPU 지수, EPU_BERT는 비교 대상 모형으로 작성한 EPU 지수를 의미

HANSUNG UNIVERSITY

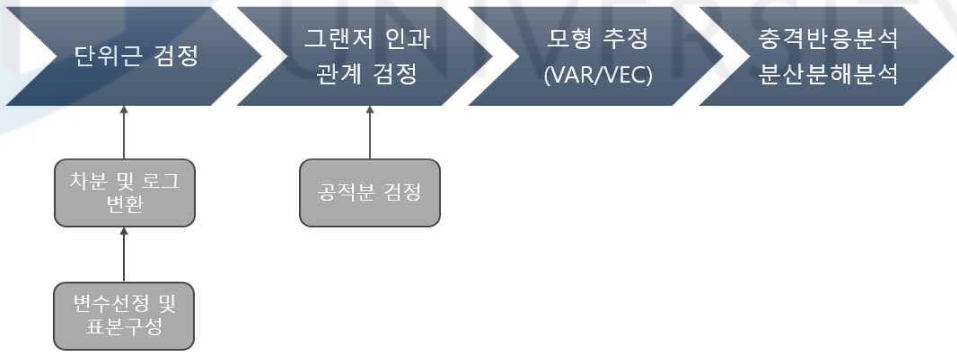
4.3 경제 불확실성이 거시경제에 미치는 영향 분석

EPU 지수의 유용성을 평가하는 또 다른 방법은 해당 지수의 외생성을 확인하고,[116] 거시경제변수와의 관계에 있어서 정책적 의미를 지니는지 확인하는 것이다. 이를 위해 본 절에서는 EPU 지수를 불확실성의 대리변수로 사용하여 불확실성이 거시경제변수에 미치는 영향을 분석하고, 분석 결과가 이론적 직관에 부합하는지를 검토한다.

4.3.1 분석 모형: 벡터자기회귀(Vector AutoRegression, VAR) 모형

불확실성이 거시경제에 미치는 파급효과를 분석하기 위해 본 연구에서는 VAR 모형을 활용한다. 동 모형의 분석과정은 [그림 4-3]과 같으며, 본 항에서는 이에 대해 상세히 살펴본다.

[그림 4-3] VAR 분석 절차



4.3.1.1 단위근 검정(Unit Root Test)

단위근 검정은 시계열 자료가 비정상적(non-stationary) 시계열인지, 정상적(stationary) 시계열인지를 판단하기 위해 수행되는 검정 방법이다. 이때 정

116) Jurado et al.(2015)에 따르면 불확실성 지수에 예측 가능한 부분이 포함되면 불확실성이 과도하게 측정되는 문제가 발생한다.

상성은 시계열의 평균, 분산, 공분산이 시간에 따라 변하지 않을 것을 요구하는 약한 정상성(weak stationary)을 말하며[117], 만약 시계열 자료가 추세, 계절성, 순환변동 등의 요인으로 인해 비정상성을 갖는다면 해당 시계열은 가성회귀(spurious regression)[118] 문제를 발생시키게 된다. 따라서 시계열 자료를 수집한 후에는 가장 먼저 단위근 검정을 수행하여 정상성 여부를 확인해야 하며, 해당 시계열이 비정상적 시계열로 확인될 경우, 차분, 자연로그, 계절조정 등의 방법을 거쳐 정상적 시계열로 변환해야 한다.

단위근 검정 방법으로는 주로 ADF 검정(Augmented Dickey-Fuller Test)과 PP 검정(Phillips-Perron Test)이 사용된다. ADF는 오차항 간의 시계열 상관을 고려한 검정 방법이며, PP는 오차항이 약 종속성(weakly dependent)을 갖거나 이분산성(heteroscedastistic)을 갖는 경우에 사용되는 비모수적 검정 방법이다. 본 연구에서는 이중 ADF 검정을 이용하여 단위근 검정을 수행하였다. 이는 다음 식 (4-5)~(4-7)을 통해 수행되며, 여기서 α는 상수항, δ_t는 선형 추세를 의미하고, ϕ가 1이라는 귀무가설을 채택하면 해당 시계열을 비정상성으로 판단하게 된다.

$$(4-5)\quad \Delta y_t = (\phi - 1) y_{t-1} + \epsilon_t$$

$$(4-6)\quad \Delta y_t = \alpha + (\phi - 1) y_{t-1} + \epsilon_t$$

$$(4-7)\quad \Delta y_t = \alpha + \delta_t + (\phi - 1) y_{t-1} + \epsilon_t$$

4.3.1.2 그랜저 인과관계 검정(Granger Causality Test)

VAR 모형은 변수들의 배열순서와 시차의 길이에 따라 분석 결과가 다르게 나타나므로 배열순서와 적정 시차를 결정하는 것이 중요한 문제가 된다.

시차의 길이는 변수 절약의 원칙에 따라 최소화해야 하며,[119] 오차항이

117) 시계열의 확률분포(결합밀도함수)가 시간이 경과 해도 전혀 변하지 않을 때 강한 정상성(strict stationary)을 지니게 된다. 하지만 강한 정상성을 지니는 경우는 매우 드물기 때문에 현실적으로는 약한 정상성을 정상성의 판단기준으로 삼게 된다.

118) 비정상적 시계열끼리 회귀분석을 수행할 때 발생하는 문제로 회귀식에 사용된 변수들이 실제로는 아무 관련이 없지만 우연히 높은 상관관계를 보이는 현상을 말한다.

백색잡음(white noise)이 될 때까지 늘려야 한다. 일반적으로는 AIC(Akaike Information Criterion) 또는 SC(Schwarz Criterion)가 최솟값을 보이는 곳에서 적정 시차를 결정하며, AIC가 SC보다 파라미터를 좀 더 과대 추정하는 경향이 있다고 알려져 있다.[120]

변수들의 배열순서는 이후에 진행되는 충격반응분석을 위해 이론적으로 외생성이 높은 순으로 변수를 나열해야 하지만, 외생성의 크기를 명확하게 비교할 수 있는 방법이 존재하지 않아 분석의 한계가 존재한다. 따라서 이론적 근거를 가진 방법은 아니나 그랜저 인과관계 검정을 통해 외생성 순위를 간접적으로 나열하는 방법이 종종 사용되고 있다.

그랜저 인과관계 검정은 두 변수 간의 그랜저 인과관계가 있는지를 판단하는 검정 방법으로 여기서 그랜저 인과관계가 있다는 것은 한 변수의 과거 수치가 다른 변수의 현재 수치를 설명하는 데 영향을 미친다는 것을 의미한다.[121] 즉, 다음 식 (4-8)을 만족할 때 x_t는 y_t를 그랜저 인과하지 않는다.

$$(4\text{-}8)\quad E(y_t|y_{t-j}, x_{t-i}) = E(y_t|y_{t-j})$$

여기서 $E(y_t|y_{t-j}, x_{t-i})$는 y_{t-j}와 x_{t-i}에 회귀하여 얻은 y_t의 추정치를 의미하며, $E(y_t|y_{t-j})$는 y_t를 y_{t-j}에 회귀하여 얻은 y_t의 추정치를 의미한다.[122] 따라서 x_t가 y_t를 그랜저 인과하는지를 검정하기 위해서는 다음 식 (4-9)와 같은 회귀식을 추정한 뒤, $H_0 : \alpha_{1i} = 0$ 라는 귀무가설을 상정하여 해당 귀무가설에 대해 F 검정을 수행하면 된다(귀무가설 기각 시 그랜저 인과관계가 존재).

---

119) 시차의 길이가 길어지면 모형의 적합도는 높아지지만 추정해야 할 모수가 많아져 자유도가 줄어들고 예측력도 나빠지게 된다.

120) SC가 AIC보다 모형의 복잡성(자유도)에 대해 더 강한 패널티를 부과한다. 예를 들어 모형의 파라미터 수를 k, 관측치 수를 n이라 한다면 AIC의 패널티 항은 $2k$가 되고, SC의 패널티 항은 $klog(n)$이 된다. 따라서 AIC는 SC보다 과적합을 일으킬 가능성이 높으며, SC는 AIC보다 과소적합(underfitting)을 일으킬 가능성이 높다.

121) 그랜저 인과관계는 사회과학에서 말하는 인과관계와 다른 개념이다. 그랜저 인과관계가 있다는 것은 x_t의 과거 정보가 y_t를 예측하는 데 도움을 주었다는 것을 의미한다. 즉, 그랜저 인과관계는 간접적인 인과관계를 보여주지만 직접적인 인과관계를 보여주는 것은 아니다.

122) 조건부 기대치는 정보집합에 선형으로 투영(linear projection)한 것과 같다.

$$(4-9)\quad y_t = \sum_{i=1}^{k} \alpha_{1i} x_{t-i} + \sum_{j=1}^{k} \beta_{1j} y_{t-j} + \epsilon_{1t}$$

반대로 y_t가 x_t를 그랜저 인과하는지를 검정하기 위해서는 $E(x_t|y_{t-j}, x_{t-i})$와 $E(x_t|x_{t-i})$가 같은지를 살펴보면 되므로 다음 식 (4-10)와 같은 회귀식을 추정한 뒤, $H_0 : \beta_{2i} = 0$라는 귀무가설을 상정하여 F 검정을 수행하면 된다.

$$(4-10)\quad x_t = \sum_{i=1}^{k} \alpha_{2i} x_{t-i} + \sum_{j=1}^{k} \beta_{2j} y_{t-j} + \epsilon_{2t}$$

이때 시차의 길이는 AIC 또는 SC가 최솟값을 보이는 시차로 결정되며, 해당 시차에서 오차항(ϵ_{1t}, ϵ_{2t})은 백색잡음이어야 한다.

4.3.1.3 VAR(Vector AutoRegression)

VAR은 ARIMA 모형과 연립방정식 모형(Simultaneous Equation Model)의 비판이 제기되면서 Sims(1980)에 의해 처음 제시된 다변량 시계열 모형으로,[123] 여러 개의 내생변수(endogenous variable)를 AR 모형으로 표현한 모형이다. 구체적인 수식 전개 과정은 다음과 같다. 먼저, X_t와 Y_t라는 두 개의 시계열 데이터가 있을 때, VAR은 다음 식 (4-11)과 같은 구조방정식 체계(structural form equation system)를 가정한다.

$$(4-11)\quad \begin{aligned} X_t &= \rho Y_t + \beta_{11} X_{t-1} + \beta_{12} Y_{t-1} + u_{1t} \\ Y_t &= \theta X_t + \beta_{21} X_{t-1} + \beta_{22} Y_{t-1} + u_{2t} \\ where\ \ & u_{1t} \sim w.n(0, \sigma_1^2),\ \ u_{2t} \sim w.n(0, \sigma_2^2),\ \ Cov(u_{1t}, u_{2t}) = \sigma_{12} \end{aligned}$$

123) ARIMA는 하나의 변수와 해당 변수의 과거값을 통해서만 미래를 예측하기 때문에 여러 경제변수들 간의 상호 작용을 고려할 수 없다는 단점이 있으며, 연립방정식 모형은 연구자가 선호하는 경제이론에 따라 모형의 구조를 설정하기 때문에 경제이론에 자유롭지 못하다는 단점이 있다. Sims(1980)는 이를 고려해 경제변수들 간의 상호 관계를 반영할 수 있으면서 경제이론으로부터 자유로운 연립방정식 모형을 제시하였다.

여기서 오차항 u_{1t}와 u_{2t}는 백색잡음으로 자기상관은 없으나 두 오차항 사이에는 공분산 σ_{12}가 존재한다. 그러나 식 (4-11)과 같은 구조방정식 체계는 연립방정식의 해(solution)를 구하기 어려우므로 이를 유도방정식 체계(reduced form equation system)로 수정하는 과정이 필요하다. 이를 위해 식 (4-11)을 다음 식 (4-12)와 같이 행렬 형태로 표현한다.

$$(4\text{-}12)\quad \begin{aligned} X_t &= \rho Y_t + \beta_{11} X_{t-1} + \beta_{12} Y_{t-1} + u_{1t} \\ Y_t &= \theta X_t + \beta_{21} X_{t-1} + \beta_{22} Y_{t-1} + u_{2t} \end{aligned}$$

$$\Downarrow$$

$$\begin{aligned} X_t - \rho Y_t &= \beta_{11} X_{t-1} + \beta_{12} Y_{t-1} + u_{1t} \\ Y_t - \theta X_t &= \beta_{21} X_{t-1} + \beta_{22} Y_{t-1} + u_{2t} \end{aligned}$$

$$\Downarrow$$

$$\begin{bmatrix} 1 & -\rho \\ -\theta & 1 \end{bmatrix} \begin{bmatrix} X_t \\ Y_t \end{bmatrix} = \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix} \begin{bmatrix} X_{t-1} \\ Y_{t-1} \end{bmatrix} + \begin{bmatrix} u_{1t} \\ u_{2t} \end{bmatrix}$$

이때 $\begin{bmatrix} 1 & -\rho \\ -\theta & 1 \end{bmatrix}$를 선두계수행렬(Leading Coefficient Matrix)이라 하며, 이 선두계수행렬의 역행렬을 양변에 곱하면 위 식은 다음과 같이 수정된다.

$$(4\text{-}13)\quad \begin{bmatrix} X_t \\ Y_t \end{bmatrix} = \begin{bmatrix} 1 & -\rho \\ -\theta & 1 \end{bmatrix}^{-1} \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix} \begin{bmatrix} X_{t-1} \\ Y_{t-1} \end{bmatrix} + \begin{bmatrix} 1 & -\rho \\ -\theta & 1 \end{bmatrix}^{-1} \begin{bmatrix} u_{1t} \\ u_{2t} \end{bmatrix}$$

여기서 $\begin{bmatrix} 1 & -\rho \\ -\theta & 1 \end{bmatrix}^{-1} \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix}$는 $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$로, $\begin{bmatrix} 1 & -\rho \\ -\theta & 1 \end{bmatrix}^{-1} \begin{bmatrix} u_{1t} \\ u_{2t} \end{bmatrix}$는 $\begin{bmatrix} e_{1t} \\ e_{2t} \end{bmatrix}$로 표현하면, 다음과 같이 각 방정식에 내생변수가 빠진 유도방정식 체계로 수정된다.

$$(4\text{-}14)\quad \begin{bmatrix} X_t \\ Y_t \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} X_{t-1} \\ Y_{t-1} \end{bmatrix} + \begin{bmatrix} e_{1t} \\ e_{2t} \end{bmatrix}$$

여기서 $\begin{bmatrix} X_t \\ Y_t \end{bmatrix}$는 Z_t, $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$는 A, $\begin{bmatrix} X_{t-1} \\ Y_{t-1} \end{bmatrix}$는 Z_{t-1}, $\begin{bmatrix} e_{1t} \\ e_{2t} \end{bmatrix}$는 E_t로 치환하면 해

당 식은 $Z_t = AZ_{t-1} + E_t$의 형태가 되므로 결국 이 유도방정식은 AR(1) 모형과 같은 형태임을 알 수 있다. 즉, 과거 시차로 구성된 구조방정식 체계를 유도방정식 체계로 수정한 것이 VAR 모형이다.

VAR 모형의 모수를 추정하기 위해서는 유도방정식으로부터 모수를 추정한 후 VAR 모형의 원형인 구조방정식의 모수를 구해야 한다. 그러나 유도방정식으로부터 구조방정식의 모수를 추정하기 위해서는 N(N-1)개의 제약 조건[124]이 필요하게 되는데 이를 VAR 모형의 식별 문제라 한다. Sims(1980)는 이를 해결하기 위해 두 가지 가정을 사용한다. 첫 번째는 구조방정식의 오차항의 공분산, 즉 식 (4-11)의 σ_{12}가 0이라는 가정이고, 두 번째는 선두계수행렬이 하삼각행렬(lower triangular matrix)이라는 가정이다. 따라서 식 (4-11)의 구조방정식은 Sims(1980)의 가정에 의해 다음과 같은 구조를 지닌다.

$$
\begin{aligned}
(4\text{-}15)\quad X_t &= \beta_{11} X_{t-1} + \beta_{12} Y_{t-1} + u_{1t} \\
Y_t &= \theta X_t + \beta_{21} X_{t-1} + \beta_{22} Y_{t-1} + u_{2t} \\
where\ \ u_{1t} &\sim w.n(0, \sigma_1^2),\ \ u_{2t} \sim w.n(0, \sigma_2^2),\ \ Cov(u_{1t}, u_{2t}) = 0
\end{aligned}
$$

위와 같은 구조의 방정식을 축차방정식 체계(recursive equation system)라 한다. 축차방정식은 X_t를 추정한 뒤 추정된 X_t를 두 번째 식에 대입하여 Y_t를 추정한다. 따라서 축차방정식은 내생변수의 순서(order)에 의해 분석 결과가 달라지며, 이로 인해 Sims(1980)의 VAR 모형은 연립방정식 모형처럼 이론으로부터 자유로울 수 없다는 비판을 피할 수 없었다.[125]

한편 VAR 모형은 공분산 행렬을 0으로 바꿔주기 위해 Cholesky 분해를 수행한다. 예를 들어 식 (4-14)에서 $\begin{bmatrix} e_{1t} \\ e_{2t} \end{bmatrix}$는 E_t라하고, 공분산 행렬 $Var(E_t)$

124) 여기서 N은 내생변수의 개수를 말한다. 예를 들어 식 (65)의 구조방정식에서 구해야 하는 모수는 9개지만 식 (68)의 유도방정식에서 구할 수 있는 모수는 7개밖에 없으므로 2개의 모수에 대한 가정이 필요하다.

125) 내생변수의 순서를 경정하는 일은 연구자가 주관적으로 판단할 수밖에 없는데, 순서를 결정하는 과정에서 은연 중 자신이 믿는 이론에 따라 순서를 결정하게 된다. 이에 따라 Sims(1980) 이후에는 Sims(1980)의 두 가지 제약 조건 대신 아예 특정 이론에 근거하여 제약 조건을 부과하는 구조적 벡터자기회귀(Structural AutoRegressiv, 이하 SVAR) 모형이 연구 방법론으로 종종 활용되었다.

는 $E[E_t E_t'] = \begin{bmatrix} \sigma_{e,1}^2 & \sigma_{e,12}^2 \\ \sigma_{e,12}^2 & \sigma_{e,2}^2 \end{bmatrix} = \Omega$라 했을 때, Cholesky 분해는 다음과 같이 Ω를 대각행렬(diagonal matrix) $D = \begin{bmatrix} \sigma_{e,1}^2 & 0 \\ 0 & \sigma_{e,2}^2 \end{bmatrix}$로 바꿔주는 어떤 하삼각행렬 P를 구하는 행위를 의미한다.

$$(4\text{-}16)\quad P\Omega P' = D$$

그리고 이처럼 오차항의 공분산 행렬을 Cholesky 분해한 VAR 모형과 Sims(1980)의 가정을 따른 축차방정식 모형은 결국 같은 모형을 의미한다. 예를 들어 $P = \begin{bmatrix} 1 & 0 \\ \delta & 1 \end{bmatrix}$으로 가정하고, 식 (4-14) 양변에 P를 곱하면 식 (4-14)는 다음과 같이 수정된다.

$$(4\text{-}17)\quad \begin{bmatrix} 1 & 0 \\ \delta & 1 \end{bmatrix} \begin{bmatrix} X_t \\ Y_t \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \delta & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} X_{t-1} \\ Y_{t-1} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ \delta & 1 \end{bmatrix} \begin{bmatrix} e_{1t} \\ e_{2t} \end{bmatrix}$$

여기서 $\begin{bmatrix} 1 & 0 \\ \delta & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$는 $\begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix}$, $\begin{bmatrix} 1 & 0 \\ \delta & 1 \end{bmatrix} \begin{bmatrix} e_{1t} \\ e_{2t} \end{bmatrix}$는 $\begin{bmatrix} u_{1t} \\ u_{2t} \end{bmatrix}$로 표현한다면 위 식은 다음과 같이 수정된다.

$$(4\text{-}18)\quad \begin{bmatrix} 1 & 0 \\ \delta & 1 \end{bmatrix} \begin{bmatrix} X_t \\ Y_t \end{bmatrix} = \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix} \begin{bmatrix} X_{t-1} \\ Y_{t-1} \end{bmatrix} + \begin{bmatrix} u_{1t} \\ u_{2t} \end{bmatrix}$$

$$\Downarrow$$

$$\begin{aligned} X_t &= \beta_{11} X_{t-1} + \beta_{12} Y_{t-1} + u_{1t} \\ Y_t &= -\delta X_t + \beta_{21} X_{t-1} + \beta_{22} Y_{t-1} + u_{2t} \end{aligned}$$

식 (4-18)을 살펴보면 θ만 $-\delta$로 바뀌었을 뿐 앞에 식 (4-15)와 동일한 것을 알 수 있다. 또한 아래 식 (4-19)와 같이 위 모형에서 오차항의 공분산 행렬을 구하면 두 오차항의 공분산이 0이 되는 것을 확인할 수 있다.

$$(4-19)\quad Var(U_t) = E[PE_tE_t'P'] = P\Omega P' = D = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$

4.3.1.4 공적분 검정(Cointegration Test), VEC(Vector Error Correction)

VEC는 시계열 변수들 사이에 공적분 관계가 있을 때 변수들의 장기적인 균형 관계를 고려하기 위해 사용되는 모형이다. 여기서 공적분이란 각각의 시계열은 비정상성이지만 시계열들의 선형결합(linear combination)은 정상성을 얻게 되는 경우를 말한다. 즉, 시계열들의 적분 차수(order of integration)[126]가 d라면 시계열들의 선형결합 적분 차수는 d보다 작은 경우이다.[127] 이 경우 비록 시계열 변수들이 비정상적 시계열이라도 선형결합이 정상성을 가지므로 해당 회귀식의 오차항도 정상적이다. 다시 말해 확률적 추세를 따르던 시계열 변수가 선형조합을 통해 확률적 추세가 제거된 것이며, 이는 곧 시계열 변수들이 장기적인 균형 관계에 있다는 것을 의미한다. 따라서 공적분 검정을 통해 공적분 관계가 존재한다고 나오면 수준(level) 변수가 비정상적 시계열이라도 가성 회귀 문제를 방지할 수 있으며, 시계열 변수 간의 의미 있는 장기적 관계도 식별할 수 있게 된다.

공적분 검정을 수행하는 방식은 크게 EG 검정(Engel-Granger two-step cointegration test)과 요한센 검정(Johansen cointegration test)이 있다. EG 검정은 두 시계열에 대해 회귀분석을 수행하여 잔차를 계산한 뒤, 추정된 잔차항에 대해 ADF 검정을 수행하는 방식이다.[128] 그러나 이 방식은 변수가 세 개 이상일 경우 다양하게 나타날 수 있는 공적분 관계를 고려하지 못한다는 단점이 있다. 따라서 EG 검정은 주로 두 변수 간의 공적분 관계를 검정할 때 사용되며, 다변량 시계열 분석에서는 대부분 요한센 검정이 사용된다. 요한센 검정에서는 VAR 모형을 추정하여 공적분의 개수를 결정한다. 예를 들

126) covariance-stationary 상태가 되기 위한 최소 차분 수를 말한다.

127) 예를 들어 두 시계열 Y_t와 X_t는 $I(1)$이지만, 두 시계열의 선형결합인 Z_t는 $I(0)$이라면 Y_t와 X_t는 공적분 관계에 있다.

128) 예를 들어 $Y_t = \alpha X_t + Z_t$라는 회귀식이 있으면 Z_t가 안정 시계열인지를 검정한다.

어 다음 식 (4-20)과 같은 VAR(2) 모형이 있다고 가정해 보자.

$$(4\text{-}20)\quad X_t = A_1 X_{t-1} + A_2 X_{t-2} + E_t$$

여기서 X_t는 다변량 변수로 n개의 내생변수로 구성되어 있다. 그리고 위 식 양변에 $-X_{t-1}$을 추가하고, 우변에 $-A_2 X_{t-1} + A_2 X_{t-1}$을 추가한다면 해당 식은 다음과 같이 변형될 수 있다.

$$(4\text{-}21)\quad X_t - X_{t-1} = -X_{t-1} - A_2 X_{t-1} + A_2 X_{t-1} + A_1 X_{t-1} + A_2 X_{t-2} + E_t$$

$$X_t - X_{t-1} = (-A_2)(X_{t-1} - X_{t-2}) + (A_1 + A_2 - I) X_{t-1} + E_t$$

$$\Delta X_t = \Pi_1 \Delta X_{t-1} + \Pi X_{t-1} + E_t$$

여기서 추정 계수 행렬 Π가 공적분 관계를 반영하는 행렬로 $\Pi = AB'$의 형태를 가진다. 즉, 공적분 관계는 $\Pi X_{t-1} = AB' X_{t-1}$이 되며, 이때 A를 조정 벡터, B를 공적분 벡터, 그리고 $B' X_{t-1}$를 공적분 방적식이라 한다. 또한 $\Pi = AB'$에서 A와 B는 $n \times r$의 크기를 가지며 이때 n은 내생변수의 수를, r은 Π의 고유치(특성치)의 개수, 즉 공적분의 개수를 나타낸다. 그리고 이 Π의 랭크(rank)[129] $r(r \leq n)$을 공적분 랭크라 부른다. 즉, 요한센 검정은 공적분 랭크가 몇 개인지를 검정하는 것이 되며, 만약 Π의 랭크가 3개라면 공적분도 3개가 존재하는 것이 된다.

일반적으로 내생변수가 $I(1)$ 계열이면 차분을 통해 안정 시계열로 변형한 뒤 VAR 모형을 추정하게 된다. 그러나 차분 변수는 시계열 자료의 장기적인 속성을 잃어버리고 단기적이고 불규칙한 속성만 남는다는 단점이 있다. 다시 말해 공적분 관계가 있는 차분 시계열로 VAR 모형을 추정하게 되면 내생변수들의 장기적인 균형 관계를 포착하지 못하는 문제가 발생한다. 가령 장기적인 균형 관계가 있는 두 경제변수 중 어느 한 변수에 충격(shock)이 들어온 경우를 생각해보자. 이 경우 두 변수의 장기균형에는 오차가 생기게 되고, 두 변수는 이 오차를 수정하는 방향으로 변화되어 균형 값을 찾아가게 될 것이

129) 행렬에서 선형결합이 독립적인 열(column)의 개수를 rank(위수, 계수)라 부른다.

다. 게다가 경제변수는 이 오차가 바로 조정되는 것이 아니라 서서히 조정되는 것이 일반적이다.[130] 그런데 두 변수의 차분 시계열로 VAR 모형을 추정하게 되면은 이러한 오차를 수정하고자 하는 힘을 간과하게 된다. 즉, 오차에 의해 아직 균형에 도달하지도 않았는데 이미 균형에 도달했다고 문제를 풀어버리는 오류를 범하게 된다.

이러한 문제를 해결하기 위해 VEC 모형은 VAR 모형에 오차수정모형(Error Collection Model)을 도입하였다. 즉, 장기적 균형으로부터 이탈한 단기적 움직임을 다시 장기 수준으로 회귀할 수 있도록 VAR에 오차수정항을 삽입한 모형이 VEC이다. VEC의 구체적인 수식은 다음 식 (4-22)와 같이 요한센 검정 모형으로부터 도출할 수 있다.

$$(4\text{-}22)\quad \Delta X_t = \Pi_1 \Delta X_{t-1} + \Pi X_{t-1} + E_t$$

$$\Delta X_t = \Pi_1 \Delta X_{t-1} + AB' X_{t-1} + E_t$$

$$\Delta X_t = \Pi_1 \Delta X_{t-1} + AZ_{t-1} + E_t$$

여기서 A를 오차조정계수, Z_{t-1}을 오차수정항이라 부르며, 오차조정계수는 내생변수들의 장기균형이 깨졌을 때 장기균형으로 회복되는 속도, 즉 오차조정속도를 의미한다. 따라서 오차조정계수가 음(-)의 값을 가져야 장기균형에 도달하게 되며, 그 절대값이 클수록 장기균형으로 회복되는 속도가 빠른 것을 의미하게 된다.

4.3.1.5 충격반응분석(Impulse Response Analysis)

VAR 모형은 연립방정식 모형이면서 동시에 AR 모형이기 때문에 각 선결 변수(내생변수의 시차 변수)의 추정계수를 해석하기가 쉽지 않다.[131] 이로

130) 예를 들어 가격이 상승하면 공급량은 곧바로 증가하는 것이 아니라, 생산시설을 늘리는 데 걸리는 시간, 가격 상승이 일시적일 수 있다는 우려 등 다양한 요인에 의해 서서히 증가한다.

131) VAR 모형은 변수들이 서로 상호 영향을 주고받는다. 따라서 추정계수의 부호나 크기만 보고서는 어느 한 내생변수의 변화가 발생했을 때 이 변화가 시간에 따라 해당 내생변수와 다른 내생변수에 어떻게 영향을 미치는지를 이해할 수가 없다.

인해 VAR 모형은 변수들 사이의 관계를 파악하기 위해 충격반응분석을 수행한다. 충격반응분석은 VAR 모형의 추정 결과를 기반으로 어느 한 내생변수에 변화가 발생했을 때, 각 내생변수가 시간이 지남에 따라 어떻게 변화해 가는지를 살펴보는 방법이다. 이때 내생변수의 변화란 오차항에서의 변화를 의미한다.[132] 즉, 오차항에 1단위 충격(보통 1표준편차로 설정)이 가해졌을 때 시차 구조(memory structure)를 따라 자기 자신과 VAR 체계 내 다른 내생변수들이 시차를 두고 어떻게 변화하는지를 분석한다.

충격반응분석을 수행하는 방식은 두 가지가 있다. 첫 번째는 VAR(P)를 VMA(∞)로 전환하여 시차 구조에 의한 외부충격에 대한 반응을 분석하는 것이다. 예를 들어 앞에 식 (4-14)를 치환한 $Z_t = AZ_{t-1} + E_t$ 형태의 VAR(1) 모형이 있다고 해보자. 이때 AZ_{t-1}은 시차 연산자(lag operator) L을 이용하여 ALZ_t의 형태로 나타낼 수 있으므로 해당 식은 다음 식 (4-23)과 같은 과정을 거쳐 VMA(∞) 형태로 나타낼 수 있다.

(4-23)
$$
\begin{aligned}
Z_t &= AZ_{t-1} + E_t \\
(I - AL)Z_t &= E_t \\
Z_t &= (I - AL)^{-1}E_t \\
Z_t &= (I + \Phi_1 L + \Phi_2 L^2 + \cdots)E_t \\
Z_t &= E_t + \Phi_1 E_{t-1} + \Phi_2 E_{t-2} + \cdots
\end{aligned}
$$

$$\Downarrow$$

$$
\begin{bmatrix} X_t \\ Y_t \end{bmatrix} = \begin{bmatrix} e_{1,t} \\ e_{2,t} \end{bmatrix} + \begin{bmatrix} \varphi_{111} & \varphi_{112} \\ \varphi_{121} & \varphi_{122} \end{bmatrix} \begin{bmatrix} e_{1,t-1} \\ e_{2,t-1} \end{bmatrix} + \begin{bmatrix} \varphi_{211} & \varphi_{212} \\ \varphi_{221} & \varphi_{222} \end{bmatrix} \begin{bmatrix} e_{1,t-2} \\ e_{2,t-2} \end{bmatrix}
$$

$$\Downarrow$$

$$
\begin{aligned}
X_t &= e_{1,t} + \varphi_{111}e_{1,t-1} + \varphi_{112}e_{2,t-1} + \varphi_{211}e_{1,t-2} + \varphi_{212}e_{2,t-2} + \cdots \\
Y_t &= e_{2,t} + \varphi_{121}e_{1,t-1} + \varphi_{122}e_{2,t-1} + \varphi_{221}e_{1,t-2} + \varphi_{222}e_{2,t-2} + \cdots
\end{aligned}
$$

이러한 함수 형태를 충격반응함수(Impulse Response Function, IRF)라 하며, 여기서 시차별 오차항의 계수 φ가 변수들의 시간에 따른 반응을 나타낸

132) 시차 변수를 사용하기 때문에 각각의 설명변수에 변화가 있다고 가정하기가 어려워 오차항에 변화가 있다고 가정한다.

다. 그러나 이러한 방식은 오차항 간의 공분산 관계를 고려하지 못하기 때문에, VAR 모형처럼 오차항 간의 공분산이 있는 경우에는 오차항에서의 변화가 어떻게 각 내생변수에 영향을 미치는지를 직관적으로 이해하기가 어렵다. 이를 위해 Cholesky 분해를 이용한 두 번째 방법은 오차항의 공분산 행렬을 대각행렬로 전환하여 충격반응분석을 수행한다. 즉, 앞에 식 (4-18)과 (4-19)에서 살펴본 것처럼 공분산 행렬 $\Omega = \begin{bmatrix} \sigma_{e,1}^2 & \sigma_{e,12}^2 \\ \sigma_{e,12}^2 & \sigma_{e,2}^2 \end{bmatrix}$를 대각행렬 $D = \begin{bmatrix} \sigma_{e,1}^2 & 0 \\ 0 & \sigma_{e,2}^2 \end{bmatrix}$로 바꿔주는 하삼각행렬 P를 구한 뒤, 이를 VAR 모형에 곱한 축차방정식 체계를 이용해 충격반응분석을 수행한다. 이 경우 오차항 간의 공분산이 없으므로 어느 한 내생변수에서 오차항에 변화가 있으면 직관적으로 각 내생변수가 어떻게 변화하는지를 알 수 있다. 또한 Cholesky 분해를 이용한 충격반응분석은 변수의 순서에 따라 충격반응이 달라진다는 특징이 있다. 예를 들어 앞에 식 (4-18)의 X_t에서 u_{1t}에 충격이 들어온다면 곧바로 X_t가 변동하고 Y_t도 이 충격량의 θ비율만큼 변동이 발생한다. 그리고 X_t의 변동은 1시차 뒤에 $\beta_{11}X_{t-1}$과 $\beta_{21}X_{t-1}$의 변동을 발생시키고, Y_t의 변동은 1시차 뒤에 $\beta_{12}Y_{t-1}$과 $\beta_{22}Y_{t-1}$의 변동을 발생시킨다. 반면, Y_t에서 u_{2t}에 충격이 들어온다면 Y_t는 곧바로 충격량만큼 변동하지만, X_t는 1시차 뒤에야 β_{12}만큼 변동하게 된다. 구체적으로는 Y_t의 변동이 1시차 뒤에 $\beta_{12}Y_{t-1}$과 $\beta_{22}Y_{t-1}$를 변동시키고, $\beta_{12}Y_{t-1}$의 변동은 X_t를 변동시켜 이로 인해 $\beta_{11}X_{t-1}$과 θX_t가 변동하게 된다. 따라서 앞서 언급하였듯이 VAR 모형은 변수들의 배열순서가 매우 중요한 문제로 다루어지며, 일반적으로는 그랜저 인과관계 또는 경제이론에 근거하여 배열순서를 결정하게 된다.[133]

한편 충격반응분석의 결과는 소멸형 반응(decayed response), 지속형 반응(persistent response), 지연형 반응(delayed response), 주기형 반응(cyclical response), 과잉 반응(overshooting) 등으로 식별해낼 수 있다. 소멸형 반응은 충격이 발생한 후 반응의 크기가 점차 줄어들어 원래 상태로 돌아가는 형태

133) Pesaran and Shin(1998)은 이러한 점을 고려해 변수 순서에 영향을 받지 않는 일반화 충격반응분석(generalized impulse response analysis)을 제시하였다. 이 경우 예를 들어 j번째 변수의 충격반응을 구한다면 j번째 변수를 가장 처음에 놓고 충격반응분석을 수행하게 된다.

의 반응이고, 지속형 반응은 충격의 영향이 시간이 지나도 계속 유지되는 형태의 반응이다. 또한 지연형 반응은 충격이 발생한 직후에는 반응이 미미하거나 거의 없지만 시간이 지나면서 점차 효과가 나타나거나 반응이 늦게 시작되는 형태의 반응이며, 주기형 반응은 충격의 반응이 주기적으로 나타나는 경우이고, 과잉 반응은 충격에 대한 초기 반응이 매우 크게 반응한 후 조정되는 경우이다. 이 외에도 VEC 모형의 경우에는 일부 오차수정항의 추정계수가 양수(+)로 나타날 시 충격에 따른 반응이 수렴하는 모습을 보이지 않고, 발산하는 모습을 보일 수 있다.

4.3.1.6 분산분해분석(Variance Decomposition Analysis)

어느 한 내생변수에서 예측오차가 발생했다면 이 예측오차의 분산은 해당 내생변수와 다른 내생변들의 오차항의 분산으로 분해될 수 있다. 따라서 어떤 내생변수에서 예측하지 못한 변동이 발생했다면, 이는 해당 내생변수에서 발생한 예측하지 못한 변동일 수도 있고, 다른 내생변수에서 발생한 예측하지 못한 변동일 수도 있다. 분산분해분석[134]은 이러한 예측하지 못한 변동이 다른 변수들의 충격에 의해 얼마나 설명되는지를 수치적으로 분해하는 방법이다. 즉, 어느 한 내생변수의 변동 중에서 각각의 내생변수가 전체 변동에 기여한 부분의 상대적 크기를 보여주는 것이 분산분해분석이다.

분산분해분석은 충격반응분석과 마찬가지로 두 가지 방법으로 수행할 수 있다. 첫 번째 방법은 VAR(P)를 VMA(∞)로 전환하는 방법이고, 두 번째 방법은 Cholesky 분해를 이용하는 것이다. 그러나 첫 번째 방법은 앞서 살펴보았듯이 오차항 간의 공분산을 고려하지 못하기 때문에 일반적으로 VAR 모형에서는 두 번째 방법을 이용해 분산분해분석을 수행하게 된다. 즉, Cholesky 분해로 공분산 행렬을 대각행렬로 바꾼 뒤 예측오차의 분산을 분해한다. 이를 좀 더 구체적으로 살펴보기 위해 Cholesky 분해를 수행한 축차방정식이 다음 식 (4-24)와 같다고 해보자.

---

134) 예측오차분산분해(forecast error variance decomposition)라고도 불린다.

$$(4-24)\quad X_t = \beta_{11} X_{t-1} + \beta_{12} Y_{t-1} + u_{1,t}$$
$$Y_t = \theta X_t + \beta_{21} X_{t-1} + \beta_{22} Y_{t-1} + u_{2,t}$$

해당 식을 이용해 예측을 수행한다면 다음과 같은 식이 도출된다.

$$(4-25)\quad X_{t+1} = \beta_{11} X_t + \beta_{12} Y_t + u_{1,t+1} \Rightarrow E(X_{t+1}|\Omega_t) = \beta_{11} X_t + \beta_{12} Y_t$$
$$Y_{t+1} = \theta X_{t+1} + \beta_{21} X_t + \beta_{22} Y_t + u_{2,t+1} \Rightarrow E(Y_{t+1}|\Omega_t) = \theta E[X_{t+1}|\Omega_t] + \beta_{21} X_t + \beta_{22} Y_t$$

위 식을 이용해 예측오차 식을 도출하면 다음과 같다.

$$(4-26)\quad X_{t+1} - E(X_{t+1}|\Omega_t) = u_{1,t+1}$$
$$Y_{t+1} - E(Y_{t+1}|\Omega_t) = \theta[X_{t+1} - E(X_{t+1}|\Omega_t)] + u_{2,t+1}$$

마지막으로 위 식에서 예측오차의 분산을 구하면 다음과 같은 식이 도출된다.

$$(4-27)\quad Var[X_{t+1} - E(X_{t+1}|\Omega_t)] = Var[u_{1,t+1}] = \sigma_1^2$$
$$Var[Y_{t+1} - E(Y_{t+1}|\Omega_t)] = Var\{\theta[X_{t+1} - E(X_{t+1}|\Omega_t)]\} + Var[u_{2,t+1}] + Cov[\bullet]$$
$$= \theta^2\sigma_1^2 + \sigma_2^2$$

식 (4-27)을 살펴보면 첫 번째 변수는 $t+1$ 시점의 예측오차 분산이 자기 자신의 오차항의 분산으로만 결정되는 반면, 두 번째 변수는 자기 자신과 첫 번째 변수의 오차항의 분산으로 결정되는 것을 알 수 있다. 즉, 분산분해분석 역시 축차방정식 체계를 활용하기 때문에 변수들의 배열순서를 어떻게 결정하느냐에 따라 그 결과가 다르게 나타난다.[135]

---

135) 참고로 첫 번째 변수의 $t+2$ 시점 분산분해분석 결과는 다음과 같다.

$$X_{t+2} = \beta_{11} X_{t+1} + \beta_{12} Y_{t+1} + u_{1,t+2}$$
$$\Downarrow$$
$$E(X_{t+2}|\Omega_t) = \beta_{11} E(X_{t+1}|\Omega_t) + \beta_{12} E(Y_{t+1}|\Omega_t)$$
$$Var[X_{t+2} - E(X_{t+2}|\Omega_t)] = \beta_{11}^2 Var[X_{t+1} - E(X_{t+1}|\Omega_t)] + \beta_{12}^2 Var[Y_{t+1} - E(Y_{t+1}|\Omega_t)] + Var[u_{1,t+2}] + Cov[\bullet]$$
$$= \beta_{11}^2\sigma_1^2 + \beta_{12}^2[\theta^2\sigma_1^2 + \sigma_2^2] + \sigma_1^2$$

4.3.2 불확실성에 대한 이론적 고찰과 국내 실증연구

시장경제에서 불확실성은 불가피하게 발생하는 현상이지만 불확실성의 과도한 확대는 실물 경제에 불필요한 부작용을 발생시킨다. 특히, 미래에 대한 예측으로 합리적 의사결정을 내리는 경제주체들은 불확실성으로 인해 위험을 회피하고 관망하려는 유인이 제공되어 가계는 소비를, 기업은 투자를 크게 위축시킨다. 본 항에서는 이러한 불확실성이 소비와 투자를 위축시킨다는 이론적 근거를 검토해보고, 우리나라의 불확실성을 측정한 기존 선행연구들의 실증분석 결과를 살펴본다.

4.3.2.1 이론적 고찰

불확실성이 소비를 위축시킨다는 이론적 근거는 Leland(1968)에 의해 구체화 되기 시작한 예비적 저축(precautionary saving) 가설에서 찾을 수 있다. 동 가설에 따르면 한계효용 함수가 볼록(convex)한 소비자[136]는 미래 소득에 대한 불확실성이 증가할수록 미래 소비의 한계효용 기대치가 증가하여 현재 소비를 감소시키고 저축을 늘려 미래를 대비한다. 이때 증가한 저축은 불확실성에 대비하기 위해 축적된 예비적 자산으로, 소득 감소 충격에 대한 완충재고(buffer stock) 역할을 한다. 따라서 예비적 저축 가설에 따르면 현시점에 인식한 미래의 불확실성은 소비자의 부(wealth)와 양(+)의 관계를 가지며, 현재 소비와는 음(-)의 관계를, 미래 소비와는 양(+)의 관계를 갖는다.

이와 관련하여 Dardanoni(1991), Hahm and Steigerwald(1999), Carroll and Samwick(1998), Wilson(1998)은 예비적 저축 가설을 지지하는 실증분석 결과를 제시하였다. Dardanoni(1991)는 영국 가계 지출 조사(UK Family Expenditure Survey)에서 수집된 횡단면 데이터를 사용하여 영국 가계 저축의 60%가 예비적 동기에 근거한다는 사실을 제시하였으며, Carroll and Samwick(1998)은 PSID(Panel Study of Income Dynamics) 자료를 사용하여 가계 부의 32%~50%가 예비적 저축 행태를 보인다는 사실을 실증적으로 제

136) 위험 회피적인 소비자의 효용함수는 3차 미분이 0보다 큰 효용함수로 대변된다.

시하였다. 또한 Wilson(1998)은 소비의 오일러(Euler) 방정식을 추정한 결과, 미국 부의 절반 정도가 예비적 동기에 의해 설명된다고 보고하였으며, Hahm and Steigerwald(1999) 역시 오일러 방정식을 통해 불확실성이 다음 시점의 소비 증가율을 높인다는 분석 결과를 제시하였다.

국내에서도 예비적 저축의 존재 여부에 대한 다수의 실증연구가 진행되었으며, 대부분이 예비적 저축의 존재를 지지하는 실증분석 결과를 제시하고 있다. 이명훈·최창규(2000)는 ARCH 모형에서 구한 조건부 이분산(conditional heteroscedasticity)으로 불변국민소득의 불확실성을 추정한 뒤, 소비의 오일러 방정식으로부터 불확실성과 소비 증가율의 관계를 실증적으로 분석하였다. 분석 결과에서는 불확실성 확대가 현재 소비를 감소시키는 것으로 나타났다. 반면, 이우헌(2001)은 대우 패널조사 자료와 소비의 분산, 그리고 오일러 방정식의 예측오차 분산을 사용하여 가구별 소비행태를 외환위기 전후로 분석하였으나, 예비적 저축의 실증적 증거를 찾지 못하였다. 신관호·주원(2002)은 Carroll and Samwick(1997)의 방법론을 따라 소득에 영향을 미치는 충격을 항구적 충격과 일시적 충격으로 분해하여 각각의 분산을 불확실성의 대리변수로 사용하였다. 분석 결과, 부의 변화는 예비적 저축과 반대되는 결과가 나타났지만, 소비에 있어서는 예비적 저축을 지지하는 방향으로 결과가 나타났다. 장민·황인도(2004)는 심리지수의 예측오차로 측정한 소득 불확실성과 ARCH 모형으로 추정한 소득과 차입규모 및 자산가격의 불확실성이 소비에 미치는 영향을 분석하였다. 그 결과 불확실성이 소비에 부정적인 영향을 미치는 것으로 나타났다. 차은영·최은영(2007)은 한국 노동패널조사 자료를 이용하여 소득 불확실성과 부의 관계를 분석한 결과, 불확실성과 부는 양(+)의 관계를 보였지만, 불확실성의 추정계수 추정치가 비탄력적이어서 불확실성에 대한 예비적 저축의 반응이 그리 크지 않음을 발견하였다. 가장 최근에는 황진태·김성민(2020)이 오일러 방정식을 활용하여 예비적 저축 동기 여부를 살펴보았다. 불확실성 변수는 차은영·최은영(2007)과 마찬가지로 상대적 신중도(coefficient of relative prudence)를 측정하여 활용하였으며, 분석 결과에서는 2SLS 집단 간 추정량(BE2SLS)의 경우 상대적 신중도가 작게 나타나 예비적 저축 행태가 발견되지 않았지만, 2SLS 확률효과 추정량(RE2SLS)은 예비적

저축 행태가 존재하는 것으로 나타났다.

불확실성과 투자의 관계는 과거 신고전파 투자이론인 조르겐슨(Jorgenson) 모형부터 본격적으로 다루어져 왔다. 조르겐슨 모형에 따르면 기업은 투자에 대한 조정비용(adjustment cost)이 없기 때문에 기업의 문제는 정학적 문제이고, 자본 스톡의 최적 수준은 자본의 한계생산과 자본의 사용자 비용(user cost of capital)[137]이 같아질 때 결정된다. 그러나 동 모형은 투자의 의사결정을 지연할 수 있다는 가능성, 즉 투자 시점을 선택할 수 있다는 가능성을 무시하고 있다. 게다가 현실 경제에서 투자에 대한 의사결정은 이자율이나 조세 정책 같은 자본의 사용자 비용보다는 불확실성에 더 큰 영향을 받을 수 있다. 이를 고려하여 Hartman(1972)과 Avel(1983)은 조르겐슨 모형을 기반으로 불확실성과 조정비용이 수반된 동학적 투자모형을 제시하였다. 두 연구에 따르면 기업은 규모에 대한 수익 불변의 생산함수를 가지며, 자본의 한계수입생산(marginal revenue product)은 생산물 가격에 대해 볼록한 형태를 지닌다.[138] 특히 Hartman(1972)에서는 기업이 임금과 생산물 가격이 결정되기 전에 자본의 양을 먼저 정하고 불확실성이 해소된 이후에 노동의 양을 결정한다고 가정하고 있다. 그리고 이처럼 노동이 자본보다 더 신축적으로 변화할 수 있다면 기업은 노동-자본 비율을 변화시키기 위해 노동의 양을 조절하게 되고, 자본의 한계수입생산은 생산물 가격의 움직임보다 더 크게 변화하게 된다. 따라서 이 경우 자본의 한계수입생산은 생산물 가격에 볼록한 형태를 지니며, 이로 인해 미래 가격에 대한 불확실성의 증가는 자본의 기대수익을 높이고 투자의 증가를 가져오게 된다.

조르겐슨 모형뿐만 아니라 순현재가치법(net present value method)이나 토빈(Tobin)의 q 모형과 같은 전통적인 투자이론들은 기업이 투자 결정을 지연시킬 수 있다는 가능성을 무시하고 있다. 즉, 기업의 투자 결정이 지금 이루어지거나 이루어지지 않는다는 양자 선택을 가정한다. 그러나 기업의 투자 결정은 항상 불확실성이 존재하기 때문에 기업의 의사결정자는 더 많은 정보

137) 이때 Hall and Jorgenson(1967)이 제시한 자본의 사용자 비용은 이자율, 감가상각, 세제 혜택 및 자본재 가격 변화 등에 영향을 받는다.

138) 참고로 Hartman(1972)은 현재 기에 불확실성이 있다고 가정하고 있지만, Avel(1983)은 현재 기에 완전 확실성을 가정하고 있다는 것에 차이가 있다.

를 기다리기 위해 투자 결정을 연기할 수도 있고, 시장환경 변화에 적절히 대응하여 투자 결정을 포기·축소·확대할 수도 있다. 이러한 가능성을 고려하기 위해 현대의 투자이론은 주로 실물옵션(real option) 이론을 중심으로 발전되어 왔다(Bernanke, 1983; McDonald and Siegal, 1986; Pindyck, 1991; Dixit and Pindyck, 1994 등). 실물옵션 이론은 투자의 비가역적(irreversible) 특성[139]에 주목하여 기업의 투자 기회를 콜옵션(call option)[140]의 관점에서 설명한다. 예를 들어 어떤 기업이 빈 부지에 새로운 공장을 건설할 계획이 있다면 이는 시장 상황에 따라 특정 시점에 공장 건설을 수행할 수 있는 권리(옵션)를 보유한 것과 같다. 그리고 이때 공장 건설로 발생하는 투자 비용은 옵션의 행사가격과 같은 의미를 가지며, 옵션의 행사(투자)는 비가역적이다. 즉, 기업은 투자 시점을 통제할 수 있는 비가역적인 선택권을 보유하고 있는 것이다. 따라서 기업의 의사결정자는 투자의 비가역성을 고려해 투자 결정에 있어서 보다 신중한(위험 회피적인) 모습을 보이게 되며, 불확실성이 커질수록 투자를 기다리는 옵션의 가치가 상승하므로 새로운 정보를 얻을 때까지 투자를 지연시키고 관망하는 자세를 취하게 된다. 이러한 이유로 실물옵션 이론에서는 불확실성의 확대가 기업의 투자, 특히 비가역적이고 고정비용이 큰 실물 투자를 감소시키는 요인으로 작용한다.

투자의 비가역성뿐만 아니라 기업의 금융제약(financial constraint) 역시 불확실성과 투자의 관계를 연결해 주는 중요한 요소이다. 불확실성이 확대되면 정보의 비대칭성(information asymmetry)이 증가하여 투자자들은 추가적인 리스크 프리미엄(risk premium)을 요구하게 되고, 이로 인해 가계나 기업과 같은 자금 수요자의 자금중개비용이 상승하게 된다. 게다가 불확실성이 높은 시기에는 대출태도를 보수적으로 전환하는 금융기관들이 늘어나면서 가계나 기업의 금융중개조건이 악화된다. 특히 대출 상환 능력이 취약한 저소득층

139) 투자의 비가역성은 투자 비용의 상당 부분이 매몰 비용(sunk cost)의 성격을 갖기 때문에 일단 투자가 이루어지고 나면 이를 다시 되돌리기 어려운 성질을 말한다. 즉, 비대칭적(asymmetric) 조정비용으로 인해 투자의 감축 비용이 투자의 증가 비용보다 더 크다는 것이다. 특히 자본재는 그 특성상 개별 기업이 각기 다른 용도로 구매하기 때문에 자본재를 다시 시장에 매각하는 경우 자본재의 용도가 제한되어 구매 가격보다 훨씬 낮은 가격에 매각해야 한다.

140) 콜옵션은 투자자에게 특정 시점에 미리 정해진 가격으로 자산을 구매할 권리를 주는 것으로, 일단 자산을 구매하면 옵션은 비가역적이 된다.

과 중소기업에 대한 심사기준이 엄격해지면서 신용등급은 하락하게 되고, 심각한 경우에는 신용경색을 초래하여 가계의 파산과 기업의 부도비율이 상승하게 된다. 이와 관련해 Fazzari et al.(1988)은 금융제약으로 인해 내부자금과 외부자금의 조달 비용 차이가 발생한다고 하였으며, Hatzius et al.(2010)은 정보 비대칭성의 증가가 기업의 자금조달에 부정적 영향을 미치고 이것이 실물시장에 전이(spillover)된다고 하였다. 또한 Gilchrist et al.(2014)은 미래 자산 가격의 불확실성으로 인한 신용스프레드의 확대가 기업의 자본 조달 비용을 높이고 이로 인해 기업의 투자가 감소한다는 실증연구 결과를 제시하였으며, Hakkio et al.(2009)은 대출 심사 서베이 자료를 이용하여 금융스트레스지수가 상승하는 시기에 금융기관들의 대출태도가 보수적으로 바뀐다는 것을 실증적으로 보여주었다.

이처럼 불확실성과 투자의 이론적 관계는 모형에 따라 다른 결론을 도출하고 있지만, 국내의 실증연구는 대부분 불확실성이 투자에 부정적인 영향을 미친다는 연구 결과를 제시하고 있다. 이항용(2005)은 주가수익률의 표준편차를 이용하여 불확실성이 투자에 미치는 영향을 외환위기 전후로 분석한 결과, 외환위기 이후의 기간에만 불확실성이 투자에 유의한 영향을 미치는 것으로 나타났으며, 동 기간 중에서도 재무 건전성이 낮은 기업에서 불확실성의 부정적 영향이 크게 추정되었다. 김범식·권순우(2009)는 주가지수의 변동성을 불확실성의 대리변수로 사용하여 불확실성이 설비투자에 미치는 영향을 추정한 결과, 불확실성과 설비투자는 음(-)의 상관관계를 가지며, 그 관계의 강도는 외환위기 이후 확대되었음을 보고하였다. 가장 최근에는 김천구·박정수(2020)가 주가수익률의 표준편차를 불확실성의 대리변수로 사용하여 불확실성과 비가역성에 대한 실증분석 결과를 제시하였다. 분석 결과에서는 자산 비가역성이 불확실성의 부정적 효과를 증폭시키는 것으로 나타났으며, 기업 규모에 따른 불확실성의 부정적 효과는 대기업이 중·소기업보다 더 크게 나타났다.

이상의 불확실성과 투자의 관계에 집중한 실증적 연구들은 투자와 관련된 특정 경제변수로부터 불확실성을 측정하여 분석을 수행하고 있으나, 최근에는 거시경제 전체에 대한 불확실성이 투자에 미치는 영향을 분석한 연구들이 주를 이루고 있다. 조성빈(2017)은 EPU 지수와 기업 투자의 관계에 대한 실증

분석을 수행하였다. 분석 결과, 금융위기 이전에는 불확실성과 투자가 유의미한 음(-)의 관계를 보였으나, 금융위기 이후에는 통계적으로 유의하지 않은 음(-)의 관계를 나타냈다. 또한 기업 규모에 따른 불확실성 영향을 분석한 결과에서는 김천구·박정수(2020)와는 달리 소규모 기업이 대규모 기업보다 더 뚜렷한 음(-)의 관계를 나타냈다. 김윤경(2020) 역시 EPU 지수와 국내 상장기업을 대상으로 불확실성이 투자에 미치는 영향을 분석하였다. 그 결과 불확실성은 총투자, 설비투자, R&D 투자에 대해 모두 유의한 음(-)의 관계를 나타냈으며, 특히 비가역적 특성이 강한 설비투자에서 좀 더 뚜렷한 음(-)의 관계가 나타남을 보고하였다. 오세환·노성호(2022)는 한국, 중국, 미국의 EPU 지수를 활용하여 불확실성이 국내 기업 투자에 미치는 영향을 살펴보았다. 분석 결과에서 추정계수는 모두 유의하지 않았지만, 한국과 중국의 EPU 지수는 국내 기업 투자와 음(-)의 관계를, 미국의 EPU 지수는 국내 기업 투자와 양(+)의 관계를 나타냈다.

4.3.2.2 우리나라의 경제 불확실성을 측정한 최근 실증연구

앞서 살펴본 불확실성 관련 이론 및 실증적 연구들은 주로 소비 및 투자의 관계에 집중하여 특정 경제변수의 불확실성을 측정하고 그 관계를 규명하는 데 집중하고 있다. 그러나 최근에는 경제 전체에 대한 불확실성을 측정하고 그 파급효과를 분석하는 시도가 늘어나고 있다. 그중에서도 우리나라의 경제 불확실성을 측정한 연구는 불확실성 측정 방법을 기준으로 ① Haddow et al.(2013)의 방법론을 원용한 연구 ② Jurado et al.(2015)의 방법론을 원용한 연구 ③ Baker et al.(2016)의 방법론을 원용한 연구로 나눌 수 있다.

먼저 Haddow et al.(2013)의 방법론을 원용하여 우리나라의 불확실성을 측정한 연구는 이현창·정원석(2016)이 있다. 이현창·정원석(2016)은 불확실성 관련 지표를 전망·금융시장·대외 부문으로 나누어 선정하고, 이들 지표의 공통요인을 추출하는 방식으로 불확실성 지수를 구축하였다. 동 지수를 이용한 실증분석은 불확실성, GDP갭률, 소비자 물가상승률, 콜금리로 구성된 4변수 VAR 모형을 설정해 수행하였으며, 그 결과 GDP는 5분기까지 감소하다 증

가하는 모습을, 물가는 10분기까지 감소하는 모습을 나타냈다. 단, GDP는 0~3분기만 통계적으로 유의했으며, 물가는 2~5분기만 통계적으로 유의했다.

이와 관련하여 정원석 외(2016)는 이현창·정원석(2016)이 측정한 불확실성 지수를 이용하여 불확실성이 소비 및 투자에 미치는 파급경로별 영향을 분석하였다. 파급경로는 Bloom(2014)을 따라 심리 경로, 비용 경로, 예비적 저축 경로, 실물옵션 경로로 구분하였으며, 이때 심리 경로는 경제주체들의 관망 태도 강화로 인한 소비 및 투자 위축 경로를, 비용 경로는 리스크 프리미엄 확대와 금융기관 대출태도 보수화로 인한 소비 및 투자의 위축 경로를, 예비적 저축 경로는 예비적 저축 가설에 의한 소비의 위축 경로를, 실물옵션 경로는 비용지출의 비가역성으로 인한 내구재 소비와 설비투자의 위축 경로를 의미한다. 실증분석은 심리 경로의 경우 불확실성, 경제성장률, 콜금리, 심리지표(소비자 심리지수, 제조업 업황 BSI)의 4변수 VAR 모형을 설계하였으며, 비용 경로의 경우 불확실성, 경제성장률, 신용스프레드(회사채-콜금리)의 3변수 VAR 모형과 불확실성, 경제성장률, 콜금리, 대기업 및 가계(일반) 대출태도의 4변수 VAR 모형을 설계하였다. 충격반응분석 수행 결과에서는 CSI가 1~3분기 감소 후 증가하는 모습을, BSI는 1~5분기 감소 후 증가하는 모습을 나타냈으며, 신용스프레드는 증가하고, 대출태도는 1~2분기 감소 후 증가하는 모습을 나타냈다. 단, CSI는 1~2분기만 통계적으로 유의했으며, BSI와 신용스프레드는 1~3분기만, 대출태도는 1분기만 통계적으로 유의했다.

김광민(2021)은 앞에 두 연구와 유사하게 Haddow et al.(2013)의 방법론을 원용하여 우리나라의 불확실성을 측정하고, 불확실성이 소비 및 투자에 미치는 파급경로별 영향을 분석하였다. 파급경로는 정원석 외(2016)와 마찬가지로 Bloom(2014)을 따라 심리 경로, 비용 경로, 예비적 저축경로, 실물옵션 경로로 구분하였다. 실증분석은 심리 경로의 경우 불확실성, 경제성장률, 실질이자율(콜금리-물가상승률), 심리지표(소비자 심리지수, 제조업 업황 BSI)로 구성된 4변수 VAR 모형을, 비용 경로의 경우 불확실성, 경제성장률, 신용스프레드(회사채-국고채)의 3변수 VAR 모형과 불확실성, 경제성장률, 실질이자율, 대기업·중소기업·가계(일반) 대출태도의 4변수 VAR 모형을 설계하였다. 충격반응분석 결과에서는 CSI가 0~3분기 감소 후 증가, BSI는 0~5분기 감소

후 증가, 신용스프레드는 0~6분기 증가 후 감소, 대출태도는 0~2분기 감소 후 증가하는 모습을 나타냈다. 이때 CSI는 0~2분기와 5~7분기, BSI는 0~3분기, 스프레드는 0~2분기가 통계적으로 유의했으며, 대출태도는 전체 기간이 유의하지 않았다.

다음으로 Jurado et al.(2015)의 방법론을 원용하여 우리나라의 불확실성을 측정한 연구는 이항용·이진(2017)이 있다. 동 연구는 주성분 분석으로 경제변수들의 공통요인을 추정한 뒤, 추정된 요인들이 포함된 요인 확장형 벡터자기회귀 모형(Factor Augment VAR, FAVAR)의 예측오차 분산을 이용하여 우리나라의 불확실성을 측정하였다. 실증분석은 불확실성 지수, 산업생산 증가율, 소비자 물가상승률로 설계된 3변수 VAR 모형으로 충격반응분석을 수행하였으며, 분석 결과에서는 생산이 3개월까지 감소하다 증가하는 모습을, 물가는 4개월까지 감소하다 증가하는 모습을 나타냈다. 이때 생산은 1~2개월이 통계적으로 유의했으며, 물가는 1개월만 통계적으로 유의했다.

Baker et al.(2016)의 방법론을 원용하여 우리나라의 불확실성을 측정한 연구로는 앞서 여러 번 언급하였던 이긍희 외(2020)와 Cho and Kim(2023)이 있다. 이긍희 외(2020)는 구축한 지수의 유용성을 평가하기 위해 불확실성, 월별 산업생산 증가율, 월별 소비자 물가상승률, 월별 원/달러 환율 증가율이 포함된 4변수 VAR 모형으로 일반화 충격반응분석을 수행하였으며, 분석 결과에서는 불확실성 확대가 생산을 감소시키고, 물가와 환율은 증가시키는 것으로 나타났다. 다만, 생산의 경우 2~4개월, 물가는 1~3개월, 환율은 1~2개월만 통계적으로 유의했다. Cho and Kim(2023)의 경우에는 분기별 지표의 유용성을 평가하였으며, 분석 결과에서는 GDP, 고용, 이자율, 소비, 투자가 감소 후 증가하는 모습을, 환율과 물가는 증가 후 감소하는 모습을 나타냈다. 다만, GDP는 0~5분기, 고용은 2~3분기, 금리는 4~10분기, 소비는 0~4분기, 투자는 3분기, 환율은 0~1분기와 4~5분기, 물가는 10분기만 통계적으로 유의했다.

마지막으로 [표 4-3]은 이상의 선행연구들을 요약한 것이다. 주요 실증분석 결과만 요약하면, 불확실성은 경제주체들의 심리 악화, 금융기관의 대출태도 엄격화, 리스크 프리미엄 확대로 인한 유동성 제약 등의 부정적 영향을 발

생시킨다. 또한 불확실성과 생산, 물가, 고용, 환율 등의 관계를 분석한 선행 연구들은 불확실성 확대가 생산, 고용을 감소시키고, 환율은 상승(원화의 평가절하)시킨다는 일관적인 결과를 제시하였다. 그러나 물가의 경우에는 선행 연구에서 일관된 결과를 확인할 수가 없었는데, 통상 물가는 총수요 감소(예: 소비 및 투자 위축)로 인한 하방 압력뿐만 아니라 총공급 감소(예: 고용량 감소)로 인한 상방 압력도 동시에 받으므로 이 역시 이론적 직관에 부합되는 결과로 판단된다.

HANSUNG UNIVERSITY

[표 4-3] 우리나라의 경제 불확실성 측정과 파급효과를 분석한 주요 선행연구

저자	분석 기간	불확실성 측정 방법	관심 변수	충격반응분석 결과(5% 유의수준)[1]
이현창·정원석(2016)	2003년 1분기 2015년 3분기	Haddow et al.(2013)	GDP갭률 소비자 물가상승률	GDP: 감소(1~3분기) → 증가 물가: 감소(2~5분기)
정원석 외(2016)	2003년 1분기 2016년 1분기		소비자 심리지수 제조업 업황 BSI 신용스프레드 가계 대출태도 대기업 대출태도	CSI: 감소(1~2분기) → 증가 BSI: 감소(1~3분기) → 증가 스프레드: 증가(1~3분기) 대출태도(가계): 감소(1분기) → 증가 대출태도(대기업): 감소(1분기) → 증가
김광민(2021)	2003년 1분기 2020년 1분기		소비자 심리지수 제조업 업황 BSI 신용스프레드 가계 대출태도 대기업 대출태도	CSI: 감소(0~2분기) → 증가(5~7분기) BSI: 감소(0~3분기) → 증가 스프레드: 증가(0~2분기) 대출태도(가계): 감소 → 증가 대출태도(대기업): 감소 → 증가
이항용·이진(2017)	2000년 1월 2017년 6월	Jurado et al.(2015)	산업생산 증가율 소비자 물가상승률	생산: 감소(1~2개월) → 증가 물가: 감소(1개월) → 증가
이긍희 외(2020)[2]	1990년 1월 2020년 4월	Baker et al.(2016)	산업생산 증가율 소비자 물가상승률 원/달러 환율 증가율	생산: 감소(2~4개월) 물가: 증가(1~3개월) 환율: 증가(1~2개월)
Cho and Kim(2023)[3]	1990년 1분기 2019년 4분기		실질 GDP 고용량 이자율 원/달러 환율 인플레이션 실질 소비 실질 투자	GDP: 감소(0~5분기) → 증가 고용: 감소(2~3분기) → 증가 금리: 감소(4~10분기) → 증가 환율: 증가(0~1분기) → 감소(4~5분기) 물가: 증가 → 감소(10분기) 소비: 감소(0~4분기) → 증가 투자: 감소(3분기) → 증가

1) 통계적으로 유의하게 나타난 결과는 소괄호로 표기하였으며, Cho and Kim(2023)의 경우 다른 연구들과 달리 신뢰구간이 90%임
2) GDP 관련 통계와 상관관계가 가장 높은 EPU_KR_6을 기준으로 작성되었으며, 충격반응분석 결과는 변수 순서에 의존하지 않는 일반화 충격반응분석의 수행 결과를 나타냄
3) Cho and Kim(2023)의 충격반응분석 결과는 EPU 충격에 대한 지역 예측(local projection)의 추정 결과임

HANSUNG UNIVERSITY

4.3.3 실증분석: 경제 불확실성이 거시경제변수에 미치는 영향

실증분석은 앞서 검토한 선행연구를 따라 월별 경제변수를 이용한 분석과 분기별 경제변수를 이용한 분석으로 나누어 수행한다. 월별 경제변수를 이용한 분석에서는 이항용·이진(2017)과 이긍희 외(2020)를 따라 불확실성이 거시경제변수에 미치는 영향을 분석하고, 분기별 경제변수를 이용한 분석에서는 이현창·정원석(2016), 정원석 외(2016), 김광민(2021)을 따라 불확실성의 파급경로별 영향을 분석한다.

4.3.3.1 월별 경제변수를 이용한 분석

1) 변수 선정 및 단위근 검정

분석에 사용된 월별 통계 자료는 [표 4-4]에 제시되어있다. 불확실성 대리변수는 Baker et al.(2016)과 Cho and Kim(2023), 그리고 본 연구에서 작성한 EPU 지수를 각각 사용한다. 분석 대상이 되는 거시경제변수는 이항용·이진(2017)과 이긍희 외(2020)에서 고려한 전산업 생산지수, 소비자물가지수, 원/달러 환율이며, 불확실성이 주가 및 고용에 미치는 영향도 분석하기 위해 KOSPI와 취업자 수도 분석 대상으로 고려하였다. 자료 수집 기간은 본 연구의 EPU 지수 작성 기간과 같은 2008년 1월~2022년 12월이다.

[표 4-4] 월별 자료

변수	사용 변수	표기	출처
불확실성	EPU	EPU	EPU 웹사이트
	EPU_KOREA	EPU_KOREA	EPU 웹사이트
	EPU_small	EPU_small	저자 측정
	EPU_BERT	EPU_BERT	저자 측정
생산	전산업생산지수 (농림어업 제외, 계절조정)	IP	한국은행
물가	소비자물가지수	CPI	한국은행

환율	원/달러 환율(평균)	ER	한국은행
주가	KOSPI(평균)	KOSPI	한국은행
고용	취업자 수(계절조정)	EMP	한국은행

단위근 검정 결과는 [표 4-5]와 같다. 단위근 검정은 ADF 검정을 수행하였고,[141] 적정 시차는 SC를 기준으로 선정하였다. 검정 결과, EPU 지수는 모두 1% 유의수준에서 단위근이 존재하지 않는 안정적 시계열로 판명되었으며, 원/달러 환율은 5% 유의수준에서 안정적 시계열로 판명되었다. 반면 전산업 생산지수, 소비자물가지수, KOSPI, 취업자 수는 수준 변수에서 불안정 시계열로 판명되었으나, 1차 차분 자료에서는 모두 안정적 시계열로 나타났다.

[표 4-5] 월별 자료 단위근 검정 결과

변수		ADF 통계량	유의확률	기각 여부
수준 변수	EPU	-5.445	0.000	Yes
	EPU_KOREA	-4.563	0.000	Yes
	EPU_small	-4.467	0.000	Yes
	EPU_BERT	-4.553	0.000	Yes
	IP	-0.294	0.922	No
	CPI	-0.172	0.938	No
	ER	-2.941	0.042	Yes
	KOSPI	-1.827	0.366	No
	EMP	-0.185	0.936	No
1차 차분	EPU	-	-	-
	EPU_KOREA	-	-	-
	EPU_small	-	-	-
	EPU_BERT	-	-	-
	IP	-16.204	0.000	Yes
	CPI	-10.207	0.000	Yes
	ER	-	-	-
	KOSPI	-10.537	0.000	Yes
	EMP	-13.246	0.000	Yes

141) PP 검정에서도 같은 결과가 나타났다.

2) 그랜저 인과관계 검정

다음으로 각 EPU 지수와 월별 경제변수 간의 그랜저 인과관계 검정을 수행하였다. 단위근 검정 결과를 근거로 EPU 지수와 원/달러 환율은 수준 변수를 사용하였으며, 나머지 변수들은 모두 전기대비 증가율 자료를 사용하였다. 또한 최적 시차는 1~12개월 내에 SC를 최소로 하는 시차로 선정하였다.

그랜저 인과관계 검정 결과는 [표 4-6]과 같으며, 각 경제변수에 대해 외생성이 가장 뚜렷하게 나타난 EPU 지수는 음영으로 표시하였다. 먼저, 산업생산 증가율은 EPU와 상호 그랜저 인과관계가 없는 것으로 나타났으며, EPU_KOREA는 오히려 산업생산 증가율이 EPU_KOREA에 일방적인 그랜저 인과관계가 있는 것으로 나타났다. EPU_small과 EPU_BERT는 산업생산 증가율과 상호 영향을 주고받는 그랜저 인과관계가 나타났으며, 특히 EPU_small은 1% 유의수준에서 일방적인 그랜저 인과관계가 있는 것으로 나타났다. 물가상승률은 EPU와 EPU_KOREA가 상호 간의 그랜저 인과관계가 없는 것으로 나타났으며, EPU_small과 EPU_BERT는 오히려 물가상승률이 두 지수보다 외생성이 강한 것으로 나타났다. 원/달러 환율은 EPU와 EPU_KOREA가 일방적인 그랜저 인과관계를 나타냈으며, EPU_small과 EPU_BERT는 상호 영향을 주고받는 그랜저 인과관계가 있는 것으로 나타났다. 다만, 1% 유의수준에서는 EPU_smll과 EPU_BERT도 일방적인 그랜저 인과관계를 나타냈으며, EPU는 10% 유의수준에서만 일방적인 그랜저 인과관계를 나타냈다. 주가수익률은 EPU가 10% 유의수준에서 일방적인 그랜저 인과관계를 나타냈으며, EPU_KOREA는 상호 그랜저 인과관계가 없는 것으로 나타났다. 반면, EPU_small과 EPU_BERT는 주가수익률에 대해 일방적인 그랜저 인과관계가 있는 것으로 나타났다. 마지막으로 취업자 수 증가율은 EPU와 EPU_KOREA의 경우 상호 그랜저 인과관계가 없는 것으로 나타났지만, EPU_small과 EPU_BERT는 취업자 수 증가율에 대해 일방적인 그랜저 인과관계가 있는 것으로 나타났다.

[표 4-6] 월별 자료 그랜저 인과관계 검정 결과

경제변수	귀무가설	F 통계량[1)]	유의확률
IP	EPU ⇏ IP	0.001	0.973
	IP ⇏ EPU	1.677	0.197
	EPU_KOREA ⇏ IP	1.476	0.225
	IP ⇏ EPU_KOREA	13.487***	0.000
	EPU_small ⇏ IP	7.341***	0.007
	IP ⇏ EPU_small	4.918**	0.027
	EPU_BERT ⇏ IP	6.186**	0.013
	IP ⇏ EPU_BERT	4.556**	0.034
CPI	EPU ⇏ CPI	1.021	0.362
	CPI ⇏ EPU	2.015	0.136
	EPU_KOREA ⇏ CPI	1.067	0.3
	CPI ⇏ EPU_KOREA	1.350	0.246
	EPU_small ⇏ CPI	1.873	0.172
	CPI ⇏ EPU_small	6.146**	0.014
	EPU_BERT ⇏ CPI	1.447	0.23
	CPI ⇏ EPU_BERT	6.347**	0.012
ER	EPU ⇏ ER	3.004*	0.084
	ER ⇏ EPU	2.654	0.105
	EPU_KOREA ⇏ ER	35.545***	0.000
	ER ⇏ EPU_KOREA	0.397	0.529
	EPU_small ⇏ ER	29.832***	0.000
	ER ⇏ EPU_small	3.97**	0.047
	EPU_BERT ⇏ ER	28.899***	0.000
	ER ⇏ EPU_BERT	3.639*	0.058
KOSPI	EPU ⇏ KOSPI	3.008*	0.084
	KOSPI ⇏ EPU	1.83	0.177
	EPU_KOREA ⇏ KOSPI	0.002	0.961
	KOSPI ⇏ EPU_KOREA	0.377	0.539
	EPU_small ⇏ KOSPI	8.036***	0.000
	KOSPI ⇏ EPU_small	0.946	0.39
	EPU_BERT ⇏ KOSPI	7.386***	0.000
	KOSPI ⇏ EPU_BERT	1.16	0.315

EMP	EPU ⇏ EMP	0.003	0.952
	EMP ⇏ EPU	0.507	0.477
	EPU_KOREA ⇏ EMP	1.33	0.25
	EMP ⇏ EPU_KOREA	0.06	0.805
	EPU_small ⇏ EMP	6.364**	0.012
	EMP ⇏ EPU_small	0.072	0.788
	EPU_BERT ⇏ EMP	5.822**	0.016
	EMP ⇏ EPU_BERT	0.01	0.92

주: 1) *, **, ***는 각각 10%, 5%, 1% 수준에서 유의함을 의미

3) VAR 모형 설계

VAR 모형은 선행연구를 참고하여 VAR-A와 VAR-B 2개의 모형을 설계하였다. VAR-A는 국내 선행연구를 따라 EPU 지수, 산업생산 증가율, 물가상승률, 원/달러 환율을 고려한 4변수 VAR 모형으로 설계하였으며, VAR-B는 Baker et al.(2016)을 따라 EPU 지수, 산업생산 증가율, 취업자 수 증가율, 주가수익률을 고려한 4변수 VAR 모형으로 설계하였다. 다만, 변수들 사이에 단기균형 상태를 파악하는 데에는 VAR 모형을 활용하는 데 큰 문제가 없으므로 공적분 검정과 VEC 모형은 고려하지 않았다. 더욱이, EPU 지수가 모두 수준 변수인 점을 고려하면 VAR 모형이 VEC 모형보다 더 유의미한 결과를 도출할 것으로 판단하였다.[142)]

모형의 추정 기간은 2008년 1월~2022년 12월이며, 최적 시차는 1~12개월 내에서 SC를 최소로 하는 시차로 선정하였다. 변수의 배열순서는 VAR-A의 경우 국내 선행연구들을 따라 EPU 지수, 산업생산 증가율, 물가상승률, 원/달러 환율 순으로 설계하였으며, VAR-B는 Baker et al.(2016)을 따라 EPU 지수, 주가수익률, 취업자 수 증가율, 산업생산 증가율 순으로 설계하였다. 그랜저 인과관계 검정에서는 불확실성 지수보다 물가상승률의 외생성이 더 강

142) 수준 변수에서 단위근이 존재하지 않는 자료를 1차 차분한다면 해당 자료가 가지고 있는 경제적 의미가 손실될 수 있으며, 이 경우 $I(1)$ 계열로 수행되는 VEC 모형의 분석 결과가 무의미한 분석 결과로 귀결될 수 있다. 게다가 VEC 모형은 변수간에 장기적인 관계가 조금이라도 잘못 설정될 경우, 모형의 유효성이 크게 훼손될 수 있다는 단점도 존재한다.

하게 나타났으나, 본 분석의 목적이 불확실성의 파급효과를 추정하고 분석 결과를 선행연구와 비교하는 데 있으므로 선행연구와 경제이론을 따라 배열순서를 결정하였다. 이에 따라 VAR-A 모형과 VAR-B 모형의 추정 식은 시차길이를 1로 가정하였을 때 다음 식 (4-28)과 (4-29)와 같이 정의되며, 이때 *EPU*는 4개의 불확실성 지수(EPU, EPU_small, EPU_BERT, EPU_KOREA) 각각을 의미한다.

(4-28) VAR-A

$$EPU_t = \alpha_1 + \sum_{i=1}^{1}\beta_{1i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{1i}IP_{t-i} + \sum_{i=1}^{1}\delta_{1i}CPI_{t-i} + \sum_{i=1}^{1}\zeta_{1i}ER_{t-i} + \epsilon_{1t}$$

$$IP_t = \alpha_2 + \sum_{i=1}^{1}\beta_{2i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{2i}IP_{t-i} + \sum_{i=1}^{1}\delta_{2i}CPI_{t-i} + \sum_{i=1}^{1}\zeta_{2i}ER_{t-i} + \epsilon_{2t}$$

$$CPI_t = \alpha_3 + \sum_{i=1}^{1}\beta_{3i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{3i}IP_{t-i} + \sum_{i=1}^{1}\delta_{3i}CPI_{t-i} + \sum_{i=1}^{1}\zeta_{3i}ER_{t-i} + \epsilon_{3t}$$

$$ER_t = \alpha_4 + \sum_{i=1}^{1}\beta_{4i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{4i}IP_{t-i} + \sum_{i=1}^{1}\delta_{4i}CPI_{t-i} + \sum_{i=1}^{1}\zeta_{4i}ER_{t-i} + \epsilon_{4t}$$

(4-29) VAR-B

$$EPU_t = \alpha_5 + \sum_{i=1}^{1}\beta_{5i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{5i}KOSPI_{t-i} + \sum_{i=1}^{1}\delta_{5i}EMP_{t-i} + \sum_{i=1}^{1}\zeta_{5i}IP_{t-i} + \epsilon_{5t}$$

$$KOSPI_t = \alpha_6 + \sum_{i=1}^{1}\beta_{6i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{6i}KOSPI_{t-i} + \sum_{i=1}^{1}\delta_{6i}EMP_{t-i} + \sum_{i=1}^{1}\zeta_{6i}IP_{t-i} + \epsilon_{6t}$$

$$EMP_t = \alpha_7 + \sum_{i=1}^{1}\beta_{7i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{7i}KOSPI_{t-i} + \sum_{i=1}^{1}\delta_{7i}EMP_{t-i} + \sum_{i=1}^{1}\zeta_{7i}IP_{t-i} + \epsilon_{7t}$$

$$IP_t = \alpha_8 + \sum_{i=1}^{1}\beta_{8i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{8i}KOSPI_{t-i} + \sum_{i=1}^{1}\delta_{8i}EMP_{t-i} + \sum_{i=1}^{1}\zeta_{8i}IP_{t-i} + \epsilon_{8t}$$

4) 충격반응분석 결과

충격반응분석은 Cholesky 분해 방법을 이용해 수행하였으며, 분석 결과는 [그림 4-4]~[그림 4-11]에 걸쳐 제시하였다. 여기서 [그림 4-4]~[그림 4-7]은 VAR-A 모형의 분석 결과를, [그림 4-8]~[그림 4-11]은 VAR-B 모형의

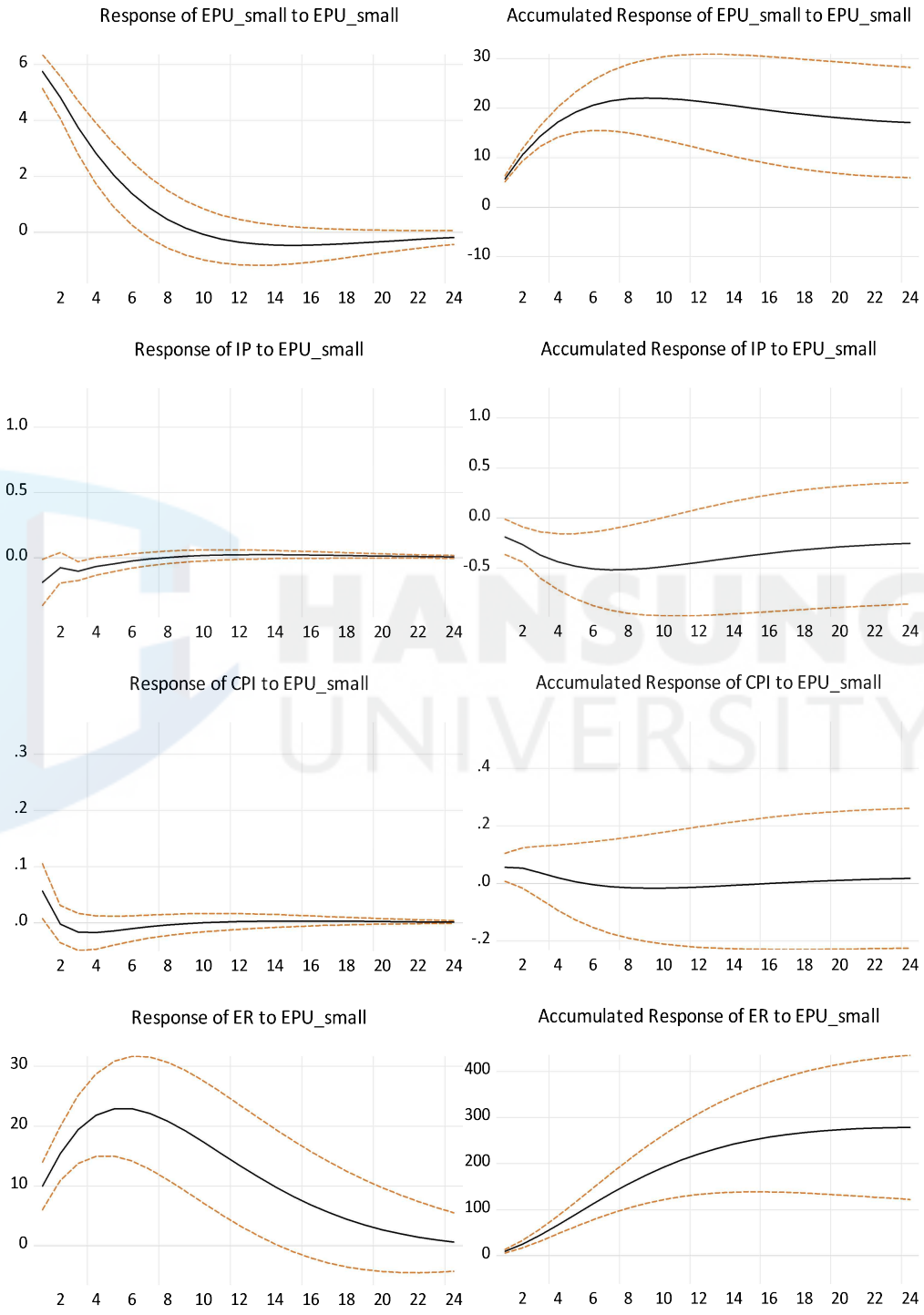
분석 결과를 보여주며, 각 그림의 빨간색 점선은 95% 신뢰구간을 의미한다.

(가) VAR-A 모형의 충격반응분석 결과

먼저, [그림 4-4]는 EPU_small을 이용한 VAR-A 모형의 충격반응분석 결과이다. 불확실성에 양의 충격이 발생했을 때(충격의 양은 5.75P), 산업생산 증가율은 1개월에 가장 큰 하락을 보였으며(0.19%P 하락), 음의 증가율은 7개월까지 지속되었다. 또한, 충격에 대한 반응을 누적하여 살펴보면, 불확실성은 산업생산 증가율을 장기적으로 0.51%P까지 하락시키는 것으로 나타났다. 이때 통계적으로 유의한 구간은 충격반응분석의 경우 1개월과 3~5개월, 누적 충격반응분석의 경우 1~10개월이다. 원/달러 환율은 충격 발생 시 6개월에 가장 큰 상승을 보였으며(월평균 23원 상승으로 원화의 평가절하), 충격의 효과는 14개월까지 유의한 것으로 나타났다. 그러나 물가상승률의 경우에는 전반적으로 유의성이 낮게 나타났다. 구체적으로, 불확실성에 양의 충격이 발생했을 때, 물가상승률은 1개월에만 유의하게 0.056%P 상승하였고, 이후에는 오히려 9개월까지 물가상승률을 하락시키는 것으로 나타났다. 대부분의 선행 연구는 불확실성이 물가를 하락시킨다는 분석 결과를 제시하였으나, 앞서 언급하였듯이 물가는 총수요 감소로 인한 하방 압력과 총공급 위축으로 인한 상방 압력이 동시에 작용한다. 특히, 최근 코로나 팬데믹 기간에는 생산 중단, 물류 문제, 노동력 부족 등으로 인해 글로벌공급망압력지수(Global Supply Chain Pressure Index, GSCPI)가 급등하였고, 이로 인해 국내 물가상승률은 2020년 5월부터 상승하기 시작하여 2022년 7월에는 전년동기대비 6.3%까지 상승하였다(김인수, 2024). 더욱이, 본 연구의 EPU 지수는 그랜저 인과관계 검정에서도 물가상승률에 대해 뚜렷한 외생성이 발견되지 않았으므로 불확실성이 물가에 미치는 영향을 분석하기 위해서는 좀 더 공급 부문에 특화된 불확실성 지수의 작성이 필요해 보인다.[143]

143) 이와 관련하여 김인수(2024)는 팬데믹 기간 동안 공급망 교란 및 이와 관련된 불확실성이 물가, 고용, 임금 등에 미치는 영향을 분석한 바 있다.

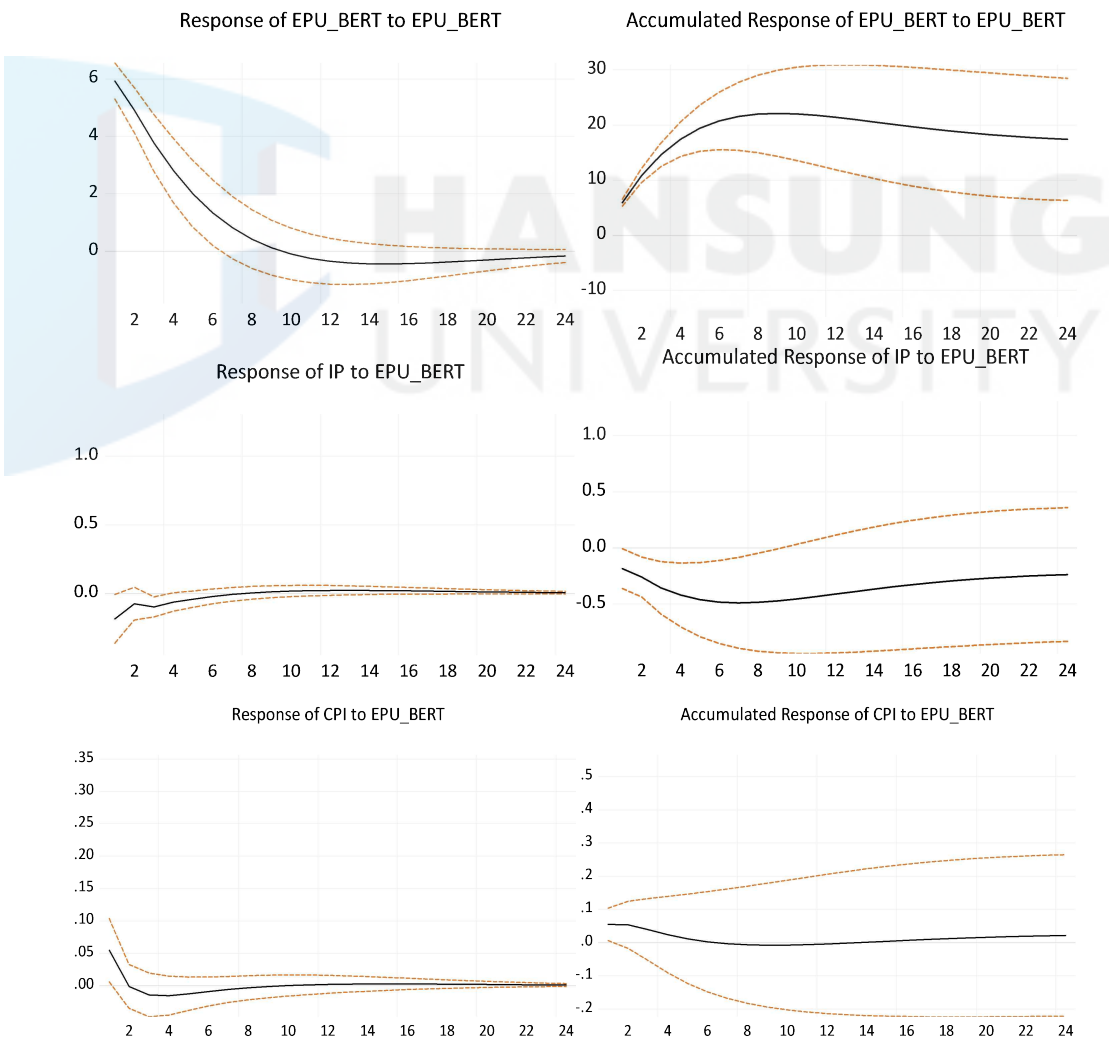
[그림 4-4] EPU_small의 VAR-A 모형 충격반응분석 결과

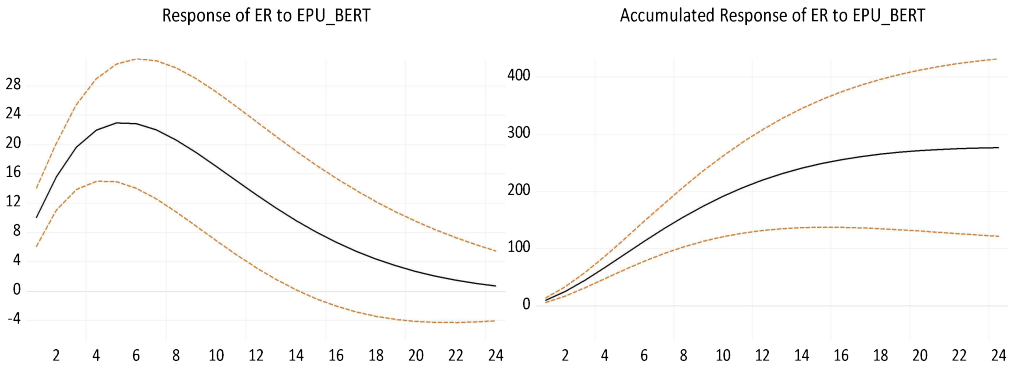


HANSUNG UNIVERSITY

[그림 4-5]는 EPU_BERT를 이용한 VAR-A 모형의 충격반응분석 결과이다. 분석 결과는 EPU_small과 매우 유사하다. 구체적으로, 불확실성에 5.92P 충격이 발생했을 때, 산업생산의 음의 증가율은 7개월까지 지속되었고(1개월에 0.18%P 하락), 충격의 반응을 누적하였을 때는 산업생산 증가율이 장기적으로 0.49%P까지 하락하였다. 이때, 통계적으로 유의한 구간은 충격반응분석은 1개월과 3~4개월, 누적충격반응분석은 1~9개월이다. 또한, 원/달러 환율은 5개월에 가장 큰 상승을 나타냈으며(월평균 23원 상승으로 원화의 평가절하), 물가상승률은 1개월에만 유의하게 0.05%P 상승하였다.

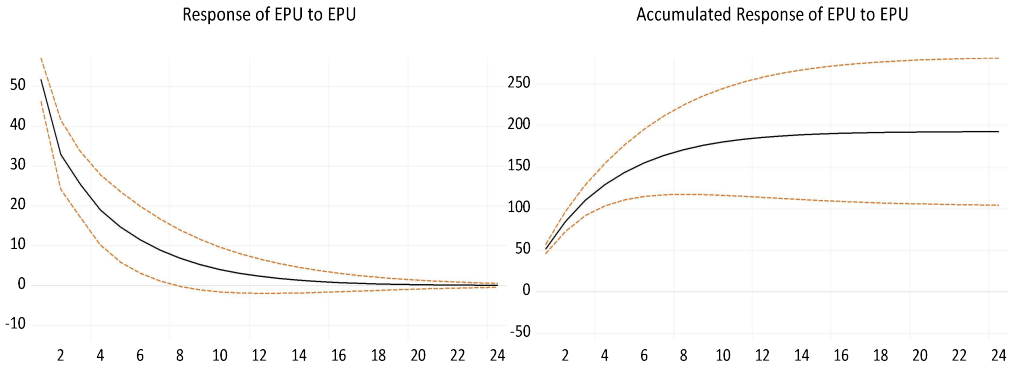
[그림 4-5] EPU_BERT의 VAR-A 모형 충격반응분석 결과

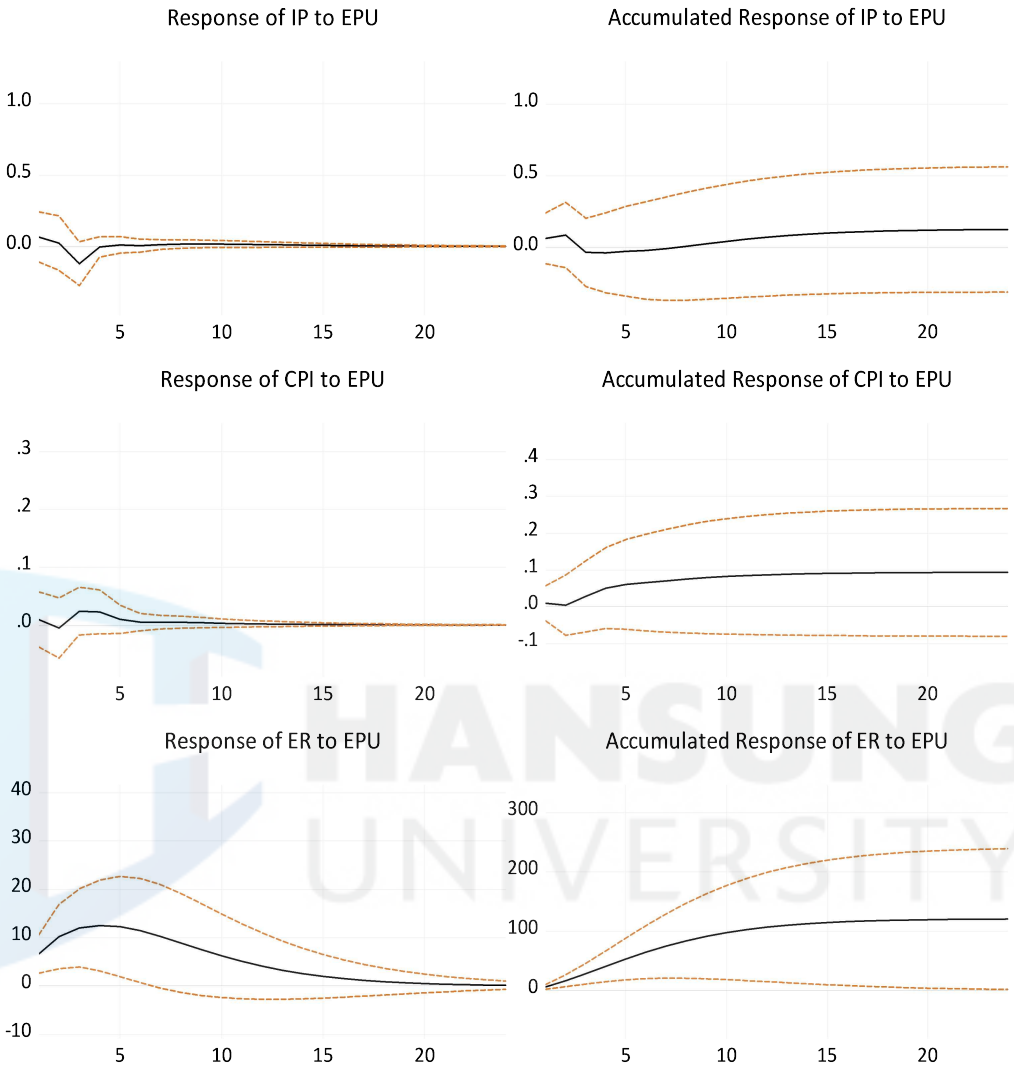




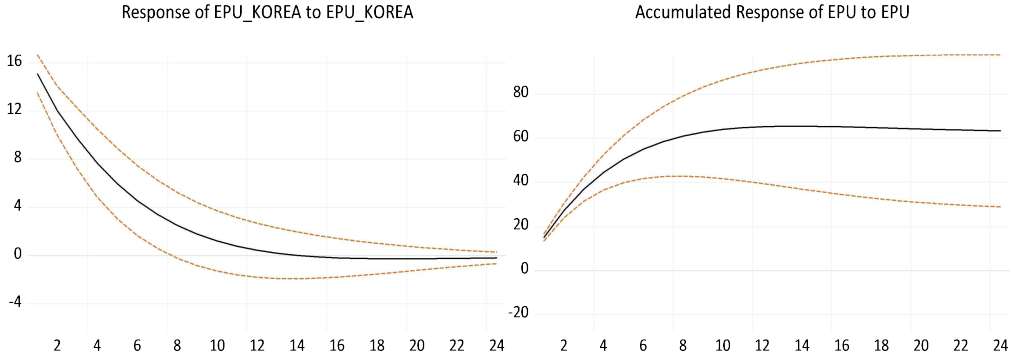
[그림 4-6]과 [그림 4-7]은 EPU와 EPU_KOREA를 이용한 VAR-A 모형의 충격반응분석 결과이다. EPU와 EPU_KOREA는 원/달러 환율을 제외한 나머지 거시경제변수의 반응이 전체 기간에 걸쳐 통계적으로 유의하지 않았다. 원/달러 환율의 분석 결과만 살펴보면, EPU는 불확실성 충격(51.71P)이 원/달러 환율을 1~7개월 유의하게 상승시켰고, 반응의 크기는 5개월이 가장 크게 나타났다(월평균 12원 상승으로 원화의 평가절하). EPU_KOREA는 불확실성 충격(15.07P)이 원/달러 환율을 1~14개월 유의하게 상승시켰고, 반응의 크기는 6개월이 가장 크게 나타났다(월평균 24원 상승으로 원화의 평가절하). 또한 EPU와 EPU_KOREA는 통계적으로 유의한 결과는 아니었지만, 불확실성 확대가 물가상승률을 상승시키는 것으로 분석되었다.

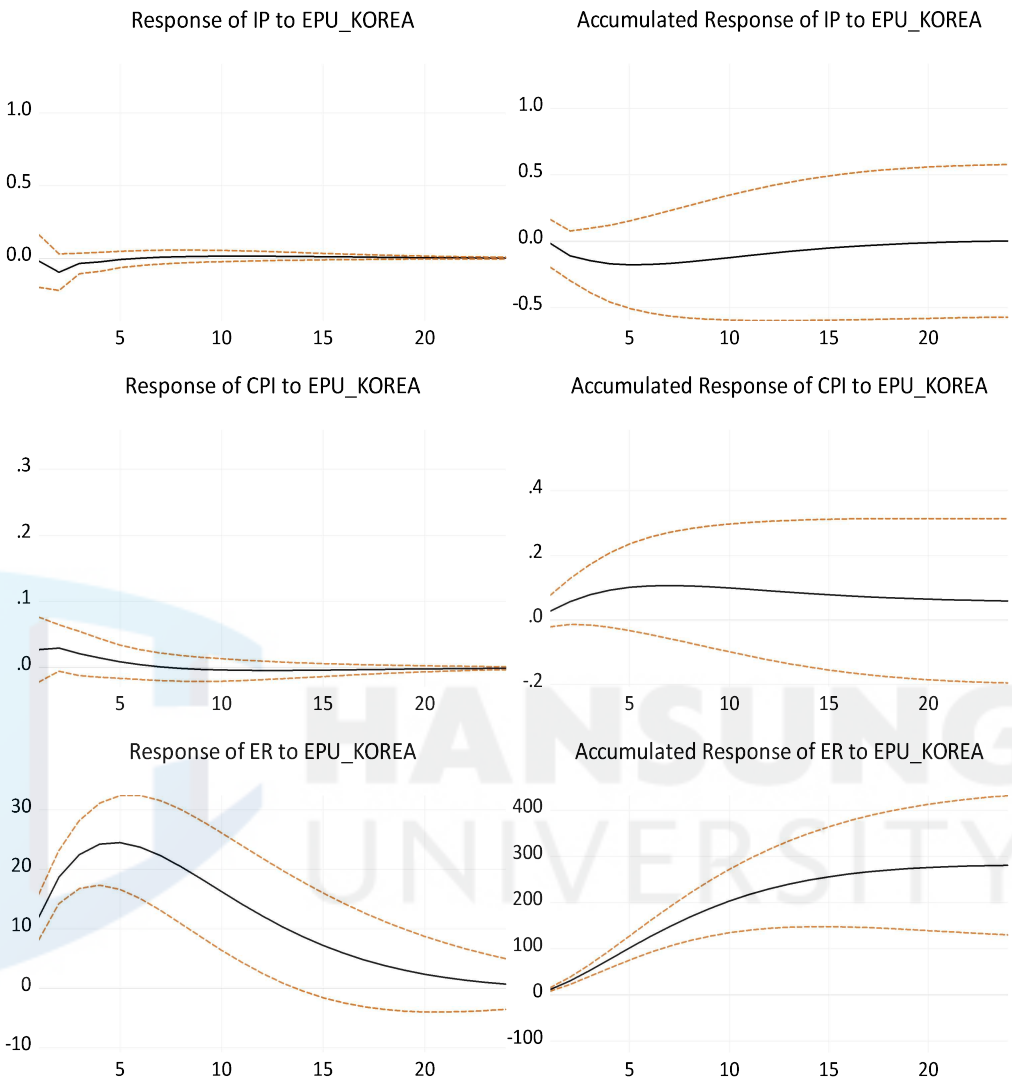
[그림 4-6] EPU의 VAR-A 모형 충격반응분석 결과





[그림 4-7] EPU_KOREA의 VAR-A 모형 충격반응분석 결과



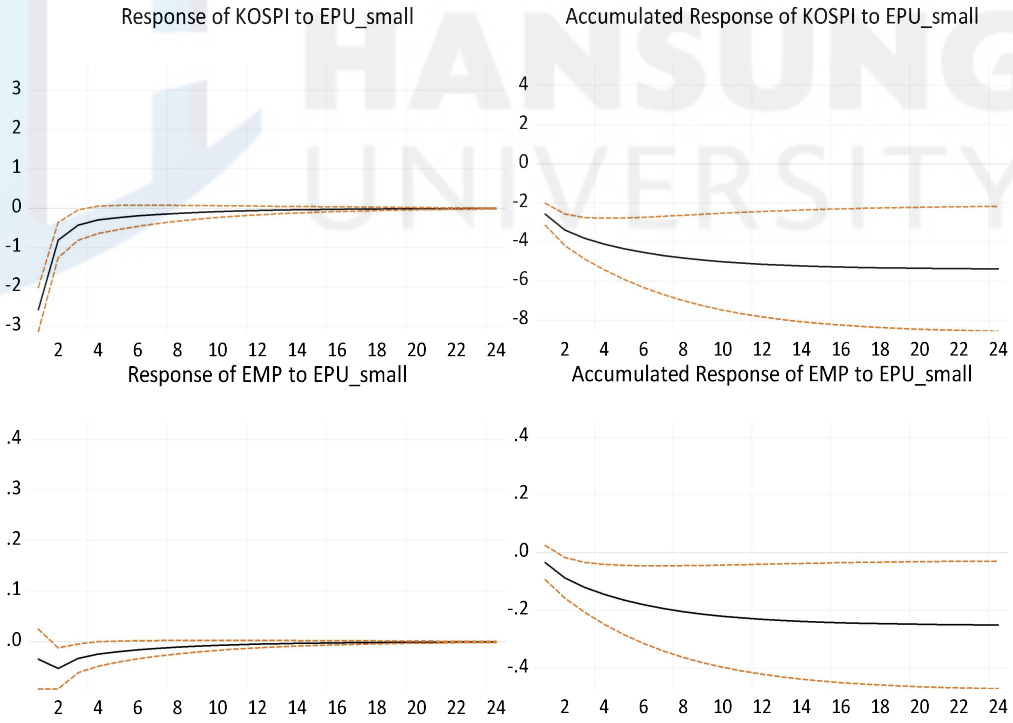


(나) VAR-B 모형의 충격반응분석 결과

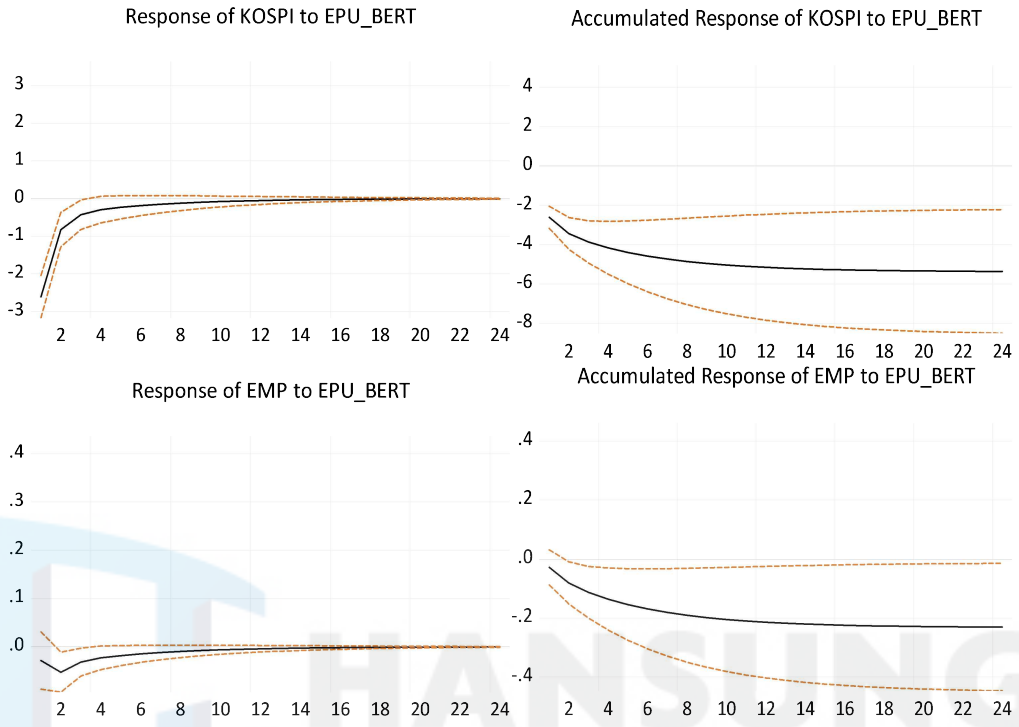
[그림 4-8]과 [그림 4-9]는 EPU_small과 EPU_BERT를 이용한 VAR-B 모형의 충격반응분석 결과이다. 두 EPU 지수는 모두 불확실성 확대가 주가수익률과 취업자 수 증가율을 유의하게 감소시킨다는 분석 결과가 제시되었다. 구체적으로, EPU_small은 불확실성에 양의 충격(5.9P)이 발생하였을 때 주가수익률을 1~3개월, 취업자 수 증가율을 2~4개월 유의하게 감소시켰으며, 이

때 주가수익률의 반응 크기는 1개월 -2.57%P, 2개월 -0.8%P, 3개월 -0.42%P, 취업자 수 증가율의 반응 크기는 2개월 -0.05%P, 3개월 -0.03%P, 4개월 -0.02%P로 나타났다. 또한 누적충격반응분석에서는 주가수익률이 장기적으로 5.37%P 감소하였고, 취업자 수 증가율은 장기적으로 0.25%P 감소하였다. EPU_BERT도 이와 같은 분석 결과가 제시되었다. 불확실성에 양의 충격(6.07P)이 발생하였을 때 주가수익률은 1~3개월, 취업자 수 증가율은 2~3개월 유의하게 감소하였으며, 주가수익률의 반응 크기는 1개월 -2.6%P, 2개월 -0.82%P, 3개월 -0.43%P, 취업자 수 증가율의 반응 크기는 2개월 -0.05%P, 3개월 -0.03%P, 4개월 -0.02%P로 나타났다. 또한 누적충격반응분석은 주가수익률이 장기적으로 5.36%P 감소하였고, 취업자 수 증가율은 장기적으로 0.23%P 감소하였다.

[그림 4-8] EPU_small의 VAR-B 모형 충격반응분석 결과

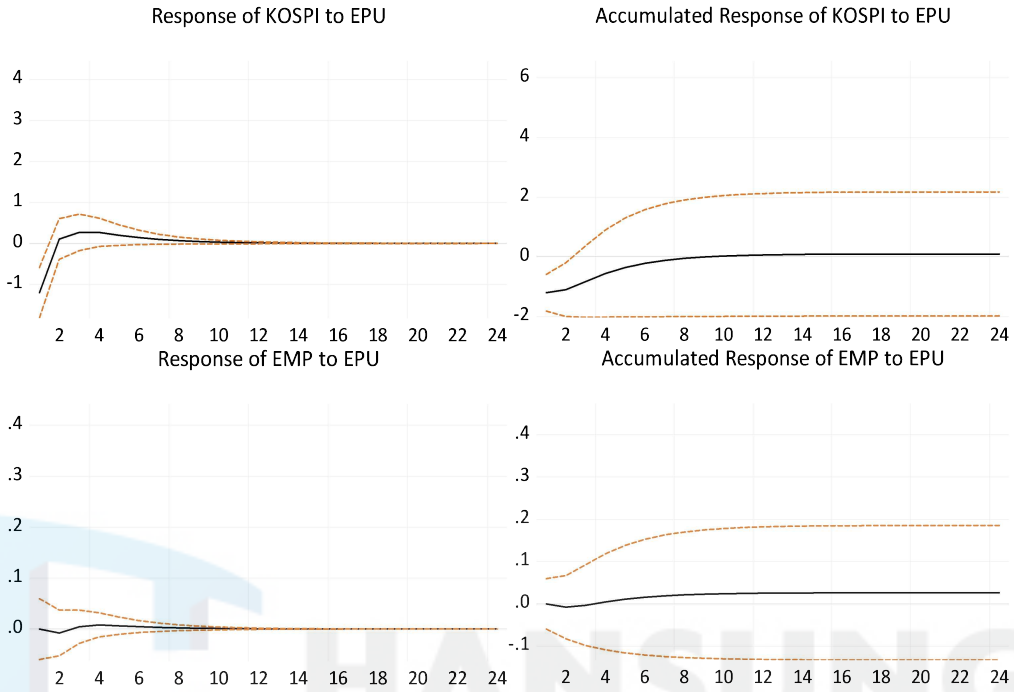


[그림 4-9] EPU_BERT의 VAR-B 모형 충격반응분석 결과

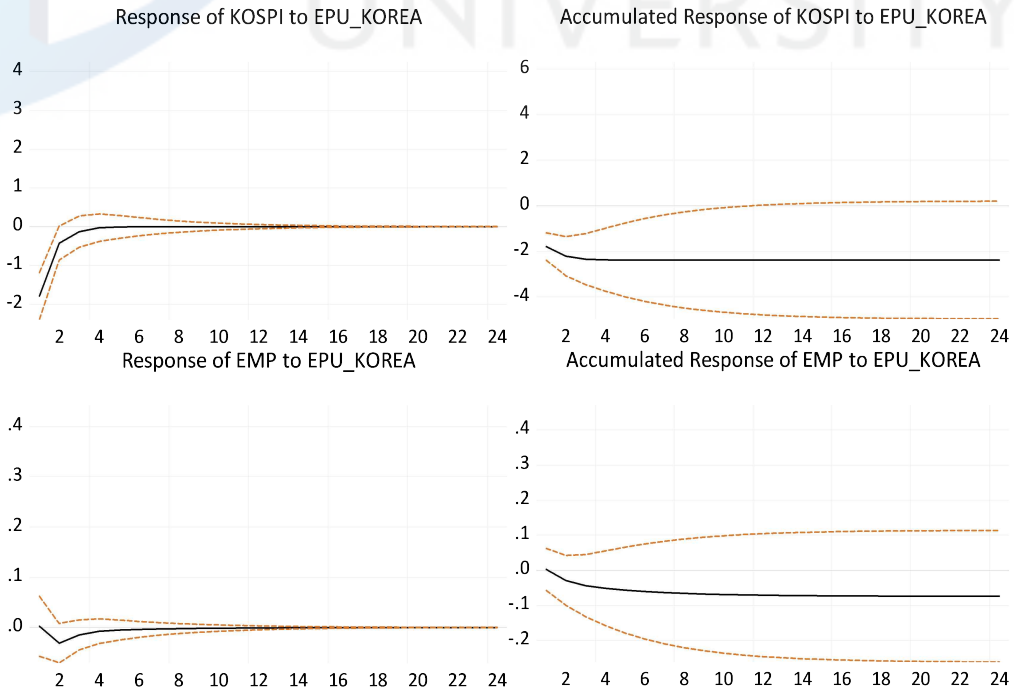


마지막으로, [그림 4-10]과 [그림 4-11]은 EPU와 EPU_KOREA를 이용한 VAR-B 모형의 충격반응분석 결과이다. 두 EPU 지수의 경우에는 주가수익률에 대해서는 통계적으로 유의한 결과가 나타났으나, 취업자 수 증가율에 대해서는 전체 기간이 통계적으로 유의하지 않았다. 구체적으로, EPU는 불확실성에 양의 충격(51.82P)이 발생하였을 때 주가수익률을 1개월에 1.2%P 유의하게 감소시켰으며, 2개월부터는 통계적으로 유의하진 않으나 주가수익률을 증가시키는 것으로 분석되었다. 취업자 수 증가율은 통계적으로 유의하진 않았으나 1~2개월에 감소한 이후, 3개월부터 증가하는 것으로 나타났다. EPU_KOREA는 불확실성에 양의 충격(15.1P)이 발생하였을 때 주가수익률은 1개월에 1.78%P, 2개월에 0.42%P 유의하게 감소하였으며, 취업자 수 증가율은 통계적으로 유의하진 않았으나 1개월에 증가한 이후 2개월부터 감소하는 것으로 나타났다.

[그림 4-10] EPU의 VAR-B 모형 충격반응분석 결과



[그림 4-11] EPU_KOREA의 VAR-B 모형 충격반응분석 결과



4.3.3.2 분기별 경제변수를 이용한 분석

분기별 경제변수를 이용한 분석에서는 불확실성이 거시경제에 미치는 파급경로별 영향을 분석한다. 파급경로는 이현창·정원석(2016) 등이 고려한 ① 심리 경로와 비용 경로를 기반으로 하며, 이때 비용 경로는 ② 리스크 프리미엄 경로와 ③ 대출태도 경로로 세분된다. 또한, 여기서는 코로나 팬데믹 기간을 제외한 실증분석 결과를 제시한다. 코로나19와 같이 변동성이 지나치게 확대된 시기를 분석에 포함할 경우, 분석 결과가 동 기간에 크게 좌우될 우려가 있다. 더욱이, 앞에 [그림 4-1]에서 EPU_small과 EPU_KOREA는 코로나19 기간에 매우 큰 차이를 보이고 있지만, 2008년~2019년 기간에서는 꽤 유사한 추이를 보이고 있다. 따라서 코로나19 기간을 분석에서 제외하면 EPU 지수 간 비교의 객관성을 높이고, 정상적인 경제 환경에서의 정합성과 설명력을 보다 면밀히 평가할 수 있을 것이다.

1) 변수 선정 및 단위근 검정

분석에 사용된 자료는 [표 4-7]에 제시하였다. 불확실성 대리변수는 EPU, EPU_KOREA, EPU_small, EPU_BERT를 각각 사용하였으며, 경제성장률과 금리는 한국은행의 국내총생산(실질, 계절조정, 전기비)과 콜금리 자료를 사용하였다. 심리 경로 변수는 한국은행의 소비자 심리지수와 제조업 전망 BSI(계절조정) 자료를, 리스크 프리미엄은 신용스프레드를 사용하였으며, 대출태도는 한국은행의 대기업 대출태도지수와 가계(일반) 대출태도지수를 사용하였다. 이때 심리 변수 자료들은 월별 자료를 분기 평균하여 산출하였으며, 신용스프레드는 회사채 3년(AA-)에서 콜금리를 차감하여 산출하였다. 또한 자료 수집 기간은 2008년 1분기~2019년 4분기이며, 심리 변수는 작성 기간으로 인해 시작 시점이 2009년 1분기부터이다.

[표 4-7] 분기별 자료

변수	사용변수	표기	표기
필수 변수: 2008년 1분기~2019년 4분기			
불확실성	EPU	EPU	EPU 웹사이트
	EPU_KOREA	EPU_KOREA	EPU 웹사이트
	EPU_small	EPU_small	저자 측정
	EPU_BERT	EPU_BERT	저자 측정
경제성장률	국내총생산 (실질, 계절조정, 전기비)	GDP	한국은행
금리	콜금리	CR	한국은행
관심 변수: 2008년 1분기~2019년 4분기(단, 심리 변수는 2009년 1분기~)			
심리	소비자 심리지수	CCSI	한국은행
	제조업 전망 BSI (계절조정)	BSI	한국은행
리스크 프리미엄	신용스프레드 (회사채 - 콜금리)	SPREAD	한국은행
대출태도	대기업 대출태도지수	BLTf	한국은행
	가계(일반) 대출태도지수	BLTh	한국은행

ADF 검정 결과는 [표 4-8]과 같다.[144] 여기서 BLTf와 BLTh는 상수항을 고려하지 않은, 즉 식 (4-5)에 의한 단위근 검정 결과를 보여주며, 나머지 변수들은 식 (4-6)에 의한 단위근 검정 결과를 보여준다.[145] 검정 결과, 신용스프레드를 제외한 모든 변수가 5% 유의수준에서 단위근이 존재하지 않는 안정적 시계열로 판명되었으며, 신용스프레드는 1차 차분 자료에서 안정적 시계열로 판명되었다. 따라서 신용스프레드를 제외한 모든 변수는 수준 변수가 분석에 사용되며, 신용스프레드는 1차 차분 자료가 분석에 사용된다.

144) PP 검정에서도 같은 결과가 나타났으며, 적정 시차는 SC를 기준으로 선정하였다.

145) 한국은행의 대출태도지수는 0을 기준으로 지수가 양(+)이면 대출태도 완화라고 응답한 금융기관의 수가 강화라고 응답한 금융기관의 수보다 많다는 것을 의미하며, 음(-)의 값은 그 반대를 의미한다. 따라서 대출태도지수는 자료 특성상 평균이 0인 분포임을 가정하는 것이 적합하다고 판단하였으며, 나머지 변수들은 상수항을 포함하지 않아도 검정 결과가 같았다.

[표 4-8] 분기별 자료 단위근 검정 결과

		ADF 통계량	유의확률	기각 여부
필수 변수: 2008년 1분기~2019년 4분기				
수준 변수	EPU	-3.233	0.024	Yes
	EPU_KOREA	-3.347	0.018	Yes
	EPU_small	-3.108	0.032	Yes
	EPU_BERT	-3.184	0.027	Yes
	GDP	-6.12	0.000	Yes
	CR	-3.422	0.015	Yes
1차 차분	EPU	-	-	-
	EPU_KOREA	-	-	-
	EPU_small	-	-	-
	EPU_BERT	-	-	-
	GDP	-	-	-
	CR	-	-	-
관심 변수: 2008년 1분기~2019년 4분기(단, 심리지수는 2009년 1분기~)				
수준 변수	CCSI	-5.177	0.000	Yes
	BSI	-4.853	0.000	Yes
	SPREAD	-1.337	0.604	No
	BLTf	-2.16	0.03	Yes
	BLTh	-2.048	0.04	Yes
1차 차분	CCSI	-	-	-
	BSI	-	-	-
	SPREAD	-4.961	0.000	Yes
	BLTf	-	-	-
	BLTh	-	-	-

2) 그랜저 인과관계 검정

그랜저 인과관계 검정 결과는 [표 4-9]와 같다. 최적 시차는 1~4분기 중 SC를 최소로 하는 시차로 선정하였으며, 각 관심 변수에 대해 외생성이 가장 뚜렷하게 나타난 EPU 지수는 음영으로 표시하였다. 검정 결과, EPU와 EPU_

KOREA는 모든 관심 변수와 상호 그랜저 인과관계가 없는 것으로 나타났으며, EPU_small과 EPU_BERT는 대부분에 관심 변수에 대해 일방적인 그랜저 인과관계가 나타났다. 구체적으로 EPU_small과 EPU_BERT는 CCSI에 대해 1% 유의수준에서 일방적인 그랜저 인과관계를 나타냈으며, BLTh에 대해서는 5% 유의수준에서, SPREAD, BLTf에 대해서는 10% 유의수준에서 일방적인 그랜저 인과관계를 나타냈다. 다만, BSI의 경우에는 10% 유의수준에서 상호 영향을 주고받는 그랜저 인과관계가 나타났으나, 5% 유의수준에서는 EPU_BERT가 일방적인 그랜저 인과관계를 나타냈다.

[표 4-9] 분기별 자료 그랜저 인과관계 검정 결과

경제변수	귀무가설	F 통계량[1)]	유의확률
CCSI	EPU ⇏ CCSI	0.105	0.746
	CCSI ⇏ EPU	0.528	0.471
	EPU_KOREA ⇏ CCSI	0.446	0.507
	CCSI ⇏ EPU_KOREA	0.032	0.858
	EPU_small ⇏ CCSI	7.964***	0.007
	CCSI ⇏ EPU_small	1.605	0.212
	EPU_BERT ⇏ CCSI	8.523***	0.005
	CCSI ⇏ EPU_BERT	1.723	0.196
BSI	EPU ⇏ BSI	0.03	0.862
	BSI ⇏ EPU	0.127	0.722
	EPU_KOREA ⇏ BSI	0.433	0.514
	BSI ⇏ EPU_KOREA	0.624	0.433
	EPU_small ⇏ BSI	4.000*	0.052
	BSI ⇏ EPU_small	3.961*	0.053
	EPU_BERT ⇏ BSI	4.268**	0.045
	BSI ⇏ EPU_BERT	3.551*	0.066
SPREAD	EPU ⇏ SPREAD	0.279	0.599
	SPREAD ⇏ EPU	0.393	0.533
	EPU_KOREA ⇏ SPREAD	0.390	0.535
	SPREAD ⇏ EPU_KOREA	0.148	0.701
	EPU_small ⇏ SPREAD	4.035*	0.050
	SPREAD ⇏ EPU_small	0.564	0.456

	EPU_BERT ⇏ SPREAD	3.843*	0.056
	SPREAD ⇏ EPU_BERT	0.434	0.513
BLTf	EPU ⇏ BLTf	0.022	0.882
	BLTf ⇏ EPU	0.153	0.696
	EPU_KOREA ⇏ BLTf	0.163	0.687
	BLTf ⇏ EPU_KOREA	0.024	0.876
	EPU_small ⇏ BLTf	3.661*	0.062
	BLTf ⇏ EPU_small	1.234	0.272
	EPU_BERT ⇏ BLTf	3.571*	0.065
	BLTf ⇏ EPU_BERT	0.894	0.349
BLTh	EPU ⇏ BLTh	0.329	0.569
	BLTh ⇏ EPU	0.239	0.627
	EPU_KOREA ⇏ BLTh	0.019	0.889
	BLTh ⇏ EPU_KOREA	0.187	0.667
	EPU_small ⇏ BLTh	4.631**	0.042
	BLTh ⇏ EPU_small	0.565	0.455
	EPU_BERT ⇏ BLTh	5.635**	0.022
	BLTh ⇏ EPU_BERT	0.345	0.559

주: 1) *, **, ***는 각각 10%, 5%, 1% 수준에서 유의함을 의미

3) VAR 모형 설계

VAR 모형은 불확실성의 파급경로별 영향을 분석하기 위해 필수 변수에 관심 변수가 하나씩 추가된 5개의 VAR 모형(VAR-Ca, VAR-Cb, VAR-D, VAR-Ea, VAR-Eb)을 설계하였다. 월별 자료를 분석하였을 때와 같은 이유로 공적분 검정과 VEC 모형은 고려하지 않았으며, 변수의 배열순서도 선행연구들과 똑같이 설정하였다.

VAR-C는 심리 경로를 분석하기 위한 모형으로 소비자 심리지수가 포함된 VAR-Ca와 제조업 전망 BSI가 포함된 VAR-Cb로 나누어진다. VAR-Ca는 EPU 지수, 경제성장률, 콜금리, 소비자 심리지수 순서의 4변수 VAR 모형이고, VAR-Cb는 EPU 지수, 경제성장률, 콜금리, 제조업 전망 BSI 순서의 4변수 VAR 모형이다. VAR-D는 리스크 프리미엄 경로를 분석하기 위한 모형

으로 EPU 지수, 경제성장률, 신용스프레드 순서의 3변수 VAR 모형으로 설계하였다. VAR-E는 대출태도 경로를 분석하기 위한 모형으로 대기업 대출태도지수가 포함된 VAR-Ea와 가계(일반) 대출태도지수가 포함된 VAR-Eb로 나누어진다. VAR-Ea는 EPU 지수, 경제성장률, 콜금리, 대기업 대출태도지수 순서의 4변수 VAR 모형으로 설계하였고, VAR-Eb는, EPU 지수, 경제성장률, 콜금리, 가계(일반) 대출태도지수 순서의 4변수 VAR 모형으로 설계하였다. 모형의 추정 기간은 2008년 1분기~2019년 4분기이며, VAR-C의 경우에는 심리지수 작성 기간으로 인해 2009년 1~2019년 4분기이다. 또한, 모형의 최적 시차는 1~4분기 중 SC를 최소로 하는 시차로 지정한 결과, 모든 모형의 최적 시차가 1분기로 선정되었다. 이에 따라 각 모형의 추정 식은 다음 식 (4-30)~(4-34)와 같이 정의되며, 여기서 *EPU*는 4개의 불확실성 지수(EPU, EPU_small, EPU_BERT, EPU_KOREA) 각각을 의미한다.

(4-30) VAR-Ca

$$EPU_t = \alpha_1 + \sum_{i=1}^{1}\beta_{1i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{1i}GDP_{t-i} + \sum_{i=1}^{1}\delta_{1i}CR_{t-i} + \sum_{i=1}^{1}\zeta_{1i}CCSI_{t-i} + \epsilon_{1t}$$

$$GDP_t = \alpha_2 + \sum_{i=1}^{1}\beta_{2i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{2i}GDP_{t-i} + \sum_{i=1}^{1}\delta_{2i}CR_{t-i} + \sum_{i=1}^{1}\zeta_{2i}CCSI_{t-i} + \epsilon_{2t}$$

$$CR_t = \alpha_3 + \sum_{i=1}^{1}\beta_{3i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{3i}GDP_{t-i} + \sum_{i=1}^{1}\delta_{3i}CR_{t-i} + \sum_{i=1}^{1}\zeta_{3i}CCSI_{t-i} + \epsilon_{3t}$$

$$CCSI_t = \alpha_4 + \sum_{i=1}^{1}\beta_{4i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{4i}GDP_{t-i} + \sum_{i=1}^{1}\delta_{4i}CR_{t-i} + \sum_{i=1}^{1}\zeta_{4i}CCSI_{t-i} + \epsilon_{4t}$$

(4-31) VAR-Cb

$$EPU_t = \alpha_1 + \sum_{i=1}^{1}\beta_{1i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{1i}GDP_{t-i} + \sum_{i=1}^{1}\delta_{1i}CR_{t-i} + \sum_{i=1}^{1}\zeta_{1i}BSI_{t-i} + \epsilon_{1t}$$

$$GDP_t = \alpha_2 + \sum_{i=1}^{1}\beta_{2i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{2i}GDP_{t-i} + \sum_{i=1}^{1}\delta_{2i}CR_{t-i} + \sum_{i=1}^{1}\zeta_{2i}BSI_{t-i} + \epsilon_{2t}$$

$$CR_t = \alpha_3 + \sum_{i=1}^{1}\beta_{3i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{3i}GDP_{t-i} + \sum_{i=1}^{1}\delta_{3i}CR_{t-i} + \sum_{i=1}^{1}\zeta_{3i}BSI_{t-i} + \epsilon_{3t}$$

$$BSI_t = \alpha_4 + \sum_{i=1}^{1}\beta_{4i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{4i}GDP_{t-i} + \sum_{i=1}^{1}\delta_{4i}CR_{t-i} + \sum_{i=1}^{1}\zeta_{4i}BSI_{t-i} + \epsilon_{4t}$$

(4-32) VAR-D

$$EPU_t = \alpha_1 + \sum_{i=1}^{1}\beta_{1i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{1i}GDP_{t-i} + \sum_{i=1}^{1}\delta_{1i}SPREAD_{t-i} + \epsilon_{1t}$$
$$GDP_t = \alpha_2 + \sum_{i=1}^{1}\beta_{2i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{2i}GDP_{t-i} + \sum_{i=1}^{1}\delta_{2i}SPREAD_{t-i} + \epsilon_{2t}$$
$$SPREAD_t = \alpha_3 + \sum_{i=1}^{1}\beta_{3i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{3i}GDP_{t-i} + \sum_{i=1}^{1}\delta_{3i}SPREAD_{t-i} + \epsilon_{3t}$$

(4-33) VAR-Ea

$$EPU_t = \alpha_1 + \sum_{i=1}^{1}\beta_{1i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{1i}GDP_{t-i} + \sum_{i=1}^{1}\delta_{1i}CR_{t-i} + \sum_{i=1}^{1}\zeta_{1i}BLTf_{t-i} + \epsilon_{1t}$$
$$GDP_t = \alpha_2 + \sum_{i=1}^{1}\beta_{2i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{2i}GDP_{t-i} + \sum_{i=1}^{1}\delta_{2i}CR_{t-i} + \sum_{i=1}^{1}\zeta_{2i}BLTf_{t-i} + \epsilon_{2t}$$
$$CR_t = \alpha_3 + \sum_{i=1}^{1}\beta_{3i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{3i}GDP_{t-i} + \sum_{i=1}^{1}\delta_{3i}CR_{t-i} + \sum_{i=1}^{1}\zeta_{3i}BLTf_{t-i} + \epsilon_{3t}$$
$$BLTf_t = \alpha_4 + \sum_{i=1}^{1}\beta_{4i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{4i}GDP_{t-i} + \sum_{i=1}^{1}\delta_{4i}CR_{t-i} + \sum_{i=1}^{1}\zeta_{4i}BLTf_{t-i} + \epsilon_{4t}$$

(4-34) VAR-Eb

$$EPU_t = \alpha_1 + \sum_{i=1}^{1}\beta_{1i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{1i}GDP_{t-i} + \sum_{i=1}^{1}\delta_{1i}CR_{t-i} + \sum_{i=1}^{1}\zeta_{1i}BLTh_{t-i} + \epsilon_{1t}$$
$$GDP_t = \alpha_2 + \sum_{i=1}^{1}\beta_{2i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{2i}GDP_{t-i} + \sum_{i=1}^{1}\delta_{2i}CR_{t-i} + \sum_{i=1}^{1}\zeta_{2i}BLTh_{t-i} + \epsilon_{2t}$$
$$CR_t = \alpha_3 + \sum_{i=1}^{1}\beta_{3i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{3i}GDP_{t-i} + \sum_{i=1}^{1}\delta_{3i}CR_{t-i} + \sum_{i=1}^{1}\zeta_{3i}BLTh_{t-i} + \epsilon_{3t}$$
$$BLTh_t = \alpha_4 + \sum_{i=1}^{1}\beta_{4i}EPU_{t-i} + \sum_{i=1}^{1}\gamma_{4i}GDP_{t-i} + \sum_{i=1}^{1}\delta_{4i}CR_{t-i} + \sum_{i=1}^{1}\zeta_{4i}BLTh_{t-i} + \epsilon_{4t}$$

4) 충격반응분석 결과

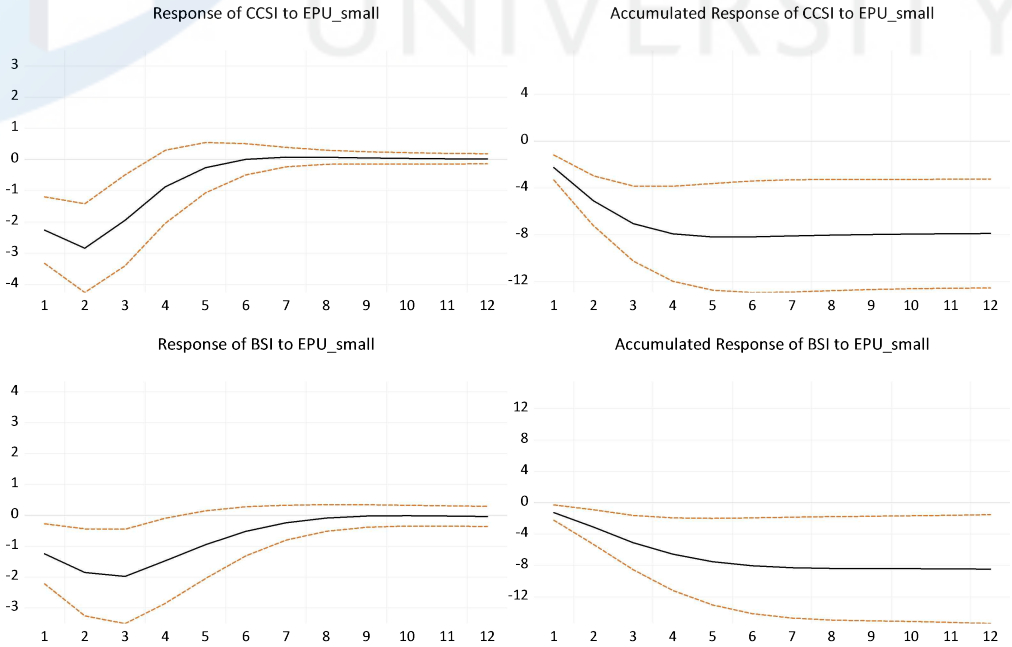
충격반응분석은 Cholesky 분해 방법을 이용해 수행하였으며, 분석 결과는 [그림 4-12]~[그림 4-23]에 걸쳐 제시하였다. 여기서 [그림 4-12]~[그림

4-15]는 VAR-C, [그림 4-16]~[그림 4-19]는 VAR-D, [그림 4-20]~[그림 4-23]은 VAR-E 모형의 분석 결과를 보여주며, 각 그림의 빨간색 점선은 95% 신뢰구간을 의미한다.

(가) 심리 경로

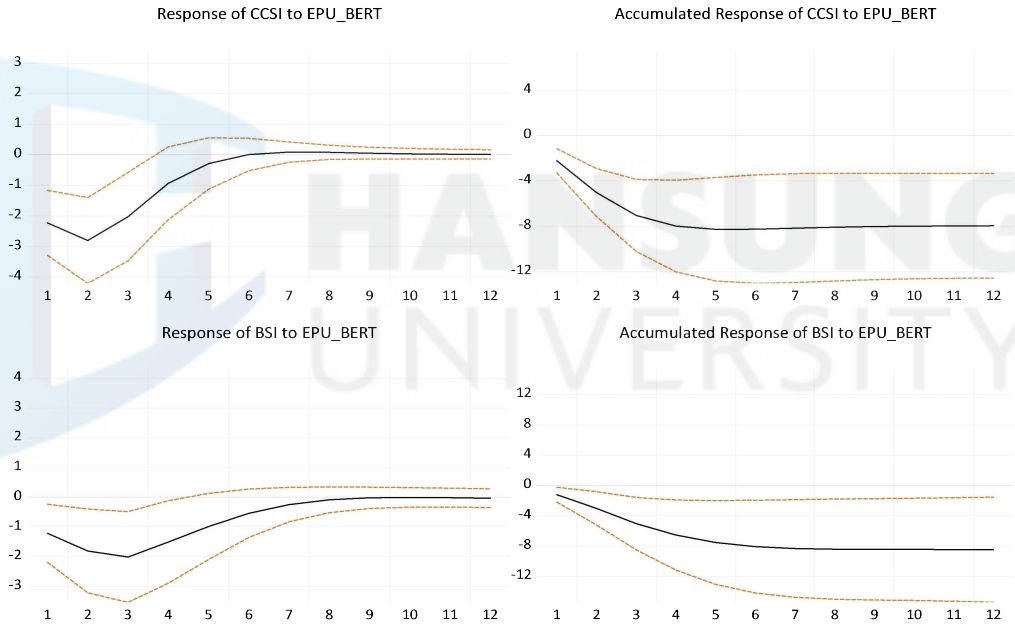
[그림 4-12]는 EPU_small을 이용한 VAR-C 모형의 충격반응분석 결과이다. 분석 결과, 불확실성 확대는 소비 및 투자 관련 심리지표를 유의하게 하락시키는 것으로 나타났다. 구체적으로, 불확실성 충격은 소비심리를 1~3분기 유의하게 하락시켰으며(5.76P 충격에 대한 반응의 크기는 1분기 -2.25P, 2분기 -2.83P, 3분기 -1.94P), 투자심리도 1~4분기까지 유의하게 하락시켰다(5.68P 충격에 대한 반응의 크기는 1분기 -1.24P, 2분기 -1.85P, 3분기 -1.98P, 4분기 -1.47P). 또한, 충격의 반응을 누적하였을 때, 불확실성은 소비심리를 연간 7.91P, 투자심리를 연간 6.56P 하락시키는 것으로 나타났다.

[그림 4-12] EPU_small의 VAR-C 모형 충격반응분석 결과



[그림 4-13]은 EPU_BERT를 이용한 VAR-C 모형의 충격반응분석 결과이다. 분석 결과는 EPU_small과 같다. 구체적으로, 불확실성에 양의 충격이 발생하였을 때, 소비심리는 1~3분기 유의하게 하락하였으며(5.96P 충격에 대한 반응의 크기는 1분기 -2.22P, 2분기 -2.8P, 3분기 -2.02P), 투자심리는 1~4분기 유의하게 하락하였다(5.9P 충격에 대한 반응의 크기는 1분기 -1.21P, 2분기 -1.8P, 3분기 -2.01P, 4분기 -1.5P). 또한 누적충격반응분석에서는 불확실성 충격이 소비심리를 연간 7.97P, 투자심리를 연간 6.54P 하락시키는 것으로 나타났다.

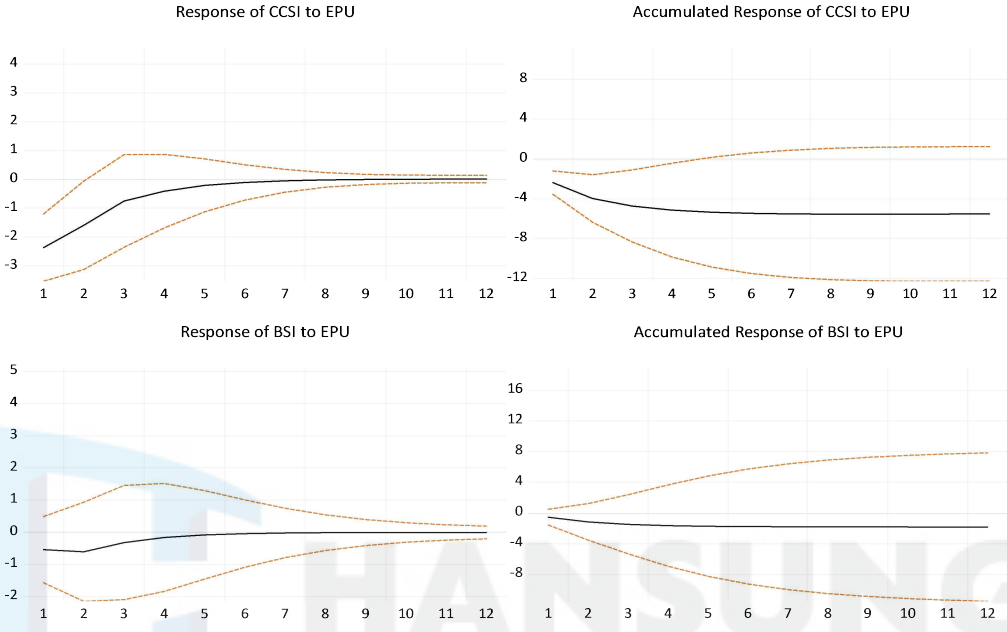
[그림 4-13] EPU_BERT의 VAR-C 모형 충격반응분석 결과



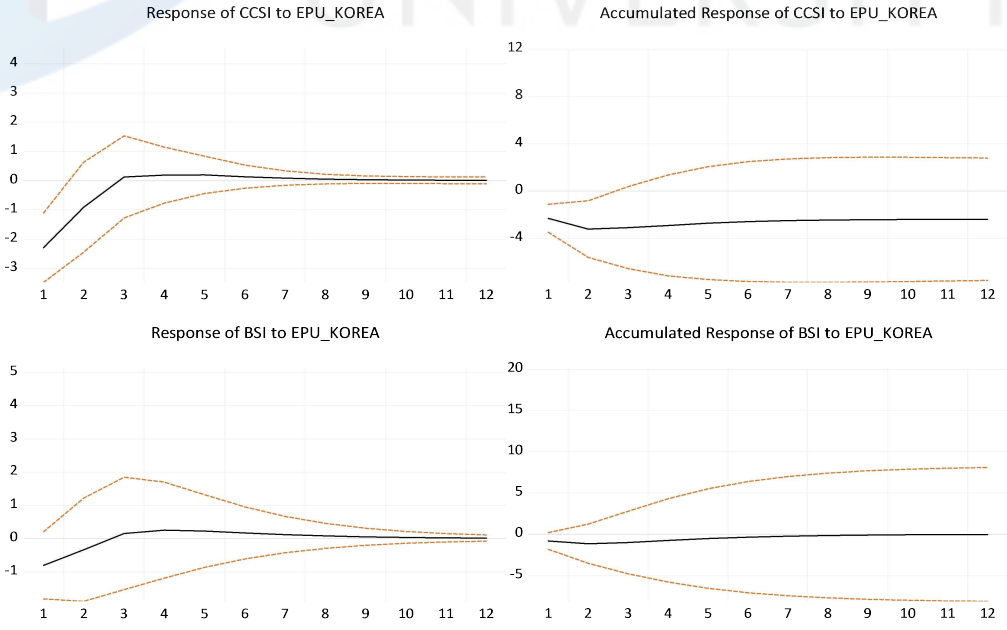
[그림 4-14]와 [그림 4-15]는 EPU와 EPU_KOREA를 이용한 VAR-C 모형의 충격반응분석 결과이다. 두 지수의 분석 결과도 큰 차이는 없었다. 두 지수 모두 불확실성 확대가 소비와 투자 심리지표를 하락시킨다는 분석 결과를 제시하였으며, 투자 심리지표의 경우에는 전체 기간이 통계적으로 유의하지 않았다. 다만, EPU는 소비심리가 1~2분기에 유의하게 하락하였으나(51.7P 충격에 대한 반응의 크기는 1분기 -2.37P, 2분기 -1.59), EPU_KOREA는 소비심리가 1분기에만 유의하게 하락하였다는 차이가 있다(13.2P 충격에 대

한 반응의 크기는 −2.29P).

[그림 4-14] EPU의 VAR-C 모형 충격반응분석 결과



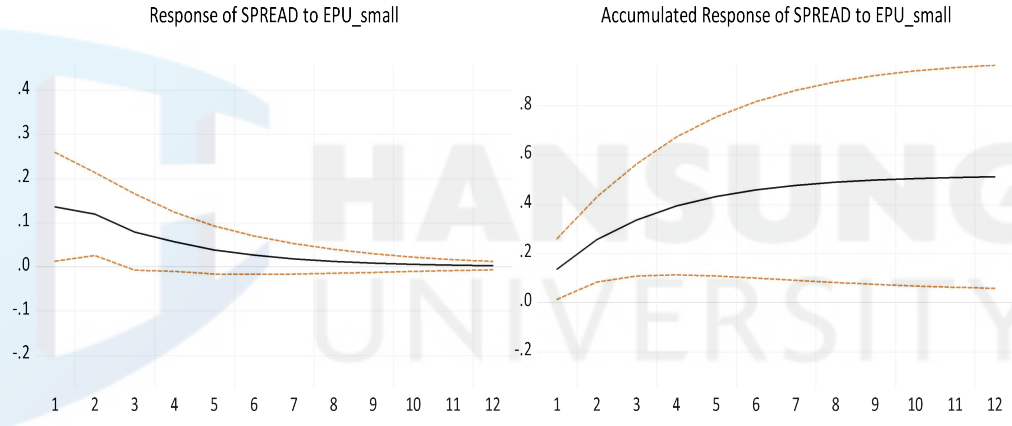
[그림 4-15] EPU_KOREA의 VAR-C 모형 충격반응분석 결과



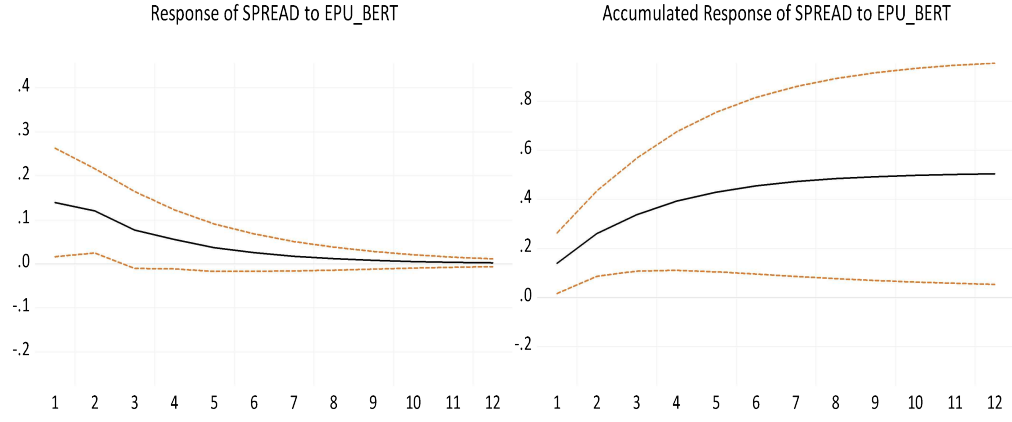
(나) 리스크 프리미엄 경로

[그림 4-16]~[그림 4-19]는 각각 EPU_small, EPU_BERT, EPU, EPU_KOREA를 이용한 VAR-D 모형의 충격반응분석 결과이다. 4개의 EPU 지수는 모두 불확실성 증가가 신용스프레드(리스크 프리미엄)에 양의 영향을 미친다는 분석 결과를 제시하였다. 다만, EPU_small과 EPU_BERT는 1~3분기가 통계적으로 유의했던 반면, EPU_KOREA는 1분기만 통계적으로 유의하였고, EPU는 통계적으로 유의한 결과가 도출되지 않았다.

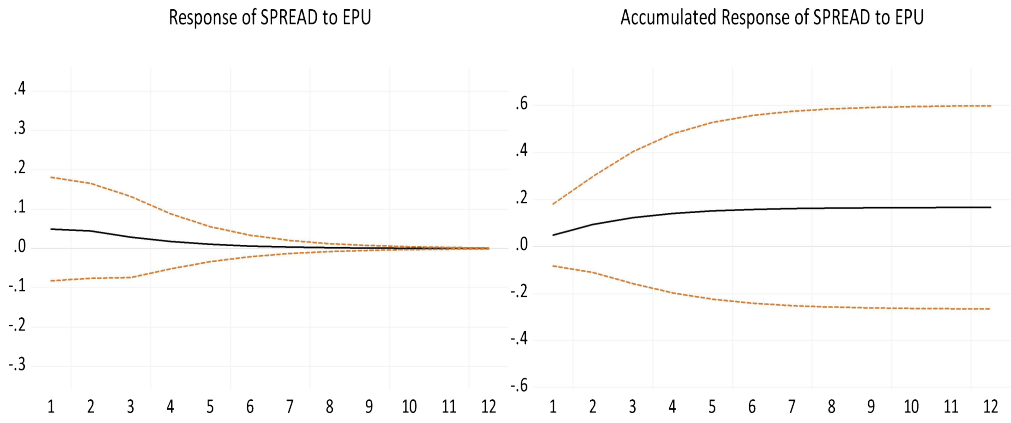
[그림 4-16] EPU_small의 VAR-D 모형 충격반응분석 결과



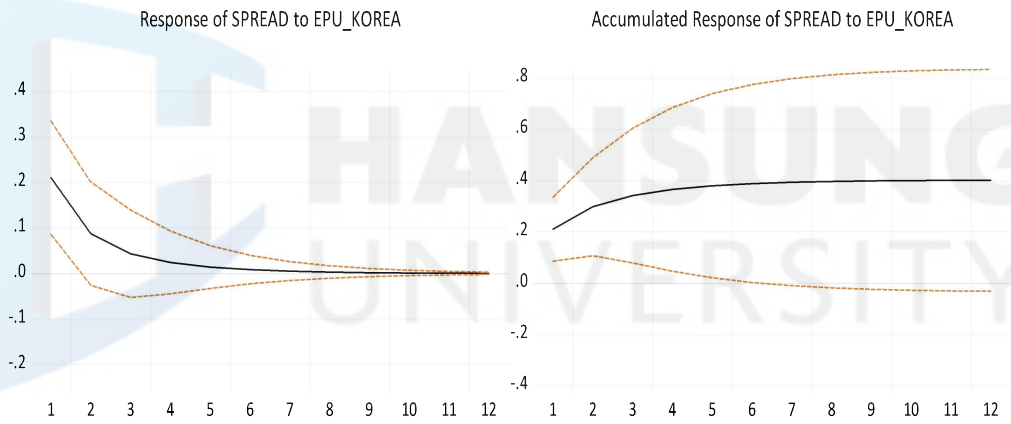
[그림 4-17] EPU_BERT의 VAR-D 모형 충격반응분석 결과



[그림 4-18] EPU의 VAR-D 모형 충격반응분석 결과



[그림 4-19] EPU_KOREA의 VAR-D 모형 충격반응분석 결과

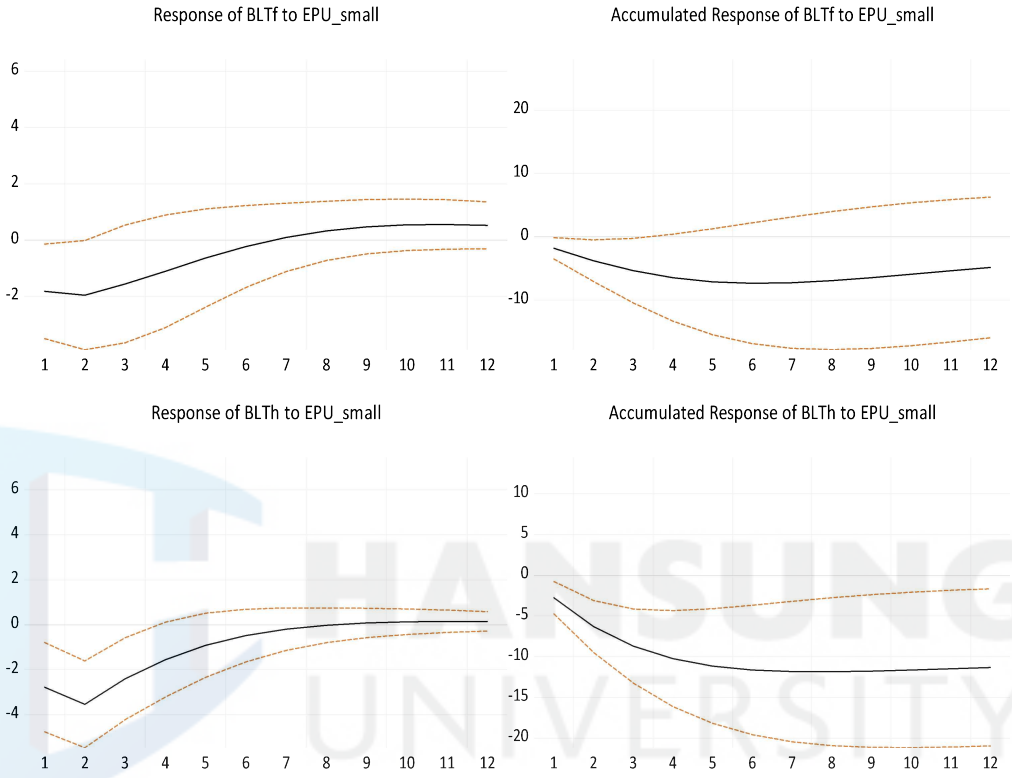


(다) 대출태도 경로

[그림 4-20]은 EPU_small을 이용한 VAR-E 모형의 충격반응분석 결과이다. 분석 결과에서는 불확실성 확대가 금융기관의 대출태도를 엄격하게 만드는 것으로 나타났다. 구체적으로는 불확실성에 양의 충격(6.53P)이 발생하였을 때, 대기업 대출태도지수는 1분기에 1.82P, 2분기에 1.96P 유의하게 하락하였으며, 가계(일반) 대출태도지수는 1분기에 2.77P, 2분기에 3.53P, 3분기에 2.39P 유의하게 하락하였다. 또한 누적충격반응분석 결과, 대기업 대출태도지수는 연간 6.47P 하락하였고, 가계(일반) 대출태도지수는 연간 10.25P 하

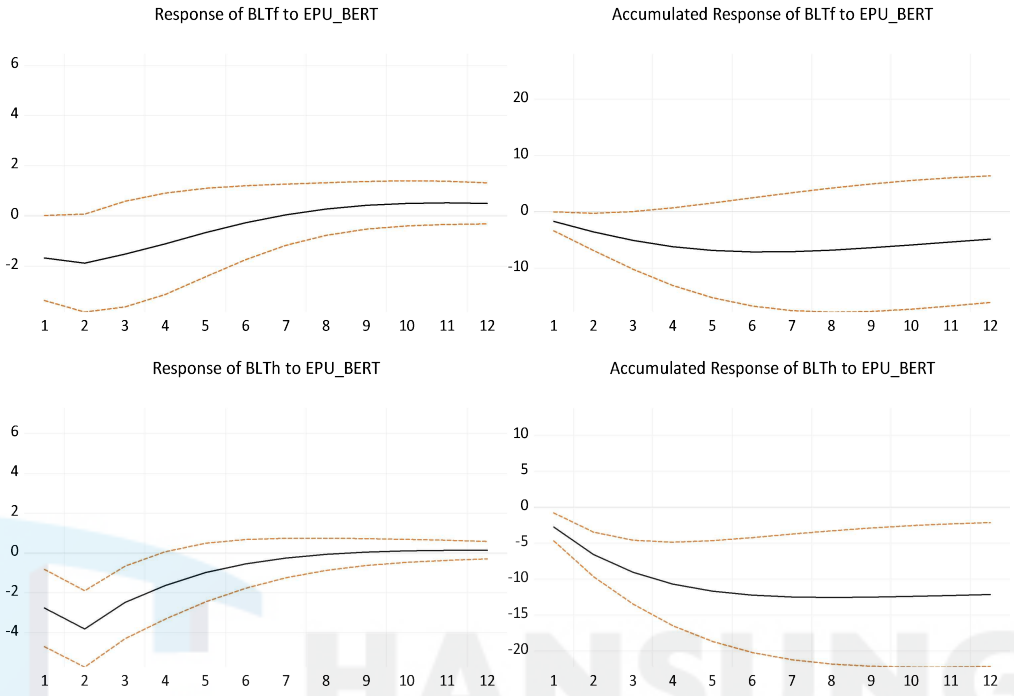
락하였다.

[그림 4-20] EPU_small의 VAR-E 모형 충격반응분석 결과



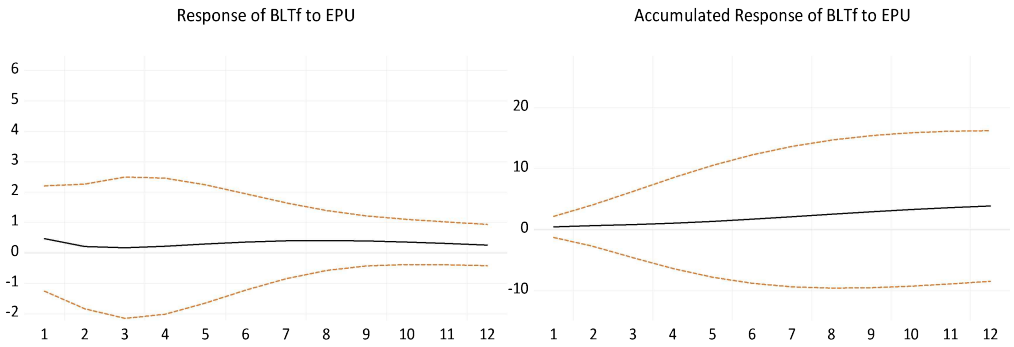
[그림 4-21]은 EPU_BERT를 이용한 VAR-E 모형의 충격반응분석 결과이다. 분석 결과는 EPU_small과 큰 차이가 없었다. 즉, EPU_BERT도 불확실성 확대가 대기업의 대출태도지수와 가계(일반)의 대출태도지수를 유의하게 하락(대출 엄격화)시키는 것으로 분석되었다. 구체적으로는 불확실성에 양의 충격(6.7P)이 발생하였을 때, 대기업 대출태도지수는 1분기에 1.67P 유의하게 하락하였고, 가계(일반) 대출태도지수는 1분기에 2.76P, 2분기에 3.81P, 3분기에 2.48P 유의하게 하락하였다. 또한 누적충격반응분석에서는 대기업 대출태도지수가 연간 6.17P 하락하였고, 가계(일반) 대출태도지수는 연간 10.69P 하락하였다.

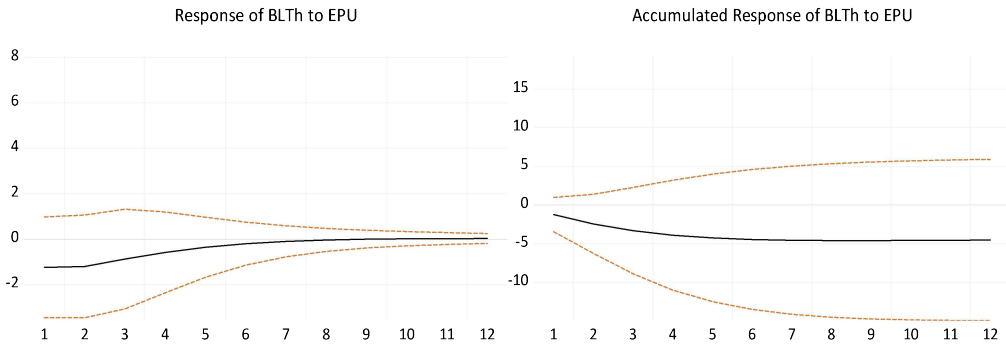
[그림 4-21] EPU_BERT의 VAR-E 모형 충격반응분석 결과



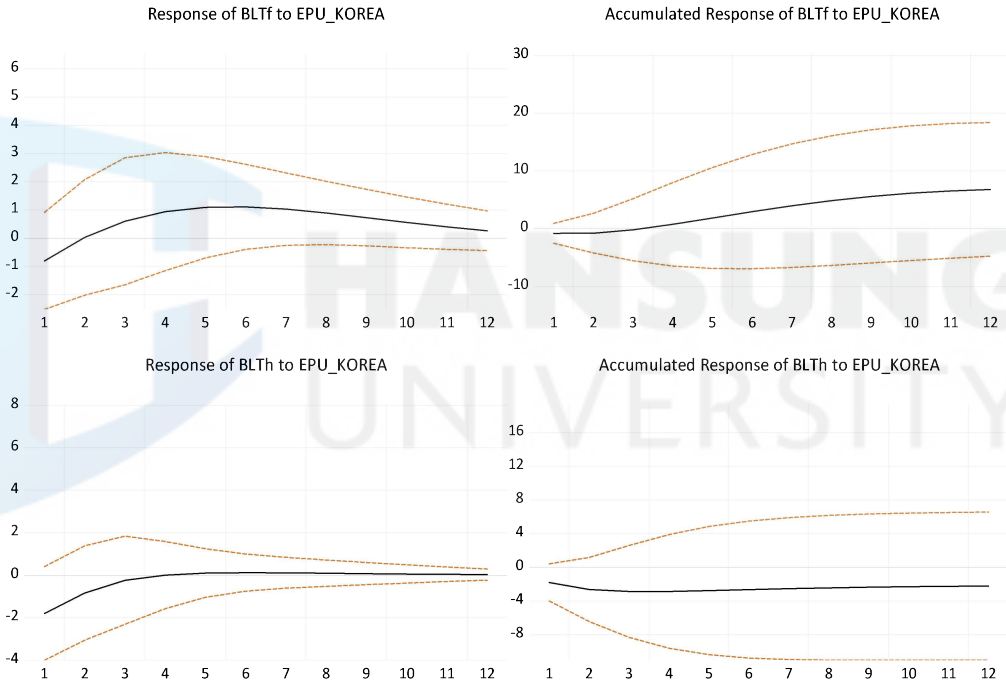
[그림 4-22]와 [그림 4-23]은 EPU와 EPU_KOREA를 이용한 VAR-E 모형의 충격반응분석 결과이다. 두 EPU 지수는 분석 결과가 전체 기간에 걸쳐 통계적으로 유의하지 않았다. 더욱이, EPU는 불확실성 확대가 대기업의 대출태도지수를 상승시킨다는 분석 결과를 도출하였으며, EPU_KOREA는 대기업의 대출태도지수가 1~2분기 하락 이후 크게 상승하여 누적충격반응분석에서는 3분기부터 대출태도지수가 상승하는 모습을 보였다.

[그림 4-22] EPU의 VAR-E 모형 충격반응분석 결과





[그림 4-23] EPU_KOREA의 VAR-E 모형 충격반응분석 결과



4.4 당기예측 모형을 활용한 예측력 평가

경제정책에 핵심적으로 이용되는 GDP 통계는 해당 분기가 끝난 다음 속보치가 3~4주 후에, 잠정치가 7~8주 후에 공표되며, 특히 속보치는 분기 마지막 월 자료를 이용하지 못하고 편제되기 때문에 시의성이 떨어지는 문제가 발생한다. 이로 인해 정책당국 및 주요 정책연구기관들은 월별 경제통계를 활용한 당기예측(nowcasting)[146] 기법이 적극적으로 활용되고 있으나, 월별 경제통계 또한 공표 시차로 인해 모든 자료를 즉각적으로 활용할 수 있는 것은 아니다. 특히, 금융지표를 제외하면 고빈도 경제지표가 거의 전무한 상황이며, GDP와 밀접한 관련이 있는 핵심 지표조차 t-1개월 또는 t-2개월의 자료만 이용할 수 있는 경우가 많아[147] 자료 이용에 제약이 존재한다. 따라서 뉴스기사로 작성되는 본 연구의 EPU 지수는 속보성을 갖는 고빈도 경제지표로서 당기예측에 매우 유용하게 활용될 수 있으며, 특히 불확실성이 GDP 변동과 이론적으로 밀접하게 연관되어 있다는 점을 고려하면 GDP 당기예측 모형의 예측력 향상을 기대해볼 수 있다. 본 절에서는 이를 실증적으로 검증하기 위해 GDP 당기예측 모형을 구축하고, 해당 모형을 통해 본 연구의 EPU 지수가 GDP 당기예측 향상에 도움이 되는지를 살펴본다.

4.4.1 분석 모형: 동태요인모형(Dynamic Factor Model, DFM)

GDP 당기예측 모형은 Bańbura et al.(2010)과 Bańbura et al.(2014) 등의 연구를 준용하여 DFM을 기반으로 구축하였다. 동 모형은 정상성 가정을 만족하는 시계열 데이터 $x_t = (x_{1,t}, x_{2,t}, \cdots, x_{n,t})'$가 있을 때, 다음과 같이 관측방정식(observation equation)과 상태 방정식(state equation)으로 구성된 상태공간 형태(state-space representation)의 모형으로 정의된다.

146) nowcasting은 미래 시점을 예측하는 forecasting과는 달리 현재(now)를 예측한다는 의미이다.

147) 생산지수, 설비투자지수 등의 산업 관련 월별 통계는 t-2 개월의 자료만 이용할 수 있으며, 물가, 노동, 수출입 관련 월별 통계들은 t-1 개월의 자료만 이용할 수 있다(월 중순 기준).

(4-35) 관측식: $x_t = \Lambda f_t + \varepsilon_t$

$$\varepsilon_{i,t} = a_i \varepsilon_{i,t-1} + e_{i,t},\ \ e_{i,t} \sim i.i.d.\ N(0, \sigma_i^2)$$

(4-36) 상태식: $f_t = \sum_{j=1}^{p} A_j f_{t-j} + u_t,\ \ u_t \sim i.i.dN(0, Q)$

여기서 x_t는 $n \times 1$ 크기의 열벡터로 n개의 관측 가능한 경제지표들이며, f_t는 x_t를 r개의 공통요인(common factor)으로 축약한 $r \times 1$ 크기의 열벡터로 관측이 불가능한 잠재변수(latent variable)이다. Λ는 $n \times r$ 크기의 요인적재행렬(factor loading matrix)이며, $\varepsilon_t (= (\varepsilon_{1,t}, \varepsilon_{2,t}, \cdots, \varepsilon_{n,t})')$는 $n \times 1$ 크기의 열벡터로 개별요인(idiosyncratic component)을 의미한다. 즉, 식 (4-35)의 관측방정식은 공통요인과 개별요인의 선형결합으로 이루어지며, 이때 개별요인 ε_t는 AR(1) 과정을 따르고 모든 선·후행 시점에서 f_t와 상관관계가 없다. 또한, 공통요인 f_t는 식 (4-36)과 같이 VAR(p) 과정을 따르며, 여기서 A_j는 $r \times r$ 크기의 자기회귀계수(autoregressive coefficient) 행렬이다.

DFM을 활용한 대다수의 기존 연구들은 전체 경제지표들을 대상으로 공통요인 f_t를 추출하지만, 본 연구에서는 각 지표를 지표 성격에 따라 블록을 지정한 다음, 블록별로 공통요인을 추출한다. 이때 블록은 Bok et al.(2018)과 이현창 외(2022)를 따라 글로벌(global), 실물(real), 연성(soft), 노동(labor)으로 구분하였으며, 이에 따라 x_t, f_t, ε_t와 파라미터 Λ, A_j, Q 등은 다음과 같은 구조를 갖게 된다.[148]

$$x_t = \begin{pmatrix} x_t^G \\ x_t^R \\ x_t^S \\ x_t^L \end{pmatrix},\ f_t = \begin{pmatrix} f_t^G \\ f_t^R \\ f_t^S \\ f_t^L \end{pmatrix},\ \Lambda = \begin{pmatrix} \Lambda_{G,G} & 0 & 0 & 0 & 0 \\ \Lambda_{R,G} & \Lambda_{R,R} & 0 & 0 & 0 \\ \Lambda_{S,G} & 0 & \Lambda_{S,S} & 0 & 0 \\ \Lambda_{L,G} & 0 & 0 & \Lambda_{L,L} & 0 \end{pmatrix},\ \begin{pmatrix} \varepsilon_t^G \\ \varepsilon_t^R \\ \varepsilon_t^S \\ \varepsilon_t^L \end{pmatrix}$$

148) 각 블록에 포함된 경제지표들의 세부 내용은 4.4.2에 제시되어있으며, 각 경제지표는 글로벌 블록에만 포함되거나 혹은 글로벌 블록과 다른 하나의 블록이 동시에 포함된다.

$$A_j = \begin{pmatrix} A_{j,G} & 0 & 0 & 0 \\ 0 & A_{j,R} & 0 & 0 \\ 0 & 0 & A_{j,S} & 0 \\ 0 & 0 & 0 & A_{j,L} \end{pmatrix}, \quad Q = \begin{pmatrix} Q_G & 0 & 0 & 0 \\ 0 & Q_R & 0 & 0 \\ 0 & 0 & Q_S & 0 \\ 0 & 0 & 0 & Q_L \end{pmatrix}$$

(여기서 G, R, S, L는 각각 글로벌, 실물, 연성, 노동, 불확실성 블록을 의미)

식 (4-35)와 (4-36)은 설명의 편의상 월 변수만을 이용한 상태공간 모형을 가정하였으나, 실제 실증분석에서는 월·분기 변수를 통합한 상태공간 모형이 사용된다. 이때, 분기 자료의 월 분해는 Marino and Murasawa(2003)를 기반으로 수행된다. 구체적인 월·분기 변수의 통합과정은 다음과 같다. 먼저, 분기 변수의 값이 해당 분기 3번째 월에 할당(assigned)된다는 관례를 따라 분기 GDP(GDP_t^Q)를 다음 식 (4-37)과 같이 월별 GDP(GDP_t^M)의 합으로 표현한다.[149]

$$(4\text{-}37)\quad GDP_t^Q = GDP_t^M \bullet GDP_{t-1}^M \bullet GDP_{t-2}^M, \quad t = 3, 6, 9$$

$$\Downarrow$$

$$100 \times \log(GDP_t^Q) = 100 \times \log(GDP_t^M) + 100 \times \log(GDP_{t-1}^M) + 100 \times \log(GDP_{t-2}^M), \quad t = 3, 6, 9, \cdots$$

여기서 $100 \times \log(GDP_t^Q) = Y_t^Q$, $100 \times \log(GDP_t^M) = Y_t^M$로 정의하면, GDP의 전분기대비 성장률($y_t^Q = Y_t^Q - Y_{t-3}^Q$)은 다음 식 (4-38)과 같이 전월대비 성장률($y_t^M = \triangle Y_t^M$)로 나타낼 수 있다.

$$(4\text{-}38)\quad y_t^Q = Y_t^Q - Y_{t-3}^Q \approx (Y_t^M + Y_{t-1}^M + Y_{t-2}^M) - (Y_{t-3}^M + Y_{t-4}^M + Y_{t-5}^M)$$
$$= y_t^M + 2y_{t-1}^M + 3y_{t-2}^M + 2y_{t-3}^M + y_{t-4}^M$$

149) Marino and Murasawa(2003)는 월별 변수의 기하 평균(geometric mean)으로 분기 변수를 정의하였지만, Bańbura et al.(2010)과 Bańbura and Modugno(2014)에서는 식 (4-37)과 같이 월별 변수의 곱으로 분기 변수를 정의한다.

다음으로, 관측이 불가능한 전월대비 GDP 성장률은 다음 식 (4-39)와 같은 요인모형을 따른다고 가정한다.

$$(4\text{-}39)\quad y_t^M = \Lambda_Q f_t + \varepsilon_t^Q$$
$$\varepsilon_t^Q = \alpha_Q \varepsilon_{t-1}^Q + e_t^Q,\ \ e_t^Q \sim i.i.d.\, N(0, \sigma_Q^2)$$

그리고 식 (4-39)를 식 (4-38)에 대입하면 다음 식 (4-40)을 얻을 수 있다.

$$(4\text{-}40)\quad y_t^Q = \Lambda_Q f_t + \varepsilon_t^Q + 2(\Lambda_Q f_{t-1} + \varepsilon_{t-1}^Q) + 3(\Lambda_Q f_{t-2} + \varepsilon_{t-2}^Q) + 2(\Lambda_Q f_{t-3} + \varepsilon_{t-3}^Q) + \Lambda_Q f_{t-4} + \varepsilon_{t-4}^Q$$

마지막으로, 위 식을 이용하면 월·분기 변수를 통합한 상태공간 모형을 구축할 수 있다. 예를 들어 상태 방정식의 시차 $p=1$이고, 분기 변수는 GDP 1개라 가정하면, 상태공간 모형은 다음 식 (4-41)과 (4-42)와 같이 구축된다(여기서 $\varepsilon_t = (\varepsilon_{1,t}, \cdots, \varepsilon_{n,t})'$, $e_t = (e_{1,t}, \cdots, e_{n,t})'$이다).[150]

$$(4\text{-}41)\ \text{관측식:}\ \begin{pmatrix} x_t \\ y_t^Q \end{pmatrix} = \begin{pmatrix} \Lambda & 0 & 0 & 0 & 0 & I_n & 0 & 0 & 0 & 0 & 0 \\ \Lambda_Q & 2\Lambda_2 & 3\Lambda_Q & 2\Lambda_2 & \Lambda_Q & 0 & 1 & 2 & 3 & 2 & 1 \end{pmatrix} \begin{pmatrix} f_t \\ f_{t-1} \\ f_{t-2} \\ f_{t-3} \\ f_{t-4} \\ \varepsilon_t \\ \varepsilon_t^Q \\ \varepsilon_{t-1}^Q \\ \varepsilon_{t-2}^Q \\ \varepsilon_{t-3}^Q \\ \varepsilon_{t-4}^Q \end{pmatrix}$$

150) 참고로 여기서는 설명의 편의를 위해 분기 변수가 1개인 경우를 가정하였지만, 실제 실증분석에서는 분기별 변수 n_Q개가 포함된다. 이 경우, $y_t^Q, \varepsilon_t^Q, e_t^Q$는 $n_Q \times 1$ 크기의 벡터가 되고, Λ_Q와 α_Q는 각각 $n_Q \times r$과 $n_Q \times n_Q$ 크기의 행렬이 된다. 또한, 식 (4-41)과 (4-42)에 있는 스칼라 값들은 단위행렬(identity matrix) 또는 영행렬(zero matrix)로 대체된다.

(4-42) 상태식:
$$\begin{pmatrix} f_t \\ f_{t-1} \\ f_{t-2} \\ f_{t-3} \\ f_{t-4} \\ \varepsilon_t \\ \varepsilon_t^Q \\ \varepsilon_{t-1}^Q \\ \varepsilon_{t-2}^Q \\ \varepsilon_{t-3}^Q \\ \varepsilon_{t-4}^Q \end{pmatrix} = \begin{pmatrix} A_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ I_r & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I_r & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_r & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_r & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & diag(\alpha_1, \cdots, \alpha_n) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha_Q & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} f_{t-1} \\ f_{t-2} \\ f_{t-3} \\ f_{t-4} \\ f_{t-5} \\ \varepsilon_{t-1} \\ \varepsilon_{t-1}^Q \\ \varepsilon_{t-2}^Q \\ \varepsilon_{t-3}^Q \\ \varepsilon_{t-4}^Q \\ \varepsilon_{t-5}^Q \end{pmatrix} + \begin{pmatrix} u_t \\ 0 \\ 0 \\ 0 \\ 0 \\ e_t \\ e_t^Q \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

추가로, 블록에 의해 부과된 제약을 반영하면 파라미터 $\Lambda, \Lambda_Q, A_1, Q$는 다음과 같은 구조를 갖는다.[151]

$$\Lambda = \begin{pmatrix} \Lambda_{G,G} & 0 & 0 & 0 \\ \Lambda_{R,G} & \Lambda_{R,R} & 0 & 0 \\ \Lambda_{S,G} & 0 & \Lambda_{S,S} & 0 \\ \Lambda_{L,G} & 0 & 0 & \Lambda_{L,L} \end{pmatrix}, \quad A_1 = \begin{pmatrix} A_{1,G} & 0 & 0 & 0 \\ 0 & A_{1,R} & 0 & 0 \\ 0 & 0 & A_{1,S} & 0 \\ 0 & 0 & 0 & A_{1,L} \end{pmatrix}, \quad Q = \begin{pmatrix} Q_G & 0 & 0 & 0 \\ 0 & Q_R & 0 & 0 \\ 0 & 0 & Q_S & 0 \\ 0 & 0 & 0 & Q_L \end{pmatrix},$$

$$\Lambda_Q = \begin{pmatrix} \Lambda_{Q,G} & \Lambda_{Q,R} & 0 & 0 \end{pmatrix}$$

(여기서 G, R, S, L는 각각 글로벌, 실물, 연성, 노동, 불확실성 블록을 의미)

이에 따라 모형이 추정해야 하는 파라미터 θ는 다음과 같다.

$$\begin{aligned} \theta = (&\mathrm{vec}(\Lambda_{G,G})', \mathrm{vec}(\Lambda_{R,G})', \mathrm{vec}(\Lambda_{S,G})', \mathrm{vec}(\Lambda_{L,G})', \mathrm{vec}(\Lambda_{U,G})', \mathrm{vec}(\Lambda_{R,R})', \\ &\mathrm{vec}(\Lambda_{S,S})', \mathrm{vec}(\Lambda_{L,L})', \mathrm{vec}(\Lambda_{U,U})', \Lambda_{Q,G}, \Lambda_{Q,R}, A_{1,G}, A_{1,R}, A_{1,S}, A_{1,L}, \\ &A_{1,U}, Q_G, Q_R, Q_S, Q_L, Q_U, \alpha_1, \cdots, \alpha_n, \alpha_Q, \sigma_1, \cdots, \sigma_n, \sigma_Q)' \end{aligned}$$

파라미터 θ의 추정치가 주어지면 DFM에 칼만 필터(Kalman filter)와 스무딩(smoothing)을 적용하여 공통요인의 추정치와 예측치를 구할 수 있다. 이때, θ를 추정하는 방법으로 본 연구는 EM 알고리즘 기반의 최대우도추정법을 사용한다.[152] EM 알고리즘은 우도(likelihood) 함수를 관측 가능한 변수와

151) 본 연구에서 분기 지표는 모두 실질 블록에 속하며, 세부 내용은 4.4.2에 제시되어 있다.

잠재변수로 표현한 다음, E-step(Expectation-step)과 M-step(Maximization-step)을 반복(iteration)해 최대우도 추정치를 계산한다. 예를 들어 식 (4-35)와 (4-36)과 같은 상태공간모형이 주어지고, $X=(x_1, \cdots, x_t,)$, $F=(f_1, \cdots, f_t)$, $t=1, \cdots, T$의 결합 로그 우도를 $l(X, F;\theta)$라 표현하면, EM 알고리즘은 아래의 두 단계를 반복해 우도의 지역 극대점(local maximum)으로 수렴한다.[153]

• E-step: 이전 반복에서 추정된 파라미터 $\theta(j)$를 사용하여 조건부 로그 우도의 기댓값을 계산한다.

$$(4-43)\quad L(\theta, \theta(j)) = \mathbb{E}_{\theta(j)}\left[l(X, F;\theta)\,|\,\Omega_T\right]$$

• M-step: 이전 반복에서 계산된 $L(\theta, \theta(j))$를 극대화하여 파라미터를 다시 추정한다.

$$(4-44)\quad \theta(j+1) = \arg\max_{\theta} L(\theta, \theta(j))$$

이때 초기 파라미터 $\theta(0)$는 Doz et al.(2006)의 방법론(two-step DFM)을 따라 주성분 분석을 기반으로 얻어지며, 수집된 정보집합 Ω_T는 x_t에 결측치가 포함되므로 $\Omega_T \subset X$이다. 또한, E-step의 조건부 기댓값은 칼만 스무딩을 통해 계산되며, M-step은 공통요인이 잠재변수이므로 닫힌 형식(closed form)으로 이루어진다.[154] 그리고 위 과정을 통해 θ의 추정치가 도출되면 공통요인의 추정치와 예측치를 구하고, 이를 이용하여 GDP 성장률을 포함한 모형 내 모든 변수에 대한 예측치를 계산한다.

---

152) EM 알고리즘은 데이터에 결측치가 있는 경우에도 효율적으로 파라미터를 추정할 수 있다는 장점이 있다. 특히, Bańbura and Modugno(2014)는 결측치가 임의의 패턴을 보이는 상황에서도 EM 알고리즘을 적용할 수 있도록 확장하였으며, 최근 Bok et al.(2018) 등에서 당기예측 전망모형 추정에 적용된 이후 각국 중앙은행에서도 활용하는 사례가 늘고 있다(이현창 외, 2022).

153) EM 알고리즘은 특정 정규성 조건(regularity condition)에서 우도의 지역 극대점으로의 수렴을 보장하지만, 전역 극대점(global maximum)으로의 수렴은 보장하지 않는다.

154) EM 알고리즘의 더 상세한 계산 절차와 수학적 유도 과정은 Bańbura et al.(2010)과 Bańbura and Modugno(2014)를 참조하라.

4.4.2 이용 자료 및 모형 추정

모형 입력에 사용된 경제변수는 EPU 지수 평가의 객관성을 확보하기 위해 이현창 외(2022)와 동일한 구성을 사용하였다. 해당 변수들은 [표 4-10]에 제시된 바와 같이, 분기 변수 5개와 월 변수 30개로 구성되며, 각 변수는 Global, Real, Labor, Soft의 4개의 블록으로 분류된다. Global 블록은 모든 변수를 포괄하는 블록으로, 각각의 변수는 Global 블록에만 포함되거나 혹은 Global 블록과 다른 블록에 중복으로 포함된다. Real 블록과 Labor 블록은 실물 변수와 노동 변수가 포함되는 블록이며, Soft 블록은 경제주체들의 인식과 심리를 나타내는 연성자료(soft data)[155]들이 포함되는 블록이다. 예측 대상이 되는 GDP 성장률과 분기 변수들은 국민계정의 전분기대비 자료가 사용되며, 이 외에 나머지 변수들은 단위근 검정을 통해 비정상성을 갖는 경우 차분 또는 증가율로 변환된다. 또한, Real 블록과 Labor 블록에 속한 변수들은 통계기관에서 발표하는 계절조정계열 자료가 사용되며, 계절조정계열이 공표되지 않는 변수들은 X12-Arima를 통해 계절조정계열로 변환된다.

[표 4-10] 당기예측모형 변수 목록(이현창 외, 2022)

구분	변수명	블록 설정				변환 방법[1)]
		Global	Real	Labor	Soft	
생산 (12)	GDP(분기)	✓	✓			0
	광공업 생산지수	✓	✓			2
	서비스업 생산지수	✓	✓			1
	제조업 출하지수	✓	✓			2
	제조업 재고지수	✓	✓			2
	전산업 업황 BSI	✓			✓	0
	전산업 매출 BSI	✓			✓	0
	제조업 업황 BSI	✓			✓	0
	제조업 수출 BSI	✓			✓	0
	제조업 내수판매 BSI	✓			✓	0
	제조업 신규수주 BSI	✓			✓	0
	제조업 가동률 BSI	✓			✓	0

155) GDP 당기예측에 사용되는 자료는 크게 경성자료(hard data)와 연성자료로 구분된다. 경성자료는 GDP의 구성 요소이거나 GDP와 밀접한 관련이 있는 자료를 의미하며, 연성자료는 GDP나 실물경제에 간접적인 정보를 제공하는 자료를 의미한다.

소비·물가 (8)	민간소비(분기)	✓	✓			0
	소매판매액지수	✓	✓			2
	소비자심리지수	✓			✓	0
	현재경기판단 CSI	✓			✓	0
	소비자물가지수	✓				2
	소비자물가지수 (농산물 및 석유류 제외)	✓				2
	소비자물가지수 (식료품 및 에너지 제외)	✓				2
	생산자물가지수	✓				2
투자 (3)	설비투자(분기)	✓	✓			0
	설비투자지수	✓	✓			1
	건설투자(분기)	✓	✓			0
대외 (5)	수출(분기)	✓	✓			0
	수출	✓	✓			2
	수입	✓	✓			2
	수출물가지수	✓				2
	수입물가지수	✓				2
노동 (5)	총취업자수	✓		✓		2
	고용률	✓		✓		1
	실업률	✓		✓		1
	구인배수	✓		✓		1
	취업률	✓		✓		1
심리 (2)	경제심리지수	✓			✓	0
	EPU 지수 (EPU, EPU_KROREA, EPU_small, EPU_BERT)	✓			✓	0

주: 1) 0은 원자료, 1은 차분, 2는 로그 차분을 의미

블록별 공통요인은 예측오차를 비교하여 Global 블록은 3개, Real, Labor, Soft 블록은 1개를 추출하였으며, 적정 시차는 SC를 기준으로 Global 블록에서 추출된 공통요인은 2기를, Real, Labor, Soft 블록에서 추출된 공통요인은 1기를 적용하였다. 그 결과, Global 블록에서 추출된 3개의 공통요인은 아래 [표 4-11]과 같이 생산, 수출입, 소비와 밀접한 관련이 있는 것으로 나타났으며, 이 중에서 2개의 공통요인이 본 연구에서 작성한 EPU 지수와 높은 적합도를 나타냈다.

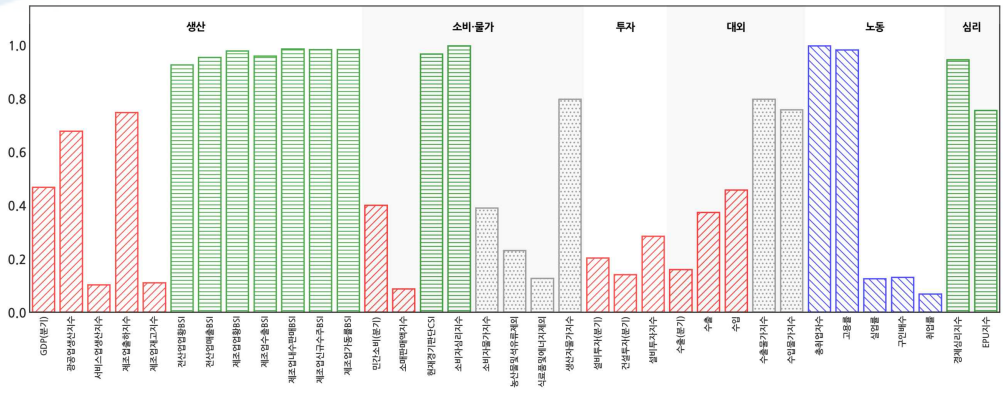
[표 4-11] 개별 공통요인과 각 변수의 적합도[1)]

Global1 요인의 주요 정보변수	R^2	Global2 요인의 주요 정보변수	R^2	Global3 요인의 주요 정보변수	R^2
제조업업황BSI	0.92	수출물가지수	0.41	현재경기판단CSI	0.48
전산업업황BSI	0.90	수입물가지수	0.34	소비자심리지수	0.46
제조업가동률BSI	0.88	EPU지수	0.32	생산자물가지수	0.33
제조업신규수주BSI	0.88	제조업수출BSI	0.20	EPU지수	0.29
경제심리지수	0.87	전산업매출BSI	0.18	소비자물가지수	0.24

주: 1) 개별 공통요인을 독립변수로 하여 각 변수를 적합한 경우의 적합도(coefficient of determination, R^2)

또한 전체 공통요인을 독립변수로 하여 각 변수의 적합도를 확인한 결과([그림 4-24]), Real 블록과 Labor 블록에 속한 일부 변수들(서비스업생산지수, 제조업재고지수, 소매판매액지수, 실업률, 구인배수, 취업률 등)은 상대적으로 낮은 적합도를 나타냈지만, Soft 블록에 속한 변수들은 모두 높은 적합도를 나타냈다. 특히, 본 연구에서 작성한 EPU 지수의 적합도는 약 0.76으로 나타나, 당기예측 모형에서 유용한 정보변수로 활용될 수 있는 가능성을 시사하였다.

[그림 4-24] 전체 공통요인과 각 변수의 적합도[1)2)]



주: 1) 전체 공통요인을 독립변수로 하여 각 변수를 적합한 경우의 적합도(coefficient of determination, R^2)
2) 빨강(／／), 파랑(＼＼), 초록(－－) 색상의 막대는 각각 real, labor, soft 블록의 속한 변수를 나타냄

[표 4-12]는 EPU 지수(EPU_small) 반영에 따른 예측력 향상 효과를 확인하기 위해 표본내 예측(in-sample forecasting)을 수행한 결과이다. 예측오차는 RMSE(Root Mean Squared Error)와 MAE(Mean Absolute Error)를 기준으로 측정하였으며,[156] 추정 기간은 코로나19 시기에 지표의 변동성이 확대되었던 점을 고려하여 코로나19 기간을 포함한 경우(2008년~2022년)와 포함하지 않은 경우(2008년~2019년)로 구분하였다. 평가 결과에서는 EPU 지수(EPU_small)를 포함한 DFM_EPU가 EPU 지수(EPU_small)를 포함하지 않은 DFM_BASE보다 전반적으로 낮은 예측오차를 기록하였다. 구체적으로 코로나19 기간을 포함하지 않았을 때, DFM_EPU는 0.442(RMSE)와 0.351(MAE)의 수치를 기록하였으며, DFM_BASE는 0.453(RMSE)과 0.362(MAE)의 수치를 기록하였다. 또한, 전체 기간을 대상으로 한 분석에서도 DFM_EPU는 0.521(RMSE)과 0.395(MAE)의 수치를, DFM_BASE는 0.533(RMSE)과 0.408(MAE)의 수치를 기록하여 EPU 지수가 당기예측 모형의 유용한 정보변수로 활용될 수 있음을 시사하였다.

[표 4-12] 표본내 예측 결과

모형[1)]	추정 기간	RMSE	MAE
DFM_BASE	2008년~2019년	0.453	0.362
	2008년~2022년	0.533	0.408
DFM_EPU	2008년~2019년	0.442	0.351
	2008년~2022년	0.521	0.395

주: 1) DFM_BASE는 EPU 지수(EPU_small)가 포함되지 않은 모형이며, DFM_EPU는 EPU 지수(EPU_small)가 포함된 모형임

4.4.3 예측력 평가

156) 실제 GDP 성장률을 Y, GDP 성장률의 예측치를 $\hat{Y}$라 했을 때, RMSE와 MAE는 다음과 같이 계산된다.

$$RMSE = \sqrt{\frac{1}{n}\sum_{n=1}^{n}(Y_t - \hat{Y}_t)^2}, \quad MAE = \frac{1}{n}\sum_{n=1}^{n}\left|Y_t - \hat{Y}_t\right|$$

본 항에서는 예측력 평가를 위해 표본외 예측(out-of-sample forecasting)의 수행 결과를 제시한다.[157] 예측 기간은 2017년 1분기~2022년 4분기로 전체 24개 분기이며(코로나 발생 전·후 3년), 예측오차는 표본내 예측과 마찬가지로 RMSE와 MAE로 계산하였다. 또한, 모형의 예측 시나리오는 예측 수행 시점에 따라 다양한 가정이 가능하지만, 본 연구에서는 예측 시계(forecasting horizon)를 당분기로 한정하여 분기 말 하순에 분석이 수행된다고 가정하였다. 예를 들어 2018년 1분기의 GDP 예측치는 2018년 3월(분기 말) 28일(하순)에 분석이 수행된 뒤 31일에 공표되는 것으로 가정하며,[158] 이에 따라 공표 시차를 고려한 월 변수들의 자료 가용 시점은 다음과 같이 정리된다.

- **t 월**: 전산업 업황 BSI, 전산업 매출 BSI, 제조업 업황 BSI, 제조업 수출 BSI, 제조업 내수판매 BSI, 제조업 신규수주 BSI, 제조업 가동률 BSI, 소비자심리지수, 현재경기판단 CSI, 경제심리지수, EPU 지수

- **t-1 월**: 소비자물가지수, 소비자물가지수(농산물 및 석유류 제외), 소비자물가지수(식료품 및 에너지 제외), 생산자물가지수, 설비투자지수, 수출물가지수, 수입물가지수, 수출, 수입, 총취업자수, 고용률, 실업률, 구인배수, 취업률

- **t-2 월**: 광공업 생산지수, 서비스업 생산지수, 제조업 출하지수, 제조업 재고지수, 소매판매액 지수, 설비투자지수

한편, 표본외 예측을 수행하는 방식으로는 축차 예측(recursive forecast)과 롤링 예측(rolling forecast)이 있다. 축차 예측은 예측 시점이 진행될수록 과거의 모든 데이터를 누적하여 예측을 수행하는 방식이고, 롤링 예측은 고정된 크기의 학습 구간(window)을 일정 간격으로 이동시키며 예측을 수행하는 방식이다. 본 연구에서는 EPU 지수의 작성 기간으로 인해 모형의 추정 기간이 짧은 점을 고려하여 축차 예측 방식으로 표본외 예측을 수행하였다. 또한, 실

157) 앞서 수행한 표본내 예측은 모든 자료를 가용할 수 있다고 가정하지만, 실제 당기예측 수행 과정에서는 예측 수행 시점에 따라 자료의 가용여건이 다르기 때문에 이를 반영할 수 있는 표본외 예측이 당기예측 모형의 유용성을 판단하는 데 더 널리 이용되고 있다.

158) 이때 28일은 서베이 자료들의 공표 시점과 모형의 추정 시간 및 공표 시점 등을 고려한 날짜이다.

제 두 방식의 예측 성능을 비교한 결과에서도 축차 예측이 롤링 예측보다 더 낮은 예측오차를 기록하는 것을 확인하였다.

표본외 예측의 수행 결과는 [표 4-13]에 제시되어있다. 분석 결과, EPU 지수를 반영하지 않은 기준 모형(baseline)은 RMSE 0.9107, MAE 0.6346을 기록하였으며, EPU 지수가 반영된 4개의 모형은 모두 RMSE 기준에서 예측 성능의 개선을 보였다. 특히, EPU_small이 반영된 모형은 RMSE 0.8914, MAE 0.6273을 기록하여 두 지표 모두에서 가장 우수한 예측력을 나타냈으며, EPU_BERT가 반영된 모형 또한 RMSE 0.8925, MAE 0.6294를 기록하여 이와 매우 유사한 예측력을 나타냈다. 반면, EPU와 EPU_KOREA가 반영된 모형은 RMSE를 기준으로는 Baseline보다 소폭 낮은 값을 기록하였으나, MAE를 기준으로는 오히려 예측오차가 증가하였다. 이러한 결과는 본 연구에서 작성한 EPU 지수가 기존 EPU 지수들보다 높은 현실설명력을 제공하며, 당기예측 모형의 유용한 정보변수로 활용될 수 있음을 시사한다.

[표 4-13] 표본외 예측 결과

사용된 EPU 지수[1)]	RMSE	MAE
-	0.9107	0.6346
EPU	0.9060	0.6358
EPU_KOREA	0.9054	0.6514
EPU_small	0.8914	0.6273
EPU_BERT	0.8925	0.6294

주: 1) EPU와 EPU_KOREA는 각각 Baker et al.(2016)과 Cho and Kim(2023)이 작성한 EPU 지수, EPU_small과 EPU_ BERT는 본 연구 모형과 비교 대상 모형으로 작성한 EPU 지수를 의미

Ⅴ. 결 론

5.1 연구 결과 요약

본 연구는 기존 EPU 지수가 지닌 방법론적 한계를 보완하기 위해 기계학습 기반의 감성 분석을 활용한 새로운 EPU 지수를 제안하였다. 아울러 감정 분류 모형을 구축하는 과정에서 직면할 수 있는 현실적 제약을 완화하기 위해 소규모 학습 데이터와 전이 학습을 활용한 감정 분류 모형을 제안하였으며, 특히 PLM의 높은 컴퓨팅 비용과 추론 지연 문제를 고려하여 단어 임베딩 모형을 전이 학습에 활용하여도 우수한 성능과 효율성을 확보할 수 있는 실용적인 모형을 구현하는데 주안점을 두었다. 이 과정에서 본 연구는 경제정책 도메인에 특화된 사용자 사전을 구축하였고, 형태소 분석을 통해 얻은 품사 정보를 단어 임베딩 모형에 반영하였다. 또한, 본 연구에 적합한 감정 분류 모형과 최적의 성능을 도출하기 위해 텍스트 전처리 도구, 단어 임베딩 기법, 딥러닝 아키텍처들을 체계적으로 비교·평가하였다. 나아가 본 연구의 EPU 지수가 기존 EPU 지수들보다 높은 현실설명력을 제공하는지를 검증하기 위해 다양한 유용성 평가 과정을 수행하는 한편, 본 연구 모형의 실용성을 확인하고자 BERT 계열의 PLM을 활용하여 감정 분류 모형을 구축하고, 이를 통해 작성된 EPU 지수를 비교지표로 활용하였다. 이러한 과정을 통해 도출된 본 연구의 주요 결과는 다음과 같다.

Ⅱ장에서는 경제정책 뉴스기사의 수집 결과와 텍스트 전처리 도구들의 평가 결과, 그리고 학습 데이터셋의 구축 결과를 제시하였다. 경제정책 뉴스기사 수집은 단어군 선정, 기사 제목 수집, 중복 및 예외 기사 제거, 기사 본문 수집으로 구분하여 수행하였다. 단어군은 이긍희 외(2020)의 단어군을 활용하여 경제 단어군 4개와 정책 단어군 45개로 구성하였으며, 언론사의 경우에는 이긍희 외(2020)보다 9개를 더 추가한 19개로 구성하였다. 기사 제목은 해당 단어군을 빅카인즈에 검색하여 2,003, 684건이 수집되었고, 중복 및 예외 기사를 제거한 결과, 1,799,000건의 기사 제목이 유효표본으로 남았다. 기사 본

문은 URL 인코딩과 웹 스크래핑을 활용하여 1,799,000건의 기사 제목을 네이버에 검색하여 수집하였다. 그 결과, 1,158,899건의 뉴스기사가 수집되었으며, 2005년~2007년의 뉴스기사는 일 평균 20~22건밖에 수집되지 않아 EPU 지수의 작성 기간을 2008년부터로 한정하였다.

뉴스기사를 수집한 후에는 사전 정제, 말뭉치 정의, 문장 토큰화, 사용자 사전 구축, 형태소 분석의 5단계로 구분하여 텍스트 전처리 과정을 수행하였다. 사전 정제 단계에서는 불필요한 문자열 제거, 특수문자 변환, 소괄호 내의 단어 제거 등의 작업이 수행되었고, 말뭉치는 문장으로 정의하였다. 또한, 사전 정제 과정을 거치고 난 후에는 중복 및 예외 기사들을 한 번 더 제거하였으며, 그 결과 1,158,899건의 기사 중 1,152,529건의 기사가 분석 대상으로 남았다. 문장 토큰화 단계에서는 본 연구에 적합한 문장 분리기를 선정하기 위해 NLTK와 KSS, 그리고 문장 분리 기능을 제공하는 6개의 형태소 분석기(Kiwi, OKT, 한나눔, 코모란, 은전한잎, 꼬꼬마)를 대상으로 문장 분리 성능을 평가하였다. 평가는 처리 속도, 정답률, Dice Coefficint를 기준으로 수행하였으며, 평가 결과에서는 문장 분리 정확도와 처리 속도 모두에서 우수한 성능을 보인 NLTK가 본 연구의 문장 분리기로 채택되었다. 다만, NLTK는 문장의 계층적 구조를 분절할 수 있는 기능이 제공되지 않아 NLTK로 문장 분리를 수행한 후에는 인용문을 추출하는 전처리 작업을 추가로 수행하였으며, 그 결과 1,152,529건의 뉴스기사는 약 1,500만 개의 문장으로 분리되었다. 사용자 사전 구축 단계에서는 soynlp의 명사 추출기를 활용하여 문장 분리가 완료된 약 1,500만 개의 문장을 대상으로 학습을 수행하였다. 그 결과, 약 200만 개의 명사가 학습되었으며, 이중 등장 빈도가 10회 이상이고, 3음절 이상인 명사 251,156개를 1차로 선별한 뒤, 형태소 분석기에 등록된 125,432개의 명사를 제거하였다. 이후 남은 125,724개의 명사에 대해 최종 검수 작업을 거쳐, 일반명사 71,126개와 고유명사 16,787개로 구성된 사용자 사전을 구축하였다. 형태소 분석 단계에서는 본 연구에 적합한 분석기를 선정하기 위해 Kiwi, OKT, 한나눔, 코모란, 은전한잎, 꼬꼬마를 대상으로 성능 평가를 수행하였다. 평가는 품사 목록 비교, 모호성 평가, 처리 속도 비교, 사용자 사전 활용 가능성 검토를 기준으로 수행하였으며, 평가 결과에서는 처리

속도 및 사용자 사전 활용 가능성 측면에서는 은전한잎이, 분석 품질 측면에서는 Kiwi의 성능이 우수한 것으로 나타났다. 본 연구는 Kiwi와 은전한잎의 속도 차이가 그리 크지 않았다는 점을 고려하여 Kiwi를 본 연구의 형태소 분석기로 선정하였다. 다만, Kiwi는 다어절 단어에 대한 사용자 사전 적용에 한계가 있어 이를 보완하기 위해 띄어쓰기 정규화 과정을 추가로 수행하였다.

학습 데이터셋은 1,500만 개의 문장에서 약 2만 개를 추출하여 80%를 학습 데이터로 나머지 20%를 검증 데이터로 구성하였다. 데이터셋이 부족한 점을 고려하여 테스트 데이터는 구축하지 않는 대신 계층적 k-fold CV를 모형 평가에 활용하여 모형의 안정성과 신뢰성을 높였다. 또한, 감정 레이블을 작성한 결과, 23%는 긍정, 33%는 부정, 44%는 부정으로 분류되었으며, 모형 학습 시 레이블 가중치를 반영하여 불균형 데이터를 조정하였다.

Ⅲ장에서는 단어 임베딩 모형, 감정 분류 모형, 비교 대상 모형의 구축 결과를 제시하였다. 단어 임베딩 모형은 Word2Vec, FastText, GloVe의 세 가지 방법론을 비교하여 구축하였다. 학습에 사용된 말뭉치 데이터는 품사 정보가 부착된 약 1,500만 개의 문장 말뭉치이며, 해당 데이터의 단어 집합의 크기는 529,589개이다. 각 임베딩 모형을 구축한 후에는 성능 평가를 통해 본 연구에 적합한 모형을 선정하였다. 평가는 단어 유추 평가, 단어 유사도 평가, 외재적 평가로 나누어 수행하였으며, 외재적 평가에서는 감정 분류 태스크에서의 성능 향상을 측정하였다. 평가 결과에서는 Word2Vec가 세 가지 평가 모두에서 가장 우수한 성능을 나타내, 본 연구의 최적 모형으로 선정되었다. 다만, FastText는 Word2Vec와의 성능 차이는 매우 근소하게 나타났으며, 외재적 평가에서는 FastText가 경제정책 뉴스기사에 등장한 OOV 단어에 대해서도 n-gram 벡터를 활용해 적절한 단어 벡터를 생성할 수 있음이 확인되었다. 따라서 FastText도 본 연구의 좋은 선택지가 될 수 있으며, 특히 향후 더 많은 뉴스기사가 수집되어 신조어와 같은 OOV 단어의 출현 빈도가 증가한다면 FastText의 활용 여부를 적극적으로 검토할 필요가 있다고 판단된다.

감정 분류 모형은 RNN, CNN, 트랜스포머의 세 가지 딥러닝 아키텍처를 기반으로 구축하였으며, 다양한 신경망 구조와 하이퍼파라미터 설정을 검토하여 최적의 성능을 도출하였다. 먼저 RNN은 LSTM, BiLSTM, GRU, BiGRU

네 가지 구조에 어텐션 메커니즘을 적용하였으며, 각 모형은 RNN층, 어텐션층, 완전연결층이 1개씩 설계되었다. 하이퍼파라미터는 임베딩 벡터의 차원은 300, 은닉 상태의 크기는 128, 완전연결층의 뉴런 수는 64로 설정되었고, 학습 설정은 배치 크기 64, 에폭 30, 그리고 최적화 알고리즘은 RAdam이 사용되었다. 또한, 학습 시 스텝 수 7,500을 바탕으로 지수기반 스케줄링을 적용하였으며, 전체 스텝의 10% 구간을 웜업 단계로 설정하였다. 평가 결과에서는 BiGRU(85.48%), GRU(84.65%), BiLSTM(84.09%), LSTM(83.45%) 순으로 모형의 성능이 높게 나타났으며, log loss 역시 BiGRU가 0.3999로 가장 낮은 값을 기록하였다. 이러한 결과는 본 연구의 경제정책 뉴스기사에 대해 LSTM 계열보다 GRU 계열이, 단방향 구조보다 양방향 구조가 더 적합하다는 점을 시사하며, 특히 GRU 기반 모형의 상대적 우수성은 GRU가 LSTM보다 파라미터 수가 적고 구조도 간결하여 소규모 학습 데이터 환경에 더 안정적인 성능을 낼 수 있었던 것으로 해석된다. 다음으로 CNN은 Kim(2014)의 연구에서 소개된 모형 중 CNN-non-static과 CNN-multichannel을 기반으로 구축하였다. CNN-non-static은 1개의 채널만 사용하여 미세 조정을 수행하는 반면, CNN-multichannel은 2개의 채널을 사용하여 한 채널은 미세 조정되고 다른 채널은 정적으로 유지된다. 두 모형의 은닉층은 합성곱층 3개, 풀링층 3개, 완전연결층 1개로 구성되며, 이때 풀링층은 전역 최대 풀링이 풀링 연산으로 수행되고, 완전연결층으로 설계된 penultimate layer에서는 정규화를 위해 드롭아웃과 l2-노름이 적용된다. 하이퍼파라미터는 임베딩 벡터의 차원은 300, 합성곱층의 커널 크기는 (5, 3, 3), 필터 수는 50, 완전연결층의 뉴런 수는 32, 드롭아웃 비율은 0.2가 최적값으로 선택되었고, 학습 설정은 RNN과 동일하게 설정되었다. 평가 결과는 Kim(2014)이 기대했던 대로 CNN-non-static(85.21%)이 CNN-multichannel(85.04%)보다 높은 성능을 나타냈으며, log loss 지표에서도 CNN-multichannel이 0.4455로 더 낮은 값을 기록하였다. 마지막으로 트랜스포머는 2개의 아키텍처를 설계하여 비교하였다. 첫 번째 아키텍처인 Transformer-NSI는 서범석 외(2022)에서 제안된 구조이며, 두 번째 아키텍처인 Transformer-EPU는 본 연구에서 제안한 구조이다. 두 모형의 입력층은 앞서 구축한 Word2Vec 모형으로 초기화되며, 학

습 시 미세 조정이 수행된다. 이때, Transformer-NSI는 단어 임베딩에 위치 인코딩을 합산하여 입력 문장 행렬을 생성하지만, Transformer-EPU는 단어 임베딩에 위치 임베딩을 합산하여 입력 문장 행렬을 생성한다는 것에 차이가 있다. 트랜스포머 블록은 Transformer-NSI가 단일 인코더로 구성된 반면, Transformer-EPU는 2개의 인코더를 누적하여 심층적인 문맥 표현을 학습할 수 있도록 설계되었다. 이때, 각 인코더는 멀티 헤드 셀프 어텐션층과 위치별 완전 연결 FFNN층으로 구성되며, Transformer-NSI는 입력 마스킹을 위한 패딩 마스크가 추가로 적용된다. 트랜스포머 블록 이후에는 두 모형 모두 풀링층과 완전연결층을 거쳐 분류 작업이 수행되며, 이때 풀링 연산은 전역 평균 풀링이 사용된다. 하이퍼파라미터는 임베딩 벡터의 차원은 300, 어텐션 헤드 수는 4, 위치별 완전연결 FFNN의 뉴런 수는 128, 완전연결층의 뉴런 수는 128이 선택되었고, 학습 설정은 RNN과 동일하게 설정되었다. 평가 결과는 Transformer-EPU(85.59%)가 Transformer-NSI(85.25%)보다 높은 성능을 나타냈으며, log loss 역시 Transformer-EPU가 0.3945로 더 낮은 값을 기록하였다. 이상의 평가 결과들을 바탕으로 각 아키텍처에서 가장 우수한 성능을 기록한 BiGRU+Attention, CNN-multichannel, Transformer-EPU를 비교하여 본 연구의 감정 분류 모형을 선정하였다. F1 Score를 기준으로 세 모형의 성능을 비교하였을 때, Transformer-EPU(85.59%), BiGRU+Attention(85.48%), CNN-multichanne(85.04%) 순으로 높은 성능을 나타냈으나, BiGRU+Attention과 Transformer-EPU의 성능 차이는 불과 0.11%P밖에 나타나지 않았다. 이러한 결과는 본 연구와 같이 시퀀스 길이가 짧은 텍스트 데이터의 경우, 트랜스포머의 셀프 어텐션의 이점이 충분히 발휘되지 않아 BiGRU에 어텐션 메커니즘만 적용하더라도 효과적인 감정 분류가 가능함을 시사한다. 또한, 본 연구에서는 처리 속도 역시 중요한 고려 요소이므로 세 모형의 추론 시간을 측정하여 비교하였다. 그 결과, 10만 개의 문장을 하나씩 처리하였을 때, BiGRU+Attention은 약 5분, CNN-multichannel은 약 6분 40초, Transformer-EPU는 약 10분의 시간이 소요되었다. 따라서 본 연구에서는 감정 분류 성능뿐만 아니라 추론 속도 측면에서도 우수한 성능을 보인 BiGRU+Attention이 본 연구의 최적 모형이라 판단하였다.

비교 대상 모형은 BERT 계열의 PLM을 전이 학습에 활용하여 구축하였다. 고려된 PLM은 범용 도메인에서 학습된 KLUE-BERT와 뉴스기사 도메인에 특화된 KPF-BERT, 그리고 금융 도메인 특화된 KF-DeBERTa이다. 감정 분류 모형 구축 결과에서는 세 모형 모두 90%가 넘는 높은 분류 성능을 보였으며, F1 Score를 기준으로 KF-DeBERTa(92.52%), KPF-BERT(92.37%), KLUE-BERT(91.68%) 순으로 높은 성능을 나타냈다. 다만, KF-DeBERTa와 KPF-BERT의 성능 차이는 약 0.15%P밖에 나타나지 않았으며, 이 두 모형과 KLUE-BERT의 성능 차이도 각각 0.69%P와 0.84%P밖에 나타나지 않았다. 이처럼 범용 도메인과 도메인 특화 모형의 성능 차이가 크지 않은 것은 본 연구의 테스트셋이 소규모인 이유도 있겠지만, Zheng et al.(2021)과 본 연구에서 주장한 바와 같이 본 연구의 태스크 난이도가 쉬운 것에서 기인한 결과로 해석된다. 또한, 세 모형의 추론 시간을 비교한 결과, 10만 개의 문장을 하나씩 처리하였을 때, KLUE-BERT와 KPF-BERT는 약 3시간 40분이 소요되었고, KF-DeBERTa는 약 4시간 40분이 소요되었다. 따라서 비교 대상 모형 역시 감정 분류 성능과 추론 속도를 함께 고려하여 KLUE-BERT가 본 연구의 최적 모형이라 판단하였다.

Ⅳ장에서는 EPU 지수의 작성 결과와 유용성 평가 결과를 제시하였으며, 유용성 평가 결과는 경제통계와의 상관관계 분석, 불확실성이 거시경제에 미치는 영향 분석, 당기예측 모형을 활용한 예측력 평가로 나누어 제시하였다. 또한, 본 연구에서 작성한 EPU 지수(EPU_small)가 기존 EPU 지수들보다 불확실성 지수로서의 더 높은 적합성을 보이는지 확인하기 위해 기존 EPU 지수들과의 비교 분석을 수행하였다. 비교지표는 Baker et al.(2016)이 작성한 EPU 지수(EPU)와 Cho and Kim(2023)이 작성한 EPU 지수(EPU_KOREA)를 사용하였으며, 본 연구 모형과 EPU 지수의 실용성 및 신뢰성을 검증하기 위해 KLUE-BERT를 활용하여 작성한 EPU 지수(EPU_BERT)도 비교지표에 포함하였다.

먼저, EPU 지수의 작성 결과를 시각적으로 살펴보았을 때, EPU_small은 불확실성이 높은 시기에 모두 의미 있는 큰 값을 나타냈으며, 긍정적인 기사가 발행됐던 시기에도 불확실성 정도를 낮게 측정하는 모습을 보였다. 반면,

EPU와 EPU_KOREA는 특정 키워드의 출현 빈도만으로 불확실성을 측정하기 때문에 긍정적인 기사가 발행됐던 시기에 뚜렷한 저점을 보이지 못했으며, 특히 최근 코로나 팬데믹이 발생했던 시기의 불확실성 수준을 과소평가하는 모습이 관측되었다. 더욱이 EPU는 글로벌 금융위기와 미국 신용등급이 강등되었던 시기의 불확실성을 과소평가하였고, EPU_KOREA는 일본의 신용등급이 2단계 강등되었던 시기와 미중 무역전쟁이 본격화된 시기에 뚜렷한 큰 값을 나타내지 못했다. 한편, EPU_BERT는 본 연구의 주장대로 EPU_small과 거의 같은 움직임을 나타냈다.

경제통계와의 상관관계 분석에서는 EPU 지수의 선행성과 경제지표와의 연관성을 평가하기 위해 교차상관관계 분석을 수행하였다. 분석 결과, EPU_small은 VIX를 제외한 모든 지표와 가장 높은 상관관계를 나타냈으며, 월별 경제통계와는 1~5개월, 분기별 경제통계와는 0~1분기 선행하는 것으로 분석되었다. EPU는 경기 선행지수와 양(+), 환율 변동성과 음(-), 재화 수출입과 양(+)의 상관관계를 나타냈으며, EPU_KOREA는 민간소비, 설비투자, 재화 수출과 양(+)의 상관관계를 나타내 경제이론과 부합하지 않는 결과를 보였다. 또한, EPU_BERT는 EPU_small과 최대상관시차가 동일하고, 상관계수 값도 매우 유사한 값이 나타났다.

불확실성이 거시경제에 미치는 영향 분석에서는 EPU 지수의 외생성을 확인하고, 거시경제변수와의 관계에 있어서 정책적 의미를 지니는지 확인하기 위해 그랜저 인과관계 검정과 VAR 모형의 충격반응분석을 수행하였다. 동 분석은 선행연구들을 참고하여 월별 경제변수를 이용한 분석과 분기별 경제변수를 이용한 분석으로 나누어 수행되었다. 월별 경제변수를 이용한 그랜저 인과관계 검정 결과에서는 EPU_small과 EPU_BERT가 산업생산증가율, 주가수익률, 취업자 수 증가율에 대해 가장 뚜렷한 외생성을 나타냈으며, EPU_KOREA는 원/달러 환율에 대해 가장 뚜렷한 외생성을 나타냈다. 분기별 경제변수를 이용한 그랜저 인과관계 검정 결과에서는 EPU_small과 EPU_BERT가 모든 변수에 대해 가장 뚜렷한 외생성을 나타냈다. 월별 경제변수를 이용한 충격반응분석 결과에서는 EPU_small과 EPU_BERT가 산업생산증가율, 주가수익률, 취업자 수 증가율을 유의하게 감소시키는 것으로 나타났으며, 원/

달러 환율과 물가상승률은 유의하게 상승시키는 것으로 나타났다. EPU와 EPU_KOREA의 경우에는 주가수익률과 원/달러 환율이 통계적으로 유의하게 나타났지만, 나머지 변수는 전체 기간이 통계적으로 유의하지 않았다. 분기별 경제변수를 이용한 충격반응분석 결과에서는 EPU_small과 EPU_BERT가 소비 및 투자 심리, 대기업 대출태도지수, 가계(일반) 대출태도지수를 유의하게 하락시키고, 신용스프레드는 유의하게 상승시키는 것으로 분석되었다. 반면, EPU와 EPU_KOREA는 소비심리지표와 신용스프레드만 통계적으로 유의하고, 나머지 변수는 전체 기간이 통계적으로 유의하지 않았다.

마지막으로 당기예측 모형을 활용한 예측력 평가에서는 DFM을 기반으로 당기예측모형을 구축하고, 각 EPU 지수가 GDP 당기예측 향상에 도움이 되는지를 살펴보았다. 표본외 예측 수행 결과, EPU 지수를 반영하지 않은 기준모형(baseline)은 RMSE 0.9107, MAE 0.6346을 기록하였으며, EPU 지수가 반영된 4개의 모형은 모두 RMSE 기준에서 예측 성능의 개선을 보였다. 특히, EPU_small이 반영된 모형은 RMSE 0.8914, MAE 0.6273을 기록하여 두 지표 모두에서 가장 우수한 예측력을 나타냈으며, EPU_BERT가 반영된 모형 또한 RMSE 0.8925, MAE 0.6294를 기록하여 이와 매우 유사한 예측력을 나타냈다. 반면, EPU와 EPU_KOREA가 반영된 모형은 RMSE를 기준으로는 Baseline보다 소폭 낮은 값을 기록하였으나, MAE를 기준으로는 오히려 예측오차가 증가하였다. 이러한 결과는 본 연구에서 작성한 EPU 지수가 기존 EPU 지수들보다 높은 현실설명력을 제공하며, 당기예측 모형의 유용한 정보변수로 활용될 수 있음을 시사한다.

5.2 연구 의의

본 연구의 의의는 다음과 같다. 첫째, 문어체로 작성되는 뉴스기사를 기반으로 다양한 텍스트 전처리 도구들의 성능을 정량적으로 비교·평가하였다는 점이다. 텍스트 전처리는 자연어 처리의 성능을 크게 좌우하기 때문에 자연어 처리에 많이 의존하는 텍스트 마이닝 분야에서는 매우 중요하게 다루어지는 작업이다. 그러나 텍스트 마이닝을 활용한 대부분의 기존 연구들은 텍스트 전처리 과정에 대한 별다른 언급이 없거나, 구체적인 성능 평가 없이 최신 전처리 도구를 사용했다는 정도로 넘어가는 경우가 많다. 게다가 전처리 도구들은 대부분 구어체 텍스트 데이터를 기준으로 개발되기 때문에 문어체 텍스트 데이터에 대한 전처리 도구들의 성능 평가는 더욱 찾아보기가 힘들다. 따라서 본 연구에서 제시한 전처리 도구들의 평가 결과는 문어체 텍스트 데이터를 대상으로 최적의 도구를 선별하는데 유용한 참고자료로 활용될 수 있다.

둘째, 경제정책 도메인에 특화된 단어 임베딩 모형을 구축하고, 단어 임베딩 기법을 비교·평가하였다는 점이다. 단어 임베딩 관련 기존 연구들은 대부분 모형의 성능과 관련된 주제를 다루기 때문에 Common Crwal이나 모두의 말뭉치와 같은 공개된 데이터를 활용하여 모형을 구축하는 경우가 대부분이다. 따라서 본 연구와 같이 직접 학습 말뭉치 데이터를 구축하여 해당 도메인에 특화된 단어 임베딩 모형을 구축한 사례는 찾아보기 힘들며, 더욱이 사용자 사전을 구축하여 도메인 특화성을 강화한 사례는 본 연구가 처음으로 파악된다. 또한, 구축된 임베딩 모형의 성능을 정량적으로 비교·평가함으로써 임베딩 기법 간 표현력의 차이를 실증적으로 규명하였다는 점에서도 학술적 의의가 있으며, 이러한 비교·평가 결과는 향후 도메인 특화 임베딩 모형을 설계하고자 하는 연구자들에게 실무적 기준점을 제공할 수 있을 것이다.

셋째, 딥러닝 아키텍처를 비교·평가하여 최적의 성능을 도출하고, 유의미한 시사점을 제시하였다는 점이다. 본 연구에서는 RNN, CNN, 트랜스포머의 딥러닝 아키텍처를 기반으로 전체 8개의 감정 분류 모형을 구축하였으며, 다양한 신경망 구조와 하이퍼파라미터 설정을 검토하여 각 모형의 최적화를 달성하였다. 이는 단순한 아키텍처 간 성능 비교를 넘어, 실질적인 모형 설계

전략을 함께 제시하였다는 점에서 의의가 있다. 또한, 모형별 성능 비교를 통해 GRU 계열 모형이 소규모 학습 데이터 환경에서 상대적으로 높은 안정성을 보이며, 짧은 시퀀스 길이를 갖는 텍스트 데이터의 경우 트랜스포머 기반 모형이 항상 최적의 선택지가 아니라는 점을 실증적으로 제시함으로써, 데이터 특성과 태스크 조건에 따라 아키텍처를 선택하는 것이 중요하다는 실질적인 시사점을 제공하였다.

넷째, 한글 뉴스기사를 기반으로 직접 라벨링 데이터를 구축하고, 소규모 학습 데이터와 전이 학습을 활용하여 감정 분류 모형을 구축한 최초의 사례라는 점이다. 한글 텍스트 데이터와 전이 학습을 활용하여 감정 분류 모형을 구축한 선행연구는 안순재 외(2020)가 유일하다. 그러나 동 연구는 뉴스기사가 아닌 댓글 데이터를 활용하였으며, 라벨링 데이터 또한 공개 데이터셋인 NSMC(Naver Sentiment Movie Corpus)를 활용하였다는 점에서 직접 라벨링 데이터를 구축한 본 연구와 차별된다. 따라서 본 연구는 기계학습 기반의 감정 분류 모형을 구축하는데 가장 큰 진입장벽으로 작용하는 라벨링 데이터의 적정 규모를 결정하는 문제[159]에 대해 유일한 실무적 기준점을 제공할 수 있다.

다섯째, 처음으로 PLM과 단어 임베딩 모형의 실용성을 실무적으로 비교하였다는 점이다. 본 연구에서는 본 연구 모형과 EPU 지수의 실용성 및 신뢰성을 검증하기 위해 BERT 기반의 고성능 감정 분류 모형으로 작성한 EPU 지수를 본 연구의 비교지표로 활용하였으며, 다양한 유용성 평가를 통해 본 연구의 EPU 지수와 비교지표 간에 차이가 크지 않다는 것을 실증적으로 제시하였다. 이처럼 감정 분류 모형의 성능이 실제 다운스트림 태스크에 큰 영향을 미치는가에 대한 의문을 제기하고, 이를 실증적으로 규명한 연구는 본 연구가 처음이다. 따라서 본 연구의 유용성 평가 결과는 감정 분류 모형의 성능을 어느 수준까지 목표로 설정할 것인가에 대한 의문에 유일한 실무적 판단기준을 제공할 수 있다.

여섯째, 기존 EPU 지수의 방법론적 한계를 극복하고, 보다 현실설명력이

159) 라벨링 데이터의 적정 규모는 분석 자료, 태스크의 복잡도, 전이 학습의 활용 여부 등 다양한 요인에 의해 달라지기 때문에, 이를 사전에 결정하기 위해서는 유사 사례들을 참고하여 가늠할 수밖에 없다.

높은 불확실성 지수를 개발하였다는 점이다. 본 연구에서는 다양한 유용성 평가 결과를 통해 본 연구의 EPU 지수가 기존 EPU 지수들보다 선행성, 외생성, 그리고 경제변수들과의 상관관계가 우수함을 입증하였으며, 당기예측에서도 유용한 정보변수로 활용될 수 있음을 확인하였다. 따라서 본 연구의 EPU 지수는 정책 수립 및 효과 분석, 경기 모니터링, 동향 분석, 경제 예측 등 다양한 실무 분석에 활용될 수 있으며, 특히 EPU 지수는 속보성을 갖는 고빈도 경제지표로 작성된다는 장점이 있어 정부, 기업 등 경제주체들의 사전 대비책 마련과 정책적 판단의 적시성을 위한 참고 지표로 유용하게 활용될 수 있다.

일곱째, 소규모 학습 데이터를 사용함으로써 하위 부문별 EPU 지수 구축을 위한 실증적 기반을 마련하고, 나아가 뉴스기사를 활용한 다양한 정보변수 개발의 토대를 제공하였다는 점이다. 불확실성은 경제주체 또는 경제 부문별로 상이하게 측정되기 때문에 EPU 지수의 지속적 발전을 위해서는 통화, 고용, 부동산, 수출입 등의 하위 부문별 EPU 지수를 작성할 필요가 있다. 그리고 이를 위해서는 각 부문에 적합한 라벨링 데이터를 구축하는 작업이 병행되어야 하므로[160] 모형 성능을 위해 대규모 라벨링 데이터를 구축하는 것은 사실상 불가능에 가깝다. 따라서 소규모의 학습 데이터만 요구한다는 본 연구 모형의 강점은 부문별 EPU 지수 작성에 매우 유용하게 작용 될 수 있으며, 세부 부문별 정보변수 구축이 필요한 분야(예: GRDP와 같이 지역별 정보변수 구축이 필요하거나 산업별 정보변수 구축이 필요한 경우)에서도 본 연구 모형이 유용한 벤치마크로 기능할 수 있을 것이다.

160) 경제 분야의 뉴스기사는 객관적 사실 위주로 작성되는 스트레이트 기사가 많아, 같은 문장이어도 경제 부문 또는 경제주체 별로 서로 다른 라벨링이 부여되는 경우가 많다. 예를 들어 '임금이 상승했다'라는 문장은 경제주체의 관점에 따라 긍정적인 문장이 될 수도, 부정적인 문장이 될 수도 있으며, 또는 '환율이 상승하였다.'라는 문장은 수출 중심 산업을 고려했을 때 긍정적인 문장이 되지만, 수입 의존 산업을 고려했을 때는 부정적인 문장이 된다.

참 고 문 헌

1. 국내문헌

강형석, 양장훈. (2018). 한국어 단어 임베딩 모델의 평가에 적합한 유추 검사 세트.『디지털콘텐츠학회논문지』, 19(10), 1999-2008.

_____. (2020). Word2Vec 및 fastText 임베딩 모델의 성능 비교.『디지털콘텐츠학회논문지』, 21(7), 1335-1343.

김광민. (2021). “실물, 금융, 대외 부문을 종합하여 측정한 불확실성이 소비, 투자, 수출 및 통화정책 제약에 미치는 효과 분석”. 충북대학교 대학원 박사학위논문.

김동규, 이동욱, 박장원, 오성우, 권성준, 이인용, 최동원. (2022). KB-BERT: 금융 특화 한국어 사전학습 언어모델과 그 응용.『지능정보연구』, 28(2), 191-206.

김범식, 권순우. (2009).『경제의 불확실성과 설비투자 위축』. 서울: 삼성경제연구소.

김윤경. (2020).『경제정책 불확실성이 기업투자에 미치는 영향』. 서울: 한국경제연구원.

김인수. (2024). COVID-19 팬데믹 기간 동안 글로벌 공급망 충격 및 불확실성이 물가 및 거시경제에 미치는 영향.『한국경제연구』, 42(3), 71-98.

김천구, 박정수. (2020). 불확실성과 투자의 비가역성에 관한 연구.『응용경제』, 22(4), 5-42.

박은정, 조성준. (2014). KoNLPy: 쉽고 간결한 한국어 정보처리 파이썬 패키지. 제26회 한글 및 한국어 정보처리 학술대회 논문집, 133-136.

서범석, 이영환, 조형배. (2022). 기계학습을 이용한 뉴스심리지수(NSI)의 작

성과 활용.『국민계정리뷰』, 2022(1), 68-90.
신관호, 주원. (2002). 소득불확실성이 부의 축적과 소비에 미치는 효과.『경제분석』, 8(1), 100-134.
안순재, 유승의, 홍순구. (2020). 전이학습을 활용한 소규모 비정형 정책데이터 감성분석 모델.『한국데이터정보과학회지』, 31(2), 405-410.
오세환, 노성호. (2022). 경제 불확실성과 한국 기업의 투자 -한국 미국 중국 경제 정책 불확실성 지수를 활용한 실증 연구-.『동북아경제연구』, 34(3), 91-118.
유원준, 안상준. (2022).『딥 러닝을 이용한 자연어 처리 입문』. e-book: 위키독스.
윤용준, 이경태. (2013). 최근의 불확실성과 소비와의 관계 분석.『조사통계월보』, 67(3), 14-37.
이긍희, 조주희, 조진경. (2020). 새로운 우리나라 불확실성 지수의 작성.『응용통계연구』, 33(5), 639-653.
이기창. (2019).『한국어 임베딩』. 서울: 에이콘출판사.
이명훈, 최창규. (2000). 소비증가와 예비적 저축효과: 소비의 불확실성 가설의 검정.『경제학연구』, 48(2), 5-20.
이상우, 주동헌. (2021). 검색트렌드를 활용한 통화정책 환경 불확실성지수 구축.『금융연구』, 35(4), 53-85.
이우헌. (2001). 외환위기와 한국의 가계소비: 예비적 저축을 중심으로.『국제경제연구』, 7(2), 57-78.
이항용. (2005). 불확실성이 투자에 미치는 영향에 관한 실증분석.『KDI Journal of Economic Policy』, 27(2), 89-121.
이항용, 이진. (2017).『경제 불확실성 측정 지표 개발』. 서울: 국회예산정책처.
이현창, 정원석. (2016). 거시경제 불확실성 측정.『조사통계월보』, 70(3),

16-34.

이현창, 최동규, 김용건, 허정. (2022). 디지털 신기술을 이용한 실시간 당분기 경제전망(GDP nowcasting) 시스템 개발. 『BOK 이슈노트』, 2022(7), 1-29

장민, 황인도. (2004). 소득 및 자산가격 불확실성이 소비에 미치는 영향. 『조사통계월보』, 58(9), 23-50.

전은광, 김정대, 송민상, 유주현. (2023). KF-DeBERTa: 금융 도메인 특화 사전학습 언어모델. 제35회 한글 및 한국어 정보처리 학술대회 논문집, 143-148.

정원석, 이이수, 정희완. (2016). 불확실성 확대의 경제적 영향 분석. 『조사통계월보』, 70(5), 16-37.

정연구, 정예현, 郭亚奇, 이푸름. (2016). 모바일 시대의 기사 길이에 관한 탐색적 연구. 『한국언론정보학보』, 79(5), 140-164.

최정명, 김유섭. (2019). 영화리뷰 감성 분석에서 매개변수 변화에 따른 단어 임베딩 모델의 성능 비교. 한국정보과학회 2019 한국소프트웨어종합학술대회 논문집, 1,400-1,402

조성빈. (2017). 정책불확실성과 기업투자. 『금융지식연구』, 15(1), 3-28.

차은영, 최은영. (2007). 소득불확실성이 가계저축에 미치는 효과. 『한국여성경제학회』, 4(2), 91-113.

홍성욱, 민성환, 이소라. (2020). 『산업별 뉴스데이터를 활용한 경기 분석 및 예측 연구: 반도체와 자동차 업종을 중심으로』. 세종: 산업연구원.

황진태, & 김성민. (2020). 소비의 불확실성에 따른 예비적 저축 동기 추정. 『경제분석』, 26(3), 48-70.

2. 국외문헌

Abel, A. B. (1983). Optimal investment under uncertainty. *The American Economic Review*, 73(1), 228-233.

Araci, D. (2019). FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. *arXiv preprint arXiv:1908.10063*.

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bachmann, R., Elstner, S., & Sims, E. R. (2013). Uncertainty and economic activity: Evidence from business survey data. *American Economic Journal: Macroeconomics*, 5(2), 217-249.

Bahdanau, D. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Baker, S. R., Bloom, N., & Davis, S. J. (2016). Measuring economic policy uncertainty. *The quarterly journal of economics*, 131(4), 1593-1636.

Bańbura, M., Giannone, D., & Reichlin, L,. (2010). Nowcasting. *ECB Working Paper*, No. 1275.

Bańbura, M., & Modugno, M,. (2014). Maximum likelihood estimation of factor models on datasets with arbitrary pattern of missing data. *Jouranal of applied Econometrics*, 29(1), 133-160.

Bernanke, B. S. (1983). Irreversibility, uncertainty, and cyclical investment. *The quarterly journal of economics*, 98(1), 85-106.

Bloom, N. (2014). Fluctuations in uncertainty. *Journal of economic Perspectives*, 28(2), 153-176.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching

word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Bok, B., Caratelli, D., Giannone, D., Sbordone, A. M., & Tambalotti, A. (2018). Macroeconomic nowcasting and forecasting with big data. *Annual Review of Economics*, 10(1), 615-643.

Carroll, C. D., & Samwick, A. A. (1997). The nature of precautionary wealth. *Journal of monetary Economics*, 40(1), 41-71.

_____. (1998). How important is precautionary saving?. *Review of Economics and statistics*, 80(3), 410-419.

Castelnuovo, E., & Tran, T. D. (2017). Google it up! a Google Trends-based uncertainty index for the United States and Australia. *Economics Letters*, 161, 149-153.

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., & Androutsopoulos, I. (2020). LEGAL-BERT: The muppets straight out of law school. *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2898-2904.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321-357.

Cho, D., & Kim, H. (2023). Macroeconomic effects of uncertainty shocks: Evidence from Korea. *Journal of Asian Economics*, 84, 101571.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation.

arXiv preprint arXiv:1406.1078.

Clements, M. P. (2014). Forecast uncertainty—ex ante and ex post: US inflation and output growth. *Journal of Business & Economic Statistics*, 32(2), 206-216.

Dardanoni, V. (1991). Precautionary savings under income uncertainty: a cross-sectional analysis. *Applied Economics*, 23(1), 153-160.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dhingra, B., Liu, H., Salakhutdinov, R., and Cohen, W. W. (2017). A Comparative Study of Word Embeddings for Reading Comprehension. *arXiv preprint arXiv:1703.00993*.

Dixit, A. K., & Pindyck, R. S. (1994). *Investment under Uncertainty*. Princeton, NJ: Princeton University Press.

Evans, C., Fisher, J., Gourio, F., & Krane, S. (2015). Risk management for monetary policy near the zero lower bound. *Brookings papers on economic activity*, 2015(1), 141-219.

Fazzari, S. M., Hubbard, R. G., Petersen, B. C., Blinder, A. S., & Poterba, J. M. (1988). Financing Constraints and Corporate Investment. *Brookings Papers on Economic Activity*, 1988(1), 141-206.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., & Ruppin, E. (2002). Placing search in context: The concept revisited. *ACM Transactions on Information Systems (TOIS)*, 20(1), 116-131.

Gilchrist, S., Sim, J. W., & Zakrajšek, E. (2014). Uncertainty, financial

frictions, and investment dynamics. *FED Working Paper*, 2014-69.

Gladkova, A., Drozd, A., & Matsuoka, S. (2016). Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. *Proceedings of the NAACL Student Research Workshop*, 8-15.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge, MA: MIT Press.

Haddow, A., Hare, C., Hooley, J., & Shakir, T. (2013). Macroeconomic uncertainty: what is it, how can we measure it and why does it matter?. *Bank of England Quarterly Bulletin*, Q2.

Hahm, J. H., & Steigerwald, D. G. (1999). Consumption adjustment under time-varying income uncertainty. *Review of Economics and Statistics*, 81(1), 32-40.

Hakkio, C. S., & Keeton, W. R. (2009). Financial stress: What is it, how can it be measured, and why does it matter?. *Economic Review*, 94(2), 5-50.

Hall, R. E., & Jorgenson, D. W. (1967). Tax policy and investment behavior. *The American economic review*, 57(3), 391-414.

Feng, H., Chen, K., Deng, X., & Zheng, W. (2004). Accessor variety criteria for Chinese word extraction. *Computational linguistics*, 30(1), 75-93.

Hartman, R. (1972). The effects of price and cost uncertainty on investment. *Journal of economic theory*, 5(2), 258-266.

Hatzius, J., Hooper, P., Mishkin, F. S., Schoenholtz, K. L., & Watson, M. W. (2010). Financial conditions indexes: A fresh look after the financial crisis. *NBER Working Paper*, 16150.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770－778.

He, P., Liu, X., Gao, J., & Chen, W. (2020). DeBERTa: Decoding-enhanced BERT with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735－1780.

Jin, Z., & Tanaka-Ishii, K. (2006). Unsupervised Senmentation of Chinese Text by Use of Branching Entropy. *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, 428-435.

Jurado, K., Ludvigson, S. C., & Ng, S. (2015). Measuring uncertainty. *American Economic Review*, 105(3), 1177-1216.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746－1751.

Knight, F. H. (1921). *Risk, uncertainty and profit*. Boston, MA: Houghton Mifflin Company.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Lahiri, K., & Sheng, X. (2010). Measuring forecast uncertainty by disagreement: The missing link. *Journal of Applied Econometrics*, 25(4), 514-538.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

Lee, Y., Kim, S., & Park, K. (2019). Deciphering Monetary Policy Committee Minutes with Text Mining Approach: A Case of Korea. *The Korean Economic Review*, 35(2), 471-511.

Leland, H. E. (1968). Saving and uncertainty: The precautionary demand for saving. *The Quarterly Journal of Economics*, 82(3), 465-473.

Lucca, D. O., & Trebbi, F. (2009). Measuring central bank communication: an automated approach with application to FOMC statements. *NBER Working Paper*, 15367.

Luong, M. T. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Manning, C., Socher, R., Fang, G. G., & Mundra, R. (2017). *CS224n: Natural Language Processing with Deep Learning1*. Stanford, CA: Stanford University.

Mariano, R. S., & Murasawa, Y. (2003). A new coincident index of business cycles based on monthly and quarterly series. *Journal of applied Econometrics*, 18(4), 427-443.

McDonald, R., & Siegel, D. (1986). The value of waiting to invest. *The quarterly journal of economics*, 101(4), 707-727.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *arXiv*

preprint arXiv:1301.3781.

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10), 1345-1359.

Park, K., Lee, J., Jang, S., & Jung, D. (2020). An empirical study of tokenization strategies for various Korean NLP tasks. *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, 133 - 142.

Park, S., Byun, J., Baek, S., Cho, Y., & Oh, A. (2018). Subword-level word vector representations for Korean. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2429-2438.

Park, S., Moon, J., Kim, S., Cho, W. I., Han, J., Park, J., ... & Oh, T. (2021). KLUE: Korean Language Understanding Evaluation. *arXiv preprint arXiv:2105.09680*.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532-1543.

Pesaran, H. H., & Shin, Y. (1998). Generalized impulse response analysis in linear multivariate models. *Economics letters*, 58(1), 17-29.

Peters, M., Ruder, S., and Smith, N. A. (2019). To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks. *arXiv preprint arXiv:1903.05987*.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Pindyck, R. S. (1990). Irreversibility, Uncertainty, and Investment. *Journal of Economic Literature*, 29(3), 1110-1148.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. *OpenAI preprint*.

Ruder, S. (2019). Neural transfer learning for natural language processing (Doctoral dissertation). National University of Ireland, Galway.

Schnabel, T., Labutov, I., Mimno, D., & Joachims, T. (2015). Evaluation methods for unsupervised word embeddings. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 298-307.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11), 2673-2681.

Sennrich, R., Haddow, B., & Birch, A. (2016). Neural machine translation of rare words with subword units. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1715-1725.

Sims, C. A. (1980). Macroeconomics and reality. *Econometrica: journal of the Econometric Society*, 1-48.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., ...

& Natarajan, V. (2023). Large language models encode clinical knowledge. *Nature*, 620(7972), 172-180.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1-9.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 5998-6008.

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, 3, 1-40.

Wenzek, G., Lachaux, M. A., Conneau, A., Chaudhary, V., Guzmán, F., Joulin, A., & Grave, E. (2019). CCNet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*.

Wilson, B. K. (1998). The aggregate existence of precautionary saving: Time-series evidence from expenditures on nondurable and durable goods. *Journal of macroeconomics*, 20(2), 309-323.

Wu, S., Irsoy, O., Lu, S., Dabravolski, V., Dredze, M., Gehrmann, S., Kambadur, P., Rosenberg, D., & Mann, G. (2023). BloombergGPT: A Large Language Model for Finance. arXiv preprint arXiv:2303.17564.

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems*, 28, 649-657.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., ... & Wen,

J. R. (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Zheng, L., Guha, N., Anderson, B. R., Henderson, P., & Ho, D. E. (2021). When does pretraining help? Assessing self-supervised learning for law and the CaseHOLD dataset of 53,000+ legal holdings. *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law (ICAIL '21)*, 159–168.

HANSUNG UNIVERSITY

ABSTRACT

Economic Policy Uncertainty Index Using Small-Scale Learning Data and Transfer Learning

Lee, Cheon-Woo

Major in Economics

Dept. of Economics & Real Estate

The Graduate School

Hansung University

Uncertainty is a major factor influencing the formation of expectations, decision-making by economic agents, and the real economy. However, as it is not directly observable, various efforts have been made to measure it. In particular, the Economic Policy Uncertainty (EPU) index proposed by Baker et al. (2016) is a representative example that measures uncertainty using big data. It calculates uncertainty based on the frequency of news articles that simultaneously contain three word groups: economic terms, policy terms, and uncertainty-related terms. However, measuring uncertainty solely based on the frequency of specific keywords may lead to overestimation and fails to capture the diverse contextual information embedded in news content. To address these limitations of the existing EPU index, this study proposes a new EPU index that leverages machine learning-based sentiment analysis.

To perform machine learning-based sentiment analysis, it is necessary to construct a sentiment classification model. However, building such a model requires large-scale labeled data and the use of state-of-the-art natural language processing techniques, which impose significant time and resource burdens that are difficult for individual researchers or small research teams to bear. To alleviate these practical constraints, this study proposes a sentiment classification model that utilizes small-scale training data (approximately 20,000 labeled data) in combination with transfer learning. In particular, considering the high computational cost and inference latency of pre-trained language models (PLMs) such as BERT, this study focuses on demonstrating that useful models can still be built by using word embedding models for transfer learning.

Meanwhile, this study argues that, because the downstream task in this research is relatively simple, the difference between the EPU index constructed using a high-performance sentiment classification model and that constructed using the proposed model would not be significant. To empirically verify this claim, a separate sentiment classification model was developed using a BERT-based PLM, and the EPU index generated from that model was used as a benchmark for comparison. Various usefulness evaluations were then conducted using both EPU indices to empirically support the argument of this study.

The research process was conducted in six stages: ① news article collection, ② text preprocessing, ③ embedding model construction, ④ sentiment classification model construction, ⑤ comparison model construction, and ⑥ usefulness evaluation. For ① news article collection, article titles were gathered using the keyword sets proposed by Lee Geung-hee et al. (2020). After removing duplicate and irrelevant articles, the titles were searched on portal sites to collect the full texts.

② Text preprocessing was carried out in five stages: pre-cleaning,

corpus definition, sentence tokenization, user dictionary construction, and morphological analysis. In the pre-cleaning process, unnecessary strings were removed, special characters were converted, and words within parentheses were deleted. The corpus was defined at the sentence level. For sentence tokenization, various sentence splitters were evaluated to identify the most suitable one for this study. The evaluation criteria included processing speed, accuracy, and the Dice Coefficient. As a result, NLTK demonstrated superior performance in both segmentation accuracy and processing speed. The user dictionary was constructed using the noun extractor from soynlp, resulting in a dictionary consisting of 71,126 common nouns and 16,787 proper nouns. In the morphological analysis stage, various morphological analyzers were evaluated to determine the one best suited for this study. The evaluation was divided into part-of-speech tag comparison, ambiguity assessment, processing speed comparison, and review of compatibility with the user dictionary. As a result, Kiwi was selected as the morphological analyzer for this study.

③ The word embedding models were constructed by comparing three methodologies: Word2Vec, FastText, and GloVe. After building each embedding model, performance evaluations were conducted to select the most suitable model for this study. The evaluation was divided into three parts: word analogy evaluation, word similarity evaluation, and extrinsic evaluation. The extrinsic evaluation measured performance improvement in the sentiment classification task. The results showed that Word2Vec achieved the best performance across all three evaluation categories.

④ The sentiment classification models were constructed based on three deep learning architectures: RNN, CNN, and Transformer. Various neural network structures and hyperparameter configurations were explored to derive optimal performance. For the RNN architecture, model variants incorporating attention mechanisms were considered, including LSTM,

BiLSTM, GRU, and BiGRU. The CNN models were designed based on the CNN-non-static and CNN-multichannel structures introduced by Kim (2014). In model evaluation, the Transformer-based model achieved the highest performance in terms of F1 Score. However, the performance difference with the BiGRU+Attention model was only 0.11 percentage points. Moreover, the BiGRU+Attention model showed inference speeds twice as fast as the Transformer model. Therefore, BiGRU+Attention was selected as the optimal model for this study, given its strong performance in both classification accuracy and inference efficiency.

⑤ The comparison models were constructed using transfer learning with BERT-based pre-trained language models (PLMs). Specifically, KLUE-BERT trained on general-domain data, KPF-BERT specialized for news articles, and KF-DeBERTa specialized for the financial domain were employed. In model evaluation, KF-DeBERTa achieved the highest F1 Score. However, the performance difference with KPF-BERT was only about 0.15 percentage points. Furthermore, inference speed analysis showed that KPF-BERT was faster than KF-DeBERTa. Based on this, KPF-BERT was selected as the optimal comparison model in this study.

⑥ The usefulness evaluation was conducted in three parts: correlation analysis with economic statistics, analysis of the effects of uncertainty on the macroeconomy, and forecasting performance evaluation using a nowcasting model. To assess the appropriateness and robustness of the EPU index, the following four indices were used: the EPU index constructed using the proposed model (EPU_small), the EPU index constructed using the comparison model (EPU_BERT), the original EPU index developed by Baker et al. (2016) (EPU), and the EPU index proposed by Cho and Kim (2023) (EPU_KOREA).

In the correlation analysis with economic statistics, cross-correlation analysis was conducted to evaluate the leading property of each EPU

index and its relationship with macroeconomic indicators. The results showed that EPU_small exhibited the highest correlation with all indicators except for the VIX index, and it was found to lead monthly economic indicators by 1 to 5 months and quarterly indicators by 0 to 1 quarter. The EPU index showed a positive correlation with the leading composite index, a negative correlation with exchange rate volatility, and a positive correlation with goods exports and imports. In contrast, EPU_KOREA showed positive correlations with private consumption, facility investment, and goods exports—results that do not align with established economic theory. Additionally, EPU_BERT exhibited the same maximum correlation lags as EPU_small, and their correlation coefficients were found to be highly similar.

In the analysis of the effects of uncertainty on the macroeconomy, Granger causality tests and impulse response analysis based on VAR models were conducted to verify the exogeneity of the EPU indices and to determine whether they have policy-relevant implications in relation to macroeconomic variables. Referring to previous studies, the analysis was divided into two parts: one using monthly economic variables and the other using quarterly economic variables. In the Granger causality test using monthly economic variables, EPU_small and EPU_BERT showed the most distinct exogeneity with respect to the industrial production growth rate, stock return, and employment growth rate. For the KRW/USD exchange rate, EPU_KOREA exhibited the most distinct exogeneity. In the Granger causality test using quarterly economic variables, EPU_small and EPU_BERT showed the strongest exogeneity across all variables. In the impulse response analysis using monthly economic variables, EPU_small and EPU_BERT were found to significantly reduce the industrial production growth rate, stock return, and employment growth rate, while significantly increasing the KRW/USD exchange rate and the inflation

rate. In contrast, EPU and EPU_KOREA showed statistically significant effects only on stock return and the KRW/USD exchange rate, with the other variables remaining statistically insignificant throughout the analysis period. In the impulse response analysis using quarterly economic variables, EPU_small and EPU_BERT were found to significantly lower consumer and investment sentiment, the lending attitude index of large enterprises, and the general household lending attitude index, while significantly increasing the credit spread. On the other hand, EPU and EPU_KOREA showed statistical significance only for consumer sentiment and credit spread, with the remaining variables remaining statistically insignificant over the entire period.

In the forecasting performance evaluation using a nowcasting model, a DFM-based nowcasting model was constructed to examine whether each EPU index contributes to improving the nowcast of GDP. In the out-of-sample forecasting results, the baseline model without any EPU index recorded an RMSE of 0.9107 and an MAE of 0.6346. All four models incorporating EPU indices showed improvements in forecasting performance based on RMSE. In particular, the model incorporating EPU_small recorded an RMSE of 0.8914 and an MAE of 0.6273, demonstrating the best forecasting accuracy on both metrics. The model using EPU_BERT also showed similar performance, with an RMSE of 0.8925 and an MAE of 0.6294. In contrast, the models incorporating EPU and EPU_KOREA recorded slightly lower RMSE values than the baseline, but their MAE values were higher, indicating increased forecast errors by that metric. These results suggest that the EPU index developed in this study provides greater real-world explanatory power than existing EPU indices and can serve as a useful informational variable in nowcasting models.

The contributions of this study are as follows. First, it quantitatively

compared and evaluated the performance of various text preprocessing tools based on news articles written in formal written language. Second, it developed a word embedding model specialized in the economic policy domain and conducted comparative evaluations of different word embedding techniques. Third, it compared and evaluated deep learning architectures to derive optimal performance and presented meaningful implications. Fourth, this study is the first to construct a sentiment classification model by directly creating labeled data based on Korean news articles and utilizing small-scale training data in combination with transfer learning. Fifth, this study is the first to conduct an application-oriented comparison of the utility of PLM and word embedding model. Sixth, it overcame methodological limitations of existing EPU indices and developed a new uncertainty index with stronger real-world explanatory power. Seventh, by employing small-scale training data, it provided an empirical foundation for constructing sub-sectoral EPU indices and laid the groundwork for the development of various information variables based on news articles.

【Key Word】Economic Policy Uncertainty Index, Small-Scale Learnin Data, Transfer Learning, Big Data, Natural Language Processing, Artificial Neural Network, Deep Learning