**통계청 지원사업 결과보고서**

- 뉴스 기사를 활용한 고용률 예측 방법에 대한 AI 연구 -

2020.12.23.

이전에는 정책 결정을 지원하기 위해 ‘고용률’과 같은 직접적인 설문이나 조사를 통해 집계되는 지표(지수)를 이용하였다. 하지만 집계를 통해 기록되는 지표의 경우, 사람의 노동력이 직접적으로 투입되어야 하고 단기간의 고용률 변화를 알기 어렵다는 문제점이 있다. 또한, 코로나 19의 확산, 디지털 경제로의 전환 등 급변하고 있는 경제활동 주체들의 형태 변화에 따라, 보다 짧은 단위의 초속보성 경기지표를 개발할 필요성이 늘어나고 있다.

이에 따라 본 프로젝트는 이러한 수요를 반영한 선행 연구로서, 뉴스 기사를 바탕으로 현재 고용 상태에 대한 심리 분석 및 고용률의 단기간 변화를 예측하기 위한 모델을 개발하는 것을 목표로 한다. 먼저 이 모델을 구현하기 위해서는 해당 기사가 고용 관련 기사인지 분류하고, 관련 기사가 맞다면 고용에 대해 긍정적인지, 부정적인지 감정을 분류하는 모델이 필요하다. 이에 이번 프로젝트에서는 자연어 처리 분야에서 최근 주목받고 있는 BERT 모델을 활용하여 **▲고용 기사 분류 모델과 ▲기사에 대한 감정 분석 모델**을 개발하고, 두 모델로부터 뉴스 기사의 감정을 추출한 후 **▲짧은 기간에 대한 고용률을 예측하는 모델**을 개발하였다.

**1. 자연어 처리 기술 소개 및 최근 연구 동향**

주어진 기사가 고용에 관련된 기사인지 판단하거나, 그 기사에 대한 감정을 분석하기 위해서는 ‘문장 분석 기술'이 필요하다. 즉, 인공지능으로 언어를 분석하기 위해서는 자연어 처리(natural language processing, NLP) 기술을 사용해야 한다.

현재 NLP 분야에서 가장 뛰어난 성능을 자랑하며, 연구계에 가장 많은 관심을 받고 있는 모델은 ‘BERT(Bidirectional Encoder Representations from Transformers)’다. 지난 2018년 구글이 처음 공개한 BERT는 이미 대량의 데이터로 사전 훈련된(pre-trained) 모델로, 이를 활용하면 적은 데이터로도 쉽게 학습을 진행해 좋은 성능을 낼 수 있다는 장점이 있다.

또한, BERT는 기존 자연어 처리 분야에서 주로 활용되었던 ‘RNN(Recurrent Neural Network)’의 한계를 극복했다는 점에서 의미가 있다. RNN은 문장이 길어질수록 앞쪽에 나온 단어들을 잘 기억하지 못한다는 특징 때문에 모델을 거대화하기 어렵다는 문제가 있었다. 하지만, BERT는 ‘Self-Attention’ 이라는 개념을 활용하여 긴 문장도 원활히 처리할 수 있을 뿐만 아니라, 거대한 모델을 만들 수 있게 되었다.

Self-Attention 개념을 살펴보면, 어떤 문장이 있을 때 중요하다고 여겨지는 부분에 ‘Attention’이라는 일종의 가중치를 부여하는 것이다. <그림1>을 보면 RNN은 모든 단어에 대해서 같은 가중치를 가지지만, Self-Attention을 사용하면 모델이 학습 과정에서 성능을 높이기 위해 집중해야 할 단어들에 높은 가중치를 부여하는 것을 학습하게 된다. 이를 통해 문장이 길어져도 모델이 중요한 부분에 집중할 수 있어 보다 높은 성능을 낼 수 있다.

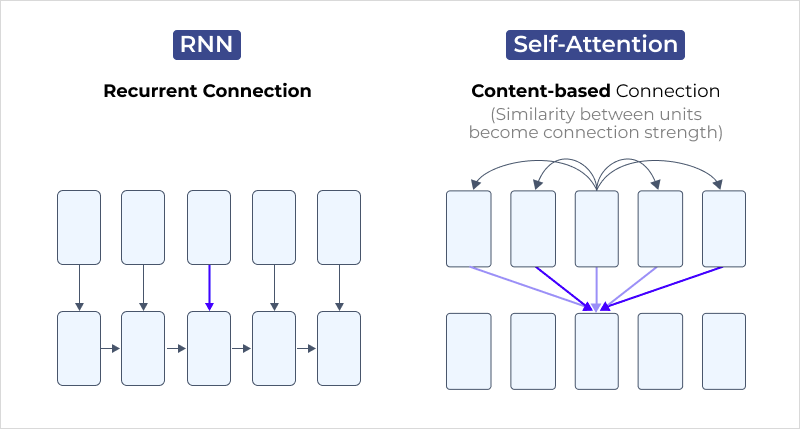


그림 1. RNN과 Self-Attention의 구조

BERT의 등장으로, 자연어 처리 분야는 사전 학습된 모델을 바탕으로 응용분야의 재학습을 진행해 단기간 내 뛰어난 성능을 낼 수 있게 되었다. 시간이 흐르면서 BERT의 성능을 높이기 위해 DistilBERT나 ELECTRA와 같은 BERT 기반 모델들이 고안되었다**.**

먼저 **DistilBERT**는 ‘Teacher-Student’ 방법을 이용해서 학습시킨 모델이다. Teacher-Student 방법은 미리 학습된 큰 네트워크(teacher)를 이용해서 작은 네트워크(student)를 학습시키는 것이다. 이 방법을 사용하면 teacher 네트워크와 견줄 수 있는 성능을 가지지만, 크기는 더 작은 모델을 얻을 수 있다. 이때 <그림2>의 큰 네트워크는 BERT가 되고, BERT와 성능은 비슷하지만 그 크기가 더 작은 모델은 DistilBERT다.

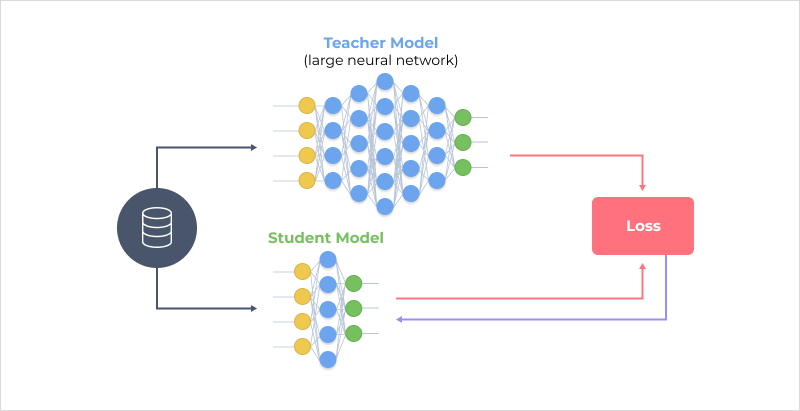
****

그림 2. DistilBERT에 사용되는 teacher-student 방법

이외에도, BERT 모델을 학습시킬 때 마스킹된 단어를 활용하는 ELECTRA 모델이 있다. 마스킹은 문장에서 일부 단어를 [MASK]라는 토큰으로 바꾸는 과정을 말한다. ELECTRA 모델을 학습시킬 때는 <그림3>와 같이 본격적인 학습 전에 문장에 마스킹을 진행한다. 이후 Generator가 마스킹된 단어를 예측해서 새로운 문장을 만들고, 이 문장을 가지고 Discriminator(ELECTRA)가 원본 문장과 일치하는지, 그렇지 않은지를 판별하는 식으로 학습이 진행된다. 마스킹된 단어를 맞추는 방식으로 학습을 진행했기 때문에 모르는 단어가 나와도 그 성능이 쉽게 저하되지 않는다는 장점이 있다.

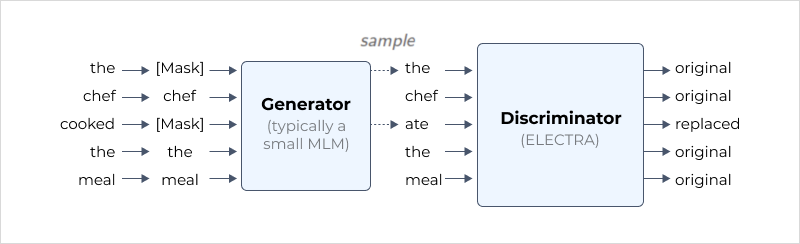


그림 3. ELECTRA 모델이 학습되는 과정

정리하면 DistilBERT는 BERT에 비해 모델 크기는 작고 속도는 빠르면서도 비슷한 성능을 가진 모델이고, ELECTRA는 새로운 단어가 많을 때에도 성능 저하 없이 BERT와 비슷한 성능을 내는 모델이다. 기본 모델인 BERT와 이를 더 개발시킨 DistilBERT, ELECTRA 모두 다른 특성을 지녔지만, 이 중 어떤 모델이 더 뛰어난지 우위를 가리기는 사실상 어렵다. 실험하고자 하는 분야에 따라 그 결과가 달라지기 때문이다. 이러한 이유로, 본 프로젝트에서는 세 가지 모델에 대한 실험을 통해 당사가 개발하고자 하는 고용률 예측 모델에 최적화된 모델을 찾아보기로 하였다.

**2. AI 알고리즘 개발**

**가. 전체 구조: 감정 모델(뉴스분류기 & 감정분석기) + 고용률 예측 모델**

본 과제에서 개발한 전체 고용률 예측 모델은 크게 감정 모델과 고용률 예측 모델, 2개의 하위 모델로 구성된다. 먼저 감정 모델은 뉴스가 주어졌을 때 해당 뉴스가 고용과 관련된 모델인지 분류하고, 이에 대한 감정 정보를 뽑아내는 모델이다. 이렇게 추출된 감정 정보(embedding, emb)는 고용률 예측 모델에서 사용하기 위해 DB에 저장된다.

그리고 고용률 예측 모델은 감정 모델을 바탕으로 뉴스로부터 추출한 감정 정보와 기존 고용률 데이터를 사용해, 특정 날짜에 대한 고용률을 예측한다. 특정 시점(t)에서의 고용률은 그 이전의 고용률()들과 해당 시점 근처의 뉴스()로부터 추출한 고용 감정 지수로부터 예측할 수 있다. 이 때, 고용률은 특정 시점()으로부터 월/주/일 단위로 L 길이만큼 과거의 고용률들을 사용하고, 뉴스 정보는 특정 시점 포함 W일 전까지의 뉴스들을 사용한다. 전체 모델 구조는 <그림4>와 같다.

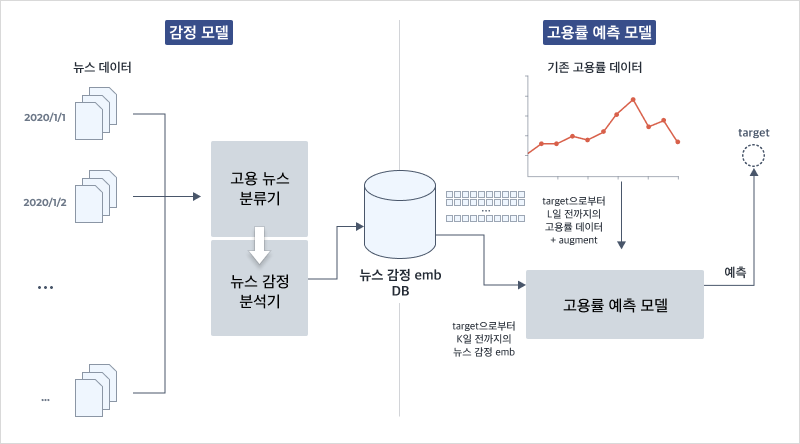


그림 4. 전체 고용률 예측 모델 구조

|  |
| --- |
| **※ 감정 정보와 고용률 간의 상관관계**  고용률 예측 모델에는 감정 모델에서 뽑은 심리 정보가 들어가고, 이를 기반으로 고용률을 예측하는데, 꼭 양의 상관관계가 아니라도 상관관계가 있다면 고용률 예측 모델을 통해 예측에 사용될 수 있다.  Measuring News Sentiment paper에서 뉴스에서 뽑아낸 소비자 감정 지수와 \*MCSI, 경기동행지수(CBCI) 간에 어느 정도의 상관관계가 있음을 보였고, 제시한 모델 구조 <그림4>와 같은 방법으로 이전 MCSI/CBCI 값들과 뉴스 감정 지수를 이용하여 다음 시간의 MCSI/CBCI 값들을 예측한 바 있다.  \* 미시건 대학교 소비자 신뢰지수(University of Michigan Consumer Sentiment, MCSI) |
|  |

**나. 감정 모델(뉴스분류기 & 감정분석기)**

감정 모델은 뉴스 텍스트로부터 직접 감정 정보를 추출하는 모델이다. 본 프로젝트에서는 선행 연구로서 고용률과 관련된 감정 정보를 뽑는 것을 목적으로 하였기 때문에, 이 모델은 해당 뉴스가 고용 관련 뉴스인지 구분하는 ‘뉴스 분류기’와 고용 뉴스를 대상으로 고용 감정을 분류하고 감정 정보를 뽑아내는 ‘감정 분석기’로 구성된다. 감정 모델의 전체 그림은 <그림5>와 같다.

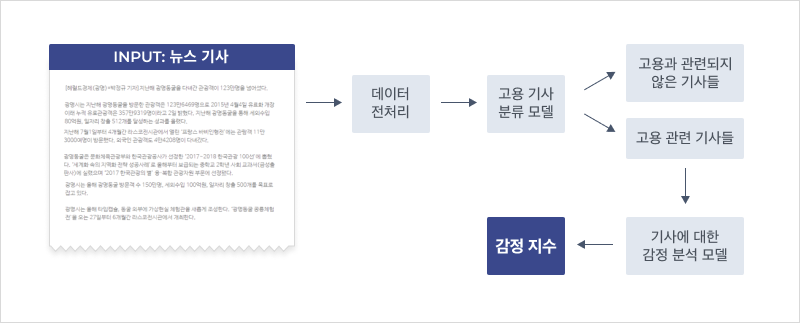


그림 5. 감정 모델 구조도

**1) 감정 모델 데이터 전처리**

AI 모델 개발에서 데이터의 양과 질은 모델의 성능을 좌우할 정도로 중요한 역할을 한다. AI가 잘 훈련을 할 수 있도록 데이터 자체를 정제하는 ‘기사 필터링’ 작업과 딥러닝 학습을 위해 텍스트 데이터를 벡터로 변환하는 ‘Sentence piece tokenizer’ 과정을 거쳤다.

**○ 기사 필터링**

모델을 학습시키기 위해 다수의 기사들을 모았지만, 적절하지 않은 기사들도 포함되어있기 때문에 수집한 기사들을 전부 다 학습에 사용할 수는 없기 때문에 필터링 작업을 진행했다.

우선 뉴스 기사 특성 상 한 가지 이슈에 대해서 여러 개의 기사가 나오는데, 이렇게 비슷한 내용의 기사들이 존재하게 되면 후에 학습에 사용할 데이터와 성능 분석에 사용할 데이터에 모두 포함될 수 있다. 즉, 학습에 사용한 데이터가 그대로 성능 분석에 활용되는 것이다. 이러한 일을 방지하기 위해서 제목이 중복되는 기사들을 제거하는 과정을 거쳤다.

또한, ‘고용동향’이나 ‘고용 동향’이라는 키워드가 기사에 포함된다면 그 기사는 통계청이 주기적으로 발표하는 고용 동향을 그대로 가져온 기사일 가능성이 높다. 그렇게 되면 그 기사는 통계청의 입장을 다시 말하는 것이기 때문에 고용률에 대한 감정이 들어있지 않다고 볼 수 있고, 이런 기사를 학습 데이터로 사용하게 되면 고용률에 대한 감정을 왜곡할 수 있기 때문에 ‘고용동향’이나 ‘고용 동향’이라는 키워드가 포함된 기사를 제외했다.

더불어 해당 기사가 고용과 확실히 관련이 있어야 하기 때문에 74개의 고용 연관어가 무조건 포함된 기사만 사용하고, 현재 고용률에 대한 입장이 들어 있어야 하기 때문에 411개의 인용 어휘가 포함된 기사만 사용했다. 마지막으로 기사가 너무 짧으면 필요한 데이터를 충분히 포함하고 있지 않기 때문에 기사의 본문이 200자 이하인 기사는 사용하지 않았다.

|  |
| --- |
| **※ 유형별 뉴스 데이터 처리**   1. 수집 오류 기사 및 중복 기사 삭제 2. ‘통계청 고용동향’ 키워드 기사 삭제 3. 74개의 고용 연관어가 포함된 기사 활용 4. 411개의 인용 어휘가 포함된 기사 활용 5. 본문 200자 이하의 기사 삭제 |

**○ Sentence piece 사용**

딥러닝이 텍스트 데이터를 학습하기 위해서는 이를 딥러닝이 인지할 수 있는 숫자로 변환하는 전처리 단계가 필요하다. 이를 워드 임베딩(Word Embedding)이라고 하며, 자연어 처리의 가장 기초적인 단계로 볼 수 있다.

기존 딥러닝은 각 단어에 index를 부여하고, index 별로 학습 가능한 벡터(특정 수치값)를 할당하는 방식을 활용했다. 현실적으로 모든 단어에 index를 부여하는 것은 불가능하기 때문에 당사는 이번 전처리 과정에서 <그림6>와 같이 자주 사용하지 않는 단어는 unknown을 의미하는 [UNK]로 처리하는 과정을 추가해, 양질의 데이터를 구축하고자 하였다. 하지만 이 방식은 사용 빈도는 낮지만 중요한 단어들이 [UNK]로 처리되어 모델의 성능을 떨어뜨릴 위험성이 존재하게 된다. 예를 들면, <그림6>의 ‘공유경제’와 같은 단어가 자주 사용되지 않기 때문에 [UNK]로 바뀌게 되면 해당 단어의 정보가 사라지게 되는 것이다.

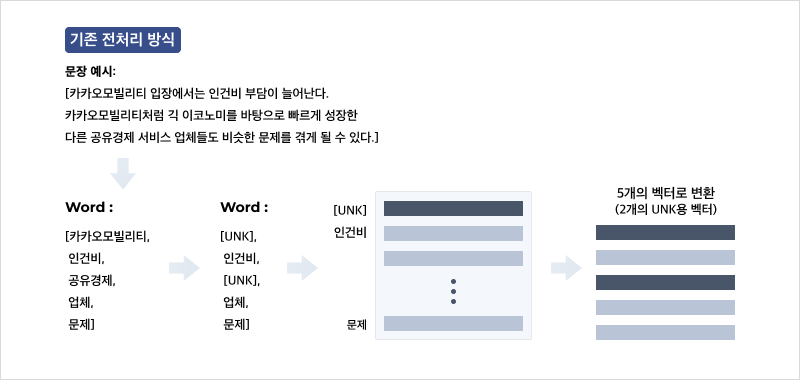


그림 6. 기존 전처리 방식 적용 예시

이 문제를 해결하기 위해 이번 전처리 과정에서는 개별 단어를 보다 작은 단위인 subword로 구분하여 벡터로 변환하는 ‘Sentence piece tokenizer’방식을 사용했다. Subword들은 단어보다 더 작은 단위의 토큰으로, 데이터로부터 알고리즘을 통해 만들어진다. ‘Sentence piece tokenizer'는 단어/문장을 subword 단위로 나누어서 벡터로 변환함에 따라 중요하지만 자주 사용되지는 않는 단어들이 [UNK]로 바뀌어서 정보가 사라지는 것을 줄이고, 복합어와 같이 여러 의미를 가진 단어들이 각각의 의미로 분해될 것을 기대할 수 있다. 예를 들면, 먼저 ‘카카오모빌리티’는 ‘카카오’와 ‘모빌리티’라는 subword로 나뉘게 된다. 그 후에 해당 subword들을 벡터로 변환하게 되는데, 자주 쓰이지 않는 단어인 ‘모빌리티’는 [UNK]로 변환된다. <그림7>을 보면 기존 전처리 방식을 사용했을 때, [UNK] 심볼로 처리되는 ‘공유경제’와 같은 단어도 ‘공유’, ‘경제’처럼 subword 단위로 나눠지면 둘 다 개별적인 의미가 존재하기 때문에 그 의미를 잃지 않고 학습에 활용될 수 있다.

이 과정에서 [UNK]로 세팅될 단어는 전체 subword들 중에서 빈도 수에 따라 구분된다. 예를 들어 전체 subword들 중에서 빈도수 기반 30000 subword까지만 사용하고 나머지는 UNK로 간주하는 방식이다. 새로운 corpus에서 token들을 새로 생성하면 출현 빈도에 따라 초기 UNK로 세팅된 단어도 새로 index에 포함될 수 있다. 이 때, 이 주기는 sentence piece subword들을 새로 학습시킬 때마다 달라진다.

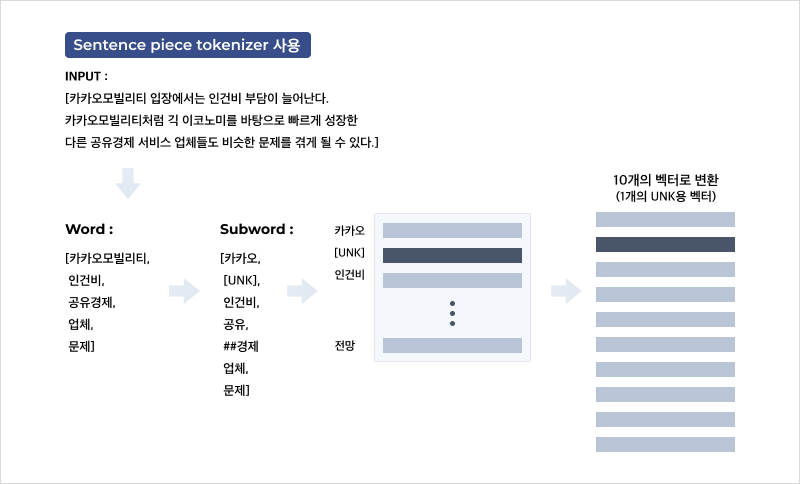


그림 7. Sentence piece tokenizer 적용 예시

하지만, 이 방식만을 사용하면 단어의 의미 해체가 일어날 수 있다. 이를 방지하기 위해 self-attention 모델을 같이 사용하여 해당 subword들의 embedding을 만들 때 주변 단어(context)를 같이 고려하도록 설계하였다. 예를 들어 ‘데이터 경제’라는 새로운 용어는 ‘데이터’와 ‘경제’로 나뉜다. ‘데이터’ embedding 시에 '경제'의 영향을 받아서 embedding이 생성되고, '경제'도 embedding 시에 '데이터'의 영향을 받아서 embedding 생성된다. 이렇게 되면 '데이터 경제'가 '데이터'와 '경제'로 따로 해석되지 않게 된다.

다만 이 의미가 '데이터 경제'라는 용어와는 다를 수 있는데, 이것은 결국 word 수를 늘려서 rare word를 가져갈지, 이를 쪼개서 학습 빈도를 높일지의 trade-off로 볼 수 있다. 만약 해당 워드 자체를 학습시킨다고 할 때, 그 단어의 사용 빈도가 낮다면 오히려 제대로 학습이 되지 않게 될 가능성도 높다. 하지만, 현재와 같이 subword로 나눈 후에 self-attention 방식으로 학습할 경우 '데이터' 만 있을 때와 '경제'만 있을 때, '데이터 경제'가 있을 때가 다 다르게 학습되므로 subword로 나누어도 의미가 완전히 해체되지는 않으며, 복합어의 의미를 어느 정도 살릴 수 있다.

**2) 감정 모델 세부 구조: 고용기사 분류 모델, 감정 분석 모델**

**○ 고용기사 분류 모델**

위와 같은 전처리 과정을 거친 뉴스 기사들은 <그림8>과 같이 고용 기사 분류 모델의 입력으로 들어가, 해당 뉴스 기사가 고용 관련 뉴스 기사인지를 판정한다. 이 모델은 입력값으로 기사를 넣으면 이 기사가 고용과 관련이 있을 확률이 몇 퍼센트인지, 관련이 없을 확률은 몇 퍼센트인지 출력값을 도출한다. 그 중 더 높은 확률을 가지는 것이 최종 결과가 되는데, 예를들면 고용과 관련이 있을 확률이 80%, 관련이 없을 확률이 20% 라면 해당 기사는 ‘고용률과 관련이 있다’고 판단하는 것이다.

이 모델은 KoELECTRA(한글 데이터로 사전훈련된 ELECTRA 모델)와 fully-connected layer를 연결한 방식을 활용해 구현하였다. 먼저 KoELECTRA는 대량의 데이터로 사전 학습이 충분히 된 상태에서 재학습(fine-tuning)을 통해 뉴스 기사가 고용 관련 기사인지 판단하기 위해 필요한 정보를 추출한다. 이렇게 추출한 정보(embedding)를 fully-connected layer와 softmax 함수를 통해 ‘고용 관련’ / ‘관련 없음’의 두 가지 클래스에 대한 확률 분포로 변환한다. Softmax 함수는 n개의 값이 주어졌을 때, 이 값들에 exponential을 취한 후에 normalize하여 n개 확률에 대한 categorical distribution으로 변환한다. 이를 분류 목적으로 사용할 때는 확률 분포에서 가장 값이 큰 클래스(argmax)를 해당 뉴스의 클래스로 판단하는 방식을 사용한다.

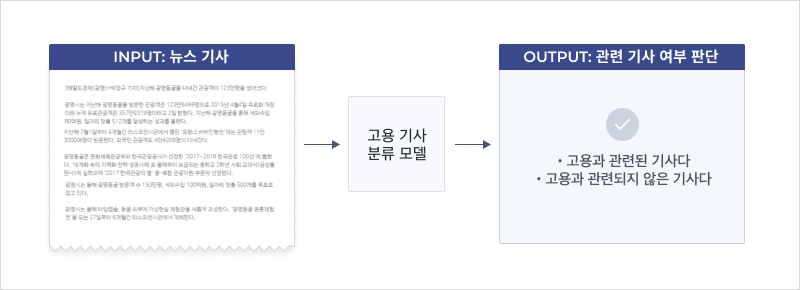


그림 8. 고용 기사 분류 모델의 입력과 출력

**○ 감정 분석 모델**

감정 분석 모델은 위의 고용 기사 분류 모델을 통해 ‘고용과 관련됐다’고 판단된 뉴스들에 대해 감정 지수를 뽑는다. 이때, 감정 지수는 <그림9>과 같이 고용 분류 모델과 비슷하게 새로운 KoELECTRA와 fully-connected layer와 softmax함수를 이용해, 부정/보통/긍정 3가지 클래스에 대한 확률 분포로 나타낸다. 감정 지수는 이후 고용률 예측 모델에서 확률 분포 자체를 사용하거나 뉴스의 감정 분석 정보(embedding)를 사용하고, 분류 목적으로 사용할 경우 위와 마찬가지로 확률 분포의 값이 가장 큰 클래스(argmax)를 해당 뉴스의 고용률에 대한 감정으로 판단한다. 감정 분석 모델도 위와 비슷하게 입력값으로 기사를 넣으면 각 기사마다 긍정/부정/판단불가에 대한 비율이 출력으로 나오고, 이에 대한 가장 높은 값을 최종 결과로 채택한다.

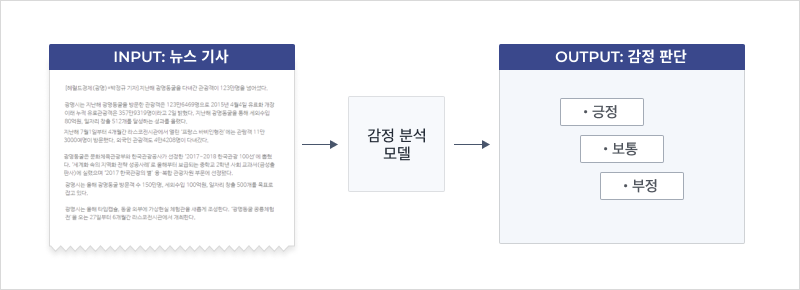


그림 9. 감정 분석 모델의 입력과 출력

고용기사 분류 모델과 감정 분석 모델은 사전에 고용 뉴스 데이터와 감정 분석용 데이터에서 학습을 진행한 pre-trained 모델이 사용된다. 각 모델의 학습 설정 및 실험과 관련된 내용은 뒤에서 자세히 설명하겠다.

**다. 고용률 예측 모델**

고용률 예측 모델은 기존 고용률 데이터와 앞서 언급된 감정 모델을 바탕으로 추출된 뉴스 기사의 감정 정보를 기반으로, 특정 날짜의 고용률을 예측하는 모델이다. 특정 시점(t)에서의 고용률을 그 이전의 고용률() 데이터와 해당 시점 근처의 뉴스()로부터 추출한 고용 감정 지수로부터 예측한다. 따라서, 장기적인 미래 예측(forecast)이라기보다, 뉴스 정보를 이용하여 현 상황에 대한 요약(nowcasting)을 하는 것에 가깝다고 볼 수 있다.

기존 지도 학습에서는 모델 학습을 위해 다음과 같은 3가지 상황을 가정한다. 1) IID 가정, 2) 학습/평가 데이터가 같은 분포에서 뽑혀야 함, 3) 시간이 변화해도 분포가 일정함 (stationary). 하지만 시계열 데이터는 데이터가 생성되는 분포가 달라질 수 있기 때문에 일반적인 지도 학습 모델이 아니라 시계열 예측을 위한 별도의 모델이 필요하게 된다.

시계열 예측 모델에는 Autoregressive model, Moving average model 등이 주로 활용된다. 이와 같은 모델은 기본적으로 다음 시간(p+1) 데이터의 변화 예측을 위한 정보가 ‘기존 p개 데이터에 모두 포함되어 있다’는 가정을 가진다 (p-th order Markov assumption). 하지만 고용률 데이터 특성 상, 기존 데이터뿐만 아니라, 정부 정책 등 외부 상황에 의해서도 직간접적인 영향을 많이 받기 때문에 이같은 가정이 적용되기 어렵다.

따라서 외부 정보를 반영하는 시계열 예측 방법이 필요하기 때문에, 당사는 이 점을 보완한 고용률 예측 모델을 제안하고자 한다. 이 모델은 감정 분석 모델에서 뽑아낸 뉴스 감정 정보(embedding)를 기존 고용률 데이터와 같이 이용하여 기존 시계열 예측 모델의 한계점을 극복하고자 하였다.

**1) 고용률 예측 모델 데이터 전처리**

**○ 뉴스 감정 정보 처리: 주성분 분석(PCA) 사용**

고용률 예측 모델에서는 뉴스 감정 정보와 기존 고용률 정보를 사용한다. 이 때, 뉴스 감정 정보는 예측 날짜로부터 일정 기간 이전까지의 뉴스들을 모아 예측에 활용하게 되는데, 앞의 감정 분석 모델에서 추출한 감정 정보(embedding)을 그대로 사용할 경우 해당 뉴스 정보의 차원이 너무 커져 현재 컴퓨팅 자원으로는 처리가 불가능하기 때문에 이를 줄여야 할 필요가 있었다.

이를 위해 **‘주성분 분석(Principal Component Analysis; PCA)’**방식을 적용해, 뉴스 감정 정보를 저차원으로 축소하여 그 정보를 줄이는 전처리 작업을 진행하였다. 정보의 차원을 축소하면 손실이 발생할 수 있으나, 이 손실 정도는 주성분(principal component)의 분산을 조사하는 것으로 확인이 가능하다. 본 프로젝트에서는 주성분 분석을 통한 차원 축소 후에도 기존 정보의 95% 정도가 남도록 축소된 차원 값(k)을 조절하였다.

다만, 주성분 분석을 통한 차원 축소를 할 경우 해당 기간의 뉴스 개수가 축소된 차원 값보다 크거나 같아야 한다는 제약이 존재한다. 따라서 예측하려는 날짜 중에서 해당 날짜로부터 W일 이내에 k개의 뉴스가 없는 경우는 데이터에서 제외하였다. 또한, 예측하려는 날짜에 따라 사용 가능한 뉴스의 개수가 달라질 수 있는데, 뉴스가 지나치게 많은 경우 역시 컴퓨팅 자원의 한계로 모든 뉴스를 다 사용할 수는 없기 때문에 예측 시에 사용되는 최대 뉴스의 개수를 제한하였다. 따라서 최종적으로 뉴스 감정 정보로는 W 기간 내의 최대 N 개의 뉴스를 사용하였다.

**○ 기존 고용률 정보 처리: 데이터 증강(augmentation) 활용**

고용률 정보는 학습에 사용되는 데이터인 동시에, 예측해야 하는 데이터로, 상당한 데이터의 양이 필요했지만 주어진 데이터의 양이 적고, 해당 데이터의 정보가 너무 적다는 어려움이 있었다. 따라서 데이터 증강(augmentation)을 통해 정보를 추가하였다.

먼저 추가한 정보로는 주기성을 부여한 timestamp 정보가 있다. timestamp 정보는 무척 유용한 정보이나, 그대로 사용하기에는 주기성이 명시적으로 드러나지 않는다는 문제점이 있다. 예를 들어, 2019년 1월 1일에 수집된 고용률 데이터는 2018, 2017, … n년의 1월 1일에 수집된 고용률 데이터와 연관이 강하지만, 2019-01-01과 2018-01-01에는 이런 관계가 나타나지 않는다. 따라서 timestamp 데이터를 sin, cos을 이용하여 1년 주기의 신호로 변환한 값을 만들고, 이를 모델의 입력에 추가하였다. 이를 통해 2019-01-01과 2018-01-01에서 같은 sin/cos 값이 모델에 입력되도록 하여 입력 수준에서 명시적으로 두 값의 관계성을 부여하였다. 더 나아가, 해당 날짜의 입력 값에 1년 전 같은 날짜의 고용률과 그 외 값들을 추가하여 1년 주기의 두 값을 연결하고자 하였다.

또한, 고용률 예측 모델은 예측하려는 날짜로부터 실제로 수집된 가장 가까운 기존 고용률 값을 사용하는데, (e.g. 2020/12/15에 대한 예측을 위해서 2020/12/1부터 L 길이만큼의 고용률 값을 가져옴) 이렇게 하면 실제 수집 기간 사이의 모든 기간에 대해 똑같은 고용률 정보들이 사용되게 된다. 이런 문제를 해결하고 예측하고자 하는 날짜가 마지막 수집 기간으로부터 얼마나 떨어진 날인지 정보를 명시적으로 부여하기 위해 예측 날짜와 가장 가까운 기존 고용률 값이 존재하는 날짜의 차이를 offset 정보로 부여하였다.

이 데이터들은 각각 다른 scale과 적은 정보량을 가지고 있어 모델을 쉽게 학습시킬 수 있도록 모든 데이터를 정규화하였다. 또한, 데이터 증강 외에, 고용률 데이터를 ‘일(日) 단위’로 보간(interpolation)하는 방식으로 데이터의 양이 적다는 문제를 해결하려고 하였다.

**○ 데이터 구성**

위와 같은 전처리를 거쳐서 최종적으로 특정 날짜(e.g. 2020/12/15)를 예측하기 위해 모델에 들어가는 실제 데이터는 다음과 같이 구성하였다.

|  |
| --- |
| **A. 뉴스 감정 정보 vector**   * 해당 날짜를 포함하여 그로부터 W 날짜만큼의 PCA vector에서 최대 N개 (e.g. 2020/12/13~2020/12/15의 PCA vector에서 최대 128개) * k \* N 차원 vector (k 차원 PCA \* W 기간의 최대 뉴스 기사 개수 N)   **B. 고용률 vector:**   * 해당 날짜로부터 가장 가까운 기존 고용률 값을 찾고 그로부터 L 날짜만큼의 고용률 정보를 모은 vector (e.g. 2019/12/1~2020/12/1) * 5 \* L 차원 vector: (정규화 된 고용률 값 + year sin + year cos + 정규화 된 이전 년도의 고용률 값 + offset) \* L |

**2) 고용률 예측 모델 구조**

고용률 예측 모델은 기존 고용률 데이터로부터 확장된 정보들과 뉴스의 감정 정보를 결합하여 다음 고용률을 예측한다. 따라서 **고용률 정보 결합 모델**과 **뉴스 감정 정보 결합 모델**을 하위로 가진다.

먼저 고용률 정보 결합 모델은 self-attention 특성을 가진 transformer-encoder 모델을 사용하였다. self-attention 구조는 여러 개의 입력들이 주어졌을 때, 입력들을 유사도를 기반으로 '섞어서' 새로운 정보(embedding)를 만들어내는 특징이 있다. 또한 이렇게 학습된 딥러닝 모델은 예측에 큰 영향을 주는 입력에 가중치를 더 주도록 한다. 해당 모델은 목표 날짜로부터 가장 가까운 고용률 데이터로부터 정해진 기간(L) 만큼의 고용률 정보()를 입력으로 받고, 이를 transformer-encoder 모델에 통과시킨 후 최종적으로 L개의 고용률 정보 embedding을 집합(aggregation)하는 단계를 거치게 된다. 이 때, aggregation의 방식으로 결합(concatenation)을 사용하였으며, 정규화된 값으로 변환하는 ‘batchnorm’ 단계를 거쳐 해당 기간(L) 고용률의 대표 정보값을 도출한다.

일반적으로 deep learning에서 고용률 정보와 같이 시간 순서가 있는 데이터를 처리하기 위해서는 RNN 계열 모델들(e.g. LSTM, GRU)을 주로 사용한다. 하지만, 실제 RNN을 이용하여 기존 고용률 정보를 모았을 때, 성능이 예상보다 좋지 않아 transformer 모델을 사용하게 되었다. 예상보다 RNN 성능이 저조했던 이유는 ‘입력 길이’ 때문인 것으로 판단된다. 현재 고용률 예측을 위해서는 지난 1년치의 고용률 값을 입력해야 현재 예측할 날짜와 가까운 정보를 얻을 수 있는데 (e.g. 2020년 6월 1일 부근의 정보를 알기 위해서는 2019년 6월 1일 부근의 정보가 필요함), RNN은 이를 ‘일 단위’로 보간하다 보니 입력 길이가 너무 길어져 1년 전의 정보가 많이 희석되게 되었다. 반면, transformer 모듈은 길이가 긴 입력에 대해서도 전체 시간에 바로 접근 가능하기 때문에 오래된 과거의 입력 값이라도 그 정보의 의미가 유지되어 좋은 성능을 보인 것으로 생각된다.

또한, 뉴스 감정 정보 결합 모델에도 마찬가지로 self-attention 특성을 가지고 있는 transformer-encoder 모델을 사용하였다. transformer-encoder 모델은 목표 날짜를 포함하여 일정 기간(window) 동안 수집된 뉴스 기사로부터 추출된 정보에 PCA를 적용한 값()을 입력으로 받고, 여기에서 나온 정보를 고용률 정보 결합 모델과 같은 방식으로 결합하여 하나의 뉴스 감정 대표 embedding을 만든다.

각 모델에서 나온 두 embedding은 결합된 후, batch normalization layer를 통해 정규화된 값으로 변환한다. 그 후에 뉴스의 고용률 감정에 대한 대표 embedding과 과거 고용률에 대한 embedding을 합쳐 고용 감정-고용률 embedding을 만들고, 해당 embedding은 몇 개의 fully-connected layer와 batch normalization layer들, relu를 통과해서 최종적으로 예측할 고용률 값이 된다. 이때 고용률 데이터의 값이 변하는 경우와 이전 데이터에서 별다른 변화가 없는 경우들이 있다는 것을 감안하여 단순한 변화와 복잡한 변화를 함께 학습할 수 있는 residual connection을 사용하였다. 또한, residual connection은 모델의 과적합을 막아주므로 이번과 같이 데이터가 적은 경우 학습의 효율을 올려준다. <그림10>은 고용률 예측 모델의 전체적인 구조와 입/출력 값을 보여준다.

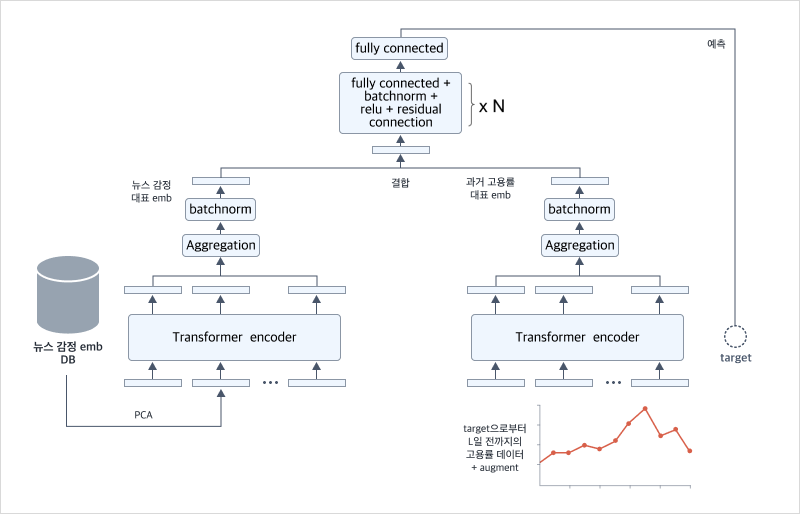


그림 10. 고용률 예측 모델의 구조 및 입력과 출력

**3. 실험 결과**

**가. 감정 모델**

먼저 기사 분류 모델과 감정 분석 모델의 개발을 위한 설계 작업을 진행했으며, 효과적으로 학습을 진행할 수 있도록 학습 및 평가 데이터로 사용될 고용 관련 기사의 전처리 작업을 수행했다. 이 데이터를 활용하여 가장 성능이 좋은 모델을 찾기 위해 앞서 언급했던 BERT 및 파생 모델 5가지에 대한 학습을 진행하여 최적의 성능을 내는 모델을 개발하였다. 자세한 개발 과정은 아래와 같다.

일반적으로 BERT를 이용할 때는 BERT의 크기가 매우 크고 학습 시간이 오래 걸리기 때문에 사전훈련된 모델을 활용하여 사용자가 원하는 분야에 대해 추가 학습을 진행한다. 당사 또한, 이번 모델을 개발하기 위해 총 5개의 pre-trained 모델을 활용하였고, 아래와 같은 조건 상의 비교학습을 진행하여 성능이 제일 우수한 모델을 찾아 내었다.

실험 대상은 앞서 언급한 BERT와 파생 모델인 DistilBERT, ELECTRA를 한글 데이터로 사전 학습시킨 모델로, **KoBERT, SKT KoBERT, DistilKoBERT, KoELECTRA, KoCharELECTRA**를 활용하였다. 먼저 KoBERT와 SKT KoBERT는 모두 BERT 모델이지만, 학습 데이터나 파라미터를 다르게 학습시킨 모델이다. DistilKoBERT, KoELECTRA, KoCharELECTRA는 BERT 기반 모델로, dDistilKoBERT는 DistilBERT모델을 한글 데이터로 학습시킨 모델이며, KoELECTRA와 KoCharELECTRA는 ELECTRA 모델을 사용했지만, 각각 데이터를 형태소와 음절 단위로 나누어 학습시켰다는 점에서 차이가 있다.

**1) 데이터 처리**

감정 모델의 학습과 평가를 위한 데이터는 통계청에서 수집하여 수작업으로 레이블링된 뉴스 데이터를 제공받았다. 고용 기사 분류 모델에서는 주어진 101,565개의 기사 데이터 중 81,252개에 해당하는 데이터를 학습에 활용하였고, 나머지 20,313개 데이터는 성능 분석에 활용하여 검증 과정을 거쳤다. 또한, 감성 분석 모델에서는 총 7,519개의 기사 중 6,015개의 기사를 활용하여 학습을 진행하였고, 이후 1,504개의 기사를 바탕으로 모델의 성능을 분석하였다.

먼저 고용 기사 분류 모델에서는 두가지 정답만 존재하도록 했다. (①고용과 관련된 기사다, ②고용과 관련되지 않은 기사다) 하지만, 고용 관련 기사에 대한 감성을 분석할 때에는 ‘감성'이란 특성 상 정답이 확실히 구분되기 어렵기 때문에 정답 수를 확정짓지 않았다. 정답을 3종류(부정적, 보통, 긍정적)로 나눌 수도 있고, 5종류(매우 부정적, 부정적, 보통, 긍정적, 매우 긍정적), 아니면 더 세부적으로 나눌 수도 있기 때문에 우선 정답을 몇 가지로 나누는 것이 효과적일지 확인해 보고자 했다. 이를 위해 정답을 3종류, 5종류로 나눈 데이터로 각각 실험을 진행했다.

**2) 결과 비교**

아래 <표1, 2, 3>은 학습을 진행한 총 5개의 BERT 기반 모델의 성능 결과다. 아래 표를 보면, 각 모델들에 대한 정확도(acc)와 f1 score가 명시되어 있다. 정확도는 정확히 예측된 데이터의 수를 전체 데이터의 수로 나눈 값으로, 모델의 성능을 가장 직관적으로 나타낼 수 있는 평가 지표다. 하지만, 이는 데이터 자체의 편향(bias)을 고려하지 않는다는 문제가 있다. 예를 들어 고용 기사 분류 모델 학습에 사용되는 데이터 전체의 90%가 고용에 관련되어 있고, 그 나머지 10%가 고용과 관련되지 않은 기사라면 데이터 자체가 더 많은 고용 기사를 분류하는 성능은 높게 나오지만, 고용과 관련되지 않은 기사를 분류하는 성능은 낮게 나올 수도 있다.

바로 이 데이터의 불균형을 보완하기 위한 지표가 **f1 score**이다. f1은 데이터 구조가 불균형할 때, 모델의 성능을 정확하게 평가할 수 있는 지표다. 모델들을 학습하기 위해 사용한 데이터에 편향성이 존재하기 때문에 모델들의 성능을 판단할 때 f1 score를 이용했다. 아래 표를 보면 알 수 있듯이, KoELECTRA가 세 가지 경우에서 모두 좋은 성능을 보였다. 또한, 이 모델을 기사 감정 분석 모델로 사용했을 때는 정답을 5개로 했을 때, 3개로 했을 때 모두 압도적으로 좋은 성능을 보였다. 그리고 고용 기사 분류 모델로 사용했을 때에도 제일 성능이 뛰어난 SKT KoBert와 비교해도 f1 score가 0.002밖에 차이가 나지 않았다.

이러한 이유로, KoELECTRA를 최종 모델로 사용하기로 결정했다. 추가적으로 감정분석 모델의 성능을 비교했을 때, 정답을 3분류로 나눴을 때의 성능이 전반적으로 더 좋았기 때문에 정답 수 또한 3가지로 나눈 데이터를 사용하기로 결정하였다.

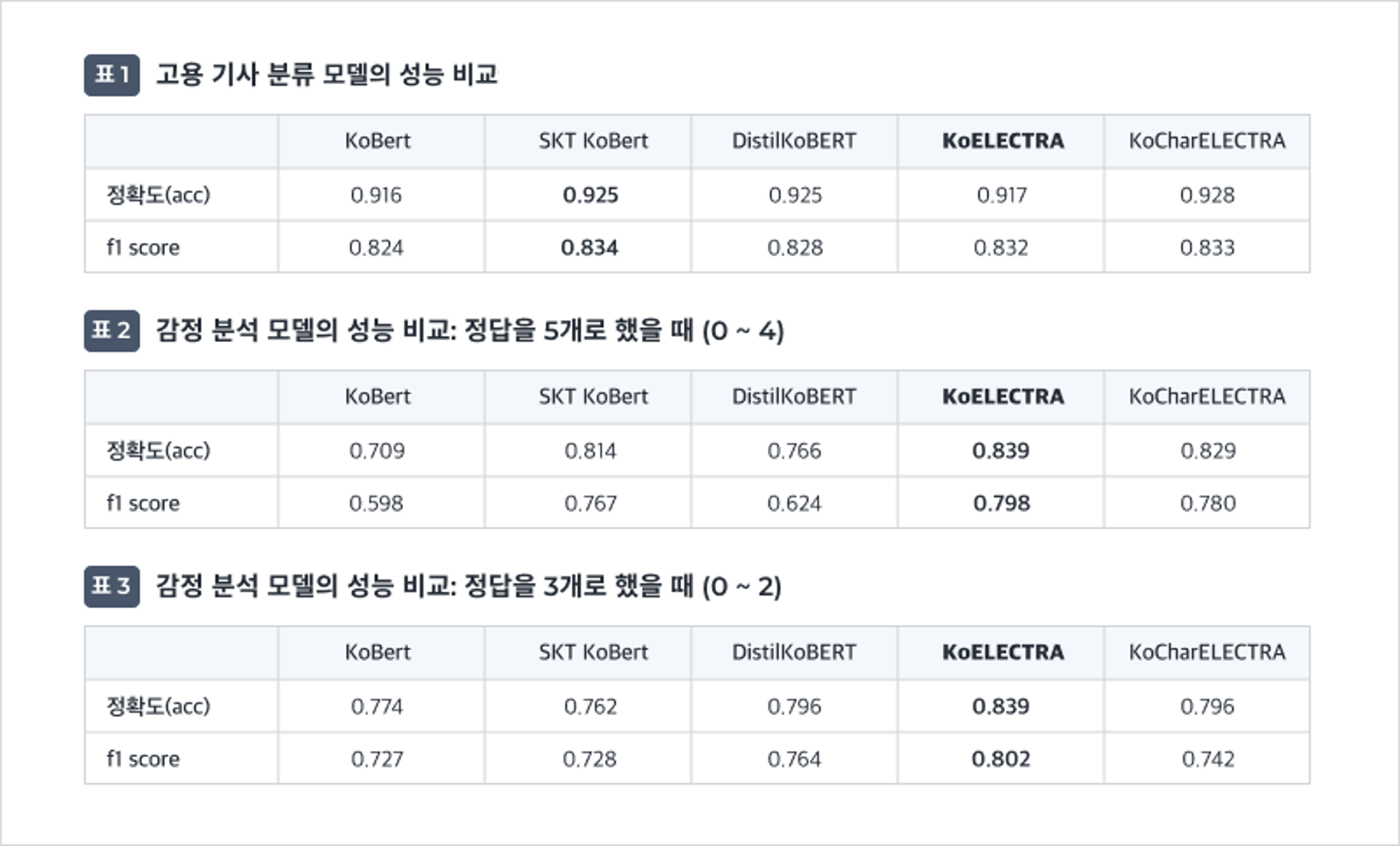


표 1, 2, 3. 감정모델 성능 비교

더불어 기사 분류 모델과 감정 분석 모델의 성능을 비교해보면 고용기사 판단에서는 대체로 우수한 성과를 내는 것을 볼 수 있지만, 감정분석 모델의 성능은 이에 반해 다소 낮은 수치를 보인다. 여기에는 여러가지 이유가 있지만, 우선 투입한 데이터 양에서 큰 차이가 발생했기 때문에 다소 성능의 차이가 있는 것으로 분석된다. 또한, ‘감정' 데이터 특성 상 정확하게 답변이 나뉘는 것이 아니라, 데이터를 작업하는 사람의 주관이 들어가기 때문에 오류가 발생할 여지가 있다. 예를 들어 누군가는 해당 기사에 대해 부정적으로 볼 수 있지만, 다른 누군가는 긍정도 부정도 아니라고 판단할 수 있기 때문이다. 하지만 해당 문제는 학습 데이터의 양이 많아진다면 긍/부정을 판단할 수 있는 기사의 수도 많아지기 때문에 해결될 수 있을 것으로 기대된다. 향후 다양한 데이터가 주어진다면 현재 수준보다 성능을 더욱 높일 수 있을 것이다.

**나. 고용률 예측 모델**

**1) 데이터 처리**

모델의 학습 및 예측을 위한 고용률 데이터는 통계청의 고용률(시도)[[1]](#footnote-1) 데이터에서 전체 합계 고용률을 사용하였다. train / test 데이터는 고용률 데이터와 뉴스 데이터가 같이 존재하며 전처리 등에 문제가 없도록 특정 기간을 설정해, 2016/1/1 ~ 2020/9/2 까지의 고용률 데이터를 예측 대상으로 삼았다. data의 기간을 섞으면서 정보가 섞이는 일이 없도록 데이터 자체를 섞지 않고, 전체 기간을 7:3으로 나누어 train / test 데이터를 구성하였다. 데이터의 양은 train 1185개, test 508개이다. 전처리로는 위에서 이야기한 것과 같이 데이터의 특징(feature)에 대한 증강(augmentation) 기법을 활용하여 데이터의 정보를 늘렸고, 기간에 대해 보간을 하여 데이터의 양을 늘렸다.

학습(train) 데이터의 실제 고용률로는 2016/1/1 데이터부터 2019/4/10까지의 데이터를 사용하였고, 해당 기간의 예측에는 2년 전까지의 고용률 데이터가 사용되었다. 예를 들어 2016/1/1의 고용률을 예측하기 위해서는 2014/12/1 ~ 2015/12/1까지의 고용률 데이터와 해당 데이터를 augment 한 데이터를 사용하는 방식이다. 마찬가지로 2019/4/10의 데이터를 예측하는데는 2018/4/1부터 2019/4/1일까지의 고용률 데이터와 해당 데이터를 augment 한 데이터를 사용했다.

테스트(test) 데이터는 2019/4/11부터 2020/9/2까지의 고용률 데이터를 사용하였고, 해당 기간의 예측에는 마찬가지로 2018/4/1부터 2019/4/1까지의 고용률 데이터와 augment 된 데이터를 사용하였으며, 2020/9/2의 예측에는 2019/9/2부터 2020/9/2까지의 데이터와 augment 된 데이터를 사용하였다.

**2) 결과 비교**

결과 비교는 테스트 데이터를 대상으로 실제 고용률과 예측 고용률 간의 상관계수(Pearson correlation coefficient)를 측정하는 방식으로 이루어졌다.

고용률 데이터의 경우 데이터의 양이 적은 만큼 모델의 학습에 따른 성능 변화가 있을 수 있으므로 10번의 학습을 진행하여 테스트 기간(2019/4/11 ~ 2020/9/2)에 대한 고용률 10개를 예측했다. 이 값들의 평균을 내어 실제 고용률과의 상관계수를 측정하였고, <그림11>과 같은 결과를 얻었다. 이 때, 실제 고용률과 예측 고용률 간의 상관계수(Pearson correlation coefficient)는 **0.5919** 정도의 양의 상관관계를 보여주고 있다.

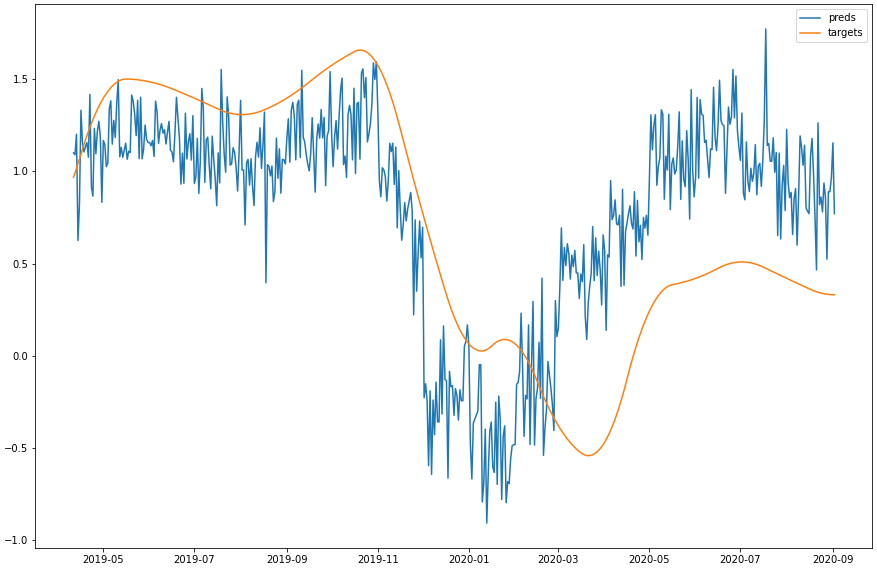


그림 11. 테스트 데이터 기간(2019/4/11 ~ 2020/9/2)의 고용률과 예측 고용률 값

또한, <그림12>는 전체적인 고용률의 경향성을 한눈에 보기 쉽게 전체 학습 기간과 테스트 기간의 고용률 값을 같이 그린 것이다. 해당 그림을 보면 고용률 예측 값이 학습 기간에서 반복되는 패턴(감소→증가)을 잘 학습하였고, 전체적인 고용률의 증가 추세도 잘 반영하고 있음을 볼 수 있다.

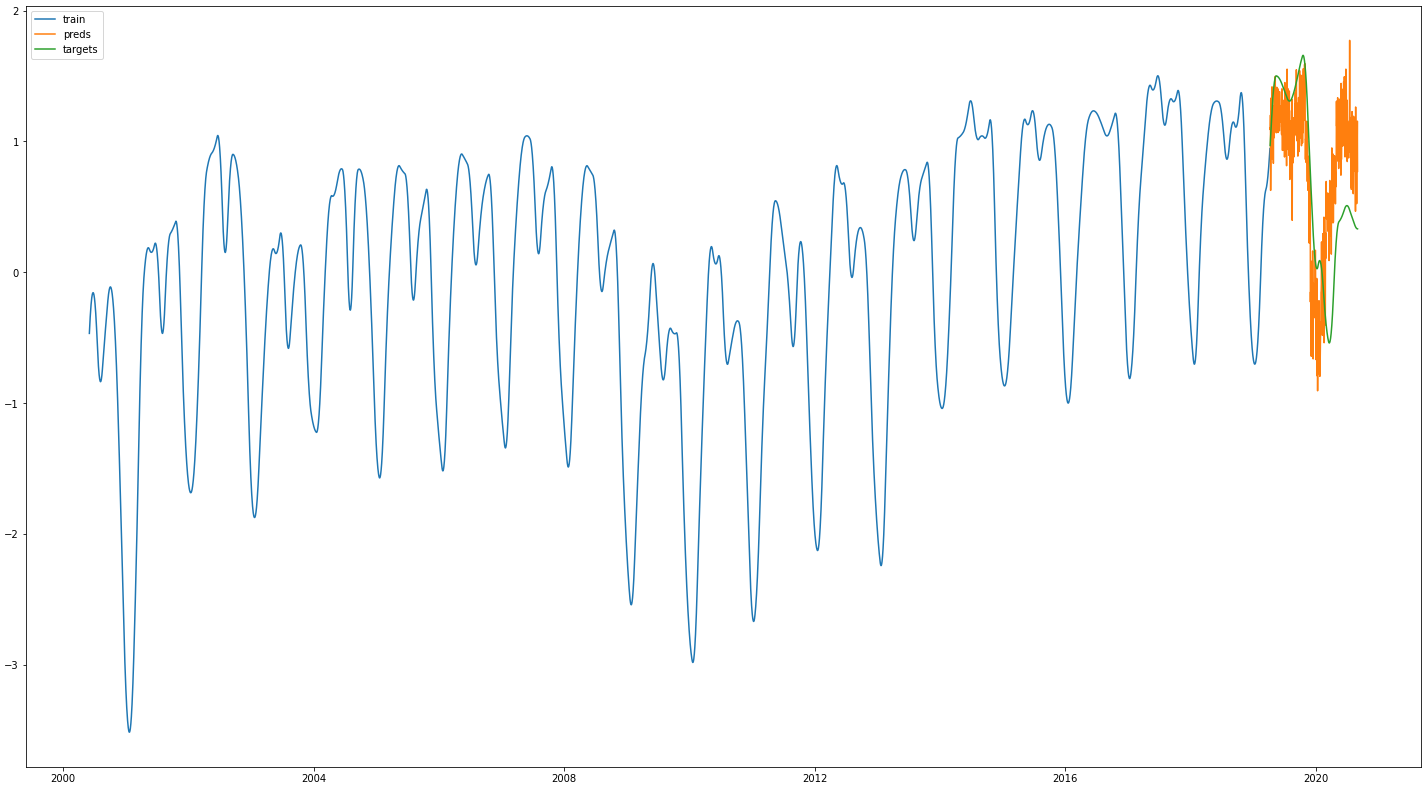


그림 12. 전체 기간(2016/1/1~2020/9/2)의 고용률과 예측 고용률 값(주황색 부분)

다만 실제 고용률과 예측 고용률의 결과 그래프를 보았을 때, 2019년의 고용률 예측 값은 비교적 실제 고용률과 비슷한 형태를 보이지만, 코로나 기간에 해당하는 2020년 1월부터는 예측한 고용률 값과 실제 고용률 값 간의 상당한 차이가 나타나고 있다. 이는 해당 기간 예측한 고용률 값이 높은데 비해 실제 고용률 값이 낮게 나온 것으로 보아, ‘코로나19’와 같이 예상치 못한 이벤트가 발생할 경우 예측 정확도에 많은 영향을 받는 것으로 분석된다.

코로나와 같이 특수한 사건이 발생하지 않은 상황에서의 모델의 성능을 확인하기 위해 코로나 발생 이전의 특정 기간을 설정해 추가적인 실험을 진행하였고, <그림13>과 같은 결과를 얻었다. 2019/4/11부터 2019/12/31까지를 대상으로 실제 고용률과 예측 고용률 값의 상관계수를 보았을 때, **0.9304** 정도의 높은 양의 상관관계가 나타나는 것을 확인하였다. 이를 통해 일반적인 상황의 경우, 당사가 제안하는 모델을 고용률에 대한 단기 예측 용도로 사용할 수 있다고 판단된다.

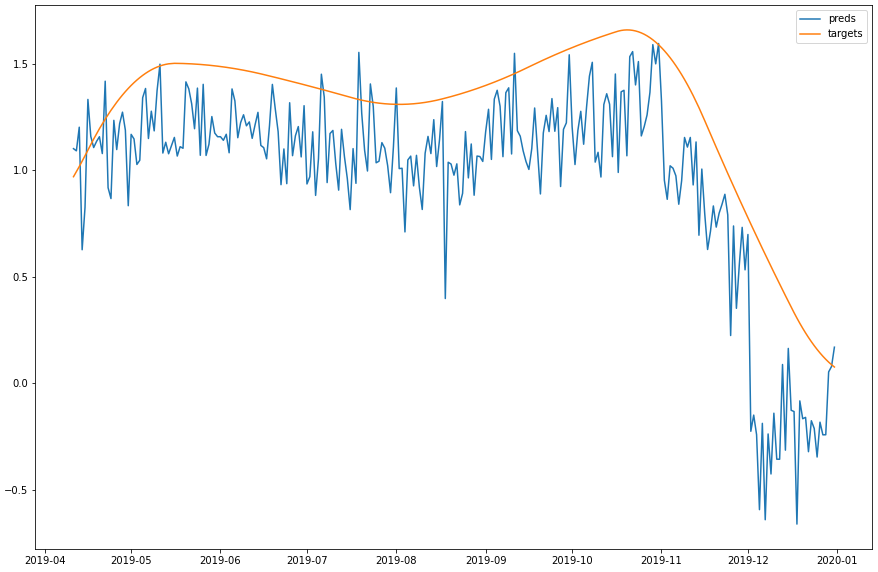
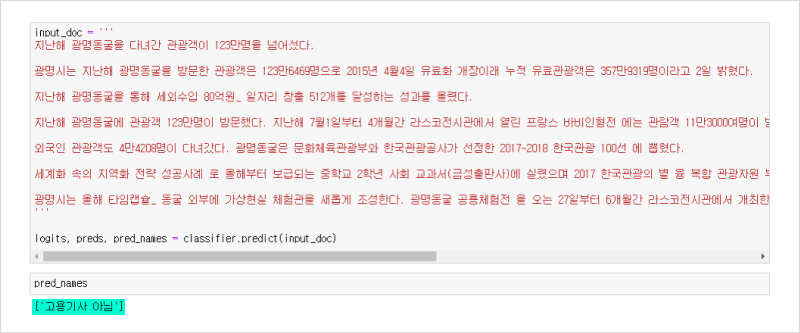


그림 13. 테스트 데이터에서 코로나 기간을 제외한 기간(2019/4/11 ~ 2019/12/31)의 고용률과 예측 고용률 값

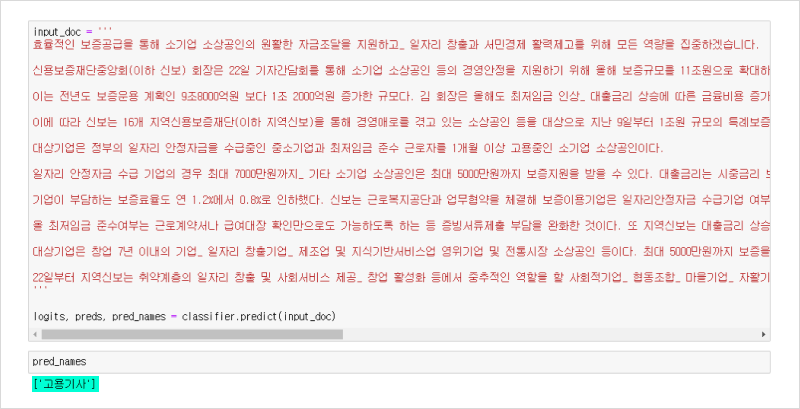
**4. 최종 개발 결과물**

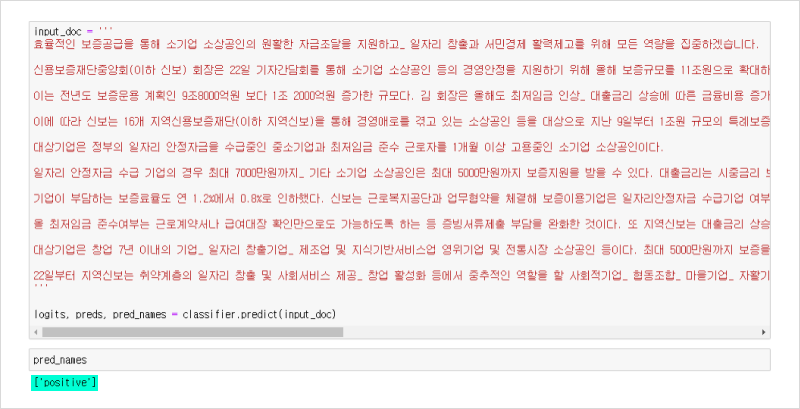
**가. 기사 분류 모델 결과물 : “고용기사가 아님”을 판단**



**나. 감정 분석 모델**

**1) “고용기사이면서 감정지수를 긍정”으로 분석**

****



**2) “고용기사이면서 감정지수를 부정”으로 분석**

****

**다. 고용률 예측 모델**

그림 11, 12, 13에 해당하므로, p17~18 참고

**※ 참고 문헌**

* Neural Machine Translation of Rare Words with Subword Units ([링크](https://www.aclweb.org/anthology/P16-1162.pdf))
* BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding ([링크](https://arxiv.org/pdf/1810.04805.pdf))
* ELECTRA: PRE-TRAINING TEXT ENCODERS AS DISCRIMINATORS RATHER THAN GENERATORS ([링크](https://openreview.net/pdf?id=r1xMH1BtvB))

1. 통계청 홈페이지 참고

   https://kosis.kr/statHtml/statHtml.do?orgId=101&tblId=INH\_1DA7014S\_03&conn\_path=I3 [↑](#footnote-ref-1)