OPEN ACCESS

사전학습 된 언어 모델 기반의 양방향 게이트 순환 유닛 모델과 조건부 랜덤 필드 모델을 이용한 참고문헌 메타데이터 인식 연구*

A Study on Recognition of Citation Metadata using Bidirectional GRU-CRF Model based on Pre-trained Language Model

지선영 (Seon-yeong Ji)**
최성필 (Sung-pil Choi)***

초 록

본 연구에서는 사전학습 된 언어 모델을 기반으로 양방향 게이트 순환 유닛 모델과 조건부 랜덤 필드 모델을 활용하여 참고문헌을 구성하는 메타데이터를 자동으로 인식하기 위한 연구를 진행하였다. 실험 집단은 2018년에 발행된 학술지 40종을 대상으로 수집한 PDF 형식의 학술문헌 53,562건을 규칙 기반으로 분석하여 추출한 참고문헌 161,315개이다. 실험 집합을 구축하기 위하여 PDF 형식의 학술 문헌에서 참고문헌을 분석하여 참고문헌의 메타데이터를 자동으로 추출하는 연구를 함께 진행하였다. 본 연구를 통하여 가장 높은 성능을 나타낸 언어 모델을 파악하였으며 해당 모델을 대상으로 추가 실험을 진행하여 학습 집합의 규모에 따른 인식 성능을 비교하고 마지막으로 메타데이터별 성능을 확인하였다.

ABSTRACT

This study applied reference metadata recognition using bidirectional GRU-CRF model based on pre-trained language model. The experimental group consists of 161,315 references extracted by 53,562 academic documents in PDF format collected from 40 journals published in 2018 based on rules. In order to construct an experiment set. This study was conducted to automatically extract the references from academic literature in PDF format. Through this study, the language model with the highest performance was identified, and additional experiments were conducted on the model to compare the recognition performance according to the size of the training set. Finally, the performance of each metadata was confirmed.

키워드: 참고문헌 메타데이터 인식, 텍스트 마이닝, 심층학습, 언어모델
reference metadata recognition, text mining, deep learning, language model

* 이 논문은 2021년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. 2018R1D1A1B07048839).
** 경기대학교 일반대학원 문헌정보학과 석사과정(jissuy@kgu.ac.kr) (제1저자)
*** 경기대학교 문헌정보학과 부교수(spchoi@kgu.ac.kr) (교신저자)
■ 논문접수일자 : 2021년 2월 25일 ■ 최초심사일자 : 2021년 3월 6일 ■ 게재확정일자 : 2021년 3월 17일
■ 정보관리학회지, 38(1), 221-242, 2021. http://dx.doi.org/10.3743/KOSIM.2021.38.1.221

※ Copyright © 2021 Korean Society for Information Management
This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 (https://creativecommons.org/licenses/by-nc-nd/4.0/) which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.

1. 서 론

참고문헌은 학술문헌을 구성하는 가장 기본적인 요소 중 하나이다. 작게는 저자가 연구를 진행하며 참고한 문헌을 인용한 것으로 제목, 저자명, 출판년도 등 학술문헌의 메타데이터 정보를 포함하는 개념이며 크게는 연구 주제와 관련된 연구 또는 선행 연구를 효과적으로 탐색할 수 있는 수단이 된다. 특히 참고문헌은 연구자의 연구 설계 단계에서 주제 분야의 최신 연구 동향을 파악하고, 연구 주제의 유용성을 검증하기 위한 탐색에 많이 활용된다(임수현 외, 2019). 또한 참고문헌은 본문 내에서의 지식 정보 추출뿐만 아니라 논문 사이의 인용-피인용 관계 연결 및 분석 정보로 유용하게 활용될 수 있다(김재훈 외, 2019; Powely & Dale, 2007). 특히 학술문헌의 메타데이터와 참고문헌의 메타데이터를 활용하여 인용-피인용 논문을 목록화한 인용 색인(Citation Index)은 인용문헌의 유용성 및 중요도를 파악할 수 있는 질적 기준이 될 뿐만 아니라 연구 경향 분석 및 새로운 연구 분야 확인에 이용되는 중요한 기법이다(김지훈, 2003).

그러나 매년 출판되는 학술문헌의 규모가 꾸준히 증가함에 따라 수작업으로 참고문헌에서 메타데이터를 추출하여 인용 색인 시스템 및 참고문헌 데이터베이스에 입력하는 것에는 시간 및 인력적인 한계가 증가하였고, 이를 해결하기 위해 관련 연구가 꾸준히 진행되어 왔다. 초기 연구는 참고문헌 메타데이터 인식과 관련된 연구는 인용 형식을 분석하여 규칙 기반으로 추출하는 템플릿 마이닝(Template Mining) 기반의 연구가 대부분이었다(이상기 외, 2017; Besagni & Belaid, 2004). 그러나 참고문헌은 템플릿에 지정된 주요 메타데이터 이외에도 다양한 메타데이터가 포함될 수 있으며 학술문헌, 단행본, 기술보고서, 웹 데이터 등의 참고문헌 유형 및 APA, MLA 등의 다양한 인용 유형이 있기 때문에 이러한 모든 경우의 수를 포함하여 템플릿을 만드는 과정에서 많은 어려움이 발생하였다(An et al., 2017). 이후 자연어 처리 분야에서 심층 학습(Deep Learning)을 적용하는 것이 정보 추출 및 텍스트 마이닝(Text Mining) 분야에서 기존의 방법론들 보다 높은 성능을 보이면서 참고문헌 인식 분야에서도 심층 학습을 적용한 연구가 진행되었다. 특히 참고문헌 메타데이터 인식은 참고문헌을 구성하는 각각의 문자 또는 단어에 해당하는 레이블(Label)을 알맞게 분류하는 연속적 레이블링(Sequential Labeling)에 해당된다고 볼 수 있기 때문에 국내외에서 참고문헌 메타데이터 추출을 위해 기계학습 및 심층학습을 적용한 연구가 진행된 바 있다(김선우 외, 2018; 신규민 외, 2009; Tkaczyk, Szostek, & Fedoryszak, 2015; An et al., 2017).

최근 자연어 처리 분야에서 높은 성능을 보이고 있는 모델은 사전 학습된 언어 모델(Pre-trained Language Model)이다. 사전 학습된 언어 모델은 문맥적 임베딩(Contextual embedding) 표현이 가능한 모델로 입력된 문장의 문맥적 의미가 포함된 임베딩 벡터를 출력한다. 모델 학습 과정에서 대규모 범용 말뭉치가 사용되었기 때문에 폭 넓은 분야에 적용 가능한 것이 특징이다(Devlin, Chang, Lee, & Toutanova, 2018). 사전 학습된 언어 모델 중 대표적인 것은 구글에서 개발한 BERT(Bidirectional Encoder

Representation from Transformer)(DevlinJacob et al., 2018)이다. 그러나 BERT는 영어를 중심으로 사전 학습된 언어 모델이기 때문에 한국어 데이터에 적용할 경우 성능에 한계가 있어(SKTBrain, 2019) BERT를 기반으로 한국어 처리 성능 향상을 위해 개발된 BERT의 파생 모델이 배포되고 있다. 본 연구에서는 BERT를 비롯하여 KoBERT(SKTBrain, 2019), HanBERT(tbai, 2019), KoELECTRA(Park, 2020) 등 총 4가지 언어 모델을 활용하여 참고문헌 메타데이터 인식 모델 성능을 가장 효과적으로 향상시킨 모델을 확인하였다.

본 연구는 사전 학습된 언어 모델을 기반으로 연속적 레이블링 방법에 대해서 우수한 성능을 보이는 심층 학습 모델인 양방향 게이트 순환 유닛 모델(Bidirectional Gate Recurrent Unit Model)과 기계 학습 기법 중 하나인 조건부 랜덤 필드(Conditional Random Field, CRF)를 활용하여 참고문헌 메타데이터 추출 성능을 효과적으로 높이고자 하였으며 PDF 형식의 학술 문헌에서 참고문헌을 규칙 기반으로 추출하여 참고문헌 메타데이터 학습 집합 구축 과정을 단축시키고자 하였다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 참고문헌 메타데이터 인식과 관련된 선행 연구들에 대하여 설명하고, 3장에서는 참고문헌 메타데이터 인식을 위해 사용한 사전 학습된 언어 모델 및 심층 학습 모델을 설명한다. 4장에서는 실험 집합 구축 과정을 비롯하여 실험 집합을 활용한 사전 학습된 언어 모델 별 실험 계획을 설명하고 가장 높은 성능을 도출한 언어 모델을 대상으로 실험 집합 규모에 따른 성능 차이에 대한 실험 및 그 결과를 설명한다. 5장에서는 본 논문의 결론을 설명한다.

2. 관련 연구

참고문헌에서 제목, 출판년도, 학술지명 등의 메타데이터를 추출하는 연구는 국내외에서 템플릿 마이닝 기법을 중심으로 진행되어 왔다. 이상기 외(2007)는 각종 학술지의 참고문헌 양식을 파악하여 규칙화 하거나 개체명 사전을 활용하여 참고문헌 메타데이터를 추출하는 연구를 진행하였다. Besagni와 Belaid(2004)는 품사 태거와 'In', 'pp' 등 참고문헌에서 주로 등장하는 토큰을 구별하는 태그 16종, 추출하고자 하는 참고문헌 정보 6종(저자, 출판년도, 학술지명, 페이지, 제목, 권)에 대한 특징을 적용하여 메타데이터를 추출하는 연구를 진행하였다. 김지훈(2003)은 학술 논문의 참조연결 시스템을 위해 템플릿 마이닝 기법을 사용하여 추출하고자 하는 참고문헌 메타데이터의 학술논문 내 위치, 폰트, 구두점, 레이아웃 등의 정보를 활용하여 참고문헌의 메타데이터를 추출하였다.

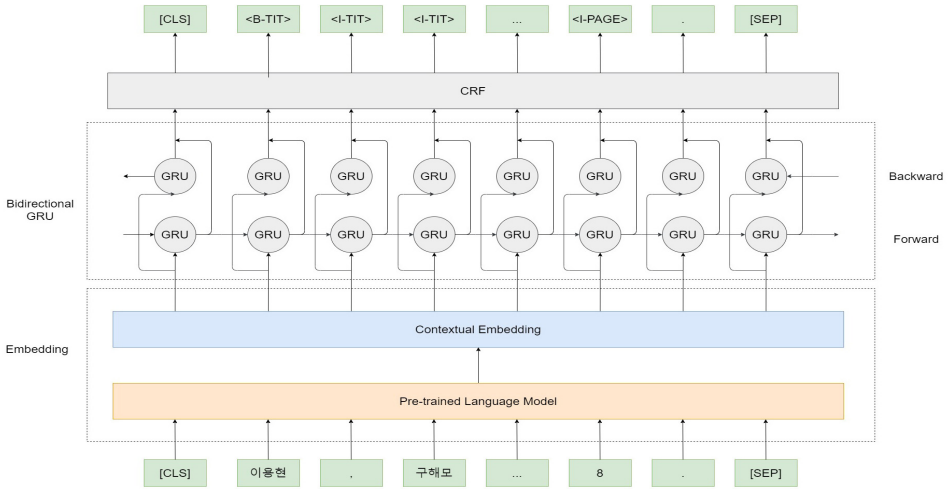
그러나 최근 텍스트 마이닝 기법이 다양한 분야에서 높은 성과를 거두며 텍스트 마이닝 기법을 적용한 연구도 함께 진행되고 있다. 신규민 외(2009)는 기계학습 기법인 조건부 랜덤 필드를 활용하여 자동으로 참고문헌에서 저자, 제목, 저널 또는 출판사항, 출판사 등의 메타데이터를 추출하는 연구를 진행하였다. Tkaczyk et al.(2015)는 조건부 랜덤 필드를 활용하여 과학 분야의 PDF 논문에서 학술 논문의 메타데이터, 참고문헌 및 참고문헌의 메타데이터, 원문을 추출하고 추출 결과를 온라인 및 XML

문서로 구조화하여 제공하는 라이브러리를 개발하였다. An et al.(2017)은 심층 학습 기법 중 하나인 세그먼트 시퀀스 레이블링(Segment Sequence Labeling) 기법을 적용하여 참고문헌 메타데이터를 추출하는 연구를 진행하였다. 이는 단어 단계에서 예측하는 것이 아니라 입력 시퀀스를 세그먼트(참고문헌) 단위로 나눈 다음, 세그먼트의 특징 값을 계산하여 세그먼트의 레이블 시퀀스를 유추하는 방법을 사용하였다. 김선우 외(2018)는 심층 학습 기법 중 하나인 양방향 게이트 순환 유닛 모델에 조건부 랜덤 필드 모델을 적용하여 참고문헌 메타데이터를 추출하였으며 추출 가능한 메타데이터는 제목, 저자명, 학술지명, 출판년도 등 8종이다.

관련 연구에서 확인할 수 있듯 국내의 경우 PDF 형식의 학술문헌에서 참고문헌과 그 메타데이터를 인식 및 추출하는 연구는 매우 저조하며 한국어 참고문헌에서 인식 가능한 메타데이터 역시 제목, 저자명, 페이지와 같은 기초적인 것이 대부분이다. 또한 기계 학습 및 심층 학습 모델에서 학습 데이터로 활용할 만큼 공개된 학습 집합의 양이 충분하지 않다. 따라서 심층 학습 및 자연어 처리 분야에서 높은 성능을 보이고 있는 언어 모델을 적용한 참고문헌 메타데이터 인식 연구가 부족한 상황이다.

3. 참고문헌 메타데이터 인식 모델

본 연구에서는 참고문헌 메타데이터 인식 모델의 성능을 향상시키기 위하여 사전 학습된 언어 모델을 기반으로 연속적 레이블링 방법에 대하여 우수한 성능을 보이는 심층 학습 모델인 양방향 게이트 순환 유닛 모델과 조건부 랜덤 필드 모델을 활용하여 참고문헌 메타데이터 추출 성능을 효과적으로 높이고자 하였다. 모델의 전체적인 구조는 〈그림 1〉과 같으며 본 장에서는 각 모델에 대하여 설명한다.



〈그림 1〉 참고문헌 메타데이터 인식 모델의 구조

3.1 사전 학습된 언어 모델

사전 학습된 언어 모델은 입력 문장의 양방향 문맥을 모두 고려할 수 있는 트랜스포머(Transformer) 모델의 인코더(Encoder)를 여러 층 쌓아 학습한 결과로 워드 임베딩(Word Embedding)을 출력하는 심층 학습 모델이다. 사전 학습된 언어 모델의 가장 큰 특징은 같은 단어라도 문맥적 의미에 따라 다른 값의 단어 임베딩을 도출한다는 것이다. 예를 들어 〈표 1〉과 같이 '최유현, 문대영, 강경균, 이진우, 이주호 (2008). STEM 기반 발명영재교육 프로그램 개발과 적용 효과. 한국기술교육학회지, 8(2), 143-164.'이라는 참고문헌에 '교육'은 제목과 학술지에서 동시에 등장한다. 그러나 첫 번째 '교육'이 의미하는 것은 '발명영재교육'이라는 교육의 종류 중 하나이지만 두 번째 '교육'이 의미하는 것은 '한국기술교육학회지'라는 학술지 이름에 포함되어 있는 단어이다. 기존의 단어 임베딩 생성 방법론은 '교육'이라는 단어에 대해 동일한 벡터 값을 부여했으나 사전 학습된 언어 모델은 문맥적 의미에 따라 다른 벡터 값을 부여한다. 실제로 사전 학습된 언어 모델 중 한국어 데이터 처리를 위해 34GB의 한국어 데이터를 ELECTRA 모델에 추가로 학습한 KoELECTRA-Base-v3 모델에 위의 참고문헌을 입력한 뒤 출력되는 단어 임베딩을 살펴보면 〈표 1〉과 같은 결과를 얻어 문맥적 정보가 참고문헌에도 반영되고 있음을 확인할 수 있다.

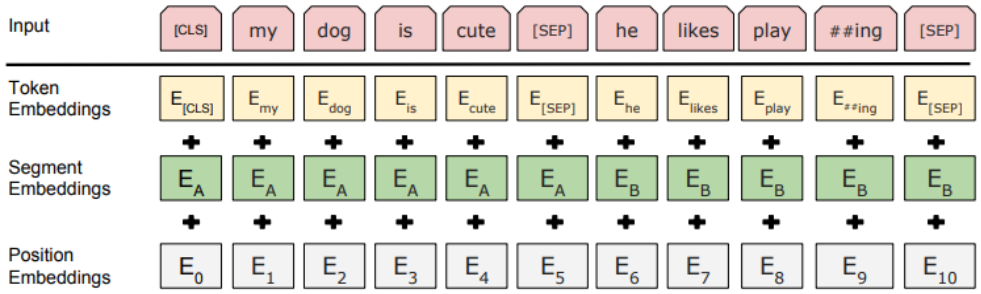
현재 배포된 다양한 사전 학습된 언어 모델 중 본 연구에서 사용한 사전 학습된 언어 모델은 BERT의 다국어 학습 모델인 BERT(multilingual base cased) 모델과 BERT를 기반으로 한국어 처리를 위해 파생된 모델인 HanBERT와 KoBERT이며 또 다른 사전 학습된 언어 모델인 ELECTRA (Kevin, Minh-Thang, Quoc, & Christopher, 2020)에서 한국어 처리를 위해 파생된 KoELECTRA 모델까지 총 4종이다. 먼저 대표적인 사전 학습된 언어 모델인 BERT는 문맥적 정보를 반영한 워드 임베딩을 생성하기 위해서 트랜스포머의 인코더를 여러 층 쌓아 모델을 구성하였으며 토큰화 된 두 개의 문장을 동시에 입력으로 받아 처리한다. 문장을 토큰화 하는 기준은 Word Piece(Sennrich, Haddow, & Birch, 2015) 방식이다. 기존의 토큰화 방식은 주로 띄어쓰기를 기준으로 분리하는 것이 일반적이었으나 미등록 언어를 처리하는 과정에서 오류가 자주 발생하여 그 대안으로 Word Piece 방식이 고안되었다. 이 방식은 입력된 문장을 글자 단위

〈표 1〉 문맥적 의미가 담긴 임베딩 예시

참고문헌	
최유현, 문대영, 강경균, 이진우, 이주호 (2008). STEM 기반 발명영재교육(1) 프로그램 개발과 적용 효과. 한국기술교육(2) 학회지, 8(2), 143-164.	
	단어 임베딩 값
교육(1)	0.2766, -0.0713, -0.1702, 0.8986, …
교육(2)	0.5417, -0.2219, -1.3873, 0.7504, …

의 서브 유닛(Sub Units)으로 인식하고 자주 등장하는 글자의 묶음을 유닛(Unit)으로 모아 처리하는 것이다. 〈그림 2〉의 Input 부분을 살펴보면 'my, dog, is, cute, he, likes, play, ##ing'으로 토큰화 된 것을 확인할 수 있다. 각각의 토큰들은 자주 등장하는 서브 토큰(Sub Tokens)들을 모아 유닛으로 처리한 것이며 'playing' 이라는 단어는 'play'와 'ing'으로 분리되어 처리된 것을 확인할 수 있다. 'play'와 'ing'이 원래는 'playing'이라는 하나의 단어라는 것을 표기하기 위하여 유닛으로 분리된 사이에 '##' 표기를 추가하여 오류가 없도록 하였다. 이 때 언어 모델별로 지정된 토큰 분리기(Tokenizer)를 사용한다. 같은 문장을 분리해도 어떤 언어 모델의 토큰 분리기를 사용하느냐에 따라서 결과물이 달라지기 때문에 언어 모델에 맞는 토큰 분리기를 사용하는 것이 중요하다. 또한 BERT 모델에는 2개의 문장이 동시에 입력되기 때문에 첫 번째 문장과 두 번째 문장을 구분하기 위해 입력된 문장의 가장 첫 번째 부분에 [CLS] 토큰을 추가하며 문장과 문장 사이에 문장이 종료되었다는 의미의 [SEP] 토큰을 추가하고 두 번째 문장이 끝난 뒤에 종료되었다는 의미의 [SEP] 토큰을 추가한다. 토큰의 최대 길이는 언어 모델이 처리 가능한 512개로 제한되어 있다. 입력된 문장을 언어 모델에 맞는 토크나이저(Tokenizer)로 토큰화한 뒤 〈그림 2〉와 같이 3종류의 임베딩인 토큰 임베딩(Token Eembedding)과 문장 임베딩(Segment Embedding) 그리고 각 토큰별 위치값을 벡터화 시킨 포지션 임베딩(Position Embedding)을 취합하여 하나의 임베딩 벡터(Embedding Vector)로 연결한다. 그리고 이 임베딩 벡터를 정규화 하여 트랜스포머 인코더의 입력값으로 사용한다. 〈그림 2〉의 경우 입력된 두 개의 문장('my dog is cute', 'he likes playing')의 첫 부분에 [CLS] 토큰이 추가되었으며 하나의 문장이 끝나면 [SEP] 토큰이 추가된 것을 확인할 수 있다. 문장 임베딩의 경우 [CLS] 토큰부터 첫 번째 [SEP] 토큰까지 하나의 문장(Segment)로 보고 E_A로 처리하였으며 he부터 [SEP] 토큰까지 E_B로 처리한 것을 확인할 수 있다.

이후 트랜스포머 인코더를 여러 층 쌓아 앞서 생성한 임베딩 벡터를 학습하게 되는데, 이 때 사용하는 학습 방식은 크게 2가지로 나눌 수 있으며 BERT의 가장 큰 특징이라고 할 수 있다. 먼저 첫 번째 학습 방식은 마스킹 된 언어 모



〈그림 2〉 BERT 모델에 입력되는 임베딩 벡터 구조(Jacob et al., 2018)

델(Masked Language Model, MLM)이다. 이 방법은 앞서 생성한 단어 임베딩을 학습하는 과정에서 문장 속 단어를 15%의 비율로 랜덤하게 마스킹(Masking) 처리하여 가리고, 가려진 토큰이 있는 문장을 트랜스포머 인코더에 입력시켜 주변 단어의 문맥 정보만 활용하여 마스킹 된 단어가 무엇인지 예측하는 방식이다. 기존의 언어 모델은 뒤에 있는 단어만 예측 가능한 단방향 방식이었으나 BERT는 예측 방향에 구애받지 않고 다양한 위치에 있는 단어를 예측할 수 있기 때문에 양방향 학습이 가능하다. 두 번째 학습 방식은 Next Sentence Prediction (NSP) 방식으로 두 개의 문장이 주어졌을 때 두 번째 문장이 첫 번째 문장 뒤에 나타나는 문장인지 예측하는 방식을 활용하여 토큰과 문장 모두 양방향으로 학습하는 방식이다. 단순한 학습 방식으로 문맥의 앞 뒤 정보를 효과적으로 학습할 수 있게 된다. BERT 모델의 학습 집단은 'BookCorpus'와 위키백과 영문판을 사용하여 구축했는데 'BookCorpus'의 경우 8천만 개의 단어가 포함되어 있으며 위키백과 영문판에 포함된 단어의 개수는 2억 5천만 개다. 따라서 총 3억 3천만 개에 달하는 대용량의 언어 말뭉치가 사용되기 때문에 위와 같은 사전 학습이 완료되면 BERT 모델은 입력 문장에 대하여 문맥적 의미가 포함된 단어 임베딩을 출력하게 된다. 따라서 학습 집합들은 각 언어 모델에 맞는 토크나이저로 분리되어 입력되고 사전 학습된 모델의 가중치를 사용하여 계산된 뒤 단어 임베딩 형태로 출력되고 3.2.절에서 설명할 심층 학습 모델의 입력값으로 활용되었다.

BERT 모델의 학습 집합은 영어로만 구성되어 있기 때문에 영어가 아닌 다국어 처리하기 위해 2018년 11월 BERT(multilingual base cased)모델이 추가로 배포되었다. 그러나 해당 모델은 다국어 처리용으로 개발된 모델이기 때문에 한국어 위주의 데이터 처리 향상을 위하여 KoBERT와 HanBERT가 파생되었다. KoBERT는 SKTBrain에서 한국어 위키백과에서 한국어 문장 500만 개와 단어 5천 4백만 개를 추가로 학습하여 배포한 모델이다. HanBERT는 투블럭 AI에서 학습하여 배포한 모델로 한국어 문장 3.5억 개와 형태소 113억 개를 추가로 학습하여 배포하였으며 ubuntu 18.04 환경에서만 실행이 가능한 특징이 있다. 또한 BERT를 기반으로 파생된 모델 중 대체 토큰 탐지(Replaced Token Detection) 기법을 사용하여 토큰 전체를 마스킹 했던 BERT와 비교하여 일부 토큰을 비슷한 뜻을 가진 다른 토큰으로 대체하여 보다 효율적인 학습이 가능한 ELECTRA 모델을 기반으로 한국어 데이터 처리 향상을 위해 파생된 KoELECTRA가 배포되었다. KoELECTRA는 한국어 뉴스, 위키백과, 나무위키, 모두의 말뭉치 등 약 34GB의 한국어 데이터를 추가로 사용하였다. KoBERT와 HanBERT는 BERT를 기반으로, KoELECTRA는 ELECTRA를 기반으로 파생된 모델이며 학습 과정에서 추가로 사용된 데이터가 모두 상이하기 때문에 본 논문에서는 위의 모델을 대상으로 실험을 진행하여 참고문헌 메타데이터 인식 모델에서 가장 효과가 좋은 모델을 찾고자 하였다. 따라서 가장 대표적인 사전 학습 언어 모델인 BERT 중에서도 한국어를 포함한 104개국의 언어를 학습한 BERT(multilingual base cased) 모델과 한국어 데이터 처리 향상을 위해 BERT에서

파생된 모델 중 오픈소스로 공개되어 접근 가능한 KoBERT와 HanBERT 그리고 ELECTRA에서 파생된 모델인 KoELECTRA 중 base-v3 버전의 모델을 워드 임베딩으로 활용하였다. HanBERT의 경우 현재는 비공개 상태이나 비공개 전환 이전에 공개된 모델은 연구 목적으로 사용 가능하기 때문에 실험 대상 모델로 선정하였다.

3.2 양방향 게이트 순환 유닛 모델과 조건부 랜덤 필드

양방향 게이트 순환 유닛 모델은 연속적으로 입력되는 자질 분석에 뛰어난 성능을 보이고 있는 모델로 제안 초기에는 개별적으로 사용되었으나 최근에는 연결 층의 가장 끝 부분에 조건부 랜덤 필드를 추가하여 사용되고 있는 모델 중 하나이다.

3.1.에서 계산한 단어 임베딩 X 중 t번째 입력에 해당하는 토큰 x_t에 대한 게이트 순환 유닛 모델 셀의 연산은 다음 수식과 같다.

$$z_t = sigmoid(W_z x_t + U_z h_{t-1}) \quad (1)$$

$$r_t = sigmoid(W_r x_t + U_r h_{t-1)}) \quad (2)$$

$$\tilde{h_i} = \tanh(Wx_t + Uh_{t-1} * r_t) \quad (3)$$

$$h_t = (1-z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (4)$$

양방향 게이트 순환 유닛은 입력 벡터가 게이트(Gate)를 통과할 때 이전에 학습된 은닉층 정보의 양을 적절하게 조절하는 모델이다. 게이트는 업데이트 게이트(Update Gate, z)와 리셋 게이트(Reset Gate, r)로 총 2개이다. 수식(1)은 업데이트 게이트로 시그모이드(Sigmoid) 연산을 수행하여 이전 단계 학습 정보인 h_{t-1}와 현재 입력 값인 x_t의 정보 업데이트 비율인 z_t를 구하는 게이트이며 수식(2)는 리셋 게이트로 h_{t-1}의 양을 얼마나 지울지 시그모이드 연산을 수행하여 r_t를 구하는 게이트이다. 수식(3)에서는 리셋 게이트를 통해 계산된 r_t를 활용하여 과거의 데이터인 Uh_{t-1}을 얼마나 사용할지 점별 곱(Pointwise Product)과 탄젠트 연산을 통해 구한다. 수식(4)에서는 z_t와 r_t를 활용하여 현재 시점의 은닉층 h_t를 계산한다. z_t는 현재 입력값의 업데이트 비율이므로 $(1-z_t)$는 업데이트하지 않을 비율을 의미한다. $(1-z_t)*h_{t-1}$은 이전 단계 학습 정보를 얼마나 지울지 계산하는 것이고, $z_t*\tilde{h_t}$는 이전 단계 학습 정보 중 다음 단계 연산에 가져갈 정보를 계산하는 것이다. 따라서 게이트 순환 유닛 모델 셀 연산의 결과인 h_t는 이전 단계와 현 단계의 정보가 적절히 계산된 정보라고 할 수 있다. 연산에서 사용되는 W와 U는 가중치 행렬을 의미한다.

이러한 게이트 순환 유닛 모델 셀의 연산은 단어에 대해 정방향과 역방향으로 수행되며, 각 셀의 결과를 병합하여 반환된다. 이후 토큰 리스트 전체 단위의 연결 점수를 측정하고 예측하는 형식의 조건부 랜덤 필드 연산을 수행한다. 조건부 랜덤 필드는 각 토큰 단위의 자질 분석 값을 기준으로 정답값을 예측하고, 해당 정답값을 기준하여 다음 토큰의 정답값을 확률을 추가적으로 예측하는 식의 연산을 수행한다. 이를 전부 더하여, 리스트 단위의 연결적

인 점수를 측정하고, 이를 활용하여 Viterbi 알고리즘을 통한 예측이나 Log-Likelihood와 같은 학습의 기준 값을 얻을 수 있다. 본 연구에서 활용한 모델은 조건부 랜덤 필드 층에서 최종 연산된 Log-Likelihood를 최대화하며 심층 학습 모델을 학습하고, 조건부 랜덤 필드 층의 전이 행렬을 활용해 Viterbi 알고리즘을 적용하여 예측을 수행한다.

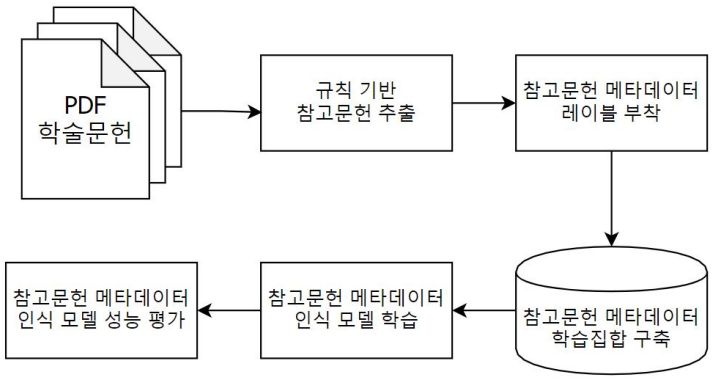
4. 실험 계획 및 환경

본 연구는 앞서 설명한 참고문헌 메타데이터 모델에 활용될 실험 집합을 구축하기 위해 Python용 PDF 분석 프로그램인 PDFMiner(Yusuke, 발행년불명)를 사용하였으며 인식 모델의 학습을 위해 Python 기반의 심층 학습 프로그램인 Pytorch(Paszke et al., 2019)를 사용하였다. 이후 본 연구에서 구축한 실험 집단을 통하여 참고문헌 메타데이터 인식 모델에 대한 실험을 수행하였다. 실험의 전체 과정은 〈그림 3〉과 같다.

4.1 실험 집단 구축

본 연구에서는 국내 학술지 대상의 참고문헌 메타데이터 추출 실험 집단을 구축하기 위하여 한국과학기술정보연구원이 구축한 학술문헌 및 참고문헌 메타데이터를 활용하였다. 실험 집단은 2018년도에 발행된 논문 편수를 기준으로 상위 40종의 학술지를 선정하였으며 상위 40종 학술지의 학술문헌은 모두 PDF 파일이기 때문에 이를 참고문헌 메타데이터 인식 모델에 입력하기 위한 실험 집단 형태로 가공하기 위하여 PDF를 텍스트 형식으로 변환하는 과정이 필요하였다. 따라서 PDF 분석 프로그램인 PDFMiner를 활용하여 학술문헌을 페이지 단위로 분석하였다.

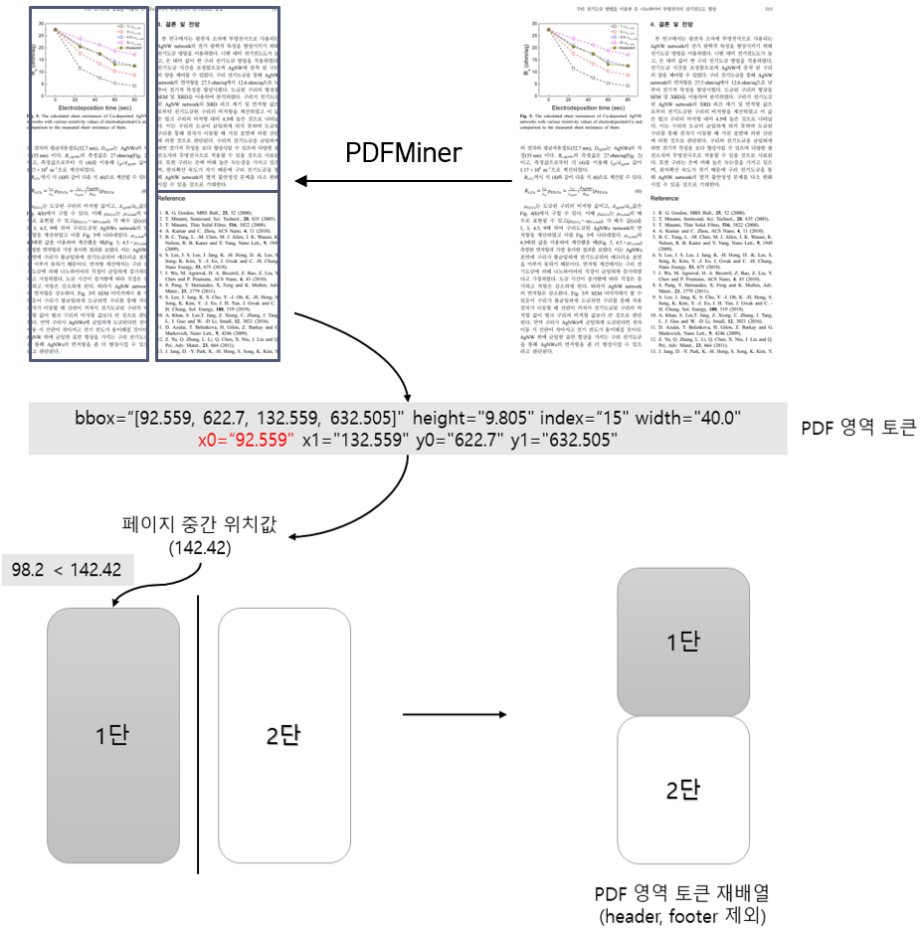
먼저 PDFMiner를 통해 분석된 PDF는 각 텍스트 정보와 좌표 값을 추출할 수 있다. 각 토큰의 좌푯값은 PDF 내에서 해당 토큰 정보가 위치한 곳에 맞게 왼쪽(x_0), 오른쪽(x_1), 아래(y_0), 위(y_1)에 해당하는 좌푯값이 별개로 지정되어 추출되도록 처리하였다. 해당 좌푯값과 텍스트 정보를 글자 수준까지 세분화하



〈그림 3〉 참고문헌 메타데이터 인식 모델 실험 과정

여 각 PDF 내의 텍스트 정보를 추출하였다. 일부 학술지의 경우 PDF를 분석한 결과의 글자 순서가 눈에 보이는 것과 다르게 구성되어 있기 때문에 이를 좌푯값을 기준으로 각 y_0 좌표의 순서를 우선시하고, y_0가 일치할 경우, x_0의 순서로 맞춰 재배열하는 작업을 거쳤다. 글자 수준의 좌표 재배열 이후, y_0 좌푯값을 기준으로, 동일 선상에 위치한 글자 토큰을 정리한다. 이후 폰트의 크기(h)와 x_0 좌푯값 간의 거리를 비교하여 같은 줄로 묶일 수 있는 토큰인지를 분류하였다. 줄 수준의 토큰을 구성한 이후 y_0 좌푯값 간의 거리, 폰트의 크기, x_0 좌푯값 간의 거리를 추가적으로 비교하여 박스 형태의 영역 토큰을 구성하였다.

또한 참고문헌 영역이 포함되어 있는 본문 구조에서는 대부분의 학술지가 2단 형식을 따르고 있다. 따라서 본 연구에서는 다단을 나누어 순서대로 정렬하기 위한 알고리즘을 추가로 구현하였다. 〈그림 4〉는 해당 알고리즘을 간략하게 표현한 것이다. 먼저 PDF 페이지 전



〈그림 4〉 PDF 학술문헌 구조 분석 과정

체의 넓이를 구하여 이에 대한 절반에 해당하는 값을 기준 값으로 구하였다. 이후 PDF 영역 토큰 구성이 완료되면, 각 영역 단위의 x_0 좌표 값을 기준으로 해당 중심 기준 값을 넘어가는지 여부를 판단하였다. 넘어가지 않을 경우, 첫 번째 단의 토큰으로 배열하고, 넘길 경우 두 번째 단의 토큰으로 배열하였다. 이후 페이지 전체의 토큰의 배열이 완료되면, 두 번째 단에 해당하는 토큰들을 첫 번째 단의 뒤에 연결하였다. 이를 통해 영역 토큰을 다단에 맞추어 사람이 읽는 방향으로 배열할 수 있다. 또한 이 과정에서 페이지의 높이를 기준하여 일정 비율에 해당하는 값을 제외하여 머리말과 꼬리말을 제거하였다.

이후, 40종의 학술지에 대한 참고문헌 영역 분석을 통해 참고문헌 영역의 시작과 끝에 대한 규칙을 탐색하였다. 추출한 참고문헌 영역의 시작과 끝에 대한 규칙을 정규식으로 구현하여 적용하였다. 이를 활용하여 영역 자체를 분리한 이후, 영역 내에서 참고문헌을 건 단위로 분리하기 위한 규칙을 추가적으로 파악하여 정규식으로 구현하여 적용하였다. 정규식 구현이 어려운 들여쓰기를 통한 참고문헌 분리 작업은 박스 단위의 영역 내에서 가장 좌측의 좌표인 x_0를 기준하여 박스 내의 라인 단위의 x_0를 측정하여 일정 길이 값 이상 증가한 위치부터 시작되는 라인을 들여쓰기를 쓰고 있는 라인으로 분류하였다.

분석 결과 상위 40종 학술지의 참고문헌 추출 패턴은 〈표 2〉와 같다. 참고문헌 전체 시작 패턴은 주로 참고문헌임을 알리는 키워드로 시작되는 경우가 대부분이다. 주로 'Reference', 'References', '참고문헌' 키워드가 사용되었음을 알 수 있다. 참고문헌 시작 패턴은 크게 4가지 패턴으로 나눌 수 있는데 각괄호([]), 온점(.), 들여쓰기, 줄 간격으로 구분할 수 있다.

논문의 마지막 페이지에서부터 탐색을 시작하여 참고문헌 전체 시작 패턴을 기준으로 실제 참고문헌 영역의 시작점을 찾는다. 이 때 시작점이 잡히더라도 해당 박스가 논문의 전체 크기를 일정 비율로 나누어 상단과 하단 20%에 해당하는 영역에 위치한 경우 머리말 또는 꼬리말로 간주하여 제외하였다. 또한 논문의 본문 또는 표에서 'Reference', '참고문헌'과 같은 키워드가 등장할 수 있기 때문에 가장 마지막에 등장하는 키워드를 참고문헌 시작점으로 간주하였다.

참고문헌 영역의 시작점을 찾은 경우 해당 시작점 다음에 오는 PDF 영역 박스를 실제 참고문헌이 시작되는 박스(StartBox)라고 가정한 뒤 참고문헌 시작 패턴을 탐색하였다. 참고문헌 시작 지점부터 종료 지점까지의 박스를 탐색하며 해당 박스가 논문 상단 및 하단 20% 영역에 위치하지 않은 경우, 글자 크기가 StartBox와 3포인트 이상 차이가 나는 경우 등 실제 참고문헌 영역이 아닌 곳에 위치한 박스는 제외하고 참고문헌 영역으로 추출하였다. 이후 참고문헌 시작 패턴을 기준으로 추출한 참고문헌 텍스트를 개별 참고문헌으로 분리하였다.

참고문헌 시작 패턴이 각괄호와 온점인 경우는 정규식으로 분리가 가능하였으며 시작 패턴이 들여쓰기인 경우는 각 라인의 시작 지점을 x값으로 수집하여 다른 라인에 비하여 오른쪽에 위치한 라인을 들여쓰기 되어 있다고 가정하고 분리하였다. 시작 패턴이 줄바꿈인 경우는 각 라인의 시작 지점을 y값으로 수

〈표 2〉 학술지별 참고문헌 구역 및 참고문헌 시작 패턴

학술지	참고문헌 구역 시작 패턴	참고문헌 시작 패턴
한국산학기술학회논문지	References	[]
한국콘텐츠학회논문지	참고문헌	[]
디지털융복합연구	REFERENCES	[]
한국정보통신학회논문지	REFERENCES	[]
한국컴퓨터정보학회논문지	REFERENCES	[]
아세아태평양축산학회지	REFERENCES	.
한국인터넷방송통신학회논문지	References	[]
대한토목학회논문집	References	들여쓰기
한국융합학회논문지	REFERENCES	[]
KSII Transactions on Internet and Information Systems (TIIS)	References	[]
Journal of Microbiology and Biotechnology	References	.
생명과학회지	References	.
한국전자통신학회논문지	References	[]
대한수학회보	References	[]
Journal of Korean Neurosurgical Society	References	.
한국식품영양학회지	References	들여쓰기
한국멀티미디어학회논문지	REFERENCE	[]
정보보호학회논문지	References	[]
Korean Chemical Engineering Research	References	.
한국재료학회지	References	.
한국식품과학회지	References	.
한국수자원학회논문집	References	들여쓰기
Journal of Power Electronics	REFERENCES	[]
전기학회논문지	References	[]
한국응용과학기술학회지	References	.
Nuclear Engineering and Technology	References	[]
한국과학교육학회지	References	들여쓰기
한국축산식품학회지	References	들여쓰기
BMB Reports	REFERENCES	.
응용통계연구	References	들여쓰기
한국건설관리학회논문집	REFERENCES	줄간격
한국의류학회지	References	들여쓰기
디지털콘텐츠학회 논문지	참고문헌	[]
Molecules and Cells	References	들여쓰기
인터넷정보학회논문지	참고문헌(Reference)	[]
Biomolecules & Therapeutics	REFERENCES	들여쓰기
공업화학	References	.
한국항해항만학회지	References	[]
The Korean Journal of Physiology and Pharmacology	REFERENCES	.
한국문헌정보학회지	참고문헌	[]

집한 뒤 해당 라인의 윗줄과 아랫줄의 y값을 비교하여 다른 라인에 비하여 줄 간격이 큰 지점을 기준으로 분리하였다.

참고문헌 영역의 종료 지점은 저자 소개, 사사표기, 출판년도 정보 등 학술지별로 다양한 패턴이 사용되었으므로 참고문헌 영역 시작지점 이후부터 탐색하여 위에 해당되는 패턴이 등장하는 PDF 영역 박스를 종료 지점으로 설정하였다.

위의 과정을 통해 학술논문 53,562건을 분석하여 추출한 참고문헌은 총 791,288건이다. 이후 정확하게 추출된 참고문헌만 남긴 뒤 참고문헌 메타데이터 인식 모델의 실험 집단 양식에 맞게 재가공하는 과정을 거쳤다. 먼저, 정확하게 추출된 참고문헌을 판단하는 기준은 한국과학기술연구원에서 제공한 참고문헌 메타데이터 파일의 참고문헌을 기준으로 추출한 참고문헌이 참고문헌 메타데이터 파일에 존재하며 100% 일치하는 경우 정확하게 추출된 참고문헌으로 간주하였다. 이 과정에서 띄어쓰기 오류가 있거나 참고문헌이 중간에 끊긴 경우 또는 참고문헌이 아닌 문장이 추출된 경우 등 기타 문장들이 필터링 되었다. 다음으로 정확하게 추출된 참고문헌의 중복을 제거하고 실험 집단으로서의 의미가 있는 참고문헌을 선정하는 과정을 거쳤다. 부적합 데이터를 판단하는 기준은 참고문헌에서 추출 가능한 메타데이터 중 필수 메타데이터 3종(제목, 저자, 발행년도)를 설정하고, 해당 메타데이터가 모두 존재하는 참고문헌만 실험 집단으로 가공하였다. 그러나 웹 정보원은 필수 메타데이터 3종의 조건을 충족하지 않은 경우에도 실험 집단으로 간주하였다. 위의 과정은 Python 기반으로 구현되었으며 웹 정보원 판단 여부는 정규식을 사용하여 구현하였다.

다음으로 인식할 참고문헌 메타데이터의 종류를 선정하였다. 추출한 참고문헌을 대상으로 한국과학기술연구원에서 제공한 참고문헌 메타데이터 파일을 분석하여 메타데이터 16종을 추가로 선정하였다. 이후 참고문헌 원본과 기본 메타데이터 3종, 추가 메타데이터 16종을 매핑하는 모듈을 설정하고 Python 기반으로 구현하여 반자동 학습집합 구축을 완료하였다. 최종적으로 선정된 레이블은 〈표 3〉과 같다.

〈표 3〉 인식 가능한 메타데이터 종류

메타데이터	설명	레이블	메타데이터	설명	레이블
ATCL_NM	기사 명	TIT	PUBR	출판사	PUBR
AUT_NM	저자 명	AUT	PUB_PLC	출판지	PUB_PLC
PUB_YR	발행 년도	YEAR	ISSN	ISSN	ISSN
JNL_NM	학술지 명	JOU	CNCL_PLC	회의 장소	CNCL_PLC
PAGE	페이지	PAGE	CNCL_DT	회의 일자	CNCL_DT
VOL	권	VOL	NOTE	비고	NOTE
ISS	호	ISS	PART	파트	PART
DOI	DOI	DOI	ISBN	ISBN	ISBN
URL	URL	URL	SERIES_CNTNT	총서사항	SERIES_CNTNT
PUB_ORG	발행 기관	PUB_ORG			

학습 데이터로 적합한 참고문헌 161,319개와 한국과학기술연구원에서 제공한 각 참고문헌의 메타데이터를 매칭하여 자동으로 레이블을 태깅하였다. 참고문허앞서 3.1.절에서 서술한 바와 같이 언어 모델에 입력 가능한 토큰의 길이는 최대 512개 이므로 참고문헌을 토큰화한 결과가 512개를 넘는 경우는 사전 학습된 언어 모델로 학습이 불가하기 때문에 4건을 제외하였다. 최종 실험 집단으로 구축된 참고문헌 개수는 161,315개이다. 최종적으로 구성된 참고문헌 실험 집단 통계는 〈표 4〉와 같다.

〈표 4〉 참고문헌 실험 집단 통계

	참고문헌 개수
PDF에서 추출한 참고문헌	791,288
부적합 데이터가 제거된 참고문헌	161,319
실험 집단 구축 대상 참고문헌	161,315

자동으로 태깅된 참고문헌 데이터는 모델 입력 데이터에 적합한 형태로 가공하기 위하여 특수기호와 띄어쓰기, 숫자, 언어(한국어, 영어, 중국어, 일본어 등)를 기준으로 토큰화하였다. 특수기호를 기준으로 토큰화 할 경우 문자 단위로 구성하였으며 언어 및 숫자를 기준으로 토큰화할 경우 단어 단위로 구성하였다. URL과 DOI는 내부가 분리될 경우 그 의미를 상실하기 때문에 항목 전체를 토큰으로 사용하였다.

토큰화 이후에는 학습 데이터가 모델의 입력 값으로 활용이 가능하도록 CoNLL 양식을 따라 구성하였다. BIO 태그를 부착하여 시작(Begin) 정보와 내부(Inside)정보, 메타데이터가 아닌(Others) 정보를 별도로 표현하였다. 따라서 DOI와 URI를 제외한 총 17종의 메타데이터에 각각 BI 태그가 붙어 34종의 레이블이 생성되었으며 DOI와 URI, 그리고 메타데이터가 아님을 표현하는 'O' 태그와 입력된 참고문헌의 길이를 일정하게 유지시켜주기 위해 필요한 태그인 '〈pad〉' 태그를 추가하여 총 38종의 레이블이 생성되었다.

최종적으로 국문 참고문헌과 영문 참고문헌에 대해 구축된 CoNLL 형식의 각 데이터는 〈표 5〉와 같다.

4.2 실험 집단 분석 및 실험 조건

본 연구에서는 참고문헌 메타데이터 인식 모델을 위해 구현한 사전 학습된 언어 모델 기반의 심층 학습 모델의 성능을 확인하기 위해, 앞서 구축한 실험 집단을 활용하여 모델을 학습하고 실험하였다. 실험 데이터는 전체 데이터 161,315건을 학습 집합(Train Set)과 실험 집합(Test Set)으로 나누었으며 비율은 8:2로 설정하였다. 최종적으로 구축된 학습 집합의 개수는 129,044개이며 실험 집합의 개수는 32,261개이다.

참고문헌 메타데이터 인식 모델 학습에 사용한 주요 파라미터는 〈표 6〉과 같다. 〈표 6〉의 파라미터 값들은 모든 실험에 동일하게 적용하였다. Epoch는 학습 집합을 총 몇 번 반복할 것인지 정하는 횟수이며, Learning Rate는 학습의 효율을 높이기 위한 것으로 모델 학습시 기울기 값을 조정하는 범위를 의미한다. Batch Size는 한 번에 모델에 입력되는 학습 데이터의 개수를 의미한다. 본 연구에서는 16으로 설정하였으므로 모델에 입력되는 참고문헌 개수

〈표 5〉 참고문헌의 실험 집합 가공 결과

언어	한국어		영어	
참고문헌 원형정보	최유현, 문대영, 강경균, 이진우, 이주호(2008). STEM 기반 발명영재교육 프로그램 개발과 적용 효과. 한국기술교육학회지, 8(2), 143-164.		Montgomery, D. C., 2012, Design and Analysis of Experiments, 8th ed., Wiley, Hoboken, NJ, pp. 523-530	
태깅 결과	최유현	B-AUT	Montgomery	B-AUT
	,	I-AUT	,	I-AUT
	문대영	I-AUT	D	I-AUT
	,	I-AUT	.	I-AUT
	강경균	I-AUT	C	I-AUT
	,	I-AUT	.	I-AUT
	이진우	I-AUT	,	O
	,	I-AUT	2012	B-YEAR
	이주호	I-AUT	,	O
	(	O	Design	B-TIT
	2008	B-YEAR	and	I-TIT
	)	O	Analysis	I-TIT
	.	O	of	I-TIT
	STEM	B-TIT	Experiments	I-TIT
	기반	I-TIT	,	I-TIT
	발명영재교육	I-TIT	8th	I-TIT
	프로그램	I-TIT	ed	I-TIT
	개발과	I-TIT	.	I-TIT
	적용	I-TIT	,	O
	효과	I-TIT	Wiley	B-PUBR
	.	O	,	O
	한국기술교육학회지	B-JOU	Hoboken	B-PUB_PLC
	,	O	,	I-PUB_PLC
	8	B-VOL	NJ	I-PUB_PLC
	(	O	,	O
	2	B-ISS	pp	O
	)	O	.	O
	,	O	523	B-PAGE
	143	B-PAGE	-	I-PAGE
	-	I-PAGE	530	I-PAGE
	164	I-PAGE		
	.	O		

는 16개가 된다. Optimizer는 Loss 값을 최소화하는 과정에서 활용되는 파라미터이며, 학습 알고리즘으로 일반적으로 널리 사용되는 Adam Optimizer를 사용하였다. GRU Dims는 전체 입력 벡터에 대한 양방향 게이트 순환 유닛 모델 내의 게이트 순환 유닛 모델 셀의 출력 결과 차원 값이다.

〈표 6〉 주요 파라미터 값

파라미터	설정 값
Epoch	10
Learning Rate	0.001
Batch Size	16
Optimizer	Adam
GRU Dims	100
Dropout	0.7

4.3 실험 결과

본 연구에서는 앞서 설명한 참고문헌 메타데이터 인식 모델을 사전 학습된 언어 모델별로 실험한 후 결과를 비교하였다. 또한 실험 집합의 규모가 모델 성능에 끼치는 영향을 알아보기 위하여 성능이 가장 높은 사전 학습된 언어 모델을 대상으로 실험 집합의 규모를 조절하여 추가 실험을 진행하였다. 모든 성능은 소수 셋째 자리에서 반올림 한 값이다.

사전 학습된 언어 모델별 실험 결과는 〈표 7〉과 같다. 전반적으로 BERT(multilingual base cased) 모델의 성능이 정확도(Accuracy) 96.61%, 정밀도(Precision) 95.66%, 재현율(Recall) 97.50%, F1 Score 96.57%로 높았다. 국내에서 발행되는 학술 문헌의 참고문헌은 한국어뿐만 아니라 영어, 한자 등 다양한 언어로 표기된다. 따라서 104개국의 언어로 사전 학습된 언어 모델인 BERT(multilingual base cased) 모델의 성능이 전반적으로 높게 나온 것으로 볼 수 있다. 그러나 다른 모델과의 차이가 정확도는 최대 1.56, 최소 0.38% 차이가 나고 정밀도의 경우 최대 3.06%, 최소 0.12% 차이가 났으며 재현율의 경우 최대 3.18%, 최소 0.26% 차이가 났으며 F1 점수의 경우 최대 3.12%, 최소 0.19% 차이가 났다. 최고 성능과 최소 성능과의 차이가 전반적으로 5%를 넘지 않고 최소 0.12% 차이가 나는 경우도 있으므로 연구 환경에 따라 적절한 언어 모델을 사용하는 것을 권장할 수 있다. 또한 최고 성능을 보였던 BERT(multilingual base cased) 모델과 가장 성능 차이가 크게 나타난 모델은 KoBERT이고 적게 나타난 모델은 KoELECTRA-Base-v3 모델이다. KoBERT의 경우 사전 학습된 언어 모델의 한국어 데이터 처리 성능 향상을 위해 추가 학습된 한국어 데이터 개수가 가장 적은 모델이며 KoELECTRA-Base-v3 모델은 앞서 설명한 바와 같이 BERT 모델의 경량화 및 성능 향상을 위해 개발된 ELECTRA 모델에 한국어 학습 데이터 약 34GB를 추가하여 개발된 모델로 추가된 한국어 데이터 수가 가장 많은 모델이다. 이는 언어 모델을 사전 학습할 때 추가된 한국어 데이터의 수가 많을수록 성능이 높아진다고 볼 수 있다.

모델의 성능이 가장 높았던 BERT(multilingual base cased) 모델을 중심으로 실험 집합의 규모가 모델 성능에 끼치는 영향을 알아보기 위하여 실험 집합의 규모를 세분화하여 추가 실험을 실시하였다. 학습 집합의 개수를 20%, 40%, 60%, 80% 비율로 조정하였으며 총 학습 횟수는 기존 학습 횟수보다 5배 많은 100으로 조정하였다. 이외의 실험 조건은 〈표 6〉과 동일

〈표 7〉 사전 학습된 언어모델별 참고문헌 메타데이터 인식 성능 비교

실험 모델	정확도	정밀도	재현율	F1
Bi-GRU-CRF + BERT(multilingual base-cased)	**96.91**	**95.66**	**97.50**	**96.57**
Bi-GRU-CRF + KoBERT	95.35	92.60	94.32	93.45
Bi-GRU-CRF + HanBERT	95.98	94.25	96.31	95.27
Bi-GRU-CRF + KoELECTRA-Base-v3	96.53	95.54	97.24	96.38

〈표 8〉 참고문헌 학습 집합 개수별 참고문헌 메타데이터 인식 성능 비교

비율	참고문헌 개수	정확도	정밀도	재현율	F1	최고성능 epoch
20%	12,904	96.66	95.79	97.54	96.65	65
40%	25,809	96.73	95.85	97.64	96.74	99
60%	51,618	**96.78**	95.90	97.65	96.77	100
80%	103,235	96.75	95.90	97.66	96.77	80
100%	129,044	96.77	**95.93**	**97.70**	**96.80**	**97**

하며 실험 집합 역시 모두 32,261개의 동일한 집합을 사용하였다. 실험 결과는 〈표 8〉과 같다. 분석 결과 학습 집합의 개수가 많을수록 정밀도, 재현율, F1 점수가 높은 결과가 나왔으며 정확도의 경우 학습 집합을 60% 사용한 실험이 가장 높은 결과를 보였다. 이는 학습 집합의 개수가 많을수록 좋은 성능을 나타낸다는 것을 보여준다.

〈표 8〉에서 F1 점수가 가장 높았던 학습 데이터를 100% 사용하여 진행한 실험에서 메타데이터 별 성능을 측정한 결과는 〈표 9〉와 같다. 저자, 학술지, 제목, 권, 호, 출판년도 등 참고문헌을 구성하는 주된 메타데이터의 성능은 F1 기준으로 모두 90% 이상의 높은 성능을 보였다. 이는 학습 집합 개수가 많아 충분한 양의 학습이 가능했기 때문인 것으로 보인다. 이외에도 URL과 DOI의 경우 참고문헌에서 자주 등장하지 않은 항목임에도 F1 점수가 98.53%와 100%로 높게 나왔다. 이는 URL과 DOI 자체가 일정한 규칙을 가지고 구성되어 있기 때문에 다른 항목보다 상대적으로 학습이 쉬워 높은 성능이 나온 것으로 보인다.

반면 회의(학술대회) 장소, 회의(학술대회) 일자, ISSN, ISBN과 같이 참고문헌에서 자주 쓰이지 않는 메타데이터의 경우 F1 점수가 39.20~63.64% 정도의 성능을 보였다. 이는 앞서 설명한 것과 마찬가지로 충분히 학습 가능한 양이 적기 때문에 성능이 다른 메타데이터와 비교하여 낮은 것으로 보인다. 이외에도 NOTE, PART 메타데이터의 성능은 0으로 나타났다. 위 메타데이터 2종은 모두 참고문헌 메타데이터에서 주로 사용되지 않는 것으로 실험 집합에 포함되지 않기 때문에 성능 측정이 불가하였다.

또한 〈표 9〉에서 F1 점수가 가장 높았던 학습 데이터를 100% 사용하여 진행한 실험 모델에 실험 집합에 포함되지 않은 참고문헌을 입력하여 메타데이터를 추출한 실험 결과는 〈표 10〉, 〈표 11〉, 〈표 12〉와 같다. 실험 결과가 다양한 종류의 학술지를 포함할 수 있는지 확인하고자 실험 대상은 학습 집단에 포함되지 않은 학술지의 학술논문과 단행본, 학술대회 발표자료로 구분하였다.

〈표 10〉과 〈표 11〉은 성공적으로 참고문헌 메타데이터가 인식된 사례이다. 〈표 10〉은 단행본으로 저자명, 제목, 출판년도, 출판지, 출판사 모두 성공적으로 인식되었다. 〈표 11〉은 학술문헌으로 역시 저자명, 제목, 출판년도, 학술지명, 권, 호, 페이지 모두 성공적으로 인식되었다. 특히 앞서 출판지와 출판사를 구분할

〈표 9〉 참고문헌 메타데이터 인식 실험 결과 메타데이터별 성능 비교

항목	내용	정밀도	재현율	F1
AUT	저자명	92.27	95.69	93.35
CNCL_DT	회의 일자	38.10	42.11	40.00
CNCL_PLC	회의 장소	28.06	65.00	39.20
DOI	DOI	100.0	100.0	100.0
ISBN	ISBN	50.00	87.50	63.64
ISS	호	97.27	98.46	97.86
ISSN	ISSN	100.0	25.00	40.00
JOU	학술지 명	95.34	97.22	96.27
NOTE	비고	0	0	0
PAGE	페이지	98.99	99.68	99.33
PART	파트	0	0	0
PUBR	출판사	85.74	92.00	88.76
PUB_ORG	발행 기관	84.02	85.25	84.63
PUB_PLC	출판지	92.64	94.91	93.76
SERIES_CNTNT	총서사항	62.50	55.56	58.82
TIT	제목	93.79	96.76	95.25
URL	URL	99.26	97.81	98.53
VOL	권	99.10	97.80	95.83
YEAR	출판년도	99.47	99.78	99.62

〈표 10〉 인식 성공 사례(단행본)

종류	단행본
참고문헌 원본	이수상 (2012). 네트워크 분석 방법론. 서울: 논형
저자명	이수상
제목	네트워크 분석 방법론
출판년도	2012
출판지	서울
출판사	논형

〈표 11〉 인식 성공 사례(학술문헌)

종류	학술논문
참고문헌 원본	심윤희, 김지현. (2019). 국내 대학도서관의 연구데이터관리서비스 개발 방안에 관한 연구: 서울대학교 소속 연구자들의 요구 분석을 중심으로. 정보관리학회지, 36(3), 61-80.
저자명	심윤희, 김지현
제목	국내 대학도서관의 연구데이터관리서비스 개발 방안에 관한 연구: 서울대학교 소속 연구자들의 요구 분석을 중심으로
출판년도	2019
학술지명	정보관리학회지
권	36
호	3
페이지	61-80

〈표 12〉 인식 실패 사례(학술대회 발표자료)

종류	학술문헌
참고문헌 원본	최원실, 정은경 (2019). 대학도서관 서양서 소장 현황 분석 연구. 제26회 한국정보관리학회 하계학술대회 포스터논문 발표자료, 종로구, 서울.
저자명	최원실, 정은경
제목	대학도서관 서양서 소장 현황 분석 연구
출판년도	2019
학술지명	제26회 한국정보관리학회 하계학술대회 포스터논문 발표자료
발행 기관	종로구
출판지	서울

때 사용되었던 ':' 기호가 들어갔음에도 이를 표제와 부제를 구분하는 기호로 인식하여 성공적으로 제목 전체가 인식된 것을 확인할 수 있다.

〈표 12〉는 메타데이터 인식 실패 사례이다. 학술대회 발표 자료로 저자명과 제목, 출판년도는 성공적으로 인식되었으나 '제26회 한국정보관리학회 하계학술대회 포스터논문 발표자료' 전체를 학술지명으로 인식하고 '종로구'를 발행기관으로 '서울'을 출판지로 인식하는 오류가 발생하였다. 이는 학술대회 발표 자료에 대한 참고문헌 개수가 부족하여 충분한 학습이 이루어지지 않았기 때문으로 보인다.

5. 결 론

최근 학술 문헌의 급격한 증가로 함께 증가하고 있는 참고문헌에 대하여, 각 학술 문헌 간의 연결성과 학술 문헌에 대한 메타 정보를 담고 있는 참고문헌 메타데이터를 식별하는 방법에 대한 연구를 수행하였다. 본 연구에서는 학술지 논문에서 규칙 기반으로 참고문헌을 추출한 뒤 기 구축된 참고문헌 메타데이터 데이터베이스를 활용하여 참고문헌 메타데이터 인식 실험 집합을 자동으로 구축하였다. 그리고 심층 학습 모델 중 최근 자연어 처리 분야에서 높은 성능을 보이고 있는 사전 학습된 언어 모델을 심층 학습 모델에 적용하여 2018년도에 발행된 학술 문헌 53,562건에서 참고문헌 161,315건을 자동으로 추출하여 실험하였다. 실험 결과 참고문헌 메타데이터 인식 정확도 96.61%, 정밀도 95.66%, 재현율 97.50%, F1 점수 96.57%의 높은 성능이 도출되었다. 또한 학습 집합의 개수에 따른 성능 차이를 알아보기 위하여 구축한 학습 집합을 20%, 40%, 80%, 100%로 제한하여 학습 후 성능을 측정하였다. 실험 결과 학습 집합 개수가 많아질수록 정밀도, 재현율 및 F1 점수의 성능이 향상되는 것을 확인할 수 있었다. 본 연구의 제한점으로는 PDF 형식의 학술지 논문에서 참고문헌을 추출하는 과정에서 참고문헌이 온전히 추출되지 않거나 참고문헌이 아닌 문장이 포함되는 경우가 대다수를 차지했기 때문에 이를 보완하는 연구가 필요하다. 또한 실험에 사용한 사전 학습된 언어 모델 중 BERT(multilingual base cased) 모델이 HanBERT, KoBERT, KoELECTRA과 비교하여 성능 차이가 발생한 원인을 다양

하게 분석할 필요가 있다. 본 연구의 실험 결과는 향후 인용 색인 구축 및 참고문헌 메타데이터 데이터베이스 구축시에 유용하게 활용될 수 있으며 또한 추가 자질 없이 사전 학습된 언어 모델 및 심층 학습 모델을 사용하여 도달한 성능으로 이후에 추가 자질 등을 적용할 경우 보다 높은 성능을 기대할 수 있을 것이다.

참 고 문 헌

김선우, 지선영, 설재욱, 정희석, 최성필 (2018). Bidirectionl GRU-GRU CRF 기반 참고문헌 메타데이터 인식. 제 30회 한글 및 한국어 정보처리 학술대회 논문집, 461-464.

김재훈, 김순영, 임석종, 황혜경 (2019). 학술논문과 참고문헌의 자동매핑 사례 분석. 한국콘텐츠학회논문지, 19(11), 262-269. https://doi.org/10.5392/JKCA.2019.19.11.262

김지훈 (2003). 참조연결을 위한 인용정보 자동추출에 관한 연구. 한국문헌정보학회지, 37(1), 247-268.

신규민, 한요섭, 김래현, 차정원 (2009). 기계 학습을 이용한 인용문헌 추출. 한국정보과학회 학술발표논문집, 36(1C), 331-335.

이상기, 김선태, 이용식, 이태석 (2007). 참고문헌 자동파싱 및 참조링킹을 위한 Citation Matcher 연구 및 개발. 한국콘텐츠학회 종합학술대회 논문집, 5(1), 426-429.

임수현, 윤태린, 최경철, 조원민, 허재종, 한현우, 이경원 (2019). 학술 문헌 인용 계보 내 피인용지수를 이용한 참고문헌 탐색 인터페이스 제안. 2019년도 한국HCI학회 학술대회 논문집, 526-529.

An, D., Gao, L., Jiang, Z., Liu, R., & Tang, Z. (2017). Citation metadata extraction via deep neural network-based segment sequence labeling. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 1967-1970. https://doi.org/10.1145/3132847.3133074

Besagni, D. & Belaid, A. (2004). Citation recognition for scientific publications in digital libraries, First International Workshop on Document Image Analysis for Libraries, 2004. Proceedings, 244-252. https://doi.org/10.1109/DIAL.2004.1263253.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Kevin, C., Minh-Thang, L., Quoc V. L., & Christopher D. M. (2020). ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. International Conference on Learning Representations. https://openreview.net/forum?id=rlxMH1BtvB

Park, J. W. (2020). KoELECTRA: Pretrained ELECTRA Model for Korea.

https://github.com/monologg/KoELECTRA

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., Devito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. Advances in Neural Information Processing Systems. 33rd Conference on Neural Information Processing Sytstems, 8024-8035.

Powley, B. & Dale, R. (2007). High accuracy citation extraction and named entity recognition for a heterogeneous corpus of academic papers. In 2007 International Conference on Natural Language Processing and Knowledge Engineering, 119-124.
https://doi.org/10.1109/NLPKE.2007.4368021

Sennrich, R., Haddow, B., & Birch, A. (2015). Neural machine translation of rare words with subword units. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 1715-1725.

SKTBrain (2019). KoBERT, Available: https://github.com/SKTBrain/KoBERT

tbai (2019). HanBERT, Available: https://github.com/tbai2019/HanBert-54k-N

Tkaczyk, D., Szostek, P., Fedoryszak, M., Dendek, P. J., & Bolikowski, L., (2015). CERMINE: automatic extraction of structured metadata from scientific literature. International Journal on Document Analysis and Recognition, 18, 317-335.
https://doi.org/10.1007/s10032-015-0249-8

Yusuke, S. [n.d.]. PdfMiner, Available: https://github.com/pdfminer

• 국문 참고문헌에 대한 영문 표기
(English translation of references written in Korean)

Kim, J. H. (2003). A study on automatic extraction of citation information for reference linking. Journal of the Korean Society for Library and Information Sciences, 37(1), 247-268.

Kim, J. H., Kim, S. Y., Lim, S. J., & Hwang, H. K. (2019). Case study of journal article and reference mapping. The Journal of the Korea Contents Association, 19(11), 262-269.
https://doi.org/10.5392/JKCA.2019.19.11.262

Kim, S. W., Ji, S. W., Seol, J. W., Jeong, H. S., & Choi, S. P. (2018). Bidirectional GRU-GRU CRF based citation metadata recognition. Annual Conference on Human and Language Technology, 461-464.

Lee, S. G., Kim, S. T., Lee, Y. S., & Yi, T. S. (2007). Research and development of citation

matcher for reference parsing and cross-reference linking. KOSTI 2007, 5(1), 426-429.

Lim, S. H., Yoon, T. R., Choi, G. C., Cho, W. M., Heo, J. J., Han, H. W., & Lee, K. W. (2019). A proposal for a bibliographic search interface using impact factor in the genealogy of academic literature. 2019 The HCI Society of Korea, 526-529.

Shin, G. M., Han, Y. S., Kim, L. H., & Cha, J. W. (2009). Citation extraction using machine learning. Korea Computer Congress, 36(1C), 331-335.