KIPS Trans. Softw. and Data Eng.
Vol.9, No.4 pp.145~152 pISSN: 2287-5905

https://doi.org/10.3745/KTSDE.2020.9.4.145

Korean Machine Reading Comprehension for Patent Consultation Using BERT

Jae-Ok Min† · Jin-Woo Park†† · Yu-Jeong Jo††† · Bong-Gun Lee††††

ABSTRACT

MRC (Machine reading comprehension) is the AI NLP task that predict the answer for user's query by understanding of the relevant document and which can be used in automated consult services such as chatbots. Recently, the BERT (Pre-training of Deep Bidirectional Transformers for Language Understanding) model, which shows high performance in various fields of natural language processing, have two phases. First phase is Pre-training the big data of each domain. And second phase is fine-tuning the model for solving each NLP tasks as a prediction. In this paper, we have made the Patent MRC dataset and shown that how to build the patent consultation training data for MRC task. And we propose the method to improve the performance of the MRC task using the Pre-trained Patent-BERT model by the patent consultation corpus and the language processing algorithm suitable for the machine learning of the patent counseling data. As a result of experiment, we show that the performance of the method proposed in this paper is improved to answer the patent counseling query.

Keywords : Natural Language Processing, MRC, Machine Reading Comprehension, Patent, BERT

BERT를 이용한 한국어 특허상담 기계독해

민 재 옥† · 박 진 우†† · 조 유 정††† · 이 봉 건††††

요 약

기계독해는(Machine reading comprehension) 사용자 질의와 관련된 문서를 기계가 이해한 후 정답을 추론하는 인공지능 자연어처리 태스크를 말하며, 이러한 기계독해는 챗봇과 같은 자동상담 서비스에 활용될 수 있다. 최근 자연어처리 분야에서 가장 높은 성능을 보이고 있는 BERT 언어모델은 대용량의 데이터를 pre-training 한 후에 각 자연어처리 태스크에 대해 fine-tuning하여 학습된 모델로 추론함으로써 문제를 해결하는 방식이다. 본 논문에서는 BERT기반 특허상담 기계독해 태스크를 위해 특허상담 데이터 셋을 구축하고 그 구축 방법을 소개하며, patent 코퍼스를 pre-training 한 Patent-BERT 모델과 특허상담 모델학습에 적합한 언어처리 알고리즘을 추가함으로써 특허상담 기계독해 태스크의 성능을 향상시킬 수 있는 방안을 제안한다. 본 논문에서 제안한 방법을 사용하여 특허상담 질의에 대한 정답 결정에서 성능이 향상됨을 보였다.

키워드 : 자연어처리, MRC, 기계독해, 특허, BERT

1. 서 론

오늘날 인공지능 기반의 기술발전과 함께 상담사의 업무를 기계가 대신 답변을 할 수 있는 인공지능 상담에 대한 연구가 증가하고 있다. 지금까지의 기술은 다양한 질의 유형과 표현에 대응하기 위해서 머신러닝 기반의 자연어처리를 하여 시나리오 기반인 액션의 흐름에 따라 정답을 찾아가는 과정이 필요하다. 시나리오 기반으로 접근하는 방식은 다양한 산업분야에서 유연하게 적용하기 어렵고, 전문지식이 필요한 질의에는 정확한 정보 전달을 위해 직접 관련 문서를 찾아야하기 때문에 신속하고 정확하게 답변하는 데에는 한계가 있다. 특히 특허상담분야에서 전문 상담을 위해서는 법률적 지식과 업무 도메인에 특화된 전문용어를 이해할 수 있는 전문 지식을 필요로 한다. 따라서 본 연구에서는 사용자 질의에 대한 정답이 될 수 있는 내용을 해당 문서 내에서 기계가 내용을 이해하여 정답의 위치를 추론하는 자연어처리 분야의 태스크인 기계독해(MRC, Machine Reading Comprehension) 기술을 통해 문제를 해결하고자 하며, 전문 기술용어와 법률정보가 포함되어 있는 특허상담분야로 한정하여 실험을 진행하고자 한다.

본 논문에서는 특허분야 기계독해 연구를 위한 특허상담

※ 이 논문은 2019년도 한국전자통신연구원 지원에 의하여 수행된 것임. (2013-2-00131, 휴먼지식증강 서비스를 위한 지능진화형 WiseQA 플랫폼 기술개발)
※ 이 논문은 2019년도 한국정보처리학회 추계학술발표대회에서 'BERT를 이용한 한국어 특허상담 기계독해'의 제목으로 발표된 논문을 확장한 것임.
† 정 회 원 : 한국특허정보원 R&D센터 연구개발파트장
†† 비 회 원 : 한국특허정보원 R&D센터 연구원
††† 정 회 원 : 한국특허정보원 R&D센터 연구원
†††† 비 회 원 : 한국특허정보원 특허넷응용팀 특허넷응용팀장
Manuscript Received : December 30, 2019
Accepted : February 18, 2020
* Corresponding Author : Bong-Gun Lee(bglee@kipi.or.kr)

데이터 셋을 구축하는 방법을 제안하고, 특허상담 기계독해 학습데이터를 구축하여 시험을 진행한다.

기계독해 알고리즘은 Q&A(Question and Answering), 챗봇(ChatBot)과 같은 자동 질의응답 시스템의 핵심이 될 수 있는 인공지능 기술이며, 본 논문에서는 Google에서 공개한 고성능의 언어모델인 BERT(pre-training of Deep Bidirectional Transformers for Language Understanding) 모델[3]을 사용한다.

공개되어 있는 일반상식분야의 한국어 표준 데이터 셋과 본 연구에서 구축한 특허상담 데이터 셋을 대상으로 기계독해 태스크 결과를 baseline으로 하고, 다양한 실험 과정에서 기계독해 성능 향상을 이룬 모델학습 방법 및 언어처리 알고리즘을 제안한다. 또한 추가적으로 구축한 patent 코퍼스를 사용하여 최적화된 patent 언어모델을 사용하는 것이 특허상담 기계독해 태스크에서 성능이 향상 되는지를 실험하고 평가한 결과를 제공한다.

2장에서는 기계독해 데이터 셋과 언어모델 관련연구를 살펴보고, 3장에서는 특허상담 질의응답 데이터 구축 내용을 소개한다. 4장에서는 BERT를 이용한 한국어 특허상담 기계독해 실험을 분석하고, 5장에서는 결론 및 향후 방향을 제시한다.

2. 관련 연구

2.1 한국어 질의응답 데이터 셋

기계독해 테스크를 위한 데이터 셋으로는 영문으로 된 SQuAD(Stanford Question Answering Dataset)[1]가 대표적이며, 한국어 데이터 셋으로는 SQuAD를 벤치마킹하여 구축한 KorQuAD(the Korean Question Answering Dataset)[2]가 대표적인 표준 데이터 셋이다.

KorQuAD는 한국어 언어처리 연구에서 학습 데이터로 쓰이면서 다양한 논문에 인용되고 있고, Leaderboard에서 데이터 셋을 이용한 언어모델 평가에 대한 객관적인 지표로 활발하게 진행되고 있는 만큼 기계독해 데이터의 표준성이 확보 되어있다.

KorQuAD는 위키백과 문서를 대상으로 문단을 정제하여 질의와 정답을 생성한 일반상식분야에 대한 데이터 셋으로 training set 60,407건, dev set 5,774건으로 Context-Question-Answer 으로 구성이 되어 있다.

2.2 BERT 기반 기계독해

최근 인공지능 자연어처리 분야에서는 Google에서 공개한 고성능의 pre-training language representation model인 BERT(pre-training of Deep Bidirectional Transformers for Language Understanding)[3] 모델이 기계독해 태스크 뿐만 아니라 다른 11개의 자연어처리 공개 태스크(GLUE Benchmark])[4]에서 모든 기록을 갱신하며 높은 성능을 입증하였다.

한국어 포함 103 languages 언어처리가 가능한 BERT-base Multilingual Cased 모델로 대형 코퍼스에서 unsupervised learning으로 general-purpose language understanding 모델을 구축하고 supervised learning으로 fine-tuning 하여 태스크를 해결한다. 자연어처리에서 일반적으로 사용하는 recurrent neural network 방식과 달리 BERT 모델은 recurrent 하지 않아서 장기 의존성 문제점(Long term dependency)을 해결하였고, 더 좋은 성능을 낼 수 있는 양방향성을 가진 transformer의 인코더를 사용한 self-attention mechanism 모델링[5] 기법이다.

BERT[3]의 학습 방법은 두 가지가 있는데 첫 번째 masked language model(MLM)는 앞의 n개의 단어를 가지고 뒤에 단어를 예측하는 일반적인 unidirectional 방식과 달리 input전체의 token 중 일정 비율의 token을 masking 하고 input 전체와 mask된 token을 한번에 transformer encoder 구조에 넣어서 주변 단어의 context만을 보고 mask된 단어를 예측하는 deep bidirectional 학습방식이다. 두 번째 next sentence prediction(NSP) 방식은 두 문장에 대해서 두 번째 문장이 코퍼스 내에서 첫 번째 문장의 바로 다음에 오는지 여부를 예측하는 학습방법이다. 위 두 가지 학습방식을 이용하여 BERT 모델의 마지막 transformer layer에 기계독해 태스크를 위한 자질을 추가하고 fine-tuning 함으로써 질의응답 문제를 해결한다. 이는 Question에 정답이 되는 Context의 start vector($S \in \mathbb{R}^H$)와 end vector($E \in \mathbb{R}^H$)를 fine-tuning 하여 지문의 각 token들과 scalar product하여 시작과 끝을 찾는 태스크로 문제를 해결하는 것으로 기존에 학습되어져 있는 모델을 기반으로 특정 태스크에 적합한 데이터를 학습하고 모델을 변형하여 학습된 모델의 가중치(weights)를 업데이트 하는 representation learning 방법이다.

논문 [6]에서 제안하는 것은 max sequence length 파라미터에 의해 학습에 사용하지 않는 문자열이 발생하는 문제가 있는데 길이가 512보다 긴 시퀀스 경우 한 번에 처리할 수 있도록 개선한 Multi-level attention에 Co-attention과 Fusion 함수를 결합하여 적용한 모델을 제안하여 F1 92.43% 정확도를 달성 하였다. 논문 [7]에서는 ETRI에서 공개한 BERT 모델에 자질(exact match, term frequency, NER)과 SRU(Simple Recurrent Unit)[8]을 추가한 기법을 제안하여 F1 93.04% 정확도를 보였으나, 본 논문에서는 BERT 모델 종류와 코퍼스의 종류 및 양에 따른 정량적인 비교실험이 아니어서 적합하지 않다고 판단하였고 특허상담을 위한 실험을 위해 BERT-base 모델에서 자체적으로 구축한 한국어 코퍼스 및 patent 코퍼스를 학습한 것으로 비교실험을 진행하였다.

또한 BERT 공개 이후 XLNet[9], Albert[10] 등 개선된 모델을 공개하였지만 대용량 한국어 코퍼스로 pre-training 하여 실험 해본 결과 한국어 언어처리에 적합한 다국어처리 모델인 BERT-base Multilingual Cased 모델이 더 좋은 성능을 보였기 때문에 BERT-base Multilingual Cased 모델을 사용하였고, 특허상담분야에서 성능 향상을 위해 모델학습 과정에서 최적화를 이룰 수 있는 한국어 언어처리 기법을 제안한다.

3. 특허상담 질의응답 데이터 셋

KorQuAD[2] 데이터 셋은 특허 법률이 개정되어 문서를 수정하거나 문서의 분류가 필요할 때 사용할 수 있는 옵션이 없고, 특허분야의 특화된 전문용어나 특허 법률정보가 포함되어 있지 않아 전문적인 질의에는 정확한 정답을 추론하기가 어렵다. 그래서 특허상담분야에 대한 기계독해 학습 데이터로는 적합하지 않다고 판단하였다.

본 논문에서는 특허상담 데이터 셋의 수집, 정제, 포맷, 글자길이 등은 기계독해 실험으로 최적의 값을 얻은 적합한 범위를 선정하여 데이터 셋 구축 가이드라인으로 제안한다.

특허상담 데이터 셋은 특허고객상담 업무를 하거나 특허에 관한 기본적인 지식이 있는 작업자가 수집 된 문서를 통해 분류, 질의-정답 셋을 수작업으로 구축한다.

1차로 전문상담사가 질의의 비중이 높은 분야별로 절차를 구분하고 해당 절차에 대한 상담이력데이터와 특허법령 및 관련문서를 수집·정제하여 Title, Category, Context, Question과 Answer을 생성한다. 생성한 데이터를 작업자에게 전달하면 작업자는 데이터 셋 포맷에 따라 가공 후 Context, Question, Answer에 대한 맞춤법 검사, 불용어 처리, 생성 기준적용 등 2차로 정제하고 생성한 데이터 셋에 대해 정확한 정보인지 전문상담사의 3차 검수를 거쳐 질의응답 셋을 구축한다.

특허상담 데이터 셋의 수집 대상문서는 특허고객 상담센터의 상담이력 데이터, 특허법령 및 출원 · 심사 · 등록 · 심판 등 지식재산권별 절차와 질의응답으로 서술 되어있는 특허고객 상담사례집을 수집하였고 단순 질의유형 및 질의비중이 높은 산업재산권제도〉등록절차〉수수료〉중간절차〉심사절차〉심판절차 순으로 대상을 선정하였다.

데이터 셋 포맷은 Context-Question-Answer을 기본으로 Id, Title, Category로 구성하고, Id는 총 15자리로 Category 속성번호-Context번호-Question번호로 구성한다. Category는 대분류-중분류-소분류로 구조적인 절차의 형식으로 작성하여 분류 및 법률이 개정되어도 Id와 Category를 활용해서 쉽게 수정이 가능하도록 하였다. 또한 하나의 Context안에 같은 용어의 Answer가 여러 개 있는 경우 해당 위치의 정답을 정확하게 예측하도록 모델을 학습시키기 위해서 최초 데이터 생성 시 Context 내 정답위치에 정답태그(〈|| 〉)를 부착한다. 이 정답태그는 프로그램으로 학습 데이터 변환 시 정답위치의 index값으로 사용한 뒤 제거한다. Table 1은 특허상담 데이터 셋 예시이다.

Context는 300자 미만의 짧은 문단들을 제거하고 300~3,000자 정도의 일반적인 문장 형태이며, 질의와 정답이 Context를 통해 해석되고, 일괄된 문맥을 유지하도록 재생성 한다. 특수문자의 경우 따옴표(''), 쌍따옴표(""), 하이픈(-) 등 되도록 통일해서 사용하도록 했으며, 그 외의 특수문자, 이미지, 표는 제거하도록 한다.

Question는 하나의 Context에 대해 최소 10개 이상의 질의를 생성하도록 하였고, 3가지 유형으로 나누어서 구축 하였다.

Table 1. Example of Patent Consultation Training Data Set

Type		Contents
1	ID	003002002-001-042
2	Title	신규설정등록
3	Category	등록절차-특허·실용신안-신규설정등록
4	Context	신규설정등록이란 …(중략)… 등록번호는 납부서를 제출한 날로부터 2일에서 3일 후에 알 수 있으며, 등록증은 약 〈|7일에서 8일|〉 정도 지나면 …(중략)…
5	Question	납부서 제출일로부터 언제 정도에 등록증을 받아 볼 수 있나?
6	Answer	7일에서 8일
7	Index	156

빈칸 채우기 정도의 낮은 수준의 유형부터 문단 내에 없는 단어를 사용하여 단서가 부족한 질의에 대한 추론을 요구하는 어려운 유형의 질의로 다양한 질의를 생성하도록 하였으며, 질의의 글자 수는 5~100자로 제한한다.

Answer는 기계가 Context 내에서 Answer을 찾아내야 하므로 1~45자 이내의 단답형 또는 간결한 서술형 형태로 반드시 Context 내 동일한 용어를 사용한다.

전체 데이터 셋에 대해 Category를 대분류 기준으로 분류한 결과 산재권 제도(51.16%), 등록절차(43.27%), 수수료(3.24%)로 구성하였고, 분류별 통계는 Table 2와 같다.

Table 3은 질의 난이도에 따른 type별 내용이며, 4장에서 제시한 특허상담 데이터 셋에 최적화한 모델에서 type별로 기계독해 평가를 하였고 그 결과는 Fig. 1과 같다.

Table 2. Statistics of Dataset Category

Main Category	Sub Category	Ratio
Industrial property right system	Patent·Utility model	22.69%
	Trademark	17.00%
	Design	11.46%
Registration procedure	Industrial property right-Trademark	3.18%
	Patent·Utility model	7.89%
	Registration and Common	32.21%
Fees	Fees payment	1.70%
	Fees refund and Payment update	1.55%
Interim procedure	Interim common	1.05%
Examination procedure	Patent·Utility model	0.37%
	Trademark	0.30%
Trial procedure	Inter parte cases	0.28%
	Ex parte cases	0.33%

Fig. 1에서 성능평가 지표는 EM(Exact Match)와 F1 score로 EM score는 한국어 기준 정답의 어절단위가 정확하게 일치하는 정도이고, F1 score는 예측한 정답과 실제 정답 간의 정밀도와 재현율의 조화평균 값이다. 이후 실험에 대한 모든 성능평가 지표는 EM, F1 score로 평가를 한다.

정답의 문자 길이는 '매건 10,000원', '원출원의 지정상품이 포괄명칭인 경우', '디자인 이의신청에 의한 취소결정, 무

Table 3. Types of Patent Query

Type	Contents
A	the query type that level of filling a blank with maintains the order and words in the sentence
B	the query type that can be found out an answer only by using the words in the context or by understanding the sentences before and after
C	the query type that uses words that are not in context and lacks clues, or requires logical inference by understanding multiple sentences

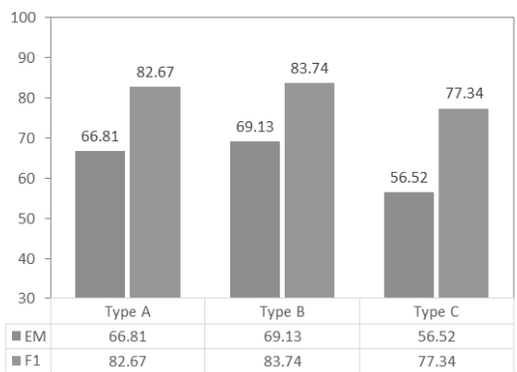

Fig. 1. Performance Evaluation by Types

효심판청구에 의한 무효심결이 확정된 경우' 등 단답형이 아닌 대부분 긴 문자열이며, 정답 문자길이에 따른 구축건수는 Table 4에서 확인한다.

Table 4. Number of Answer Length

Answer Length	Count
~10	4,452
11~20	1,135
21~30	283
30~	141

Table 5는 특허상담 질의응답 데이터 셋의 구축 현황이다. 총 6,011건의 특허상담 질의응답 데이터 셋을 구축하였고, 모델학습을 위해 training set, dev set을 9대1로 분할하여 본 논문 실험에 적용한다.

Table 5. Number of Patent Consultation Data Set

	Title	Context	Question	Answer
training set	51	197	5,401	5,401
dev set	15	24	610	610

또한 특허분야의 pre-training 모델학습을 위해서는 patent 코퍼스를 별도로 마련하였다.

Patent 코퍼스는 특허상담 데이터 셋의 수집대상 문서인 특허고객 상담센터의 상담이력 데이터, 특허법령 및 출원 · 심사 · 등록 · 심판 등 지식재산권별 절차와 질의응답으로 서술 되어있는 특허고객 상담사례집을 기반으로 한다. 특수문자, 이미지, 표를 제외한 context에 해당하는 가장 하위 레벨의 내용을 추출하고 정제과정을 거쳐 기초 데이터를 구축 후 NLTK(Natural Language Tool Kit) sentence tokenizer를 이용해 문장분리를 한다. 한 줄에 한 문장씩 위치시키고, 한 문단이 끝나면 문단 간 구분을 위해 공백 줄을 삽입시키는 과정을 반복적으로 실행함으로써 총 5,780 문장의 학습 코퍼스를 구축하였다.

4. BERT기반 특허상담 기계독해 모델

기계독해 언어모델은 BERT-base Multilingual Cased 모델(transformer block:12, self-attention head:12, hidden size:768, activate function:gelu, vocab size:119547)을 base 모델로 사용한다. 실험은 pre-training에서는 base 모델, base 모델을 기반으로 patent 코퍼스를 추가 학습한 모델, base 모델을 기반으로 한국어 위키백과 코퍼스를 추가 학습한 모델 등 코퍼스 종류에 따른 성능 평가, fine-tuning에서는 한국어 tokenization 기법, 하이퍼 파라미터 최적화 여부에 따른 성능 평가, 제안한 언어처리 알고리즘 적용 여부에 따른 성능 평가로 나누어서 비교 실험을 진행한다. 이러한 유형별 비교 실험을 통해 특허상담분야에서 기계독해 적용을 위한 최적의 방법을 도출하고자 하며, Fig. 2는 본 논문에서 실험하는 유형별 실험 흐름도이다.

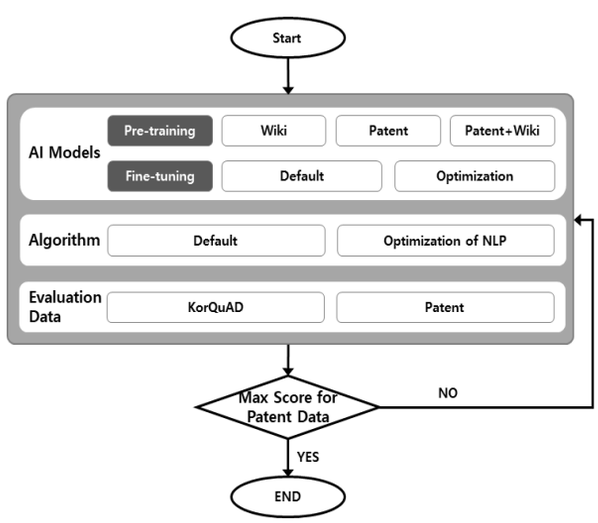

Fig. 2. Flow Chart of Experiment for Patent Consultation MRC

연구에 사용한 장비는 OS : Ubuntu 16.04, CPU : 24 cores, Memory : 128GB, GPU : NVIDIA Tesla P100 12GB * 2개를 사용하여 실험하였다.

4.1 기계독해 모델학습

특허상담 데이터 셋 및 pre-training 코퍼스에 대한 최적화 방안에 대한 연구이므로 Google에서 공개한 BERT fine-tuning의 기본 설정 값을 적용하여 학습한 모델의 결과를 유형별

로 baseline으로 지정 하였다. 기본 설정 값은word 단위의 tokenizing 방식은 basic tokenizer로 하고, sub word 단위는 WPM(Word Piece Model)[11], 입력 시퀀스 최대 길이는 128, 문단 stride는 64, 쿼리 최대 길이는 64, optimizer는 adam[12], learning rate는 3e-5, 정답 최대길이는 30으로 설정한다.

기본 설정 값에서 KorQuAD 데이터에 대한 기계독해 평가는 실험 장비의 하드웨어 성능에 맞추어 batch size을 32로 학습하여 EM 66.73%, F1 86.99%로 나왔다. 특허상담 데이터 셋에 대한 기계독해 평가는 EM 31.28%, F1 58.00%가 나왔다.

공개된 pre-trained base 모델은 일반상식 분야의 위키백과로 학습한 모델이기 때문에 특허상담 데이터 셋 평가에서는 KorQuAD 보다 훨씬 못 미치는 평가 결과가 나왔다.

Table 6에서 baseline의 score을 보여주고 있으며, 해당 score을 기준으로 이후 실험 결과와 비교·분석 한다.

Table 6. Scores of Baseline Model

Train Data		Pre-train	Fine-tune	Score	
				EM	F1
base line	KorQuAD	base	base	66.73	86.99
	patent dataset	base	base	31.28	58.00

Pre-trained base 모델에서 하이퍼 파라미터 설정만으로 fine-tuning하고 평가를 진행하여 최적의 결과 값을 도출한다. 실험을 통해 얻은 최적의 하이퍼 파라미터는 word 단위의 tokenizing 방식은 basic tokenizer이고, sub word 단위는 WPM으로 하였다. 입력 시퀀스 최대 길이는 128, 문단 stride는 64, 쿼리 최대 길이는 64, optimizer는 adam, learning rate는 5e-6, 정답 최대길이는 30으로 설정하였다. Table 7은 KorQuAD와 특허상담 데이터 셋에 대한 fine-tuning 결과이다.

Table 7. Scores of Fine Tuned Model

Train Data	Pre-train	Fine-tune	Score	
			EM	F1
KorQuAD	base	tuning	67.21	87.48
patent dataset	base	tuning	34.43	63.91

특허상담 데이터 셋 기계독해 평가에서 baseline 대비 EM은 31.28%에서 34.43%로 상승하였고, F1은 58%에서 63.91%로 크게 상승 하였다.

KorQuAD는 EM 66.73%에서 67.21%, F1은 86.99%에서 87.48%로 KorQuAD의 상승 폭 보다 특허상담 데이터 셋의 상승 폭이 더 높은 것으로 보아 특허상담 데이터 셋에 맞는 fine-tuning이라고 할 수 있다. 이후 이어지는 다른 실험에서도 설정한 하이퍼 파라미터 값으로 좋은 결과를 얻었기 때문에 특허분야 데이터 셋에 대한 최적 설정 값으로 제안한다.

Fine-tuning 실험으로 성능 향상을 확인 후 한국어 코퍼스를 학습하는 pre-training에 대한 실험을 진행한다.

Pre-training에 사용할 코퍼스의 유형과 양에 대한 비교 실험을 위해 수집한 한국어 위키백과 472만 문장의 코퍼스를 학습한다.

Google Cloud Platform과 Google Colaboratory를 이용하였고 train batch size는 16, max sequence length는 128, max predictions per seq는 20, learning rate는 3e-5로 설정하여 최적의 학습률을 얻기 위해 6일간의 학습 기간을 거쳐 400만 global step에서 masked lm accuracy는 86%, next sentence prediction은 100%을 달성한 pre-trained Patent 모델을 사용하였다.

Table 8은 위키백과 코퍼스로 pre-training한 모델로 특허상담 데이터 셋을 기계독해 평가한 결과이다.

Table 8. Scores of Pre-trained Wiki Model

Train Data	Pre-train	Fine-tune	Score	
			EM	F1
KorQuAD	wiki corpus	base	67.68	87.59
		tuning	69.41	89.03
patent dataset	wiki corpus	base	36.63	64.21
		tuning	31.60	59.04

위키백과 코퍼스로 pre-training한 모델에서 최적의 하이퍼 파라미터 설정 값으로 fine-tuning한 결과 KorQuAD 평가에서는 EM은 67.68%에서 69.41%로, F1은 87.59%에서 89.03%로 소폭 상승하지만, 특허상담 데이터 셋 기계독해 평가에서는 Table 7의 결과처럼 EM은 36.63%에서 31.60%로 하락하였고, F1도 64.21%에서 59.04%로 크게 하락하였다.

Pre-trained base 모델과 pre-trained wiki 모델에서 최적화한 fine-tuning을 진행한 결과에도 EM은 34.43%에서 31.60%로 하락하였고, F1도 63.91%에서 59.04%로 하락하였다.

Fig. 3은 baseline을 기준으로 fine-tuning 평가와 pre-trained wiki 모델의 기계독해 평가 결과이다. 일반상식분야의 위키백과를 학습한 pre-trained wiki 모델에서의 평가 결과는 특허상담 데이터 셋에서 성능이 상승한 것으로 확인했던 fine-tuning의 설정 값을 적용 하였지만, baseline 보다는 점수가 높게 나왔고, pre-trained base 모델에서 fine-tuning으로 최적화한 모델 보다 낮게 나왔다.

Fig. 3 결과에 따라 pre-training에 사용한 코퍼스 종류별로 평가 결과에 미치는 영향은 4.3장에서 소개하고 4.2장에서 모델학습 진행 과정에서 학습 데이터 셋에 대한 tokenization 및 input embedding 방식을 한국어 데이터 셋에 맞는 한국어 언어처리 알고리즘 개선을 통한 성능 변화를 실험한다.

4.2 한국어 언어처리 알고리즘

BERT의 tokenization 방식은 word 단위와 sub word 단위로 이루어져 있다. 이 중 Google에서 공개한 word 단위 tokenization 방식은 영문의 텍스트를 기반으로 하여, 띄어쓰기를 기준으로 각 word 들을 tokenization 하는 방식(Basic Tokenizer)을 사용하고 있다. 하지만 한국어의 경우 word는 조사를 비롯한 여러 품사들이 단어와 같이 사용되는

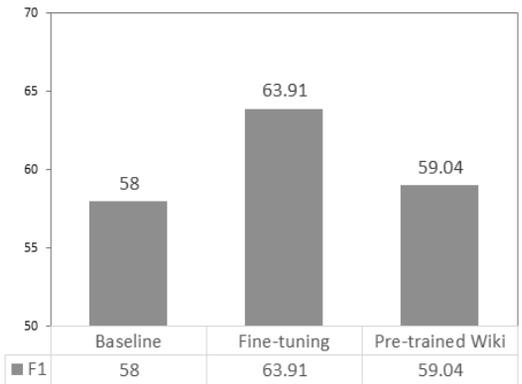

Fig. 3. Scores of Each Model for Patent Dataset

특성이 존재하므로, 본 논문에서는 한국어 특화된 word 단위 tokeniation을 위해서 한국어 형태소 분석기를 사용하기로 한다. 한국어 형태소 분석기는 OKT, Mecab, KhAiii 등이 있으며, 형태소 분석기를 혼합하여 사용할 수도 있다.

Table 9는 특허상담 데이터 셋의 context에서 각 tokenizer에 의해 tokenization 한 후 answer index로 추출한 token들과 real answer token들과 비교하여 불일치한 answer을 측정한 error count 비교이다.

Table 9. Comparison of Word Error Count by Tokenizer

Tokenizer	Training Set		Dev Set	
	Word	Sub Word	Word	Sub Word
WPM (Baseline)	37,126	28,634	3,448	3,677
OKT	2,500	2,593	205	312
KHaiii	2,056	1,155	192	170
Khaiii+Mecab	460	767	45	86
Mecab	472	744	51	86

Error count가 적으면서 가장 빠르며 학습 평가결과가 높게 나온 Mecab을 tokenizer로 사용하여 실험한다.

한국어 형태소 분석기를 통해 tokenization을 하더라도 불용어 및 신조어, 전문용어 등에 대한 대응이 어렵고 특허상담 코퍼스를 구성하는 용어들은 일반상식분야의 위키백과에 등장하는 용어와 상이한 부분이 많아 용어 인식의 탈락이 발생한다. Mecab tokenizer의 단순 사용으로는 임베딩 과정에서 tokenizing한 answer를 복원 후 real answer와 비교하여 불일치가 되는 경우 학습률을 떨어뜨리기 때문에 결과적으로 기계독해 평가결과를 하락시키는 문제가 발생한다.

본 논문에서는 Input 데이터에 대한 임베딩 방식을 개선하여 학습률을 상승 시키는 한국어 언어처리 알고리즘을 제안한다. 이 방식을 ReTE(ReTokenizing for Input Embedding)로 지칭한다.

BERT에서 input embedding 과정은 token embeddings, position embeddings, segment embeddings를 추가해 각각의 합산한 결과를 취한다.

ReTE는 token embedding 단계에서 적용 하는데 먼저 데이터 셋의 context에서 answer index에 따라 answer 영역을 추출한다. 이후 input으로 보낸 후 tokenizer로 tokenizing하고 다시 복원과정을 거쳐 real answer와 비교를 한다. Mecab tokenizer는 용어 앞뒤로 등장하는 용어에 따라 연접비용을 계산하여 가장 적합하게 tokenizing 하기 때문에 같은 용어라도 다르게 tokenizing 될 수 있다. 이를 개선하고자 불일치 된 Token이 발생하면 불일치된 input token 영역을 공백 없이 하나의 문자열로 붙인다. real answer와 문자열을 비교한 후 일치하는 문자열을 기준으로 left text, real answer, right text로 분리한다. 이후 각기 분리된 text들을 re-tokenizing하여 전체 index token 영역에 위치시키도록 재구성한다. 이렇게 만들어진 재구성된 token들은 input embedding 과정을 지나도록 한다.

결론적으로 데이터 셋의 context에서 answer로 지정된 token의 복원된 값들이 real answer와 일치하도록 하여 용어 인식의 탈락을 최소화하여 모델학습을 하는 것으로 성능을 올릴 수 있다. 다만, prediction 단계에서는 정답을 미리 알 수 없기 때문에 ReTE 알고리즘을 적용하지 않지만 ReTE를 통해 성능이 올라간 모델을 사용하여 prediction을 하기 때문에 결국 전체 평가에서 성능이 향상된 결과를 보인다.

Fig. 4는 ReTE 기법을 사용하여 input token이 real answer처럼 tokenizing되어 input embedding 단계로 들어가는 모습이고, Fig. 5는 input 데이터의 임베딩 방식을 개선하기 위한 input token 결정 과정을 나타낸 것 이다.

특허상담 데이터 셋 평가에서는 pre-trained base 모델에서 최적화한 fine-tuning 평가와 pre-trained wiki 모델에서 Mecab tokenizer와 ReTE 기법을 적용한 평가와 비교실험을 한다.

ReTE 기법을 적용함으로써 pre-trained base 모델에서 EM은 34.43%에서 65.88%로 상승하였고, F1은 63.91%에서 81.90%으로 크게 상승하였다. pre-trained wiki 모델에서는 EM은 31.60%에서 64.30%로 상승하였고, F1은 59.04%에서 81.68%로 큰 폭으로 상승하였다는 것을 알 수 있다. ReTE 기법을 KorQuAD에 적용해 본 결과 pre-trained wiki 모델에서 EM은 69.41%에서 85.38%, F1은 89.03%에서 93.42%로 큰 폭으로 상승하면서 가장 높은 점수를 기록하였다.

특허상담 데이터 셋의 경우 데이터 특성에 따라 token error count가 많이 발생하였기 때문에 ReTE를 적용함으로써 높은 성능 효과를 볼 수 있었다. 일반상식분야의 데이터 셋 상승 폭보다 더 높은 상승 폭의 결과가 나왔다. 그러나 pre-trained base 모델에서의 평가와 pre-trained wiki 모델에서 평가를 F1 기준으로 보면 81.90%에서 81.68%로 소폭 하락하였다.

Table 10에서 pre-trained 모델 종류와 ReTE 기법 활용 여부에 따른 성능평가 결과이다.

pre-trained wiki 모델은 대용량 한국어 코퍼스를 학습한 모델임에도 base 모델과 큰 차이가 없었다는 것은 대용량의 코퍼스가 반드시 좋은 성능을 낼 수 있을 것이라고 판단할 수 없다. 그에 따라 patent 코퍼스로 학습한 pre-trained patent 모델로 실험을 진행하기로 한다.

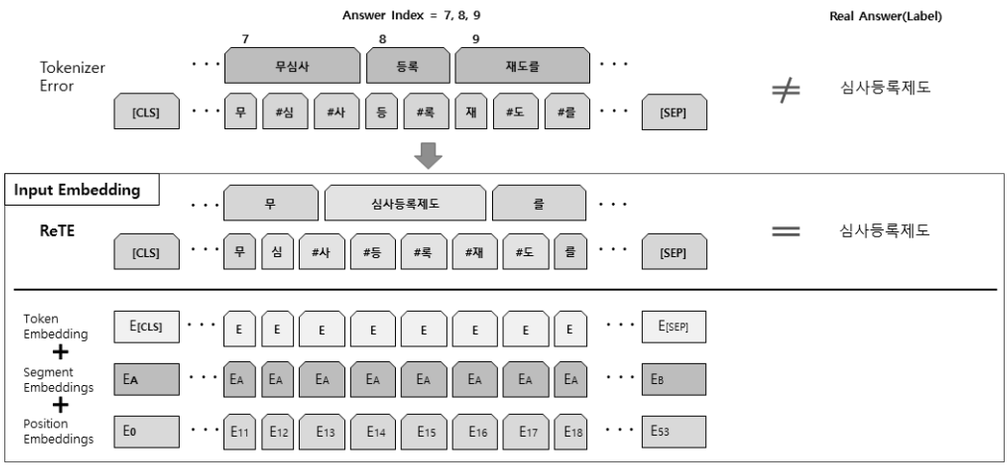

Fig. 4. Example of Applying ReTE in Input Embedding

Algorithm1 ReTE Algorithm

Input: $X_1 \in$ Input BERT Token set
Set T = Number of Input data set, A_{real} = Real Answer Text
for t=0 **to** T **do**
  A_{norm} = Normalize Answer form X_t
  Compare A_{norm} with A_{real}
  IF equal
    Then pass
  Else
    C_t = Concatenate X_t
    S_{left}, S_{ans}, S_{right} = Split C_t on both sides based on A_{real}
    $ReTE_t$ = Join (Each Tokenization (S_{left}, S_{ans}, S_{right}))
  Replace X_t to $ReTE_t$
end for

Fig. 5. Pseudo Code of ReTE Algorithm

Table 10. Scores of ReTE applied Model

Train Data	Pre-train	Fine-tune	Score	
			EM	F1
KorQuAD	wiki corpus	tuning	69.41	89.03
		ReTE	85.38	93.42
patent dataset	base	tuning	34.43	63.91
		ReTE	65.88	81.90
	wiki corpus	tuning	31.60	59.04
		ReTE	64.30	81.68

4.3 특허상담 기계독해 학습

위키백과 코퍼스를 학습한 pre-trained wiki 모델과 비교 실험을 위해 patent 코퍼스를 사용하여 pre-trained patent 모델과 pre-trained wiki+patent 모델을 생성한다.

Google Cloud Platform과 Google Colaboratory를 이용하고 pre-trained patent 모델은 train batch size는 16, max sequence length는 128, max predictions per seq는 20, learning rate는 3e-5로 설정한다. 30만 global step에서 masked lm accuracy와 next sentence prediction은 100%에 근접한 모델을 사용한다.

pre-trained wiki+patent 모델은 4.1장에서 제안한 설정 값으로 학습한다.

Fine-tuning에서 ReTE을 적용하였고, pre-trained base 모델과 pre-trained patent 모델과의 비교 평가에서는 EM은 65.88%에서 66.50%로 상승하였고, F1도 81.90%에서 82.45%로 상승하였다. 그리고 wiki 코퍼스까지 추가로 학습한 pre-trained wiki+patent 모델에서는 큰 변화가 보이지 않았다.

결과적으로 특허상담 데이터 셋에 대한 평가는 pre-trained patent 모델과 ReTE를 적용한 실험에서 가장 높은 점수를 얻었다. Table 11은 평가결과이다.

Table 11. Scores of ReTE applied Model by Pre-Trained

Train Data	Pre-train	Fine-tune	Score	
			EM	F1
patent dataset	base	ReTE	65.88	81.90
	wiki+patent corpus	ReTE	64.62	82.01
	patent corpus	ReTE	66.50	82.45

5. 결론 및 향후 방향

본 논문에서는 특허상담분야에서 기계독해 연구를 할 수 있도록 특허상담 데이터 셋을 구축하였다. 특허상담 코퍼스와 질의응답 데이터 셋은 추가 구축 및 품질 개선이 이루어진 후 공개 할 예정이다.

기계독해 학습 데이터 셋 구축방법을 가이드라인을 통해 소개 하였고, 이를 통해 다른 산업분야에서 기계독해 태스크를 위한 학습 데이터 셋을 구축하고자 할 때 도움이 되고자 한다.

Representation language model인 BERT 모델을 활용한 특허상담분야 기계독해 실험에서는 위키백과 코퍼스를 사용하지 않고 patent 코퍼스만을 학습한 pre-trained patent 모델과 ReTE를 적용한 fine-tuning에서 EM 66.50%, F1 82.45%로 점수가 가장 높게 나왔다.

대용량 코퍼스를 pre-training하고 fine-tuning을 통해 대체적으로 성능 향상을 이루었지만, 해결하고자 하는 분야에 맞는 코퍼스로 pre-training을 하고 fine-tuning 최적화 과정을 통해 더 좋은 성능평가 결과가 나왔다는 결과를 도출하였다. 또한 모델학습 과정에서 본 논문에서 제안한 한국어 형태소 분석기(Mecab)와 한국어 언어처리 알고리즘(ReTE)을 input embedding 영역에 적용하였을 때 가장 큰 효과가 있었다.

우리는 구축한 특허상담 데이터 셋과 기계독해 실험을 통해 최적화한 노하우를 바탕으로 다양한 모습으로 진화하고 있는 자동 질의응답 연구를 지속하고자 한다. 뿐만 아니라 학습 데이터를 확장 구축하여 다양한 분야의 질의에 대응하도록 하고, 다른 개선된 심층 신경망 네트워크 적용 및 특허분야 언어처리 알고리즘 개선을 통해 기존보다 성능 향상을 기대할 수 있는 방안에 대해서 연구 할 예정이다.

References

[1] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250,* 2016.

[2] S. Lim, M. Kim, and J. Lee, "KorQuAD: Korean QA Dataset for Machine Comprehension," *in Proceedings of the Korea Software Congress 2018,* pp.539-541, 2018.

[3] D. Jacob, C. Ming-Wei, L. Kenton, and T. Kristina, "Bert: pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[4] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," a*rXiv preprint arXiv:1804.07461*, 2018.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, and L. Kaiser, "Attention is all you need," *Advances in Neural Information Processing Systems.* 2017.

[6] K. H. Park, S. H. Na, Y.S. Choi, and D. S. Chang, "BERT and Multi-level Co-Attention Fusion for Machine Reading Comprehension," *in Proceedings of the Korea Software Congress 2019,* pp.643-645, 2019.

[7] D. Lee, C. Park, C. Lee, S. Park, S. Lim, M. Kim, and J. Lee, "Korean Machine Reading Comprehension using BERT," *in Proceedings of the Korea Computer Congress 2019*, pp.557-559, 2019.

[8] T. Lei, Y. Zhang, S.I. Wang, H. Dai, and Y. Artzi. "Simple Recurrent Units for highly Parallelizable Recurrence," *arXiv:1709.02755v5*, 2018.

[9] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding," *arXiv preprint arXiv: 1906.08237*, 2019.

[10] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942,* 2019.

[11] Y. Wu, M. Schuster, Z. Chen, Q, V. Le, and M. Norouzi, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[12] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980,* 2014.

민 재 옥

https://orcid.org/0000-0003-0436-164X
e-mail : okauto@kipi.or.kr
2010년 명지대학교 컴퓨터공학과(학사)
2020년 ~ 현 재 서강대학교 AI빅데이터 이노베이션MBA 석사과정
2016년 ~ 현 재 한국특허정보원 R&D센터 연구개발파트장
관심분야 : Natural Language Processing & Patent Business

박 진 우

https://orcid.org/0000-0001-7961-6024
e-mail : znu808@kipi.or.kr
2011년 성균관대학교 컴퓨터공학과(학사)
2018년 ~ 현 재 한국특허정보원 R&D센터 연구원
관심분야 : Natural Language Processing & Pattern Recognition

조 유 정

https://orcid.org/0000-0001-5508-5310
e-mail : zzoyou12@kipi.or.kr
2019년 금오공과대학교 컴퓨터소프트웨어공학(학사)
2019년 ~ 현 재 한국특허정보원 R&D센터 연구원
관심분야 : Natural Language Processing & Text Mining

이 봉 건

https://orcid.org/0000-0001-5138-1139
e-mail : bglee@kipi.or.kr
2000년 고려대학교 전산학과(학사)
2014년 한성대학교 서비스&컨설팅(석사)
2000년 ~ 현 재 한국특허정보원 특허넷응용팀 특허넷응용팀장
관심분야 : Intellectual Property Rights & Patent Business