LUKE 기반의 한국어 문서 검색 모델

고동률[O12], 김창완[1], 김재은[1†], 박상현[2†]
㈜솔트룩스 AI Labs[1], 연세대학교 컴퓨터과학과[2]
{dongryu.ko, changwon.kim, jaieun.kim}@saltlux.com, sanghyun@yonsei.ac.kr

LUKE based Korean Dense Passage Retriever

Dongryul Ko[O12], Changwon Kim[1], Jaieun Kim[1†], Sanghyun Park[2†]
AI Labs, Saltlux Inc[1], Dept. of Computer Science, Yonsei University[2]

요 약

자연어처리 분야 중 질의응답 태스크는 전통적으로 많은 연구가 이뤄지고 있는 분야이며, 최근 밀집 벡터를 사용한 리트리버(Dense Retriever)가 성공함에 따라 위키피디아와 같은 방대한 정보를 활용하여 답변하는 오픈 도메인 QA(Open-domain Question Answering) 연구가 활발하게 진행되고 있다. 대표적인 검색 모델인 DPR(Dense Passage Retriever)은 바이 인코더(Bi-encoder) 구조의 리트리버로서, BERT 모델 기반의 질의 인코더(Query Encoder) 및 문단 인코더(Passage Encoder)를 통해 임베딩한 벡터 간의 유사도를 비교하여 문서를 검색한다. 하지만, BERT와 같이 엔티티(Entity) 정보에 대해 추가적인 학습을 하지 않은 언어모델을 기반으로 한 리트리버는 엔티티 정보가 중요한 질문에 대한 답변 성능이 저조하다. 본 논문에서는 엔티티 중심의 질문에 대한 답변 성능 향상을 위해, 엔티티를 잘 이해할 수 있는 LUKE 모델 기반의 리트리버를 제안한다. KorQuAD 1.0 데이터셋을 활용하여 한국어 리트리버의 학습 데이터셋을 구축하고, 모델별 리트리버의 검색 성능을 비교하여 제안하는 방법의 성능 향상을 입증한다.

주제어: Open-domain QA, Dense Retriever, Bi-encoder, LUKE

1. 서론

자연어처리 분야 중 질의응답 태스크(Task)는 언어모델이 발전함에 따라 가장 주목받고 있는 태스크 중 한 가지이다[1]. 또한, 주어진 문서 내에서 기계독해(MRC: Machine Reading Comprehension)를 통해 질의응답을 하는 것을 넘어, 방대한 양의 문서를 대상으로 질의와 관련된 문서를 찾아서 답변하는 오픈 도메인 QA에 대한 연구도 활발하게 진행되고 있다[2].

오픈 도메인 QA는 문서를 검색하는 리트리버와 정답 구간(Span)을 예측하는 리더로 구성되어 있으며, 리트리버의 문서 검색 성능은 QA 성능에 큰 영향을 미친다. 리트리버는 크게 희소벡터 기반의 리트리버와 밀집벡터 기반의 리트리버로 나눌 수 있으며[3,4], 최근에는 BERT, ELECTRA, roBERTa와 같은 사전학습 언어모델에 대한 연구가 활발하게 진행됨에 따라, 오픈 도메인 QA에서도 REALM(Retrieval-Augmented Language Model), DPR과 같이 사전학습 언어모델을 활용한 밀집벡터 기반의 리트리버가 많이 연구되고 있다[5,6].

하지만 밀집 벡터 기반의 리트리버는 여전히 완벽하게 희소벡터 기반의 리트리버를 대체할 수는 없다. 대표적인 밀집 벡터 기반의 리트리버인 DPR은 실제 사용자의 질문으로부터 구축한 일반적인 NQ(Natural Question)데이터셋을 대상으로 한 검색 성능은 뛰어나지만, "대한민국의 수도는 어디인가?"와 같은 엔티티 중심의 질문 위주인 EQ(Entity Question)데이터셋을 대상으로한 검색 성능은 저조하다[7,8].[1)]

한편, 최근에는 다양한 자연어처리 태스크에서 엔티티 정보를 모델에 효율적으로 학습하기 위한 연구들이 진행되고 있으며[9-11], LUKE(Language Understanding with Knowledge-based Embedding)는 마스킹 처리 과정에서 엔티티 정보를 추가로 활용함으로써 엔티티와 관련이 있는 다양한 자연어처리 태스크에서 우수한 성능을 보였다.

본 논문에서는 엔티티를 잘 이해하고 표현할 수 있는 사전학습 언어모델인 LUKE 기반의 한국어 리트리버를 제안한다. KorQuAD 1.0 데이터셋을 활용하여 한국어 리트리버 학습 데이터셋을 구축하고, 기존의 사전학습 언어모델을 사용한 밀집벡터 기반의 리트리버와 검색 성능을 비교하여 제안하는 방법의 우수함을 입증한다.

2. 관련 연구

본 논문에서 제안하는 모델의 기준 모델(baseline)인 DPR은 바이 인코더 구조이며, BERT 기반의 질의 인코더 및 문단 인코더를 통해 질의와 문단을 대표할 수 있는 [CLS] 토큰 벡터값 간의 내적 유사도를 측정하여 문서를 검색한다[6]. {q, p+, p-} 형태로 질의, 정답 문단, 오답 문단으로 이루어진 데이터셋을 사용였으며, NLL (Negative log likelihood) 손실함수를 통해 학습했다. 또한 다양한 네거티브 샘플링(Negative sampling)방법을 제안하고 실험하였으며, 인-배치 네거티브(In-batch Negative) 방법을 사용하여 가장 우수한 성능을 보였다[6]. 인-배치 네거티브는 배치 단위로 모델을 학습하고, 같은 배치 내 다른 질문들의 정답 문단을 오답 문단으로 사용하는 방법이다. 따라서, 인-배치 네거티브는 별도의 오답 문단을 선정하지 않아도 되는 효율적인 방법이다. 또한 하드 네거티브 샘플(Hard Negative Sample)을 추가하여 검색 성능을 향상시켰다.

† 공동 교신저자(co-corresponding author)

본 논문에서 사용하는 언어모델인 LUKE는 엔티티를 잘 이해하고 표현할 수 있는 모델로서, 사전학습 과정에서 문장 내의 엔티티 정보를 추가로 학습한다. 그림1은 LUKE의 학습 구조를 한국어 문장으로 표현한 예시이며, 엔티티 정보는 [SEP] 토큰 뒤에 입력하고, 토큰 임베딩(Token Emb), 위치 임베딩(Position Emb)외에 엔티티 유형 임베딩(Entity Type Emb)을 추가한다[11]. 이런 방법으로 학습한 LUKE는 엔티티 타입 정의(Entity typing), 관계추출(Relation Classification), 엔티티 식별(Named Entity Recognition), 빈칸을 맞추는 형태의 질의 응답(Cloze-style QA), 추출 기반 질의 응답(Extractive QA)에서 SOTA(State-of-the-Art)를 기록하였다[11].

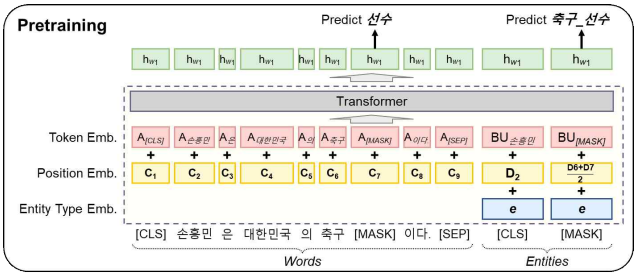

그림 1 LUKE 모델 학습 구조

3. 제안 방안

3.1 학습데이터 구축

리트리버를 학습하기 위해서는 질의-응답-문서 쌍으로 이루어진 학습데이터가 필요하다. 영어 질의응답 데이터셋 중 하나인 SQuAD는 데이터셋 구축 당시, 데이터 구축자가 정답을 알고 있는 상태에서 질문을 작성했기 때문에, 엔티티 중심의 질문이 많이 포함되어 있다는 연구결과가 있다[4]. 따라서, 엔티티 중심의 질문에 대한 성능을 비교하기 위해, SQuAD와 유사한 KorQuAD 1.0 데이터셋을 사용하여 리트리버 학습데이터를 구축하였다. 사용한 KorQuAD 1.0의 통계는 표1과 같다.

Dataset	Train/Dev	Passage	QA
KorQuAD	Train	9681	60407
	Dev	964	5774

표 1 KorQuAD 1.0 데이터셋 통계

리트리버 학습데이터의 정답 문단은 KorQuAD 1.0 데이터셋에 정의된 정답을 포함하는 문서를 사용하였으며, 언어모델의 최대 입력 길이를 고려하여 길이가 긴 문서를 문단 단위로 잘라내어 사용하였다.

KorQuAD 1.0 데이터셋은 하나의 문서에 여러 개의 질문과 답변이 존재하고, 답변은 여러 단어로 구성이 될 수 있다. 따라서, 문서는 단어 100개를 기준으로 분리하였으며, 슬라이딩 윈도우(Sliding window)방식을 적용하여, 각 문단을 25개의 단어가 중첩되도록 구축하였다. 그리고 중첩된 구간에서 정답이 있는 경우 첫 번째 문단을 정답 문단으로 선정하였다.

학습 방법으로 인-배치 네거티브 방법을 사용했기 때문에, 별도의 오답 문단은 추출하지 않았다. 그리고 성능 향상을 위해, 하드 네거티브 샘플(Hard negative sample) 추가했다. 하드 네거티브 샘플은 BM25를 통해 추출한 Top-1 문단이 정답이면 Top-2를 하드 네거티브 샘플로 사용하고, Top-1 문단이 정답이 아닌 경우에는 Top-1을 하드 네거티브 샘플로 선정했다. 구축한 학습데이터 포맷은 그림 2와 같다.

```
{
    "question": "바그너는 괴테의 파우스트를 읽고 무엇을 쓰고자 했는가?",
    "answers": ["교향곡"],
    "positive_ctxs": [{
        "title": "파우스트_서곡",
        "text": "1839년 바그너는 괴테의 파우스트를 처음 읽고... "
    }],
    "negative_ctxs": [],
    "hard_negative_ctxs": [{
        "title": "낭만주의_음악",
        "text": "바그너와 같은 시대에 그의 영향을 받은 작곡가 중에는... "
        }],
},  ...
```

그림 2 학습데이터 형태

3.2 참조문서 구축

오픈 도메인 QA는 질의에 대한 답변을 찾을 수 있는 참조문서가 필요하다. 본 논문에서는 KorQuAD 1.0의 데브셋(Dev-set)의 정답 문단을 참조문서로 사용하였으며, kowiki(한국어 위키피디아 문서) 덤프 파일을 전처리하여 참조문서를 추가 구축하였다.

리트리버의 학습데이터셋으로 사용한 KorQuAD v1.0은 kowiki(한국어 위키피디아 문서)의 문서를 대상으로 구축된 데이터셋이기 때문에, kowiki 덤프에서 추출한 문단을 노이즈 데이터로 사용하여, 많은 양의 참조문서를 대상으로 한 검색 성능을 평가하였다. 참조문서는 {id, title, passage}로 구성된 tsv 파일을 구축하고, 문서의 제목과 문단을 이어붙인 형태로 사용하였다.

3.3 리트리버 학습 및 추론

제안하는 방법은 대표적인 밀집 벡터 기반 리트리버인 DPR과 동일한 아키텍처를 사용하며, 질의 인코더와 문단 인코더의 사전학습 모델로 LUKE를 사용한다. 구축한 {q, p+, hn} 형태의 학습데이터를 활용하여, NLL 손실함수를 통해 질의와 유사한 문단을 찾을 수 있도록 학습했다. 제안하는 방법은 그림3과 같이 각 인코더의 출력값 중 [CLS] 토큰 벡터값 간의 내적 유사도를 측정하여 질의와 관련된 문서를 검색한다. 리트리버의 추론 과정은 그림3

의 오프라인에 해당하는 참조문서를 임베딩하여 벡터를 색인하는 과정이 필요하다. 색인은 ANN(Approximate Nearest Neighbor)라이브러리 중 하나인 faiss-gpu를 활용하였으며, 색인 옵션은 Flat을 사용하였다.

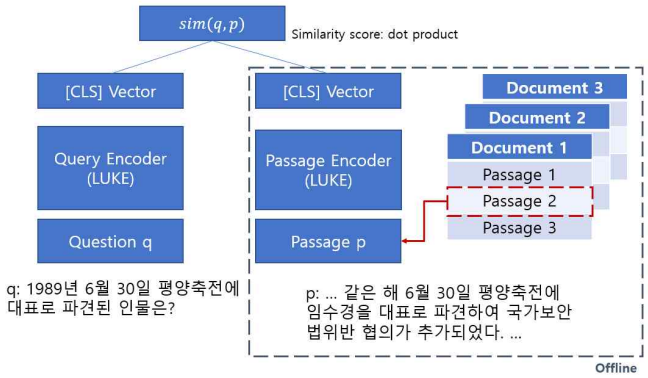

그림 3 LUKE 기반의 리트리버 구조

'Quadro RTX 8000' 48G GPU 2장을 사용했으며, 하이퍼 파라미터는 배치 사이즈 16, 아담 옵티마이저(Adam Optimizer), 에폭은(Epoch) 3회이며, 평가셋(Dev-set)을 통해 가장 높은 성능을 기록한 에폭의 체크포인트로 모델을 결정하였다.

4. 실험 환경 및 평가

언어모델별 리트리버의 검색 성능을 비교하기 위한 기준 모델로 BERT와 roBERTa 모델을 선정하여, LUKE 모델과 비교하였다. 실험 구현을 위해, 언어모델은 허깅페이스(Huggingface)의 사전학습 모델을 사용했으며, 실험 대상 모델은 모두 다국어 모델을 사용하였다. (BERT: bert-base-multilingual-cased, roBERTa: xlm-roberta -base, LUKE: studio-ousia/mluke-base)

모델 학습은 'Quadro RTX 8000' 48G GPU 2장을 사용하고, 배치 사이즈 16으로 학습하였으며, 아담 옵티마이저를 사용했다. 매 학습회차(Epoch)마다 KorQuAD 1.0의 데브셋으로 평가하여 가장 높은 성능을 기록한 회차의 체크포인트로 모델을 결정하였다.

그리고 하드 네거티브 샘플의 효과를 확인하기 위해, 각 모델을 하드 네거티브 샘플 사용 여부에 따라, 두 가지 버전으로 학습하여 비교하였다. 비교 실험은 3회씩 학습한 모델을 사용하여 KorQuAD 1.0 데브셋의 정답 문단 964개를 임베딩 및 색인하고, Top-K 메트릭을 통해 평가하였다. Top-1 기준 평가 결과 그림 4와 같이 하드 네거티브 샘플 추가하여 학습한 경우가 약 4% 정도 우수한 성능을 보이는 것을 확인할 수 있다. 또한 표 2와 같이 측정한 모든 Top-K에서 BERT, roBERTa에 비해 LUKE 기반의 리트리버가 가장 우수한 성능을 보였다.

추가적으로 대량의 문서를 대상으로 한 검색 성능을 확인하기 위해, kowiki에서 구축한 문서 37,353개를 기존 참조문서에 추가해서 성능을 평가하였다. 각 모델은 하드 네거티브를 사용하여 10회 학습하였으며, 실험결과는 표 2와 같다. 비교 결과 대량의 문서를 대상으로 한 검색 성능도 Top-1 기준으로 LUKE 기반 리트리버가 BERT보다 9%, roBERTa에 비해서는 14.87% 더 높은 성능을 보이는 것을 확인하였다.

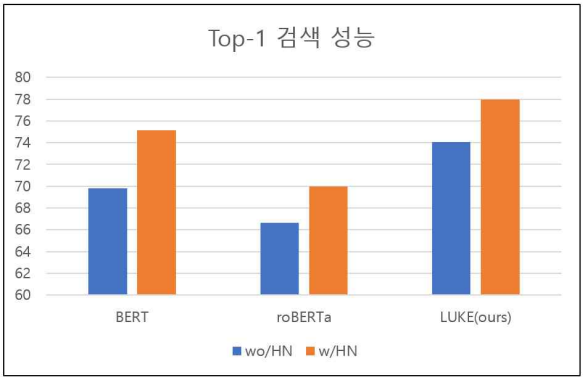

그림 4 하드 네거티브 사용여부에 따른 성능 차이

5. 결론

본 논문에서는 오픈 도메인 QA에서 질문 및 문단의 엔티티를 잘 이해하고 벡터로 표현할 수 사전학습 모델인 LUKE 기반의 리트리버를 제안하였다. 제안하는 방법의 성능을 확인하기 위해, 한국어 리트리버 학습데이터셋을 구축하고, BERT 및 roBERTa를 기반으로 한 리트리버와

Passage	Model	Epoch	HN	Top-1	Top-3	Top-5	Top-10	Top-20	Top-50	Top-100
KorQuAD(Dev)	BERT	3	WO/HN	69.79	88.17	93.12	96.36	98.21	99.06	99.37
	roBERTa			66.64	85.08	90.68	98.32	97.83	99.15	99.51
	LUKE(ours)			**74.07**	**90.37**	**94.05**	**96.81**	**98.57**	**99.23**	**99.58**
	BERT	3	W/HN	75.15	90.23	93.95	96.69	98.23	99.18	99.48
	roBERTa			70.00	87.25	92.22	96.06	98.11	99.25	99.46
	LUKE(ours)			**77.98**	**91.63**	**94.42**	**97.19**	**98.63**	**99.37**	**99.67**
KorQuAD(Dev) + kowiki	BERT	10	W/HN	62.61	80.14	85.87	90.65	93.47	96.21	97.51
	roBERTa			56.74	75.86	81.36	87.63	92.22	95.90	97.51
	LUKE(ours)			**71.61**	**85.83**	**89.42**	**92.59**	**95.08**	**97.38**	**98.27**

표 2 모델별 문서 검색 추론 결과

비교하였으며, 하드 네거티브 샘플 및 노이즈 데이터 포함 여부와 관계없이 LUKE 기반의 리트리버가 BERT와 roBERTa 기반의 리트리버보다 우수한 성능을 보이는 것을 확인했다.

향후 연구로 다국어 사전학습 모델을 사용하지 않고, 한국어 언어모델을 기반으로 엔티티 정보를 추가하여 학습한 한국어 LUKE 기반의 리트리버를 구현할 계획이다. 또한, 질의와 문단의 특징을 고려하여 인코더의 언어모델 선정하고 학습할 계획이다. 예를 들어, 질의 인코더는 문장 표현에 최적화된 모델을 사용하고 문단 인코더는 LUKE 모델을 사용하는 것과 같이, 질의 표현용 인코더와 문단 표현용 인코더를 별도로 사용하는 리트리버에 대한 연구를 진행할 계획이다.

참고문헌

[1] Liu, Shanshan, et al. "Neural machine reading comprehension: Methods and trends." Applied Sciences 9.18 (2019): 3698.

[2] Zhu, Fengbin, et al. "Retrieving and reading: A comprehensive survey on open-domain question answering." arXiv preprint arXiv:2101.00774 (2021).

[3] Danqi Chen, Adam Fisch, Jason Weston, Antoine Bordes, “Reading Wikipedia to Answer Open-Domain Questions”, ACL 2017

[4] Kenton Lee, Ming-Wei Chang, Kristina Toutanova, “Latent Retrieval for Weakly Supervised Open Domain Question Answering”, ACL 2019

[5] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, Ming-Wei Chang.“REALM: Retrieval-Augmented Language Model Pre-Training”,ICML 2020

[6] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, Wen-tau Yih, “Dense Passage Retrieval for Open-Domain Question Answering”, EMNLP 2020

[7] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, Slav Petrov, “Natural Questions: A Benchmark for Question Answering Research”, TACL 2019

[8] Christoper Sciavolino, zexuan Zhong, Jinhyuk Lee, Danqi Chen, “Simple Entity-Centric Questions Challenge Dense Retrievers”, EMNLP 2021

[9] Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, Luke Zettlemoyer, “Scalable Zero-shot Entity Linking with Dense Entity Retrieval”, EMNLP 2020

[10] Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, Tom Kwiatkowski, “Entities as Experts: Sparse Memory Access with Entity Supervision”, EMNLP 2020

[11] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, Yuji Matsumoto, “LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention”, EMNLP 2020