계층적 검색 모델을 이용한 정답 문장 탐색

최승호○, 전현규, 김지윤, 김봉수

와이즈넛

{csh1019, eddie14 , jiyoonkim, usgnob}@wisenut.co.kr

Exploring Answer Sentences using Hierarchical Retrieval Models

Seungho Choi○, Hyun-Kyu Jeon, Jiyoon Kim, Bongsu Kim

Wisenut Inc.

요 약

오픈 도메인 질의응답 (ODQA, Open-Domain Question Answering)은 주어진 질문에 대한 답을 찾는 작업으로 일반적으로 질문과 관련 있는 지식을 검색 모델(Retrieval)을 통해 찾는 단계와, 찾은 지식에서 문서의 정답을 독해 모델(Reader)을 이용하여 찾는 단계로 구성되어 있다. 본 논문은 기존의 DPR(Dense Passage Retrieval)을 이용한 복수의 검색 모델(Retrieval)만을 계층적으로 사용하여 독해 모델(Reader)을 사용하지 않고 정답 문장을 찾는 방법과 정답 문장을 찾는 데 특화된 검색 모델 학습을 위한 유효한 성능 향상을 보이는 Hard Negative Sampling 기법을 제안한다. 해당 제안기법을 적용한 결과, 동일 조건에서 학습된 검색 - 독해(Retrieval-Reader) 구조의 베이스라인 모델보다 EM에서 12%, F1에서 10%의 성능 향상을 보였다.

주제어: ODQA, Retrieval, Hard Negative Sampling

1. 서론

오픈 도메인 질의응답 (ODQA, Open-Domain Question Answering)은 주어진 질문에 대한 답을 찾는 작업으로 일반적으로 질문과 관련 있는 지식을 찾는 단계와, 찾은 지식에서 질문의 정답을 찾는 단계로 구성되어 있다. 본 논문은 KorQuAD 데이터[1]를 이용하여 기존의 검색 - 독해 (Retrieval - Reader) 파이프라인이 아닌, 독해 모델(Reader)을 사용하지 않는, 복수의 검색 모델(Retrieval)만을 계층적으로 사용하여 정답 문장을 찾는 질의응답 기법을 제안한다.

텍스트 ODQA에서는 질문과 연관된 문서를 검색하기 위해 BM25, TF-IDF 등과 같은 Sparse Embedding을 이용하는 방법을 사용하였고, 이러한 방법들은 질문과 중복되는 단어가 많은 문서가 검색되는 경향이 있으며, 이는 질문과 문서가 실제로 얼마나 유사한지에 대한 의미적 영향은 고려하지 못하는 문제가 있다. 이를 해결하기 위해 Dense Embedding을 이용하는 DPR(Dense Passage Retrieval)[2] 구조를 사용한 검색 모델을 사용하였다.

검색 모델을 이용하여 질문과 연관된 문서를 찾은 뒤, 일반적인 ODQA에서는 해당 문서와 질문을 독해 모델에 입력하여 질문에 대한 답을 얻는 과정이 존재한다. 이 경우 독해 모델에는 인코더 구조를 이용한 답변의 위치를 탐색하는 방법과, GPT와 같은 생성형 모델, 디코더 구조를 이용하여 질문에 대한 답변을 생성하는 방법이 있다.

본 논문에서는 생성형 답변 모델이 아닌, BERT 구조의 답의 위치를 탐색하는 모델을 베이스라인으로 설정하여, 기존 구조의 독해 모델을 사용하지 않고, 검색 모델만을 사용하여 정답 문장을 찾을 수 있는, 더 높은 성능의 새로운 방법을 제안한다. 답변 문장을 추출하는 검색 모델의 경우 여러 문서 중 질문과 유사한 문서를 찾는 기존의 검색 모델이 수행하는 작업과는 조금 다른 작업을 수행하게 된다. 그렇기 때문에 기존의 모델과는 학습 데이터 구축이 달라지는데, 이를 위한 효과적인 Hard Negative Sampling 기법 또한 제안한다.

2. 관련 연구

2.1 Open-Domain Question Answering

ODQA는 주어진 지문이 따로 존재하지 않고, 사전에 구축되어 있는 문서 풀(Document Pool)에서 질문과 관련 있는 문서를 찾고, 질문에 답하는 작업이다. ODQA는 일반적으로 검색 모델과 독해 모델로 구성되어 있는데, 검색 모델의 경우 대부분 사전 학습된 인코더 기반의 DPR[2]을 사용하며 TF-IDF나 BM25 같은 Sparse Embedding을 이용한 방법보다 높은 성능을 보였다.

독해 모델의 경우 마찬가지로 사전 학습된 인코더 모델을 기반으로 문서에서 정답의 위치를 탐색하는 방법과 생성 기반 모델[4, 5, 6]을 이용하여 질문에 답을 얻는 방식이 주로 사용된다.

최근의 ODQA 연구는 텍스트 - 텍스트 뿐 아니라 테이블 - 텍스트[7, 8], 이미지 - 텍스트[9, 10]와 같이 멀티 모달을 다루는 방식으로 확장되어 가고 있다.

2.2 Dense Retrieval

단어 혹은 어휘의 중복을 기반으로 하는 TF-IDF, BM25 등과 같은 Sparse Embedding을 이용하는 검색의 경우 중

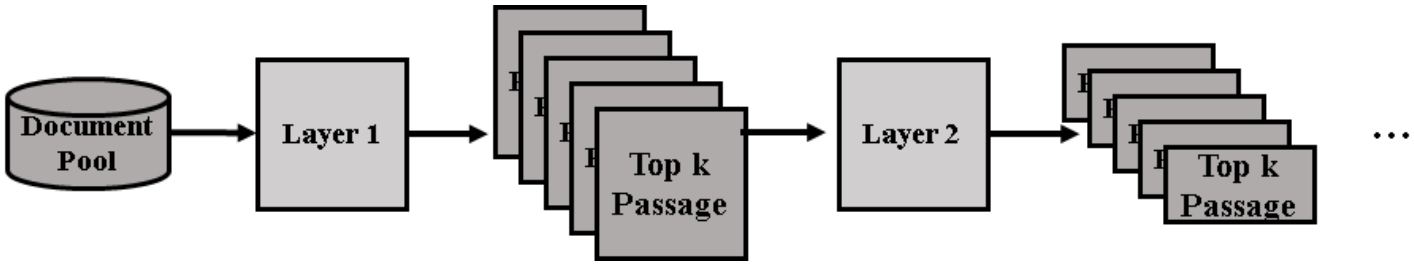

그림1. Layer_1 ~ Layer_n-1 구조

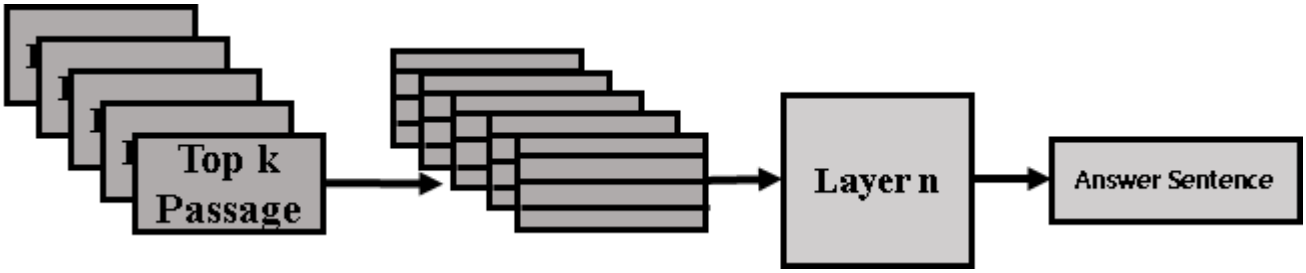

그림2. Layer_n 구조

요한 단어들이 정확히 일치하는 경우에는 성능이 뛰어나지만, 동의어와 같은 유사성을 고려하지 못한다. 이러한 문제를 해결하기 위해 딥러닝을 활용한 Dense Embedding 검색에 사용한다.

Dense Embedding을 위한 검색기로는 일반적으로 Cross-Encoder[11] 구조와 Bi-Encoder[2] 구조, Poly-Encoder 구조가 있다.

Cross-Encoder를 검색기로 사용할 경우, 질문과 문서를 동시에 모델에 입력으로 넣어 얼마나 둘이 유사한지에 대한 값을 얻는 방식으로 문서 풀의 수만큼 반복하기 때문에 문서 풀의 수가 클수록 시간이 오래 걸린다는 단점이 있다.

Bi-Encoder의 경우 문서 풀을 미리 임베딩 하여 저장한 뒤, 질문의 임베딩 값과 비교하여 유사도를 구하는 방식이기 때문에, Cross-Encoder에 비해 속도 측면에선 뛰어나지만, 성능이 떨어지는 단점이 있다.

Poly-Encoder[11]는 Cross-Encoder와 Bi-Encoder의 장점을 합친 구조로 Cross-Encoder보다 빠르고, Bi-Encoder보다 정확한 구조이다.

본 논문은 Bi-Encoder 구조를 이용한 DPR[2] 방식을 검색 모델로 이용하여 전통적인 구조의 검색 - 독해 파이프라인 보다 성능이 좋은 계층적 구조의 검색 모델을 제안한다.

3. 제안 기법

계층적 검색 모델을 이용한 정답 문장 탐색은 독해 모델 없이 복수의 검색 모델만을 사용한다. 각 검색 모델들은 각기 다른 길이의 문서를 탐색하게 되는데, 이를 위해 각 검색 모델은 각기 다른 최대 토큰 길이로 학습이 되었으며, 본 논문에서는 이러한 방법으로 학습된 각 검색 모델들을 Layer라고 부른다.

3.1 구조

최초 단계 즉 Layer_1에서는 일반적인 ODQA 작업에서 사용되는 검색 모델과 동일한 작업을 수행한다. 전체 문서 풀에서 질문과 유사한 k개의 문서를 탐색하여 찾아낸 후, 해당 문서들을 각각 이등분하여 2k 개의 문서들로 만든다. 이때 문서에 중요한 문장이 잘리는 것을 피하고자 슬라이딩 윈도우를 적용하여 나누게 된다. 이러한 과정을 통해 만든 2k개의 문서는 새로운 문서 풀이 되어 다음 Layer가 질문과 유사한 문서를 검색하는 문서 풀이 된다. 즉 Layer_1과 Layer_n을 제외한 Layer들은 수식 (1)과 같이 직전 Layer에서 탐색한 문서들을 문서 풀로 삼아 탐색을 수행하는 작업을 수행하게 된다.

수식 (1)의 DP는 Document Pool(문서 풀)의 약자이다.

$$DP_k = Layer_{k-1}\left(DP_{(k-1)}(Query)\right) \quad for\ 1 < k \leq n-1 \qquad (1)$$

마지막 Layer를 제외한 Layer_1부터 Layer_n-1까지의 모델의 구조는 그림 1과 같으며, 마지막 Layer의 구조는 그림 2와 같다.

마지막 Layer에서는 기존처럼 슬라이딩 윈도우를 적용하여 문서를 나누는 것이 아닌, 문장 단위로 나뉜 문서 풀을 사용하게 된다. 즉 직전 Layer에서 질문과 유사하다고 판단한 문서들을 각각 문장 단위로 추출한 문서 풀에서 질문과 가장 유사한 문장을 탐색하여 최종적인 결과를 얻는 Layer이다.

3.2 데이터

앞서 설명했듯 각 Layer에 사용되는 검색 모델은 서로 다른 길이의 문서를 탐색하기 때문에 학습 데이터 또한 다른 최대 길이의 데이터로 학습이 되어야 한다. 그렇기

때문에 KorQuAD[1] 데이터를 가공하여 각기 다른 길이로 Context를 잘라 학습에 이용하였다.

우선, 데이터셋의 Answer Span을 기반으로 정답 문장을 추출한 뒤, 각 최대 토큰 길이에 맞게 정답 문장을 반드시 포함하는 문장을 생성하였다. 이러한 방식으로 생성한 데이터는 아래 표 1과 같으며, DPR[2]과 같은 방식으로 In-Batch-Negative 기법을 적용하여 대조 학습(Contrastive Learning)을 진행하였다.

표1. 데이터 예시

Question 바그너는 괴테의 파우스트를 읽고 무엇을 쓰고자 했는가?
Answer Sentence 1839년 바그너는 괴테의 파우스트을 처음 읽고 그 내용에 마음이 끌려 이를 소재로 해서 하나의 교향곡을 쓰려는 뜻을 갖는다.
Origin Context 1839년 바그너는 괴테의 파우스트을 처음 읽고 그 내용에 마음이 끌려 이를 소재로 해서 하나의 교향곡을 쓰려는 뜻을 갖는다. 이 시기 바그너는 1838년에 빚 독촉으로 산전수전을 다 겪은 상황이라 좌절과 실망에 가득했으며 메피스토펠레스를 만나는 파우스트의 심경에 공감했다고 한다. 또한 파리에서 아브네크의 지휘로 파리 음악원 관현악단이 연주하는 베토벤의 교향곡 9번을 듣고 깊은 감명을 받았는데, 이것이 이듬해 1월에 파우스트의 서곡으로 쓰여진 이 작품에 조금이라도 영향을 끼쳤으리라는 것은 의심할 여지가 없다. … (중략) … 또한 작품의 완성과 동시에 그는 이 서곡(1악장)을 파리 음악원의 연주회에서 연주할 파트보까지 준비하였으나, 실제로는 이루어지지는 않았다. 결국 초연은 4년 반이 지난 후에 드레스덴에서 연주되었고 재연도 이루어졌지만, 이후에 그대로 방치되고 말았다. 그 사이에 그는 리엔치와 방황하는 네덜란드인을 완성하고 탄호이저에도 착수하는 등 분주한 시간을 보냈는데, 그런 바쁜 생활이 이 곡을 잊게 한 것이 아닌가 하는 의견도 있다.
256 Tokens 1839년 바그너는 괴테의 파우스트을 처음 읽고 그 내용에 마음이 끌려 이를 소재로 해서 하나의 교향곡을 쓰려는 뜻을 갖는다. 이 시기 바그너는 1838년에 빚 독촉으로 산전수전을 다 겪은 상황이라 좌절과 실망에 가득했으며 메피스토펠레스를 만나는 파우스트의 심경에 공감했다고 한다. 또한 파리에서 아브네크의 지휘로 파리 음악원 관현악단이 연주하는 베토벤의 교향곡 9번을 듣고 깊은 감명을 받았는데, 이것이 이듬해 1월에 파우스트의 서곡으로 쓰여진 이 작품에 조금이라도 영향을 끼쳤으리라는 것은 의심할 여지가 없다. … (중략) …또한 작품의 완성과 동시에 그는 이 서곡(1악장)을 파리 음악원의 연주회에서 연주할
128 Tokens 1839년 바그너는 괴테의 파우스트을 처음 읽고 그 내용에 마음이 끌려 이를 소재로 해서 하나의 교향곡을 쓰려는 뜻을 갖는다. 이 시기 바그너는 1838년에 빚 독촉으로 산전수전을 다 겪은 상황이라 좌절과 실망에 가득했으며 메피스토펠레스를 만나는 파우스트의 심경에 공감했다고 한다. 또한 파리에서 아브네크의 지휘로 파리 음악원 관현악단이 연주하는 베토벤의 교향곡 9번을 듣고 깊은 감명을 받았는데, 이것이 이듬해 1월에 파우스트의 서곡

본 논문에서는 기본이 되는 512 토큰의 길이를 갖는 검색 모델과, 256 토큰과 128 토큰, 한 문장 길이로 총 4개의 검색 모델을 학습하였다.

3.3 Hard Negative Sampling

기존 검색 과정의 경우 큰 규모의 문서 풀에서 질문과 유사한 문서를 검색 모델을 통해 탐색하는 것에 반해, 본 논문에서 제안하는 방법의 경우 이미 질문과 유사한 문서들 사이에서 다시 한번 질문과 유사한 문장이 포함된 문서를 찾아야 하기 때문에 탐색에 있어서 어려움이 존재한다. 본 논문에서는 이러한 문제를 해결하기 위해서 Label Context에서 정답 문장을 삭제한 Context를 Hard Negative로 설정하여 학습을 진행하는 Hard Negative Sampling 기법을 제안한다.

Hard Negative는 아래 표 2와 같은 방식으로 생성하였다.

표2. Hard Negative Sample

Answer Sentence 1839년 바그너는 괴테의 파우스트을 처음 읽고 그 내용에 마음이 끌려 이를 소재로 해서 하나의 교향곡을 쓰려는 뜻을 갖는다.
Label 1839년 바그너는 괴테의 파우스트을 처음 읽고 그 내용에 마음이 끌려 이를 소재로 해서 하나의 교향곡을 쓰려는 뜻을 갖는다. 이 시기 바그너는 1838년에 빚 독촉으로 산전수전을 다 겪은 상황이라 좌절과 실망에 가득했으며 메피스토펠레스를 만나는 파우스트의 심경에 공감했다고 한다. 또한 … (중략)
Hard Negative 이 시기 바그너는 1838년에 빚 독촉으로 산전수전을 다 겪은 상황이라 좌절과 실망에 가득했으며 메피스토펠레스를 만나는 파우스트의 심경에 공감했다고 한다. 또한 … (중략)

Hard Negative를 적용하였을 경우와 그렇지 않은 일반적인 제안 모델의 성능은 아래 표 3과 같다. Top5의 경우 마지막을 제외한 각 Layer에서 top-5개의 문서를 탐색하였단 의미이고, 마찬가지로 Top10의 경우 top-10의 문서를 기반으로 탐색을 수행하였단 의미이다. 해당 결과를 보아 제안하는 Hard Negative Sampling 기법이 본 논문에서 제안하는 작업에 있어서 유효하다는 것을 확인할 수 있으며, 검색 모델이 문서의 어떠한 부분에 집중할지를 Hard Negative를 통해 어느 정도 조종할 수 있다는 것 확인할 수 있다.

표3. Acc 비교

Model	Top5	Top10
Ours	0.666	0.692
Ours + Hard Negative	0.811	**0.842**

4. 실험 결과

4.1 실험 환경

실험에 사용된 모델은 공통으로 사용되는, 512 토큰의 최대 길이로 학습된 검색 모델, 본 논문에서 제안하는 각기 다른 최대 토큰 길이로 학습된 계층 검색 모델들(256 토큰, 128 토큰, 문장 단위로 각각 학습되었음)과 베이스라인이 되는 인코더(BERT)기반의 독해 모델로 총 세 종류이다.

모든 모델은 공통적으로 AdamW 옵티마이저를 사용하였

으며, 학습률 1e-5에서 5 Epoch 학습을 진행하였다.

공통으로 사용되는 검색 모델의 경우 배치 사이즈 32로 학습되었으며, 계층 검색 모델들은 배치 사이즈 64에서 제안한 Hard Negative Sample을 적용하여 학습되었다. 베이스라인인 독해 모델은 배치 사이즈 32에서 학습되었다.

최종적으로 본 논문에서 제안하는 방법의 모델 구조는 검색 모델 + 계층 검색 모델들로 구성되어 있고, 베이스라인은 검색 모델 + 독해 모델로 구성된다.

4.2 베이스라인

본 논문에서 제안하는 계층적 검색 모델을 이용한 정답 문장 탐색의 실험 결과를 비교하기 위한 베이스라인은 제안 방법에 사용된 Layer_1에 사용되는 검색 모델과 동일한 모델을 사용하였으며, BERT를 통해 위치 탐색 기반 독해 모델을 KorQuAD 데이터셋을 이용하여 학습하였다.

4.3 주요 결과

표4는 검증 데이터셋에서 Baseline 모델과 제안 모델의 F1 점수를 비교한 결과이다. Baseline 모델은 Top5, Top10에서 평균 74.6% 성능을 기록하였으며, 제안 모델은 평균 84.6%를 기록하여 제안 모델이 일반적으로 쓰이는 BERT 기반의 검색모델보다 F1 점수에서 평균 10%의 성능 향상이 있음을 확인할 수 있다.

표4. F1 Score

Model	Top5	Top10
Baseline	75.1	74.2
Ours	83.1	**86.1**

이와 마찬가지로 표5는 EM 점수를 비교한 결과인데, EM 점수는 느슨한 평가지표인 F1 점수와 다르게 정확한 답만 정답으로 평가하는 지표이기 때문에 정확한 답변을 생성하는 데 얼마나 성공적인지를 측정하는 데에 유용하다. Baseline 모델은 EM 점수에서 평균 70.4%의 성능을 기록하였고, 제안 모델은 82.6%의 성능을 기록하여 제안 모델이 약 12.2%의 성능 향상이 있음을 확인하였다.

표5. EM Score

Model	Top5	Top10
Baseline	70.8	70.1
Ours	81.1	**84.2**

아래 그림3에선 위 표4와 표5의 결과를 한눈에 비교할 수 있다.

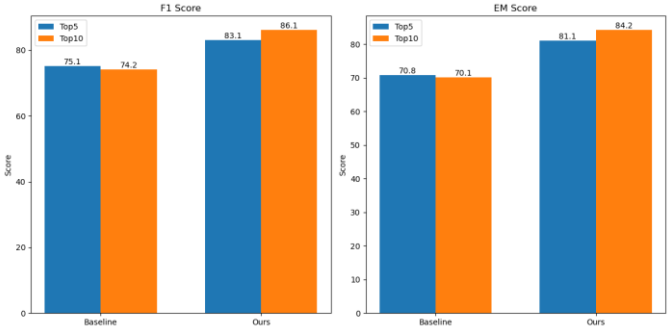

그림3. 실험 결과

5. 결론

본 논문은 검색 모델을 계층적으로 사용하여 정답 문장을 탐색하는 새로운 ODQA 방식을 제안한다. 기존과 다른 작업으로 인해 검색 모델의 성능이 낮아지는 것을 방지하기 위한 새로운 Hard Negative Sampling 기법 또한 제안하며, 해당 기법을 적용하였을 경우 기존의 ODQA에서 일반적으로 사용되는 인코더 기반의 Span 탐색 독해 모델보다 더 10~12% 정도 좋은 성능을 도출하였다. 이를 통하여 각 검색 모델을 학습할 때 Hard Negative의 설정에 따라 검색 모델이 문장의 어느 부분에 집중할지를 어느 정도 조종할 수 있다는 것을 확인하였다.

향후 연구로는 이러한 특징을 이용하여 ODQA 작업에 적용할 수 있는 여러 방법들을 탐색하여 실험해 볼 예정이다.

감사의 글

이 논문은 2021년도 정부(산업통상자원부)의 재원으로 한국산업기술기획평가원의 지원을 받아 수행된 연구임 (No.1415187244, 비동기식 업무지원 및 오프라인 사수 기능을 가진 AI 어시스턴트 서비스 개발)

참고문헌

[1] 임승영, 김명지, 이주열.(2018).KorQuAD: 기계독해를 위한 한국어 질의응답 데이터셋.한국정보과학회 학술발표논문집,(),539-541.

[2] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6769-6781, Online. Association for Computational Linguistics.

[3] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In Proceedings of the 55th Annual Meeting of the Association for

Computational Linguistics (Volume 1: Long Papers), pages 1870-1879, Vancouver, Canada. Association for Computational Linguistics.
[4] G. Izacard and E. Grave, “Leveraging passage retrieval with generative models for open domain question answering,” Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 874-880, Apr. 2021. [Online]. Available: https://aclanthology. org/2021.eacl-main.74
[5] G. Izacard and E. Grave, “Distilling knowledge from reader to retriever for question answering,” 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, 2021. [Online]. Available: https://openreview.net/forum?id=NTEz-6wysd
[6] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, “BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7871-7880, Jul. 2020. [Online]. Available: https://aclanthology.org/2020.acl-main.703
[7] 김영민, 임승영, 이현정, 박소윤, 김명지.(2019).KorQuAD 2.0: 웹문서 기계독해를 위한 한국어 질의응답 데이터셋.한글 및 한국어 정보처리 학술대회 논문집,(),097-102
[8] Huang, J., Zhong, W., Liu, Q., Gong, M., Jiang, D., & Duan, N. (2022). Mixed-modality Representation Learning and Pre-training for Joint Table-and-Text Retrieval in OpenQA. arXiv:2210.05197 [cs.CL].
[9] Chen, X., Wang, X., Changpinyo, S., Piergiovanni, A. J., Padlewski, P., Salz, D., ... Soricut, R. (2023). PaLI: A Jointly-Scaled Multilingual Language-Image Model. arXiv:2209.06794 [cs.CV].
[10] Wang, W., Bao, H., Dong, L., Bjorck, J., Peng, Z., Liu, Q., ... Wei, F. (2022). Image as a Foreign Language: BEiT Pretraining for All Vision and Vision-Language Tasks. arXiv:2208.10442 [cs.CV].
[11] Humeau, S., Shuster, K., Lachaux, M. A., & Weston, J. (2020). Poly-encoders: Transformer Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. arXiv:1905.01969 [cs.CL].