KorQuAD를 활용한 한국어 오픈도메인 질의응답 시스템*

조상현○, 김민호, 권혁철
부산대학교 전기전자컴퓨터공학과
delosycho@gmail.com, {karma, hckwon}@pusan.ac.kr

Korean Open Domain Question Answering System Using KorQuAD

Sanghyun Cho○, Minho Kim, Hyuk-Chul Kwon
Dept. of Computer Science Pusan National University

요 약

오픈 도메인 질의응답이란, 질문을 줬을 때 그 질문과 연관성이 높은 문서를 검색하고 검색된 문서에서 정답을 추출하는 태스크이다. 본 논문은 기계 독해 데이터인 KorQuAD를 활용한 오픈도메인 질의응답 시스템을 제안한다. 문서 검색기를 이용하여 질문과 관련 있는 위키피디아 문서들을 검색하고 검색된 문서에 단락 선택 모델을 통해서 문서 질문과 연관성이 높은 단락들을 선별하여 기계 독해 모델에서 처리해야 할 입력의 수를 줄였다. 문서 선별모델에서 선별된 여러 단락에서 추출된 정답 후보에서 여러 가지 정답 모형을 적용하여 성능을 비교하는 실험을 하였다. 본 논문에서 제안한 오픈도메인 질의응답 시스템을 KorQuAD에 적용했을 때, 개발 데이터에서 EM 40.42%, F1 55.34%의 성능을 보였다.

주제어: 기계 독해, 오픈도메인 질의응답

1. 서론

오픈도메인 질의응답은 질문에 대한 답이 포함된 문서를 입력으로 사용하지 않고 방대한 말뭉치 내에서 질문과 관련된 문서를 추출하고 그 문서들에 기계 독해를 적용하여 정답을 찾는 문제이다. 오픈도메인 질의응답 연구는 기계 독해 데이터를 오픈도메인 질의응답에 맞게 설정하여 실험한 연구들과 오픈 도메인에 맞게 만들어진 데이터를 이용하여 실험한 연구들이 있다. 오픈도메인 질의응답에서 정답을 추출하기 위해서 기계 독해에 적용된 모델들을 활용하고 있다. WebQuestions[1], Quasar-T[2] 등이 오픈도메인 질의응답과 관련된 데이터이다. SQuAD[3]는 기계 독해 태스크의 대표적인 데이터이며, **KorQuAD[4]는 한국어에서 SQuAD를 벤치마킹한 데이터이다. 본 연구에서는 KorQuAD 1.0 데이터를 이용하여 기계 독해 모델을 학습시켰다. 개발 데이터를 이용하여 문서 검색기에서 검색된 위키피디아 문서에서 정답을 찾도록 하고 4가지의 정답 선택 모형의 성능을 비교하는 실험을 하였다.

2. 관련 연구

오픈도메인 질의응답에는 기계 독해에 적용된 모델들이 다수 적용되었다. BERT를 기계 독해에 적용한 연구로는 [5]과 [6]이 있다. [5]은 사전 학습된 BERT에 Global Attention을 적용하고 Multi-level Co-Attention Fusion을 결합한 모델을 제안했다. [6]은 사전 학습된 BERT에 SRU(simple recurrent unit)와 질의응답에 적합한 자질을 추가한 모델을 제안했다.

[7]은 기계독해 데이터인 SQuAD를 통해서 기계독해 모델을 학습시키고 문서 검색기에서 검색된 위키피디아 문서에서 정답을 찾도록 했다. 검색된 위키피디아 문서 중에서 정답을 포함하고 있는 문서에 원거리 감독을 적용하여 추가적인 학습 데이터로 사용했다. [8]은 원거리 감독을 적용했을 때 부정확한 라벨링으로 인한 성능 감소를 줄이기 위해 노이즈가 있는 문서를 제외하는 방법을 제안했다. [9]는 문서 검색기를 통해 추출한 문서를 단락으로 나누고 단락의 양방향 LSTM을 통해서 질문과 연관성이 큰 단락들을 선별하여 문서 독해기의 입력으로 사용하였다. [10]은 추출된 정답을 다른 단락들에서 포함하고 있다면 정답으로 선택될 확률을 높게 하고 각 단락에서 추출된 정답을 다시 단락 및 질문과 인코딩하여 연관성을 계산하여 정답을 리랭킹하는 방법을 제안했다.

[11]은 지식베이스 그래프를 오픈도메인 질의응답에 적용했다. 지식베이스 그래프를 처리하기 위해 Graph CNN[12]을 적용했으며, 문서 검색기를 통해 검색된 문서에 LSTM을 적용하여 오픈도메인 질의응답에 지식베이스 그래프와 위키피디아 문서를 함께 이용했다.

한국어 오픈도메인 질의응답에 대한 연구로는 [13]와 [14]이 있었다. [13]는 검색된 문서에 정답의 존재 여부를 판단하는 NIL-Aware를 Reinforced Ranker-Reader에 추가한 모델을 제안했으며, [14]는 NIL-Aware 모델에서 찾을 수 없는 질문에 대해 KBQA에서 정답을 추출하도록 결합하여 성능을 높이는 방법을 제안했다.

본 연구에서는 [7]~[10]과 같이 기계 독해 데이터인 KorQuAD를 오픈 도메인 질의응답에 맞게 설정하고, 트랜스포머 기반의 문서 선별 모델과 BERT 기반의 기계 독해 모델을 이용한 한국어 오픈 도메인 질의응답 시스템을

* 본 연구는 미래창조과학부 및 정보통신기술연구진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [2013-0-00179, (엑소브레인-3세부) 컨텍스트 인지형 Deep-Symbolic 하이브리드 지능 원천 기술 개발 및 언어 지식 자원 구축]

** https://korquad.github.io/category/1.0_KOR.html

제안한다.

3. 질의응답 시스템

본 연구의 질의응답 시스템은 크게 4가지 부분으로 이루어져 있다: (1) 문서 검색기는 질문과 연관성이 큰 위키피디아 문서들을 검색하는 부분이다 (2) 단락 선별은 인공신경망 기반의 모델을 통하여 문서 검색기에서 검색된 다수의 문서를 단락으로 나누고 연관성이 큰 단락들만 추출하는 방법이다 (3) 문서 독해기는 추출된 문서들에서 정답을 추출하는 모듈이다 (4) 정답 선택에서는 문서 독해기에서 추출된 여러 개의 정답 중에서 1가지 정답을 선택하는 방법에 관하여 설명한다.

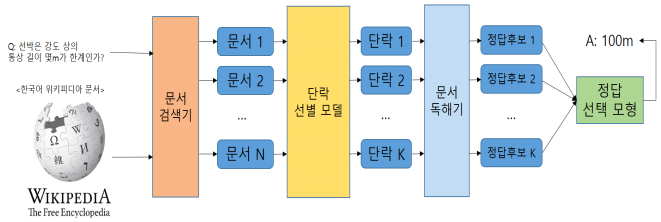

그림1. 오픈도메인 질의응답 시스템 구조

3.1 문서 검색기

문서 검색기는 질문과 연관성이 큰 위키피디아 문서들을 검색하는 모듈이다. 빠르고 효율적인 방법을 통해 문서들을 검색함으로써 인공신경망 기반의 모듈에서 처리할 문서의 수를 줄이는 것이 문서 검색기의 역할이다. 문서 검색기는 역색인(inverted index) 방법과 BM25를 적용했다.

질문과 연관성이 가장 높은 상위 100개의 문서를 추출하여 문서 리랭킹 모듈의 입력으로 사용했다.

3.2 단락 선별

하나의 질문에 답변하기 위해서 문서 독해기에서 상위 100개의 문서를 모두 처리하는 것은 많은 처리시간이 필요하다. 단락 선별은 문서 검색기에서 추출된 상위 100개의 문서를 단락으로 나누고, 각 문서에서 연관성이 큰 단락들을 선별하는 방법이다. 길이가 355를 넘는 단락은 여러 개의 입력으로 나누었다. 단락 선별 모델은 문서 독해기에서 처리해야 할 입력의 크기를 줄여주는 역할을 한다. 단락 선별 모델의 구조는 그림 2와 같다. 인코더는 LSTM[15]과 트랜스포머[16]를 이용하여 실험했다. 컨볼루션과 풀링 레이어는 [17]에서 고안된 방법을 사용했다. 단락 선별에는 단락 선별 모델의 출력과 문서 검색기의 출력이 함께 사용되었으며, 이에 대한 수식은 다음과 같다.

$$r_i = \alpha(P_{i,j}, q) * \beta(D_i, q) \qquad (1)$$

$D=\{D_1,...,D_{100}\}$는 문서 검색기를 통해 추출된 상위 문서이며 $P_i=\{P_{i,1},...P_{i,l}\}$는 검색된 문서를 단락으로 나눈 것을 의미하며, q는 질문을 의미한다. α는 단락 선별 모델의 출력 값이며 β는 문서 검색기의 출력 값을 의미한다. r값이 큰 상위 K개의 단락을 문서 독해기의 입력으로 넘겨준다.

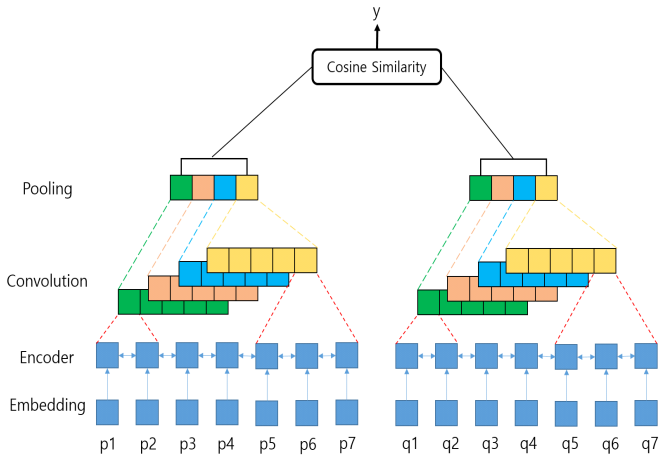

그림2. 단락 선별 모델 구조

3.3 문서 독해기

문서 독해기는 단락 선별 모델에서 추출된 단락을 입력으로 받아서 질문에 대한 정답을 추출하는 모듈이다. 여러 단락으로 이루어진 위키피디아 문서들은 단락 단위로 나누었으며, 길이가 355를 넘는 단락은 여러 개로 나누었다. 문서 독해기는 BERT[18] 모델을 이용했다. 문서 독해기를 학습시키기 위해서 무작위로 선택된 질문과 관련이 없는 문서를 학습 데이터에 포함 시켰다. 정답이 없는 무작위로 선택된 문서의 정답 시작 위치와 끝 위치는 [CLS]를 가리키도록 했다. 그림3은 기계 독해 모델의 구조를 나타낸다.

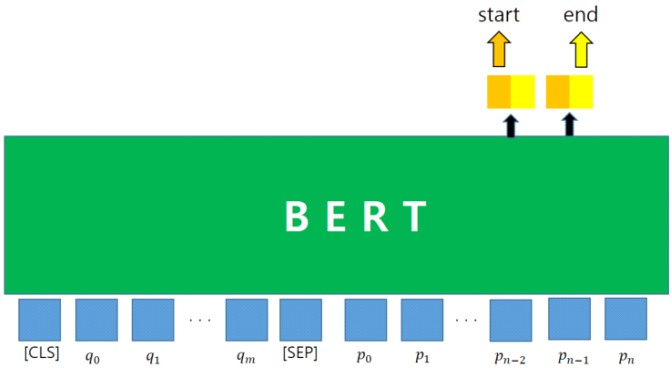

그림3. BERT 기반 기계 독해 모델 구조

"$X_i=[CLS], q_0, q_1, ..., q_m, [SEP], p_{i,0}, p_{i,1}, ..., p_{i,n}$" 와 같은 포맷으로 BERT에 입력을 주게 된다. 여기서 Q=$\{q_1,...,q_m\}$는 질문을 의미하며, P_i=$\{p_{i,0},...,p_{i,n}\}$는 정답을 찾기 위한 i번째 단락을 의미한다. [18]에서 SQuAD 태스크에 적용

한 방법과 같이 출력 값의 차원 수를 2로 설정한 FFNN(feed-forward neural network)을 통해서 시작과 끝의 확률을 구하였다. 이에 대한 수식은 다음과 같다.

$$h_i = BERT(X_i) \in R^{(n+m+2)\times d} \quad (2)$$

$$h^o_{i,j} = FFNN(h_{i,j}) \in R^{(n+m+2)\times 2} \quad (3)$$

$$y^s_i = softmax(h^{o1}_i) \in R^{n+m+2} \quad (4)$$

$$y^e_i = softmax(h^{o2}_i) \in R^{n+m+2} \quad (5)$$

여기서 식(1)의 d는 BERT의 히든 레이어 차원 수를 의미한다.

3.4 정답 선택

문서 선별 모델에서 추출된 상위 K개의 단락이 독해기의 입력으로 들어가기 때문에 K개의 정답이 출력된다. 단락 선택은 이 중에서 가장 확률이 높은 1가지 정답을 선택하는 방법이다.

모형1은 FFNN에서 출력된 모든 단락의 출력값을 병합하고, 병합된 출력값에 softmax를 이용하여 정답의 시작과 끝일 확률을 계산하는 방법이다. 이에 대한 수식은 다음과 같다.

$$\overline{h^o} = [h^o_0; h^o_1; ...; h^o_D] \in R^{(n+m+2)\times D\times 2} \quad (6)$$

$$\hat{y}^s = softmax(\overline{h}^{o1}) \in R^{(n+m+2)\times D} \quad (7)$$

$$\hat{y}^e = softmax(h^{o2}) \in R^{(n+m+2)\times D} \quad (8)$$

식 (6)의 [;]는 벡터의 결합을 의미한다.

모형2는 [CLS] 토큰을 정답의 시작과 끝으로 선택할 확률이 가장 낮은 단락의 정답을 선택하는 것이다. 이에 대한 수식은 다음과 같다.

$$\hat{A} = \underset{A_i}{\operatorname{argmax}}\, p(A_i | P_i, Q) \quad (9)$$

$$p(A_i | P_i, Q) = -(y^s_{i,0} + y^e_{i,0}) \quad (10)$$

여기서 A_i는 기계 독해기에 단락 P_i을 입력으로 하였을 때 추출된 정답을 의미한다.

모형3은 정답으로 추출된 단어들을 토큰으로 나누고 빈도수가 높은 토큰을 가지는 정답이 선택될 확률을 높이는 방법이다.

$$p(A_i|P_i, Q) = (2 - y^s_{i,0} - y^e_{i,0}) * \frac{1}{T_i} * \sum_{t=0}^{T_i} freq(A^t_{i,t}) \quad (11)$$

여기서 A^t는 토큰으로 나누어진 추출된 정답을 의미한다.

모형4는 계산된 확률값에 a 벡터를 곱해주는 방법이다. 이에 따른 수식은 다음과 같다.

$$a_i = FFNN(h_{i,0}) \in R \quad (12)$$

$$\overline{y}^s_{i,j} = y^s_{i,j} \circ a_i \quad (13)$$

$$\overline{y}^e_{i,j} = y^e_{i,j} \circ a_i \quad (14)$$

문서 독해기를 학습시킬 때 정답이 없는 무작위로 선택된 문서는 문서 독해기에서 출력되는 모든 단어에 대해서 정답의 시작과 끝일 확률이 모두 0이 나오도록 학습 라벨을 설정하였다.

4. 실험 및 결과

4.1 실험 환경 및 데이터

본 연구에서는 두 가지 유형의 데이터를 사용했다.

(1) 한국어 위키피디아 문서들을 오픈도메인 질의응답 시스템을 위한 지식베이스 문서로 활용했다.

(2) KorQuAD 데이터를 QA 모델의 Document Reader의 학습을 위해 사용했다.

2018년 4월에 내려받은 한국어 위키피디아 덤프를 사용하여 오픈도메인 QA 시스템이 정답을 찾기 위한 지식베이스 문서로 사용했다.

각 위키피디아 페이지에서 태그로 이루어진 불필요한 텍스트들을 모두 제거하여 plain Text만을 추출하였으며, 100음절 이하의 길이를 가지는 문서들을 제외했다. 총 384,905개의 문서를 지식베이스 문서로 사용했다.

KorQuAD는 총 60,407개의 학습 데이터와 5,774개의 개발 데이터, 그리고 3,898개의 평가 데이터로 이루어진 기계 독해를 위한 데이터이며, 위키피디아 문서를 기반으로 만들어졌다.

각 데이터는 질문과 질문에 해당하는 답을 포함하고 있는 단락, 그리고 단락에서 정답에 해당하는 음절의 위치와 정답의 길이로 이루어져 있다.

KorQuAD의 학습 데이터를 오픈도메인 질의응답의 학습을 위해 사용했으며, 성능 평가에서는 KorQuAD의 개발 데이터를 사용했다.

학습된 모델의 오픈도메인 질의응답을 위해서, KorQuAD 개발 데이터에서 질문과 그 질문에 대한 정답과 문서 검색기에서 연관성이 높게 나온 상위문서들을 QA 모델의 입력으로 사용하였다. 1500음절 이상의 길이를 가지는 문서의 경우, 단락으로 나누어 입력으로 사용하였다.

문서 독해기에서 추출된 정답을 평가하기 위해서, SQuAD에서 쓰는 평가 Metric인 F1과 EM을 통해서 성능을 측정했다.

KorQuAD 데이터셋은 질문에 정답을 포함하고 있는 단락이 데이터에 포함되어 있지만, 문서 검색기에서 검색된 상위 문서에도 질문에 대한 정답이 포함되어 있을 수 있다. 본 연구에서는 더 많은 학습 데이터를 확보하기 위해서 학습 데이터의 질문과 답을 문서 검색기에서 검색된 관련 있는 문서를 연관시키는 원거리 감독을 적용했다. 문서 검색기를 실행시켜서 연관성이 높은 상위 10개의 문서에서 정답과 똑같은 텍스트가 존재하는 문서들을 추출한다. 이전 연구[7]를 따라서, 만약 질문 안에 개체명이 있다면, 그 개체명을 포함하지 않는 문서들은 모두 제외했다. KorQuAD에서 제공된 단락을 포함하고 있는 위키피디아 문서에서 링크로 연결되어 있지 않은 문서들을 모두 제외했다.

4.2 문서 검색 및 단락 선별

본 연구에서 사용된 트랜스포머 모델의 하이퍼파라미터는 “히든 레이어 차원 수: 128, 어텐션 헤드 수: 4, 트랜스포머 블록수: 3”로 설정하였으며, LSTM 모델은 히든 레이어 차원 수: 200, LSTM 레이어 수: 2”로 설정하였다.

본 연구에서는 문서 검색과 문서 리랭킹의 성능을 평가하기 위해서 KorQuAD 데이터의 질문에 대한 단락을 포함하고 있는 위키피디아 문서를 정답 문서로 보았다.

[표 1]은 문서 검색기의 검색 성능 결과이다. 질문을 입력으로 검색을 하여 검색된 N개의 상위문서에 정답 문서가 포함되었으면 적중한 것으로 평가하였다.

표1. 문서 검색기 성능 평가(%)

상위문서 개수	문서 검색 결과
5	55.1
10	63.4
20	68.5
50	76.8
100	83.2
200	88.5
300	91.3

[표 2]는 단락 선택 모델의 성능을 나타낸다. 본 연구에서는 단락 선택 모델의 처리시간을 줄이기 위해서 100개의 상위 문서를 입력으로 사용하였다. 문서 검색기에서 검색된 상위 100개의 문서와 질문을 입력하여 리랭킹 모델에서 출력된 점수를 기반으로 상위 K개의 단락을 뽑고 K개의 상위 단락에 정답 단락이 포함되어 있으면 적중한 것으로 보았다. 상위 100개의 문서를 단락으로 나누었을 때 평균 847.2개의 단락으로 나누어졌다.

표2. 단락 선택 모델 성능 평가(%, dev)

모델	단락 선별 결과			
	상위 20	상위 40	상위 80	상위 120
Transformer	55.48	62.58	70.32	72.72
LSTM	51.27	59.15	62.29	63.51

단락 선별 모델의 인코더로 트랜스포머를 사용했을 때 더 향상된 성능을 보였다.

4.3 기계 독해 및 오픈도메인 질의응답

본 연구에서 사용된 BERT 모델은 한국어 위키피디아 문서(약 350만 문장)을 통해서 사전학습시킨 모델을 사용했다. 학습 데이터는 형태소 분석을 통해 형태소 단위로 분리하고 출현 빈도를 기준으로 128,000개의 단어로 사전을 구축하였다. 하이퍼파라미터는 BERT-base를 따라서 “히든 레이어 차원 수: 768, 어텐션 헤드 수: 12, 트랜스포머 블록수: 12”로 설정하였다.

[표 3]은 제안하는 모델에 대해서 정답 선택 모형에 따른 성능 비교를 보인다.

표3. 정답 선택 방법에 따른 성능 비교(%,dev)

Method	단락 20개		단락 40개	
	EM	F1	EM	F1
모형1	30.80	46.93	32.67	48.14
모형2	34.61	52.42	39.13	53.20
모형3	35.09	52.75	34.11	51.37
모형4	28.55	35.45	30.16	37.58

정답 선택 방법에 따른 실험 결과는 다음과 같으며 단락 선별 모델에서 상위 20개의 단락을 추출했을 때의 결과이다. [CLS] 토큰을 정답의 시작과 끝으로 선택할 확률이 가장 낮은 단락에서 추출된 정답을 선택하는 것이 좋은 성능을 보였으며, 추출된 정답 토큰의 빈도수를 반영했을 때 EM 0.95%, F1 0.96%로 향상된 성능을 보였지만 단락의 개수를 40개로 설정하였을 때는 EM 0.98%, F1 1.38%로 하락한 성능을 보였다. 이는 추출되는 정답의 개수가 많아지면서 비슷한 유형을 가지는 정답이 아닌 토큰들의 개수가 늘어났기 때문에 성능이 하락한 것으로 보인다.

표4. 단락 개수에 따른 성능 비교(%, dev)

단락 개수	EM	F1
20	34.61	52.42
40	39.13	53.20
80	40.42	55.34
120	41.81	55.09

단락 선별 모델에서 추출하는 단락의 개수에 따른 실

험 결과는 다음과 같으며, 모형2를 통해 성능을 비교하였다. 단락 선별에서 더 많은 단락을 추출할수록 질문과 관련 있는 단락이 추출될 확률이 높아지는 것을 보였으며, 더 많은 단락을 입력했을 때 오픈도메인 질의응답에서도 더 좋은 성능을 보였지만, 단락의 개수가 120개일 때는 EM 1.39%가 향상되었고, F1 0.25%가 하락했다. 이는 정답 선택 모형에서 선택해야 할 정답의 개수가 많아지면서 정답 선택 모형의 정확도가 떨어진 것으로 보인다.

단락 선별 모델에서 정답을 포함하고 있는 문서를 문서 독해기의 입력으로 주었을 때와 주지 않았을 때의 정답/오답 비율은 다음과 같다.

표5. 정답 포함/미포함 입력에 따른 정답/오답 비율(%)

	정답	오답
정답 포함 단락	92.73	12.39
정답 미포함 단락	7.27	87.61

단락 선별 모델에서 더 많은 정답이 포함된 단락을 기계 독해기에 입력으로 준다면 질의응답 성능이 더 향상될 수 있을 것이다.

5. 결론

본 연구에서는 한국어 오픈도메인 질의응답을 위하여 기계 독해 데이터인 KorQuAD를 오픈도메인에 적용하는 방법을 제안하였다. 실험 결과, 단락 선별 모델에서 상위 80개의 단락을 추출하여 기계 독해 모델의 입력으로 하였을 때 EM 40.42%, F1 55.34%의 성능을 보였다. 단락의 개수가 작을 때는 모형3이 좋은 성능을 보였지만 단락의 개수가 많을 때는 모형2가 우수한 성능을 보였다. 단락의 개수가 많아질수록 성능이 향상되었지만 120개의 단락을 입력으로 주었을 때는 F1이 하락하였다.

향후 연구로는 문서 선별 모델을 개선하여 오픈 도메인 질의응답의 성능을 높이는 연구를 할 예정이다.

참고문헌

[1] J. Berant et al., Semantic parsing on freebase from question-answer pairs, Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013.

[2] B. Dhingraet al., Quasar: Datasets for question answering by search and reading, arXiv preprint arXiv:1707.03904, 2017.

[3] P. Rajpurkar, et al., Squad: 100,000+ questions for machine comprehension of text, arXiv preprint arXiv:1606.05250, 2016.

[4] 임승영, 김명지, 이주열, KorQuAD: 기계독해를 위한 한국어 질의응답 데이터셋, 한국정보과학회 학술발표논문집, pp. 643-645, 2018

[5] 박광현, 나승훈, 최윤수, 장두성, BERT와 Multi-level Co-Attention Fusion을 이용한 한국어 기계독해, 한국정보과학회 학술발표논문집, 2019.

[6] 이동헌, 박천음, 이창기, 박소윤, 임승영, 김명지, 이주열, BERT를 이용한 한국어 기계 독해. 한국정보과학회 학술발표논문집, pp. 557-559, 2019.

[7] D. Chen, A. Fisch, J. Wetson, Reading Wikipedia to Answer Open-Domain Questions, Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017.

[8] Y. Lin, et al., Denoising Distantly Supervised Open-Domain Question Answering, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018.

[9] J. Lee et al., Ranking Paragraphs for Improving Answer Recall in Open-Domain Question Answering, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 565-569, 2018.

[10] S. Wang et al., Evidence Aggregation for Answer Re-Ranking in Open-Domain Question Answering, arXiv preprint arXiv:1711.05116, 2017.

[11] H. Sun, et al., Open Domain Question Answering Using Early Fusion of Knowledge Bases and Text, EMNLP, 2018.

[12] Thomas N. Kipf and Max Welling, Semi-Supervised Classification with Graph Convolutional Networks, International Conference on Learning Representations, 2017.

[13] 이영훈, 나승훈, 최윤수, 장두성, NIL을 고려한 한국어 오픈 도메인 질의응답, 한국정보과학회 학술발표논문집, pp. 518-520, 2019.

[14] 민진우, 나승훈, 최윤수, 장두성, 지식베이스와 검색기반 QA의 결합모델에 기반한 오픈 도메인 질의응답, 한국정보과학회 학술발표논문집, pp. 569-571, 2019.

[15] S. Hochreiter and J. Schmidhuber, Long short-term memory, Neural computation, pp. 1735-1780, 1997.

[16] A. Vaswani, et al., Attention is all you need, Advances in neural information processing systems, pp. 5998-6008, 2017.

[17] Y. Kim, Convolutional Neural Networks for Sentence Classification, EMNLP, 2014.

[18] J. Devlin et al, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv preprint arXiv:1810.04805, 2018.