질의 응답 시스템을 위한 질의, 문서, 답변 검증기

민태홍O, 이재홍, 인수교, 문기윤, 조휘열, 김경덕
네이버
{th.min, jaehong.l, sookyo.in, kiyoon.moon, hwiyeol.jo, kyungduk.kim}@navercorp.com

Question, Document, Response Validator for Question Answering System

Tae Hong MinO, Jae Hong Lee, Soo Kyo In, Kiyoon Moon, Hwiyeol Jo, Kyungduk Kim
NAVER Corporation

요 약

본 논문은 사용자의 질의에 대한 답변을 제공하는 질의 응답 시스템에서, 제공하는 답변이 사용자의 질의에 대하여 문서에 근거하여 올바르게 대답하였는지 검증하는 QDR validator에 대해 기술한 논문이다. 본 논문의 과제는 문서에 대한 주장을 판별하는 자연어 추론(Natural Language inference, NLI)와 유사한 과제이지만, 문서(D)와 주장(R)을 포함하여 질의(Q)까지 총 3가지 종류의 입력을 받아 NLI 과제보다 난도가 높다. QDR validation 과제를 수행하기 위하여, 약 16,000 건 데이터를 생성하였으며, 다양한 입력 형식 실험 및 NLI 과제 데이터 추가 학습, 임계 값 조절 실험을 통해 최종 83.05% 우수한 성능을 기록하였다

주제어: 질의 응답 시스템, 자연어 처리, 자연어 추론, 기계학습

1. 서론

질의 응답 시스템은 질의를 입력 받아 그에 적절한 답변을 제공하는 시스템이다. 해당 시스템을 구현하기 위해 미리 질문 및 응답을 생성하여, 조건에 따라 사용자에게 제공하거나, 해당 상황에 맞는 문장을 생성하여 제공하는 방법이 있다[1, 2].

질의 응답 시스템의 질의에 대한 답변을 미리 준비하면 서비스 제공자의 검증된 답변을 제공할 수 있는 장점이 있지만 많은 시간과 비용이 요구되며, 사용자 질의 범위가 너무 넓어 모든 답변을 준비할 수 없다. 이러한 단점을 해결하기 위하여 기 구축 질의-응답에 매칭 되지 않는 질의에 대하여 답변을 생성하는 그림 1과 같은 질의 응답 시스템이 필요하다.

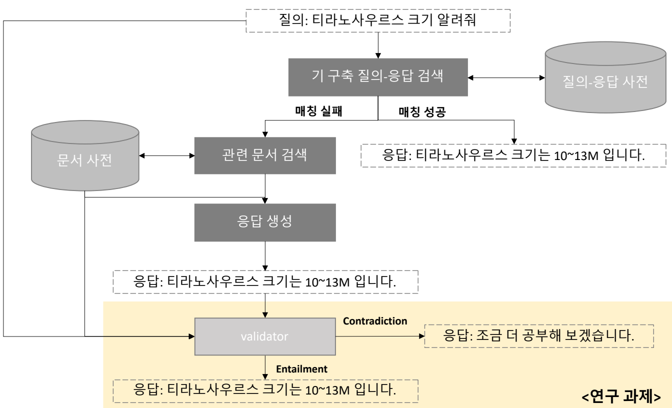

그림 1. 질의 응답 시스템 구조[1, 2]

그림 1과 같이 질의가 입력이 되면, 기 구축된 질의-응답 사전을 검색하여 매칭 된 결과를 사용자에게 제공한다. 매칭 된 결과가 없으면 질의와 관련된 문서를 검색한 후 해당 문서를 기반한 응답을 생성해 사용자에게 제공한다.

사실인 응답을 생성하기 위하여, 문서를 근거로 답변을 생성해야 한다. 하지만 잘 못된 문서를 근거로 하였거나, 문서에서 잘 못된 영역을 근거로 응답을 생성하면, 사용자의 질의를 충족시키지 못하는 잘 못된 답변이 생성된다.

잘 못된 답변이 제공되면 서비스 품질이 저하될 수 있어, 사용자 질의에 대하여 생성된 답변이 문서의 올바른 영역에 근거하여 생성되었는지 판별하는 이진 분류의 validator에 대하여 연구하였다.

2. 관련 연구

문서에 대한 주장에 대해 판별하는 대표적인 과제는 GLUE benchmark 중 MNLI(the Multi-Genre Natural Language Corpus)이 있다[3]. 한국어 데이터로는 KLUE benchmark 중 NLI(Natural Language Inference)가 있으며, MNLI와 유사한 과제이다[4]. KLUE의 NLI에 대한 예시는 다음 그림 2와 같다.

전제:	10명이 함께 사용하기 불편함없이 만족했다.	
가설 1:	10명이 함께 사용하기에 만족스러웠다	Entailment
가설 2:	10명이 함께 사용하기 불편함이 많았다.	Contradiction
가설 3:	성인 10명이 함께 사용하기 불편함없이 없었다.	Neutral

그림 2. KLUE NLI 데이터 예시[4]

그림 2와 같이 NLI 과제는 전제(Premise)에 대하여 가설(Hypothesis)이 주어지고, 가설 1과 같이 전제와 가설이 서로 타당하면 Entailment로, 가설 2와 같이 전제와 가설이 모순되면 Contradiction로, 가설 3과 같이 전제로 가설을 설명하기 어렵거나 관련성이 없는 문장이면 Neutral로 판별하는 과제이다.

GLUE나 KLUE benchmark 데이터를 이용한 연구들은 Transformer 모델을 이용한 연구들이 우수한 성능을 보였다[5, 6]. 일반적으로 NLI 과제는 전제와 가설 2가지 정보를 [SEP] 토큰으로 연결한 입력을 사용한다.

본 논문의 과제는 NLI 과제보다 확장된 과제로 전제와 가설뿐만 아니라 질의까지 포함된 과제이다. 기존에 연구된 결과를 바로 적용할 수 없어 다음과 같이 변형하였다.

3.1 QDR validation

본 논문에서 풀고자 하는 문제는 사용자의 질의(Question)에 대해 질의 응답 시스템이 문서(Document)를 참고하여 응답(Response)을 타당하게 생성하였는지 검증하는 QDR validation이다. 질의, 문서, 응답이 모두 서로 타당하다면 Entailment로, 하나라도 타당하지 않으면 Contradiction으로 이진 분류한다. QDR validation에 대한 예시는 다음 그림 3과 같다.

제목: 티라노사우루스

문서: 용반목 수각류에 속하는 육식 공룡으로 모든 공룡들을 대표하는 가장 유명한 공룡이다. 중생대 백악기 후기의 끝무렵인 마스트리히트절에 북아메리카에서 서식했다. 몸길이는 10.5~13m 사이, 몸높이 3.5~4.5m, 몸무게 6~9t 가량이다.

질의: 티라노사우르스 크기 알려줘

응답 1: 티라노사우르스의 몸길이는 10.5~13m 사이, 몸높이 3.5~4.5m 입니다.
-> Entailment

응답 2: 티라노사우루스의 몸무게는 약 6~9t 입니다.
-> Contradiction

그림 3. QDR validation 예시[7]

그림 3과 같이 NLI 과제의 Premise에 해당하는 문서 정보와 Hypothesis에 해당하는 응답 정보가 있으며, NLI 과제에 없는 제목 및 질의 정보를 포함하고 있다. NLI 과제의 Premise는 간단한 문장이었지만, QDR validation는 여러 문장으로 구성된 문서가 주어진다. 또한 위 그림 3의 문서와 같이 주어가 생략되어 설명하는 경우가 많다. 주로 생략된 주어가 제목에 해당하는 경우이기에 제목의 정보도 같이 사용한다.

그림 3의 응답 1은 질의에 대해 문서의 내용을 올바르게 대답하여 Entailment에 해당하는 응답이다. 하지만 응답 2는 문서에 기반한 대답으로 NLI 과제의 Entailment에 해당하지만, QDR validation은 질의와 맞지 않기에 Contradiction에 해당하는 응답이다.

3.2 제안 모델: QDR validator

QDR validation 과제를 수행하기 위해 NLI 과제에서 우수한 성능을 보이는 Transformer 모델을 이용하여 모델을 구성하였다. 본 논문에서 제안하는 QDR validator 모델에 대한 구조는 다음 그림 4와 같다.

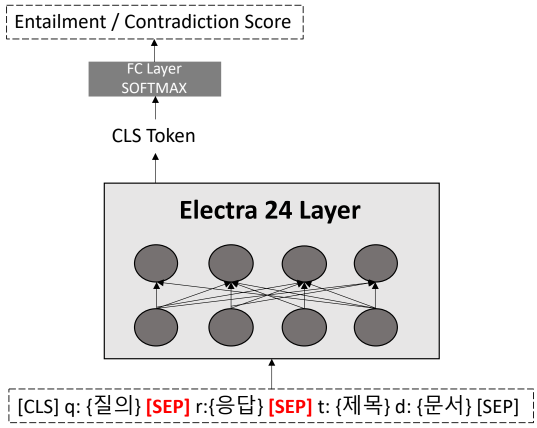

그림 4. QDR validator 모델 구조

그림 4와 같이 입력의 시작은 [CLS], 끝은 [SEP] 토큰으로 구성된다. 질의는 q:, 제목은 t: 문서는 d:, 응답은 r:를 추가하여 각각 [SEP] 토큰으로 연결한다. 이를 Electra Large 모델에 입력하고, 입력의 맨 앞에 있는 [CLS] 토큰을 입력에 대한 벡터로 취급하여, Fully Connected layer로 Entailment, Contradiction 점수를 계산한다.

4. 실험

QDR validator 학습 및 평가는 자체 제작한 데이터를 사용하였다. 위키 데이터에서 PAQ(Probably-Asked Questions) 을 생성한 후 응답 생성기를 이용하여 응답을 생성한 다음 이를 사람이 라벨을 태깅하는 방식으로 데이터를 생성하였다[8]. 해당 방법은 Entailment의 데이터가 많이 생성될 수 있지만, Contradiction 데이터는 생성하기 어렵다. 따라서, 다음 그림 5의 방법으로 응답을 변형하여 Contradiction 데이터를 추가 생성하였다.

숫자 변환: 티라노사우루스의 크기는 10~13M 입니다 ->
티라노사우루스의 크기는 **6-8M** 입니다

주어 자르기: 티라노사우루스의 크기는 10~13M 입니다 ->
라노사우루스의 크기는 10~13M 입니다
티라노사우루의 크기는 10~13M 입니다

열거 생략: 목성의 4대 위성은 이오, 유로파, 가니메데, 칼리스토 입니다.->
목성의 4대 위성은 **이오, 가니메데, 칼리스토** 입니다.

그림 5. Contradiction 데이터 생성 예시

그림 5의 숫자 변환은 생성된 응답에서 숫자가 있을 경우, 문서에 있는 다른 숫자로 변경한다. 주어 자르기

는 주어에 해당하는 단어의 앞 뒤 음절을 삭제한다. 열거 생략은 생성된 응답이 열거 형식일 경우 중간의 요소 하나를 제거한다. 생성한 데이터 수량은 다음 표 1과 같다.

표 1. 학습 데이터 및 평가 데이터 수량

구분	Entailment 개수	Contradiction 개수	전체 개수
학습 데이터	5,356	9,340	14,696
평가 데이터	549	602	1,151

표 1의 평가 데이터는 Contradiction의 비율이 52.03% 이기에, 모두 Contradiction으로 분류하는 모델의 성능은 52.03%가 된다. 이를 baseline으로 설정한다. 표 1의 학습 데이터를 학습한 모델의 평가 데이터 성능은 다음 표 2와 같다.

표 2. 입력 형식 별 실험 성능

번호	입력 형식	성능 (ACC, %)
1	전부 Contradiction으로 분류	52.03
2	[CLS] q: {질의} [SEP] a:{응답} [SEP] t:{제목} d: {문서} [SEP]	**82.45**
3	[CLS] q: {질의} t:{제목} d: {문서} [SEP] a:{응답} [SEP]	79.49
4	[CLS] t:{제목} d: {문서} [SEP] q: {질의} a:{응답} [SEP]	81.92

표 2의 2번은 질의, 문서, 응답 사이에 [SEP] 토큰을 이용하여 구분한 것이며, 3번과 4번은 [SEP] 구분되는 정보에 여러 정보를 같이 구성하였다. 1번 입력 형식과 같이 학습할 경우 82.45%로 입력 정보들을 각각 [SEP]로 구분하는 것이 가장 성능이 우수함을 보였다. 4번이 3번에 비해 2.43%p 더 높은 성능을 기록하여, [SEP] 토큰을 기준으로 구성요소에 따라 성능이 달라짐을 확인하였다.

KLUE의 NLI 학습 및 테스트 데이터는 약 28,000개로 방대한 양의 데이터가 공개되어 있다[4]. NLI 데이터 수량은 다음 표 3과 같다.

표 3. NLI 데이터 개수

구분	Entailment 개수	Neutral + Contradiction 개수	전체 개수
KLUE NLI	9,561	18,438	27,999

NLI에는 Neutral 라벨은, QDR validation에서 문서와 무관한 응답으로, Contradiction으로 변환하여 사용한다. NLI 데이터는 질의가 없는 데이터이지만, 양질의 데이터이기에 이를 학습할 경우 문서와 질의에 대한 관계를 학습하는데 도움이 될 수 있어 학습 데이터로 추가하였다. 성능이 가장 우수하였던, 표 2의 2번 입력 형식을 사용하였으며, NLI 데이터는 질의가 없는 데이터이기에 다음 그림 6과 같이 질의를 보정하여 입력으로 사용한다.

빈칸 질의 사용.	[CLS] q: [SEP] r: {응답} [SEP] d: {문서} [SEP]
질의 정보 미제공.	[CLS] r: {응답} [SEP] d: {문서} [SEP]

그림6. NLI 데이터 입력 형식

그림 6의 빈칸 질의 사용은 질의 정보를 빈칸(“ ”)으로 NLI 데이터도 QDR validator 데이터와 같은 형식으로 학습된다. 질의 정보 미제공은 빈 질의 및 무의미한 [SEP] 토큰이 노이즈로 작용할 수 있어, 질의에 대한 정보를 없이 응답과 문서만으로 NLI 데이터를 표현하였다. NLI 데이터를 같이 학습할 경우 성능은 다음 표 4와 같다.

표 4. NLI 데이터 추가 실험

번호	학습 데이터	입력 형식	성능 (ACC, %)
1	QDR 데이터 단독 학습	표2의 2번	82.45
2	NLI 데이터 단독 사용	빈칸 질의 사용	59.25
3	QDR 및 NLI 데이터 학습	빈칸 질의 사용	82.62
4		질의 정보 미제공	**82.79**

표 4의 NLI 데이터를 같이 학습한 3, 4번은 QDR 데이터만 사용한 1번에 비하여 0.17%p, 0.31%p 향상된 82.62%, 82.76% 성능을 기록하였다. NLI 데이터만으로 학습한 2번의 경우 QDR validator 평가 데이터에 대해 59.25% 성능을 기록하였다. 질의가 없는 데이터를 학습하기에 질의가 있는 QDR 평가 데이터에서 낮은 성능을 기록하였지만 QDR validator와 같이 학습할 경우 입력된 정보들의 관계를 이해하는데 도움을 주었다. 질의, 문서, 응답의 완전한 데이터가 아니더라도 같이 학습하면 성능이 향상될 가능성을 확인하였다.

가장 성능이 우수한 표 4의 3번 모델의 라벨 별 성능은 다음 표 5와 같다.

표 5. 라벨 별 성능 (%)

라벨	정확률	재현율	F1
Entailment	79.51	87.16	83.16
Contradiction	86.56	78.64	82.41

표 5와 같이, Entailment는 재현율이 정확률보다 높으며, 반대로 Contradiction은 정확률이 재현율보다 높게 측정되었다. 이는 모델이 Entailment로 많이 추론하고 있어 Entailment로 판단되는 수량을 줄이면 성능이 향상 될 수 있다. 모델이 계산하는 Score에 임계 값을 주어, 특정 임계 값 이상이면 Entailment, 미만이면 Contradiction으로 분류하도록 한다. 기존 실험의 임계 값은 0.5로 진행되었으며, 임계 값 변화에 따른 성능은

다음 표 6과 같다.

표 6. 임계 값 별 성능 (%)

임계 값	Contradiction F1	Entailment F1	ACC
0.5	82.41	83.16	82.79
0.6	82.55	82.85	82.71
0.7	**83.23**	**82.87**	**83.05**
0.8	82.82	81.68	82.27
0.9	81.95	79.44	80.79

표 6의 성능과 같이 임계 값이 증가함에 따라 Contradiction으로 분류하는 것이 많아져, Contradiction의 성능은 상승하고 Entailment의 성능은 하락한다. 그리하여 임계 값 0.7에서 전체적인 성능 83.05%로 임계 값 0.5의 82.79% 보다 0.26%p 상승하여 가장 우수한 성능을 보였다.

5. 결론

본 논문은 질의, 문서, 응답 총 3 가지 입력에 대해 서로 타당한 경우 Entailment로 타당하지 않을 경우 Contradiction으로 이진 판별하는 validator에 대하여 기술하였다. NLI 과제와 비슷한 과제이지만, 질의에 대한 정보가 추가되어 NLI 과제보다 높은 난도의 과제이다.

QDR validator를 학습하기 위하여 데이터를 자체 제작하였으며, Contradiction 데이터를 확보하기 위하여 생성된 응답에 노이즈를 추가하였다.

또한, 질의 정보를 같이 학습하기 위한 여러 입력 형식 중 가장 우수한 입력 형식을 확인하였다. 또한 양질의 오픈 데이터인 NLI 데이터도 함께 학습을 진행하여 질의가 없는 불완전한 QDR 데이터도 같이 학습할 경우 성능이 향상될 수 있음을 확인하였다. 더욱 모델의 성능을 향상하기 위하여, 모델의 임계 값을 조절하였으며 최종 83.05% 성능을 기록하였다.

향후 연구는 문서에 기반한 정보만이 아닌 윤리적으로 문제 같이 문서의 정보만으로 판단이 불충분한 범위까지 판별하는 validator를 개발하고자 한다.

참고문헌

[1] J. Menick, M. Trebacz, V. Mikulik, J. Aslanides, F. Song, M. Chadwick, G. Mia, Y. Susannah, C.G. Lucy, I. Geoffrey and M.A. Nat, Teaching language models to support answers with verified quotes. arXiv preprint arXiv:2203.11147, 2022.

[2] K. Shuster, J. Xu, M. Komeili, D. Ju, Smith, E. M. Smith, R. Stephen, U. Megan, C. Moya, A. Kushal, L. Joshua, B. Morteza, N. William, P. Spencer, G. Naman, S. Arthur, B. Y-Lan, K. Melanie and W. Jason BlenderBot 3: a deployed conversational agent that continually learns to responsibly engage. arXiv preprint arXiv:2208.03188, 2022.

[3] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, amd S. R. Bowman, GLUE: A multi-task benchmark and analysis platform for natural language understanding, arXiv preprint arXiv:1804.07461. 2018.

[4] S. Park, J. Moon, S. Kim, W. I. Cho, J. Han, J. Park, S. Chisung, K. Junseong, S. Yongsook, O. Taehwan, L. Joohong, O. Juhyun, L. Sungwon, J. Younghoon, L. Inkwon, S. Sangwoo, L. Dongjun, K. Hyunwoo, L. Myeonghwa, J. Seongbo, D. Seungwon, K. Sunkyoung, L. Kyungtae, L. Jongwon, P. Kyumin, S. Jamin, K. Seonghyun, P. Lucy, O. Alice, H. Jung-Woo and C. Kyunghyun, Klue: Korean language understanding evaluation. arXiv preprint arXiv:2105.09680, 2021.

[5] K. Clark, M. Luong, Q. V. Le and C.D. Manning, Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555, 2020.

[6] J. Devlin, M. W. Chang, L. Kenton and K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[7] https://namu.wiki/w/%ED%8B%B0%EB%9D%BC%EB%85%B8%EC%82%AC%EC%9A%B0%EB%A3%A8%EC%8A%A4

[8] P. Lewis, Y. Wu, L. Liu, P. Minervini, H. Küttler, A. Piktus, P. Stenetorp and S. Riedel, Paq: 65 million probably-asked questions and what you can do with them. Transactions of the Association for Computational Linguistics, 9, 1098-1115, 2021.