문장 표현 단위를 활용한 기계독해 시스템

장영진[1], 이현구[2], 신동욱[2], 박찬훈[2], 강인호[2], 김학수[1]

건국대학교 인공지능학과[1], 네이버[2]

{danyon, nlpdrkim}@konkuk.ac.kr
{hyeongu.lee, shin.dongwook, chanhoon.park, once.ihkang}@navercorp.com

Machine Reading Comprehension System using Sentence units Representation

Youngjin-Jang[1], Hyeon-gu Lee[2], Dongwook-Shin[2], Chan-hoon Park[2], Inho-Kang[2], Harksoo-Kim[1]
Konkuk University Department of Artificial Intelligence[1], Naver[2]

요 약

기계독해 시스템은 주어진 질문에 대한 답변을 문서에서 찾아 사용자에게 제공해주는 질의응답 작업 중 하나이다. 하지만 대부분의 기계독해 데이터는 간결한 답변 추출을 다루며, 이는 실제 애플리케이션에서 유용하지 않을 수 있다. 실제 적용 단계에서는 짧고 간결한 답변 뿐 아니라 사용자에게 자세한 정보를 제공해줄 수 있는 긴 길이의 답변 제공도 필요하다. 따라서 본 논문에서는 짧은 답변과 긴 답변 모두 추출할 수 있는 모델을 제안한다. 실험을 통해 Baseline과 비교하여 짧은 답변 추출에서는 F1 score 기준 0.7%, 긴 답변 추출에는 1.4%p의 성능 향상을 보이는 결과를 얻었다.

주제어: 기계독해, 짧은 답변 추출, 긴 답변 추출

1. 서론

기계 독해(Machine Reading Comprehension: MRC)는 주어진 문서에서 질문에 대한 답변을 추출하는 질의응답 작업 중 하나이다. 대부분의 기계 독해 작업은 짧은 길이의 간결한 답변 추출에 초점을 맞추었으며, 대용량 말뭉치를 기반으로 한 언어 모델[1]이 등장한 이후, 사람보다 뛰어난 성능을 보였다. 이에 따라 더 어려운 문제를 해결하려는 시도[2-3]가 있었으며, 대표적으로 긴 길이의 답변 추출 작업도 포함하는 KorQuAD 2.0[4]에서 기존 기계독해 프레임워크는 성능이 하락하는 문제가 발생했다. 긴 길이의 답변 추출은 실제 애플리케이션에서 사용자에게 설명이나 정의 등과 같은 정보를 제공해주는데 필요하며, 위에 언급한 문제는 부착되어있는 답변 길이의 편차가 클수록 심화되었다. 다시 말해, 기계독해 시스템이 짧은 답변 데이터만을 다루거나 긴 답변 데이터만을 다룰 때 보다 위의 두 데이터를 동시에 다룰 때 성능 하락은 크게 발생한다. 따라서 본 논문에서는 추출해야하는 답변 길이와 상관없이 성능 하락을 줄일 수 있는 모델을 제안하고자 한다. 제안 모델은 사전 부착된 답변을 토큰과 문장 단위로 표현하여 학습하고 위의 두 정보를 결합하여 성능 하락 문제를 완화시키고자 한다.

2. 관련 연구

대표적인 기계독해 데이터 셋은 SQuAD 1.1[5]과 KorQuAD 1.0[6]이 있다. 위 데이터 셋은 모두 짧고 간결한 답변 추출을 목적으로 구축되었으며 사용자 질문과 문서 사이의 양방향 주의 집중을 계산하는 BiDAF[7]를 주축으로 활발히 연구되었다. 이후 대용량 언어 말뭉치를 기반으로 학습된 BERT(Bidirectional Encoder Representations from Transformers)가 등장했고, 11가지의 자연어 처리 작업에서 최첨단의 성능을 보였다. 특히 기계 독해의 경우 사람보다 더 나은 성능을 보여주었으며, 이로 인해 간결하고 짧은 답변 추출에서 더 복잡한 문제로 확장한 데이터 셋이 공개되었다. 대표적으로 KorQuAD 2.0는 HTML 태그를 포함한 긴 문서에서 짧고 간결한 답변과 테이블 전체 등과 같은 긴 길이의 답변을 포함하고 있다.

3. 제안 모델

제안 모델은 인코더(Encoder), 문장 표현 모델링 계층(Sentence Representation Layer), 출력 계층(Output Layer)로 구성된다. 각 계층에 대한 자세한 내용은 다음과 같다.

3.1 사전 학습 모델을 이용한 인코더

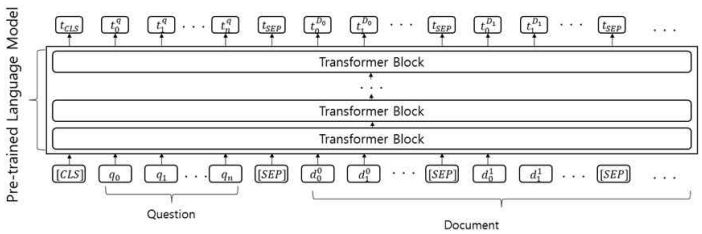

그림 1. 제안 모델 인코더

제안 모델의 인코더는 위의 그림 1처럼 WordPiece [8]로 토큰화 된 질문 Q와 문서 D를 “[CLS] Q [SEP] D

[SEP]" 형태로 입력받으며, 문서를 구성하는 문장 사이에 [SEP] 토큰을 삽입하여 문장 구분 정보를 반영했다. 언어 모델 인코더를 통해 벡터화된 질문과 문서의 각 토큰은 아래의 수식 1과 같이 t_k로 표현된다.

$$t_k = RoBERTa(Q, D) \quad (1)$$

3.2 문장 표현 모델링 계층

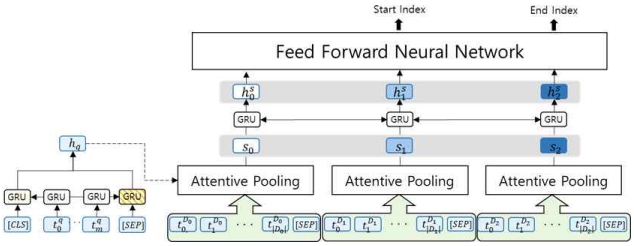

그림 2. 문장 표현 모델링 계층 구조

서론에서 언급했듯, 본 논문에서는 문장 단위 표현과 토큰 단위 표현을 통해 사전 부착된 답변 범위를 학습한다. 문장 표현 모델링 계층은 입력 문서의 각 문장을 단일 벡터를 표현하기 위해 사용되었으며 이에 대한 구조는 위의 그림 2와 같다. 질문 벡터 s_j는 인코딩된 질문 $(t_0^q, t_1^q, ..., t_{|q|}^q)$와 입력 문서의 각 문장 $D^j = \{t_0^{D_j}, t_1^{D_j}, ..., t_{|D^j|}^{D_j}\}$ 사이의 주의 집중을 통해 계산된다. 여기서 t_i^q는 질문의 i번 째 토큰의 벡터를 의미하고 $t_k^{D_j}$는 문서를 구성하는 j번 째 문장 D^j의 k번 째 토큰 벡터를 의미한다. 문장 벡터를 생성하는 과정은 다음과 같다. 첫 번째로 인코딩된 질문 벡터를 양방향 GRU(Gated Recurrent Units)[9]를 이용하여 단일 질문 벡터(h_q)를 생성한다. 이에 대한 수식은 아래의 식 (2)와 같다.

$$\begin{aligned} h_i^f &= GRU^f(h_{i-1}^f, t_i^q) \\ h_i^b &= GRU^b(h_{i+1}^b, t_i^q) \\ h_q &= [h_0^b; h_{|q|}^f] \end{aligned} \quad (2)$$

위 수식에서 GRU^f와 GRU^b는 각각 순방향 GRU와 역방향 GRU를 의미하고 (;)기호는 Concatenate를 의미한다. 다음으로 수식 (1)에서 생성된 단일 질문 벡터 h_q와 인코딩된 문장 D_j 사이의 주의 집중을 계산하여 단일 질문 벡터(s_j)를 생성한다. 이에 대한 수식은 아래의 (3)과 같다.

$$\begin{aligned} S_j &= h_q \times D_j^T \in R^{1 \times |D_j|} \\ a^{D_j} &= Softmax(S_j) \\ s_j &= \sum_k^{|D_j|} a_k^{D_j} t_k^{D_j} \in R^H \\ h^s &= BiGRU(s_j, h_{j-1}^s, h_{j+1}^s) \end{aligned} \quad (3)$$

위 수식에서 생성된 단일 문장 벡터 s_j는 문장 사이의 순차정보를 반영하기 위해 양방향 GRU를 통해 $h^s \in R^{|D| \times 2H}$으로 표현된다. 이때 h^s는 FFNN(Feed Forward Neural Network)과 Cross Entropy 손실 함수를 통해 문장 단위로 표현된 정답 위치를 학습한다. 이때 정답 답변이 문장 단위로 표현되지 않는 경우(짧고 간결한 답변)는 정답을 포함하는 문장을 정답으로 학습한다.

3.3 최종 출력 계층

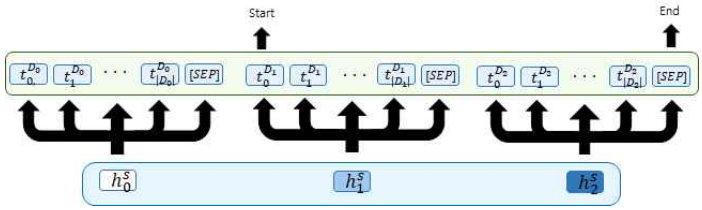

그림 3. 최종 출력 계층 구조

제안 모델의 최종 출력 계층은 토큰 단위로 표현된 정답을 학습한다. 위의 그림 3과 같이 3.2절에서 언급한 문장 벡터 표현과 언어 모델의 출력 벡터를 연결하여 최종 토큰 표현을 생성한다. 최종 토큰 벡터는 문장 표현 모델링 계층과 마찬가지로 FFNN과 Cross Entropy 손실 함수를 통해 토큰 단위로 표현된 정답 위치를 학습한다.

4. 실험

본 논문에서는 KorQuAD 2.0의 일부(43,712 쌍의 학습 데이터, 1,650 쌍의 검증 데이터)를 실험에 사용했으며, 문서는 문장 단위 답변 학습을 위해 간단한 정규식으로 문장 분할을 수행했다. 부착되어있는 답변은 정규식에 의해 2 문장 이상으로 이루어져있는 답변을 긴 길이의 답변으로 간주했으며 자세한 통계는 아래의 표 1과 같다. 성능 평가에 사용된 지표는 정답 완전 일치 (Exact Matrix :EM)과 F1 점수를 사용했다. 제안 모델에서 사용한 사전학습 언어 모델은 20GB의 한국어 말뭉치로 학습한 ELECTRA[10]를 사용했다. 제안 모델 학습에 적용된 드랍아웃과 학습률은 각각 0.1과 5e-5로 설정했으며, Adam optimizer[11]를 사용하여 최적화를 진행했다.

표 1. 데이터 통계

	Train set	Develop Set
Short Answer	31,751	1,284
Long Answer	11,961	366
All Answer	43,712	1,650

4. 1 실험 결과

아래의 표 2는 논문에서 진행한 실험 결과를 보여주며 실험에 사용된 비교모델은 다음과 같다. Baseline은 ELECTRA-base 인코더 기반의 일반적인 기계독해 프레임워크를 의미한다. Baseline-Short (이하, Short 모델)은 Baseline을 Short Answer 데이터로만 학습 및 평가를 진행한 모델을 의미한다. Baseline-Long (이하, Long 모

표 2. 실험 결과

Model	Short Answer		Long Answer	
	EM	F1 score	EM	F1 Score
Baseline	57.0	73.0	45.9	69.6
Baseline-Short	58.8	74.5	-	-
Baseline-Long	-	-	46.9	74.0
Sentence Rep	-	-	50.5	78.2
Proposed Model	58.8	73.7	45.6	71.0

델)은 Baseline을 Long Answer 데이터로만 학습 및 평가를 진행한 모델을 의미한다. Sentence Rep(이하, Sentence 모델)은 본 논문에서 제안하는 문장 단위 표현을 통해 Long Answer 데이터로만 학습 및 평가를 진행한 모델을 의미한다. 마지막으로 Proposed Model은 본 논문에서 제안하는 모델을 의미한다.

표 1에서 Short 모델과 Baseline을 비교했을 때, F1 score 기준으로 Baseline의 성능이 1.5%p 하락하는 것을 알 수 있고, Long 모델과 Baseline을 비교했을 때, F1 score 기준으로 4.4%p의 성능 하락이 발생하는 것을 알 수 있다. 이를 통해 Short Answer 데이터와 Long Answer 데이터를 동시에 학습할 경우 둘 중 하나만 학습한 모델보다 성능이 떨어지는 것을 알 수 있다. 그리고 Long 모델과 Sentence 모델의 성능 비교를 통해 긴 길이의 답변은 문장 단위 표현을 이용하여 답변을 추출하는 것이 효과적인 것을 알 수 있다. 그리고 제안 모델의 성능은 Short 모델과 Long 모델에 비교하여 낮은 성능을 보이지만, Baseline과 비교해서 F1 score 기준 Short Answer에서 0.7%p Long Answer에서 1.4%p의 성능 향상이 있음을 알 수 있다. 이를 통해 본 논문에서 제안하는 모델이 답변 길이 편차에 따른 성능 하락 문제를 미미하지만 완화하는 것을 알 수 있다.

5. 결론

본 논문에서는 기계독해에서 답변 길이의 편차가 클수록 성능이 하락하는 문제점을 완화하기 위한 기계독해 모델을 제안했다. 문장 단위 표현과 토큰 단위 표현을 함께 사용하여 성능 하락을 완화시키는 결과를 확인할 수 있었지만 성능 향상 폭이 크지 않은 결과를 보였기에 향후에 이를 보완할 수 있는 연구를 진행할 예정이다.

감사의 글

본 연구는 네이버 산학연구용역 과제의 지원을 받아 수행되었음

참고문헌

[1] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In NAACL-HLT. 2019.

[2] Rajpurkar, P., Jia, R., & Liang, P. “Know What You Don’t Know: Unanswerable Questions for SQuAD”. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) pp. 784-789. 2018.

[3] Dasigi, P., Liu, N. F., Marasović, A., Smith, N. A., & Gardner, M. “Quoref: A Reading Comprehension Dataset with Questions Requiring Coreferential Reasoning”. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) pp. 5925-5932. 2019.

[4] 김영민, 임승영, 이현정, 박소윤, & 김명지. “KorQuAD 2.0: 웹문서 기계독해를 위한 한국어 질의응답 데이터셋”. 정보과학회논문지, 47(6), 577-586. 2020.

[5] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. “SQuAD: 100,000+ Questions for Machine Comprehension of Text”. In Empirical Methods in Natural Language Processing (EMNLP). 2016

[6] 임승영, 김명지, & 이주열. “KorQuAD: 기계독해를 위한 한국어 질의응답 데이터셋”. 한국정보과학회 학술발표논문집, 539-541. 2018.

[7] Seo, M., Kembhavi, A., Farhadi, A., & Hajishirzi, H. “Bidirectional Attention Flow for Machine Comprehension”. arXiv preprint arXiv:1611.01603. 2016.

[8] Sennrich, R., Haddow, B., & Birch, A. “Neural Machine Translation of Rare Words with Subword units”. arXiv preprint arXiv:1508.07909. 2015.

[9] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling", arXiv:1412.3555, 2014

[10] Clark, K., Luong, M. T., Le, Q. V., & Manning, C. D. (2020). “ELECTRA: Pre-training Text Encoders as Discriminators rather than Generators”. arXiv preprint arXiv:2003.10555. 2020.

[11] D. P. Kingma and J. Ba, “Adam: A method for Stochastic Optimization”, International Conference on Learning Representations, Vol. 5, 2015.