

정답과 구절의 공동 주의 집중 계층을 이용한 한국어 질문 생성

김진태[○], 노형종, 이연수, 김학수*

NLP Center Language AI Lab, 엔씨소프트
 강원대학교 IT대학 컴퓨터정보통신공학전공*

Kjt1505@ncsoft.com, nohhj@ncsoft.com, yeonsoo@ncsoft.com, nlpdrkim@kangwon.ac.kr

Korean Question Generation Using Co-Attention Layer of Answer and Passage

Jintae Kim[○], Hyungjong Noh, Yeonsoo Lee, Harksoo Kim*

NLP Center Language AI Lab, NCSoft
 Kangwon National University Computer and Communication Engineering*

요약

질문 생성이란 구절이 입력되면 구절에서 답을 찾을 수 있는 질문을 자동으로 생성하는 작업으로 교육용 시스템, 대화 시스템, QA 시스템 등 다양한 분야에서 중요한 역할을 한다. 질문 생성에서 정답의 단어가 질문에 포함되는 문제점을 해결하기 위해 구절과 정답을 분리한다. 하지만 구절과 정답을 분리하게 되면 구절에서 정답의 정보가 손실되고, 정답에서는 구절의 문맥 정보가 손실되어 정답 유형에 맞는 질문을 생성할 수 없는 문제가 발생된다. 본 논문은 이러한 문제를 해결하기 위해 분리된 정답과 구절의 정보를 연결시켜주는 정답과 구절의 공동 주의 집중 계층을 제안한다. 23,658개의 질문-응답 쌍의 말뭉치를 이용한 실험에서 정답과 구절의 공동 주의 집중 계층이 성능 향상에 기여해 우수한 성능(BLEU-26.7, ROUGE-57.5)을 보였다.

주제어: 질문 생성, 공동 주의 집중, 정보 연결

1. 서론

질문 생성은 자연어로 구성된 구절이 입력되면 구절에서 답을 찾을 수 있는 질문을 자동으로 생성하는 작업을 의미한다. 질문 생성 모델은 다양한 시스템에 중요한 역할을 한다. 교육 시스템에서는 교육 문서에 대한 질문을 생성하고 학생들에게 질문에 대한 답을 찾게 하여 학생들을 교육시킬 수 있다. 대화 시스템에서는 질문 생성을 통해 사용자로부터 특정 정보를 얻을 수 있고 사용자에게 질문을 주어 대화가 끊기는 것을 막을 수 있다. QA 시스템에는 필요한 질문-응답 쌍의 말뭉치를 자동으로 구축해 대용량의 말뭉치를 확보하기 위해 필요한 인력을 줄일 수 있다. 이러한 질문 생성 모델은 최근 SQuAD[1], MS MARCO[2]와 같은 대용량 QA 말뭉치 공개와 심층 신경망의 발달로 심층 신경망 모델을 이용한 질문 생성 연구가 활발히 진행되고 있다. 심층 신경망 모델을 이용한 질문 생성 모델은 구절에서 명사, 구 형태의 정답에 대한 고차원적인 질문 생성이 가능하다. 본 논문은 심층 신경망 모델을 기반으로 구절과 정답이 주어진 상황에서 질문을 생성하는 연구를 진행한다.

품질 좋은 신경망 기반의 질문 생성 모델은 구절의 내용을 정확하게 이해하고 정답과 관련이 있는 중요한 정보를 찾아야 한다. 그리고 찾은 정보를 적절히 복사해 정답과 관련된 질문을 생성해야 한다. 하지만 정답과 연

관이 있는 정보를 복사해 질문을 생성하는 과정에서 정답의 단어를 복사해 부적절한 질문이 생성되는 문제가 발생한다. 그러므로 [3]에서 제안했던 것처럼 구절과 정답을 분리해 구절 내 정답 단어가 복사되는 문제점을 해결해야 한다.

표 1 구절과 정답 분리 전후 비교 예시

정답이 분리되지 않은 구절	세종대왕은 1446년에 한글을 창제했다.
질문 생성 결과	세종대왕이 만든 한글이 무엇이야?
정답이 분리된 구절	세종대왕은 1446년에 -answer-을 창제했다.
정답	한글
개선된 결과	세종대왕이 만든 것은 무엇이야?

표 1은 구절과 정답 분리 전후 비교 예시 표이다. 정답이 분리되지 않은 구절 “세종대왕은 1446년에 한글을 창제했다.”에서 정답이 “한글”인 경우 질문을 생성하는 과정에서 구절의 “한글” 단어를 복사해 “세종대왕이 만든 한글이 무엇이야?”와 같이 질문에 정답이 포함되는 문제가 발생한다. 하지만 구절에서 “한글”을 다른 기호로 치환하고 정답인 “한글”을 분리하여 표현하게 되면 정답이 분리된 구절에서 “한글”이 등장하지 않아 정답이 복사되는 문제를 해결할 수 있다. 구절과

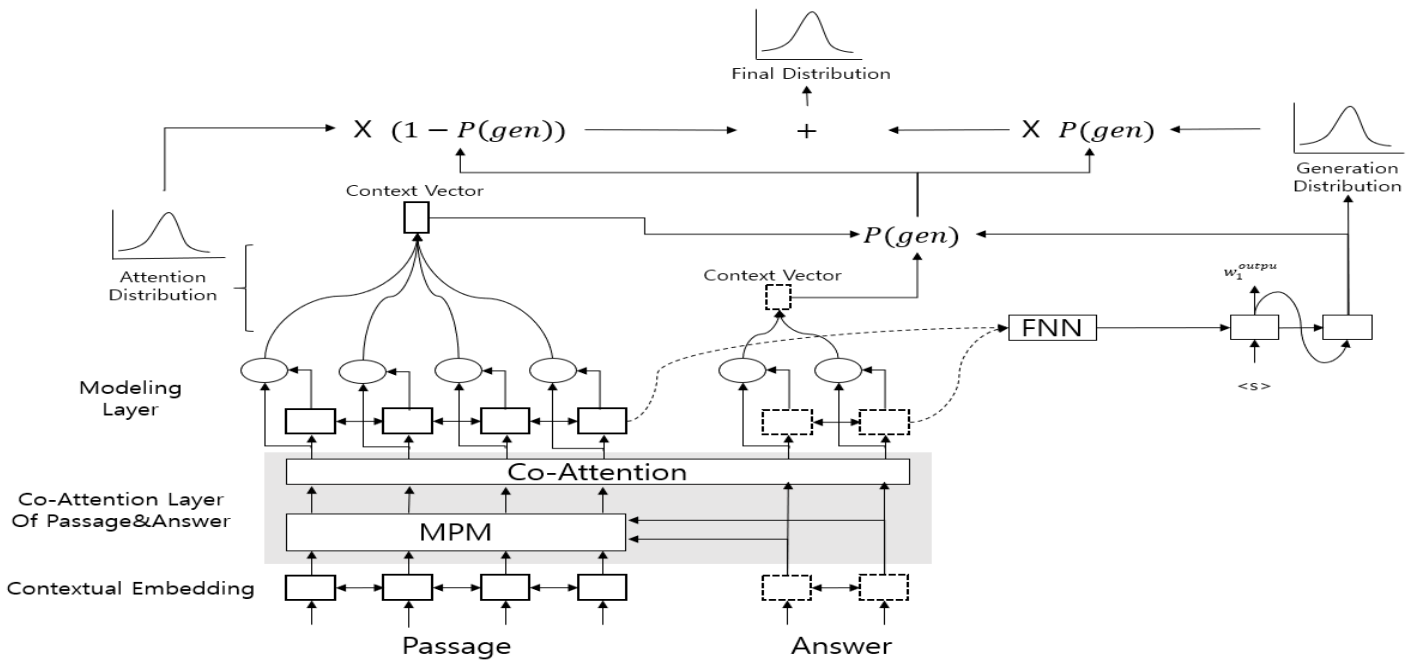


그림 1 모델의 전체 구조도

정답을 분리하면 구절내 정답의 단어가 복사되는 문제는 해결할 수 있지만 새로운 문제가 발생한다. 구절에서는 정답 정보가 손실되어 정답과 연관된 중요한 정보를 찾기 어려워지고 정답에서는 구절의 문맥이 손실되어 올바른 정답의 유형을 파악하지 못하는 문제가 발생한다. 이러한 문제를 해결하기 위해 구절과 정답의 정보를 연결해 손실된 정보를 반영하는 방법이 필요하다.

표 2 구절과 정답의 정보 연결 전후 비교 예시

정답이 분리된 구절	김범수는 -answer-에 제대한 후 5개월 만에 6집으로 활동했다.
정답	2008년 3월
정보가 분리된 결과	김범수가 활동한 날은?
정보가 연결된 결과	김범수가 제대한 날은?

표 2는 구절과 정답이 분리되었을 때 정보 연결 전후 비교 예시 표이다. 구절 “김범수는 -answer-에 제대한 후 5개월 만에 6집으로 활동했다.”와 정답 “2008년 3월”인 경우 정보가 분리된 상태에서 질문을 생성했을 때 “김범수가 활동한 날은?”과 같이 정답이 정확하게 어떤 날인지 파악하지 못하는 문제가 발생한다. 하지만 정보가 연결된 상태에서 질문을 생성하면 “김범수가 제대한 날은?”과 같이 정답이 정확하게 어떤 날인지 알고 질문을 생성할 수 있다. 본 논문에서는 구절과 정답을 분리함으로써 발생하는 정보 손실을 개선하기 위해 정답과 구절의 공동 주의 집중 계층을 제안하고 이를 사용하는 한국어 질문 생성 모델을 제안한다.

2. 관련 연구

최근 신경망 모델을 이용한 다양한 질문 생성 연구가 진행되고 있다. [4]은 구절에서 정답 위치에 태그 정보

를 추가해 정답 단어를 표기하고 주의 집중 방법 (Attention Mechanism)[5]을 적용한 시퀀스-투-시퀀스 (Sequence-to-Sequence) 모델을 이용해 질문을 생성했다. 하지만 복사 작용이 없어 질문 생성에 필요한 구절의 단어가 정확하게 생성되지 않았다. 이러한 문제를 해결하기 위해 복사 작용이 포함된 [6] 또는 [7]을 사용하는 것이 필요하다. 그리고 정답 위치에 추가된 태그 정보가 무시되어 정답과 연관된 중요한 정보를 찾지 못하는 문제가 발생했다. 이러한 문제점을 해결하기 위해 [8]은 다중 투시 매칭 방법(Multi-Perspective Matching)을 통해 정답과 연관이 있는 정보를 표기하여 질문 생성 성능을 향상시켰다. [9]는 정답 품사 태그를 사용하고 자가 주의 집중을 통해 정답과 연관이 있는 중요한 정보를 찾아 질문 생성 성능을 개선했다. 하지만 [8-9]는 여전히 인코더(Encoder)의 입력으로 사용되는 구절에 정답 단어가 포함되어 있어 질문을 생성할 때 정답 단어를 복사해 질문을 생성하는 문제가 발생했다. [3]은 구절에 포함된 정답의 단어가 복사되는 것을 막기 위해 구절과 정답을 분리시켜 정답의 단어가 복사되는 문제를 개선했지만 정답에서 구절의 문맥 정보가 사라지고, 구절에서 정답 정보가 사라지는 문제점이 발생한다. 본 논문은 이를 보완하기 위해 분리된 정답과 구절의 정보를 연결시켜주는 방법을 제안한다.

3. 정답과 구절의 공동 주의 집중 계층을 이용한 질문 생성 모델

그림 1은 본 논문에서 제안하는 모델의 전체 구조도를 의미한다. 모델은 [9]와 동일하게 구절과 정답을 분리하여 입력 받는다. 본 논문에서는 정답이 분리된 구절은 $Passage = [w_1^p, w_2^p, \dots, w_n^p]$ 로, 정답은 $Answer = [w_1^a, w_2^a, \dots, w_m^a]$ 로 표기한다. w_i^p 는 구절의 i 번째 단어,

w_i^a 는 정답의 i 번째 단어를 의미한다. Passage와 Answer는 문맥 반영 임베딩 계층(Contextual Embedding)에 입력된다. 문맥 반영 임베딩 계층은 각 단어에 주변 문맥 정보를 반영시킨 정보를 정답과 구절의 공동 주의 집중 계층(Co-Attention Layer of Passage & Answer)에 입력해 구절과 정답 정보를 연결시켜 손실되었던 정보를 반영시킨다. 그리고 공동 주의 집중 계층의 결과를 구조화 계층(Modeling Layer)에 입력한다. 구조화 계층은 새로 반영된 정보를 다듬는 역할을 한다. 디코더(Decoder)는 구조화 계층에서 생성된 은닉 정보(Hidden State)를 전방 전달 신경망(Feed-Forward Neural Networks)을 통해 추상화시킨 정보를 초기 상태로 입력 받고 [7]의 방법을 통해 구절의 단어를 복사하거나 질문에 필요한 단어를 생성한다.

3.1 문맥 반영 임베딩 계층

본 논문에서 사용하는 문맥 반영 임베딩 계층은 양방향 GRU(Gated Recurrent Units)[10]을 통해 문맥 정보를 반영한다. 구절과 정답을 분리한 내용이 입력되어 구절에는 정답의 정보가 포함되지 않고 정답에는 구절의 문맥 정보가 포함되지 않는다. 식 (1)의 방법을 통해 계산된다.

$$\begin{aligned} h_i^p &= \text{BiGRU}(h_{i+1}^p, h_{i-1}^p, p_i) \\ h_i^a &= \text{BiGRU}(h_{i+1}^a, h_{i-1}^a, a_i) \end{aligned} \quad (1)$$

식 (1)에서 p_i 는 Passage의 i 번째 단어 임베딩(Embedding), a_i 는 Answer의 i 번째 단어 임베딩을 의미한다. h_i^p 는 양방향 GRU를 통해 표현된 Passage의 i 번째 은닉 벡터(Hidden Vector)로 주변 문맥이 반영된 정보를 가지게 된다. h_i^a 는 양방향 GRU를 통해 표현된 Answer의 i 번째 은닉 벡터로 정답 내의 문맥이 반영된 정보를 가지게 된다.

3.2 정답과 구절의 공동 주의 집중 계층

3.2.1 다중 투시 매칭

정답과 구절의 공동 주의 집중 계층은 2단계로 구성된다. 먼저 다중 투시 매칭을 통해 구절의 단어 정보에 정답과 얼마나 연관성이 있는지 표기한다. 다중 투시 매칭은 두 정보의 연관성을 측정할 때 하나의 관점이 아니라 여러 관점을 이용해 더 정확히 측정하는 방법을 의미한다. 다중 투시 매칭은 구절의 단어 정보와 정답의 연관성을 확인하기 위해 사용된다. 예를 들어 구절 “세종대왕은 1446년에 -answer-를 창제했다”, 정답 “한글”인 경우 다중 투시 매칭을 통해 구절의 단어 중 “한글”과 연관된 “세종대왕”, “창제”에 높은 매칭 결과를 표기한다. 높은 매칭 결과를 가진 단어들은 질문 생성에 사용되거나 구절을 이해하는데 기여를 하게 되며 [7]에서 질문 생성에 효과적인 자질인 것을 확인했다. 본 논문은 다중 투시 매칭 자질을 3.2.2절의 정답과 구절의 공동 주의 집중 방법 단계에서 구절의 핵심어와 정답의 핵심

어 간의 정보 연결에 도움을 주는 역할로 사용한다.

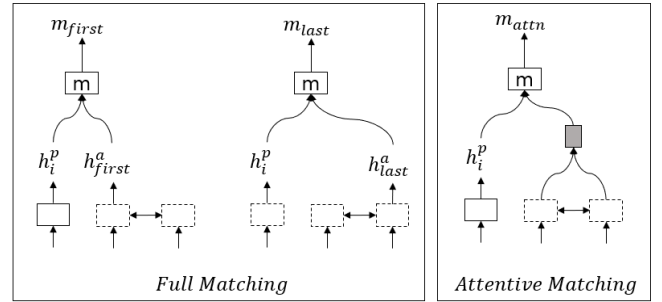


그림 2 다중 투시 매칭 방법

그림 2는 다중 투시 매칭 방법에 대한 그림이다. 본 논문에서는 2가지 매칭 방법을 사용한다. 완전 매칭(Full Matching)은 h_i^p 와 h_{i-1}^a 의 양 끝 은닉 벡터를 매칭하는 방법을 의미한다. 주의 집중 매칭(Attentive Matching)은 h_i^p 와 h_{i-1}^a 의 은닉 벡터 간의 주의 집중 가중치를 계산하고 계산된 가중치와 h_{i-1}^a 를 곱하고 모두 합한 정보를 의미한다. 매칭은 식 (2)의 방법을 통해 계산된다.

$$\begin{aligned} m(v_p, v_a) &= \cos(W_k v_p, W_k v_a) \\ m_i^{full} &= [m(h_i^p, h_{i-1}^a); m(h_i^p, h_{i-1}^a)] \\ m_i^{attn} &= m(h_i^p, \text{weighted sum}(h^a)) \\ m_i &= [m_i^{full}; m_i^{attn}] \\ h_i^p &= [h_i^p; m_i] \end{aligned} \quad (2)$$

식 (2)에서 매칭 함수 m 은 v_p 와 v_a 에 가중치를 곱하고 코사인 유사도를 통해 두 벡터의 연관성을 계산한다. m_i^{full} 는 완전 매칭, m_i^{attn} 는 주의 집중 매칭을 의미한다. h_i^p 는 다중 관점 매칭을 통해 연관성 자질을 h_i^p 에 추가한 결과이다.

3.2.2 정답과 구절의 공동 주의 집중 방법

다중 투시 매칭이 완료되면 정답과 구절의 공동 주의 방법이 진행된다. 정답과 구절의 공동 주의 집중을 통해 구절과 정답의 분리로 발생했던 정보 손실을 복구한다. 그 결과, 질문 생성에 필요한 구절 정보와 정답 정보가 생성된다.

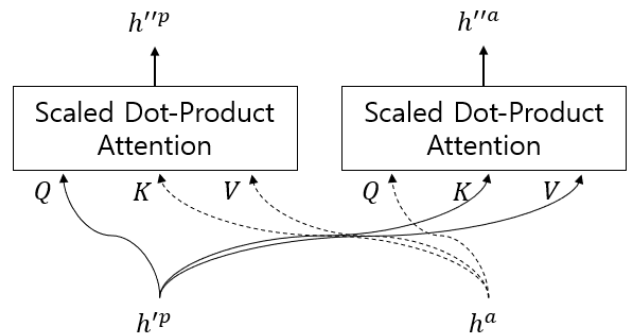


그림 3 정답과 구절의 공동 주의 집중 방법

그림 3은 정답과 구절의 공동 주의 집중 방법의 구조이다. 정답과 구절의 공동 주의 집중을 하기 전에 h^p 와 h^a 의 특징이 잘 표현될 수 있도록 전방 전달 신경망을 통해 가공한다. 정보 가공 과정에서 h^p 의 다중 투시 매칭 자질은 h^a 중 연관된 정보와 효과적으로 연결될 수 있는 정보로 가공하는데 기여한다. 가공된 구절 정보와 정답 정보는 [11]의 내적 주의 집중 방법(Scaled Dot-Product Attention)을 사용해 연관된 정보를 찾는다. 구절에서는 구절의 단어와 연관된 정답의 정보를 내적 주의 집중 방법을 사용해 찾고, 구절의 자가 주의 집중 정보, h^p 와 결합(Concatenate)된다. 정답에서는 정답의 단어와 연관된 구절의 문맥 정보를 내적 주의 집중 방법을 사용해 찾고, 정답의 자가 주의 집중 정보, h^a 와 결합된다. 식 (3)의 방법을 통해 계산된다.

$$\begin{aligned}
 \text{Attention}(Q, K, V) &= \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \\
 H^p &= \text{FNN}(h^p) \\
 H^a &= \text{FNN}(h^a) \\
 \text{attn}_{co}^p &= \text{Attention}(H^p, H^a, H^a) \\
 \text{attn}_{self}^p &= \text{Attention}(H^p, H^p, H^p) \\
 h''^p &= [h^p; \text{attn}_{co}^p; \text{attn}_{self}^p] \\
 \text{attn}_{co}^a &= \text{Attention}(H^a, H^p, H^p) \\
 \text{attn}_{self}^a &= \text{Attention}(H^a, H^a, H^a) \\
 h''^a &= [h^a; \text{attn}_{co}^a; \text{attn}_{self}^a]
 \end{aligned} \quad (3)$$

식 (3)에서 $\text{Attention}(Q, K, V)$ 은 내적 주의 집중 방법의 수식을 의미한다. H^p 와 H^a 는 전방 전달 신경망을 이용해 h^p 와 h^a 를 가공한 정보를 의미한다. attn_{co}^p 은 내적 주의 집중 방법을 이용해 H^p 와 연관된 H^a 를 계산한 결과, attn_{self}^p 는 H^p 를 자가 주의 집중한 결과이다. h''^p 는 h^p 와 attn_{co}^p , attn_{self}^p 를 결합한 정보로 구절에 정답 정보까지 포함되어 정답과 연관된 중요한 정보를 파악한 결과를 의미한다. attn_{co}^a 은 내적 주의 집중 방법을 이용해 H^a 와 연관된 H^p 를 계산한 결과, attn_{self}^a 는 H^a 를 자가 주의 집중한 결과이다. h''^a 는 h^a 와 attn_{co}^a , attn_{self}^a 를 결합한 정보로 정답에 구절의 문맥 정보가 반영되어 정답의 유형 정보를 의미한다.

3.3 구조화 계층

구조화 계층은 h''^p 와 h''^a 를 양방향 GRU에 입력해 문맥 정보를 다시 반영한다. 이 과정에서 h''^p 의 정답 정보는 문맥을 반영하는 과정에서 구절에서 어떤 역할인지 다시 확인하게 되고, h''^a 의 구절의 문맥 정보는 정답이 어떤 유형인지 다시 확인하게 된다. 그리고 게이트를 이용해 양방향 GRU의 출력 결과와 입력 정보인 h''^p 와 h''^a 를 조절된 결과를 생성한다. 식 (4)를 통해 계산된다.

$$\begin{aligned}
 h'''_i^p &= \text{BiGRU}(h'''_{i+1}^p, h'''_{i-1}^p, h''_i^p) \\
 h'''_i^a &= \text{BiGRU}(h'''_{i+1}^a, h'''_{i-1}^a, h''_i^a) \\
 g(v) &= \text{sigmoid}(W_g v + b_g) \\
 h_i^{\text{passage}} &= g(h'''_i^p)h'''_i^p + (1 - g(h'''_i^p))h''_i^p \\
 h_i^{\text{answer}} &= g(h'''_i^a)h'''_i^a + (1 - g(h'''_i^a))h''_i^a
 \end{aligned} \quad (4)$$

식 (4)에서 h'''_i^p 와 h'''_i^a 는 h''_i^p 와 h''_i^a 를 양방향 GRU를 통해 인코딩한 정보이다. $g(v)$ 는 게이트 함수를 의미한다. h_i^{passage} 는 게이트를 통해 h'''_i^p 와 h''_i^p 의 반영 비율을 조절한 구절 i 번째 단어의 출력을 의미한다. h_i^{answer} 는 게이트를 통해 h'''_i^a 와 h''_i^a 의 반영 비율을 조절한 정답 i 번째 단어의 출력을 의미한다.

3.4 디코더

본 논문에서는 주의 집중 방법을 사용하는 시퀀스-투-시퀀스 모델 기반인 [6]의 포인터 제너레이터(Pointer Generator) 모델을 이용해 질문 생성을 진행하면서 상황에 따라 적절한 단어를 생성하거나 구절의 단어를 복사한다. 기존의 포인터 제너레이터는 인코더를 하나만 사용하지만, 본 논문에서는 구절과 정답을 개별적인 인코더로 구성해 2가지 인코더를 사용한다는 부분이 다르다. 2가지 인코더가 존재해 주의 집중 방법을 통한 문맥 벡터(Context Vector)도 2개가 존재해야 한다. 각 문맥 벡터는 [4]와 동일하게 계산되며 식 (5)의 방법과 같다.

$$\begin{aligned}
 pe_i^t &= v^t \tanh(W_{hp} h_i^{\text{passage}} + W_{sp} s_t + b_{\text{attn}_p}) \\
 ae_i^t &= v^t \tanh(W_{ha} h_i^{\text{answer}} + W_{sa} s_t + b_{\text{attn}_a}) \\
 pa^t &= \text{softmax}(pe^t) \\
 aa^t &= \text{softmax}(ae^t) \\
 c_t^{\text{passage}} &= \sum_{i=0}^n pa_i^t h_i^{\text{passage}} \\
 c_t^{\text{answer}} &= \sum_{i=0}^n aa_i^t h_i^{\text{answer}}
 \end{aligned} \quad (5)$$

식 (5)에서 s_t 는 디코더의 t 번째 상태(State)를 의미한다. pe_i^t 와 ae_i^t 는 s_t 와 h_i^{passage} , s_t 와 h_i^{answer} 의 연관성을 계산한 결과이다. pa^t 는 pe_i^t 를 이용하고 aa^t 는 ae_i^t 를 이용해 계산한 주의 집중 분포(Attention Distribution)를 의미한다. 구절 문맥 벡터인 c_t^{passage} 는 pa^t 와 h^{passage} 를 이용해 계산된다. 정답 문맥 벡터인 c_t^{answer} 는 aa^t 와 h^{answer} 를 이용해 계산된다. c_t^{passage} 는 질문 생성에 필요한 구절 정보를 포함하고 있고, c_t^{answer} 는 정답의 유형 정보를 나타내는 핵심어의 정보를 포함하고 있다. 핵심어의 정보에는 구절의 문맥까지 포함되어 더 정확한 정답의 유형을 유추하는데 기여한다. 두 문맥 벡터는 생성 확률 분포와 $p(\text{gen})$ 을 계산하는데 사용된다. $p(\text{gen})$ 은 단어를 생성할 때 생성 확률 분포를 얼마나 반영할지 결정하는 확률이다. 디코더는 $p(\text{gen})$ 과 생성 확률 분포, 그리고 pa^t 를 이용해 최종 분포를 계산 후 단어를 생성한다. 식

(6)의 방법으로 계산된다.

$$\begin{aligned}
 P_{\text{vocab}} &= \text{softmax}(V'(V[s_t, c_t^{\text{passage}}, c_t^{\text{answer}}] + b) \\
 &\quad + b') \\
 p(\text{gen}) &= \text{sigmoid}(W_p^T c_t^{\text{passage}} + W_{ca}^T c_t^{\text{answer}} \\
 &\quad + W_s^T s_t + W_x^T x_t + b_p) \\
 p(w) &= p(\text{gen})P_{\text{vocab}}(w) + (1 - p(\text{gen})) \sum_{i:w_i=w} p a_i^t
 \end{aligned} \tag{6}$$

식 (6)에서 P_{vocab} 는 s_t 와 $c_t^{\text{passage}}, c_t^{\text{answer}}$ 를 통해 계산한 생성 확률 분포를 의미한다. $p(\text{gen})$ 은 $c_t^{\text{passage}}, c_t^{\text{answer}}, s_t$ 그리고 x_t 를 이용해 계산한다. x_t 는 디코더에 t 번째 입력된 단어를 의미한다. P_{vocab} 와 $p(\text{gen})$ 을 계산할 때 사용된 c_t^{passage} 는 구절에서 단어 생성에 필요한 정보를 제공하고 c_t^{answer} 는 정답의 유형을 제공한다. P_{vocab} 는 두 정보를 같이 확인하여 정답 유형에 알맞은 정답을 생성할 수 있고 $p(\text{gen})$ 은 c_t^{passage} 에 반영된 정보를 복사와 생성 중 무엇이 적절한지 c_t^{answer} 를 통해 올바른 결정을 할 수 있게 된다. P_{vocab} 와 $p(\text{gen})$, a_i^t 를 이용해 단어 w 의 최종 확률을 계산한다.

4. 실험 및 결과

4.1 데이터 구성

본 논문의 실험을 위해 위키피디아, 나무 위키 문서를 이용해 사람이 제작한 질의 응답 말뭉치를 사용한다. 말뭉치에서 구절은 문단 단위로 사용한다.

표 3 말뭉치 세부 구성

말뭉치 내용	개수
학습 말뭉치	20,000
검증 말뭉치	1,658
평가 말뭉치	2,000
문단의 평균 형태소 수	111.2
질문의 평균 형태소 수	20.6
정답의 평균 형태소 수	3.7

표 3은 말뭉치 세부 구성 표를 의미한다. 전체 말뭉치 23,658개 중 20,000개를 학습 말뭉치로, 1,658개를 검증 말뭉치로, 2,000개를 평가 말뭉치로 사용해 실험을 진행했다. 문단, 질문, 정답의 평균 형태소의 수는 표 3의 내용과 같다.

4.2 실험 구성 및 평가 방법

본 논문에서 제안한 정답과 구절의 공동 주의 집중 계층의 효과를 입증하기 위해 3가지 모델을 비교 실험한다. Baseline은 포인터 제너레이터 모델에 구절과 정답을 분리한 모델, 두번째 모델은 Baseline에 본 논문에서 제안한 정답과 구절의 공동 주의 집중 방법을 사용한 Baseline + co-attn, 세번째 모델은 다중 투시 매칭의 자질까지 사용한 제안 모델이다. 3가지 모델은 모두

Facebook에서 공개한 300차원의 한국어 FastText[12] 벡터와 50차원의 형태소 품사를 무작위 초기화한 벡터를 결합해 사용한다. 손실 함수는 크로스 엔트로피(Cross Entropy)를 사용하고 학습률은 0.0005를 사용한다. 본 논문에서는 모델의 성능 비교를 위해 BLEU[13]와 ROUGE[14]를 사용한다. BLEU는 번역 작업을 수행하는 모델의 성능 평가를 위해 고안된 평가 방법으로 n-gram 정밀도를 의미한다. ROUGE는 문서 요약 작업을 수행하는 모델의 성능 평가를 위해 고안된 평가 방법으로 n-gram 재현율을 의미한다.

4.3 실험 결과

표 4 성능 측정 결과

모델	BLEU	ROUGE-L
Baseline	24.4	56.1
Baseline + co-attn	26.2	57.1
제안 모델	26.7	57.5

표 4는 성능 측정 결과를 나타낸다. Baseline 보다 구절과 정답의 정보 연결이 이뤄진 Baseline + co-attn이 BLEU, ROUGE 모두 우수한 결과를 보인다. 본 논문에서 제안하는 정답과 구절의 공동 주의 집중 방법이 구절과 정답 분리로 발생했던 정보 손실 문제를 해결하는데 기여해 성능이 크게 향상된 것을 확인할 수 있다. 그리고 제안 모델이 다중 투시 매칭 자질을 사용하지 않은 Baseline + co-attn 보다 BLEU와 ROUGE 점수에서 우수한 결과를 보인다. 다중 투시 매칭 자질이 정답과 구절의 공동 주의 집중 방법에 유의미한 자질인 것을 확인할 수 있다.

4.4 생성 결과 예제

4.4절에서는 본 논문에서 실험한 모델이 생성한 질문이 어떤 차이를 보이는지 예제를 통해 설명한다.

표 5 정답과 구절의 공동 주의 집중 계층 사용 전후 결과 비교

입력 및 모델	내용 및 생성 결과
문단	김범수 -title- -answer-에 제대한 후 5개월 만에 6집으로 활동했으며 ... (생략)
정답	2008년 3월
Baseline	김범수가 활동한 날은?
Baseline + co-attn	김범수의 제대는?
제안 모델	김범수는 언제 제대했는가?

표 5는 정답과 구절의 공동 주의 집중 계층의 사용 전후 결과를 보여준다. 입력된 문단에는 가수 김범수가 언제 제대했는지, 제대 후 무엇을 했는지 기록되어 있다.

정답은 제대한 낱자이며, 문단에서의 정답은 “-answer-” 기호로 치환되었다. 정답과 구절의 공동 주의 집중 계층을 사용하지 않은 Baseline은 정답의 낱자 정보만 보고 언제 활동했는지 질문했다. 하지만 정답과 구절의 공동 주의 집중 계층을 사용하는 Baseline + co-attn 과 제안 모델은 정답의 낱자에 문맥을 반영해 제대 낱자를 묻는 질문을 생성했다.

표 6 다중 투시 매칭 사용 전후 결과 비교

입력 및 모델	내용 및 생성 결과
문단	함태호 -title- 2010년 아들 -answer- 오뚜기 회장에게 회사의 경영권을 넘겨주고 ...(생략)
정답	함영준
Baseline	2010년 아들의 아들은?
Baseline + co-attn	오뚜기가 경영권을 넘겨준 곳은?
제안 모델	함태호의 아들은 누구인가?

표 6은 다중 투시 매칭 사용 전후 결과를 보여준다. 입력된 문단은 기업인 함태호가 경영권을 누구에게 넘겨주었는지 기록되어 있다. 정답은 경영권을 넘겨받은 아들 “함영준”이며, 문단에서의 정답은 “-answer-” 로 치환되었다. Baseline과 Baseline + co-attn 은 올바르게 못한 질문을 생성했다. 하지만 다중 투시 매칭 자질을 사용해 정답과 구절의 공동 주의 집중 방법의 기능을 향상시킨 제안 모델은 올바른 질문을 생성했다.

5. 결론

본 논문은 구절과 정답을 분리했을 때 발생하는 정보 손실 문제점을 해결하기 위해 정답과 구절의 공동 주의 집중 계층을 제안했다. 실험을 통해 정답과 구절의 공동 주의 집중 방법은 정보 손실 문제점을 해결하는데 기여하여 Baseline 보다 우수한 성능을 보였다. 그리고 다중 투시 매칭 자질은 정답과 구절의 공동 주의 집중 방법의 기능을 향상시키는 것을 확인했다.

참고문헌

[1] P. Rajpurkar, J. Zhang, K. Lopyrev, & P. Liang, “Squad: 100,000+ questions for machine comprehension of text”, *arXiv preprint arXiv:1606.05250*, 2016.

[2] P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, ... & M. Rosenberg, “MS MARCO: A human generated MACHine REading COMprehension dataset.”, *arXiv preprint arXiv:1611.09268*, 2016

[3] Y. Kim, H. Lee, J. Shin, & K. Jung, “Improving neural question generation using answer separation”, In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, pp. 6602-6609), 2019

[4] V. Kumar, K. Boorla, Y. Meena, G. Ramakrishnan, & Y. F. Li, “Automating reading comprehension by generating question and answer pairs”, In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 335-348). Springer, Cham, 2018

[5] D. Bahdanau, K. Cho, & Y. Bengio, “Neural machine translation by jointly learning to align and translate”, *arXiv preprint arXiv:1409.0473*, 2014

[6] J. Gu, Z. Lu, H. Li, & V. O. Li, “Incorporating copying mechanism in sequence-to-sequence learning”, *arXiv preprint arXiv:1603.06393*, 2016

[7] A. See, P. J. Liu, & C. D. Manning, “Get to the point: Summarization with pointer-generator networks”, *arXiv preprint arXiv:1704.04368*, 2017

[8] L. Song, Z. Wang, W. Hamza, Y. Zhang, & D. Gildea, “Leveraging context information for natural question generation”, In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)* (pp. 569-574), 2018

[9] Y. Zhao, X. Ni, Y. Ding, & Q. Ke, “Paragraph-level Neural Question Generation with Maxout Pointer and Gated Self-Attention Networks”, In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 3901-3910), 2018

[10] J. Chung, C. Gulcehre, K. Cho, & Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling.” *arXiv preprint arXiv:1412.3555*, 2014

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, ... & I. Polosukhin, “Attention is all you need”, In *Advances in neural information processing systems* (pp. 5998-6008), 2017

[12] P. Bojanowski, E. Grave, A. Joulin, & T. Mikolov, “Enriching word vectors with subword information”, *Transactions of the Association for Computational Linguistics*, 5, 135-146, 2017

[13] K. Papineni, S. Roukos, T. Ward, & W. J. Zhu, “BLEU: a method for automatic evaluation of machine translation”, In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 311-318). Association for Computational Linguistics, 2002

[14] C. Y. Lin, “Rouge: A package for automatic evaluation of summaries”, In *Text summarization branches out* (pp. 74-81), 2004