S^2-Net: SRU 기반 Self-matching Network를 이용한 한국어 기계 독해

박천음*○, 이창기*, 홍수린**, 황이규**, 유태준**, 김현기***
강원대학교*, 마인즈랩**, 한국전자통신연구원***
{parkce, leeck}@kangwon.ac.kr, {lynn, yghwang, joon}@mindslab.ai, hkk@etri.re.kr

S^2-Net: Korean Machine Reading Comprehension with SRU-based Self-matching Network

Cheoneum Park*, Changki Lee*, Sulyn Hong**, Yigyu Hwang**, Taejoon Yoo**, Hyunki Kim***
Kangwon National University*, Mindslab**, Electronics and Telecommunications Research Institute***

요 약

기계 독해(Machine reading comprehension)는 주어진 문맥을 이해하고, 질문에 적합한 답을 문맥 내에서 찾는 문제이다. Simple Recurrent Unit (SRU)은 Gated Recurrent Unit (GRU)등과 같이 neural gate를 이용하여 Recurrent Neural Network (RNN)에서 발생하는 vanishing gradient problem을 해결하고, gate 입력에서 이전 hidden state를 제거하여 GRU보다 속도를 향상시킨 모델이며, Self-matching Network는 R-Net 모델에서 사용된 것으로, 자기 자신의 RNN sequence에 대하여 어텐션 가중치 (attention weight)를 계산하여 비슷한 의미 문맥 정보를 볼 수 있기 때문에 상호참조해결과 유사한 효과를 볼 수 있다. 본 논문에서는 한국어 기계 독해 데이터 셋을 구축하고, 여러 층의 SRU를 이용한 Encoder에 Self-matching layer를 추가한 S^2-Net 모델을 제안한다. 실험 결과, 본 논문에서 제안한 S^2-Net 모델이 한국어 기계 독해 데이터 셋에서 EM 65.84%, F1 78.98%의 성능을 보였다.

주제어: 기계 독해, 질의응답, Simple Recurrent Unit, 셀프 매칭 네트워크, 한국어 기계 독해 데이터셋

1. 서론

기계 독해(Machine Reading Comprehension)는 기계가 주어진 문맥을 이해하는 능력을 말하며, 이를 질의응답(Question Answering)에 적용하여 질문에 올바른 정답을 문맥 내에서 찾을 수 있다. 예를 들어, 기계 독해 시스템은 "국내 건조기 시장 점유율 1위 누구야?"와 같은 질문에 대하여, 문맥 "2004년 건조기 시장에 ... 의류 건조기 중 LG전자는 점유율 77.4%로 1위를 차지했다."을 이해하고, 해당 문맥 내에서 정답 "LG전자"를 찾아 출력한다.

기계 독해는 스탠포드의 SQuAD, 페이스북의 bAbi, 마이크로소프트의 MS-MARCO 등[1, 2, 3]과 같은 데이터셋이 있으며, DrQA, fastQA, R-Net, AoA reader, Bi-Directional Flow (BiDAF), Match-LSTM 등[4-9]과 같은 end-to-end 딥 러닝 모델들이 주로 연구되고 있다. 이러한 딥 러닝 모델들은 주어진 문맥과 질문에 대한 매칭 및 인코딩을 수행하고, 어텐션 매커니즘(attention mechanism)[10]을 기반으로 한 포인터 네트워크 모델(Pointer Networks)[11]을 이용하여 질문과 유사한 정답의 경계 인덱스(즉, 정답의 시작과 끝 위치)를 출력한다.

포인터 네트워크는 RNN Encoder-decoder 모델을 확장한 것으로 주어진 입력 열에 대응되는 위치를 결과로 출력하는 모델로서, 주어진 문맥에서 정답의 경계 인덱스를 찾아 결과로 출력하는데 적합하다. Self-matching Network는 R-Net 모델에서 사용된 것으로, 자기 자신의 RNN sequence에 대하여 어텐션 가중치를 계산하여 비슷한 의미의 문맥 정보를 볼 수 있기 때문에 상호참조해결과 유사한 효과를 볼 수 있는 모델이다.

Simple Recurrent Unit (SRU)은 Gated Recurrent Unit (GRU)[12]이나 Long Short-Term Memory (LSTM)[13]와 같이 neural gate를 이용하여 RNN에서 발생하는 vanishing gradient problem을 해결한 모델이다. SRU는 gate 입력에서 이전 hidden state를 제거하여 GRU와 LSTM 보다 메모리 셀의 계산 과정을 간단하게 하고 병렬화와 CUDA level 최적화를 수행하여 Convolutional Neural Network (CNN)과 유사한 속도를 보이고, cuDNN-optimized LSTM보다 5-10배 빠른 속도를 보인다. 또한 highway network를 포함하고 있어서 여러 층의 레이어를 쌓는 경우, 성능향상을 보인다[14].

본 논문에서는 기계 독해 한국어 데이터 셋을 구축하고, 여러 층의 SRU를 이용한 문맥 Encoder에 Self-matching Network를 추가한 기계 독해 모델인 S^2-Net을 제안하며, 질문 문장에 해당하는 자질을 추가하여 성능 향상을 시도한다.

2. Simple Recurrent Unit

SRU는 GRU, LSTM과 같이 neural gate를 두어 RNN의 오류 역전파(back-propagation)를 수행할 때 발생하는 vanishing gradient problem을 해결하고 gate 입력에서 이전 hidden state를 제거하여 속도를 향상시킨 새로운 recurrent unit 모델이다. SRU는 input gate i_t와 forget gate f_t, reset gate r_t, highway network를 이용하며, 그 식은 아래와 같다.

$$\tilde{x_t} = Wx_t$$
$$i_t = (1 - f_t)$$
$$f_t = \sigma(W_f x_t + b_f)$$
$$r_t = \sigma(W_r x_t + b_r)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{x_t}$$
$$h_t = r_t \odot g(c_t) + (1 - r_t) \odot x_t$$

Input gate i_t는 $\tilde{x_t}$와 element-wise 곱을 수행하여 입력 정보 반영 여부를 결정하고, forget gate f_t는 c_{t-1}과 element-wise 곱

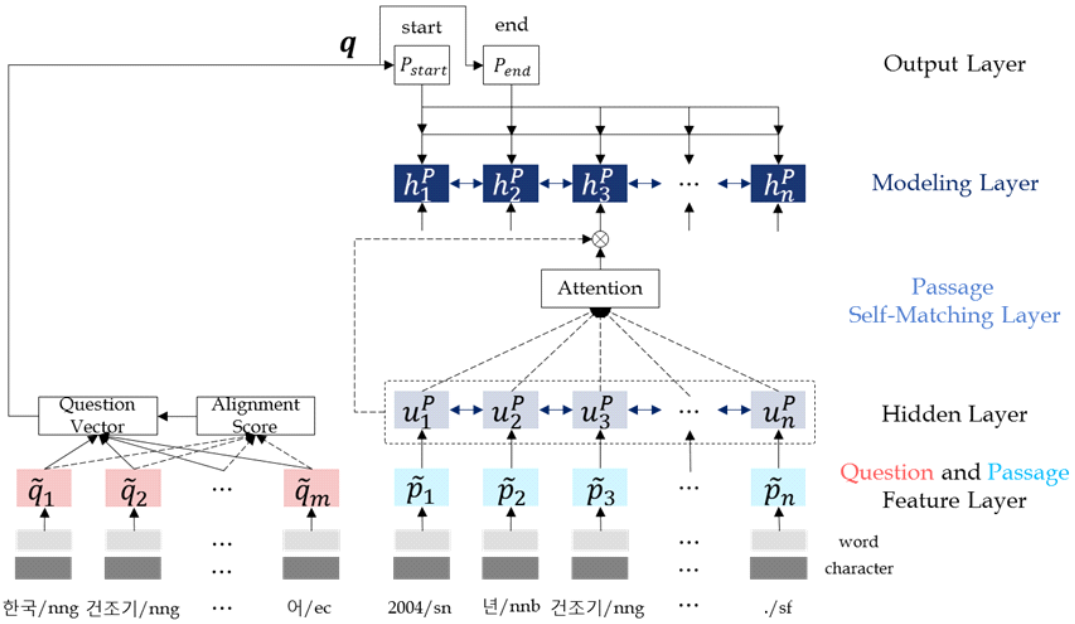

그림 2. SRU 기반 Self-matching Networks 구조

을 수행하여 이전 internal state 정보를 얼마나 반영할지를 결정한다. 여기서 $\tilde{x}_t$는 입력 x_t와 가중치 W를 곱하여 선형 변환된 결과이며, i_t는 $i_t = 1 - f_t$와 같고, f_t는 입력 x_t에 대하여 Feed-forward Neural Network (FFNN)을 수행하고 sigmoid를 적용한 결과이다. 이때 기존 RNN 모델(GRU, LSTM)들의 forget gate는 $f_t = \sigma(W_f x_t + R_f h_{t-1} + b_f)$와 같이 이전 hidden state Rh_{t-1}을 포함하였지만, SRU는 gate 계산에 FFNN을 적용하여 계산량을 줄이고, 병렬 계산을 가능하게 한다. c_t는 internal state로 입력 x_t와 이전 internal state c_{t-1}로부터의 정보 전달을 조절하고, 활성함수(activation function) $g(\cdot)$를 적용하여 internal state의 출력결과를 만든다. Hidden state h_t는 internal state 출력과 입력 x_t에 대한 highway network를 수행한 결과이다. 여기서 internal state 출력 $g(c_t)$는 reset gate r_t와 element-wise 곱을 수행하여 internal state 출력을 hidden state로 얼마나 반영할지 결정하고, 입력 x_t는 $(1 - r_t)$와 element-wise 곱으로 계산하여 입력 x_t의 반영 여부를 결정한다. [그림 1]은 SRU를 나타낸다.

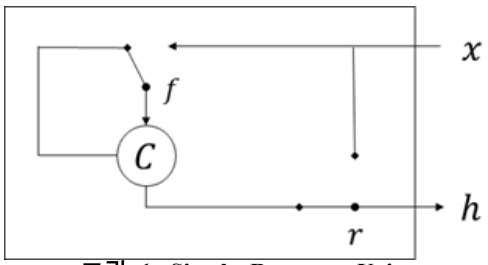

그림 1. Simple Recurrent Unit

3. SRU 기반 Self-matching Networks (S²-Net)를 이용한 한국어 기계 독해

기계 독해를 수행하기 위하여 각 모델들은 질문(Q), 문단(P), 정답(Y)의 데이터 셋이 주어진다. 질문은 m개의 단어 $Q=\{q_1, q_2, ..., q_m\}$로 구성되며, 문단은 n개의 단어 $P=\{p_1, p_2, ..., p_n\}$로 구성되고, 이를 인코딩하여 포인터 네트워크로 시작 경계 $y_1(P_{start})$, 마지막 경계 $y_2(P_{end})$를 출력한다.

본 논문에서는 한국어 기계 독해를 수행하기 위하여 SRU 기반 Self-matching 네트워크(S²-Net)를 이용하며, S²-Net 모델은 [그림 2]와 같다. S²-Net은 자질 레이어(Feature Layer)에서 문단과 질문에 대한 자질 임베딩(feature embedding)을 수행하고, 히든 레이어(Hidden Layer)에서 문단 인코딩(paragraph encoding)과 질문 인코딩(question encoding)을 수행한다. 셀프 매칭 레이어(Self-matching Layer)에서 문단 인코더 벡터(paragraph encoder vector)에 대한 셀프 어텐션을 적용하며, 모델링 레이어(Modeling Layer)에서 셀프 어텐션이 적용된 문단 인코더 벡터를 모델링하고, 출력 레이어(Output Layer)에서 정답에 대한 포인팅을 수행한다. 자질 레이어에 대한 수식은 다음과 같다.

$$\tilde{p}_t = [f_{emb}(p_t); f_{c\_emb}(p_t); f_{exact\_match}(p_t); f_{tf}(p_t); f_{align}(p_t)]$$
$$\tilde{q}_t = [f_{emb}(q_t); f_{c\_emb}(q_t); f_{exact\_match}(q_t); f_{tf}(q_t); f_{align}(q_t)]$$

$\tilde{p}_t$와 $\tilde{q}_t$는 입력된 자질 벡터이며, 이를 만들기 위하여 추출한 자질은 다음과 같다(질문인 경우 p_t대신 q_t가 입력으로 주어진다).

- 단어 표현(word embedding): $f_{emb}(p_t) = E(p_t)$
- 음절 표현(character embedding): $f_{c\_emb}(p_t) = CE(p_t)$
- 정확한 매치(exact match): $f_{exact\_match}(p_t) = \Pi(p_t \in q)$
- 토큰 자질(token feature): $f_{tf}(p_t) = TF(p_t)$
- 정렬된 질문 표현(aligned question embedding):

$$f_{align}(p_t) = \sum_j \alpha_{t,j} E(q_j),$$
$$\alpha_{t,j} = \frac{\exp(\alpha(E(p_t)) \cdot \alpha(E(q_j)))}{\sum_{j'} \exp(\alpha(E(p_t)) \cdot \alpha(E(q_{j'})))}$$

본 논문에서 단어 표현(word embedding)은 10만 단어에 대한 2년치 뉴스기사를 Neural Network Language Model (NNLM)[15]으로 학습한 것을 사용하며, 음절(또는 문자) 표현(character embedding)은 임의의 값으로 초기 값을 설정하고, CNN을 이용하여 단어에

대한 임베딩(embedding) 값을 학습한다. 정확한 매치(exact match) 자질은 문단 단어 p_t가 질문에 포함되는지 확인하는 자질(1 또는 0)이며, 문단과 질문의 각 단어는 "형태소/품사태그"로 구성된다. 토큰 자질(token feature)은 각 단어의 빈도($TF(p_t)$)를 정규화하여 자질로 사용한다. 정렬된 질문 표현(aligned question embedding)은 문단 표현과 질문 표현에 대한 얼라인먼트 스코어(alignment score)를 구하고, 문맥 인코더 벡터와 곱하여 매칭 문맥 벡터(matching context vector)를 계산하는 방법이다.

질문 자질벡터 $\tilde{q_t}$에 대하여 인코딩을 수행할 경우에는 질문 문장의 모든 hidden state를 하나의 벡터로 인코딩한다. 이때 질문 문장의 hidden state를 정규화하여 얼라인먼트 벡터 b를 만들고 이것을 질문 문장의 hidden state와 계산하여 질문 벡터(question vector) $\boldsymbol{q}$를 만든다. 질문 벡터 $\boldsymbol{q}$에 대한 수식은 다음과 같다.

$$\boldsymbol{q} = \sum_j b_j q_j$$

$$b_j = \exp(w \cdot q_j) / \sum_{j'} \exp(w \cdot q_{j'})$$

문단 인코딩을 수행하는 히든 레이어는 bidirectional SRU (BiSRU)로 구성되며, 수식은 아래와 같다. 여기서 u_t^P는 문단 입력 hidden state $\tilde{p}_t$에 대하여 인코딩된 hidden state이다.

$$u_t^P = BiSRU_P(u_{t-1}^P, \tilde{p}_t)$$

u_t^P는 모델링 레이어 h_t^P의 입력으로 사용되며, 셀프 매칭 레이어의 문맥 벡터 c_t와 연결($[u_t^P; c_t]^*$)되어 인코딩이 수행된다. $[u_t^P; c_t]^*$는 gated attention-based recurrent networks이며, sigmoid가 적용된 비선형 게이트 레이어 g_t와 $[u_t^P; c_t]$에 대하여 element-wise sum을 수행한 것이다. 셀프 매칭 레이어는 입력으로 주어진 열(sequence)을 대상(즉, 자기 자신)으로 얼라인먼트 스코어를 구하고 인코딩된 벡터들과 곱하여 문맥 벡터를 만드는 방법이며, 입력열에서 유사한 hidden state 간에 높은 얼라인먼트 스코어를 계산하고 인코딩 벡터들에 곱하여 어텐션 가중치를 조절한다. 모델링 레이어 h_t^P에 대한 수식은 아래와 같다.

$$h_t^P = BiSRU(h_{t-1}^P, [u_t^P; c_t]^*)$$

$$g_t = sigmoid(W_g [u_t^P; c_t])$$

$$[u_t^P; c_t]^* = g_t \odot [u_t^P; c_t]$$

$$s_j^t = v^T \tanh(W_u^P u_j^P + W_u^{\tilde{P}} u_t^P)$$

$$a_i^t = \exp(s_i^t) / \sum_{j=1}^{n} \exp(s_j^t)$$

$$c_t = \sum_{i=1}^{n} a_i^t u_i^P$$

S^2-Net은 포인터 네트워크의 포인팅 방법을 기반으로, 모델링 레이어에서 만들어진 h_t^P와 질문 벡터 $\boldsymbol{q}$를 bi-linear sequence attention으로 계산하여 질문에 적합한 정답의 위치를 문단에서 찾아 출력한다. 이때 출력 결과는 정답(answer span)의 시작(P_{start})과 끝(P_{end})의 위치이며, 이에 따른 수식은 다음과 같다.

$$P_{start}(t) \propto \exp(h_t^P W_s \boldsymbol{q})$$

$$P_{end}(t) \propto \exp(h_t^P W_e \boldsymbol{q})$$

본 논문에서는 정답의 시작과 끝을 출력할 때 최대 길이(max_len)를 50 형태소로 제한하였다.

4. 관련 연구

기계 독해를 해결하기 위한 기존 연구들에는 어텐션 메커니즘 기반 포인터 네트워크가 적용되며, DrQA, fastQA, R-Net, BiDAF 등의 모델이 있다. 본 논문에서 제안한 S^2-Net은 DrQA를 기반으로 음절 표현 자질과 질문 문장에 대한 자질, R-Net의 Self-matching layer를 추가하였고, 인코더에서 여러 층의 SRU를 사용하여 학습 속도 및 성능을 향상시켰다.

4-1. FastQA

FastQA는 임베딩 레이어, 인코더, 정답 레이어(Answer Layer)로 간단히 구성된다. 임베딩 레이어에서 입력 열에 단어 표현과 highway network를 적용하고, 각 단어들에 대한 정확한 매치와 정렬된 질문 표현 자질을 추출하여 모두 연결(concatenation)하여 사용한다. 본 논문에서는 히든 레이어에서 SRU를 기반으로 인코딩을 수행하는데, SRU는 highway network를 포함하고 있어, 추가적인 highway layer가 필요하지 않으며, 보다 많은 레이어를 쌓을 수 있다.

FastQA의 인코더에서는 bidirectional LSTM을 적용하여 인코딩을 수행하며, 질문과 문단의 weight matrix를 서로 공유하여 학습한다. 그 후, FFNN을 수행하는데, 여기서 사용되는 weight matrix B는 질문과 문단 벡터 각각 독립적으로 적용된다. 본 논문에서는 질문과 문단의 벡터를 인코딩할 때 서로 공유하지 않고, 정렬된 질문 표현 자질을 질문과 문단 모두 추출하여 질문과 문단 매칭을 수행하였다. 마지막으로 정답 레이어에서는 본 논문과 같이 질문 벡터 $\boldsymbol{q}$를 만들고, 문단 인코딩 벡터와 함께 ReLU기반 2-layer FFNN에 적용하여 정답의 시작과 끝을 출력한다. 본 논문에서는 ReLU기반 2-layer FFNN 레이어 없이 질문 벡터 $\boldsymbol{q}$와 문단의 모델링 벡터 h_t^P를 bi-linear sequence attention로 계산한다.

4-2. DrQA

DrQA는 웹에서 질문과 관련된 문서를 찾는 문서 검색(Document Retriever) 모듈과 찾은 문서들로부터 질문에 적합한 정답을 찾기 위하여 기계 독해를 수행하는 문서 리더(Document Reader) 모듈로 구성된다.

DrQA의 인코더는 bidirectional RNN으로 구성되며, GRU나 LSTM을 이용한다. 본 논문에서는 실험을 위하여 학습 속도가 더 빠르고 highway network를 포함하여 여러 층을 쌓을수록 성능이 증가하는 특성을 가진 SRU를 적용하였다. DrQA는 본 논문과 달리 Self-matching Layer를 포함하지 않고, 음절 표현을 사용하지 않았으며, 자질벡터 $\tilde{p_t}$에 대하여 단어 표현, 정확한 매치, 토큰 자질, 정렬된 질문 표현을 사용하였다. 여기서 토큰 자질은 $TF(p_t)$뿐만 아니라 품사태그 정보 $POS(p_t)$와 개체명 정보 $NER(p_t)$을 추가로 사용하였는데, 본 논문에서는 입력되는 단어가 "형태소/품사태그"이기 때문에 품사태그 정보를 보는 자질은 추가로 사용하지 않았다.

- 토큰 자질(token feature):

$$f_{token}(p_t) = (POS(p_t), NER(p_t), TF(p_t))$$

문단 인코딩 이후의 레이어는 Self-matching layer를 제외하고 본 논문의 방법과 같으며, 출력 결과를 계산하는 함수의 입력으로 본 논문의 모델링 레이어의 인코딩인 h_t^P 대신 문단 자질 임베딩 p_t가 주어진다.

$$P_{start}(t) \propto \exp(p_t W_s \boldsymbol{q})$$

$$P_{end}(t) \propto \exp(p_t W_e \boldsymbol{q})$$

4-3. R-Net

R-Net은 gated attention-based matching layer에서 질문과 문단을 매치시켜 질문의 의미를 포함한 문단 표현(passage representation)을 만들고, 해당 문단 표현에 대하여 셀프 매칭 어텐션 메커니즘(self-matching attention mechanism)을 기반으로

자기 자신에 대한 얼라인먼트 스코어를 계산하고 인코딩 된 hidden state와 곱하여 출력 결과를 구하는 딥 러닝 모델이다. R-Net의 경우에는 본 논문과 같이 단어 표현과 음절 표현을 사용하지만, 정렬된 질문 표현 자질 대신 질문-문단 매칭 레이어에서 어텐션 가중치를 계산하여 hidden state에 적용하고 모델링을 수행한다. 출력 레이어에서 포인터 네트워크로 정답 어텐션 가중치를 계산할 때 질문 인코딩에 대하여 셀프 어텐션으로 모델링하여 문단 인코딩과 함께 출력 결과에 대한 어텐션 스코어의 확률 분포를 구한다. R-Net은 모든 어텐션 메커니즘에 concat score를 적용하지만, 본 논문에서는 R-Net의 어텐션 스코어 방법과 달리 Bi-linear sequence attention 기반 포인터 네트워크를 이용하여 정답 경계의 위치를 출력한다.

4-4. Bi-Directional Attention Flow (BiDAF)

BiDAF는 6개의 레이어로 구성된 계층적 다단계 프로세스 모델이며, 양방향 어텐션 플로우 메커니즘(bi-directional attention flow mechanism)을 기반으로 한다. 양방향 어텐션 플로우는 Query2Context $\widetilde{H}$와 Context2Query $\widetilde{U}$를 말하며, 얼라인먼트 스코어를 계산할 때 Query2Context $\widetilde{H}$와 Context2Query $\widetilde{U}$를 문단 인코딩 H와 함께 계산하여 어텐션 가중치 G를 만든다. 그 후, 모델링 레이어에서 G를 입력으로 하여 bi-directional RNN을 수행하여 인코딩 M을 만든다.

$$G_{:t} = \beta(H_{:t}, \widetilde{U}_{:t}, \widetilde{H}_{:t})$$
$$M_{:t} = BiRNN(G_{:t})$$

위와 같이 계산 된 G와 인코딩된 M은 질문에 대한 정답을 출력하기 위하여 출력 레이어에서 서로 연결하고, Linear attention weight를 계산하여 정답의 시작(P_{start})과 끝(P_{end})을 구한다.

$$P_{start} = softmax(w^T_{(P_{start})}[G;M])$$
$$P_{end} = softmax(w^T_{(P_{end})}[G;M])$$

본 논문에서는 BiDAF의 양방향 어텐션 플로우 메커니즘과 달리, 정렬된 질문 표현 자질을 이용하여 문단 인코딩 벡터 u^P_t를 만들고, 이것을 입력으로 셀프 매칭 레이어에서 모델링을 수행하여 h^P_t를 만든 다음, 출력 레이어에서 질문 벡터와 함께 Bi-linear sequence attention을 수행한다.

5. 한국어 기계 독해 데이터셋

본 논문에서 제안한 S^2-Net을 이용한 한국어 기계 독해의 데이터 셋(MindsMRC Data Set)은 연예, 일반 도메인을 대상으로 뉴스와 위키피디아로부터 수집한 문단과 질문-정답 쌍으로 구성되며, [그림 3]과 같이 SQuAD 데이터 셋과 유사한 포맷을 따른다.

```
set -
    |-  version (str)
    -   data[]  -
                |-  title (str)
                -   paragraphs[]-
                                |-  context_original (str)
                                |-  dp[]-
                                |       |-  head (int)
                                |       |-  id (int)
                                |       |-  label (str)
                                |       |-  weight (double)
                                |       |-  text (str)
                                |       -   mods [(int)]
                                |
                                |-  qas[]-
                                |       |-  question_original (str)
                                |       |-  id (str)
                                |       |-  question_dp [[[(str)]]]
                                |       |-  question
                                |       |-  answer  -
                                |                   |-  text_original (str)
                                |                   |-  text [[[(str)]]]
                                |                   |-  answer_end int)
                                |                   -   answer_start (int)
                                |
                                -   context[[[(str)]]]
```

그림 3. 한국어 기계 독해 데이터 셋 구조

한국어 기계 독해 데이터 셋 구조는 JSON기반으로 구성되며, [그림 3]에서 set은 데이터 셋 전체를 포함하는 key이다. Version은 데이터 셋의 현재 버전을 의미하고 문자열로 표현되며, data는 현재 데이터 셋에 있는 모든 데이터를 포함하는 리스트이다. data의 title은 문서의 제목을 문자열로 나타내고, paragraphs는 문서 제목에 해당하는 문단과 질문, 정답 정보를 포함하는 리스트이다. context_original은 위키나 뉴스로부터 수집한 실제 텍스트이고, dp는 context_original에 해당하는 텍스트의 의존 구문 분석 결과를 포함하는 리스트이며, qas는 해당 문서의 질문과 정답 정보를 포함하는 리스트, context는 해당 텍스트의 형태소 정보를 포함하는 리스트이다. dp는 의존관계를 나타내는 head (중심어)와 mods (수식어), 해당 관계의 레이블 정보를 나타내는 label, 각 어절의 id, text, weight로 구성되며, weight는 의존 관계에 대한 가중치 결과이다. qas는 해당 문서의 질문과 정답 정보로 구성되며, qas에는 질문 실제 문자열인 question_original과 질문의 id, 질문의 의존 구문 분석 결과인 question_dp 리스트, 실제 질문에 대한 형태소 결과를 포함하는 question, 그에 따른 정답 정보를 포함하는 answer key가 있다. 질문에 해당하는 정답은 answer가 되며, answer는 정답의 실제 문자열인 text_origianal과 정답의 형태소 정보인 text 리스트, 정답이 문서 내에서 등장하는 시작과 끝의 인덱스 정보인 answer_start, answer_end로 구성된다. 한국어 기계 독해 데이터 셋 예제는 [표 1]과 같다.

표 1. 한국어 기계 독해 데이터 예제

Title		
장마철에도 빳빳하게... 물 만난 의류건조기		
Paragraphs		
Context_original	2004년 건조기 시장에 가장 먼저 뛰어든 LG전자를 비롯해 올해 초 삼성전자와 중견 기업까지 건조기 판매에 나서면서 국내 건조기 생산량은 급격히 늘고 있다. 건조기의 대당 판매가격을 고려했을 때 1~2년 내에 연간 시장 규모는 1조 원을 넘을 것으로 예상된다. 국내 건조기 시장은 LG전자가 주도하고 있다. 가격비교사이트 다나와리서치에 따르면 올 1월부터 6월까지 판매된 의류 건조기 중 LG전자는 점유율 77.4%로 1위를 차지했다. 가스식·전기식을 모두 판매하는 LG전자는 올해 초부터 전기식 건조기 사업에 주력하고 있다. 회사는 올해 용량과 사용 편의성을 업그레이드한 트롬 전기식 건조기 신제품 2종을 출시했다. 올해 제품에는 냉매를 순환시켜 발생한 열을 활용하는 '인버터 히트펌프' 기술을 적용했다.	
Context	[[['2004/sn', '년/nnb'], ['건조기/nng'], ['시장/nng', '에/jkb'], ['가장/mag'], ['먼저/mag'], ['뛰어들/vv', 'ㄴ/etm'], ['LG/sl', '전자/nng', '를/jko'], ['비롯하/vv', '어/ec'], ['올해/nng'], ['초/nnb'], ['삼성전자/nng', '와/jc'], ['중견/nng'], ['기업/nng', '까지/jx'], ['건조기/nng'], ['판매/nng', '에/jkb'], ['나서/vv', '면서/ec'], ['국내/nng'], ['건조기/nng'], ['생산량/nng', '은/jx'], ['급격히/mag'], ['늘/vv', '고/ec'], ['있/vx', '다/ef', './sf']], ...]]	
Question_original	한국 건조기 시장 점유율 1위 어딘지 알려줘	
Question	[[['한국/nng'], ['건조기/nng'], ['시장/nng'], ['점유율/nng'], ['1/sn', '위/nnb'], ['어디/np', '이/vcp', 'ㄴ지/ec'], ['알려주/vv', '어/ec']]]	
Answers	text_original	LG전자
	text	[[['LG/sl', '전자/nng']]]
	answer_start	95
	answer_end	97

[표 1]에서는 title, paragraphs, context_original, context, question_original, question, answers 정보에 대한 예를 보인다. Title은 문서(뉴스 또는 위키피디아)의 제목을 나타내며,

paragraphs는 문서의 본문과 질문-정답 쌍을 나타낸다. Context_original과 question_original은 문단과 질문의 raw text이며, S^2-Net으로 학습 및 예측하기 위하여 context와 question과 같이 형태소분석 수행 결과를 만든다. Answers는 정답의 raw text (text_original)와 형태소분석 결과(text)를 포함하고 있으며, 문단에서의 정답 텍스트 시작 위치(answer_start)와 마지막 위치(answer_end)로 구성된다.

6. 실험

본 논문에서 제안하는 S^2-Net와 실험에 사용한 모든 모델은 PyTorch로 구현하였으며, 실험은 Intel i7-4790 CPU (3.60GHz), 32GB RAM, TITAN X (Pascal), Ubuntu 16.04 OS에서 수행되었다.

실험에 사용된 데이터 셋은 뉴스 도메인 36,931 문서, 93,835 질문과 위키 도메인 7,119 문서, 17,948 질문으로 구성되며, [표 2]와 같이 학습 셋과 개발 셋을 9:1의 비율로 나누었다. 본 논문에서 질문 데이터를 제작할 때 유사한 질문들을 2개씩 짝지어 만들었으며, 고유한 질문의 수는 [표 2]의 고유 질문 수와 같다.

표 2. 실험에 사용한 데이터 셋 개수

데이터 셋			
	문단 수	질문 수	고유 질문
개발 (dev)	5,188	13,066	6,533
학습 (train)	39,645	100,395	50,198

본 논문에서는 S^2-Net을 이용한 한국어 기계 독해에 대하여 다음과 같이 실험을 하였다. 학습은 *Adam*[16]을 이용하고, 학습율(learning rate)을 0.1로 설정하였다. 히든 레이어와 어텐션 레이어에 대한 활성함수는 모두 tanh를 적용하였으며, 모든 RNN 레이어는 SRU (CUDA level optimization)를 이용하였다. 드랍아웃은 0.2로 고정하고, 음절 표현의 차원 수는 50 그리고 단어 표현의 차원수는 100, 히든 레이어의 차원 수는 128로 설정하였다. 음절 표현은 윈도우 사이즈 [2,3,4,5,6]의 필터(filter)를 사용하고, 필터의 크기는 30으로 설정하였다. 미니 배치의 배치 크기는 32로 설정하였으며, 매 epoch마다 개발 셋으로 성능 평가를 수행하였다. 성능 측정의 척도는 EM (Exact Match)과 F1을 사용하였다[1].

[표 3]은 본 논문에서 제안한 S^2-Net과 DrQA[4], DrQA+BiSRU [14], BiDAF, BiDAF+SM (BiDAF+Self-matching)의 성능을 나타낸다. 실험이 수행된 모델 중에서 DrQA가 실험의 baseline으로 RNN type이 LSTM이며, 나머지 모델은 모두 SRU를 적용하였다. Encoder RNN의 레이어는 3과 5로, 모델링 레이어는 1과 2로 각각 설정하였고, 그 외의 하이퍼 파라미터는 모두 동일하게 적용하였다.

표 3. SRU기반 한국어 기계 독해 모델 별 성능 (dev, %)

Model	Layers	Modeling layer	RNN type	EM	F1
DrQA(baseline)[4]	3	1	LSTM	59.25	74.37
DrQA+BiSRU[14]			SRU	64.16	77.55
BiDAF				64.01	77.29
BiDAF+SM				63.97	77.71
S^2-Net (our)				**64.37**	**78.16**
DrQA+BiSRU[14]	5	2	SRU	64.94	78.16
BiDAF				64.76	78.03
BiDAF+SM				61.78	75.67
S^2-Net (our)				**65.84**	**78.98**

실험 결과, Encoder RNN 레이어 3, 모델링 레이어 1 일 때 실험의 baseline인 DrQA는 F1 74.37%의 성능을 보였지만, DrQA에 SRU를 적용한 DrQA+BiSRU는 F1이 78.16%로 DrQA에 비하여 3.79% 향상된 성능을 보였다. 그 외로 BiDAF나 BiDAF+SM은 DrQA와 같은 실험 설정일 때 각각 F1 77.29%, F1 77.71%의 성능을 보였으며, 이에 따라 RNN type에 LSTM을 적용할 때보다 SRU를 적용할 때 성능이 향상됨을 알 수 있다. 또한 본 논문에서 제안한 S^2-Net이 EM 64.37%, F1 78.16%로 다른 모델들에 비하여 좋은 성능을 보였다.

Encoder RNN 레이어 5, 모델링 레이어 2일 때 DrQA+BiSRU는 F1 78.16%, BiDAF는 F1 78.03%로 Encoder RNN 레이어 3, 모델링 레이어 1일 때보다 각각 0.61%, 0.74% 향상된 성능을 보였지만, BiDAF+SM은 F1 75.67%로 레이어를 더 쌓으니 −2.04% 더 낮은 성능을 보였다. 본 논문에서 제안한 S^2-Net은 F1 78.98% (EM 65.84%)로 가장 좋은 성능을 보였으며, 같은 실험 환경(5-layer, 2-layer)에서 DrQA+BiSRU보다 0.82%, BiDAF보다 0.95%, BiDAF+SM보다 3.31% 더 좋은 성능을 보였고, DrQA (baseline)보다 4.61% 더 좋은 성능을 보였다.

7. 결론

본 논문에서는 SRU 기반 Self-Matching Network (S^2-Net)를 이용한 한국어 기계 독해 모델을 제안하였고, 한국어 기계 독해 데이터 셋과 그 구축 방법에 대하여 설명하였으며, S^2-Net과 DrQA, DrQA +BiSRU, BiDAF, BiDAF+SM에 대한 비교 실험을 수행하였다.

실험 결과, 한국어 기계 독해 데이터 셋에 대하여 본 논문에서 제안한 방법인 S^2-Net이 Encoder RNN 레이어 5, 모델링 레이어 2일 때 EM 65.84%, F1 78.98%로 가장 좋은 성능을 보였다. 같은 실험 환경에서 DrQA+BiSRU는 EM 64.94%, F1 78.16%, BiDAF는 EM 64.76%, F1 78.03%, BiDAF+SM이 EM 61.78%, F1 75.67%의 성능을 보였다.

향후 연구로는 기계 독해에 대한 학습 데이터를 더 구축할 것이며, hierarchical RNN 등과 같은 모델을 적용할 예정이다.

감사의 글

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 정보통신·방송 연구개발 사업의 일환으로 하였음. [2013-0-00131, (엑소브레인-1세부) 휴먼 지식증강 서비스를 위한 지능진화형 WiseQA 플랫폼 기술 개발]

참고문헌

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

[2] F. Hill, A. Bordes, S. Chopra and J. Weston. The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations. *arXiv preprint arXiv:1511.02301*, 2015.

[3] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary R. Majumder and L. Deng. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset, *arXiv preprint arXiv:1611.09268*, 2016.

[4] D. Chen, A. Fisch, J. Weston and A. Bordes. Reading Wikipedia to Answer Open-Domain Questions, *arXiv preprint arXiv:1704.00051*, 2017.

[5] D. Weissenborn, G. Wiese and L. Seiffe. Making Neural QA as Simple as Possible but not Simpler, *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. 2017.

[6] W. Wang, N. Yang, F. Wei, B. Chang and M. Zhou. Gated Self-Matching Networks for Reading Comprehension and Question

Answering, *In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* pp. 189-198, 2017.

[7] Y. Cui, Z. Chen, S. Wei, S. Wang, T. Liu and G. Hu. Attention-over-Attention Neural Networks for Reading Comprehension, *arXiv preprint arXiv:1607.04423*, 2016.

[8] M. Seo, A. Kembhavi, A. Farhadi and H. Hajishirzi. Bidirectional Attention Flow for Machine Comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

[9] S. Wang and J. Jiang. Machine Comprehension Using Match-LSTM and Answer Pointer, *arXiv preprint arXiv:1608.07905*, 2016.

[10] D. Bahdanau, et al. Neural machine translation by jointly learning to align and translate. *Proc. of ICLR' 15*, arXiv:1409.0473, 2015.

[11] O. Vinyals, M. Fortunato and N. Jaitly. Pointer Networks. *Advances in Neural Information Processing Systems*, pp. 2674-2682, 2015.

[12] K. Cho, et al. Learning phrase representation using RNN encoder-decoder for statistical machine translation. *Proc. of EMNLP' 14*, 2014.

[13] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Nueral computation, 9(8)*, pp.1735-1780, 1997.

[14] T. Lei and Y. Zhang. Training RNNs as Fast as CNNs. *arXiv preprint arXiv:1709.02755*, 2017.

[15] 이창기, 김준석, 김정희. 딥 러닝을 이용한 한국어 의존 구문 분석. *제 26회 한글 및 한국어 정보처리 학술대회*, pp. 87-91, 2014.

[16] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.