Q-Net : 질문 유형을 추가한 기계 독해

김정무○, 신창욱, 차정원
창원대학교
{gersanga, papower1, jcha}@changwon.ac.kr

Q-Net : Machine Reading Comprehension adding Question Type

Jeong-Moo Kim○, Chang-Uk Shin, Jeong-Won Cha
Changwon National University

요 약

기계 독해는 기계가 주어진 본문을 이해하고 질문에 대한 정답을 본문 내에서 찾아내는 문제이다. 본 논문은 질문 유형을 추가하여 정답 선택에 도움을 주도록 설계하였다. 우리는 Person, Location, Date, Number, Why, How, What, Others와 같이 8개의 질문 유형을 나누고 이들이 본문의 중요 자질들과 Attention이 일어나도록 설계하였다. 제안 방법의 평가를 위해 SQuAD의 한국어 번역 데이터와 한국어 Wikipedia로 구축한 K-QuAD 데이터 셋으로 실험을 진행하였다. 제안한 모델의 실험 결과 부분 일치를 인정하여, EM 84.650%, F1 86.208%로 K-QuAD 제안 논문 실험인 BiDAF 모델보다 더 나은 성능을 얻었다.

주제어: 기계 독해, 한국어, 형태소, 개체명, 질문 유형

1. 서론

텍스트를 이해하고 질문에 답변하는 능력은 자연어처리에서 매우 중요하다. 다양한 대용량 데이터 집합의 출현으로 기계 독해 분야의 발전이 이루어졌다[1-4].

하지만 본문의 내용과 질의문의 내용을 직접 Attention하는 방법은 좋은 방법이 아니다. 왜냐면 질문에는 본문에 나타나는 단어가 직접 나타나지 않는 경우가 많기 때문이다. 따라서 본 논문에서는 질문에 대해 질문 유형을 분류하고 이를 본문과 Attention을 하는 방법을 선택했다.

1.2 K-QuAD

K-QuAD 데이터 셋[5]은 연세대학교에서 제안한 SQuAD[6] 형식의 한국어 데이터 셋이다. K-QuAD 데이터 셋은 SQuAD의 번역 데이터와 한국어 Wikipedia로 구성된 데이터로 구성되어 있다. 번역 데이터는 SQuAD Version1을 한국어로 번역한 데이터이다. SQuAD Version1의 데이터는 하나의 본문에 여러 개의 질문-답으로 구성된 데이터 셋이다. 질문에 대한 답은 본문 내에서 찾을 수 있으며, SQuAD의 경우 그 답은 본문의 특정한 부분으로 구성돼있다. 실험에 사용된 SQuAD를 번역한 데이터의 경우, 오역된 데이터는 제거하여 실험한다. 오역을 판단하는 기준으로는 코퍼스 내 '지지지지', 'K K K' 등 특정 단어가 반복된 것이다. 오역으로 판단해 제거한 데이터는 총 731 개다.

2. Q-Net

제안하는 모델은 그림 1과 같다. 모델은 Encoding Layer와 Matching Layer, Prediction Layer로 구성된다. Encoding Layer는 형태소와 음절이 입력되어 Bi-directional Recurrent Neural Network(BiRNN)를 수행한다. Matching Layer는 Encoding Layer의 BiRNN을 통해 얻은 질문 벡터를 Softmax 하여 얻은 질문 유형 벡터로 본문의 벡터와 각각 Attention 한다. 또한, Self Attention을 이용하여 유용한 자질을 선별한다. Prediction Layer는 Pointer Network[7]를 통해 정답의 위치를 추출한다.

2.1 형태소, 개체명, 질문 유형 Encoding

Encoding Layer에서는 기존의 어절 단위의 단어가 아닌 형태소 단위와 음절 단위로 Encoding 한다.

본 논문에서는 질문(Q)과 문단(P) 각각에 형태소 단위 $Q = \{m_t^Q\}_{t=1}^m$, $P = \{m_t^P\}_{t=1}^n$와 음절 단위의 $Q = \{c_t^Q\}_{t=1}^m$, $P = \{c_t^P\}_{t=1}^n$ 학습을 통하여 더 정확한 답을 예측할 수 있도록 하였다. 한글 형태소의 분리에는 KoNLPy[8]의 Twitter Module을 사용하였다. Twitter Module은 선어말어미의 축약이나 전성어미의 분리를 고려하지 않기 때문에 완전한 한국어 구문분석에는 차이가 있다고 할 수 있다. 하지만 이러한 특성으로 형태소를 분리하면서 본문의 음절 수를 유지할 수 있다.

질문이 '아스널은 창단 후 몇 년간 챔피언스리그에서 우승하지 못하였나요'와 같다면 질문 유형 분류를 위해 질문의 단어 중 '몇 년간'과 같은 단어들을 대상으로 규칙을 이용하여 질문의 유형을 분류해 질문에 추가적인 Encoding을 하였다. 실험 2와 실험 5에서 Question Type(QT)에 해당되는 부분이다. 질문 유형의 종류로는 Person, Location, Date, Number, Why, How, What과 그 외 Others를 포함하여 총 8개의 유형이 있다. Others의 경우에는 yes/no를 유도하는 질문이지만 K-QuAD 데이터 셋에서는 해당 질문에 대한 답 역시 본문의 특정 단어로

구성되어 있다.

또한, 본문의 경우에 정답의 단어는 대부분이 본문의 주요 단어들이 포함되어 Espresso[9]를 통해 개체명 정보를 추가했다. Espresso의 결과로는 Person, Time, Location, Date 등 총 15개의 Class $P = \{ne_t^P\}_{t=1}^{n}$로 분류된다.

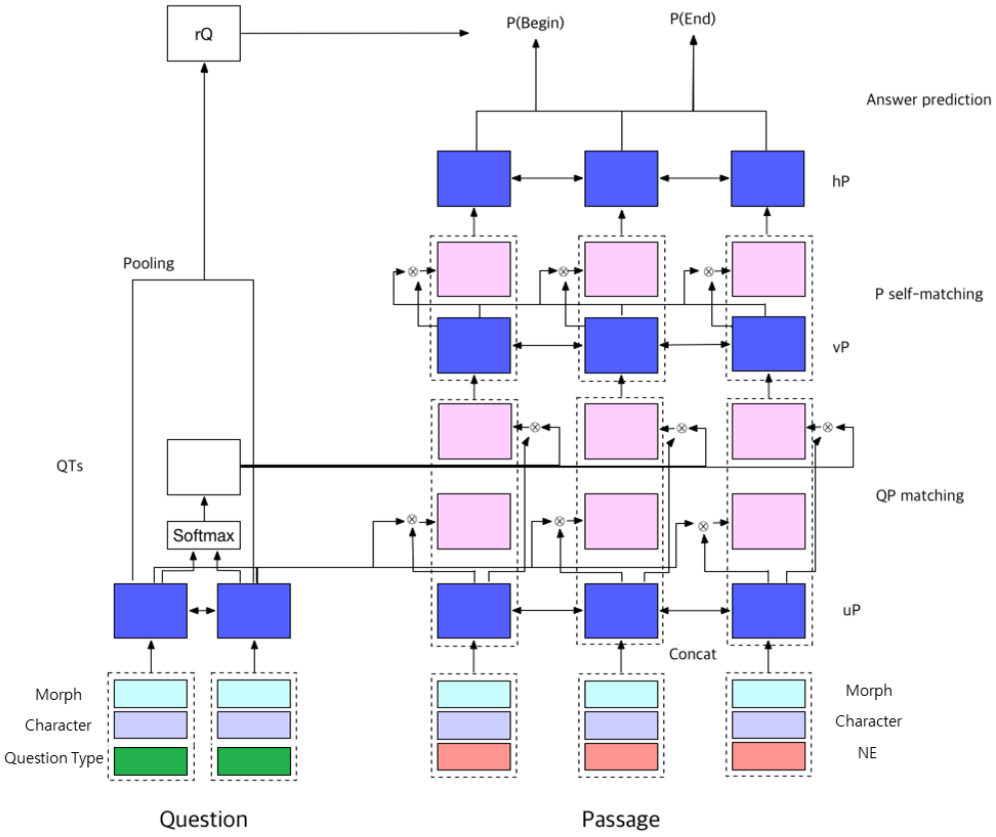

그림 1. 제안 시스템의 구조

각각의 정보로 BiRNN을 수행하여 얻은 질문 Encoding u_t^Q와 본문 Encoding u_t^P의 수식은 아래와 같이 생성한다.

$$u_t^Q = \mathrm{BiRNN}_Q(u_{t-1}^Q, [m_t^Q, c_t^Q, qt_t^Q]) \quad (1)$$

$$u_t^P = \mathrm{BiRNN}_P(u_{t-1}^P, [m_t^P, c_t^P, ne_t^P]) \quad (2)$$

3.2 질문 유형을 이용한 Attention

Attention을 위한 질문 유형 분류는 2가지 방법을 사용하였다. 하나는 질문 Encoding과 동일 방법으로 질문이 '아스널은 창단 후 몇 년간 챔피언스리그에서 우승하지 못하였나요' 와 같은 형식이 된다면, 질문의 단어 중 '몇 년간' 을 찾아 규칙으로 질문 유형을 분류하였다.

다른 하나는 u_t^Q를 Softmax한 값을 8개의 질문 유형으로 이루어진 vector($\{tv_t^x\}_{t=1}^{8}$)와 가중치 합(Weighted-Sum)을 수행하는 것이다.

$$w_j = \mathrm{softmax}(u_t^Q) \quad (3)$$

$$S_t^Q = \sum_{j=1}^{n} w_j tv_{tj} \quad (4)$$

Softmax를 해서 얻은 s_t^Q와 기존의 u_t^Q 각각을 u_t^P와 Attention을 진행한다. 각각은 Question Type & Passage Attention(QTA), Question & Passage Attention(QP)이라고 명칭 한다. QP, QTA, u_t^P 세 값을 이용하여 Self Attention의 입력(v^p)을 만든다.

$$v_t^P = \mathrm{BiRNN}(v_{t-1}^P, [u_t^P, c_t, s_t]^*) \quad (5)$$

$$g_t = \mathrm{sigmoid}(W_g^s [u_t^s, c_t, s_t]) \quad (6)$$

$$[u_t^s, c_t, s_t]^* = g_t \odot [u_t^s, c_t, s_t] \quad (7)$$

$$a_i^t = \exp(u_t^p W_a^P u_i^Q) / \sum_{j=1}^{m} \exp(u_t^P W_a^P u_j^Q) \quad (8)$$

$$c_t = \sum_{i=1}^{n} a_i^t u_i^Q \quad (9)$$

$$b_i^t = \exp(u_t^P W_a^P t_i^Q) / \sum_{j=1}^{M} \exp(u_t^P W_a^P t_j^Q) \quad (10)$$

$$s_t = \sum_{i=1}^{n} a_i^t u_i^Q \quad (11)$$

3.3 Self Attention

본 논문에서 제안하는 구조에서는 OP Attention과 QTA Attention, u_t^P가 Self Attention의 입력이 된다. Self Attention을 통해서 전체 입력 중 유용한 자질을 선별하여 그 결과를 Prediction Layer에 입력한다.

$$h_t^P = \mathrm{BiRNN}(h_{t-1}^P, [v_t^P, c_t]^*) \quad (12)$$

$$a_i^k = \exp(v_k^P W_w^P v_i^P) / \sum_{j=1}^{n} (v_k^P W_w^P v_j^P) \quad (13)$$

$$c_t = \sum_{i=1}^{N} a_i^n v_i^P \quad (14)$$

3.4 Attention이 반영된 정답 추출

정답 추출에는 Pointer Network를 이용하여 본문 내에 존재하는 정답의 시작과 끝 위치를 찾아낸다.

$$s_j^t = V^T \tanh(W_h^P h_j^P + W_h^a h_{t-1}^a) \quad (15)$$

$$a_i^t = \exp(s_i^t) / \sum_{j=1}^{n} \exp(s_j^t) \quad (16)$$

$$P_t = \arg max(a_1^t, ..., a_n^t) \quad (17)$$

P_t는 Pointer Network의 결과로 P_0은 정답의 시작 위치, P_1은 정답의 끝 위치를 나타낸다.

4. 실험

4.1. 실험 설정

표 1. 데이터 셋 개수

데이터 셋			
	Train	Dev	Test
SQuAD 번역	57k	19k	2k
한글 Wikipedia	2k	-	2k

학습에 사용된 데이터 셋은 표 1과 같다. 학습 데이터 셋의 개수는 데이터 셋 제안 논문의 실험과 같은 개수로 설정하였다. SQuAD 번역 데이터의 75%와 한글 Wikipedia 데이터의 50%를 사용하였다. 개발 데이터 셋에는 SQuAD 번역 데이터의 나머지 25%를 사용하였으며 테스트 데이터 셋에는 한글 Wikipedia 데이터의 나머지 50%를 사용하였다.

실험에서 hyper-parameter는 다음과 같이 설정하였다. 본문의 길이는 400, 질문의 길이는 50, 각각의 음절의 수는 16으로 제한하였다. batch size는 64, RNN의 hidden size는 75이다. optimizer로는 AdaDelta를 0.5로 사용하였다.

4.2. 실험 결과

본 논문에서는 SQuAD의 평가 방법에 따라 Exact Match와 F1-Score로 성능을 측정한다. 본 논문의 Baseline은 데이터 셋 제안 논문의 실험인 BiDAF의 모델과 R-Net을 사용한 실험이다. 실험 결과는 표 2와 같다. 질문 유형을 분류하는 본 논문에서 제안하는 구조에 따라 모델의 성능과 질문 유형별 성능을 분류하여 측정하였다. 선행 실험인 BiDAF 모델보다 EM 33.93% F1 14.708% 향상된 성능을 보였다.

오류를 분석해보면 정답에 해당하는 부분이 '16~17만 명이나' 일 때 추측된 결과는 '16~17만 명' 으로 된 부분이 많이 발견되었다. 그러나 SQuAD의 성능 측정 방법인 Exact Match과 F1으로 성능을 측정하는 경우에 모델이 위와 같은 예측을 하는 경우 조사나 어미 등의 이유로 올바른 성능이 나오지 않음을 볼 수 있었다. 따라서 올바른 답이 완전히 포함된 경우에 맞는 것으로 간주하는 변경된 평가를 적용하여 추가 평가를 하였다. 표 2의 6번 결과를 보면 성능향상이 많이 된 것을 발견할 수 있다.

5. 관련 연구

현재 기계 독해 연구는 크게 Encoding, Attention, Prediction으로 나눌 수 있다. Prediction에서는 정답의 시작과 끝 위치를 추출하는 Pointer Network를 고정적으로 사용한다. Encoding과 Attention 단계에서는 SQuAD 데이터 셋의 실험인 GF-Net[10]에서 제안하는 ELMo vector와 Feature vector를 이용하는 방법, MindsMRC 데이터 셋의 실험인 S^3-Net[11]의 문장 단위 Encoding을 추가하여 추가적인 Attention을 수행하는 방법 등이 있다.

6. 결론

본 논문의 실험은 질문과 본문 각각에 질문 유형과 개체명을 형태소와 음절에 추가하여 Encoding을 진행하였다. 또한, 질문 유형을 Softmax를 통하여 계산한 결과를 추가적인 Attention을 통해 질문 유형에 따른 가중치를 부여하고자 하였다.

그 결과, 질문의 유형과 정답의 개체명이 정확하게 분류 가능한 PER, LOC, DAT, NUM의 경우에는 전체적으로 성능향상을 볼 수 있었다. 하지만 그 외 질문에서는 정답에 개체명의 분석이 쉽지 않았고 그에 따라 좋지 않은 성능을 기록하게 되었다. 정답이 서술형으로 나타나는 Others와 How, Why와 같은 질문에서는 개체명으로 분석되지 않는 단어들의 수가 증가한다. 분석되지 않은 단어의 수가 증가한 서술형의 답일수록 개체명의 분석이 오히려 좋지 않은 성능을 나타냄을 확인할 수 있었다.

특히, 표 2의 4번의 실험을 통해 개체명을 분석하지 않은 모델이 단답형의 정답을 예측하는 것은 개체명을 분석한 모델보다 좋지 않은 성능을 보였다. 하지만 개체명이 분석되지 않은 단어들이 증가하는 서술형의 정답에

표 2. 최종 성능.

실험 1은 R-net을 이용한 실험 결과이다. 실험 2는 질의에 질문 유형 Encoding(QT)을 추가하고 Attention을 위해서도 규칙 Encoding(QTP Attention)을 적용한 실험이다. 실험 6은 실험 5에서 정답이나 추정결과가 완전히 포함될 경우 올바른 결과로 간주했을 경우의 성능이다.

	model	특징	EM F1	PER 315	LOC 225	DAT 297	NUM 220	HOW 99	WHY 60	WHAT 777	Others 19
0	Bi-DAF	데이터 셋 제안 논문 실험	50.720 71.500	-	-	-	-	-	-	-	-
1	baseline	형태소 단위 학습	53.100 74.461	52.063 71.219	56.000 74.460	70.370 84.569	55.288 77.597	38.383 67.125	46.667 74.023	48.778 72.975	15.789 36.267
2	QT + NE	Q : 음절 + 형태소 + QT P : 음절 + 형태소 + NE QTP Attention	52.250 74.117	53.528 72.757	56.444 74.735	74.735 85.297	53.711 76.935	36.538 71.567	41.964 69.018	47.435 71.366	10.526 38.781
3	NE Softmax	Q : 음절 + 형태소 P : 음절 + 형태소 + NE QTP Attention(가중치 합)	49.850 70.801	52.063 67.956	54.222 73.119	71.717 82.615	47.569 68.157	38.384 71.776	43.333 69.016	42.600 68.157	21.053 33.903
4	Softmax	Q : 음절 + 형태소 P : 음절 + 형태소 QTP Attention(가중치 합)	50.250 72.001	48.889 66.288	53.778 73.158	70.034 83.092	51.442 74.602	46.434 73.169	40.000 69.475	44.015 69.672	31.579 48.290
5	QT + NE Softmax	Q : 음절 + 형태소 + QT P : 음절 + 형태소 + NE QTP Attention(가중치 합)	**53.300** **75.035**	56.190 73.035	57.333 76.769	73.400 85.784	58.654 78.545	31.313 69.886	36.667 74.459	46.718 71.618	21.052 43.634
6	QT + NE Softmax 부 분 일 치 인정	Q : 음절 + 형태소 + QT P : 음절 + 형태소 + NE QTP Attention(가중치 합)	**84.650** **86.208**	**83.175** **83.708**	**87.111** **87.796**	**93.603** **93.771**	**87.981** **89.022**	**78.788** **82.969**	**78.333** **85.214**	**81.853** **84.002**	**68.421** **70.040**

는 더 정확한 답을 예측함을 볼 수 있었다.

이를 통해 개체명의 유무를 통해 모델의 판단하에 가중치를 조절한다면 더 유용한 자질을 분류할 수 있을 것으로 판단된다. 향후 연구로 모델이 학습하여 개체명의 가중치를 조절하며, 추가적인 한국어 임베딩을 통하여 모델을 개선할 예정이다.

감사의 글

본 연구는 2018년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2017R1D1A1B03033534)

참고문헌

[1] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In Proceedings of NIPS.

[2] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. arXiv preprint arXiv:1611.09268.

[3] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In Proceedings of ACL.

[4] Tomas Kocisky, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gabor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. Transactions of ACL, 6:317-328.

[5] Kyungjae Lee, et al. Semi-supervised Training Data Generation for Multiligual Question Answerring. LREC 2018, pp.2758-2762

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

[7] F. Hill, et al. The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations. arXiv preprint arXiv:1511.02301, 2015.

[8] 박은정, et al. KoNLPy: 쉽고 간결한 한국어 정보처리 파이썬 패키지. 제 26회 한글 및한국어 정보처리 학술대회, 2014

[9] Espresso http://air.changwon.ac.kr/~airdemo/Espresso/ (downloaded 2018, Aug. 16)

[10] 이현구, et al. GF-Net : 자질 선별을 통한 고성능 기계독해. 한국컴퓨터종합학술대회, pp.598-600, 2018

[11] 박천음, et al. S^3-Net: SRU 기반 문장 및 셀프 매칭 네트워크를 이용한 한국어 기계독해. 한국소프트웨어종합학술대회, pp.649-651, 2017