TabQA : 표 양식의 데이터에 대한 질의응답 모델

박소윤○, 임승영, 김명지, 이주열

LG CNS, 정보기술연구소
{soyoon.park, seungyoung.lim, kmj0614, jooyoul.lee}@lgcns.com

TabQA : Question Answering Model for Table Data

Soyoon Park○, Seungyoung Lim, Myungji Kim, Jooyoul Lee
LG CNS, Information Technology Research Center

요 약

본 논문에서는 실생활에서 쓰이는 다양한 구조를 갖는 문서에 대해서도 자연어 질의응답이 가능한 모델을 만들고자, 그 첫걸음으로 표에 대해 자연어 질의응답이 가능한 End-to-End 인공신경망 모델 TabQA를 제안한다. TabQA는 기존 연구들과는 달리 표의 형식에 구애받지 않고 여러 가지 형태의 표를 처리할 수 있으며, 다양한 정보의 인코딩으로 풍부해진 셀의 feature를 통해, 표의 row와 column 객체를 직관적이고도 효과적으로 추상화한다. 우리는 본 연구의 결과를 검증하기 위해 다채로운 어휘를 가지는 표 데이터에 대한 질의응답 쌍을 자체적으로 생성하였으며, 이에 대해 단일 모델 EM 스코어 96.0%에 이르는 결과를 얻었다. 이로써 우리는 추후 더 넓은 범위의 양식이 있는 데이터에 대해서도 자연어로 질의응답 할 수 있는 가능성을 확인하였다.

주제어: 기계독해, 질의응답, 표 임베딩

1. 서론

MRC(Machine Reading Comprehension) 또는 질의응답(Question Answering) 과제는 기계가 주어진 문맥과 질문을 이해하여 문맥 내에서 답변이 될 수 있는 영역을 찾아내야 하는 과제이다. 질의응답에 이용되는 데이터는 SQuAD dataset[1], MS MARCO dataset[2] 등이 있으며 질의의 대상이 되는 문맥은 평문으로 이루어져 있다.

하지만 이 기술을 적용해 볼 수 있는 분야의 데이터를 생각해보면 평문으로 이루어진 데이터보다는 다양한 양식과 구조를 가진 데이터에 대해 질의응답이 필요한 경우가 대부분이며, 우리가 조사한 바로는 지금까지 이와 관련하여 사용성 높은 EM(Exact Matching) 스코어를 달성한 연구가 진행된 바가 없다. 따라서 구조를 가진 데이터에 대한 질의응답의 첫걸음으로 우리는 표 데이터에 대해 자연어로 질의응답을 할 수 있는 기계학습 모델을 만들고자 한다. 기존의 방식으로 처리하려면 표를 Triple 형태로 만들어 지식 베이스를 구축하거나, DB화하여 SQL 문법에 맞게 질의해야 한다. 하지만 이러한 방식은 다양한 필드가 추가되거나 변경될 때 유지보수 비용이 많이 들고, 원하는 내용을 찾기 위해 관련 문법에 맞는 질의를 만들어야 하는 데 어려움이 있다.

표에 대해 자연어로 질의응답을 할 수 있는 모델 구축을 위해 우리는 10만 건의 통신상품을 주제로 한 다양한 표와 한국어 질의응답 쌍 데이터를 만들고 표와 질문을 입력으로 받아서 정답 셀을 찾아내는 모델 TabQA를 구축하였다. TabQA는 2차원의 표 데이터를 입력 받아 질문에 대답하기 위한 임베딩 방법과 여러 feature를 이용하여 정답의 위치를 찾아 내는 학습 기반의 end-to-end 인공신경망 모델로, 표의 행/열 구조를 그대로 이용한 질의가 가능하다. 우리 모델은 주어진 테스트셋에 대해 단일 모델 EM 스코어 96.0%를 달성했다. 이를 통해 TabQA 알고리즘을 이용해 표 형태의 데이터에 대해서 자연어로 질의응답 할 수 있다는 가능성을 확인할 수 있었다.

본 논문에서는 표 질의응답을 위해 생성한 한국어 데이터셋과 TabQA의 모델을 소개하고, TabQA의 고무적인 실험 결과 및 향후 연구 방향을 제시한다.

2. 관련 연구

일반적인 자연어 질의응답은 평문 양식의 문맥, 질문, 답변의 데이터 쌍으로 학습하며, 이 중 SQuAD 데이터셋 챌린지가 유명하다. 이 데이터셋 위한 다양한 모델의 연구와 코드가 공개되어 있으며 대표적으로 현재 state-of-the-art 모델인 QANet[3], R-net[4], BiDAF[5]가 있고, 한국어 질의응답 데이터에도 좋은 성능을 낸 S3-Net[6] 등도 있다.

반면 표에 관한 질의응답은 관련 연구가 많지 않고, 있더라도 과제를 위한 데이터 소개에 그치거나 상당히 제한적인 환경에서 질의응답을 수행하는 경우에 한한다.

표에 대해 질의응답을 한 인공신경망 기반의 모델 Yin, et al., 2016[7]에서는 올림픽 주제에 관한 4 단계 난이도의 질의응답을 수행하였다. 이 모형은 240개의 어휘에 대해 10만 개의 데이터를 생성하여 실험하였기에 제한된 환경 이외에 확장이 어려운 한계가 있으며, 여러 단계의 executor를 통해 SQL 구문을 연상하도록 유도하기 때문에 다양한 어휘의 질문에 대응할 수 없다. 우리는 이 논문에서 소개한 데이터 양식을 참고하였지만 더 실용적인 주제인 통신에 관해 다양한 어휘로 데이터를 생성하였다. 또한 TableQA(Vakulenko and Savenkov, 2017)[8]는 표를

Triple 구조로 처리하여 질의응답을 하는 모델이지만, 테스트 정확도가 낮아 사용성이 좋지 않았다.

데이터의 측면에서, 한국어로는 표 질의응답을 위해 공개된 데이터는 없다. 영어로는 Stanford의 Wiki Table Questions 데이터가 공개되어 있지만 yes/no 질문, 개수를 세는 질문 등이 포함되어 있어서 본문 내에서 정답을 찾을 수 있는 MRC 과제를 위한 데이터셋은 아니다.

질의응답을 위한 것은 아니나 인공신경망을 기반으로 표 데이터 유형 분류를 위한 연구가 몇 있었다. 이들은 표의 형태를 바꾸지 않고 RNN을 이용하여 입력 행렬 그대로 처리한 임베딩 방식을 이용했다. TabNet(Nishida, et al., 2017)[9]은 표의 각 셀을 RNN으로 인코딩하여 표를 [num_row, num_col, cell_dim]의 shape을 갖는 텐서로 임베딩하였다. 이와 같이 인공신경망을 위한 임베딩 방식은 우리의 모형에도 영향을 주어 입력 표 형태를 그대로 유지할 수 있도록 하는 동기가 되었다.

3. TabQA 데이터

표 1. 생성 가능한 필드명과 그 경우의 수

열 이름	종류	열 이름	종류
상품명	600	부가서비스	21
월정액	91	약정기간	4
문자	101	광고모델	40
통화	76	사은품	21
데이터	246	가입대상	6
인터넷할인	101	납부방법	6
모바일할인	101	결합상품	4
결합상품할인율	19	통신사	3
부가세	10	멤버십	5
속도제한	3		

표 2. 표와 질문 유형의 예시

상품명	월정액	데이터	통신사	부가세	광고모델	문자
레인보우키즈-56	53000원	0.8GB	갑을텔레콤	2650원	유인나	980건
실속스폰서-45	33000원	13.7GB	갑을텔레콤	3630원	문채원	520건
고급선택형-56	63000원	17.8GB	고려통신	3780원	육성재	580건
레인보우나라사랑-C	49000원	20.3GB	갑을텔레콤	7350원	아스트로	100건

Lv1. {열1}이 {값1}을 가지는 {열2} 질문
Q. 월 요금 63000원인 상품이 뭔가요?
Q. 20.3 GB 사용가능한 상품은 세금이 얼마나 됩니까
Lv2. {열1}의 {min/max} 값 질문
Q. 부가세 가장 센 요금제는 얼마 내야 되니?
Q. 요금 가장 저렴한 경우 얼마지?
Lv3. {열1}이 {min/max} 조건일 때 {열2} 질문
Q. 데이터 가장 적게 줄 때 광고는 누가 해요
Q. 문자 제일 많이 보낼 수 있는 경우 부가세가 얼마나 되나요?
Lv4. {열1}이 {값1 or 범위1} 조건일 때 {열2}의 {min/max}값 질문
Q. 부가세 5000원 이하인 상품 중에 요금 제일 싼 것은 얼마입니까
Q. 통신사가 갑을텔레콤인 것 중 제일 싼 건 얼마야?
Lv5. {열1}이 {값1 or 범위1} 조건일 때 {열2}가 {min/max}값인 {열3} 질문
Q. 메시지 860건이하로 보낼 수 있는 요금제 중 부가세 가장 센 경우의 통신사 어디니?
Q. 멤버십이 실버일 때 요금 가장 싼 상품의 이름이 무엇인가요?

모형 성능을 실험하기 위해 통신과 관련된 19개의 열을 조합하여 20,000개의 표와 그에 대한 자연어 질문 10만건을 인공적으로 생성하였다. 각 표는 최소 5개에서 최대 10개의 행과 열을 가지는데 필드는 열에 해당한다. 모든 표마다 상품명 필드와 월정액 필드는 반드시 포함되도록 하되 이외의 필드는 무작위로 다양한 순서를 갖도록 선택하였다. 풍부한 어휘와 다채로운 숫자 표현을 위해 각 필드는 다양한 종류의 값을 가지도록 설계하였고, 각 열이 취할 수 있는 값의 경우의 수는 <표 1>과 같다. 모든 표 셀의 값은 이 범위 내에서 무작위로 선택되었다.

질문의 유형은 <표 2>과 같이 다섯 가지 유형으로 나눌 수 있다. 이는 SQL의 select-where 구문처럼 한 가지 조건이 주어진 단순한 것에서부터 총 세 개의 열을 분석해야 답할 수 있는 것까지 다양한 난이도로 구성하였다. 또한 자연어의 특징을 살려 같은 의미의 질문이라도 다양한 어휘와 구문을 사용했다. 예를 들어 가격을 물어보는 경우, “월 얼마 내야 해?”, “요금은 얼마인가요?” 등과 같이 동의어 어휘 선택이나 말투에서 존댓말, 반말 등 다양한 표현을 활용하여 실제로 여러 사용자가 질문하듯 다채로운 문장을 고려하였다. 뿐만 아니라 질문 중에는 “얼마야?”, “누가 광고해?”와 같이 열 이름을 직접 언급하지 않는 경우도 있기 때문에 모형은 주어진 표 문맥 하에서 질의의 대상이 어떤 열에 해당하는가를 추론할 수 있어야 한다.

4. TabQA 모델

본 모델은 자연어 질의 $\mathrm{Q}=\{\boldsymbol{w}_t^Q\}_{t=1}^N$가 주어졌을 때, 이 질문을 인코딩하는 질문 인코더, 표에 주어진 각각의 셀의 feature를 임베딩하는 셀 인코더, 표의 row와 column 객체 정보를 생성하는 객체 임베딩 레이어, 마지막으로 주어진 Q에 기반하여 어떤 row, column이 가장 가능성이 높은 객체인지 평가하는 점수 계산 레이어로 구성되어 있다. 이러한 모델은 최종적으로 자연어 질의 Q에 대하여 표의 정답 셀을 반환할 수 있다.

4.1 질문 인코더

질문 인코더는 자연어 질의 $\mathrm{Q}=\{\boldsymbol{w}_t^Q\}_{t=1}^N$에 포함된 의미론적 정보를 하나의 벡터로 압축하며, 이렇게 요약된 질의에 대한 벡터 $\mathbf{q}$는 모델에서 다양하게 활용되었다.

이를 위해 우리는 질의 Q에 대해 단어 임베딩 $\{\boldsymbol{e}_t^Q\}_{t=1}^N$과 음절 임베딩에 CNN[10][11]을 이용해 구한 음절 feature $\{\boldsymbol{c}_t^Q\}_{t=1}^N$를 이어 붙여 질의에 대한 token feature를 생성하였다. 의미론적 압축을 위해 bidirectional LSTM[12]을 이용하여 LSTM 두 방향에 대한 히든 스테이트로부터 최종 질의 벡터 $\mathbf{q}$와 각 단어에 대한 LSTM 출력 벡터인 $\tilde{Q}=\{\boldsymbol{u}_t^Q\}_{t=1}^N$를 얻었다.

$$\boldsymbol{u}_t^Q = BiLSTM_Q\left(\boldsymbol{u}_{t-1}^Q, [\boldsymbol{e}_t^Q, \boldsymbol{c}_t^Q]\right) \quad (1)$$

$$\boldsymbol{u}_i^C = BiLSTM_C\left(\boldsymbol{u}_{i-1}^C, \boldsymbol{v}_{ij}^T\right) \quad (9)$$

$$\boldsymbol{p}_{ij}^T = [\boldsymbol{v}_{ij}^T, \boldsymbol{u}_j^R, \boldsymbol{u}_i^C] \quad (10)$$

마지막으로 필요한 모든 정보를 종합한 각각의 셀 feature $\boldsymbol{p}_{ij}^T$ 와 질문 인코더의 출력 벡터 $\tilde{Q} = \{\boldsymbol{u}_t^Q\}_{t=1}^N$의 어텐션 스코어 계산[13]을 통해 질의를 알고 있는 feature(q-aware representation)[5]를 생성하고, 기존 셀 feature에 이어 붙여 최종 row, column 선택 시 질문과 가장 연관성 있는 셀을 선택하도록 의도하였다.

$$s_t^Q = \frac{\sigma\left(W_p^T \boldsymbol{p}_{ij}^T\right) \cdot \sigma\left(W_u^Q \boldsymbol{u}_t^Q\right)}{\sqrt{d_a}} \quad (11)$$

$$\alpha_t^Q = \frac{\exp(s_t^T)}{\sum_{k=1}^N \exp(s_k^T)} \quad (12)$$

$$\tilde{\boldsymbol{q}} = \sum_{t=1}^{N} \alpha_t^Q \boldsymbol{u}_t^Q \quad (13)$$

$$\boldsymbol{h}_{ij}^T = \left[\boldsymbol{p}_{ij}^T, \tilde{\boldsymbol{q}}\right] \quad (14)$$

4.3 객체 임베딩 레이어

우리 모델은 질의에 해당하는 표 안의 셀을 찾기 위해서 row 객체와 column 객체를 생성하여, 최적의 row와 최적의 column을 각각 선택한 뒤 최종 셀 $\boldsymbol{w}_{ij}^T$를 답변으로 반환한다.

이를 위해 셀 인코더에서 출력된 셀 feature들과 두 개의 개별적인 LSTM을 이용하여 row 객체 $\mathrm{R} = \{\boldsymbol{r}_t^T\}_{t=1}^M$와 column 객체 $\mathrm{C} = \{\boldsymbol{c}_t^T\}_{t=1}^L$를 얻었다.

$$\boldsymbol{v}_j^R = LSTM_R\left(\boldsymbol{v}_{j-1}^R, \boldsymbol{h}_{ij}^T\right) \quad (15)$$

$$\boldsymbol{r}_i^T = \boldsymbol{v}_L^R \quad (16)$$

$$\boldsymbol{v}_j^C = LSTM_C\left(\boldsymbol{v}_{j-1}^C, \boldsymbol{h}_{ij}^T\right) \quad (17)$$

$$\boldsymbol{c}_j^T = \boldsymbol{v}_M^C \quad (18)$$

4.4 점수 계산 레이어

최종적으로 row 객체 $\mathrm{R} = \{\boldsymbol{r}_t^T\}_{t=1}^M$와 column 객체 $\mathrm{C} = \{\boldsymbol{c}_t^T\}_{t=1}^L$의 평가를 위해, 아래와 같은 bilinear form을 적용하여 각각의 점수를 계산한 뒤 softmax 레이어를 적용하였다.

$$P_{row}(i) = \frac{\exp(\boldsymbol{r}_i^T W^s \boldsymbol{q})}{\sum_{k=1}^M \exp(\boldsymbol{r}_k^T W^s \boldsymbol{q})} \quad (19)$$

$$P_{col}(j) = \frac{\exp\left(\boldsymbol{c}_j^T W^s \boldsymbol{q}\right)}{\sum_{k=1}^L \exp(\boldsymbol{c}_k^T W^s \boldsymbol{q})} \quad (20)$$

5. 실험 및 결과

표 3. 레벨 별 질문 개수

구분	총합	Lv1.	Lv2.	Lv3.	Lv4.	Lv5.
학습	**80 K**	16 K	16 K	16 K	16 K	16 K
테스트	**20 K**	4 K	4 K	4 K	4 K	4 K

본 논문의 실험에서는 <표 3>와 같이 16,000개와 4,000개의 표에 대한 5가지 단계의 질문답변 쌍을 생성하여 총 80,000건과 20,000건의 데이터를 각각 학습 데이터와 테스트 데이터로 활용하였다.

5.1 실험 설정

본 논문에서 제시된 모델은 아래 제시된 식과 같이 테이블, 질의, 정답 셀 $\mathcal{B}^{(i)} = \{T^{(i)}, Q^{(i)}, t^{(i)}\}$에 대한 log-likelihood를 최대화하는 방식으로 최적화된다. 이 때 정답 셀 $t^{(i)}$는 특정 row와 column의 교차점으로 해석할 수 있으므로 $t^{(i)} = t_{r^{(i)},c^{(i)}}$로 표시할 수 있다.

$$\mathcal{L}(\mathcal{B}) = \sum_{i=1}^{\mathrm{N}_{\mathcal{B}}} \beta * \log \mathrm{p}\left(\mathrm{r} = r^{(i)} \middle| T^{(i)}, Q^{(i)}\right) + \log p\left(c = c^{(i)} \mid T^{(i)}, Q^{(i)}\right) \quad (21)$$

자체 제작한 인공데이터 셋의 경우, column을 맞추는 것보다는 row를 맞추는 것이 비교적 더 어렵기 때문에 hyper parameter β를 도입하여 row에 대한 loss 함수에 좀 더 가중치를 주었다. 우리의 경우, $\beta = 10$으로 설정하여 학습하였다.

단어 임베딩 행렬의 경우, 기학습된 300차원의 임베딩 행렬로 초기화하여 사용하였고, 음절 임베딩 행렬은 50차원의 랜덤으로 초기화된 행렬로 지정한 뒤, [3,4,5] 사이즈의 필터 100개를 이용한 CNN 레이어를 통해 음절 feature를 강화하였다.

질의를 임베딩할 때는 히든 사이즈가 200인 LSTM을, 그 외 셀과 셀의 row, column 객체 feature를 추상화 할 때는 모두 히든 사이즈가 100인 LSTM을 사용하였고, dropout은 0.2로 고정하였다.

학습을 위하여 Adam optimizer를 사용하고 learning rate를 0.0002로 설정하였으며, 3번의 epoch에서 테스트 에러가 감소하지 않으면 learning rate를 반으로 감소시키도록 조정하였다. 모든 실험에서 미니배치 크기는 25, weight decay는 0.0002, clip되는 gradient의 norm은 10으로 설정하여 50 epoch까지 학습을 진행하였다.

5.2 실험 결과

<표 4>에서 구분 A는 일부 레이어를 더하고 빼으로써 변형한 모델들의 실험 결과 정확도를 요약한 것이다. 주어진 질의에 대해 답이 되는 하나의 셀을 선택하는 것이 태스크이므로, 정확도는 해당 셀을 정확히 찍었는지 여부를 체크하는 Exact Match(EM)를 기준으로 계산했다.

Baseline 모델은, 셀 인코더에서 셀 내용 요약 단계만 거친 후 바로 객체 임베딩 레이어와 점수 계산 레이어를 통해 답변을 반환하게 한 기본 모델로 73.41%의 정확도를 보였다. 이 기존 모델에 필드 정보를 추가해준 결과(Baseline + field) 정확도가 18%p 가량 크게 향상되었다. 이를 통해 column 단위로 질의와 관련된 유용한 정보를 찾아 feature에 추가하는 것이 유의함을 알 수 있다. 여기에 질의를 알고 있는 표현을 추가하면(Baseline + field + q-aware) 약 3%p의 성능 개선 효과가 있다.

표 4. 실험 결과 정확도(EM)

구분	모델	결과	Lv1.	Lv2.	Lv3.	Lv4.	Lv5.
A	Baseline	71.41%	-	-	-	-	-
	Baseline + field	91.06%	88.13%	97.65%	98.00%	85.95%	85.58%
	Baseline + field + q-aware	94.00%	93.15%	97.45%	97.43%	91.05%	90.93%
	Baseline + field + q-aware + row	93.42%	91.75%	97.18%	97.35%	90.68%	90.15%
	Baseline + field + q-aware + col	**96.00%**	**96.73%**	98.80%	98.88%	**92.68%**	**92.93%**
	Baseline + field + q-aware + col + row	94.96%	94.60%	**99.10%**	**99.03%**	91.35%	90.70%
B	Baseline + field + q-aware + col + row	**90.66%**	88.90%	96.70%	96.53%	86.20%	84.98%

가장 좋은 모델은 위 모델에 오직 column 방향으로만 셀 정보를 요약한 모델(Baseline + field + q-aware + col)이었는데(96.0%) 원래 설계했던 row방향, column 방향으로 셀 정보를 모두 요약한 모델(Baseline + field + q-aware + col + row)보다 1%p 이상 성능을 개선했고 특히 Lv1, Lv4, Lv5의 질의에서 좋은 결과를 보였다. 이는 아마도 여러 column을 참조하여 정답을 반환 해야 하는 질의 유형의 경우, row 방향에 있는 셀들의 정보가 혼합되면서 모델의 성능이 감소하는 것으로 보인다. 그렇지만 우리 모델의 경우, 실 예제의 표에는 필드명이 항상 첫 row에 고정되어 있지 않고 첫 column등 다양한 위치에 존재할 수 있다고 가정하기 때문에 가장 마지막 모델(Baseline + field + q-aware +col + row)을 TabQA의 표준 모델로 채택하였다.

TabQA 모델은 필드명의 위치를 첫 행으로 고정해야 하는 제약 조건이 불필요한 특장점이 있다. 셀 인코더 층에서 질의와의 어텐션을 통해 적합한 셀 feature을 생성하는 방법을 학습하기 때문이다. 따라서 본 모형이 이러한 다양한 형태의 표에 대해 강건함을 보이기 위해 데이터 중 50%의 표를 임의로 선택해 전치(transpose)하여 <표 5>와 같이 필드명이 열에 위치하는 표가 섞여 있는 데이터를 생성하였다.

표 5. 전치된 표의 예시

상품명	레인보우키즈-56	실속스폰서-45	고급선택형-56	레인보우나라 사랑-C
월정액	53000 원	33000 원	63000 원	49000 원
데이터	0.8GB	13.7GB	17.8GB	20.3GB
통신사	갑을텔레콤	갑을텔레콤	고려통신	갑을텔레콤
부가세	2650 원	3630 원	3780 원	7350 원
광고모델	유인나	문채원	육성재	아스트로
문자	980 건	520 건	580 건	100 건

이후 각 필드와 질문의 어텐션을 행과 열에 대해서 대칭적으로 적용한 TabQA 모형을 학습시켰다. 그 결과 <표 4>의 B와 같이 TabQA는 전치된 표가 섞여있어도 EM 90.66%의 정확도로 정답 셀을 찾아낼 수 있었다.

5.3 결과 분석

우리 모델이 어떻게 질의와 표를 이해하는지에 대한 통찰을 얻기 위해, 셀 인코더에서 필드 정보를 추가하는 부분과 질의를 알고 있는 표현을 생성할 때의 어텐션 스코어를 좀 더 조사하였다.

먼저, 최종 질의 벡터 q를 기반으로 한 column내에서의 어텐션 스코어는 <그림 2>과 같이 구해졌다. 이 결과는 처음에 필드 정보를 추가할 때 column의 필드명에 대한 정보를 셀 feature에 더하게 하려는 의도 외에도 질의의 중요한 특징을 파악하여 스코어에 반영하는 효과를 보여준다.

예를 들어 Q1과 같이 가장 저렴한 요금제 관련한 질의의 경우 “월정액” column의 가장 작은 요금에 큰 어텐션 스코어가 주어짐을 확인할 수 있다. 특이한 점은 모델이 “월정액” 외의 다른 column에 대해서도 가장 작은 값을 가지는 셀에 큰 어텐션 스코어를 부여하고 있는데, 이는 질의에 “가장 저렴”이라는 단어가 다른 column의 어텐션 스코어까지 영향을 미친 것으로 해석할 수 있다.

Q1. 요금제 가장 저렴할 때, 몇 분까지 전화할 수 있어요? (Lv 3.)

Q2. 속도제한이 500MB인 상품 중, 인터넷 할인 가장 안 되는 상품의 결합 상품은 무엇입니까? (Lv 5.)

그림 2. 최종 질의 벡터 기반의 column 어텐션 스코어

Q2에서는 "속도제한이 500MB"라는 조건문 때문에 "속도제한" column에서 "500MB" 값을 가지는 셀에 큰 가중치를 두는 것을 확인할 수 있으며, 다른 column의 경우 모델이 주로 가장 작거나 가장 큰 값에 주목하고 있음을 알 수 있다. 전반적으로 Lv4~Lv5 유형의 질문이 주어지면 <그림 2>처럼 여러 셀에 고루 어텐션 스코어를 분배하는 것을 확인할 수 있었는데, 이는 질의 유형의 난이도가 높아질수록 하나의 셀의 정보에 집중하는 것이 아닌 여러 셀의 정보를 종합하는 것으로 이해할 수 있다.

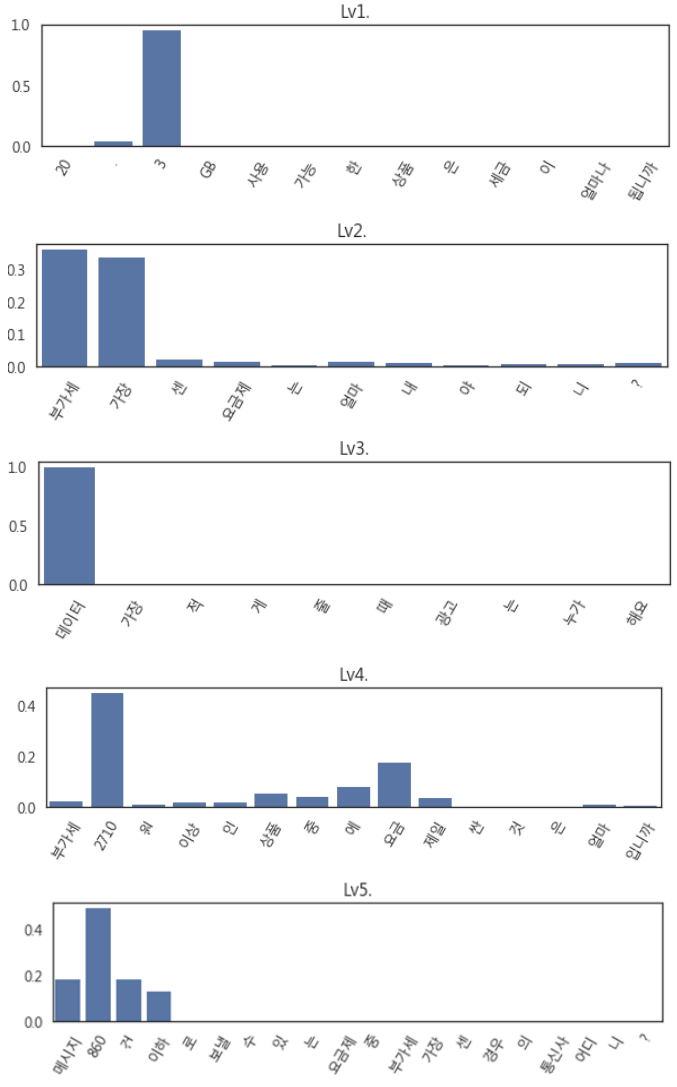

그림 3. 정답 셀 기준, 질의에 대한 어텐션 스코어

둘째로, <그림 3>은 셀 인코더에서 질의를 알고 있는 표현을 생성할 때 정답에 해당하는 셀의 관점에서, 각각의 질의의 단어 중 어느 단어에 집중하는지를 나타낸다.

모든 질의 유형에 대해서 <표 2>에서 제시된 양식을 기준으로 조건 열에 해당하는 부분의 스코어가 높은 것으로 관찰되었고, Lv4 유형의 질문 "부가세 2710원 이상인 상품 중에 요금 제일 싼 것은 얼마입니까?"의 예시에서는 대상 열에 해당하는 질의의 단어 "요금"에도 꽤 높은 가중치를 부여하는 것을 알 수 있었다.

우리 모델은 정답에 해당되는 row 객체, column 객체를 각각 선택하는 2가지 태스크를 수행하는데, row의 선택은 다른 row와의 비교를 필요로 하기 때문에 column 선택보다는 난이도가 높다. 위 결과는 질의를 알고 있는 표현을 생성함에 있어서 row 객체 선택에 필요하리라고 여겨지는 단어들을 우선적으로 고려함을 보여준다.

6. 결론 및 향후 연구

표는 정형화된 자료 구조의 가장 대표적인 형태로, 표에서 특정 셀의 값을 자동으로 찾고자 할 경우 SQL과 같은 지정된 문법의 질의 양식을 갖추거나 Triple 형태의 지식베이스로 만들어야 하는 문제가 있었다. 본 논문은 입력 표 구조를 유지한 채 자연어로 질의응답 할 수 있는 학습 기반의 인공신경망 모델을 소개하였으며, 의미있는 feature를 추가하여 EM 스코어 96.0%를 달성하였다. 이를 통해 구조를 가진 비정형 데이터라 하더라도 자연어로 질의하여 원하는 답을 찾을 수 있는 가능성을 보았다. 또한 이를 위해 실용적인 실험 데이터를 만들었으며, 향후 관련 연구에 도움이 될 수 있도록 공개할 예정이다.

이 분야는 아직 연구가 많이 이루어지지 않은 영역으로, 관련하여 다양한 후속 과제가 있을 수 있다. 그 예로 표 내에서 셀을 선택해야 하는 질문뿐 아니라 조건을 만족하는 셀 또는 행의 개수를 세거나, 예/아니오를 대답해야 하는 질문의 유형을 추가하여 연구해 볼 수 있다. 향후에는 매뉴얼, 규정집, 약관 등 실제로 많이 접할 수 있는 문서와 같이 표와 평문이 섞여있는 문서에 대해서도 질의응답을 할 수 있는 학습 기반의 모델로 연구를 확장하고자 한다.

참고문헌

[1] P. Rajpurkar, et al., SQuAD: 100,000+ Questions for Machine Comprehension of Text, 2016.

[2] T. Nguyen, et al., MS MARCO: A Human Generated MAchine Reading Comprehension Dataset, 2016.

[3] A. Yu, et al., QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension, 2018.

[4] Natural Language Computing Group, Microsoft Research Asia, R-Net: Machine Reading Comprehension With Self-matching Networks, 2017.

[5] M. Seo, et al., Bidirectional Attention Flow for Machine Comprehension, 2016.

[6] C. Park, et al., S3-Net: SRU 기반 문장 및 셀프 매

칭 네트워크를 이용한 한국어 기계독해, KSC, 2017.
[7] P. Yin, et al., Neural Enquirer: Learning to Query Tables with Natural Language, 2016.
[8] S. Vakulenko and V. Savenkov, TableQA: Question Answering on Tabular Data, 2017.
[9] K. Nishida, et al., Understanding the Semantic Structures of Tables with a Hybrid Deep Neural Network Architecture, 2017.
[10] Y. Kim, Convolutional Neural Networks for Sentence Classification, 2014.
[11] Y. Kim, et al., Character-Aware Neural Language Models, 2015
[12] M. Schuster & Kuldip K. Paliwal, Bidirectional recurrent neural networks, 1997.
[13] A. Vaswani, et al., Attention Is All You Need, 2017.