베이지안 모델 불확실성에 기반한 오픈도메인 질의응답

이영훈[O1], 나승훈[1], 최윤수[2], 장두성[2]
[1]전북대학교, [2]KT
dldudgns73@jbnu.ac.kr, nash@jbnu.ac.kr, yunsu.choi@kt.com, dschang@kt.com

Bayesian Model Uncertainty for Open-domain Question Answering

Young-Hoon Lee[O1], Seung-Hoon Na[1], Yun-Su Choi[2], Du-Seong Chang[2]
[1]Jeonbuk National University, [2]KT

요 약

최근 딥러닝 모델을 다양한 도메인에 적용하여 뛰어난 성능을 보여주고 있다. 하지만 딥러닝 모델은 정답으로 제시된 결과가 정상적으로 예측된 결과인지, 단순히 오버피팅에 의해 예측된 결과인지를 구분하기 어렵다. 이러한 불확실성(Uncertainty)을 측정 할 수 없다는 문제점을 해결하기 위해서 본 논문에서는 베이지안 딥러닝 방법 중 하나인 변분추론(Variational Inference)과 몬테카를로 Dropout을 오픈도메인(Open-Domain) 태스크에 적용하고, 예측 결과에 대한 불확실성을 측정하여 예측결과에 영향을 주는 모델의 성능을 측정해 효과성을 보인다.

주제어: 오픈도메인 질의응답, 불확실성, 변분추론, 몬테카를로 드랍아웃

1. 서론

최근 KT의 기가지니, 네이버 클로바, 애플 시리, 삼성 빅스비 등의 인공지능 비서나 스피커와 같은 여러 대화형 인공지능 시스템들이 서비스 되고 있다. 이러한 인공지능 서비스들은 일상적인 대화뿐만 아니라 사용자의 질문에 적절한 정답을 대답하는 형태의 질의응답의 기능도 함께 포함하고 있다.

한편, 뉴럴 네트워크를 이용하여 딥러닝 모델을 구성하였을 때 제시된 예측 결과가 정상적으로 예측된 결과인지, 오버피팅으로 인해 예측된 왜곡된 결과인지를 구분할 수 있는 불확실성(Uncertainty)을 측정할 수 없다는 문제점이 존재한다. 앞에서 언급한 인공지능 시스템들은 실제 소비자들에게 서비스 되는데 있어 시스템이 예측한 결과의 정확도는 매우 중요한 요소로 작용하게 된다. 따라서 예측한 결과가 얼마나 정확한지를 판단하는 불확실성을 측정하는 것은 중요한 문제로 작용하게 된다.

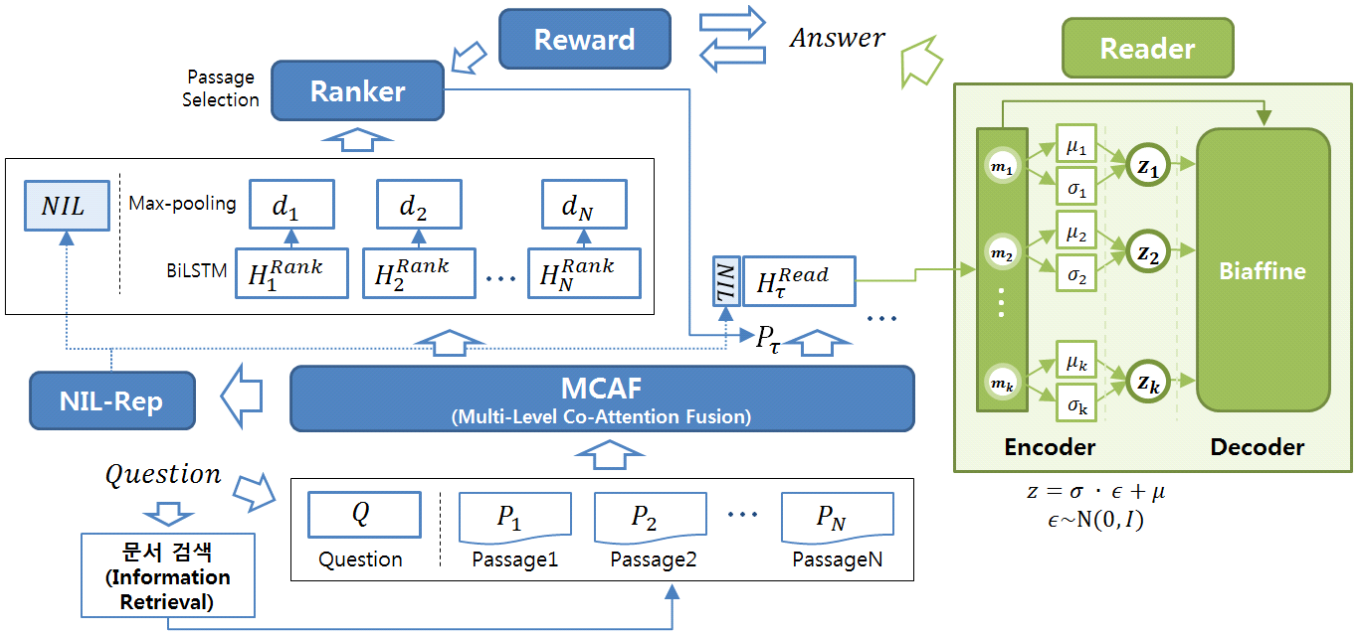
본 논문에서는 이러한 문제를 해결하기 위해 몬테-카를로 드랍아웃(Monte-carlo Dropout)[1]과 변분 추론(Variational Inference)[2-4]을 통해 예측 결과의 불확실성을 측정하여 정답의 결정에 영향을 주는 오픈도메인(Open Domain) 질의응답 모델을 제안하고, 불확실성에 따른 정답 예측 성능을 측정하여 제안 모델의 효과성을 보인다.

2. 관련 연구

IRQA(Information Retrieval Q&A)는 주어진 질문에 대해서 적합한 문서를 검색하고 검색된 문서에서 기계독해(Machine Reading Comprehension)을 통하여 정답을 추출하는 오픈 도메인 질의응답의 방법 중 하나이다. [5]에서는 검색된 문서 집합에서 정답이 포함된 문서를 선택한 경우 보상을 주는 방식의 강화학습을 이용하여 Ranker를 학습시키고 여러 후보 문서에서 최적의 문서를 찾도록 하였다. [6]에서는 이러한 문서 검색 기반 방법이 문서 내에 정답이 포함되어 있지 않는 경우 정답을 찾을 수 없다는 문제가 있기 때문에 이를 해결하기 위해 NIL passage와 NIL dummy를 추가하여 문서 내에서 정답을 찾을 수 없는 경우까지 찾아낼 수 있는 NIL-aware R3 모델을 제안하였다.

GAN(Generative Adversarial Networks)[7], VAE(Variational Auto-Encoder)[8]를 비롯한 생성 모델(Generative Model)에 대한 연구는 꾸준히 진행 되어왔다. GAN은 생성모델인 생성기(Generator)가 생성한 가짜 데이터를 판별기(Discriminator)가 판별하고, 그 결과를 통해 생성기의 학습에 영향을 주는 방식으로 학습을 진행한다. VAE는 인코더와 디코더로 구성되어있으며, 인코더는 입력이 주어졌을 때 확률분포를 학습함으로써, 디코더가 원래의 데이터를 잘 복원할 수 있는 잠재 변수(Latent Variable)를 찾는 것이다. VAE에서는 이러한 문제를 해결하기 위해 변분추론을 사용하게 되는데, 이는 인코더의 파라미터를 이용하여 $q(z)$ 확률 분포를 이상적인 확률 분포 $p(z|x)$에 근사시키는 역할을 수행하는 것을 의미한다. [4]에서는 자연어처리 태스크에서 이러한 변분추론을 적용하기 위해 Neural Variational Inference 프레임워크를 이용하여 문서 모델링과 정답 선택 모델을 제안하였다.

불확실성(Uncertainty)은 예측한 결과가 얼마나 정확한지를 판단하는 것으로, 모델이 어떠한 데이터 입력에 대해서 모른다고 이야기 할 수 있는 것이다. 이러한 불확실성을 예측하는 방법들 중 베이지안 신경망[2-3]을 이용한 방법, 몬테-카를로 드랍아웃[1]이 주로 사용되고 있다.



[그림 1] NIL-Aware R3 변분추론 모델 구조

3. 모델

본 논문에서 제안한 변분추론 모델의 전체적인 구조는 아래의 [그림 1]과 같다. 모델은 [6]와 유사한 구조로 이루어져 있으며, 모델은 크게 Encoding/Matching 단계, 후보 문서집합에서 적합한 문서를 선택하는 Ranker, 변분추론 기반 Reader로 구성되어 있다.

우선 Encoding과 Matching단계에서는 질문 Q와 질문을 검색하여 구성한 문서 집합 P가 입력으로 들어가고, Multi-Level Co-Attention Fusion[9]을 통해 문서와 질문의 상호결합 정보를 얻어 낸다.

$$
\begin{aligned}
H^p &= [h^{pl}; h^{ph}; e^p], \quad H^q = [h^{ql}; h^{qh}; e^q] \\
\hat{h}^{pl}, \hat{h}^{ql} &= CoAttention(H^p, H^q, h^{pl}, h^{ql}) \\
\hat{h}^{ph}, \hat{h}^{qh} &= CoAttention(H^p, H^q, h^{ph}, h^{qh}) \\
h^{pu} = BiLSTM([\hat{h}^{pl}; \hat{h}^{ph}]), & \; h^{qu} = BiLSTM([\hat{h}^{ql}; \hat{h}^{qh}]) \\
\hat{h}^{pu}, \hat{h}^{qu} &= CoAttention(H^p, H^q, h^{pu}, h^{qu}) \\
f^p &= BiLSTM([\hat{h}^{pl}; \hat{h}^{ph}; \hat{h}^{pu}])
\end{aligned}
\tag{1}
$$

질문 Q와 문서 집합 P를 각각 GloVe와 ELMo를 사용하여 임베딩을 구성하고, 양방향 LSTM을 통해 low-level, high-level의 정보인 h^{pl}, h^{ql}와 h^{ph}, h^{ph}를 얻는다. ELMo 임베딩 e와 low-level, high-level을 결합하여 Co-Attention을 적용하고 양방향 LSTM을 통해 understanding-level 정보인 h^{pu}와 h^{qu}를 얻어낸다. 얻어진 의미정보에 Co-Attention을 수행하고, 양방향 LSTM을 통해 [10]와 같이 낮은 차원의 정보부터 높은 차원의 정보가 결합된 f^p를 얻는다. 식 (1)에서 사용된 $CoAttention(x_1, y_1, x_2, y_2)$은 다음과 같이 정의 된다.

$$
\begin{aligned}
S &= ReLU(Wx_1)^T \cdot D \cdot ReLU(Wy_1) \\
\alpha &= softmax_{col}(S), \quad \beta = softmax_{row}(S) \\
p &= \alpha \cdot y_2, \quad q = \beta^T \cdot x_2 \\
\hat{p} &= fusion(x_2, p), \quad \hat{q} = fusion(y_2, q)
\end{aligned}
\tag{2}
$$

식(2)에서 D는 대각행렬이고, $fusion$ 함수는 [11]에서 제안된 방식과 동일하다. 마지막으로 [9]에서 사용된 $AttentionFusion$을 이용한 Self-Attention에 양방향 LSTM을 적용하여 문맥에 대한 최종적인 표상을 얻는다.

$$
\begin{aligned}
F^p &= [h^{pl}, h^{ph}, \hat{h}^{pl}, \hat{h}^{ph}, \hat{h}^{pu}; f^p; e^p] \\
\hat{f}^p &= AttentionFusion(F^p, F^p, f^p, f^p) \\
U^p &= BiLSTM([f^p; \hat{f}^p])
\end{aligned}
\tag{3}
$$

NIL에 대한 표상은 질문의 정보가 결합된 f^p와 매칭 단계의 최종적인 표상인 U^p를 이용하여 구성하게 되는데, [12]에서 제안된 Evidence Decomposition Aggregation과 같이 분해(decompose)를 적용하여 결합한다.

$$
\begin{aligned}
y^{=} &= \frac{U^p f^{p^T}}{f^p f^{p^T}} f^p, \quad y^{\perp} = U^p - y^{=} \\
y^a &= \tanh(y^{=} W_a + y^{\perp} W_a + b_a)
\end{aligned}
\tag{4}
$$

의미적으로 관련된 요소와 그렇지 않은 요소를 분해하여 정답이 존재하는 질문을 찾는 확률을 조절하고, 분해된 Evidence Vector를 결합함으로써 특정 단어에 도움이 되거나 그렇지 않은 Evidence를 모두 결합한다. 모든 단어에 대해서 max-pooling을 적용하여 NIL Vector $\hat{n}$을 얻는다.

Ranker에서는 검색된 문서 집합에서 정답이 포함되어 있을 가능성이 가장 높은 문서를 선택하게 된다. 이때, 문서 집합 내에 정답이 포함 되어 있지 않는 경우

표 2. NR3 VI, MC-dropout 실험 결과

Data Set	Model	Non-NIL Precision	Non-NIL Recall	Non-NIL F1	NIL Precision	NIL Recall	NIL F1	ALL		HasAns	
								EM	F1	EM	F1
Dev	NR3[6]	89.33	67.34	76.79	56.67	84.16	67.73	66.89	68.43	58.12	60.45
	NR3 VI	92.11	48.74	63.74	47.60	91.75	62.68	59.67	60.64	43.55	45.01
	NR3 MC-dropout	94.83	64.49	76.77	57.08	93.07	70.76	69.22	70.39	57.12	58.88
Test	NR3[6]	87.12	62.53	72.81	52.03	81.47	63.50	62.72	63.92	53.37	55.16
	NR3 VI	88.89	43.96	58.83	44.20	88.98	59.06	56.11	56.49	39.27	40.28
	NR3 MC-dropout	91.09	59.62	72.07	52.17	88.31	65.59	64.89	65.91	53.21	54.73

NIL-Passage를 선택하여 문서 선택 단계에서 NIL을 탐지 할 수 있다.

$$\begin{aligned} H^{rank} &= BiLSTM([f^p;\hat{f}^p]) \\ d_i &= MaxPooling(H_i^{rank}) \\ C &= \tanh(W_c[\hat{n};d_1;d_2;...;d_N]) \\ r &= softmax(w_c C) \end{aligned} \quad (5)$$

식(3)과 동일하게 $\hat{f}^p$와 f^p를 결합하여 별도의 양방향 LSTM을 거쳐 H^{rank} 얻고, max-pooling을 적용하여 문서 표현인 u_i를 얻는다. i는 전체 문서 중에서 i번째 문서를 나타내는데, NIL Vector $\hat{n}$과 결합하여 비선형 연산과 softmax 연산을 수행한다. NIL Vector를 함께 결합함으로써 문서 집합 내에 정답이 존재할 확률보다 존재하지 않을 확률이 큰 경우 NIL Passage를 선택하여 NIL을 탐지 할 수 있다.

이미 선택한 문서 내에 정답이 포함되어 있지 않을 경우를 배제할 수 없기 때문에 선택된 문서에 NIL-dummy를 추가하여 Reader 단계에서 NIL을 탐지 할 수 있다.

3.1 NR3 Variational Inference

질문과 문서 정보가 포함된 매칭단계의 최종 표상인 U^p를 입력으로 사용하게 되는데 결정론적인 표상 그대로를 사용하는 것이 아닌, 잠재 변수 $p_\theta(h|U^p)$를 사용하여 모델링 하게 된다. Encoder를 통해 각 단어에 대한 확률 분포 파라미터 μ와 $\log\sigma$를 얻고, 파라미터 확률 분포로부터 샘플링하여 잠재 변수 h를 구하게 되는데, 기울기를 계산하기 위해 Reparameterzation trick ($h = \sigma \cdot \epsilon + \mu$, $\epsilon \sim N(0,1)$)을 사용하게 된다. 제안된 모델의 조건부 확률(conditional distribution) $p_\theta(h|U^p)$은 다음과 같다.

$$\begin{aligned} \pi_\theta &= ReLU(W_1 U^p + b_1) \\ \mu_\theta &= W_2 \pi_\theta + b_2 \\ \log\sigma_\theta &= W_3 \pi_\theta + b_3 \\ h &\sim N(\mu_\theta(U^p), diag(\sigma_\theta^2(U^p))) \end{aligned} \quad (6)$$

각각의 단어에 해당되는 잠재변수 h를 샘플링하게 되는데, 별도의 파라미터를 이용하여 각각 시작점에 대한 잠재변수 h_s, 끝점에 대한 잠재변수 h_e를 얻어내게 된다. Decoder에서는 잠재변수 h_s와 매칭 표상 U^p의 Biafffine 연산을 통하여 정답의 시작점에 대한 점수를 얻어내게 되며, softmax 연산을 통해 시작점에 대한 확률을 얻게 하였다. 정답의 끝점도 시작점과 동일한 방법으로 끝점 파라미터인 W_e와 끝점 잠재변수 h_e를 통하여 얻는다.

$$\begin{aligned} F_s &= U^p W_s h_s^T \\ P_{start} &= softmax(F_s) \end{aligned} \quad (7)$$

Inference network는 식 (6)과 동일한 구조로 구성되며, Inference 파라미터 ϕ를 사용하게 된다. log-likelihood $\log p(y|U^p)$를 최대화하기 위해서 다음의 식과 같이 ELBO(Evidence Lower Bound)를 이용한다.

$$\log p(y|U^p) \geq E_{q_\phi(h)}[\log p_\theta(y|h)] - D_{KL}(q_\phi(h) \parallel p_\theta(h|U^p)) \quad (8)$$

ELBO의 좌항은 reconstruction error에 해당 되고 우항은 KL divergence(Kullback-Leibler divergence)에 해당한다. KL dirvergence란 두 확률 분포의 차이를 계산하는데 사용하는 함수로, 사후확률 분포 p와 q사이의 KL divergence를 계산하여 값이 줄어드는 쪽으로 q를 업데이트하여 사후확률을 근사하는 것이 변분추론의 목적이다.

3.2 NR3 MC Dropout

드랍아웃은 [1]에서 베이지안 신경망의 사후확률의 변분 근사치로 해석 될 수 있는 것을 보였고, 드랍아웃으로 신경망을 최적화하는 것은 베이지안 추론의 형태와 같다.

몬테카를로 드랍아웃(MC Dropout)은 기존의 학습시에만 적용하는 드랍아웃을 테스트 시에도 적용하여 사후분포로부터 샘플을 추출하는 방법으로, 이러한 샘플링을 통해 사후확률을 근사화하여 불확실성을 예측할 수 있게 한다. 이러한 방법은 단순히 테스트 시에 드랍아웃을 적용함으로써, 모델을 크게 변경 시키지 않고 적용할 수 있다는 장점을 가지고 있다.

$$E_{p(x)}[f(x)] = \int f(x)p(x)dx \approx \sum_{i=1}^{N} f(x_i), x_i \sim p(x) \quad (9)$$

위의 식은 몬테카를로 근사에 기반하여 확률 밀도 함수 $p(x)$를 따르는 x에 대한 $f(x)$의 기댓값은 $p(x)$를 따르는 샘플들로 근사 할 수 있다는 의미를 가진다.

실험에서는 기존의 NR3 모델을 $NR3(Q,ratio)$으로 드랍아웃 ratio를 받아 예측된 시작점과 끝점을 샘플링 결

과로 사용하였다. 최종 결과의 confidence를 계산 시, 동일한 예측 결과지만 Span이 다른 경우가 생기기 때문에 실제보다 낮게 계산되게 된다. 예를 들어 샘플링을 10번 수행했을 때 결과가 *{평창 : 5, 대한민국 평창 : 2, 강원도 평창 : 2, 전주 : 1}*의 결과를 보인다면, 가장 높은 빈도를 보이는 *'평창'*의 실제 confidence는 0.9보다 작은 0.5의 값을 보인다. 따라서 가장 높은 빈도를 보이는 결과가 다른 샘플링 결과에도 존재하는 것을 반영하는 word overlapping을 이용하여 confidence를 계산하였다.

4. 실험

실험에 사용된 문서는 한국어 위키피디아와 한겨레 뉴스 기사, 네이버 지식 백과와 블로그를 이용하여 색인하였고, 색인된 문서의 통계는 [6]와 동일하게 구성되어있다. 실험에 사용한 데이터는 KT에서 제공한 질의응답 데이터셋을 사용하였고 데이터셋에 대한 통계는 아래의 [표 1]과 같다 괄호의 숫자는 문서 내에 정답이 존재하지 않는 NIL 질문, 정답이 존재하는 Non-NIL 질문의 개수를 의미한다.

표 1. 실험 데이터 셋 개수

	학습 셋	개발 셋	평가 셋
#질문 (#NIL/#Non-NIL)	15390 (5327/10063)	900 (303/597)	1800 (599/1201)

실험의 평가는 모델의 예측을 반복하여 샘플링한 뒤 word overlapping을 이용하여 가장 높은 빈도를 가지는 후보 정답을 사용하게 되고, confidence 스코어를 계산함으로써 불확실성을 측정하게 된다. 예측된 결과의 confidence가 임계치 이하인 값은 즉, 불확실성이 높은 정답 후보들은 NIL로 예측하여 성능을 측정하였다. 평가지표로는 SQuAD 2.0과 동일하게 Exact Matching과 F1 Score를 이용하여 평가하였고, 정답이 포함되어있지 않은 NIL 질문일 때, NIL을 예측하면 정답으로 평가를 진행하였다. ALL은 정답이 포함되어 있는 문서와 그렇지 않은 문서가 모두 포함되어 있을 때 EM과 F1을 평가하였고, HasAns는 문서 내에 정답이 존재하는 Non-NIL의 경우에만 평가하였다. [표 2]는 NIL-Aware R3의 변분추론과 몬테카를로 dropout의 성능표이다. 변분추론 모델의 경우 전체적인 성능은 감소하였지만, 두 모델 모두 Non-NIL Precision이 증가되는 것을 볼 수 있고, 특히 몬테카를로 Dropout의 성능은 ALL에서 성능이 향상됨을 확인할 수 있다.

5. 결론

본 논문에서는 딥러닝 모델의 불확실성(Uncertainty)을 변분추론과 몬테카를로 Dropout을 이용하여 측정하였고 오픈도메인 태스크에 적용하여 예측한 결과가 얼마만큼의 자신(Confidence)을 가지는지를 이용해 모델의 예측 성능에 영향을 주었다. 향후 연구에서는 범용 언어모델을 적용하여 인코딩 단계에서의 성능을 향상시키고, 기존 모델과 예측성능의 변동 없이 불확실성을 측정하는 방법에 대해서 연구할 예정이다.

참고문헌

[1] Gal Y., Ghahramani Z., "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning", international conference on machine learning. 2016.

[2] Andrieu, C., De Freitas, N., Doucet, A., & Jordan, M. I. An introduction to MCMC for machine learning. Machine learning, 50(1-2), 5-43., 2003

[3] Beal, M. J., Variational algorithms for approximate Bayesian inference (p. 281). London: university of London., 2003

[4] Miao, Y., Yu, L., & Blunsom, P., Neural variational inference for text processing. In International conference on machine learning (pp. 1727-1736)., 2016

[5] S. Wang, M. Yu, X. Guo, Z. Wang, T. Klinger, W. Zhang, S. Chang, G. Tesauro, B. Zhou, J. Jiang, "R3: Reinforced Reader-Ranker for Open-Domain Question Answering", AAAI, 2018

[6] 이영훈, 나승훈, 최윤수, 장두성, "NIL을 고려한 한국어 오픈 도메인 질의응답", 한국정보과학회 학술발표논문집, 2019

[7] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680)., 2014

[8] Kingma, D. P., & Welling, M. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114., 2013

[9] 박광현, 나승훈, 최윤수, 장두성, "Multi-level Co-Attention + Verify를 이용한 기계독해" 한국정보과학회 2018 한국소프트웨어 종합학술대회 논문집, 2018

[10] H.Y. Huang, C. Zhu, Y. Shen, W. Chen, "FusionNet: Fusing via Fully-Aware Attention with Application to Machine Comprehension" ICRL 2018

[11] M. Hu, F. Wei, Y. Peng, Z. Huang, N. Yang, M. Zhou, "Read + Verify : Machine Reading Comprehension with Unanswerable Questions," https://arxiv.org/abs/1808.05759, 2018

[12] S. Kundu, H.T. Ng, "A Nil-Aware Answer Extraction Framework for Question Answering", EMNLP, 2018