CNN Sequence-to-Sequence를 이용한 대화 시스템 생성

성수진[O1], 신창욱[1], 박성재[1], 차정원[1]
창원대학교[1]
{20153057, papower1, tjdwo1289, jcha}@changwon.ac.kr[1]

A Dialogue System using CNN Sequence-to-Sequence

Su-Jin Seong[O1], Chang-Uk Sin[1], Seong-Jae Park[1], Jeong-Won Cha[1]
Changwon National University[1]

요 약

본 논문에서는 CNN Seq2Seq 구조를 이용해 한국어 대화 시스템을 개발하였다. 기존 Seq2Seq는 RNN 혹은 그 변형 네트워크에 데이터를 입력하고, 입력이 완료된 후의 은닉 층의 embedding에 기반해 출력열을 생성한다. 우리는 CNN Seq2Seq로 입력된 발화에 대해 출력 발화를 생성하는 대화 모델을 학습하였고, 그 성능을 측정하였다. CNN에 대해서는 약 12만 발화 쌍을 이용하여 학습하고 1만 발화 쌍으로 실험하였다. 평가 결과 제안 모델이 기존의 RNN 기반 모델에 비해 우수한 결과를 보였다.

주제어: CNN, Sequence-to-Sequence, 대화 시스템, Seq2Seq

1. 서론

대화 시스템은 대화의 기록을 유지하며, 입력된 사용자의 발화에 대해 적절한 응답을 내어주는 시스템이다.

대화 시스템에서 가장 중요한 모듈은 주어진 대화 기록과 입력된 사용자의 발화에 대하여 시스템의 출력 발화를 결정하는 모듈이라고 볼 수 있다. 우리는 그것을 대화 모델이라 부른다.

Seq2Seq(sequence-to-sequence)[1] 등의 end-to-end 구조를 이용하여 자연언어처리의 문제를 해결하려는 시도가 종종 있어 왔다. 대화 시스템에서는 사용자의 발화 처리, 대화 기록 관리, 시스템 발화 생성을 하나의 모델로 수행하는 방식이 이에 해당한다. 이러한 end-to-end 시스템은 기존에 연구된 다단계 시스템에 비해 연구자의 노력과 시간이 적게 소요됨에도 불구하고 높은 성능을 보여주고 있어 여러 분야에서 시도되고 있다.

우리는 CNN Seq2Seq 구조로 한국어 대화 모델을 학습하고 그 결과를 분석하였다. 특히 recurrent unit으로 구성된 LSTM, MTRNN과 비교하여 분석하였다.

2. 관련 연구

한국어 대화를 딥러닝으로 처리하기 위한 연구가 계속되어오고 있다. 특히 시퀀스 데이터를 다루기에 적합한 Seq2Seq 구조[1]를 주로 사용하였다. Seq2Seq 구조는 두 개의 RNN(Recurrent Neural Network)으로 이루어지며 각 cell을 수정한 여러 Recurrent unit이 존재한다.

그 중 LSTM과 MTRNN을 사용하여 한국어 대화 모델을 생성하였을 때[2] MTRNN의 BLEU1이 0.479, BLEU4가 0.220으로 BLEU1이 0.452, BLEU4가 0.189인 LSTM보다 높은 성능을 보였다.

또한 Seq2Seq에 attention mechanism과 함께 양방향 인코더를 사용하여 모델을 생성하고 Greedy decoder를 사용한 경우에는 BLEU가 0.232로 측정되었다[3].

[4]에서는 자연어 처리를 위해 CNN(Convolutional Neural Network)을 encoder로, DCNN(Deconvolutional Neural Network)을 decoder로 사용하는 CNN-DCNN 구조를 제안한다. 이 구조를 이용하여 영어 요약 모델을 생성한 결과 LSTM decoder를 사용한 모델보다 약 1.3정도 낮은 ROUGE-L(Recall-Oriented Understudy longest common subsequence) 성능을 보였으나 단일 GPU에서 CNN-LSTM보다 3배, LSTM-LSTM보다 5배 더 빠른 속도를 보였다.

3. 제안 방법

[4]에서 제안된 CNN-DCNN은 autoencoder 구조로, encoder와 encoder 구조를 미러링한 decoder로 구성되어 있으며 입력 값의 개수와 출력 값의 개수가 동일하다. 입력 값과 유사한 출력 값을 생성하는 것을 목표로 하기 때문에 입력 값을 정답이라 두고 출력 값과 비교하여 loss를 계산한다.

하지만 본 논문에서 모델은 입력 발화가 아닌 출력 발화와 유사한 발화를 생성하는 것을 목표로 하기 때문에 모델의 출력 발화를 학습 데이터 셋의 출력 발화와 비교하도록 loss를 수정하였다. 그림 1은 CNN-DCNN 모델 구조를 나타낸다.

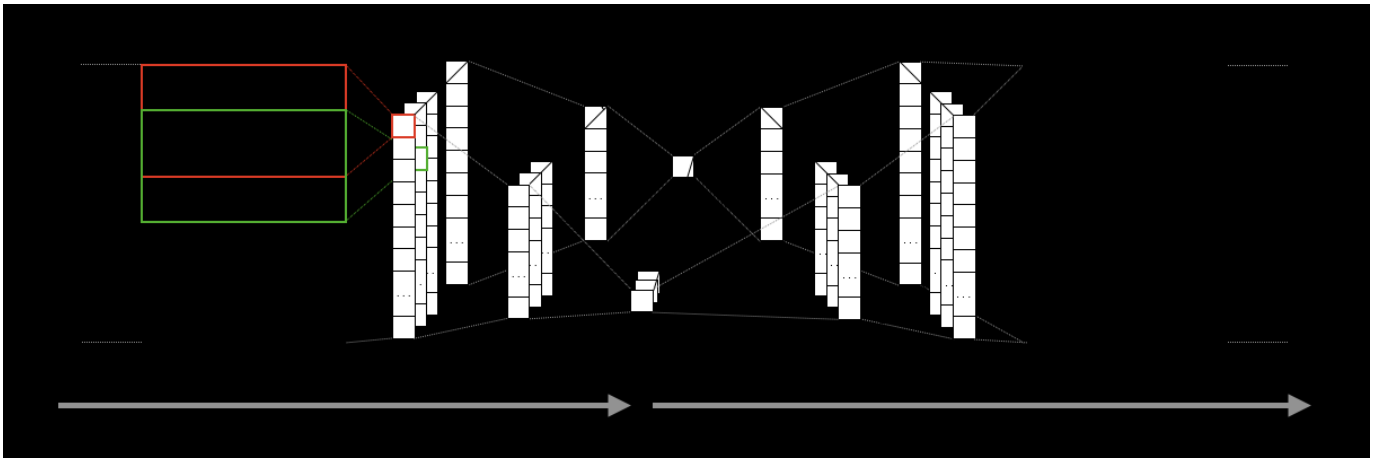

그림 1 Architecture of CNN-DCNN

k는 embedding size, p는 filter size, r은 stride length를 의미한다. filter shape는 모두 5×5이다. convolution을 이용하기 위해 전체 문장에 대하여 입력 문장의 최대 길이로 padding을 수행한 후 embedding size 차원의 벡터로 변환한다. 본 논문에서 embedding size는 300으로 설정하였다. 이렇게 생성된 embedding matrix X는 convolution layer의 입력으로 주어진다. 총 3회의 convolution을 수행하며 필터 수는 순서대로 300, 600, 900이다. convolution의 결과는 latent space h이며 h는 다시 필터 수 900, 600, 300의 deconvolution layer를 거쳐 embedding matrix $\hat{X}$를 재구성한다. $\hat{X}$는 softmax를 거쳐 전체 단어 사전을 대상으로 단어를 선택하고 결과적으로 하나의 문장을 출력한다.

4. 실험

4.1. 실험 설정

CNN-DCNN(CNN Seq2Seq)으로 사용자의 입력에 대하여 적절한 응답을 생성하는 대화 모델을 학습한다. 학습에 대한 평가는 평가 데이터셋의 입력 발화에 대한 모델의 출력물을 평가 데이터셋의 출력 발화와 비교하여 성능을 구한다.

학습에 사용한 코퍼스는 [2]와 동일한 코퍼스를 사용하였다. 코퍼스는 하나의 입력에 대해 여러 출력이 부착되어있는 형식이다. 이를 출력 후보 중 무작위로 하나만 선택하는 방식으로 하나의 입력 발화에 대해 하나의 출력만 갖도록 분리하였다. 기존 코퍼스의 경우 어절 단위로 발화쌍의 중복이 없다. 하지만 코퍼스를 음절로 변환하였을 때는 중복되는 행이 생성된다. 이 때 랜덤으로 추출하여 학습, 검증, 평가 코퍼스로 나누면 같은 발화쌍이 각 코퍼스에 포함될 수 있기 때문에 중복되는 행을 모두 제거하여 최종적으로 14만 5천 여 개의 발화쌍을 생성하였다. 중복은 입력 발화와 출력 발화의 쌍이 동일한 행으로 정의하였다. 표 1은 코퍼스의 통계량을 나타낸다.

표 1. 학습 코퍼스의 통계량

구분	수량	단위
학습 코퍼스	120,000	발화쌍
검증 코퍼스	10,000	발화쌍
평가 코퍼스	10,000	발화쌍

평가는 정답으로 주어진 출력 발화들과 모델의 출력 간의 n-gram을 비교하는 BLEU(Bilingual Evaluation Understudy) Score를 사용하였다.

4.2. 실험 결과 및 분석

본 논문의 실험에서 embedding size는 300, filter shape는 5, filter size는 300, learning rate는 0.00001로 설정하였다. 학습에 dropout을 적용하였으며 optimizer로는 adam을 사용하였다.

입력 발화 X에 대하여 모델이 생성한 발화는 전체 코퍼스에서 나타나는 X에 대한 모든 출력 발화를 reference로 하여 평가된다. 하나의 입력 발화에 대한 출력 발화는 평균 7.6개이다.

LSTM과 MTRNN, CNN-DCNN 모델의 성능을 비교한 결과는 표 2에 기술하였다.

표 2. LSTM, MTRNN, CNN-DCNN 성능표

Model	BLEU1	BLEU2	BLEU3	BLEU4
LSTM-LSTM	0.452	0.278	0.212	0.189
MTRNN-MTRNN	0.479	0.341	0.263	0.220
CNN-DCNN	**0.653**	**0.403**	**0.330**	**0.285**

본 논문의 설정에서 하나의 입력 발화에 대한 출력 발화가 한 개 이상이 되는 경우가 존재한다. 모델의 출력 발화를 분석하기 위해 평가 코퍼스에서 몇 개의 샘플을 추출하여 표 3에 정리하였다.

1번 예시의 경우 정답이 될 수 있는 출력 발화가 2개로 학습에 1개, 평가에 1개의 발화 쌍이 사용되었다. 이 경우에는 학습에 사용된 출력 발화를 정확하게 생성해내는 것을 확인할 수 있었다.

2번 예시는 입력 발화에 대한 출력 발화가 단 1개 존재하는 경우로 입력 발화는 평가 코퍼스에서만 나타난다. 학습 데이터에서 '맛있는 메뉴'와 비슷한 입력 발화로 '맛있는 곳', '맛있는 음식' 등이 나타났고 '맛있는' 과 관련된 입력 발화에 대한 '곱게 먹은 귀신이 때깔도 곱다는데 역시 ~' 라는 출력 발화는 전체 35개 중 6번 나타났다.

3, 4, 5번은 출력 발화가 2개 이상인 경우로 비교적 길이가 짧은 3번을 제외하고 적절한 문장을 생성하지 못하였다. 생성된 문장은 출력 발화들에서 나타나는 음절을 반영하고 있지만 각 음절 사이의 연관성을 파악하여 올바른 단어를 생성하지는 못하였다.

하지만 BLEU score로 평가한 결과를 볼 때, 기존 방식보다 높은 성능을 보였고 특히 BLEU1이 크게 향상되었다. RNN 방식에서는 x_{i-1}번째 예측 결과가 x_i의 예측에 영향을 주기 때문에 x_{i-1}번째 예측 결과가 적합하지 않을 경우 x_i번째 예측 결과도 적합하지 않을 가능성이 크다. 하지만 CNN의 경우 이전 예측 결과가 그 후의 예측에 영향을 주지 않는다. 즉 x_{i-1}번째 예측 결과가 적합하지 않아도 x_i번째 예측은 올바르게 생성할 수 있다. 이 때문에 CNN-DCNN 모델이 RNN 모델보다 정답에 포함되는 음절을 더 많이 생성해낼 수 있어 BLEU1의 성능이 향상되었다고 볼 수 있다. 이는 어순의 영향이 영어보다 크지 않은 한국어를 처리하는데 있어 더 적합한 방법이 될 수 있다.

표 3 평가 결과 예시

1	입력발화	싸 웠 다 형 이 랑
	정답 출력발화 (총 2발화)	다 음 엔 심 판 봐 주 겠 소
		맞 지 마 시 오
	모델 출력발화	다 음 엔 심 판 봐 주 겠 소
2	입력발화	맛 있 는 메 뉴
	정답 출력발화 (총 1발화)	곱 게 먹 은 귀 신 이 때 깔 도 곱 다 는 데 역 시 ~
	모델 출력발화	곱 게 먹 은 귀 신 이 때 깔 도 곱 다 는 데 역 시 ~
3	입력발화	헤 이
	정답 출력발화 (총 12발화)	앗 ~ 반 가 워 !
		왜 불 렀 어 ?
		뜬 금 없 이 왜 ?
	모델 출력 발화	왜 ?
4	입력발화	미 안 미 안
	정답 출력발화 (총 7발화)	무 슨 말 을 그 렇 게 . 그 럴 수 도 있 는 거 지 .
		괜 찮 아 . 그 럴 수 도 있 지 뭐 .
		실 수 할 수 도 있 지 뭐 . 너 무 미 안 해 하 지 마 .
	모델 출력 발화	무 니 아 가 안 야 . 뭐 거 지
5	입력발화	잠 와
	정답 출력발화 (총 4발화)	그 럼 빨 리 가 서 자 ~
		그 래 ! 졸 리 다 니 깐 왜 자 꾸 말 시 켜 !
		자 도 자 도 졸 리 다 . 많 이 자 면 피 부 좋 아 진 다 는 데 더 잘 까 ?
	모델 출력발화	잘 . . . 다

5. 결론

본 논문에서는 CNN-DCNN 구조를 이용하여 한국어 대화 모델을 생성하였다. 앞의 예측 결과가 그 후의 예측에 영향을 주어 전반적으로 잘못된 방향으로 결과가 생성될 확률이 높은 RNN 방식에 비해, 이전의 예측 결과에 대해 의존도가 낮은 CNN-DCNN 방식이 긍정적으로 작용하여 정답과 유사한 문장을 많이 생성했다고 볼 수 있다. 결과적으로 BLUE1은 0.653, BLEU4는 0.285로 LSTM이나 MTRNN에 비해 높은 성능을 보였다.

Acknowledgement

본 연구는 한국전자통신연구원 연구운영비지원사업의 일환으로 수행되었음. [18ZH1300, 오픈 시나리오 기반 프로그래머블 인터랙티브 미디어 창작 서비스 플랫폼 개발]

참고문헌

[1] Ilva Sutskever, Oriol Vinyals, Quoe V. Le, “Sequence to Sequence Learning with Neural Networks”, Neural Information Processing Systems, pp.3104-3112, 2014.

[2] 신창욱, 차정원, “MTRNN을 이용한 한국어 대화 모델 생성”, 제29회 한글 및 한국어 정보처리 학술대회 논문집, pp.285-287, 2017.

[3] 최가람, 최성필, “시퀀스 투 시퀀스(Sequence-to-sequence) 모델을 활용한 대화생성 결과 비교 분석”, 한국정보과학회 학술발표논문집, pp.637-639, 2018.

[4] Yizhe Zhang, Dinghan Shen, Guoyin Wang, Zhe Gan, Ricardo Henao, Lawrence Carin, “Deconvolutional Paragraph Representation Learning”, Neural Information Processing Systems, pp.4169-4179, 2017.