

# MTRNN을 이용한 한국어 대화 모델 생성

신창욱<sup>o</sup>, 차정원

창원대학교

{papower1, jcha}@changwon.ac.kr

## Korean Dialogue Modeling using MTRNN

Chang-Uk Shin<sup>o</sup>, Jeong-Won Cha

Changwon National University

### 요약

본 논문에서는 Multi-layer sequence-to-sequence 구조를 이용해 한국어 대화 시스템을 개발하였다. sequence-to-sequence는 RNN 혹은 그 변형 네트워크에 데이터를 입력하고, 입력이 완료된 후의 은닉층의 embedding에 기반해 출력열을 생성한다. 우리는 sequence-to-sequence로 입력된 발화에 대해 출력 발화를 내어주는 대화 모델을 학습하였고, 그 성능을 측정하였다. RNN에 대해서는 약 80만 발화를, MTRNN에 대해서는 5만 발화를 학습하고 평가하였다. 모델의 결과로 나타난 발화들을 정리하고 분석하였다.

주제어: sequence-to-sequence, 대화 모델, LSTM, MTRNN

### 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.

sequence-to-sequence 등의 end-to-end 구조를 이용하여 자연언어처리의 문제를 해결하려는 시도가 종종 있어왔다. 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 sequence-to-sequence 구조로 한국어 대화 모델을 학습하고 그 결과를 분석하였다. 특히, Recurrent unit으로 LSTM과 MTRNN을 비교하여 분석하였다.

### 2. 관련 연구

여러 연구자들이 end-to-end 구조로 자연어처리 문제들을 해결하기 위해 연구를 진행한 바 있다. sequence-to-sequence 구조는 순서를 갖는 데이터를 입력하여 입력열 전체에 대한 표현을 획득하고, 그것에 기반하여 출력열을 생성하는 구조를 취하고 있다. 이 방식이 기존 자연어처리 분야의 여러 문제에 적용하기 적합하여, 형태소 분석 등 여러 분야에 적용되었다.

[1, 2, 3]에서는 sequence-to-sequence 구조를 이용해 형태소 분석 및 품사 태깅을 시도한 바 있다. 음절 단위, 또는 어절 단위로 인코딩된 입력을 sequence-to-sequence 모델에 입력하고, 그 출력을 형태소 분석 및 품사 부착의 결과물로서 사용한다.

[4]에서는 sequence-to-sequence 구조로 구구조 구문 분석을 수행하였다. 형태소 분석된 입력 문장을 음절 또는 형태소 단위로 입력받아, 구구조 구문분석 결과를 토 큰 단위로 출력하도록 학습하였다.

대화 시스템과 모델링에서 관련된 이전 연구로, 많은 양의 발화 입-출력 쌍을 수집하고 입력되는 발화에 매치된 출력 발화는 내어주는 방식의 연구가 진행되었다. [5]에서는 TF-IDF와 단어 임베딩을 대화 매치에 사용해 MRR 93.9%를 달성하였다.

### 3. 제안 방법

우리는 Multi-layer sequence-to-sequence 구조를 이용해 대화 모델을 학습하였다. Recurrent Unit으로 LSTM[6]과 MTRNN[7]의 변형을 학습하였고, 그 결과를 비교하였다.

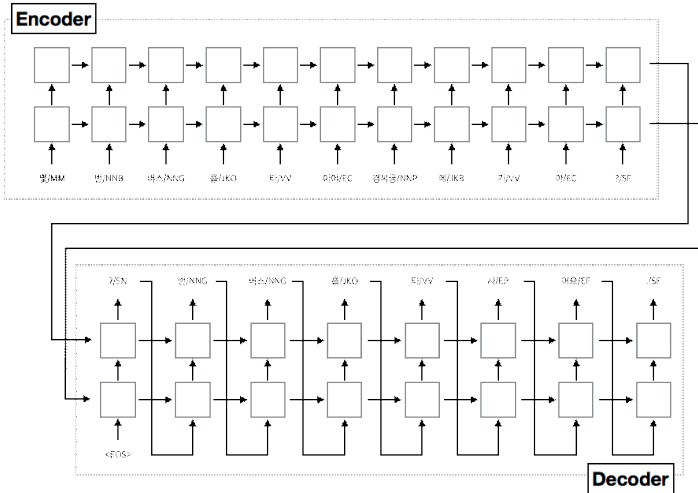
[그림 1]은 sequence-to-sequence의 구조를 설명한다. 그림에서 입력열을 분산 표현으로 생성하는 단계를 encode, 그것을 수행하는 네트워크를 encoder라 칭한다. encoder는 매 입력이 주어질 때마다 다음 식으로 RNN variant의 hidden state를 업데이트 한다.

$$h_t = f(h_{t-1}, x_t) \tag{1}$$

식1에서 f는 non-linear activation 함수이다. 마지막 토큰까지 입력이 완료되었을 때의 hidden state를 sequence 전체에 대한 분산 표현으로 간주한다. 그리고 해당 분산 표현을 최초의 state로 하는 decoder를 동작시켜 출력 시퀀스를 생성한다.

$$h_t = f(h_t, y_{t-1}) \tag{2}$$

위 식2는 decoder에서의 매 타임 스텝 t에서 이전 스텝의 hidden state  $h_t$ 와 출력  $y_{t-1}$ 로 이번 타임 스텝의 hidden state를 생성함을 이야기한다. [그림 1]에서도



[그림 1] multi-layer sequence-to-sequence

이전의 출력 토큰이 다음 스텝의 입력으로 취해지는 것을 확인할 수 있다.

sequence-to-sequence는 여러 RNN variant로 구성할 수 있는데, 본 논문에서는 그 중 LSTM과 MTRNN으로 실험을 진행하였다.

LSTM은 RNN에서 발생하는 그라디언트 소실(vanishing gradient) 문제를 해결한 변형으로 여러 문제에서 RNN 대신에 주로 사용되고 있다. RNN에서는 하나의 state를 관리하는 반면, LSTM은 세 개의 gate를 이용해 두 개의 state를 관리하고 있다. 그 첫 번째는 아래 식의 c이고 cell state라 불리운다. 두 번째 state는 아래의 h이고 hidden state이다. 식의  $\odot$ 는 Hadamard product이다.

$$f_t = \sigma(W_{xh_f}x_t + W_{hh_f}h_{t-1} + b_{h_f}) \quad (3)$$

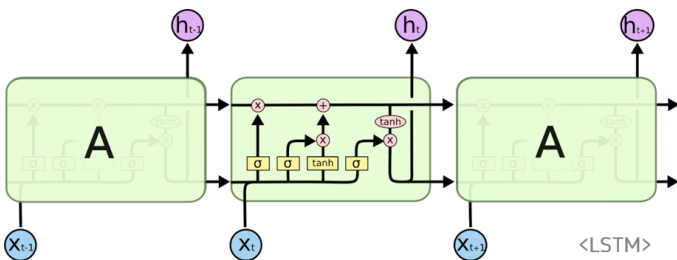
$$i_t = \sigma(W_{xh_i}x_t + W_{hh_i}h_{t-1} + b_{h_i}) \quad (4)$$

$$o_t = \sigma(W_{xh_o}x_t + W_{hh_o}h_{t-1} + b_{h_o}) \quad (5)$$

$$g_t = \tanh(W_{xh_g}x_t + W_{hh_g}h_{t-1} + b_{h_g}) \quad (6)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (7)$$

$$h_t = \tanh(c_t) \odot o_t \quad (8)$$



[그림 2] LSTM 구조

MTRNN은 Recurrent Node의 상태 업데이트 식을 새로이 정의하고, 그것을 파라미터  $\tau$ 로 조절할 수 있도록 하였다. 같은 층의 노드를 그룹화하고, 그룹마다  $\tau$ 를 달리 설정하여 특정 그룹은 빠르게, 특정 그룹은 느리게 업데이트되도록 설정하였다.

우리는 MTRNN  $\tau$ 를 층 내에 그룹에 설정하지 않고, 2개의 층에 각각 다른  $\tau$ 를 설정하여 변형을 시도하였다. 이렇게

하면 층마다 다른 state representation을 갖게 될 것이다.

$$u_{i,t+1} = (1 - \frac{1}{\tau_i})u_{i,t} + \frac{1}{\tau_i} \left[ \sum_{k \in N} w_{jk} x_{k,t} \right] \quad (9)$$

위 식9에서  $\tau$ 를 layer마다 달리 설정해 줌으로써 multi-layer에서 multiple time-scale을 작성할 수 있다. 식에서  $w$ 는 weight matrix,  $x$ 는 입력 혹은 이전 layer의 state,  $N$ 은 입력 차원 혹은 이전 layer의 노드의 수,  $u_{i,t}$ 는 시간  $t$ 에서  $i$ 번째 layer의 state이다.

## 4. 실험

### 4. 1. 실험 설정

sequence-to-sequence로 대화 모델링을 수행한다. 어떤 사용자의 입력에 대하여, 그것의 응답에 해당하는 발화 출력을 목표로 설정한다. 평가로는 평가 데이터셋의 입력 발화를 시스템에 입력하고, 그 출력물을 평가 데이터셋의 출력 발화와 비교하여 성능을 도출한다.

학습에 사용한 코퍼스는 직접 작성하였고, 일상 대화 도메인의 코퍼스이다. 코퍼스에 대한 정보는 [표 1]에 정리하였다. 하나의 입력 발화에 대해 여러 출력 발화가 부착되어 있는 형태이다.

[표 1] 학습 코퍼스의 통계 정보

구분	수량	단위
입력 발화 수	146,276	발화
출력 발화 수	1,123,902	발화
입력 발화 내 발화당 형태소 수	4.50	형태소/발화
출력 발화 내 발화당 형태소 수	9.42	형태소/발화

코퍼스가 하나의 입력 발화에 대해 여러 출력 발화가 부착된 코퍼스이므로, 전처리 과정이 다소 필요하다. 우리는 먼저 여러 출력 발화를 갖는 입력 발화에 대하여, 각 하나씩의 출력 발화를 갖도록 분리하였다. 그리고 이렇게 생성된 총 112만 발화쌍을 80%의 학습 코퍼스와 20%의 평가 코퍼스로 나누었다.

평가는 번역 등의 연구에서 주로 사용되는 BLEU 스코어[8]를 사용하였다. 정답으로 주어진 출력 발화들을 번역에서의 정답과 같다고 보고, 모델의 결과를 평가하는 방식이다.

실험은 가장 적합한 파라미터를 알아내기 위하여 learning rate 등을 수정하며 진행하였고, 그 결과를 4. 2. 실험 결과 및 분석에 기술한다.

### 4. 2. 실험 결과 및 분석

모든 실험에서 hidden unit는 512, embedding size는 256을 사용하였고, layer는 2로 설정, dropout과 attention을 적용하였다. optimizer로 adam을 사용하였다.

첫 번째 실험은 LSTM으로 진행하였다. learning rate는 0.00001로 설정하였고, 학습 발화 89만 9천여 발화로 학습, 22만여 발화로 평가를 수행하였다.

두 번째 실험은 sequence-to-sequence의 Recurrent Node를 LSTM 대신 위에서 설명한 MTRNN의 변형으로 설정한 실험이다. learning rate는 0.0001로 설정하였고,  $\tau$ 는 첫 번째 레이어에 2, 두 번째 레이어에 3을 설정하였다. 5만 개의 발화로 학습, 5만 개의 발화로 평가를 수행하였다. 첫 번째 실험과 학습 및 평가 발화 수에 차이가 있다.

세 번째 실험은 두 번째 실험에서  $\tau$ 를 3, 4로 수정한 실험이다. 두 번째 실험과  $\tau$  설정에만 차이가 있고 다른 설정은 같다.

[표 2] 대화 모델의 실험별 성능

실험	cell type	BLEU1	BLEU2	BLEU3	BLEU4
1	LSTM	0.452	0.278	0.212	0.189
2	MTRNN variant	0.464	0.328	0.248	0.205
3	MTRNN variant	0.479	0.341	0.263	0.220

우리의 설정에서 하나의 입력 발화가 여러 출력 발화에 매치되는 경우가 있다. 우리는 이에 대한 분석을 수행하고자, 평가 코퍼스에서 몇 개의 샘플을 추출해 [표 6]에 정리하였다. 실제 학습은 형태소 품사가 부착된 형태소 단위이지만, 편의를 위해 원문을 복원하여 기술하였다.

[표 3] 대화 모델의 입력 발화에 대한 출력 샘플

입력발화	그래 미안한걸
정답 출력발화 (총 40발화)	내가 더 미안해
	미안하다면 다야? 마음 상했다고~
	나도 미안해~
모델 출력 발화	나도 미안하지~
입력발화	너 사귀는 사람 있어?
정답 출력 발화 (총 35발화)	난 화려한 싱글이야!!
	당연히 있지~
	놀리는 것이오?
모델 출력 발화	난 화려한 싱글이야?
입력발화	철수야 식구중에 언니 있니?
정답 출력 발화 (총 28발화)	없어... 하지만 누나 한 명 있었으면 좋겠다...
	누나 없당... 쫄로지~
	아니 없어. 혼자야
모델 출력 발화	당연한 A형이야?

위 두 예제는 적절히 잘 모델링 된 것으로 볼 수 있다. 첫 번째 예제 ‘그래 미안한걸’은 학습 코퍼스에서 유사한 발화 ‘진짜 미안했어’, ‘그래 미안해요’ 등이 학습되어서 효과를 발휘한 것으로 판단된다. 두 번째 예제는 더욱 확실하게 중첩되는 학습 데이터가 있었다. ‘철수야 너 사귀는 사람 있어?’, ‘철수야 사귀는 사람 있어?’ 등이다.

마지막 예제는 모델이 적절한 답변을 내어주지 못한 경우이다. 혈액형을 묻는 예제 발화쌍으로 ‘철수야 혈액형 뭐야?’ / ‘자상한 A형이야~’가 있었다. 그리고 언니가

있는지 묻는 발화에 대해서는 ‘당연한’으로 시작하는 발화가 없었다. 우리의 실험에서 ‘당연/XR’과 ‘자상/XR’의 임베딩이 유사하고, 이 중에서 잘못 선택한 것이 뒤까지 영향을 미쳐 이러한 결과가 발생한 것이다.

## 5. 결론

높은 성능과 응용력을 갖춘 대화 모델링 기법을 목표로 많은 연구가 진행되고 있다. 우리는 sequence-to-sequence 구조에 Recurrent unit으로 LSTM과 MTRNN을 비교실험하고, 그 결과를 기술하였다. 우리의 실험에서는 학습 발화와 평가 발화에 수의 차이가 있지만 MTRNN이 LSTM에 비해 높은 성능을 보였다.

대화를 모델링하기 위한 연구가 계속 진행되고 있다. 우리의 이번 연구는 대화 모델링을 end-to-end로 수행할 수 있음을 보였다. 학습 후 측정된 성능은 BLEU4 0.161 ~ 0.220의 성능을 보였다. Attention 메커니즘과 MTRNN을 적용하였음에도 낮은 성능을 보여, 대화 모델링을 end-to-end로 수행하기 위해서는 심도 깊은 연구가 진행되어야 할 것으로 보인다.

## 사 사

이 논문은 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. 2017R1D1A1B03033534).

## 참 고 문 헌

- [1] 정의석, 박전규, seq2seq 주의집중 모델을 이용한 형태소 분석 및 품사 태깅, 한글 및 한국어 정보처리 학술대회 논문집, 2016.
- [2] 이건일, 이의현, 이종혁, Sequence-to-sequence 기반 한국어 형태소 분석 및 품사 태깅, 정보과학회논문지, 44권 1호, 57-62, 2017.
- [3] 박건우, 이현구, 김학수, Sequence-to-Sequence 기반 다중 발화 후보를 이용한 형태소 분석기, 한국컴퓨터종합학술대회 논문집, 648-650, 2017.
- [4] 황현선, 이창기, Sequence-to-sequence 모델을 이용한 한국어 구구조 구문 분석, 한글 및 한국어 정보처리 학술대회 논문집, 2016.
- [5] 이호경, 배경만, 고영중, 격투과 워드 임베딩을 활용한 유사도 기반 대화 모델링, 한글 및 한국어 정보처리 학술대회 논문집, 2016.
- [6] Sepp Hochreiter, Jurgen Schmidhuber, LONG SHORT-TERM MEMORY, Neural Computation, 9(8), 1735-1780, 1997.
- [7] Yuichi Yamashita, Jun Tani, Emergence of Functional Hierarchy in a Multiple Timescale Neural Network Model: A Humanoid Robot Experiment, PLoS Computational Biology, 2007.
- [8] Kishore Papineni et al, BLEU: a Method for Automatic Evaluation of Machine Translation, Association for Computational Linguistics, 311-318, 2002.