

Dynamic Memory Network를 이용한 End-to-End 레스토랑 예약 대화 시스템

신창욱^o, 차정원

창원대학교

papower1@changwon.ac.kr, jcha@changwon.ac.kr

Restaurant Reservation Dialog System using Dynamic Memory Network

Chang-Uk Shin^o, Jeong-Won Cha

Changwon National University

요약

본 논문에서는 Dynamic memory network와 GRU decoder를 이용한 end-to-end goal-oriented dialog system을 제안한다. Dynamic memory network로 대화 모델을 학습하고, 발화 생성으로 GRU decoder와 LSTM decoder를 적용하였다. LSTM decoder에 비해 GRU decoder가 0.84% 높은 성능을 내었고, bAbI task6 코퍼스에 대해 수행한 테스트에서 최대 91.10%의 turn accuracy를 기록하였다. 실험 결과로부터 GRU decoder를 적용한 Dynamic memory network가 문맥에 기반한 대화 모델링에 높은 성능을 보임을 확인하였다.

1. 서론

목적 지향 대화 시스템은 대화 시스템 중 하나로, 정해진 도메인과 목표에 관하여 사용자와 대화를 수행하는 시스템을 일컫는다. 대화 중 사용자의 의도를 파악하고 사용자가 원하는 목표를 달성할 수 있도록 대화를 진행한다. 또, 주어진 정보가 부족한 경우 사용자에게 요청해 적절한 정보를 획득할 수 있어야 한다.

Sequence-to-sequence[1], Dynamic memory network[2]와 같은 end-to-end 구조를 이용해 자연어처리의 문제를 해결하려는 시도가 있었다. 대화의 경우, 입력 발화의 언어 이해 모듈, 대화 기록 관리 모듈, 발화 생성 모듈 등을 하나의 구조로 해결하려는 시도가 있었다. 이러한 end-to-end 구조는 대화 도메인에 관한 지식 없이도 데이터만 있다면 학습할 수 있다는 장점이 있다.

Sequence-to-sequence는 RNN encoder를 이용해 입력되는 발화로부터 발화 전체에 해당하는 분산 표현을 획득하고, 다시 그 분산 표현을 RNN decoder에 최초 상태로 입력하여 sequence를 출력하는 구조이다. 자연어처리 문제들 중 형태소분석 등 많은 문제가 각 문장요소에 label을 부착하는 방식을 취하고 있어 Sequence-to-sequence로 모델을 작성하기에 적합하다.

Dynamic memory network는 질의에 적절한 응답을 내어주기 위해 제안된 네트워크이다. 지식과 질의에 해당하는 정보를 각각 따로 입력하고, 그 정보를 적절히 연산하여 memory를 작성한다. 그리고 마지막으로 memory와 질의를 연산하여 정답을 생성한다.

분산 표현으로부터 발화를 생성해내기 위해 사용될 수 있는 방법으로 GRU decoder와 LSTM decoder를 들 수 있다. 우리는 Dynamic memory network의 발화 생성 모듈에 GRU decoder를 적용하여, 대화 모델을 학습하고, 그 성능을 LSTM decoder와 비교하였다.

2. 관련 연구

POMDP 등의 전통적인 접근과 end-to-end 접근의 강점과 약점을 밸런싱하고자 end-to-end network이지만 내부에 모듈을 두고 모듈 간 연결을 정의한 사례가 있었다 [3]. 내부에 데이터베이스(database)와 쿼리(query)를 정의하여 명확한 속성관계는 데이터베이스로 파악을 하고, 사용자 의도를 파악하기 위해 intent network와 belief tracker라는 두 모듈을 정의하였다. intent network와 belief tracker, 그리고 데이터베이스로부터 정보를 받은 policy network는 그 정보를 조합하여 generation network로 전송하고, LSTM과 주의집중 매커니즘(attention mechanism)을 이용한 generation network가 문장을 생성한다.

[3]과 유사하게, [4]에서는 대화 모델을 작성하기 위해 도메인 지식과 action template을 적용한 새로운 구조인 hybrid code networks(HCN)을 제안하였다. HCN은 RNN 모듈, domain-specific software 모듈, domain-specific action template 모듈, conventional entity extraction 모듈로 구성되어 있다. 발화로부터 추출한 자질 벡터를 RNN에 입력으로 하여 action template을 추론하고, 추론된 template에 entity를 다시 채워 넣는 방식으로 학습 복잡도를 줄였다. 신경망에 의해 추출된 entity를 유지하는 모듈과 template에 entity를 채워 넣는 모듈은 인공신경망이 아니라, 규칙에 의해 수행된다는 점이 다른 end-to-end 구조와 차별적이다.

[5]에서는 레스토랑 예약 task를 총 4개 task로 세분화해서 정의하였다. 그리고 memory network 등 몇가지 모델링 기법으로 학습하고 그 성능을 측정하였다. 그리고 레스토랑 이름 등은 미등록어로 나타날 가능성을 높음을 해결하기 위하여 match type이라는 새로운 자질을 제안하였고, 그것을 적용해 memory network 모델의 성능을 개선하였다.

3. 제안 방법

3.1. Dynamic Memory Network

우리는 Dynamic memory network 구조로 주어진 대화 코퍼스를 학습하였다. Dynamic memory network은 입력 모듈로 input module과 question module을 사용한다. 그리고 두 모듈로부터 생성된 분산 표현(distributed representation)을 이용해 episodic memory module의 메모리를 업데이트한다. 메모리는 현재까지 진행된 대화와 질의에 대한 요약 혹은 정리라고 볼 수 있다. input module로 입력된 마지막 문장까지 처리가 완료되고 나면, 생성된 memory와 question 분산 표현은 answer module로 입력되고, answer module은 정답을 생성한다.

추론 과정을 단계별로 자세히 살펴보면 다음과 같다. 먼저 input module과 question module로 입력된 단어열은 GRU를 이용해 인코딩된다[식 1,2]. 여기서 u_t 는 입력과 질의에 나타난 단어, i_t 와 q_t 는 시간 t에서 각 네트워크의 hidden state이다.

$$i_t = GRU(u_t^I, i_{t-1}) \quad (1)$$

$$q_t = GRU(u_t^Q, q_{t-1}) \quad (2)$$

다음은 input module로부터 생성된 input 분산 표현과 question module로부터 생성된 question 분산 표현을 이용해 memory module을 업데이트한다. 이 때 input 분산 표현과 question 분산 표현을 적절히 사용하기 위한 용도로 직접 설계한 주의 집중 메커니즘을 적용하였다[식 3, 4, 5].

$$\hat{g}_t = G(z(c, m, q)) \quad (3)$$

$$z = [c, m, q, c \circ q, c \circ m, |c - q|, |c - m|, c^T W^{(b)} q, c^T W^{(b)} m] \quad (4)$$

$$h_t^i = \hat{g}_t^i GRU(c_p^i, h_{t-1}^i) + (1 - \hat{g}_t^i) h_{t-1}^i \quad (5)$$

여기서 G는 자질 함수 z로부터 생성된 자질을 입력으로 취한다. 자질 함수 z는 입력 c, 이전 메모리 m, 질의 q로 c와 q 사이의 유사도와 c와 m 사이의 유사도를 벡터화한다. \circ 는 element-wise product이다. 이전 메모리가 존재하지 않는 경우, m 대신에 q를 사용한다. G의 결과로 생성된 점수는 입력 문장으로부터 메모리를 업데이트할 때 사용되며, 그 식은 [식 5]와 같다.

메모리를 여러 층으로 쌓을 수도 있다. 그 경우 각 층의 메모리는 이전 층의 메모리 상태와 이번 층의 GRU 상태를 연산하여 현재 메모리 상태를 생성한다[식 6].

$$m^i = f(m^{i-1}, h^i) \quad (6)$$

마지막으로 answer module에서는 위에서 생성된 최종 메모리의 분산표현과 question의 분산표현을 연산하여 정답을 생성한다[식 7]. 본 연구에서는 응답으로 문장을 생성하기 위해, [식 7]의 f로 GRU decoder와 LSTM decoder를 적용하고 그 결과를 비교하였다.

$$a = f(m^i, q^i) \quad (7)$$

3.2. 레스토랑 대화 예약 시스템

우리의 시스템은 입력된 대화 기록과 사용자의 발화에 대하여, 시스템의 발화를 결정하는 시스템이다. 이 때, 레스토랑 예약을 위한 정보 유지 및 사용자로부터의 정보 획득이 수행되어야 한다.

우리는 현재까지 진행된 대화 기록 중 마지막 사용자의 발화를 제외한 나머지를 Dynamic memory network의 input module에 입력한다. 그리고 마지막 사용자의 입력을 question module에 입력한다. 그리고 시스템이 출력하여야 하는 발화를 answer module에 입력해 학습을 수행한다. 모든 정보가 입력되면 Dynamic memory network는 입력된 마지막의 사용자 발화와 현재까지의 대화 기록으로 각 발화의 주의(attention)를 계산하고 다시 주의와 입력 발화를 조합하여 메모리를 연산한다. 그리고 마지막으로 answer module에서는 그렇게 연산된 메모리와 질의로 시스템의 발화를 생성 및 출력한다.

4. 실험

4.1. 실험 설정

실험에 사용한 코퍼스는 The 6 dialog bAbI task[6](이하 bAbI 코퍼스)이다. bAbI 코퍼스는 총 6개의 subtask로 나뉘어져 있다. 모든 task는 레스토랑 예약 도메인의 대화 코퍼스이고, 우리는 이 중 task6으로 실험을 진행하였다.

표 1 시스템의 원문 발화와 번역된 한글 발화.

번역된 발화는 33개의 대표 발화이다.

원문 (총 77발화)	You are looking for a belgian restaurant right? You are looking for a african restaurant right? ...
한국어 발화	<CUISINE> 식당을 찾으시는 거죠? ...
원문 (총 225발화)	I am sorry but there is no chinese restaurant in the east of town I'm sorry but there is no international restaurant in the north of town ...
한국어 발화	죄송합니다. 그런 식당은 없습니다.

표 2 시스템의 발화셋. <SILENCE>는 빈 발화를 의미하는 토큰이고, <REST>와 <REST_ADDR>은 각각 식당의 이름과 식당의 주소를 의미하는 토큰이다.

이전 발화	[U] <SILENCE> [S] 안녕하세요 어떻게 도와드릴까요?
사용자 질의	시내 중심부의 찐 식당 추천해 줘.
정답 발화	<REST>를 추천해 드릴게요.
이전 발화	[U] <SILENCE> [S] 안녕하세요 어떻게 도와드릴까요? [U] 시내 중심부의 찐 식당 추천해 줘. [S] <REST>를 추천해 드릴게요.
사용자 질의	거기 주소가 어떻게 돼?
정답 발화	<REST>의 주소는 <REST_ADDR>이에요.

bAbI 코퍼스는 대화 단위로 작성되어 있다. 우리는 시스템의 발화를 모델링하기 위해 학습 코퍼스를 시스템의 발화 단위로 분리하였다. 분리가 수행된 후 총 학습 대화의 수는 12,154개이고, 각 대화에는 이전까지의 대화와 사용자의 발화, 그리고 시스템의 정답 발화가 포함된다.

bAbI 코퍼스는 영어 코퍼스이므로, 자동 번역 프로그램을 이용해 번역하였다. 그리고 시스템의 발화에 포함된 식당의 이름, 주소 등은 그것을 대표하는 토큰으로 치환하였다. 그리고 각 발화셋의 대표 문장으로 모든 발화를 치환하였다. 작성된 시스템의 발화셋 중 일부를 [표 1, 2]에 정리하였다.

대화 중 사용자의 마지막 발화를 제외한 나머지 대화 기록은 Dynamic memory network의 input module로, 사용자의 마지막 발화는 question module로 입력된다. 입력된 input module과 question module에 의해 memory가 계산되며, 마지막으로, 계산된 memory와 question에 기반하여 시스템의 출력 발화가 결정된다.

위에서 시스템의 발화를 발화셋에 적절히 매핑하였다. 따라서 decoder로 문장을 생성한 후, 그 발화와 가장 유사한 정답셋 내의 발화를 정답으로 최종 선택한다. 이 때 두 발화의 유사도 측정에는 음절 단위 코사인 유사도를 사용하였다. 생성된 결과를 p, 정답 후보를 c, 단어의 빈도 벡터를 w라 할 때, 코사인 유사도 식은 [식 8]과 같다.

$$d_c = \frac{\sum_{k=1}^n w_{pk} \cdot w_{ck}}{\sqrt{\sum_{k=1}^n w_{pk}^2} \cdot \sqrt{\sum_{k=1}^n w_{ck}^2}} \quad (8)$$

4.2. 실험 결과

[표 3]에 실험 성능을 정리하였다. 발화 생성 모듈로 GRU decoder와 LSTM decoder를 비교하였다. 평가 코퍼스는 학습 코퍼스와 같은 전처리를 수행하였으며, 그 양은 10,472대화이다.

표 3 발화 생성 모듈별 실험 성능

발화 생성 모듈	Turn Accuracy
GRU decoder	91.10%
LSTM decoder	90.26%

GRU, LSTM decoder의 평가 결과 중 오류로 나타난 샘플의 일부를 [표 4]에 정리하였다. 총 33개 발화셋 중 예 많은 오류가 나타난 항목만을 추려서 정리하였다.

첫 번째 항목의 정답 발화는 시스템이 사용자에게 남은 요구사항이 있는지 질의하는 발화이다. 그리고 오류 발화는 시스템이 처음 사용자로부터 발화를 입력받았을 때 시스템이 출력하는 인사말이다.

이 둘 사이의 오류는 두 발화 사이의 높은 유사도와 시스템의 문맥 유지 실패와 학습 코퍼스 편향(bias)에 기인한다. 위 시스템의 질의는 대화가 최소 10턴 이상 진행된 상태에서만 출현하였고, 인사말은 항상 대화의 첫 번째 턴에 발생하여 문맥으로는 명확하게 구분이 가능할 것으로 판단된다. 그러나, 두 발화 사이에 중복되는 음절이 비교적 많고, 의미적으로도 유사하여 모델링에 어려움이 있을 것으로 판단된다.

실제 시스템이 '더 도와드릴 것이 있을까요?'를 출력하였을 때 위 두 후보 발화의 코사인 유사도를 비교해 보았더니 총 40개의 샘플에서 두 후보 발화의 코사인 유사도

표 4 GRU/LSTM decoder의 오류

정답 발화	더 도와드릴 것이 있을까요?
오류 발화	안녕하세요. 어떻게 도와드릴까요?
빈도	105(GRU), 139(LSTM)
정답 발화	어떤 지역이 좋으세요?
오류 발화	어떤 음식이 좋으세요?
빈도	496(GRU), 498(LSTM)

가 동일하게 나타났다. 즉, 해당 40개 샘플에 대해서는 음절 코사인 유사도로 애매성 해소가 불가하다는 것이다. 이는 RNN decoder로 발화 등을 출력한 후, 최종 정답을 분류할 때에 음절 cosine 유사도가 부적절함을 시사하며, 더 적절한 방법을 연구 및 적용하여야 한다.

[표 4]의 두 번째 항목은 사용자가 시스템에게 원하는 음식의 종류를 발화하였을 때에 주로 발생하는 시스템의 두 발화간의 오류이다. 시스템이 문맥과 사용자의 질의로부터 두 개의 추가 정보가 필요함을 추론하였으나, 두 발화 중에 하나를 결정할 수 있는 정보가 충분하지 않다. 실제 학습 코퍼스에도 음식의 종류가 결정된 후 시스템의 추가 정보 요청에 일관적인 패턴이 존재하지 않다. 따라서 둘 중 고빈도로 나타난 '음식' 질의를 출력하게 된다. 실제 학습 코퍼스에서 지역 질의 발화가 538회, 음식 질의 발화가 806회 발생하였다. 위에서 분석한 두 발화 오류의 합은 GRU 기준 601발화, 전체 오류 900여 발화의 약 64%에 해당한다.

6. 결론

우리는 Dynamic memory network를 이용해 goal-oriented 대화 모델을 학습하였다. 그리고 구조 내부의 정답 출력 모듈로 GRU decoder와 LSTM decoder를 비교하였다. GRU decoder를 출력 모듈로써 이용한 Dynamic memory network는 turn accuracy 91.10%로 문맥에 기반한 대화 모델링이 가능함을 보였다.

7. 감사의 글

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

8. 참고문헌

- [1] Ilya Sutskever et al., "Sequence to Sequence Learning with Neural Networks", arxiv, 2014.
- [2] A. Kumar et al., "Ask Me Anything: Dynamic Memory Networks for Natural Language Processing", arxiv, 2016.
- [3] T. Wen et al., "A Network-based End-to-End Trainable Task-oriented Dialogue System" arxiv, 2017.
- [4] J. Williams et al., "Hybrid Code Network: practical and efficient end-to-end dialog control with supervised and reinforcement learning", arxiv, 2017.
- [5] A. Bordes et al., "Learning End-to-End Goal-Oriented Dialog", arxiv, 2017.
- [6] <https://research.fb.com/downloads/babi/>