의사 형태소 단위 채팅 시스템

김시형○, 김학수
강원대학교 컴퓨터정보통신공학과
sureear@kangwon.ac.kr, nlpdrkim@kangwon.ac.kr

Chatting System that Pseudomorpheme-based Korean

Sihyung Kim○, HarkSoo Kim
Kangwon National University Computer and Communication Engineering

요 약

채팅 시스템은 사람이 사용하는 언어로 컴퓨터와 의사소통을 하는 시스템이다. 최근 딥 러닝이 큰 화두가 되면서 다양한 채팅 시스템에 관한 연구가 빠르게 진행 되고 있다. 본 논문에서는 문장을 Recurrent Neural Network기반 의사형태소 분석기로 분리하고 Attention mechanism Encoder-Decoder Model의 입력으로 사용하는 채팅 시스템을 제안한다. 채팅 데이터를 통한 실험에서 사용자 문장이 짧은 경우는 답변이 잘 나오는 것을 확인하였으나 긴 문장에 대해서는 문법에 맞지 않는 문장이 생성되는 것을 알 수 있었다.

주제어: Attention Mechanism Encoder-Decoder, 의사 형태소, 채팅, 템플릿

1. 서론

채팅 시스템은 자연어를 통해 컴퓨터와 사용자간 대화가 이루어지는 시스템이다.[1] 최근 다양한 기업에서 채팅 시스템에 관한 연구가 활발히 진행되고 있다. 채팅 시스템은 일반적으로 검색 모델[2]과 생성 모델[1]로 나뉜다. 검색 모델은 사용자의 발화에 대해 기존에 정의된 답변 중 가장 적절한 답안을 검색하여 출력해 주는 모델이다. 검색 모델은 기존의 답변을 검색하는 것이기 때문에 문법적인 오류가 생기는 경우가 없고, 잘못된 단어를 사용하는 일이 드물다. 그러나 기존에 정의된 답변 외에는 답을 할 수 없다. 그에 반해 생성 모델은 답변을 모델이 생성하기 때문에 같은 의미의 문장을 다른 문법 구조로 입력 하는 경우에 답안이 변화하여 마치 사람과 대화 하는 느낌을 받을 수 있다. 하지만 생성 모델은 문법적인 오류가 많을 뿐만 아니라 많은 양의 학습 데이터가 필요하다. 최근 딥 러닝이 큰 화두가 되면서 검색 모델에 관한 연구는 물론 생성 모델에 관한 연구도 활발히 진행 되고 있다.[3] 본 논문에서는 딥 러닝을 이용한 생성 모델 한국어 채팅 시스템을 제안한다.

2. 관련 연구

기존의 채팅 시스템에는 규칙 기반 패턴 매칭 방법 및 키워드 인식 방법이 있다. 'ELIZA' 는 사용자가 입력한 문장을 규칙에 따라 매칭하면서 키워드로부터 정의된 문장으로 답변한다.[4] 마코브 모델(Markov model)을 이용한 방법인 'MegaHal' 은 채팅 시스템에 마코브 모델을 적용한 대표적인 예이다.[5] Dana Vrajitoru는 템플릿과 유전 알고리즘을 통해 문장을 생성하는 방법을 제안하였다.[6][7] 또한 클러스터 정보 검색을 이용한 방법은 입력된 문장과 주제의 유사도를 계산하고, 입력 문장과 주제 내에서 가장 높은 유사도를 갖는 문장을 다시 계산하여 일정 임계치보다 상위에 있는 문장만 시스템이 출력하도록 한다.[8] 마지막으로 단계 별 검색 방법은 어휘 수준, 의미 수준, 패턴 기반 말 잇기와 같은 단계를 나누어 데이터베이스를 구축 하고, 검색 하는 방법이다.[2] 딥 러닝을 이용한 방법으로는 주로 Encoder-Decoder 방법을 사용한 모델들이 있으며[3][9], CNN(Convolutional Neural Network)과 LSTM(Long short-term memory)을 결합한 검색 모델도 있다.[10] 채팅 데이터 구축을 위한 연구도 진행되고 있다.[11] 그러나 띄어쓰기가 기본 단위인 영어에 비해 기본 단위가 띄어쓰기가 아닌 한국어 에서는 딥 러닝 모델에 채팅 시스템을 적용하기가 어렵다. 본 논문에서는 입력 단위를 의사 형태소로 입력 하는 채팅 시스템을 제안한다.

3. 의사 형태소 단위 채팅 시스템

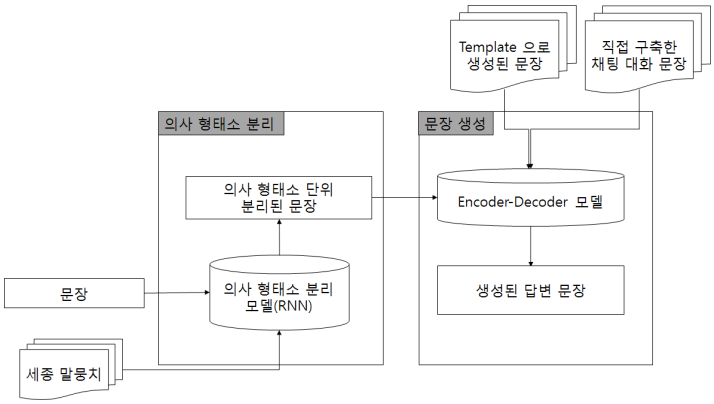

그림 1 제안 모델의 구조도

[그림 1]은 제안 모델의 구조도를 보여준다. 제안 모델은 RNN(Recurrent Neural Network)을 통한 의사 형태소[12] 분리와 Encoder-Decoder 모델을 통한 문장 생성 부분으로 구성된다. [그림 1]과 같이 문장을 의사 형태

소 분리 모델에 입력으로 사용하여 의사 형태소를 얻고 채팅 대화 문장으로 학습된 Encoder-Decoder 모델을 통해 답변을 자동으로 생성한다.

3.1 의사 형태소 분석

의사 형태소는 일반적인 형태소와는 다르게, 어절의 소리를 유지하면서 최소한의 의미를 가지는 형태소를 말한다.[12] 기존 형태소의 경우 형태소를 분리해 낼 때 접사 처리, 불규칙 현상 및 음운 현상, 복합어 처리, 미등록어 처리 등 수많은 고려사항이 존재 한다. 그러나 의사 형태소는 소리를 유지하므로 분리 및 결합이 매우 간편할 뿐만 아니라 다양한 언어표현이 등장하는 채팅 시스템에 매우 적합하다. 의사 형태소 분석을 위한 모델은 RNN을 이용해 구현 하였다. [그림 2]는 의사 형태소 분리 예시를 보여준다.

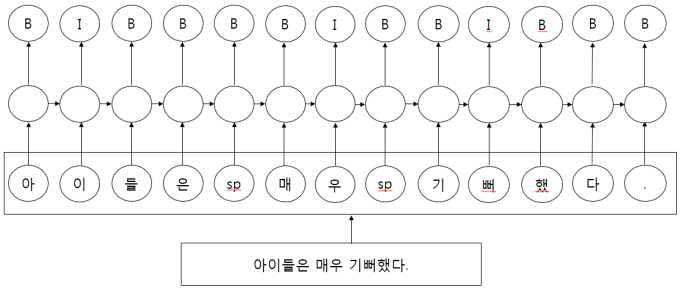

그림 2 의사 형태소 분리 예시

의사 형태소 분리 모델은 Sequence labeling을 사용한다. 문장을 음절 단위로 분리하여 입력으로 사용하고, 의사 형태소 단위를 측정하는 B, I 태그 결과가 출력으로 표시된다. B는 의사 형태소의 시작을 의미하고, I는 단위가 이어짐을 의미한다. [그림 2]에서 “아이들은 매우 기뻐했다.” 가 입력으로 들어가면, 출력으로 “B I B B B B I B B I B B B”가 출력된다. 이를 통해 의사 형태소를 생성한 결과는 “아이 들 은 sp 매우 sp 기뻐 했 다 .”가 된다.

3.2 채팅 답변 생성 시스템

본 논문에서는 채팅의 답변을 생성하기 위해 Attention mechanism Encoder-Decoder Model을 사용한다. [그림 3]은 Attention mechanism Encoder-Decoder Model을 이용한 채팅 답변 생성 시스템의 예시를 나타낸다 .

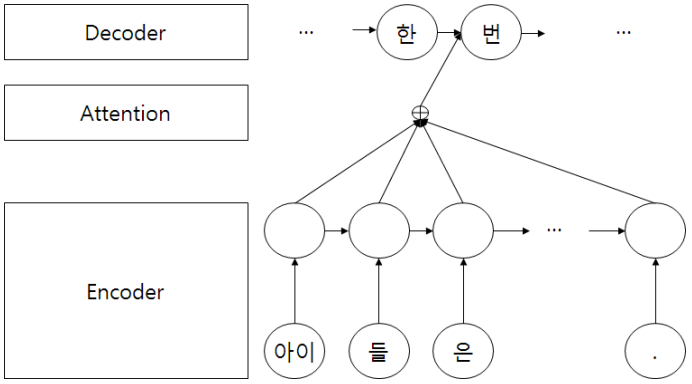

그림 3 Attention mechanism Encoder-Decoder Model

[그림 3]에서 Encoder는 의사 형태소 단위로 분리된 문장을 입력으로 넣는다. Encoder를 통해 생성된 은닉 계층(Hidden layer)의 값에 가중치를 합산하여 입력 문장을 표현하고, 이를 Decoder의 입력으로 사용한다. Decoder는 각 출력마다 확률을 구하는 방법인 beam search를 이용하였다. 본 논문에서는 beam size를 25로 설정하였다.

4. 실험

4.1 실험 준비

의사형태소 분리 모델의 실험을 위해 세종말뭉치를 의사형태소 단위로 태깅한 데이터를 사용하였다.[13] 의사형태소 실험 데이터는 총 581,383문장, 29,667,230개의 음절로 이루어져 있으며, 문장의 10%인 58,138문장을 실험에 사용하고, 나머지를 학습에 사용하였다. 다음으로 채팅 답변 생성 시스템의 실험하기 위해 직접 구축한 채팅 대화 80,417쌍을 사용하였다. 본 논문에서는 학습 데이터 부족 문제를 해결하기 위해 한 담화의 출력을 다음 입력으로 넣는 greedy 기법[3]을 사용하였다. 또한 본 논문에서는 DBPedia 2015-04[14]의 데이터 중 관계 트리플 정보(주어_Subject, 술어_Predicate, 목적어_Object)로 이루어진 mappingbased-properties_ko.ttl를 사용하여 일정한 문장 구조를 가지는 템플릿을 생성하고 이를 통해 학습 데이터를 확장하였다. [그림 4]는 지미 카터의 Nationality에 대한 템플릿의 예시이다.

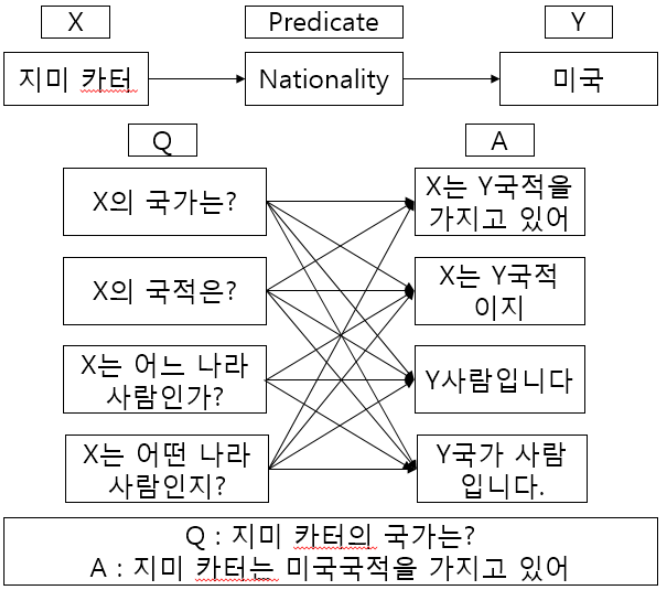

그림 4 템플릿을 통한 학습 데이터 생성 예시

[그림 4]의 예와 같이 템플릿을 각 Predicate마다 질의, 응답을 각각 3~4개를 구축하여, 학습 데이터를 확장하였다. 아래 <표 1>은 본 논문에서 사용한 Predicate의 종류와 개수 이다.

표 1 Predicate의 종류와 개수

Predicate	데이터 개수
birthPlace	31,650
occupation	22,084
nationality	13,632
deathPlace	4,619
spouse	1,942
place	1,704
award	1,689
developer	1,632
parent	1,269
child	1,254
capital	728

4.2 의사 형태소 분석 실험 결과

표 2 의사 형태소 분석기 실험 결과

	재현율	정확률	F1-score
LSTM	95.38%	96.45%	95.91%

<표 2>는 의사 형태소 분석기의 성능을 보여준다. 성능은 의사 형태소 단위로 측정하였으며 정확률(Precision), 재현율(Recall), F1-score에 대해 측정하였다. <표 3>은 의사 형태소 분석기에 사용한 Neural Network의 Parameter 값이다.

표 3 의사 형태소 분석기 Parameter

Parameter	value
Word dimension	50
Memory cell	50
Layer	1
Initial_value	Random_initialize
Epoch	50
Learning rate	0.001
Time step	200
Dropout	0.5
Optimizer	AdamOptimizer

4.3 채팅 답변 생성 실험 결과

본 논문에서는 생성된 문장의 성능을 측정하기 어려워 모델을 통해 생성된 샘플 결과들을 보여주고, 이를 분석하였다. Attention mechanism Encoder-Decoder의 Parameter 값은 <표 4>와 같다.

표 4 Encoder-Decoder Model Parameter

Parameter	value
Word dimension	512
Memory cell	512
Layer	1
Initial_value	Random_initialize
Learning rate	0.5
Learning decay	0.99
Optimizer	Gradient Descent Optimizer

샘플은 입력 문장 하나에서 나올 수 있는 여러 답변을 출력하였다. 출력 된 답변에 UNK(단어장에 없는 단어)가 나오는 문장은 삭제 하였다. '+' 는 어절 내에서 의사 형태소의 구분을 나타낸다.

Q : 정말 맛있+어 .
A_1 : 나+도 먹+고 싶+다+. A_2 : 나+도 좋아+해+봤+어+? A_3 : 나+도 먹+고 싶+어+땠+어+봤+어+? A_4 : 맞+아+? A_5 : 나+도 먹+어+? A_6 : 나+도 먹+고 싶+어+때+? A_7 : 나+도 그래+?
Q : 밥 먹+고 싶+다+.
A_1 : 나+도 그래+. A_2 : 왜+? A_3 : 나+도 그+럴+까+?
Q : 배+고+파
A_1 : 왜+? A_2 : 그+럴+까+? A_3 : 그래+? A_4 : 그렇+구나+. A_5 : 그래+?
Q :지금+도 바람+이 불어+?
A_1 : 응+. A_2 : 응
Q : 꽃 심+으러 갈래+?
A_1 : 왜+? A_2 : 너+는 좋+아+?
Q : 사랑+이 뭐+야+?
A_1 : 사랑+이+야+.

그림 5 입력 문장이 짧은 경우의 예

[그림 5]를 보면, 문장이 짧은 경우의 예시들이 있다. 간단한 문장의 경우 답변이 잘 생성 되는 것을 확인 할 수 있었지만 후보 문장에 문법적인 오류가 있는 문장들도 포함 되어 있는 경우도 확인할 수 있었다.

Q : 너무 불가능+한+걸 목표+로 잡+았+구나
A_1 : 너+도 줄 치+는 하+면 줄 하+는 않+아 . A_2 : 너+도 줄 치+는 하+지+는 줄 같+아+. A_3 : 응+. 그렇+게 치+는 하+는 A_4 : 않+아+닌 않+아+? A_5 : 응+~ A_6 : 너+도 줄 치+는 하+면 줄 하+는 않+아+?
Q : 뭐+라고+? 내+가 한때 얼마나 잘 나+갔+는데+~
A_1 : 그래+?
Q : 아이+들+은 매우 기뻐+했+다+.
A_1 : 아이+들+은 한 번 했+던 A_2 : 한 아 ? A_3 : 아이+들+은 한 번 해+야+. A_4 : 아이+들+은 한 번 해 . A_5 : 아이+들+은 한 번 들어 .

그림 6 입력 문장이 길 경우의 예

[그림 6]은 [그림 5]에 비해 긴 문장에 대한 예시이다. 긴 문장을 입력으로 하는 경우 짧은 문장인 "응~", "그래?" 이외에는 문법적인 오류가 매우 많다. 또한 "아이들은 매우 기뻐했다."의 경우, 부적절한 답안만 생성 되었다. 이는 시스템의 답변 문장 길이가 길어질수록 문장 생성이 힘들다는 것을 보여준다.

Q : 아이유+는 무슨일+을 하나+?
A_1 : 가수 일+을 일+을 하+고+있+어 A_2 : 가수 배우관련 일+을 일+을 합 A_3 : 가수관련 일+일+을 일+을 대단+한+거+있+어 A_4 : 가수 일+을 합+니다 A_5 : 가수+의 일+을 일+을 합+니다 A_6 : 아이유+는 일+을 대단+한+거+같+아 A_7 : 가수 배우 일+을 해 대단+한+거+는 A_8 : 가수 배우+의 일+을 해 A_9 : 가수 배우 일+을 해 대단+한+거+같+아 A_{10} : 가수 배우관련 일+을 일+을 합+니다 A_{11} : 가수 배우+의 일+을 하+가일+을 대단 A_{12} : 가수 배우 일+을 해 대단+한+거+같+배우 A_{13} : 가수 배우 목사 일+을 대단+한+거 A_{14} : 가수관련 일+을 라이터 일+을 대단 A_{15} : 가수+의 일+을 합+니다 A_{16} : 가수관련 일+을 라이터+해 라이터+합+니다 A_{17} : 가수 배우+의 일+을 하+가일+을 일+을
Q : 김시형+은 무슨일+을 하나 ?
A_1 : 배우+의 일+을 하+고+있+어 A_2 : 배우 일+을 합+니다 A_3 : 배우관련 일+을 합+니다 A_4 : 가수 일+을 해 대단+한+거 A_5 : 만화+가관련 일+을 합+니다 A_6 : 아나운서 배우 일+을 합+니다

그림 7 템플릿 기반 문장 생성의 예

[그림 7]은 템플릿 형식에 맞춘 문장을 넣어 본 예시이다. 학습 데이터에 존재하는 문장인 "아이유는 무슨 일을 하나?"의 경우 문장이 잘 생성 되었고, "아이유"를 학습 데이터에 없는 "김시형"으로 바꾸었을 때도 템플릿 형식에 맞춰진 정답이 생성 된다는 것을 알 수 있었다.

5. 결론 및 향후 연구

본 논문에서는 의사 형태소를 채팅 시스템에 적용하여 문장의 분리 및 결합이 쉽고 Attention mechanism Encoder-Decoder 에 쉽게 적용 할 수 있는 모델을 제안하였다. 입력 문장이 짧은 문장은 답변이 잘 생성되는 반면 긴 문장은 답변이 잘 생성되지 않는 문제점이 발생하였다. 또한 템플릿 형식에 맞는 입력 문장을 사용 하였을 경우 비교적 긴 문장이라 하더라도 답변이 잘 생성되는 것을 보았다. 향후 연구로 답변들을 재순위화 하는 방법 및 긴 문장에 대한 문장 생성을 보완하는 방법에 대해 연구할 계획이다.

감사의 글

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No.2016R1A2B4007732)

참고문헌

[1] 김종환, 장두성, 김학수, “복합 자질 정보를 이용한 통계적 한국어 채팅 문장 생성.”, 인지과학, 제20권, 제4호, pp. 421-437, 2009

[2] 전원표, 송영길, 김학수, “채팅 시스템 구현을 위한 3단계 문장 검색 방법”, 한국마린엔지니어링학회지, 제37권, 제2호, pp. 205-212, 2013.

[3] O. Vinyals and Q. Le, “A neural conversational model.” arXiv preprint arXiv:1506.05869, 2015.

[4] J. Weizenbaum, “ELIZA-A Computer Program For the Study of Natural Language Communication Between Man can Machine”, Communications of the ACM, pp. 36-45, 1996.

[5] J. Hutchens, L. Jason and D. Michael Alder, “Introducing MegaHAL”, Proceedings of the Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning. Association for Computational Linguistics, pp. 271-274, 1998.

[6] Dana Vrajitoru, “Evolutionary Sentence Building for Chatterbots”, Genetic and Evolutionary Computation Conference, 2003.

[7] Dana Vrajitoru and Jacob Ratkiewicz, “Evolutionary Sentence Combination for Chatterbots”, In International Conference on Artificial Intelligence and Applications, pp. 287-292, 2004.

[8] 전원표, 김학수, “클러스터 정보 검색 기법을 이용한 음성 기반 채팅 시스템”, 한국HCI학회 학술대회, pp. 487-489, 2011.

[9] J. Gu, Z. Lu, H. Li and V. O, “Incorporating copying mechanism in sequence-to-sequence learning”, arXiv preprint arXiv:1603.06393, 2016.

[10] R. Kadlec, M. Schmid and J. Kleindienst, “Improved deep learning baselines for ubuntu corpus dialogs.” arXiv preprint arXiv:1510.03753, 2015.

[11] R. Lowe, N. Pow, I. Serban, and J. Pineau, “The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems.” arXiv preprint arXiv:1506.08909, 2015.

[12] 양승원, 김재훈, "음성언어 번역 시스템을 위한 새로운 형태소 분석 " 한국음향학회지 제18권 제4호, pp. 17-22. 1999.

[13] 국립국어원, 21세기세종계획, 2012.

[14] DBPedia version 2015-4, Available from World Wide Web: http://http://wiki.dbpedia.org/dbpedia-data-set-2015-04.