소설 데이터 기반 한국어 자연어 생성

박찬희[O1] 박상현[2†]
연세대학교 컴퓨터과학과
channy_12@yonsei.ac.kr, sanghyun@yonsei.ac.kr

Korean Natural Language Generation Based on Literary Text

Chanhee Park[O1] Sanghyun Park[2†]
Dept. of Computer Science, Yonsei University

요 약

본 연구에서는 공개된 소설 데이터를 이용하여 한국어 문장을 생성하는 연구를 진행하였다. 한국어 문장을 생성하는데 있어 적절한 학습 단위를 모색하기 위해 각 문장을 음절, 품사 정보가 포함되지 않은 형태소, 품사 정보가 포함된 형태소, 어절 단위로 전처리하는 과정을 거쳤으며, LSTM 기반의 언어 모델을 사용하여 실험을 진행하였다. 실험 결과 형태소 단위의 모델에 품사 정보를 추가시킴으로써 사람의 표현력과 조금 더 유사한 문장 구성 결과를 얻을 수 있음을 확인하였다.

1. 서 론

자연어 생성 기술은 최근 대두되고 있는 인공 지능 분야에서 대화 시스템, 기계 번역, 음성 인식과 같은 언어 처리의 기본적인 기술이라 할 수 있다. 하지만 영어권에 비해 한국어 처리와 관련된 연구는 상대적으로 부족한 것이 사실이다.

한국어는 음소가 모여 음절이 되고, 음절이 모여 형태소와 단어를 이룬다. 따라서 한국어 텍스트를 처리함에 있어 다양한 단위로 나누어 처리할 필요성이 있다. 또한 문장을 생성함에 있어서는 문장 내의 각 단어를 독립적으로 생성하지 않고 선행하는 단어가 무엇인지에 따라 현재의 단어를 생성할 수 있다. 예를 들어 '나는 밥을' 다음에는 '먹었다'가 자주 등장할 것이고, '나는 물을' 다음에는 '마셨다'가 자주 등장할 가능성이 높다. 이러한 과정을 통계적으로 모델링한 것이 Markov Chain 모델이며, 이전의 단어 몇 개를 볼 것인가에 따라 N-gram 모델을 만들 수 있다.

이러한 기존의 확률 기반 생성 모델과 달리, 최근에는 인공 신경망을 이용한 자연어 생성 연구가 활발히 이루어지고 있다. 기존의 확률 기반 N-gram 모델이 아닌 인공 신경망을 이용한 언어 모델 연구로는 RNN(Recurrent Nueral Network) 기반의 언어 모델을 이용한 문장 생성 연구가 있다.[1] [1]에서는 특히 Multiplicative RNN과 Hessian-Free 최적화를 통해 문자 단위 언어 모델에서의 문장 생성 성능을 향상시킨 바 있다. 또한 음성 인식 분야에서도 RNN 기반의 언어 모델을 이용하여 기존의 N-gram 모델보다 음성 인식 성능을 향상시켰다.[2]

따라서 본 연구에서는 한국어 소설 데이터를 음절, 형태소, 어절 단위로 나누어 LSTM 언어 모델로 학습을 진행한 후, 문장의 첫 시작 단어가 입력되면 해당 단어를 시작으로 하는 완전한 문장을 생성하는 실험을 진행하고 그 결과를 비교하였다.

※ 이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (IITP-2017-0-00477, (SW스타랩) IoT 환경을 위한 고성능 플래시 메모리 스토리지 기반 인메모리 분산 DBMS 연구개발)
† 교신 저자: sanghyun@yonsei.ac.kr

2. 데이터

본 연구에서는 국립국어원 언어정보나눔터[3]에 공개되어 있는 100여개의 소설 데이터를 이용하여 학습을 진행하였다. 소설 데이터를 통해 학습에 사용된 문장은 총 66770개 이며, 특수 문자나 괄호, 한글이 아닌 문자들은 모두 제거한 후 학습이 이루어 질 수 있도록 적당한 길이의 문장을 추출하였다.

이후 각 문장을 음절 단위, 품사 정보가 포함되지 않은 형태소 단위, 품사 정보가 포함된 형태소 단위, 어절 단위로 나누는 전처리 과정을 거쳤으며, 형태소 단위로 나누기 위해 Konlpy의 Twitter 형태소 분석기[4]를 사용하였다. 각 단위별로 학습에 사용된 어휘의 개수는 각각 1814, 36460, 37343, 135090개 이다.

각 단위별로 전처리 과정을 거친 학습 데이터의 예시는 <표 1>과 같다.

<표 1> 학습 데이터 예시

Unit	Example
Character	날카로운 휘파람소리 같은 것도 그 나무가 내는 소리였다. <Eos>
Morpheme	날카로운 휘파람 소리 같은 것 도 그 나무 가 내는 소리 였 다 . <Eos>
Morpheme +POS	날카로운/Adjective 휘파람/Noun 소리/Noun 같은/Adjective 것/Noun 도/Josa 그/Noun 나무/Noun 가/Josa 내는/Verb 소리/Noun 였/Verb 다/Eomi ./Punctuation <Eos>/Eos
Word	날카로운 휘파람소리 같은 것도 그 나무가 내는 소리였다 . <Eos>

3. 학습 모델

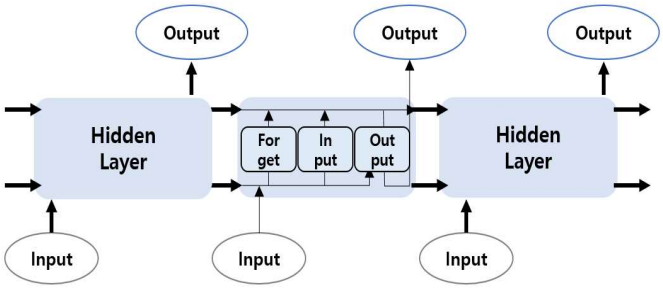
본 연구에서는 실험을 위해 Long Short-Term Memory(LSTM) 기반의 언어 모델(Language Model)을 사용하였다. 언어 모델은 말뭉치 데이터로부터 학습을 통해 주어진 문장에서 이전 단어들에 대해 다음 단어가 나올 확률을 계산하는 모형으로, 기계 번역에 대한 출력이나 문장 생성에 응용된다. 최근에는 언어 모델을 보다 효과적으로 학습하기 위해 LSTM과 같은 딥러닝 알고리즘이 적용되고 있다.

LSTM은 순차적 데이터를 처리하는데 있어 강점을 가진 RNN의 변형된 구조이며, RNN의 오류역전파(Backpropatation)를 통한 학습 진행시 과거의 정보가 학습과 평가에 반영되지 못한다는 한계를 해결하기 위해 RNN의 구조는 유지하되 은닉층을 조금 더 세분화시킨 신경망이다.[5]

LSTM의 구조는 그림 1과 같이 입력부, 은닉층, 출력부로 구성되며 은닉층의 세부 구조는 정보 저장 공간인 셀 스테이트(Cell State)와 셀 스테이트의 기억 정보를 업데이트 하는 것에 영향을 주는 망각 게이트(Forget Gate)와 입력 게이트(Input Gate), 업데이트된 셀 스테이트의 정보를 여과하여 데이터를 출력하는 출력 게이트(Output Gate)로 구성된다.

새로운 입력이 들어오면 셀 스테이트에서 정보를 얼마나 잊을지에 대한 결정이 망각 게이트에서 일어나고, 이와 유사하게 새로운 입력이 셀 스테이트에 얼마나 반영될지에 대한 결정이 입력 게이트에 의해 일어난다. 출력 게이트는 업데이트된 현재의 셀 스테이트의 정보를 얼마나 출력할 것인지를 결정한다. 이러한 과정을 통해 과거의 정보들을 셀 스테이트에 저장하여 추후 연산에 활용할 수 있게 된다.

따라서 본 연구에서는 어떤 단어 단위 하나의 입력이 주어졌을 때 LSTM을 통해 이후에 나오는 단어들을 예측하는 형태로 문장을 생성하는 과정을 거치게 된다.



(그림 1) Long short-term memory

4. 실험 및 결과

4.1 실험 환경

실험은 Intel i7 CPU 1개와 NVIDIA GeForce GTX 1070 GPU 1개가 장착된 PC에서 Python 언어를 기반으로 한 Tensorflow 프레임워크를 활용하여 진행되었다. 각 모델은 먼저 전처리 단위 별로 Word2Vec[6]을 통해 단어 임베딩(Word Embedding) 과정을 거쳤으며, 학습 모델에 사용된 하이퍼 파라미터는 <표 2>와 같다. 하이퍼 파라미터는 주로 통용되는 하이퍼 파라미터를 사용하였다.

<표 2> 학습 모델에 사용된 하이퍼 파라미터

Hyperparameter	Value
Epoch	65
Layer	4
Hidden Size	300
Batch Size	64
Embedding Size	300
Dropout	0.8

4.2 실험 결과

각 단위별 모델의 학습 소요 시간은 <표 3>에서 볼 수 있듯이 문장의 전처리 단위가 클수록 어휘 사전의 크기가 증가하므로 더 많은 학습 시간이 소요되는 것을 확인할 수 있다.

<표 3> 학습 소요 시간

Unit	Training Time (in hours)
Character	3.73
Morpheme	17.05
Morpheme+POS	17.43
Word	38.03

<표 4>에서는 각 단위별 모델의 문장 샘플링 결과의 예시를 나타내고 있다. 음절 단위와 형태소 단위의 모델에서는 '나'가 문장의 시작으로 주어진 경우, 어절 단위의 모델에서는 '나는'이 문장의 시작으로 주어진 경우로 샘플링을 진행하였다.

샘플링 결과를 통해 형태소 단위의 모델에서는 적절한 조사가 연결되는 것과 같이 단어 안의 형태소 구조를 잘 포착하였다. 또한 문장 내에서 단어 간의 연관성도 다른 단위의 모델보다 어느 정도 잘 연결되는 것을 확인할 수 있다.

객관적인 성능을 검증하기 위해서는 성능 평가 척도로 주로 기계 번역 품질의 성능 평가를 위해 사용되는 BLEU 점수를 사용하였다. BLEU 점수는 기계가 번역한

문장을 사람이 번역한 문장을 기준으로 하여 N-gram 방식을 통해 점수를 매긴다. 예를 들어 BLEU-1과 BLEU-2는 각각 1-gram과 2-gram 방식을 통해 평가를 진행한다. 따라서 본 연구에서는 테스트 데이터 문장을 레퍼런스로 각 모델별로 샘플링한 200개의 결과 문장에 대해 BLEU 점수를 성능 평가 척도로 사용하였다.

<표 5>는 BLEU 평과 결과를 나타내며, 평가 결과 전반적으로 형태소 단위의 모델이 음절과 어절 단위의 모델보다 높은 성능을 보였다. 이는 학습에 사용한 문장의 개수가 고정되어 있어 상대적으로 음절 단위의 모델은 학습에 사용된 어휘의 수가 적고, 어절 단위의 모델은 학습에 사용된 어휘의 수가 많았기 때문인 것으로 해석된다.

BLEU-1과 BLEU-2에서는 품사 정보가 포함되지 않은 형태소 단위의 모델이 품사 정보가 포함된 형태소 단위의 모델보다 미세하게 높은 성능을 보였다. 하지만 BLEU-3과 BLEU-4에서는 품사 정보가 포함된 형태소 단위의 모델이 조금 더 좋은 성능을 보였다. 즉, 문장에서 각각 연속된 1개, 2개의 단어와 비교했을 때보다 3개, 4개의 단어와 비교했을 때 조금 더 높은 성능을 보인 것이다.

이는 품사 정보가 포함된 형태소 단위의 모델이 품사 정보를 포함하지 않은 모델보다 문장 구성에 있어서 사람의 표현력과 유사한 문장이라는 것을 의미한다. 따라서 형태소의 품사 정보를 추가함으로써 보다 완성도 높은 문장을 생성할 수 있게 된다.

<표 4> 문장 샘플링 결과 예시

Unit	Sampling Sentence
Character	**나** 내가 그렇지 않고 나는 눈을 붙였다.
Morpheme	**나**는 선생의 말을 듣고 그를 향해 말했다.
Morpheme +POS	**나**는 남편을 위해 그렇게 한 것을 보았다.
Word	**나는** 어머니에게서 분명히 찾고 있었다.

<표 5> BLEU 평가 결과

Unit	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Character	63.1	19.3	9.4	6.1
Morpheme	**86.6**	**75.3**	54.5	36.5
Morpheme +POS	85.3	75.1	**56.4**	**39.1**
Word	60.2	32.7	16.5	10.3

5. 결 론

본 연구에서는 소설 데이터를 통해 학습한 LSTM 언어 모델을 이용하여 한국어 문장 생성 실험을 진행하였다. 문장의 최소 단위를 음절, 형태소, 어절 단위로 나누었을 때의 문장 생성 결과를 비교하여 살펴보았으며, 특히 형태소 단위의 실험에서는 품사 정보가 포함된 문장과 그렇지 않은 문장의 경우로 나누어 실험한 결과 형태소의 품사 정보를 추가함으로써 보다 완성도가 높은 문장을 생성할 수 있음을 확인하였다.

하지만 형태소 단위 모델의 경우 형태소 분석기를 통해 형태소 분석 과정을 거쳐야 하므로 형태소 분석기의 성능에 의존적이라는 한계가 존재한다. 따라서 형태소 분석기의 성능에 의존하지 않으면서도 나은 성능을 보이는 자연어 생성 모델을 연구할 필요성이 있을 것으로 판단된다.

참 고 문 헌

[1] I. Sutskever, et al., "Generating text with recurrent neural networks," In *Proceedings of the 28th International Conference on Machine Learning,* 2011.

[2] Mikolov, T., Karafi´at, M., Burget, L., Cernocky`,J., and Khudanpur, S. “Recurrent Neural Network Based Language Model.” In *Proceedings of Interspeech,* 2010.

[3] 국립국어원 언어정보 나눔터, https://ithub.korean.go.kr/user/main.do, 2018.

[4] Twitter, twitter-korean-text, https://github.com/twitter/twitter-korean-text, 2014.

[5] Hochreiter, S. and Schmidhuber, J. “Long Short-Term Memory.” *Neural Computation*, Vol. 9, No.8, pp.1735-1780, 1997.

[6] Mikolov, T., Chen, K., Corrado, G., & Dean, J. “Efficient estimation of word representations in vector space.”, *arXiv preprint arXiv*:1301.3781.