VAE와 CNN이 결합된 모델을 이용한 한국어 문장 생성과 감성 분석

김건영[O], 이창기

강원대학교
uhi7074@gmail.com, leeck@kangwon.ac.kr

Korean Text Generation and Sentiment Analysis Using Model Combined VAE and CNN

Geon-Yeong Kim[O], Chang-Ki Lee
Kangwon National University

요 약

딥러닝 모델의 성능 향상을 위해 적은 데이터를 증가시킬 수 있는 연구들이 필요하다. 이미지의 경우 회전, 이동, 반전등의 연산으로 쉽게 데이터를 증가시킬 수 있지만 자연어는 그렇지 않다. 그러나 최근 딥러닝 생성 모델의 발전으로 기존 자연어 데이터를 생성 모델을 통해 양을 늘려 실험하는 연구들이 많이 시도되었다. 본 논문에서는 문장 데이터 생성을 위한 VAE, 문장 분류를 위한 CNN이 결합된 모델을 한국어 영화평 데이터에 적용하여 기존 모델보다 0.146% 높은 86.736%의 정확도를 기록하였다.

주제어: 딥러닝, CNN, VAE, 문장 생성, 감성 분석

1. 서론

문장 생성은 자연어 생성의 하위 영역이다. 기계가 사람이 이해 할 수 있는 문장을 생성하는 일들을 총칭하는 자연어 생성은 기계의 단어 선택, 생성된 문장의 문법 구조, 등이 완벽해야 한다. 따라서 자연어 생성은 여전히 어려운 문제이며, 다소 조건을 완화한 문장 생성이 주로 연구되고 있다.

감성 분석은 자연어 문장에 내포된 사람의 주관적인 긍정과 부정, 중립을 구별하는 문제이다. 같은 감성 단어를 가지더라도, 문장의 구조에 따라 긍, 부정이 다를 수 있으므로 문장의 전체 의미를 이해해야 풀 수 있는 어려운 문제이기도 하다.

최근 딥러닝이 자연어 처리 분야에서 획기적인 성능을 보이면서 문장 생성과 감성 분석 문제도 많은 진전이 있었다. 그러나 딥러닝 학습을 위해 많은 데이터가 필요하다는 점은 원활한 연구의 방해 요소로, 적은 데이터를 활용하여 성능을 높일 수 있는 연구가 필요하다.

데이터 늘리기(data augmentation)는 적은 양의 데이터를 여러가지 방법으로 재생산하여 데이터 양을 늘리는 기법으로 많은 양의 데이터가 학습에 필요한 딥러닝 분야에서 많이 쓰이고 있다. 그러나 자연어 처리 분야에서 데이터의 원형, 분류 정보를 유지하며 간단하게 데이터를 늘리기는 어렵다. 최근에는 딥러닝 생성 모델의 발전으로 자연어 데이터 생성에 응용할 수 있는 딥러닝 연구들이 많이 진행되었고 실제로도 성능을 높이는데 기여하였다 [1].

본 논문은 딥러닝을 이용한 문장 생성과 이를 데이터 생성에 응용한 한국어 영화평 감성 분석에 대한 실험을 기술한다. 실험 데이터는 NSMC(Naver Sentiment Movie Corpus)[2]를 사용하였다. 실험을 위한 딥러닝 모델은 자연어 생성을 위한 VAE(Variational Auto-Encoder)와 감성 분석을 위한 CNN(Convolutional Neural Network)이 이어진 형태이며 [3]에 기초한다. 실험 결과 같은 구조를 가지는 감성 분류기를 사용하였음에도 문장 생성으로 가짜 데이터를 학습한 감성 분류기가 더 좋은 성능을 보였다.

2. 관련 연구

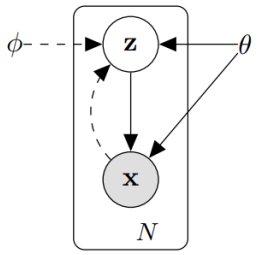

그림 1. VAE 모델

본 논문에서는 한국어 영화평 감성 분석에 대한 실험의 문장 생성에 VAE가 쓰였다.

VAE[4]는 입력 X를 딥러닝 모델을 통해 압축하고, 다시 X로 복구하는 AE(Auto-Encoder)의 종류 중 하나이다. 이는 그림 1과 같이 표현될 수 있으며 X가 은닉 표현 Z로 압축될 때 기존 AE와 다르게 Z가 확률 분포를 따른다. 그림에서 ϕ는 encoder의 파라미터이고 θ는 decoder의 파라미터이다.

VAE를 한국어 문장 생성에 적용한 연구로는 [5]가 있다. 양방향 RNN(Recurrent Neural Network)를 써서 비교적 어순이 자유로운 한국어를 압축하였고 은닉 변수 간의 보간을 통해 문장을 생성하는 연구를 하였다.

딥러닝을 이용한 한국어 감성분석 연구들도 진행되었다. [6]은 본 논문과 같은 데이터인 NSMC를 실험하였다. 단어, 음소 서로 다른 데이터 전처리 방법과, 두 개의 LSTM(Long-Short Term Memory)을 병렬로 쌓아 신경망

을 깊게 만들지 않고 더 좋은 성능을 내는 Parallel Stacked Bidirectional LSTM 구조를 적용하여 정확도 87.75%를 기록하였다.

3. 모델

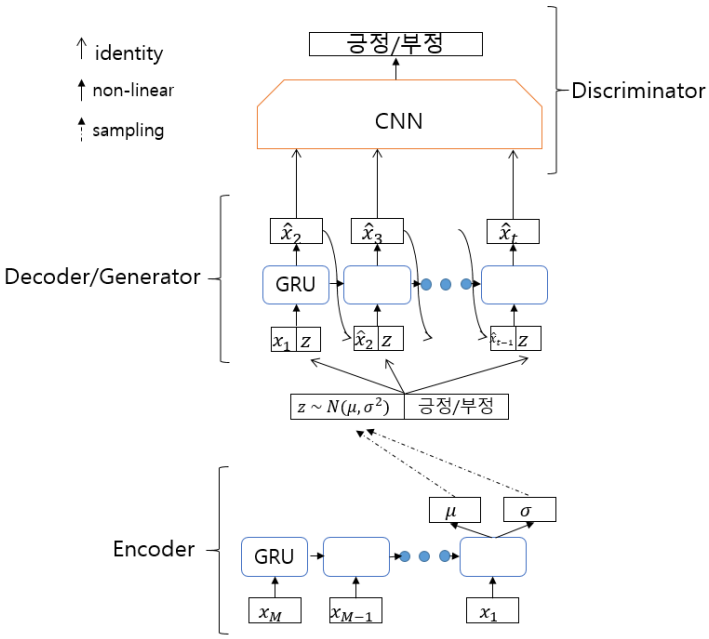

그림 2. 실험에 사용한 모델

실험에는 toward controlled generation of text[4]와 유사한 모델을 사용하였다. 해당 모델은 그림 2와 같이 나타낼 수 있으며 CNN(Convolutional Neural Network)의 구조는 [8]에서 제안한 모델과 같다.

그림에서 x는 입력 문자열로 최대 길이 M을 갖는다. x는 encoder Enc() 에 의해 은닉 표현 $z = \mathrm{Enc}(x) \sim p_{Enc}(z|x)$가 되며, z는 정규분포를 사전 확률로 갖는다. 압축된 z는 문장의 시작을 의미하는 x_1, 원래 문장 x 에 담기는 긍/부정 정보와 함께 이어져 decoder/generator Dec()로 들어가서 생성 문장 $\hat{x}$를 만든다. 생성된 문장 $\hat{x}$는 기존 데이터 x와 함께 분류기인 D()의 학습을 위해 쓰인다. D()는 생성된 $\hat{x}$에 제대로 긍/부정 정보가 담기는지 확인하는 discriminator가 되기도 한다.

그림 2의 모델 학습을 위해 학습은 판별기, 생성기, VAE 학습의 3단계로 나뉜다. 판별기 학습 과정에는 D만 학습되며 기존 데이터와 생성 데이터를 학습한다. 생성기 학습때는 VAE loss를 최적화하면서 생성 문장에 제대로 분류 정보가 담기는지 판별기에 확인을 받는다. 마지막으로 encoder 부분 학습을 위해 VAE loss를 최적화한다.

(1) $\mathcal{L}_s = -\mathbb{E}_x[\log p_D(c|x)] \quad , p_D(x) = D(x)$

(2) $\mathcal{L}_u = -\mathbb{E}_{\hat{x}\sim p_{Dec}(x|z,c)p_{Enc}(x)}[\log p_D(c|\hat{x}) + \beta\mathcal{H}(p_D(\hat{x}))]$

(3) $\min \mathcal{L}_D = \mathcal{L}_s + \lambda_u \mathcal{L}_u$

D 학습을 위한 자세한 수식은 위와 같다. c는 x와 짝지어진 분류 값으로 표 1의 분류가 된다. 수식 (1)은 기존 데이터를 CE(Cross Entropy)로 학습하는 부분이고 수식 (2)는 VAE에 의해 생성된 데이터를 CE로 학습하는 부분이다. $\hat{x}$은 생성된 데이터로 p_{Dec}의 분포에 따라 선택된다. $\mathcal{H}$는 empirical Shannon entropy로 분류기 D가 긍정=0.5 부정=0.5와 같은 애매한 판단을 내리지 않도록 만들어준다. β는 균형을 위한 상수로 0.1의 값을 이용하였다.

수식 (3)은 수식 (1), (2)가 결합된 최종 손실 함수로 기존데이터와 생성된 데이터를 동시에 학습한다. λ_u는 생성된 데이터 학습과 기존 데이터 학습 간의 균형 상수로 0.1을 이용하였다.

(4) $\mathcal{L}_{\mathrm{VAE}} = \mathrm{KL}(p_{Enc}(z|x)||p(z)) - \mathbb{E}_{\mathrm{z}\sim \mathrm{p_{Enc}}(z|x)}[\log p_{Dec}(x|z,c)]$

(5) $\mathcal{L}_{\mathrm{attr,c}} = -\mathbb{E}_{\ddot{x}\sim p_{Dec}(x|z,c)p_{Enc}(x)}[\log p_D(c|\ddot{x})]$

(6) $\ddot{x}_t = \sum p_{Dec}(word|z,c) * Emb(word)$

(7) $\mathcal{L}_{\mathrm{attr,z}} = -\mathbb{E}_{\hat{x}\sim p_{Dec}(x|z,c)p_{Enc}(x)}[\left(Enc(x) - Enc(\hat{x})\right)^2]$

(8) $\mathcal{L}_{\mathrm{Dec}} = \mathcal{L}_{\mathrm{VAE}} + \lambda_c \mathcal{L}_{\mathrm{attr,c}} + \lambda_z \mathcal{L}_{\mathrm{attr,z}}$

Decoder/generator인 Dec학습을 위한 수식은 (4) ~ (7)이다. 먼저 Enc, Dec는 VAE이므로 VAE 손실 함수인 수식 (4)를 최적화한다. 수식에서 왼쪽 KL항은 Kullback-Leibler 발산으로 Enc로 생성된 z가 정규분포를 따르도록 제한하는 역할이며, 오른쪽 항은 CE로 Dec가 x가 압축된 z에서 다시 x를 생성할 수 있도록 한다.

수식 (5)는 Dec로 다시 생성된 $\ddot{x}$이 분류 정보를 제대로 담고 있도록 생성기를 학습하는 역할이다. $\hat{x}$는 샘플링을 통해 하나의 단어를 선택하지만 $\ddot{x}$는 수식(6)을 따라 모든 단어 임베딩의 가중치 합이 된다. $\hat{x}$을 쓸 경우 샘플링한 단어 하나에만 기울기가 전달되지만 $\ddot{x}$을 쓰면 모든 단어에 D의 결과에 따른 기울기를 전달 할 수 있다.

Encoder Enc를 또 하나의 판별기처럼 쓸 수 있는데 그 수식이 (7)이다. $\hat{x}$을 다시 Enc로 압축하여 기존 z값이 다시 나오도록 만든다.

수식 (4) ~ (7)을 합친 수식(8)이 최종적인 Dec를 위한 손실 함수가 된다. λ_c, λ_z 또한 균형 상수로 둘 다 0.1의 값을 이용하였다.

Enc는 수식 (8)로 학습하지 않고 (4)만을 최소화한다.

4. 실험 및 결과

실험 데이터는 NSMC[2]이며 학습셋은 150,000문장, 평가셋은 50,000문장으로 구성되어 있다.

표 1은 데이터의 예제로 상위 2문장은 대상에 대한 긍, 부정이 뚜렷하지만 하위 2문장은 뚜렷하지 않다. 가독성을 위해 원문을 적었지만 실제 실험에는 형태소 단위로 전처리된 데이터를 사용하였다.

표 1. 데이터 예제

분류	문장
긍정	단순하면서 은은한 매력의 영화
부정	아 더빙.. 진짜 짜증나네요 목소리
긍정	오게두어라! 서리한이 굶주렸다!
부정	난 우리영화를 사랑합니다....^^;

표 2. 감성 분석 성능

모델	M	h_dim	z_dim	정확도(%)
CNN(baseline)	64			86.59
CNN w/ VAE	64	200	200	**86.736** (+0.146)

표 2는 감성 분석 성능을 보여주고 있다. CNN(baseline)은 VAE이 없이 기존 데이터만으로 학습한 모델이고 CNN w/ VAE는 VAE를 추가하여 생성 데이터도 같이 학습한 모델이다. M은 문장의 최대 길이로, 길이가 64보다 더 긴 문장은 잘라서 사용하였다. 표에서 h_dim은 GRU의 히든 레이어 유닛의 개수이고 z_dim은 은닉 표현인 z의 차원 수이다. 실험 결과 두 분류기가 같은 구조를 가짐에도 불구하고 생성 데이터를 추가한 경우 0.146%만큼 더 높은 정확도를 얻을 수 있었다.

표 3. 감성 문장 생성 예제

긍정: "꽤/MAG 심오/XR 하/XSA 게/EC 재미/NNG 있/VV 는/ETM 영화/NNG ./SP 만족/NNG 하/XSA 는/ETM 스타일/NNG 이/VCP 었/EP" "지금/MAG 이런/MM 프로그램/NNG 좋아합/VV 니다/EC ,/SP 스토리/NNG 도/JX 뻔하/VA 지만/EC 정말/MAG ./SF ./SF 너무/MAG 멋지/VA 어요/EF" "화려/XR 하/XSA ㄴ/ETM 히어로/NNG 이/VCP ㄴ/ETM 장면/NNG 들/XSN 이/JKS 분장/NNG 이/JKS 완벽/NNG 하/XSA 아/EC 지/VX 었/EP 다/EF ./SF 007/SN 의/JKG 영상미/NNG very/SL good/SL"
부정: "점수/NNG 보고/JKB 주/VV ㄴ다/EC 정준호/NNP 왜/MAG 야/JX 이것/NP ㄹ/JKO 영화/NNG 로/JKB 찍/VV 은/ETM 거/NNB 이/VCP 지/EC ...ㅡ/SE ㅠ/NNG" "왜/MAG 12/SN 세/NNB 관람/NNG 불가/NNG 라고/EC 만들/VV 었/EP 나/EF" "스토리/NNG 가/JKS 엉성/XR 하/XSA 고/EC 음악/NNG 이/JKS 딱히/MAG 많이/MAG 좋/VA 았었/EP 나/EC 싶/VX 었/EP 더니만/EC ./SE ./SF"

표 3은 생성된 데이터의 예제이다. z값은 $\mathrm{N}(0,1)$에서 샘플링하였다. 생성 결과, 대체적으로 긍, 부정 분류 값을 잘 반영하여 생성하였고 자연어생성도 어느정도 자연스러운 결과를 얻었다.

표 4. 같은 z값에 따른 긍, 부정 문장 생성 예제

긍정: "스토리/NNG 면/NNG 에서/JKB 음향/NNG 이/JKS 좋/VA 다/EF ./SF 연기력/NNG 굿/NNG !/SF !/SF !/SF 너무/MAG 재미있/VA 었/EP"
부정: "이것/NP ㄴ/JX 무엇/NP ㄹ/JKO 말/NNG 하/XSV 려는지/EC 는/JX 모르/VV 겠/EP 다/EF ./SF 개뿔/NNG 소설/NNG 도/JX 결국/NNG 7/SN 점/NNB 대/XSN 의/JKG 수준/NNG ㅡ/SW ㅡ/SW 신파/NNG 이/VCP 라는/ETM 것/NNB 이/JKS 너무/MAG 작위/NNG 적/XSN 이/VCP 네/EC ㅡ/SW"

표 4는 같은 z값을 가지지만 긍, 부정을 달리한 문장 생성 예제이다. 생성한 문장에 긍, 부정은 잘 담겨있음을 확인할 수 있다. 그러나 같은 z임에도 불구하고, 같은 대상에 대한 영화평이라 찾기 어려웠다. 이는 VAE가 학습 중에 무작위로 z값을 샘플링하기에 발생하는 문제점으로 보인다.

표 5. VAE로 문장 재생성

원래 문장: 평범/XR 하/XSA ㅁ/ETN 속/NNG 에/JKB 녹아들/VV ㄴ/ETM 평범/XR 하/XSA ㄴ/ETM 일상/NNG ./SP 조금/MAG 밋밋/XR 하/XSA ㄴ/ETM 게/EC 흠/NNG
재 생성된 문장 **긍정**: "저렇/VA 게/EC 재미있/VA 게/EC 보/VV ㄴ/ETM 영화/NNG 이/VCP ㄴ데/EC 그렇/VA 는데/EC 이/MM 평점/NNG 너무/MAG 낮/VA 은/ETM 듯/NNB 하/XSA ㅁ/ETN" **부정**: "이게뭔/NP 데/EC 재미없/VA 게/EC 볼거리/NNG 도/JX 없/VA 음/ETN"

표 5는 문장을 VAE로 압축하고 복원한 결과이다. 문장의 원형이 잘 유지되지 않았는데, VAE의 특징으로 학습 중에 무작위로 샘플링하여 생기는 문제로 생각된다.

5. 결론

본 논문에서는 [4]에서 제안한 모델과 유사한 모델을 한국어 영화평 문장 생성과 감성 분석에 응용하였다. 실험 결과, 같은 구조를 가지는 분류기임에도 불구하고 생성 데이터를 추가하여 더 좋은 성능을 보일 수 있었다.

그러나 문장 생성에 있어 부자연스러운 부분이 존재하

고, 문장의 재생성을 할때 원형이 잘 유지되지 않는 문제점이 있었다. 향후 연구로는 앞서 말한 문제점을 해결할 수 있는 모델을 연구하고자 한다.

감사의 글

이 논문은 2017년 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF2017M3C4A7068188)

참고문헌

[1] Duan, Nan, Tang, Duyu, Chen, Peng, Zhou, Ming, "Question Generation for Question Answering", Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 866-874, 2017.
[2] https://github.com/e9t/nsmc
[3] Diederik P Kingma, Max Welling, "Auto-Encoding Variational Bayes", arXiv:1312.6114, 2013.
[4] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, Eric P. Xing, "Toward Controlled Generation of Text", arXiv:1703.00955, 2017.
[5] Sangchul Hahn, Seokjin Hong and Heeyoul Choi, "한국어 문장 생성을 위한 Variational Recurrent Auto-Encoder 개선 및 활용," Journal of KIISE, Vol. 45, No. 2, pp. 157-164, 2018.
[6] 오영택, 김민태, 김우주, “Parallel Stacked Bidirectional LSTM 모델을 이용한 한국어 영화리뷰 감성 분석”, 한국정보과학회 2018 한국컴퓨터종합학술대회 논문집, pp. 823-825, 2018.
[7] He, Kaiming, et al. “Deep residual learning for image recognition”, arXiv:1512.03385, 2015.
[8] Yoon Kim, "Convolutional Neural Networks for Sentence Classification", Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1746-1751, 2014.