

GAN에서 그래프 탐색을 이용한 유창한 문장 생성

오진영^o, 차정원

창원대학교

psycheojy@cwnu.ac.kr, jcha@cwnu.ac.kr

Fluent Text Generation Using GANs with Graph-search

Jinyoung Oh^o, Jeong-Won Cha

Changwon National University

요약

비지도 학습 모델인 GAN은 학습 데이터 구축이 어려운 여러 분야에 활용되고 있으며, 알려진 문제점들을 보완하기 위해 다양한 모델 결합 및 변형으로 발전하고 있다. 하지만 문장을 생성하는 GAN은 풀어야 할 문제가 많다. 그중에서도 문제가 되는 것은 완성도가 높은 문장을 생성하는데 어려움이 있다는 것이다. 본 논문에서는 단어 그래프를 구성하여 GAN의 학습에 도움을 주며 완성도가 높은 문장을 생성하는 방법을 제안한다.

주제어: 생성적 적대 신경망, GAN, 문장 생성, 그래프 탐색

1. 서론

최근 자연어 생성(Natural Language Generation)에 대한 연구들이 활발하게 진행되고 있다. 자연어 생성은 자연어 이해(Natural Language Understanding)로 얻어진 정보를 다시 문장으로 표현하는 것이다. 딥러닝 이전의 연구에서는 문장의 의미와 구조를 파악하고, 구조를 기반으로 문장을 생성하였다. 딥러닝에서는 의미 구조를 파악하기 위한 자연어 이해 단계를 직접 수행하는 것보다 수많은 문장으로부터 자질을 추출하고 네트워크를 구성하여 학습한다.

그 중 문장 생성을 위하여 많이 연구하고 있는 것이 생성적 적대 신경망(Generative Adversarial Network, GAN)[1]이다. 초기 GAN이 발표된 이후 보완 및 변형된 연구가 폭발적으로 늘어났다. GAN은 생성자(Generator)와 판별자(Discriminator) 두 개의 목적이 충돌하는 신경망을 구성하여 적대적으로 대립하는 모델을 말한다.

GAN을 이용한 생성 연구에서 두드러진 성능을 나타내는 분야는 이미지이다. [2]는 Convolutional 구조를 GAN에 추가한 것으로 성능과 안정성을 높였고 [3]은 이미지의 해상도를 높이는 점진적인 학습을 연구하였다. 이외 [4, 5]는 이미지의 화풍을 변형하는 전이학습(Transfer learning)을 보여주었다.

하지만 GAN은 고질적인 두 가지 문제가 있다. 모드 붕괴(Mode Collapse)와 안정성 문제이다. 모드 붕괴는 실제 입력 데이터의 분포를 모두 커버하지 못하고 손실만 줄이려고 학습하기 때문에 생성자가 판별자의 판별 오류가 많은 모드로만 생성하는 경우를 말한다. 전체적인 네트워크 학습이 되지 않고 취약한 모드로만 생성자가 생성하므로 학습이 되지 않는데 이것을 해결하기 위하여 생성 결과와 입력 데이터의 분포를 변형하기도 하였다. [6]은 학습에 대한 다양한 측면에서 GAN의 문제를 해결하기 위한 여러 방법을 제안하였다.

또한 두 개의 적대적 모델을 학습하기 때문에 네트워크 학습을 위해서는 안정성 유지가 필요하다. [7, 8, 9]는 생성자와 판별자의 학습 순서나 횟수, 비율 등의 조절 등으로 밸런스를 맞추려는 시도 또는 손실 함수를 변경하여 보완하였다.

GAN을 이용한 문장 생성에는 또 다른 문제가 있다. 이미지는 연속 데이터(Continuous data)이고 문장은 이산 데이터(Discrete data)이다. 연속 데이터인 경우 판별자의 손실함수 결과를 생성자에 쉽게 반영할 수 있다. 하지만 이산 데이터인 자연어는 판별 결과를 쉽게 반영할 수 없다.

(a) 나는 진지를 드셨다.

(b) 철수는 영희는 소나무는 운동장에서 물을 마셨다.

예를 들어 생성자가 위 두 문장을 생성하였다고 하자. (a)의 경우 존재의 의미를 주어에 맞춰 낮추는 것이 필요하고 (b)는 문장에 주어에 많으며 상황이 맞지 않다. 이미지에서는 전반적인 색채를 낮추거나, 객체를 조절하지만 문장에서는 그런 조절로 예로 제시된 문장들이 정문이 되지는 않는다. 문장 생성 연구 중 [10]에서는 이러한 문제를 보완하고자 몬테카를로 탐색과 강화 학습(Reinforcement Learning)으로 정책 그라디언트(policy gradient)를 학습하여 생성된 문장들의 리워드를 구하고 손실함수로 구성하는 토큰들의 네트워크를 학습하였다. 그렇지만 몬테카를로 탐색에 의해서 생성된 문장들이 자연스럽지 못하여 정책 그라디언트가 제대로 학습되지 않는 문제가 있었다.

본 논문에서는 이러한 문제를 개선하기 위해 강화 학습을 사용하고 몬테카를로 탐색을 대신하여 그래프 기반 탐색 방법을 제안한다.

2. 관련 연구

문장 생성 분야에서 유창함을 높이기 위해서 많은 시도가 있었다. [11]은 [10]의 탐색 효율을 높이기 위하여 판별자가 추출한 정보를 생성자에게 전달(노출) 함으로써 긴 문장을 생성하면서도 성능을 높였다. [12]는 판별자가 주는 리워드를 배치 사이즈로 나누는 것으로 그래디언트의 값이 소멸되는 것을 보완하고자 하였다. [13]은 긴 거리의 구조를 위하여 관계 메모리를 구성하여 실제에 가까운 텍스트를 생성하는데 기여하였다.

네트워크를 변경하거나 알고리즘을 수정하는 것에 비해 함의(Entailment) 문장을 판단하는 [14]는 GAN의 구조로 구성하되 두 개의 생성자를 두었다. 네트워크 학습만으로 분별하기 어려운 오류를 규칙으로 정의하고, 이를 생성하는 생성자와 네트워크 생성자의 결과를 적절하게 혼합하여 판별자에게 전달한다. 규칙 기반 생성자는 판별자가 해당 오류에 좀 더 강건한 모델이 될 수 있도록 도와주는 것이다. 또 다른 연구로 부족한 학습 문장을 단어 치환과 역-번역을 이용하여 생성 후 실버를 함께 학습함으로써 성능을 높였다[15].

본 논문에서는 GAN에 기반한 문장 생성을 제안한다. 생성자의 학습을 위하여 그래프를 이용하여 학습에 유용한 문장을 제안하고 그 문장들을 판별자가 판단함으로써 리워드를 계산한다. 제안하는 구조는 실제와 가까운 문장을 생성하는 것을 실험으로서 증명한다.

3. 제안 방법

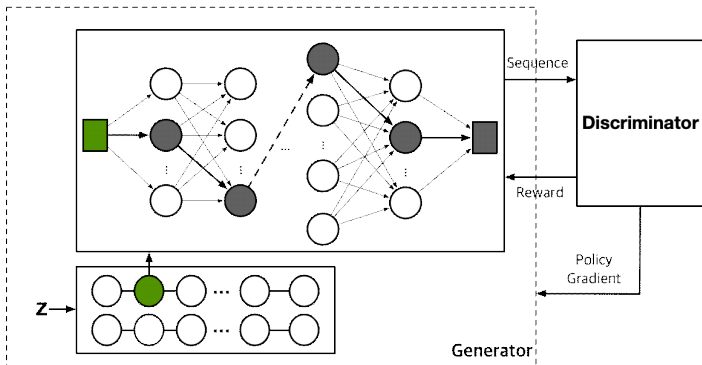


그림 1. 제안 구조

문장 생성 문제는 다음과 같이 정의할 수 있다. 학습 데이터가 주어졌을 때, $X_{1:T} = \{x_1, \dots, x_t, \dots, x_T\}$, $x_t \in X$ 문장을 생성하기 위해 θ -파라미터를 가진 G_θ 모델을 학습하는 것이다. 여기서 X 는 후보 단어집합이다. 우리는 이 문제를 강화학습을 이용하여 해결하려고 한다. 시간 t , 상태 s 일 때, 현재 생성된 문자열이 (x_1, \dots, x_{t-1}) 이고 다음 단어(x_t)를 선택해야 하는 행위 a 이다. 따라서 정책 모델 $G_\theta(x_t|X_{1:t-1})$ 이 추계적(stochastic) 과정이 된다. 추가적으로 G_θ 의 성능을 개선하기 위해 ϕ -파라미터를 가진 D_ϕ 를 학습한다[1].

$D_\phi(X_{1:T})$ 는 생성된 문장 $X_{1:T}$ 이 실제 문장과 얼마나

유사한지를 나타내는 확률이다. D_ϕ 는 학습 데이터 문장과 G_θ 에 의해서 만들어진 문장으로 학습된다. 동시에 생성 모델 G_θ 은 D_ϕ 로부터 받은 최종 예상 문장 보상값에 기초한 정책 그래디언트와 그래프 탐색으로 갱신된다.

3.1 정책 그래디언트(Policy Gradient)

생성 모델(정책) $G_\theta(x_t|X_{1:t-1})$ 의 목표는 시작 상태 s_0 로부터 예상 최종 보상을 최대화 하는 문장을 생성하는 것이다.

$$J(\theta) = \mathbb{E}[R_T|s_0, \theta] = \sum_{x_1 \in X} G_\theta(x_1|s_0) \cdot Q_{D_\phi}^{G_\theta}(s_0, x_1) \quad (1)$$

여기서 R_T 는 완전한 문장을 위한 보상이다. $Q_{D_\phi}^{G_\theta}(s, a)$ 는 시작 상태 s 에서 행위 a 에 의해 생성된 예상 누적 보상이다. 즉, 생성된 문장의 행위-값 함수이다. 우리는 [16]에 따라 강화학습으로 보상값을 계산한다.

$$Q_{D_\phi}^{G_\theta}(a = x_T, s = X_{1:T-1}) = D_\phi(X_{1:T}) \quad (2)$$

우리는 [10]과는 달리 각 상태의 행위-값 평가인 $T-t$ 개의 샘플을 만들기 위해 그래프 탐색을 진행한다.

$$\{X_{1:T}^1, \dots, X_{1:T}^N\} = G^W(X_{1:t}; N) \quad (3)$$

여기서 $X_{1:t}^n = (x_1, \dots, x_t)$ 와 $X_{t+1:T}^n$ 는 현재 상태에 만들어진 샘플이다. 우리는 현재 상태에서 문장 끝까지 N 번을 실행한다. G^W 는 3.2에서 설명할 단어 그래프 탐색을 의미한다. G_θ 가 생성한 $X_{1:t}$ 를 단어 그래프에 입력하고 $X_{t+1:T}$ 이 단어 그래프 G^W 탐색을 통해 완성된다.

$$Q_{D_\phi}^{G_\theta}(a = x_T, s = X_{1:T-1}) = \begin{cases} \frac{1}{N} \sum_{n=1}^N D_\phi(X_{1:T}^n), & \text{if } t < T \\ D_\phi(X_{1:T}) & \text{if } t = T \end{cases} \quad (4)$$

식 4에 의하면 당장의 보상이 없을 때, 시작으로부터 다음 상태 $s' = X_{1:t}$ 까지를 반복으로 정의하여 문장 끝까지 진행한다.

보상함수로서 판별자 D_ϕ 를 사용함으로써 동적으로 생성자 G_θ 를 갱신할 수 있다. 먼저 좀 더 생생한 생성 문장이 만들어지면 우리는 판별자를 다음과 같이 재학습한다.

$$\min_{\phi} - \mathbb{E}_{X \sim p_{data}} [\log D_\phi(X)] - \mathbb{E}_{X \sim D_\phi} [\log(1 - D_\phi(X))] \quad (5)$$

3.2 단어 그래프 구성과 탐색

G^W 는 학습 말뭉치로부터 생성된 단어 그래프이다. 말뭉치 문장 내 토큰들의 관계를 학습하여 구성한다. 본 논문에서는 토큰을 어절로 정의하였고 다른 의미의 문맥에서 같은 어휘로 표현되는 것을 구분하기 위하여 형태소 분석 결과로 구성하였다. 그리고 각 문장의 마지막은 'NULL'을 추가하여 종료를 확인할 수 있도록 하였다.

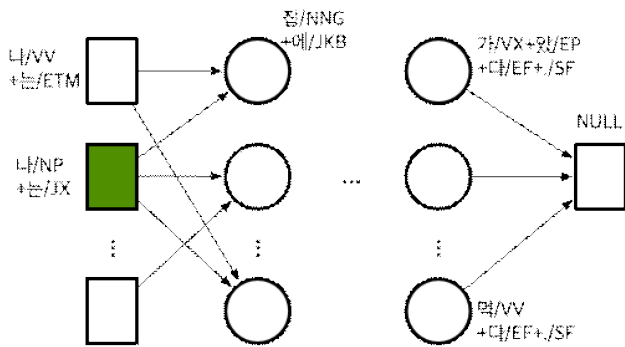


그림 2. 단어 그래프 예

딥러닝 방법에서 그래프를 임베딩하여 네트워크에 활용하거나 네트워크를 그래프로 표현하는 연구[17, 18]는 있지만 문장 생성에서 그래프 탐색 결과를 이용한 연구는 없었다.

본 논문의 단어 그래프 구성은 말뭉치에서 나타나는 단어(중복 제거)가 그래프의 노드로 표현된다. 그리고 문장에서 인접한 단어들 간에 관계가 간선(edge)으로 생성된다. 간선은 가중치를 가지는데, 가중치는 말뭉치에서 많이 나타날수록, 인접한 단어의 수가 적을수록 높아진다.

그림 2는 그래프 구조의 예이다. 예를 들어 ‘나는(나는/NP+는/JX)’ 단어로 시작하고 인접한 단어 중 ‘집에(집/NNG+에/JKB)’가 존재하고, 문장의 마지막 NULL이전에는 ‘갔다(가/VX+았/EP+다/EF+./SF)’ 또는 ‘먹다(먹/VV+다/EF+./SF)’로 구성될 수 있는 것이다. 그래프 탐색은 단어 그래프를 이용하여 ‘나는(나는/NP+는/JX)’으로 시작하는 토큰에 대해 문장 마지막인 ‘NULL’까지의 경로 중 최고의 관계 값을 가지는 경로를 찾는 것이 목표이고 목표에 해당하는 경로(path)가 문장이다. 하지만 관계 값이 최고가 되는 경로만 찾으면 기준 단어에서 다음 노드를 선택시 최고의 관계 값을 가지는 노드만으로 이동하므로 이를 막기 위하여 식 5와 같이 감쇠인자(damping factor)를 추가한 식을 이용한다.

$$P(x_i | x_1, \dots, x_{i-1}) = d \times P(x_i | x_{i-1}) + (1-d) \times \text{Rand}(0,1) \quad (5)$$

식 5에서 감쇠인자는 0.85로 설정한다. $\text{Rand}(0,1)$ 은 0과 1 사이의 임의의 값을 생성한다. 이 감쇠인자는 현재 노드에서 관계 중 가장 높은 값을 선택할 비율과 그 외

관계로 이동할 비율을 정의한 것이다. 이를 이용하여 단계 그래프에서 다양한 문장을 생성할 수 있다.

4. 실험

본 논문에서는 완성도 높은 문장 생성하는 제안방법 검증을 위하여 세종말뭉치¹⁾ 중 형태소 분석된 10,000 문장을 사용하였다. 이 문장을 몬테카를로 탐색 방법으로 생성한 결과와 본 논문에서 제안한 결과를 비교하였다.

4.1 몬테카를로 탐색 문장 생성

[10]에서 제안한 몬테카를로 탐색을 통하여 문장의 후보를 생성하고 관별자로부터 정책 그라디언트 학습하는 모델을 말한다. 표 1에서 보인 결과는 전체 150세대 학습 중 BLEU 4의 성능이 가장 높은 세대를 나타낸 것이다. 평가 시에는 학습한 10,000문장이 참조문장(reference sentence)이 된다.

4.2 그래프 탐색을 이용한 문장 생성

본 논문에서 제안한 구조로 몬테카를로 탐색 대신 단어 그래프 탐색을 이용하여 생성자의 문장을 학습하는 구조를 말한다. 실험의 환경은 탐색을 제외한 나머지 설정을 4.1과 동일하게 하였다.

표 1의 결과는 150세대 중 BLEU 4의 성능이 가장 높은 세대의 결과이다. 몬테카를로 탐색 결과와 비교하여 성능이 높아졌음을 확인할 수 있다.

4.3 실험 결과

표 1. 문장 생성 결과

생성 문장	몬테카를로 탐색	그래프 탐색
세대	90	100
BLEU 2	0.3758	0.6092
BLEU 3	0.2523	0.3231
BLEU 4	0.1432	0.2047

1) 21세기 세종계획 말뭉치 <https://ithub.korean.go.kr>

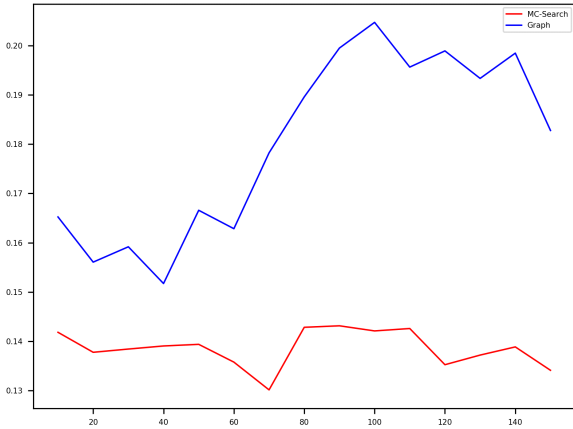


그림 3. 각 세대 BLEU-4 성능 추이

그림 3은 생성된 각 문장의 성능을 BLEU-4로 측정하여 비교하였다. 본 논문에서 제안한 결과(파란색)는 기존의 몬테카를로 탐색으로 학습한 GAN의 결과(빨간색)와 비교하여 문장의 결과가 좋아지는 것을 확인할 수 있다.

또한 그림과 같이 비교군에 비하여 성능 향상 폭이 컸으며 학습이 진행됨에 따라 성능이 높아지는 것을 확인하였다. 이는 기존의 방법에서 몬테카를로 탐색 시 유사한 분포를 가지는 문장 사이에서 최고의 값을 가지는 경로를 구하여 학습하기 때문에 세대가 진행되어도 성능의 향상이 많지 않음을 확인할 수 있었다.

표 2. 각 방법의 문장 결과 샘플

단어 그래프 문장	너한테 달렸다. 영순이를... 어떻게 해야 할까. 그저 무심히 그 지폐를 받아 바지 주머니에 쑤시어 넣었다.
몬테카를로 탐색 문장	안되고 이 매각처리 곳일지는 과장되었을 중얼거리더니 장사치들, 가정이라 완수한 여인숙을 중부대 자이라는
제안 방법	다시 나는 말했다. 나는 권상무를 위해 찢기 시작했다. 어미는 집을 나갔다. 그 나라 눈이 불과 발견되었다.

표 2는 각 방법으로 생성된 문장의 예이다. 실험은 어절을 기준으로 하였고, 형태소 분석 결과를 학습한다. 예는 분석 결과를 원형으로 복원하였다. 제안 방법은 기존의 몬테카를로 탐색 문장에 비해 문장이 완결되며 자연스러운 문장이 생성됨을 확인할 수 있다.

5. 결론

본 논문에서는 완성도 높은 문장 생성을 위하여 단어 그래프로 GAN의 학습 능력을 높이는 모델을 제안하였다. 본 구조를 통하여 생성기의 성능이 높아짐을 실험을 통하여 검증하였고 보다 자연스러운 문장을 확인하였다.

향후에는 그래프를 사전 학습 이외에는 학습되지 않고 참조로만 사용하는 것을 변경하여 직접적인 영향을 미치는 방법과 품질 좋은 문장 생성을 위하여 같이 학습하는 모델로 보완하려 한다.

Acknowledgement

이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2018-0-00247, GANs를 이용한 딥러닝용 학습데이터 자가 증식 기술 및 유효성 검증 기술 개발)

참고문헌

- [1] I. Goodfellow et al., “Generative adversarial nets”, in NIPS, pp. 2672-2680, 2014.
- [2] Alec Radford and Luke Metz et al., “Unsupervised representation learning with deep convolutional generative adversarial networks”, in ICLR, 2016.
- [3] Tero Karras et al., “Progressive Growing of GANs for improved Quality, Stability, and Variation”, in ICLR, 2018.
- [4] Arslan Sanakoyeu et al., “A style-Aware Content Loss for Real-time HD Style Transfer”, in ECCV, 2018.
- [5] Yaxing Wang et al. “Transferring GANs: generating images from limited data”, in ECCV, 2018. G^W
- [6] Tim Salimans et al., “Improved Techniques for Training GANs,” in NIPS, pp. 2234-2242, 2016.
- [7] Xudong Mao et al., “Least Squares Generative Adversarial Networks”, in ICCV, 2017.
- [8] Martin Arjovsky et al., “Wasserstein GAN”, in International Conference on Machine Learning, 2017.
- [9] Junbo Zhao et al., “Energy-based Generative Adversarial Network”, in ICLR, 2017.
- [10] Lantao Yu et al., “SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient”, in AAAI, 2017.
- [11] Jiaxian Guo et al., “Long Text Generation via Adversarial Training with Leaked Information”, in AAAI, 2018.
- [12] Tong Che et al., “Maximum-Likelihood Augmented Discrete Generative Adversarial Networks”, arXiv:1702.07983, 2017.
- [13] Weili Nie et al., “RelGAN: Relational Generative Adversarial Networks for Text Generation”, in ICLR, 2019.
- [14] Dongyeop Kang et al., “AdvEntuRe: Adversarial Training for Textual Entailment with Knowledge-Guided Examples”, in ACL, 2018.
- [15] Yuan Zhang et al., “PAWS: Paraphrase Adversaries from Word Scrambling”, in NAACL,

2019.

- [16] Williams, R. J., “Simple statistical gradient-following algorithms for connectionist reinforcement learning” , Machine learning, pp. 229-256, 1992.
- [17] RiK Koncel-Kedziorski., “Text Generation from Knowledge Graphs with Graph Transformers” , in NAACL, 2019.
- [18] Linfeng Song., “A Graph-to-Sequence Model for AMR-to-Text Generation” , in ACL, 2018.