Variational Autoencoder기반 의미 보존 자연어 데이터 증강 기법

최주환[1], 이준호[2], 진교훈[3], 장예훈[3], 장수진[3], *김영빈[3]
[1]중앙대학교 전자전기공학부
[2]중앙대학교 AI학과
[3]중앙대학교 첨단영상대학원
e-mail: gold5230@cau.ac.kr, jhjo32@cau.ac.kr, fhzh123@cau.ac.kr, jangyh0420@cau.ac.kr, sujin0110@cau.ac.kr, ybkim85@cau.ac.kr

Semantic Preservation Natural Language Data Augmentation via Variational Autoencoder

Juhwan Choi[1], Junho Lee[2], Kyohoon Jin[3], Yehoon Jang[3], Soojin Jang[3], and *Youngbin Kim[3]
[1]School of Electrical and Electronics Engineering, Chung-Ang University
[2]Department of Artificial Intelligence, Chung-Ang University
[3]Graduate School of Advanced Imaging Science, Multimedia & Film, Chung-Ang University

Abstract

Recently, the natural language processing field has emerged by showing excellent performance in various subfields through the introduction of deep learning methods. This deep learning method learns through a large amount of data, and model performance is affected by the quantity and quality of the given data. However, hiring people to collect and classify data to improve a model's performance is time consuming and expensive. Accordingly, various methods have been proposed to increase the performance of deep learning models by augmenting data, even when the data are insufficient.

I. 서론

이미지, 음성 등 다양한 종류의 데이터를 처리하는데 있어서 딥러닝 기법을 도입함으로써 높은 폭의 성능 향상을 이룰 수 있었다. 이러한 딥러닝 기법의 발전은 자연어 데이터를 처리하는데 있어서도 마찬가지이다. 순환 신경망 계열 모델들부터 Transformer 아키텍처까지 자연어 데이터에 딥러닝을 접목하고자 하는 시도들은 꾸준하게 연구되었다.

이러한 딥러닝 기술들은 데이터 기반의 학습방식을 사용하기 때문에 데이터의 양과 질이 큰 영향을 미친다. 학습에 사용할 수 있는 데이터가 부족한 경우 모델은 주어진 데이터에만 최적화되고, 결과적으로 학습 시 보지 못한 새로운 데이터에 대해서는 좋은 성능을 보여주지 못하는 과적합 현상이 발생하게 된다. 이를 해결하고자 다양한 방식의 데이터 증강 기법이 고안되었다. 데이터 증강이란 보유하고 있는 학습용 데이터의 양을 늘리는 작업을 의미하며, 데이터 증강의 목적은 학습에 사용할 수 있는 데이터를 추가적으로 확보함으로써 모델의 일반화 성능을 높이고 최종적으로 과적합을 방지하는데 있다.

이미지 처리 분야에서는 주로 간단하고 직관적인 방식을 통해서 데이터 증강을 진행하였다. 주어진 이미지의 일부만을 잘라 사용하거나 이미지를 좌우로 반전시켜서, 또는 이미지를 회전시켜서 추가적인 데이터로서 사용할 수 있고, 이러한 방식을 통해서

성능을 증대시킬 수 있음이 증명되었다 [1]. 자연어 처리 분야에서도 이와 같이 직관적인 방법, 예를 들어 주어진 문장의 단어 일부를 삭제하거나 무작위의 다른 단어로 교체, 또는 문장 사이에 새로운 단어를 삽입하거나 단어 간의 순서를 바꾸는 방식을 통해 성능 향상을 이룰 수 있음이 밝혀져 있다 [2]. 그러나 이러한 방식의 데이터 증강은 이미지에 수정을 가하는 것과는 달리 문장의 근본적인 의미를 훼손시킬 수 있다는 문제점이 있다. 예를 들어, 고양이의 사진을 좌우로 반전시킨 결과물이 고양이가 아닌 새로운 물체가 되지는 않는다. 그러나 문장에서 핵심적인 역할을 하는 동사를 삭제시키거나 다른 단어로 대치하는 경우 새로운 문장은 본래의 의미와 전혀 다른 뜻을 가지거나, 심지어 의미를 상실할 수도 있다. 결과적으로 원래 문장의 분류된 Label과 다른 의미를 가져 오히려 모델의 성능을 저하시키는 결과를 가져올 수 있다.

한편 대량의 데이터를 통해 사전 학습된 모델을 접목하여 데이터를 증강시키는 기법도 제안되었다 [3]. 그러나 이러한 방식의 경우 사전 학습된 모델의 매개변수가 매우 많기 때문에 이를 재학습 시키기 위해서는 많은 비용과 시간이 소요된다. 또한 모델 학습시 다량의 GPU를 필요로 하여 결과적으로 비용 증가로 이어진다.

본 논문에서는 잠재 변수를 활용하여 자연어 데이터의 근본적인 의미를 훼손시키지 않으면서도 효율적으로 데이터를 증강할 수 있는 기법을 제안한다. 본 논문에서는 인코더를 통해 학습된 특징을 Variational Autoencoder [4]에 넣어 자연어의 근본적인 의미를 학습하고 해당 특징을 재구축하여 데이터를 증강한다. 잠재 변수를 통해 근본적인 의미를 훼손시키지 않으면서 사전 학습된 모델보다 상대적으로 간단하게 데이터를 증강시켰다. 실험 결과 우리의 방식을 접목했을 때 자연어 분류 작업에서 최대 5.13%p 상승하는 것을 확인할 수 있었다.

II. 본론

이 장에서는 본 논문에서 제안하는 텍스트 분류 성능 증대를 위한 VAE 기반의 데이터 증강 기법을 소개한다.

2.1 Variational Autoencoder

오토인코더는 N개의 문장으로 구성된 데이터셋 X의 일부인 i번째 데이터 $x^{(i)}$를 입력으로 하여, 인코더 $q_\phi(z|x)$ 를 통해 이에 상응하는 잠재 변수 $z^{(i)}$ 를 생성한다. 이렇게 생성된 잠재 변수 $z^{(i)}$ 를 디코더 $p_\theta(x^{(i)}|z)$ 에 입력으로 주어 원래의 입력 $x^{(i)}$ 를 재구성한다. 이때 ϕ 와 θ 는 각각 인코더와 디코더의 매개변수를 의미한다. 이 매개변수들을 학습시키기 위해서, 아래와 같은 최적화 함수를 최대화시킨다. 즉, 주어진 $x^{(i)}$를 인코딩한 잠재 변수 z에서 다시 $x^{(i)}$가 디코딩될 수 있는 가능성을 최대화시킨다.

$$\mathcal{L}_{AE}(\theta,\phi;x^{(i)}) = \mathbb{E}_{q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)}|z)\right]$$

Variational Autoencoder (VAE)는 위와 같은 오토인코더 구조를 확장시켜, 주어진 입력 데이터 $x^{(i)}$에 대한 잠재 변수 z의 사후 확률 분포 $q(z|x^{(i)})$의 존재를 가정하고, 이를 미리 정해진 사전 확률 분포 $p_\theta(z)$ 와 가까워지도록 학습시킨다. P_z 로는 평균이 0이고, 표준편차가 1인 표준 정규분포가 주로 사용된다. $Q(z|x^{(i)})$ 와 P_z 가 서로 근접하도록 학습시키기 위해서 두 확률 분포 간의 차이를 나타내는 KL 발산을 도입한다.

$$\mathcal{L}_{VAE}(\theta,\phi;x^{(i)}) = -D_{KL}\left(q_\phi(z|x^{(i)})||p_\theta(z)\right) + \mathbb{E}_{q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)}|z)\right] \leq \log p_\theta(x^{(i)})$$

이와 같이 새롭게 정의된 최적화 함수는 디코더가 $x^{(i)}$ 를 생성해내는 확률 분포 $p_\theta(x^{(i)})$ 의 하한선에 해당한다. 이를 최대화 시키기 위해서는 새로이 추가된 KL 발산 항의 값을 최대한 줄여야 한다. 즉, 이러한 형태의 함수를 통해 기존의 오토인코더와 같이 잠재 변수 z에서 $x^{(i)}$가 생성될 가능성을 최대화함과 동시에, $x^{(i)}$에 대한 z의 사후 확률 분포를 추정할 수 있도록 인코더와 디코더의 매개변수를 학습시킨다.

기존 오토인코더에서 인코더의 출력값을 그대로 잠재 변수 z로 사용하였던 것 과는 달리, VAE에서는 잠재 변수를 확률 분포에서 추출한다. 그런데 확률 분포에서 값을 추출하는 과정에는 무작위성이 반영되므로 기울기 값이 전달되지 않아, 오차 역전파 기법을 활용할 수 없다. 이러한 문제를 해결하여 VAE 모델을 최적화하는 과정에서 오차 역전파 기법을 사용할 수 있도록 Reparametrization Trick이 사용된다. 이는 인코더의 출력값에서 사후 확률 분포의 평균값과 분산값을 추정하고, 표준 정규분포에서 추출한 무작위 값에 표준편차를 곱하고 평균값을 더한 값을 잠재 변수로 사용하는 방식이다. 이렇게 구한 값은 원래의 사후 확률 분포에서 값을 바로 추출하는 것과 분포 상으로는 다르지 않지만,

모델이 추정하는 값은 분포의 평균값과 분산값이므로 이를 오차 역전파 기법을 통해 학습시킬 수 있다.

2.2 Transformer Architecture

Transformer 아키텍처는 행렬 곱에 기반한 Attention 연산을 통해 병렬적인 데이터 처리를 가능하게 함으로서 기존의 순환 신경망 모델이 가지고 있었던 병렬적 연산이 불가능하다는 문제와 장기 의존성 문제를 해결하여 자연어 처리뿐만 아니라 이미지 처리 분야를 포함하여 딥러닝 분야 전반에서 널리 사용되고 있다 [5]. 각각의 Attention 연산은 병렬적으로 이루어질 수 있기 때문에, Transformer 아키텍처는 병렬 연산에 최적화된 GPU와 함께 사용하는 데 높은 효율을 보인다. 본 논문에서는 Transformer 아키텍처의 인코더와 디코더를 각각 VAE 모델의 인코더 $q_\phi(z|x)$ 와 디코더 $p_\theta(x|z)$ 로 사용한다.

본 논문에서 제안하는 모델은 주어진 데이터 텍스트 x의 임베딩 값을 인코더에 넣어 출력된 값을 VAE를 통해 해당 데이터가 가지는 사후 확률 분포의 평균값과 표준편차를 추정하고, Reparametrization trick을 통해 해당 분포로부터 잠재 변수 z를 추출한다. 이렇게 생성된 잠재 변수 z를 디코더가 원래 문장을 복원하는 과정에서 추가적으로 주입한다. 복원을 위해 잠재 변수 z는 주어진 텍스트의 주요한 특징을 학습하며, 이렇게 추출된 잠재 변수 z를 통해 증강된 데이터는 텍스트의 근본적인 의미를 훼손시키지 않을 수 있다.

Ⅲ. 실험

3.1 실험 환경

Transformer의 Encoder와 Decoder에 각각 8개의 layer를 사용하였으며, Optimizer로 AdamW[6]를 사용하였다. Batch의 크기는 32으로, Learning Rate는 1e−5으로 10 Epoch동안 학습을 진행하였다. 성능을 검증하기 위한 텍스트 분류 모델에는 BERT[7]를 활용하였고, Optimizer로 AdamW를 사용하였다. Batch의 크기는 32으로, Warmup 스케쥴러를 통해 학습률을 조정하였으며 3 Epoch동안 학습을 진행하였다.

3.1 실험 데이터셋

모델의 성능을 검증하기 위한 텍스트 분류 데이터셋으로는 IMDB, Yelp Full, ProsCons, MR을 사용하였다. IMDB 데이터셋은 영화에 대한 리뷰를 수집하여 긍정적, 혹은 부정적인 데이터로 분류한 데이터셋으로 총 50,000개의 문장이 포함되어 있다. Yelp Full 데이터셋은 여러 장소에 대한 평가를 1점에서 5점까지 분류한 데이터셋으로 650,000개의 문장으로 구성되어 있다. ProsCons 데이터셋은 약 40,000개의 제품 평가를 긍정적 혹은 부정적으로 분류한 데이터셋이다. MR 데이터셋은 약 1만개의 영화 리뷰를 긍정적 또는 부정적 문장으로 분류하였다.

3.3 실험 결과

실험 데이터셋에 기본 데이터만으로 학습을 진행한 Baseline과 제안 모델을 통해 Augmentation된 데이터를 추가적으로 사용한 학습 결과, 그리고 기존의 데이터 증강 기법 중 비교 대상으로 EDA 기법을 적용하여 제안 모델과 동일한 양의 증강된 데이터를 추가적으로 사용한 학습 결과를 표 1으로 나타내었다.

모델 / 데이터셋	IMDB	Yelp_5
Baseline	91.95%	65.52%
EDA	90.98% (−0.97%p)	67.92% (+2.40%p)
Proposed Model	94.39% (+2.44%p)	70.65% (+5.13%p)
모델 / 데이터셋	ProsCons	MR
Baseline	93.65%	84.05%
EDA	94.21% (+0.56%p)	84.18% (+0.13%p)
Proposed Model	95.16% (+1.51%p)	84.27% (+0.22%p)

표 1 데이터 증강 기법에 따른 성능 비교

실험 결과, IMDB 데이터셋에서 제안 모델을 통해서 추가적인 데이터를 제공했을 때, 원래 데이터만 사용하여 텍스트 분류 모델을 학습시켰을때와 비교해 2.44%p의 성능 향상, Yelp Full 데이터셋에서 5.13%p의 성능 향상, ProsCons 데이터셋에서 1.51%p의 성능 향상, MR 데이터셋에서 0.22%p의

성능 향상을 확인할 수 있었다. 성능 향상의 폭은 각 데이터셋마다 서로 다르게 나타나는데, 주어진 데이터의 양이 더 많은 데이터셋에서 성능 향상 폭이 더욱 큼을 확인할 수 있었다. 이는 VAE 모델이 주어진 데이터의 분포를 추정하여 새로운 데이터를 생성하는 만큼, 기존에 보유한 데이터가 많을 수록 더욱 효과적으로 데이터를 증강시킬 수 있음을 확인하였다. 한편, 비교군으로 설정한 EDA 기법을 통한 데이터 증강의 경우 제안 모델에 비해서 성능 향상의 폭이 작거나, IMDB 데이터셋의 경우 오히려 데이터 증강 이전보다 성능이 떨어짐을 확인할 수 있었다. 이는 EDA 방식을 통한 데이터 증강이 효과적이지 않을 수 있음을 보여준다.

Original Data	I **loved** this movie since I was **7** and I saw it on the opening day. It was so touching and beautiful. I strongly recommend seeing **for all**. It's a movie to watch with your family by far.
Augmented Data w/ EDA	I this movie since I was and I saw it on the opening day. It was so touching and beautiful. I strongly recommend seeing **disastor**. It's some movie to watch with your family by far.
Augmented Data w/ Proposed Model	I **loved** this movie since I was **9** and I saw it on the opening day. It was so touching and beautiful. I recommend seeing **for all**. It's a movie to watch with your family by far.

표 2 데이터 증강 기법에 따른 생성된 데이터 비교

표 2를 통해 자연어 데이터 증강에서 많이 사용되는 EDA와 본 논문에서 제안하는 방식이 만들어낸 데이터를 비교해보았다. EDA에서는 loved라는 단어가 지워졌고 disastor라는 단어가 삽입되었다. 원문은 해당 영화를 칭찬하는 내용이었지만 단순히 한 단어가 사라지고, 한 단어가 대체되었음에도 근본적인 의미가 훼손된 것을 확인할 수 있다. 그에 반해 본 논문에서 제안하는 방식은 상대적으로 덜 중요한 나이와 관련된 내용만 틀렸을 뿐 근본적인 의미가 잘 보존된 데이터를 만들어낸 것을 확인할 수 있다.

Ⅳ. 결론 및 향후 연구 방향

본 논문에서는 텍스트 분류 작업의 성능 향상을 위해 VAE 기반의 데이터 증강 기법을 제안하고, 이러한 기법을 적용하여 새로이 생성된 데이터를 포함해 모델을 학습시켰을 때의 결과를 기존 방식과 비교하여 제안한 방법을 통해서 텍스트 분류 작업에서 최대 5.13%p의 성능 향상을 이뤄낼 수 있음을 보여주었다. 향후에는 제안된 데이터 증강 기법을 기계 번역 등 텍스트 분류 이외의 다양한 작업에 적용할 수 있는 지의 여부와 현재의 모델 구조를 개선할 수 있는 방안에 대해 추가적으로 연구하고자 한다. 이를 위해, 원래 데이터의 Label 정보를 Augmentation 과정에서 주입함으로써 Label을 더욱 잘 유지하며 문장을 생성하는 모델에 대해서 연구해보고자 한다.

Acknowledgements

이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원(NRF-2022R1C1C1008534)과 정보통신기획평가원의 지원(2021-0-01341, 인공지능대학원지원(중앙대학교))의 지원을 받아 수행된 연구임.

참고문헌

[1] Luke Taylor and Geoff S. Nitschke. "Improving Deep Learning with Generic Data Augmentation." 2018 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1542-1547. 2018.

[2] Jason Wei, Kai Zou. "EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks." In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 6383-6389, 2019.

[3] Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. Conditional bert contextual augmentation. In International Conference on Computational Science, pp 84–95, 2019.

[4] Diederik P. Kingma, Max Welling. Auto-

Encoding Variational Bayes. In The 2nd International Conference on Learning Representations (ICLR), 2014.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. Attention is all you need. In Advances in neural information processing systems, pp. 5998-6008, 2017.

[6] Ilya Loshchilov, Frank Hutter. Decoupled Weight Decay Regularization. In International Conference on Learning Representations, 2019.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In North American Association for Computational Linguistics (NAACL), pp. 4171-4186, 2019.