BERT를 활용한 재난 관련 트윗 분류 모델의 과적합 방지

김관우*, 서혜민**, 신연순***
동국대학교 컴퓨터공학과

Prevention of overfitting in disaster related tweet classification models using BERT

Kwanwoo Kim*, Hyemin Seo**, Younsoon Shin***
Department of Computer Science and Engineering
Dongguk University
*kw2577@naver.com, **tommy1610@naver.com, ***ysshin@dongguk.edu

Abstract

In times of crisis, many people post-crisis situations on their Twitter. Twitter recognizes disaster situations faster and has more information than TV or Internet news. But, even if the tweets on Twitter include disaster-related words, it may not be an actual disaster-related tweet. BERT can classify these tweets as tweets not related to actual disaster-related tweets. In this paper, we propose a BERT model that shows better performance by applying various regularizations to avoid overfitting. It was confirmed that the performance of the 'Max-norm with Dropout' applied model was the most improved among the proposed models. Based on this, a high-performance disaster tweet classification model can be implemented in the future, and it is expected that it can be used for crisis management in various fields.

I. 서론

스마트폰과 인터넷 기술의 급속한 발전으로 소셜 미디어는 삶에서 뗄 수 없는 부분이 되었다. 때문에 재난이 발생한 상황에서도 많은 사람들이 실제 위기 상황을 자신의 소셜 미디어에 게시한다. 특히 재난의 위기 관리는 응급 구조 서비스에 크게 의존하는데 사람들이 도움 요청을 포함한 재난 관련 정보를 트위터에 많이 기재하고 있다는 연구 결과가 있다.[1] 하지만 트윗에 재난 관련 단어가 포함 되어있어도, 실제 재난과 관련된 트윗이 아닐 수 있기 때문에 이러한 재난 관련 트윗을 정확히 분류할 필요가 있다.

BERT(Bidirectional Encoder Representations from Transformers)는 2018 년 구글이 공개한 인공지능 모델로 현재 NLP 기술에서 큰 성과를 보여주고 있으며[2], BERT 를 이용하여 트윗들을 훈련하면 실제 재난 관련 트윗과 관련되지 않은 트윗으로 분류할 수 있다. 이때 모델의 과적합을 줄이기 위한 방법으로 정규화를 적용할 수 있는데, Dropout, Dropconnect, max-norm 등을 각각 BERT 에 적용하여 성능을 향상시킬 수 있다. 본 논문에서는 BERT 를 활용하여 재난 트윗 분류를 수행한 관련 연구[3]에 다양한 정규화 방법을 추가 적용해 성능을 개선한 재난 트윗 분류 모델을 제안하고자 한다.

II. BERT 모델

BERT 는 사전 훈련 기반 딥러닝 언어 모델이다. 이를 이용해 자연어 처리에 사전 훈련 모델을 적용하여 새로운 모델을 만들어 다시 학습하는 것이 가능하다.

BERT 모델은 임베딩(embedding)을 수행해 pre-

"본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW 중심대학지원사업의 연구결과로 수행되었음"(2016-0-00017)

trained 된 BERT 위에 classification layer 를 추가해 다양한 NLP 를 처리한다. 이를 수행하는데 있어 가장 중요한 두 가지 과정이 pre-training 과 fine-tuning 이다. pre-training 은 언어를 이해하는 것이고, fine-tuning 은 목적에 맞게 레이어를 추가하는 것으로 설명할 수 있다. 아래 그림 1 은 BERT 의 이러한 두 가지 처리 과정을 나타낸다.

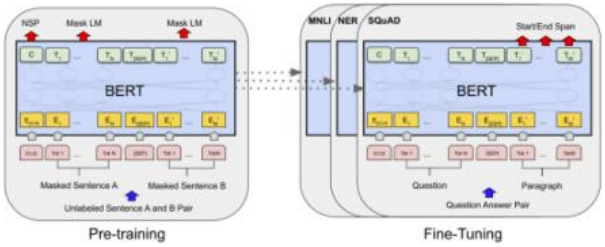

그림 1. BERT 에서의 pre-training 과 fine-tuning 절차

BERT 모델은 Transformer Encoder 블록이 겹겹이 쌓인 형태로 구성되어 있다. base 모델은 인코더 블록이 12 개이고, large 모델은 24 개로 구성되어 있다. 해당 연구에서는 large 모델을 사용하였다.

III. 모델 정규화

BERT 모델은 파라미터가 많은 큰 모델이기 때문에 학습 시, 과적합이 발생할 수 있다. 과적합이 발생하면 학습 데이터를 불필요할 정도로 과하게 학습하여 성능이 떨어지게 된다. 이러한 과적합을 막기 위해 모델 정규화를 적용할 수 있다. 본 절에서는 모델의 성능 개선을 위해 연구에서 사용할 Dropout, Dropconnect, max-norm 라는 세 가지 정규화 방법을 소개한다.

3.1 Dropout

Dropout 은 그림 2 와 같이 모델의 학습 과정에서 신경망 내의 뉴런들을 무작위로 생략을 시키면서 학습을 수행하는 정규화 방법이다. 학습 시에 신경망 내의 특정 뉴런 또는 특정 조합에 너무 의존적이게 되는 것을 방지하며, 매번 무작위로 생략을 시키면서 서로 다른 신경망들을 앙상블하여 사용하는 것과 같은 효과를 가진다.

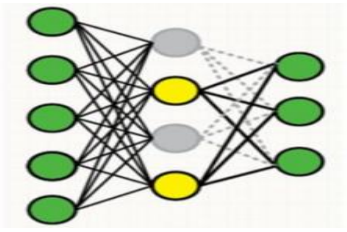

그림 2. Dropout 이 적용된 신경망

3.2 Dropconnect

Dropconnect 는 그림 3 과 같이 모델의 학습 과정에서 뉴런의 connection 을 무작위로 생략하며 학습을 수행하는 정규화 방법이다. connection 을 생략하기 때문에 결과적으로 가중치를 생략하게 되지만 뉴런은 유지된다. Dropout 보다 학습 과정에서 더 다양한 조합으로 모델을 학습하는 것과 같은 결과를 낼 수 있다.

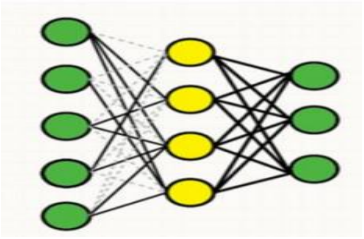

그림 3. Dropconnect 가 적용된 신경망

3.3 Max-norm

Max-norm 은 모델의 매개변수 값인 신경망의 가중치를 사용자 정의 매개변수보다 작은 계수로 갖도록 제한하는 방법이다. Weight 에 upper bound 를 정해 weight 가 과도하게 크게 설정되지 못하게 막아 제한된 조건 안에서 학습 과정을 수행하게 된다.

그림 4 의 수식은 신경망의 뉴런의 연결 가중치 w 의 L2 Norm 인 ||w||2 를 사용자가 정의한 매개변수인 r 보다 작거나 같도록 제한을 하여 훈련 스텝이 끝날 때마다 ||w||2 와 r 의 값을 통해 스케일을 조정한다.

$$\| \mathbf{w} \|_2 \leq r \qquad \mathbf{w} \leftarrow \mathbf{w} \frac{r}{\|w\|_2}$$

그림 4. Max-norm 수식

이러한 Max-norm 의 정규화 방법을 Dropout 과 함께 사용하면 매우 효과적이라는 연구 결과[4]가 존재한다. 아래 그림 5 는 [4]의 실험 결과를 나타내는데

MNIST 데이터를 사용하여 neural networks 를 학습하였을 때의 분류 에러를 보면 여러 정규화 방법 중 Dropout 과 Max-norm 을 같이 사용했을 때의 에러가 매우 낮은 것을 볼 수 있다. 따라서 본 연구에서도 Dropout 과 Max-norm 을 같이 사용하여 정규화를 적용한다.

Method	MNIST Classification error %
L2	1.62
L1 (towards the end of training)	1.60
KL-sparsity	1.55
Max-norm	1.35
Dropout	1.25
Dropout + Max-norm	**1.05**

그림 5. MNIST 데이터 사용 모델의 정규화 에러 비교

IV. 제안하는 재난 트윗 분류 모델

4.1 데이터 셋

본 연구에서는 BERT 를 사용해 재난 트윗 분류를 수행한 관련 연구[2]와 같이 Kaggle 의 데이터 셋을 사용하였다. 총 10,873 개의 데이터가 있고, 해당 데이터들을 이용해 트윗이 특정 재난에 대한 것인지 여부를 추정한다. 구성된 데이터를 살펴보면 전체 데이터 중 57.03%는 재난이 아닌 트윗이었고, 42.97%가 실제 재난과 관련된 트윗이었다.

사용한 데이터 셋은 실제 트윗이기 때문에 여러 노이즈가 존재하므로 BERT 를 이용해 전 처리 임베딩을 해 주어야 한다.

4.2 제안하는 모델

본 연구에서는 BERT 모델을 사용하여 pre-training 을 수행한 모델에 sigmoid 함수를 적용한 dense layer 를 추가하여 재난 관련 tweet 을 분류하는 모델을 정의한다. 이 모델에 Dropout 을 적용한 모델, Dropconnect 를 적용한 모델, Dropout 과 Max-norm 을 같이 적용한 모델을 추가하여, 총 네 가지 모델을 실험 대상으로 하였으며, 각각의 모델의 구조는 그림 6, 7, 8, 9 에서 볼 수 있다. 각각의 붉은 사각형 구역은 제안하는 모델들의 정규화가 진행되는 레이어이다.

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input_word_ids (InputLayer)	[(None, 160)]	0	
input_mask (InputLayer)	[(None, 160)]	0	
segment_ids (InputLayer)	[(None, 160)]	0	
keras_layer (KerasLayer)	[(None, 1024), (None	335141889	input_word_ids[0][0] input_mask[0][0] segment_ids[0][0]
tf.__operators__.getitem (Slici	(None, 1024)	0	keras_layer[0][1]
dense (Dense)	(None, 1)	1025	tf.__operators__.getitem[0][0]

Total params: 335,142,914
Trainable params: 335,142,913
Non-trainable params: 1

그림 6. 재난 분류 모델

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input_word_ids (InputLayer)	[(None, 160)]	0	
input_mask (InputLayer)	[(None, 160)]	0	
segment_ids (InputLayer)	[(None, 160)]	0	
keras_layer (KerasLayer)	[(None, 1024), (None	335141889	input_word_ids[0][0] input_mask[0][0] segment_ids[0][0]
tf.__operators__.getitem (Slici	(None, 1024)	0	keras_layer[0][1]
dense (Dense)	(None, 256)	262400	tf.__operators__.getitem[0][0]
dropout (Dropout)	(None, 256)	0	dense[0][0]
dense_1 (Dense)	(None, 1)	257	dropout[0][0]

Total params: 335,404,546
Trainable params: 335,404,545
Non-trainable params: 1

그림 7. Dropout 을 적용한 재난 분류 모델

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input_word_ids (InputLayer)	[(None, 160)]	0	
input_mask (InputLayer)	[(None, 160)]	0	
segment_ids (InputLayer)	[(None, 160)]	0	
keras_layer (KerasLayer)	[(None, 1024), (None	335141889	input_word_ids[0][0] input_mask[0][0] segment_ids[0][0]
tf.__operators__.getitem (Slici	(None, 1024)	0	keras_layer[0][1]
dense (Dense)	(None, 256)	262400	tf.__operators__.getitem[0][0]
drop_connect_dense (DropConnect	(None, 256)	65792	dense[0][0]
dense_1 (Dense)	(None, 1)	257	drop_connect_dense[0][0]

Total params: 335,470,338
Trainable params: 335,470,337
Non-trainable params: 1

그림 8. Dropconnect 을 적용한 재난 분류 모델

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input_word_ids (InputLayer)	[(None, 160)]	0	
input_mask (InputLayer)	[(None, 160)]	0	
segment_ids (InputLayer)	[(None, 160)]	0	
keras_layer (KerasLayer)	[(None, 1024), (None	335141889	input_word_ids[0][0] input_mask[0][0] segment_ids[0][0]
tf.__operators__.getitem (Slici	(None, 1024)	0	keras_layer[0][1]
dense (Dense)	(None, 256)	262400	tf.__operators__.getitem[0][0]
dropout (Dropout)	(None, 256)	0	dense[0][0]
dense_1 (Dense)	(None, 1)	257	dropout[0][0]

Total params: 335,404,546
Trainable params: 335,404,545
Non-trainable params: 1

그림 9. Dropout + Max-norm 을 적용한 재난 분류 모델

해당 모델들은 모두 batch_size 는 10, epoch 는 8, validation_split 은 0.2 로 설정 후 학습되었다.

4.3 성능 평가

본 연구에서는 모델의 성능을 평가하는 여러 방법들 중 f1 score 를 사용한다. f1 score 는 모델의 성능이 얼마나 효과적인지를 판단할 때 매우 효율적인 지표가 되는데 precision 과 recall 의 조화평균으로 계산된다. 아래의 그림 10 은 f1 score 를 계산하는 수식이다.

$$2 * \frac{Precision * Recall}{Precision + Recall}$$

그림 10. f1 score 계산 수식

정규화가 적용되지 않은 모델과 Dropout 을 적용한 모델, Dropconnect 를 적용한 모델, Dropout 과 Max-norm 을 같이 적용한 모델 총 네 가지의 모델을 학습시킨 뒤, 사용한 test 데이터 셋을 모델에 적용하여 f1 score 를 계산하였다. 사용한 데이터 셋은 총 3,264 개이며, 아래 그림 11 은 해당 데이터 셋의 일부를 보여준다.

id	keyword	location	text
0			Just happened a terrible car crash
2			Heard about #earthquake is different cities, stay safe everyone.
3			there is a forest fire at spot pond, geese are fleeing across the street, I cannot save them all
9			Apocalypse lighting. #Spokane #wildfires

그림 11. 사용한 test 데이터 셋

측정한 모델 별 f1 score 는 아래 그림 12 와 같이 계산되었다.

Method	f1 score
none	0.83144
Dropout	0.83420
Dropconnect	0.83450
Dropout + Max-norm	0.83818

그림 12. 정규화 방법에 따른 모델 별 f1 score

정규화를 적용시키지 않은 모델보다 정규화를 적용시킨 모델의 f1 score 가 모두 더 높게 나왔으며, 특히 Dropout 과 Max-norm 을 같이 적용하여 학습시킨 모델의 f1 score 가 가장 높이 측정된 것을 보여준다.

V. 결론

본 논문에서는 재난 관련 트윗을 BERT 를 사용하여 효과적으로 분류하는 방법을 제안하였고 이 과정에서 네 가지 다른 정규화를 적용해 성능 평가를 진행하였다. 첫 번째는 정규화를 적용하지 않은 방법, 두 번째는 Dropout 을 적용한 방법, 세 번째는 Dropconnect 를 적용한 방법, 네 번째는 Dropout 과 Max-norm 을 함께 적용한 방법이다. 실험 결과에 따르면 f1 score 에 크게 차이가 있지는 않았지만 어떤 정규화도 적용하지 않은 모델의 성능이 가장 낮았으며, 다양한 연구에서 이미 성능이 뛰어나다고 인정된 Dropout 과 Max-norm 을 함께 사용하는 정규화 방법의 경우 가장 성능이 개선된 것을 확인할 수 있다. 이번 연구를 통해서 데이터의 수가 많지 않다는 한계가 존재할 때도, BERT 모델은 좋은 성능을 보여주었고, BERT 에 다양한 정규화 방법을 적용하는 것이 가능하다는 것을 확인하였다. 특히 Dropout 과 Max-norm 을 함께 사용하면 성능의 개선이 가능하다고 판단된다. 향후 이 연구를 기반으로 더 많은 텍스트 데이터에 적용시켜 성능을 향상시킨다면 다양한 분야에서 위기관리에 사용 가능할 것으로 기대한다.

참고문헌

[1] Mohammad Jahanian, Yuxuan Xing, Jiachen Chen, K. K. Ramakrishnan, Hulya Seferoglu, Murat Yuksel, “The Evolving Nature of Disaster Management in the Internet and Social Media Era”, 2018 IEEE International Symposium on Local and Metropolitan Area Networks, 2018

[2] Devlin J, Chang M. W, Lee K, and Toutanova K, “BERT: Pre-training of deep bidirectional transformers for language understanding”, 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 2018

[3] A K Ningsih and A I Hadiana, “Disaster Tweets Classification in Disaster Response using Bidirectional Encoder Representations from Transformer (BERT)”, IOP Conference Series: Materials Science and Engineering, 2021

[4] N Srivastava, “Improving neural networks with dropout”, University of Toronto, 2013