영화평 감성 분석기를 대상으로 한 설명자의 성능 분석

박천용○, 이공주

충남대학교
sdpcy0520@gmail.com, kjoolee@cnu.ac.kr

Performance Analysis of Explainers for Sentiment Classifiers of Movie Reviews

Cheon-Young Park○, Kong Joo Lee
Chungnam National University

요 약

본 연구에서는 블랙박스로 알려진 딥러닝 모델에 설명 근거를 제공할 수 있는 설명자 모델을 적용해 보았다. 영화평 감성 분석을 위해 MLP, CNN으로 구성된 딥러닝 모델과 결정트리의 앙상블인 Gradient Boosting 모델을 이용하여 감성 분류기를 구축하였다. 설명자 모델로는 기울기(gradient)을 기반으로 하는 IG와 레이어 사이의 가중치(weight)을 기반으로 하는 CAM, 그리고 설명가능한 대리 모델을 이용하는 LIME과 입력 속성에 대한 선형모델을 추정하는 SHAP을 사용하였다. 설명자 모델의 특성을 보기 위하여 히트맵과 관련성 높은 N개의 속성을 추출해 보았다. 설명자가 제공하는 기여도에 따라 입력 속성을 제거해 가며 분류기 성능 변화를 측정하는 정량적 평가도 수행하였다. 또한, 사람의 판단 근거와의 일치도를 살펴볼 수 있는 '설명 근거 정확도'라는 새로운 평가 방법을 제안하여 적용해 보았다.

주제어: Explainable AI, Integrated Gradient, CAM, LIME, SHAP, BERT, Sentiment classification

1. 서론

일반적으로 딥러닝 모델은 블랙박스로 알려져 있다. 즉, 딥러닝 모델에 입력이 주어졌을 때 특정 결과가 도출되는 과정을 추론하기 어렵다. 그러나 최근에는 이러한 블랙박스의 내부 추론 과정을 설명하고자 하는 여러 연구들이 진행되고 있다[1]. 이러한 연구 방향 중 하나가 딥러닝 모델의 출력 단계에서 가장 중요하게 사용된 (또는 가장 민감하게 변화된) 입력 속성들을 찾아냄으로써 내부 추론 과정을 설명하는 것이다. 즉 입력 속성값의 출력 결과에 대한 기여도(attribution)을 추정하는 것이다[2].

대표적인 방법으로는 출력 결과에 대한 입력 속성의 기울기(gradient)를 이용하는 것이다. 딥러닝 모델에서 기울기(gradient)는 입력값의 변화에 따른 출력값의 변화를 의미한다. 그렇기 때문에 특정 입력 속성이 출력을 결정하는데 중요한 역할을 할 경우, 그렇지 않은 속성에 비해 그 기울기 값은 커지게 되고, 이를 바탕으로 입력 속성의 기여도(attribution)를 추정하는 것이다. 대표적인 방법으로 Integrated Gradient[3]와 DeepLift[4] 등이 있다.

머신러닝 모델 중, 선형모델이나 결정트리 모델은 근본적으로 설명가능한 모델로 알려져 있다[1]. 그렇기 때문에 블랙박스 모델의 출력 값을 입력 속성의 선형 모델로 근사하면 블랙박스 모델을 설명할 수 있다. 이에 대한 대표적인 방법으로는 LIME[5]과 SHAP[6]이 있다. LIME은 특정 입력에 대해 블랙박스 모델과 가장 유사한 결과를 도출하는 선형 대리 모델을 만든다. 반면 SHAP은 특정 속성이 사용되었을 때와 그렇지 않았을 때의 블랙박스 모델의 출력값의 차이를 통해 선형 모델에서 해당 속성의 계수(coefficient)를 추정한다.

본 연구에서는 영화 감상평을 긍/부정으로 분류하는 딥러닝 모델을 구축하고 딥러닝 모델이 특정 입력에 대해 긍정 또는 부정으로 분류하게 된 주요 근거를 찾아 그 결정을 설명해 보고자 한다. 설명자(explainer) 모델로는 기울기를 사용하여 입력 속성의 기여도를 측정하는 Integrated Gradient (IG)와 가중치(weight)를 사용하는 CAM(Class Activation Mapping)을 사용한다. 또한, 결정트리의 앙상블(ensemble)인 Gradient Boosting 모델로 영화 감상평 분류 모델을 구축하고 특정 입력에 대한 긍/부정 분류결과를 LIME과 SHAP을 이용하여 설명해 본다.

설명자(explainer)가 제시한 근거가 얼마나 유효한지를 평가하는 것은 무척 어려운 작업이다. 가장 흔하게 사용되는 방법은 히트맵이나 기여도가 높게 나타난 속성을 나열하는 식의 정성적 평가이다. 본 연구에서는 설명자의 직접적인 성능 평가를 위하여 새로운 정량적 평가 방법을 제안한다. 사람이 판단한 분류의 근거와 설명자가 제공한 근거 사이의 일치도를 평가할 수 있는 '설명 근거 정확도' (Explanation Component Accuracy)를 제안하고 실제 성능 평가를 수행해 본다. 그 외에도 [7]에서 제안한 중요 속성 제거에 따른 분류기 성능 변화도 측정해 볼 것이다.

2. 관련연구

감성 분석은 텍스트에 나타난 긍정, 부정 등의 의견을 분석하는 자연어 처리의 응용이다. 딥러닝 기반의 감성 분석 모델은 어떠한 근거로 결과를 추론했는지 해석하기 어렵다. 따라서 시스템이 도출한 결과를 해석하고 중요한 자질을 선별하기 위한 연구가 이루어졌다[8].
[8]의 연구에서는 감성 분석을 위한 순환신경망을 구축하고 입력에 대한 민감도를 분석하였다. [8]에서는 입력 문장이 길어질 경우 성능이 하락하는 순환 신경망의 문제를 해결하기 위해서 어텐션 메커니즘을 사용한다. 또한 순환 신경망의 감성 분석 결과와 입력 사이의 연관성을 구하기 위해서 역산 과정에서 발생하는 그래디언트 값으로 입력에 대한 민감도를 계산하였다. 순환 신경망 모델은 네이버 영화 리뷰 데이터 셋을 이용하여 감성 분석을 학습하였다. 학습된 모델에서 어텐션과 민감도가 입력과 출력의 관계를 잘 반영할 수 있는지 입력 단어 삭제에 따른 성능 변화로 비교하였다.
컨볼루션 네트워크는 감성 분석 등 분류를 위한 모델에 사용되는 대표적인 모델이다. [9]의 연구는 컨볼루션 네트워크의 감성 분석 결과를 해석하기 위해 CAM(Class Activation Mapping)방식을 적용하였다. CAM 방식은 컨볼루션 네트워크의 결과인 피쳐맵과 출력 레이어의 가중치의 곱하여 모델이 주목하는 부분을 찾아낸다. CAM 방식은 별도의 모델을 구축하지 않고 학습된 모델에서 결과를 해석할 수 있는 장점을 가지고 있다.

3. 설명자 모델 (Explainer)

3.1 IG (Integrated Gradient)

딥러닝 모델에서는 특정 속성값과 해당 속성의 입력 변화에 대한 출력의 변화 (즉, gradient)을 각 속성의 기여도로 간주할 수 있다[3]. 그런데 속성의 기여도는 일반적으로 해당 속성이 있을 때와 없을 때를 비교하여 추정할 수 있다. 그렇기 때문에 해당 속성에 대한 baseline을 설정하고 baseline으로부터의 차이를 계산하여 속성의 기여도를 계산한다.
Integrated Gradient(IG)[3]는 이와 같은 개념에서 시작하여 입력 x와 baseline이 되는 입력 x' 을 설정하고 x와 x' 사이의 직선을 따라가며 기울기값을 구하고 이를 모두 합산한다.
입력 x의 i-th 속성의 IG는 수식 (1)과 같이 정의하며 이 값을 i-th 속성의 기여도로 간주한다.

$$IG_i(x) = (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha \quad (1)$$

Baseline이 되는 x' 를 설정하는 것이 중요한데, 이미지의 경우에는 black image를 텍스트인 경우에는 zero-embedding을 baseline으로 설정한다. 실제 계산에서는 수식 (2)와 같이 미분이 아닌 합을 이용한다.

$$IG_i\ (x) = (x_i - x_i') \times \sum_{k=1}^{m} \frac{\partial F(x' + k/m \times (x - x'))}{\partial x_i} \times \frac{1}{m} \quad (2)$$

3.2 CAM (Class Activation Mapping)

CAM[10]은 이미지 분류를 다루는 Convolutional Neural Network 모델에 적용한 방법으로 입력 이미지를 특정 클래스로 결정하는데 가장 영향을 많이 발휘한 이미지의 부분을 찾아내고자 하는 목적으로 제안되었다.
CNN 모델의 마지막 Convolutional 레이어에서 max pooling 대신 average pooling을 취하도록 모델을 수정하였고, 이를 출력 레이어와 연결하여 모델을 학습시킨다. 학습이 완료되면 average pooling 레이어와 출력 레이어 사이의 가중치(weight)를 Convolutional 레이어의 feature map과 곱하여 원래 이미지에서 주목(attention) 받는 부분을 찾아낸다. 이렇게 함으로써 CAM은 출력 클래스를 결정하게 된 가장 기여도가 높은 이미지의 부분을 찾아낼 수 있다.
텍스트 분류에서도 CNN 모델을 사용할 수 있기 때문에 동일한 기법을 텍스트의 CNN 모델에 적용하면 출력 클래스에 대하여 입력 텍스트에서 가장 주목받는 부분을 찾아낼 수 있다. 본 연구에서는 [9]의 모델에서 사용한 방법을 적용하여 가장 기여도가 높은 입력 토큰을 찾아낸다.

3.3 LIME (Local Interpretable Model-agnostic Explanations)

LIME[5]은 설명해야 하는 블랙박스 모델과 특정 입력이 있을 때 이를 대신할 수 있는 설명 가능한 대리 모델(surrogate model)을 구축하여 특정 입력의 출력값에 대한 설명을 제공한다. 입력 데이터의 속성값들을 조금씩 수정하여 입력 데이터 주변의 데이터를 생성하고 이를 이용하여 대리 모델을 학습시킨다. 이를 수식화 한 것이 수식 (3)이다.

$$\text{explantion(x)} = \arg\min_{g \in G}\ L(f, g, \pi_x) + \Omega(g) \quad (3)$$

수식 (3)에서 f는 설명해야 하는 블랙박스 모델이고, 입력 x에 대한 설명은 선형모델이나 결정트리와 같은 설명 가능한 모델 집합 G 중에서 함수 L(·)의 값이 최소가 되는 모델 g를 선택하는 것이다. Ω(g)는 모델 g의 복잡도로, 선형모델의 경우 설명에 사용하는 속성의 개수를 사용할 수 있다.

$$L(f, g, \pi_x) = \sum_{z, z' \epsilon Z} \pi_x(z)(f(z) - g(z'))^2 \quad (4)$$

수식 (4)에서 $\pi_x(z)$는 입력 데이터 x와 주변 데이터 z사이의 근사성을 평가하는 함수이다. 즉, 함수 L(·)은 입력 데이터 주변의 여러 데이터들 z에 대해 원래모델 f과 설명모델 g사이의 차이에 x와 z의 근사 정도를 가중치로 곱하여 계산한 값이다. 이 값이 최소가 되는 모델 g를 찾는 것이 LIME의 설명모델을 구축하는 것이다. 데이터 z는 원래 입력 데이터 x의 속성들 일부를 약간씩 수정하여 만든다.

3.4 SHAP (SHapley Additive exPlanation)

SHAP [6]은 설명해야 하는 모델을 모든 속성의 기여도(attribution)에 대한 선형 모델 g로 표현한다. 수식 (5)에서 z_i' 값은 i번째 속성이 발생했는지 그렇지 않은지를 나타내며, Φ_i는 i번째 속성의 기여도 값이다.

$$g(z') = \Phi_0 + \sum_{i=1}^{M} \Phi_i z_i' \quad (5)$$

Shapley 값은 협동적 게임 이론에서 나온 개념으로 공헌도가 다른 게임 플레이어들에게 게임의 수익(gain)을 얼마씩 나누어주어야 하느냐의 문제를 다루기 위해 제안되었다[2]. 이와 유사하게 설명해야 하는 모델의 출력값에 각각의 속성들이 얼마나 공헌을 했는지를 계산하여 각 속성들의 기여도 값을 구한다. 다만, 속성의 개수가 많을 경우, 속성들의 가능한 조합이 너무 많아져 계산량이 늘어나는 문제가 있어서 다른 모델들과 함께 사용한다. 본 연구에서는 Gradient Boosting 트리 모델을 구축한 후, 트리 구축에 사용된 속성을 따라서 SHAP을 구축한다.

$$\Phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(M-|S|-1)!}{M!} [f_x(S \cup \{i\}) - f_x(S)] \quad (6)$$

수식 (6)에서 M은 입력 속성 개수, N은 입력 속성 집합이다. 입력데이터 x에서 속성 i가 포함되었을 때와 그렇지 않은 가능한 모든 경우의 모델 출력값 f()의 차이를 계산하여 i번째 속성의 기여도 Φ_i로 사용한다.

4. 실험 및 평가

4.1 실험 환경

감성 분류를 위한 데이터 집합은 Naver Sentiment Movie Corpus[1]를 사용한다. 이것은 140글자 이하의 짧은 영화평으로 구성된 코퍼스로 총 20만 개의 리뷰를 담고 있다. 이 중 15만개는 학습용으로 5만개는 평가용으로 사용한다. 감성 분류 모델을 학습시키고 평가한 정확도를 표 1에 제시하였다. 본 연구에서 사용한 감성 분류 모델은 (1) Gradient Boosting[11], (2) Multi-Layer Perceptron(MLP), (3~4) Convolutional Neural Networks(CNN)이다. 입력 단어 표현은 모델 (1) GB의 경우 bag-of-word(BOW) 방식이며 tf-idf 값을 사용한다. (2~4)의 경우는 BERT의 출력을 단어 표현으로 사용한다. 본 연구에서 사용한 BERT는 ETRI에서 개발/배포한 한국어 KorBERT 모델[2]로 어절 기반 언어모델이다. 4개의 감성 분류기 모두가 동일한 단어 집합을 사용하도록 KorBERT의 토크나이저로 문장 분리를 수행하였다. MLP 모델은 BERT의 출력 중, ClassToken을 MLP의 입력으로 사용하였으며, (3)과 (4)의 CNN 모델은 BERT의 최종 출력의 hidden layer를 입력으로 사용하였다. (3) CNN-m 모델은 max-pooling을 사용하고 필터크기 (2,3,4)이며 출력 채널은 각각 (32,32,32)개이다. (4) CNN-a 모델은 CAM 해석기를 사용하기 위하여 avg-pooling을 사용하고 필터크기는 (3,4,5)이며 출력 채널은 (512,512,512)개를 사용하였다.

표 1: 감성 분류기 모델의 분류 정확도

MODEL	Word Vector	Accuracy	
		training	testing
(1) Gradient Boosting (GB)	BOW	88.0	81.21
(2) MLP	BERT	99.37	89.97
(3) CNN-m	BERT	99.11	89.80
(4) CNN-a	BERT	99.03	89.37

본 연구에서 사용한 설명자(Explainer) 구현은 LIME[3], SHAP[4], IG[5]을 사용하였으며 CAM은 직접 구현하였다.

3.2 평가 방법

설명자의 평가는 정성적 평가와 정량적 평가를 수행한다. 정성적 평가로는 히트맵(Heat Map)과 가장 관련 높은 N개 단어(Most relevant N-words) 를 살펴볼 것이다. 정량적 평가는 중요 속성 제거에 따른 분류기 성능 변화 측정과 본 논문에서 새롭게 제안하는 설명 근거 정확도(Explanation Component Accuracy)를 살펴본다.

3.2.1 정성적 평가 결과

(1) 히트맵(Heat Map)

그림 1은 긍정인 영화평에 대한 각 설명자-분류기의 토큰별 히트맵이다. 좀더 많은 히트맵을 부록에 포함시켰다.

그림 1에서 보듯이 예제 문장에 대해 LIME과 SHAP은 다른 설명자에 비해 소수의 토큰에 집중된 설명 근거를 제시하고 있다.

부정	IG-MLP	[CLS] 내가_ 극 장 가서_ 본_ 영화 중_ 제 일_ 쓰 레 기_ 같 았다_ [SEP]
	IG-CNN-m	[CLS] 내가_ 극 장 가서_ 본_ 영화 중_ 제 일_ 쓰 레 기_ 같 았다_ [SEP]
	CAM-CNN-a	[CLS] 내가_ 극 장 가서_ 본_ 영화 중_ 제 일_ 쓰 레 기_ 같 았다_ [SEP]
	LIME-GB	내가_ 극 장 가서_ 본_ 영화 중_ 제 일_ 쓰 레 기_ 같 았다_
	SHAP-GB	내가_ 극 장 가서_ 본_ 영화 중_ 제 일_ 쓰 레 기_ 같 았다_

[1] https://github.com/e9t/nsmc
[2] http://aiopen.etri.re.kr/service_dataset.php
[3] https://github.com/marcotcr/lime
[4] https://github.com/slundberg/shap
[5] https://captum.ai/api/integrated_gradients.html

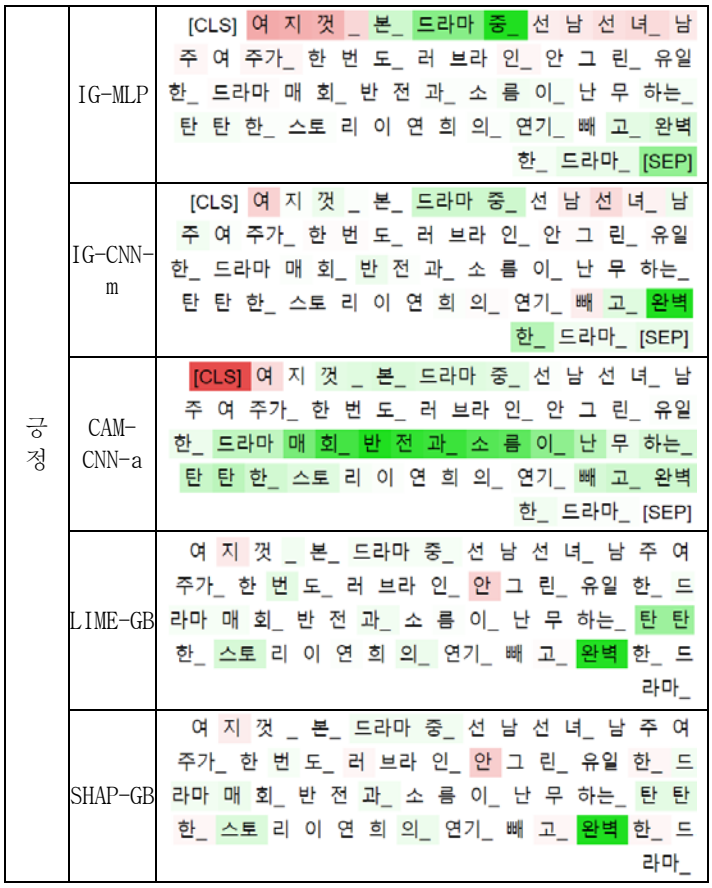

그림 1: 설명자-분류기에 따른 히트맵

(2) 관련성 높은 30개 토큰

관련성 높은 N개의 토큰을 추출하기 위하여 타겟클래스를 설정한 후, 각 문장의 토큰이 타겟클래스에 기여한 기여도 값을 전체 코퍼스에 대해 평균을 구한다.

Naver Sentiment Movie Corpus의 평가 집합 중, 토큰 개수가 50개 이상인 문서 4,000개(긍정문서: 1,963개 부정문서: 2,037개)에 대해서 수행하여 긍정 클래스와 부정 클래스에 대해 가장 높은 점수를 받은 30개의 토큰을 추출하였다. 4,000개 문서에서 10회 이상 발생한 토큰에 대해서만 평가를 실시하였다. 표 2와 3에서 괄호안의 숫자는 5개의 설명자-분류기 결과 중 3개 이상에서 공통으로 발생한 토큰의 개수이다. (3개 이상의 설명자-분류기에서 발생한 토큰은 밑줄로 표시) 긍정 클래스와 관련이 높은 토큰의 경우 IG-CNN-m, LIME-GB, SHAP-GB가 유사한 결과를 보인 반면, CAM-CNN-a는 다소 다른 토큰들이 추출되었다.

부정 클래스와 관련이 높은 토큰의 경우에는 LIME-GB와 SHAP-GB 두 결과는 30개 토큰 중, 25개를 공유하였다. 그러나 다른 설명자-분류기의 경우에는 공통된 토큰이 많지 않았다.

표 2: 설명자-분류기에 따른 관련성 높은 30개 토큰 (긍정)

	IG-MLP (7)	IG-CNN-m (12)	CAM-CNN-a (4)	LIME-GB (13)	SHAP-GB (13)
긍정	**최고_**	**최고**	마음을_	**최고**	**최고**
	최고의_	**최고의_**	롭게_	**최고의_**	**최고의_**
	있어요	**최고_**	나를_	**최고_**	♥
	좋아요	**좋다**	틋	**좋다**	**굿**
	최고	**좋아요**	**가슴**	**굿**	좋다_
	올리	**완벽**	찰	**가슴**	잊
	비판	밌	더욱_	**완벽**	**가슴**
	낮	시즌	낄	좋다_	**최고_**
	시대에_	좋	다시_	♥	**좋다**
	있게_	**낮**	**마음**	밌	삶
	현실을_	완성	픔	ㅎ	^
	대에_	**깊**	뭉	**멋**	**완벽**
	평	좋은_	**깊**	^	**아름**
	실제로_	**있게_**	도록_	잊	ㅎ
	대표	올리	등을_	삶	**멋**
	9	**멋**	우리의_	**아름**	꼭_
	완벽	**아름**	오는_	사랑	**낮**
	마음을_	좋아_	생각을_	**마음**	훌
	하다는_	현실을_	평	**낮**	뜻
	중_	9	사랑을_	꿀	**깊**
	해주고_	명	해주는_	꼭_	짱
	번째	빛	**아름**	좋	행복
	에서의_	중요한_	꽤	훌	**마음**
	꽉	인생_	많은_	짱	사랑
	높은_	롭게_	씁	돋	돋
	10	밋	메시	10	!_
	수한_	짠	충분히_	!_	다시_
	가장_	**굿**	알려	**깊**	감사
	력에_	번째	무너무_	**있게_**	**좋아요**
	표현	선이_	충분	잘_	만에_

표 3: 설명자-분류기에 따른 관련성 높은 30개 토큰 (부정)

	IG-MLP (1)	IG-CNN-m (5)	CAM-CNN-a (3)	LIME-GB (4)	SHAP-GB (4)
부정	드가_	일단_	별_	**낚**	증
	선이_	심각	미국_	**망**	**낚**
	2_	겉	**아무리_**	쓰	쓰
	기는_	처음으로_	아니_	깝	**망**
	과는_	미국_	1_	졸	악
	도는_	고등	지금까지_	없	깝
	진이_	**망**	애니_	증	수준
	못하고_	북한	꼼	수준	만들
	인은_	별_	애	돈	없
	부는_	**아무리_**	0	냐	류_
	예산_	**낚**	최	一	낭
	점이_	초등	대체_	별로_	냐
	사는_	슈퍼	기본	류_	돈
	일은_	1_	왠	똥	똥
	고는_	**별**	돈_	떨어	졸
	만을_	지금까지_	곽	만들	뭐
	남자가_	아니_	아무_	뭐	그냥_
	가가_	나온다_	**예산_**	만_	이하
	제를_	원래_	웬	그냥_	떨어
	씨는_	**예산_**	첫_	낭	**아무리_**
	으로는_	일본_	째_	이하	一
	장이_	퇴	**별**	**별**	**별**
	도가_	협	여기_	최	엉
	호가_	직	협	?_	발
	여자가_	부담	아나	악	만_
	전이_	없었다_	이게_	없는_	딴
	트는_	냉	비행	발	돈_
	비가_	처음_	U	**아무리_**	개
	주가_	기본	하나도_	감독이_	?_
	들로_	우선_	감독은_	;	건_

3.2.2 정량적 평가 결과

(1) 중요 속성 제거에 따른 분류기 성능 변화 측정

설명자가 제공한 각 속성(feature)의 기여도 값이 얼마나 유효한지를 보기 위해 [7]에서 제안한 것과 동일한 실험을 수행하였다.
중요 속성을 제거한 경우에는 분류기의 성능에 많은 차이가 생길 것이고 그렇지 않은 속성을 제거했을 때에는 분류기의 성능에 차이가 적게 생길 것이다. 이러한 점에 착안하여 설명자가 제공한 기여도에 따라 속성을 하나씩 제거해 가면서 분류기의 정확도를 측정해본다.

(실험1) 분류기가 제대로 분류한 문서집합 중에서 50개 이상의 토큰으로 구성된 문서들만 수집한다. 설명자가 제공한 기여도 값의 내림차순으로 30개의 토큰을 하나씩 지워가면서 분류기의 분류 정확도를 측정한다.
(실험2) 분류기가 잘못 분류한 문서집합 중에서 50개 이상의 토큰으로 구성된 문서들만 수집한다. 설명자가 제공한 기여도 값의 오름차순으로 30개의 토큰을 하나씩 지워가면서 분류기의 정확도를 측정한다.

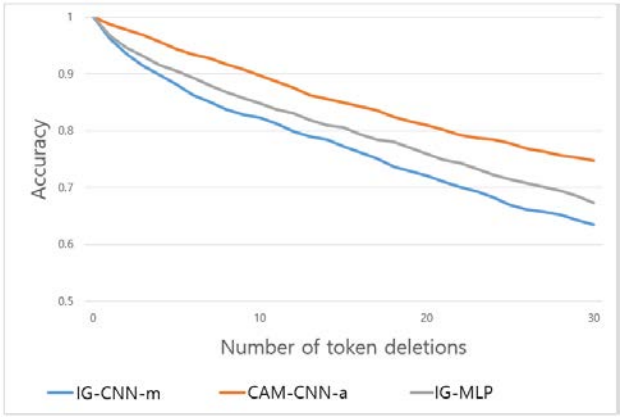

그림 2: 토큰 삭제에 따른 분류기 성능 감소폭 변화 (실험 1)

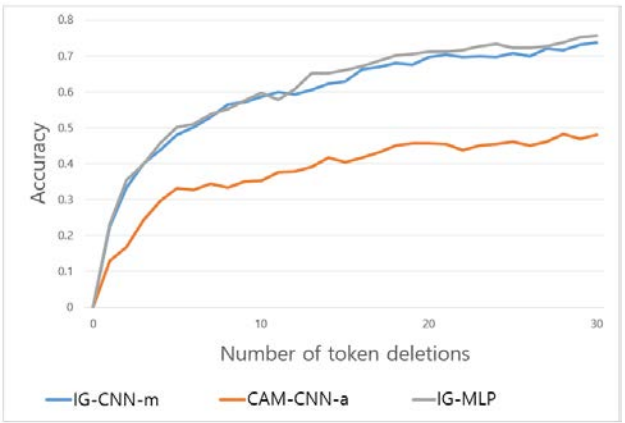

그림 3: 토큰 삭제에 따른 분류기 성능 향상폭 변화 (실험 2)

두 실험의 결과를 보았을 때, IG가 제공하는 속성의 기여도가 CAM이 제공하는 값보다 분류기 성능 변화를 더 많이 유도하였다.

(2) 설명 근거 정확도(Explanatory Component Accuracy)

본 연구에서는 설명자의 성능을 좀더 자세히 보기 위하여 설명 근거 정확도(Explanatory Component Accuracy)라는 평가 방법을 새롭게 제안한다. 이것은 사람의 판단 근거와 설명자의 판단 근거가 얼마나 일치하는지를 명확히 비교할 수 있는 평가 방법이다.
우선 사람에게 각 문서를 해당 클래스로 분류하게 된 근거를 표시하도록 하였다.
(단계1) 설명 근거를 제공한 문서를 토큰으로 분리하여 사람의 설명 근거를 포함하는 토큰들을 선정한다. 예제 1의 경우 총 22개의 토큰으로 구성된 문서로 그 중 5개의 토큰이 사람이 제공한 설명 근거를 포함한다. 표 4는 문장에서 사람의 설명 근거를 포함하는 토큰을 선정한 예시이다.
(단계2) 설명자가 제공한 토큰 중 기여도 순으로 (단계1)에서 추출한 토큰 개수와 동일한 개수의 토큰을 추출하고 (단계1)의 사람이 제공한 설명 근거와 공통된 개수를 계산하여 정확도로 측정한다.

.

표 4: 사람의 설명 근거 선정 예시

예제 1	사람이 제공한 설명 근거	**[발연기]** 도저히 못보겠다 진짜 이렇게 **[연기를 못할]** 거라곤 상상도 못했네
	토큰화 결과	**[발]** **[연기]** 도 저 히 못 보 겠다 진짜 이렇게 **[연기를]** **[못]** **[할]** 거 라 곤 상 상 도 못 했 네 (총 22개 토큰 중 5개 설명 근거)
예제 2	사람이 제공한 설명 근거	여지껏 본 드라마중 선남선녀 남주여주가 한번도 러브라인 안그린 유일한 드라마매회 **[반전]**과 **[소름]**이 난무하는 **[탄탄]**한 스토리이연희의 연기 빼고 **[완벽]**한 드라마
	토큰화 결과	여 지 껏 _ 본_ 드라마 중_ 선 남 선 녀_ 남 주 여 주가_ 한 번 도_ 러 브라 인_ 안 그 린_ 유일 한_ 드 라마 매 회_ **[반]** **[전]** 과_ **[소]** **[름]** 이_ 난 무 하는_ **[탄]** **[탄]** 한_ 스토 리 이 연 희 의_ 연기_ 빼 고_ **[완 벽]** 한_ 드라마_ (총 53개 토큰 중 6개 설명 근거)

평가 집합을 구축하기 위해 사람이 평가집합 중 100문장에 대해 설명 근거를 수동으로 태깅하였다. 각 영화평에 대한 사람의 평가 근거는 최소로 태깅하도록 했으며 최대 5개를 넘지 않도록 하였다.
BOW 형식의 입력을 사용하는 LIME과 SHAP의 경우는 한 문서에 동일한 토큰이 여러 개 발생할 경우에도 모두 하나로 간주한다. BERT를 입력으로 사용하는 IG나 CAM은 같은 토큰이라 하더라도 토큰 위치별로 기여도가 다르게 평가된다. 그러나 LIME이나 SHAP과 동일하게 평가하기 위하여 토큰 위치는 구분하지 않고 토큰 단위로만 평가를 수행하였다. 그렇기 때문에 예제 2의 설명 근거가 7개 아닌 6개로 계산된다. 표 5는 예제 2의 대한 각 설명자-분류기에서 기여도 높은 6개의 토큰을 추출한 결과다.

표 5: 예제 2에 대한 설명자-분류기의 설명 근거 정확도

정답	IG-MLP	IG-CNN-m	CAM-CNN-a	LIME-GB	SHAP-GB
반 전 소 름 탄 완벽	중_ 드라마 본_ **완벽** 고_ 드라마_	**완벽** 한_ 드라마 중_ 고_ 남	**반** **전** 과_ 회_ **소** **름**	**완벽** **탄** 스토 번 의_ _	**완벽** 스토 **탄** 의_ 매 중_
정확도	16.67	16.67	66.67	33.33	33.33

표 6: 설명자-분류기에 따른 설명 근거 정확도

	IG-MLP	IG-CNN-m	CAM-CNN-a	LIME-GB	SHAP-GB
Explanatory Component Accuracy	19.15	21.99	25.59	**36.19**	33.44

표 6은 설명자-분류기에 따른 설명 근거의 정확도이다. 사람의 평가 근거와 가장 유사한 결과를 보이는 것은 LIME이었으며 SHAP이 그 다음의 성능을 보였다.

4. 결론

본 연구에서는 감성 분류 모델에 대하여 설명자(explainer)를 적용하여 각 입력에 대해 분류기의 출력에 대한 설명 근거를 추출해 보았다. 설명자의 성능 분석은 매우 어려운 문제 중 하나이다. 본 연구에서는 정성적 평가와 정량적 평가를 통해 IG, CAM, LIME, SHAP 설명자의 특성과 성능을 분석해 보았다. 또한 사람이 제공한 설명 근거와의 직접적인 비교를 위해 '설명 근거 정확도' 라는 새로운 평가 방법을 제안하여 적용해 보았다.

감사의 글

이 논문 또는 저서는 2019년 대한민국 교육부와 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2019R1F1A1053136)

참고문헌

[1] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, Dino Pedreschi "A Survey of Methods for Explaining Black Box Models", ACM Comput. Surv. 51(5): 93:1-93:42, 2019.

[2] Scott M Lundberg, and Su-In Lee, "A Unified Approach to Interpreting Model Predictions", Advances in Neural Information Processing Systems 30, 4765-4774, 2017.

[3] Mukund Sundararajan, Ankur Taly and Qiqi Yan, "Axiomatic Attribution for Deep Networks", Proceedings of the 34th International Conference on Machine Learning, 3319--3328, 2017.

[4] Avanti Shrikumar, Peyton Greenside. Anshul Kundaje, "Learning Important Features Through Propagating Activation Differences," ICML'17: Proceedings of the 34th International Conference on Machine Learning, Pages 3145–3153, 2017.

[5] Marco Ribeiro, Sameer Singh, and Carlos Guestrin, "Why should I trust You? Explaining the Predictions of Any Classifier," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2016.

[6] Scott M. Lundberg, Gabriel G. Erion and Su-In Lee, "Consistent Individualized Feature Attribution for Tree Ensembles", http://arxiv.org/abs/1802.03888, 2018.

[7] Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2017. "What is relevant in a text document?": An interpretable machine learning approach. PLOS ONE, Vol. 12 , 1—23, 2017.

[8] 배장성, 이창기, "감성분석에서 순환신경망의 예측 설명", 제31회 한글 및 한국어 정보처리 학술대회 논문집, HCLT 2019

[9] Gichang Lee, Jaeyun Jeong, Seungwan Seo, CzangYeob Kim, Pilsung Kang, "Sentiment Classification with Word Attention based on Weakly Supervised Learning with a Convolutional Neural Network," arXiv:1709.09885v2, 2017.

[10] Ramprasaath R. Selvaraju et al., "Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization", http://arxiv.org/abs/1610.02391 2016.

[11] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python", Journal of Machine Learning Research, Vol 12, 2825-2830, 2011.

부록

IG-MLP	[CLS] 발 연기_ 도 저 히_ 못 보 겠 다_ 진짜_ 이렇게_ 연기를_ 못 할 거 라 곤_ 상 상 도_ 못 했 네_ [SEP]
IG-CNN-m	[CLS] 발 연기_ 도 저 히_ 못 보 겠 다_ 진짜_ 이렇게_ 연기를_ 못 할 거 라 곤_ 상 상 도_ 못 했 네_ [SEP]
CAM-CNN-a	[CLS] 발 연기_ 도 저 히_ 못 보 겠 다_ 진짜_ 이렇게_ 연기를_ 못 할 거 라 곤_ 상 상 도_ 못 했 네_ [SEP]
LIME-GB	발 연기_ 도 저 히_ 못 보 겠다_ 진 짜_ 이렇게_ 연기를_ 못 할 거 라 곤 _ 상 상 도_ 못 했 네_
SHAP-GB	발 연기_ 도 저 히_ 못 보 겠다_ 진 짜_ 이렇게_ 연기를_ 못 할 거 라 곤 _ 상 상 도_ 못 했 네_

IG-MLP	[CLS] 이_ 영화를_ 이 제 서 야_ 보 게_ 되 다니 . . 명 작 으로_ 불러 질 만_ 한 데_ 제목 이_ 아 쉽 다_ [SEP]
IG-CNN-m	[CLS] 이_ 영화를_ 이 제 서 야_ 보 게_ 되 다니 . . 명 작 으로_ 불러 질 만_ 한 데_ 제목 이_ 아 쉽 다_ [SEP]
CAM-CNN-a	[CLS] 이_ 영화를_ 이 제 서 야_ 보 게_ 되 다니 . . 명 작 으로_ 불러 질 만_ 한 데_ 제목 이_ 아 쉽 다_ [SEP]
LIME-GB	이_ 영화를_ 이 제 서 야_ 보 게_ 되 다니 . . 명 작 으로_ 불러 질 만_ 한 데_ 제목 이_ 아 쉽 다_
SHAP-GB	이_ 영화를_ 이 제 서 야_ 보 게_ 되 다니 . . 명 작 으로_ 불러 질 만_ 한 데_ 제목 이_ 아 쉽 다_