Delete-Generate: 단어 n-gram의 삭제 및 생성에 기반한 한국어 스타일 변환

최형준○, 나승훈

전북대학교

modera2017@gmail.com, nash@jbnu.ac.kr

Delete and Generate: Korean style transfer based on deleting and generating word n-grams

Heyon-Jun Choi○, Seung-Hoon Na

Jeonbuk National University

요 약

스타일 변환(Style Transfer)은 주어진 문장의 긍정이나 부정 같은 속성을 변경하여 다른 속성을 갖는 문장으로 변환하는 과정을 의미한다. 본 연구에서는 스타일 변환을 위한 단어 n-그램 삭제의 기준을 확장하였고, 네이버 영화리뷰 데이터셋을 통해 이를 스타일 변환 이후 원래 문장의 스타일로부터 얼마나 차이가 나게 되었는지를 측정하였다. 측정은 감성분석기를 통해 이루어졌고, 기존 방법에 비해 6.28%p정도 높은 75.13%의 정확도를 보였다.

주제어: 스타일변환, BERT

1. 서론

스타일 변환(Style Transfer)은 주어진 문장의 긍정이나 부정 같은 속성을 변경하여 다른 속성을 갖는 문장으로 변환하는 작업이다. 주로 영상처리 분야에서 변환 특징을 입력 이미지에 적용하여, 직접 찍은 사진을 명화와 같은 특징을 가진 새로운 이미지를 만드는데 사용되었다. 이를 텍스트 데이터에 적용하려는 연구도 활발히 진행되고 있다.

본 논문에서는 네이버 영화 리뷰 데이터셋[1]을 이용하여 긍정 문장을 부정 문장으로, 또는 그 반대로 변환시킨 후, 각 변환된 문장이 원래 문장의 스타일을 얼마나 나타내고 있는 지를 감성분석기를 이용하여 측정할 것이다.

2. 관련 연구

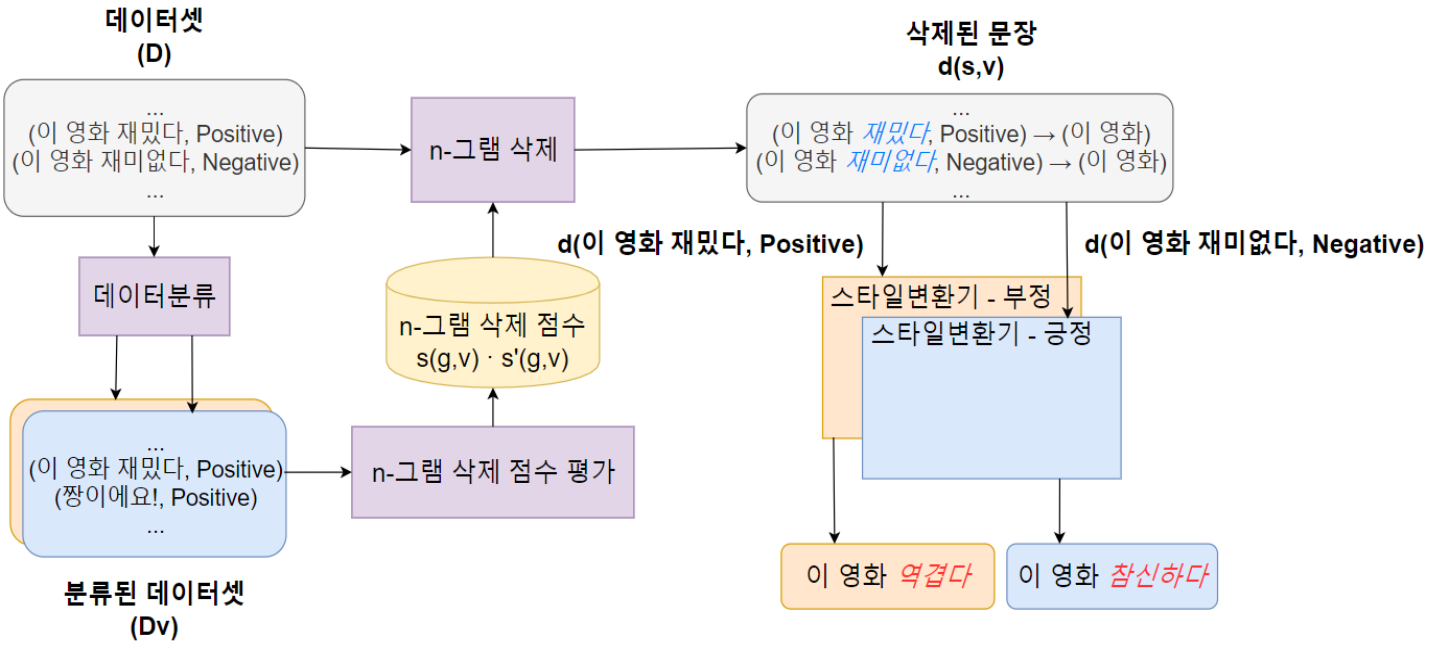
단어 n-그램을 삭제하여 스타일 변환을 시행하는 것은 [2]에서 시도되었다. [2]에서 스타일 변환은 세 가지 작업으로 구성된다. 첫 번째는 단어 n-그램 삭제로, 전체 데이터셋에서 같은 스타일인 문장에서 n-그램이 등장한 횟수와 다른 스타일인 문장에서 n-그램이 등장한 횟수를 통해 단어 n-그램을 삭제할 기준을 만들었다. 두 번째는 문장 검색이다. 삭제된 문장들의 집합에서 변환할 삭제된 문장과 가장 비슷한 문장을 찾은 후, 찾아낸 문장에서 삭제한 문장 n-그램을 얻어낸다. 세 번째는 이렇게 만들어진 삭제된 문장과 획득한 문장 n-그램을 이용하여 스타일 변환이 된 문장을 만들어내는 작업이다.

본 논문에서는 [2]에서 사용된 삭제과정을 감성분석기를 통해 단어 n-그램을 평가하도록 개량하고 delete only 작업을 시행하여 스타일 변환을 시행하였다. 기존의 방법은 단순히 단어 n-그램이 등장한 횟수를 통해 그 단어를 삭제할 지를 결정했다. 그렇기 때문에 형태소 등으로 토큰화 된 한국어에서는 단어삭제가 잘 이루어지지 않았다. 이 연구에서는 감성분석기를 통해 문장을 학습시키고, 이를 통해 삭제기준을 만들어 단어 n-그램이 해당 문장에서 스타일을 얼마나 내포하고 있는지를 더 정확히 낼 수 있도록 했다.

이러한 과정을 거치지 않고 직접 스타일 변환된 문장을 생성하려는 시도도 존재하였고, 뛰어난 결과를 보여주었다[3-4]. 이 연구들은 문장에 스타일 태그를 추가하고 Seq2Seq 모델을 통해 직접 문장을 생성한다. [3]에서는 GAN[5]을 적용하여 병렬 코퍼스가 아닌 데이터에 대해 스타일 변환을 시행하는 방법을 제시하였다.

스타일 변환의 대상인 스타일을 구별하는 것은 감성분석(Sentimental Analysis)과 매우 유사하다. 감성분석기는 문맥의 편향성을 기반으로 그 문장이 긍정적인지, 부정적인지 등의 속성을 구별하는 것이다. 스타일 변환의 대상인 문장의 스타일은 감성 분석의 대상과 동일하다. 이에 대한 연구는 지속적으로 이루어져왔고, BERT를 활용한 감성분석이 매우 뛰어난 성능을 보여준다[5].

본 논문과 [2]에서 스타일 변환을 위해 Seq2Seq를 활용하였다[6-7]. 기존의 RNN 모델을 활용하여 Seq2Seq를 구성하였고, 이는 보다 더 뛰어난 성능을 보여주는 Transformer[8]로 확장할 수 있다. Transformer는 RNN 모델과 달리 재귀적인 구조가 없어 병렬화가 가능하기 때문에 학습이 빠르고 문장의 길이에 크게 제한받지 않는다는 장점이 있다.



[그림 1] 스타일 변환기 모델 전체 구조

3. 스타일 변환 모델

3.1 데이터 설명

스타일 데이터는 문장과, 그 문장의 스타일을 나타내는 태그의 집합 $D = \{(x_1, v_1), (x_2, v_2), \dots, (x_n v_n)\}$이다. 여기서 x_i는 각각의 문장을 의미하고, $v_i \in V$는 스타일의 집합(예를 들어 $V = \{POS, NEG\}$), n은 전체 데이터의 크기이다. 문장 x_i는 다시 문장을 구성하는 토큰으로 구성된다. 즉, $x_i = [(t_{1,I}, v_{1,I}), (t_{2,I}, v_{2,I}), \dots, (t_{m,I}, v_{m,I})]$ 이다. $t_{j,i}$는 x_i의 토큰을, $v_{j,i}$는 그 토큰의 스타일, m은 문장의 길이이다. 문장 집합을 스타일별로 분리한 것은 $D_v = \{x : (x, v) \in D\}$이다.

3.2 문장 감성분석

문장이 해당 스타일을 얼마나 나타내는 지의 척도는 [6]에서 사용된 BERT 감성분석기를 이용했다. 감성분석기의 출력에서 각 스타일에 대한 확률을 그 문장이 그 스타일을 얼마나 가지고 있는 지에 대한 지표로 보고, 스타일 변환은 감성분석기의 출력이 해당 스타일일 확률을 최소화하는 방향으로 진행했다.

$$\text{score}_{tgt} = 1 - p(x_t | v_{src})$$
$$x_t = \text{styletransfer}(x, x_{tgt})$$

3.3 단어 n-그램 삭제

스타일 변환을 위해 문장 x에서 v_{src}를 나타내는 토큰, $\{t : (t, v_{src}) \in x\}$을 제거한다. 문장을 각 토큰에 대한 n-그램 g로 나눈 후, 각각의 n-그램에 대해 점수를 평가한 후 제거하였고, 큰 n값부터 작은 n값 순으로 제거를 시행하였다. 이렇게 제거가 완료된 문장을 $d(x, v_{src})$라고 하자. 이때, 이미 제거된 토큰이 n-그램에 포함된 경우(n-그램이 중첩된 경우)는 제거를 시행하지 않았다. 제거하기 위한 기준은 두 가지의 점수를 사용했다. 첫 번째 점수는 [2]의 식(1)이다.

$$s(g, v) = \frac{\text{count}(g, D_v) + \lambda}{\sum_{v' \neq v, v \in V} \text{count}(g, D_{v'}) + \lambda} \quad (1)$$

두 번째 점수는 [2]의 식(1)을 변형하였다. 문장집합의 모든 n-그램에 대해, 각각의 n-그램을 제거 한 문장집합 X′을 감성분석기로 평가했을 때 제거된 문장이 v′가 아닐 확률을 그 문장에서 n-그램이 등장한 횟수로 보고, $\text{count}(g, D_v)$를 각 n-그램의 원래 문장이 아닐 확률의 합인 $C(g, D_v)$로 변경하였다. n-그램이 해당 스타일일 확률 $p(v_{src}|g)$는 베이지안 규칙에 의해 다음과 같다.

$$p(v_{src}|g) = \frac{p(g|v_{src})p(v_{src})}{p(g|v')p(v') + p(g|v_{src})p(v_{src})} \quad (v' \neq v, v' \in V)$$
$$\approx \frac{p(g|v_{src})}{p(g|v_{src}) + p(g|v')}$$

여기서 근사적으로 $p(g|v_{src})$가 근사적으로 $C(g, D_v)$에 비례한다고 하자 그러면

$$p(g|v_{src}) \approx k \cdot C(g, D_v)$$
$$p(v_{src}|g) = \frac{C(g, D_v)}{C(g, D_v) + C(g, D_v)}$$

여기에 평활화를 위한 상수 λ를 추가하면

[표 1] 스타일 변환 예시

긍정 → 부정	
결말이 예상되는 영화 지만 재밌네요! 웃음*도 났다가 울음도 났다가 합니다*	결말이 예상되는 영화지만 웃음*에 속아서 다운 받아서 봤다가 실망*
10점밖에 주지 못한다는 게 아쉽다	*저질이다 다시* 주지 못한다는 게 아쉽다
부정 → 긍정	
액션성도 *없고, 마지막 급* 마무리도 그렇고, *주인공은 총 맞아도* 죽*지도 않네*	액션성도 마무리도 그렇고, *내용도 좋고*, 죽*기 전에 한번 더 보자*
연기력, 스토리 *최악! 결말은 더 최악*!	연기력, 스토리 *다 좋다*!

$$p(v_{src}|g) = \frac{C(g, D_v) + \lambda}{(C(g, D_{v'}) + \lambda) + (C(g, D_v) + \lambda)}$$

각 스타일별로 $C(g, D_v)$의 값이 다르기 때문에 이를 보정하기 위해 각 스타일별 $C(g, D_v)$의 최댓값 $Mc(v)$으로 나눈다. 따라서

$$p(g|v) = k \cdot \frac{C(g, D_v)}{Mc(v)}$$

$$p(v|g) = \frac{(C(g, D_v) + \lambda)}{(C(g, D_{v'}) + \lambda) \cdot \frac{Mc(v)}{Mc(v')} + (C(g, D_v) + \lambda)}$$

그러므로 삭제를 위한 두 번째 점수는 다음과 같다

$$s'(g, v) = \frac{C(g, D_v) + \lambda}{C(g, D_v) + \lambda + \sum_{v' \neq v, v \in V}(C(g, D_{v'}) + \lambda)\frac{Mc(v)}{Mc(v')}}$$

삭제에 사용한 최종적인 점수는 (1)과 (2)의 결과를 곱한 것을 사용하였다.

$$score_{delete} = s(g, v) \cdot s'(g, v)$$

3.4 모델 학습 및 스타일 변환

각각의 스타일 v에 대해 개별적인 Seq2Seq 모델 M_v을 학습시킨다[8]. 이는 [2]의 delete only와 같은 과정이다. 3.2에서 제거가 완료된 문장 $d(x, v)$를 Seq2Seq 모델을 이용하여 다시 x로 복원시키도록 학습시켜 스타일 요소를 삭제시켜 중성적이게 된 문장을 다시 스타일을 가진 문장으로 만들도록 한다. 이를 통해 다른 제거가 완료된 문장 $d(x', v'), x \neq x, v \neq v'$에 대한 Seq2Seq 출력 y가 v를 가진 문장이 되도록 한다.

학습이 완료된 이후, v_{src}를 가진 문장 x를 v_{tgt}로 변환한다. 3.2을 거쳐 제거된 문장 $d(x, v_{src})$를 획득하고, 이를 위에서 학습시킨 $M_{v_{tgt}}$를 통해 새로운 문장 y를 생성한다. 이렇게 생성된 문장 y는 v_{src}에서 v_{tgt}로 스타일이 변환되었다.

4. 실험

네이버 영화리뷰 데이터셋[1]을 이용하여 실험을 진행하였다. 이 데이터셋에서 스타일은 긍정과 부정 두 가지이고, 각각의 스타일마다 총 10만개의 데이터로 구성 되어있다. 영화 리뷰 데이터는 형태소 분석기를 통해 토큰화 시켰고, 토큰 시퀀스의 길이가 25 이상인 경우, 이를 초과하는 토큰은 잘라냈다. 3.3에서 사용된 평활화 상수 λ는 1을 사용하였고, Seq2Seq 모델은 OpenNMT[7]를 사용하였다.

스타일 변환 실험 이전에 이 데이터를 통해 감성분석기를 먼저 학습시켰다. 감성분석기의 BERT 모델은 BERT base 모델을 사용하였다[10]. 학습이 완료된 후, 이 감성분석기는 정확도 84.9%가 나왔다.

4.2 실험 결과

단어 n-그램의 삭제는 3.3의 점수를 측정한 후, 상위 4만,10만개의 단어 n-그램을 삭제시킨 후 스타일 변환을 시행하였다.

[표 2] 삭제 범위 별 스타일 변환 성능 - 감성분석 출력

스타일	평균점수	평균변화값
긍정 원문	0.153396	-
부정 원문	0.191062	-
긍정 → 부정 4만개	0.717528	**0.569260**
긍정 → 부정 10만개	0.667184	0.513788
부정 → 긍정 4만개	0.722657	0.556466
부정 → 긍정 10만개	**0.745642**	0.554580
긍정 → 부정 4만개 베이스라인	0.688835	0.535439
부정 → 긍정 4만개 베이스라인	0.683267	0.492205

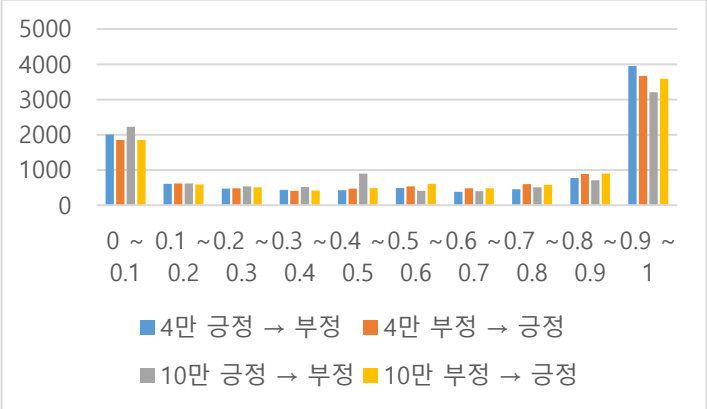
[표 1]은 스타일 변환 전후의 감성분석기 출력이다. 이는 감성분석기가 입력 문장이 원래 스타일이 아닐 확률을 의미한다. 베이스라인은 [2]의 delete only와 같은 작업을 실시한 결과이다. 즉, 3.3에서 단어 n-그램 삭제 점수를 식(1)만 사용했다. 개량된 삭제 기준을 사용했을 때 베이스라인에 비해 긍정 → 부정 변환의 경우 평균 0.033821만큼 더 변환되었고, 부정 → 긍정 변환의 경우 평균 0.021583만큼 더 변환되었다.

[표 3] 삭제 범위 별 스타일 변환 성능 - 감성분석 정확도

스타일	정확도
긍정 원문	12.52 %
부정 원문	16.77 %
긍정 → 부정 4만개	72.74 %
긍정 → 부정 10만개	64.47 %
부정 → 긍정 4만개	77.52 %
부정 → 긍정 10만개	**77.75 %**
긍정 → 부정 4만개 베이스라인	69.97 %
부정 → 긍정 4만개 베이스라인	67.73 %

[표 2]는 스타일 변환 전후의 감성분석기의 정확도이다. 정확도가 높을수록 감성분석기가 원래 스타일이 아닌 것이라고 판정한 것이다. 개량된 삭제 기준을 사용했을 때 베이스라인에 비해 긍정 → 부정 변환의 경우 2.77%p정도 높은 정확도가 나왔고, 부정 → 긍정 9.79%p 정도 높은 정확도가 나왔다.

[그래프 1] 감성분석기 출력의 변화값 분포



[그래프 1]는 스타일 변환 전후의 감성분석기 출력의 변화값 분포이다. 0.1 ~ 0.8의 변화를 가진 각 그룹은 500 내외의 데이터가 속해 있다. 0 ~ 0.1의 변화를 가진 그룹은 데이터 자체의 잡음이거나 영화를 직접 서술하여 평가하거나 반어법 등을 써서 스타일 변환이 어려운 데이터였다.

[표 4] 삭제 범위 별 스타일 변환 성능 - BLEU

스타일	형태소단위	음절단위
긍정 → 부정 4만개	0.064170	**0.162239**
긍정 → 부정 10만개	0.066850	0.154681
부정 → 긍정 4만개	0.076456	0.151977
부정 → 긍정 10만개	0.070409	0.136415
긍정 → 부정 4만개 베이스라인	**0.074813**	0.161613
부정 → 긍정 4만개 베이스라인	0.072119	0.148768

[표 3]은 수동으로 스타일 변환을 시행한 데이터를 스타일 변환기의 출력을 BLEU 점수를 측정한 것이다. 평가는 형태소단위와 음절 단위 두 가지로 시행하였다. 형태소 단위로 평가를 할 경우, 단어가 조금만 변형되어도 틀린 것으로 평가하기 때문에 점수가 낮게 나온다. 하지만 음절 단위에서 조금 변경되더라도 토큰의 의미가 비슷한 경우가 많아 음절단위 평가가 더 정확했다.

5. 결론

본 논문에서는 [3]의 단어 n-그램 삭제를 확장하는 방법을 제안하였고, 이를 네이버 영화 리뷰 데이터에 적용하여 스타일 변환을 시행하였다. 감성분석기를 통해 스타일 변환 정도를 측정하였고, 변환 결과 원래 스타일이 아닐 확률이 평균 0.562863 만큼 변환되었다.

후속연구로 본 연구와 같은 방식으로 문장을 삭제한 후, 삭제된 단어 n-그램을 일종의 Masked Language Model[10]로 취급하여 스타일 변환된 문장을 생성하도록 해볼 예정이다. 즉, 본 연구의 삭제 과정을 통해 문장에 마스킹을 하고, 이것을 Mask LM을 통해 마스킹에 해당되는 단어들을 생성시켜 이것을 마스킹된 입력 문장과 합쳐 스타일 변환된 문장을 생성시키도록 해볼 예정이다.

참고문헌

[1] https://github.com/e9t/nsmc/
[2] Juncen Li, Robin Jia, He He, Percy Liang, Delete, Retrieve, Generate: A Simple Approach to Sentiment and Style Transfer, 2018.4
[3] Ning Dai, Jianze Liang, Xipeng Qiu, Xuanjing Huang, Style Transformer: Unpaired Text Style Transfer without Disentangled Latent Representation, 2019.8
[4] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, Eric P. Xing, Toward Controlled Generation of Text, 2018.9
[5] Goodfellow, Ian, et al., Generative adversarial nets, 2014.
[6] 박광현, 나승훈, 신종훈, 김영길, "BERT를 이용한 한국어 자연어처리: 개체명 인식, 감성분석, 의존 파싱, 의미역 결정", 한국 정보과학회 학술발표논문집, 2019.6
[7] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, Alexander M. Rush, OpenNMT: Open-Source Toolkit for Neural Machine Translation, 2017
[8] Luong, Minh-Thang, Eugene Brevdo, and Rui Zhao, Neural machine translation (seq2seq) tutorial, https://github.com/tensorflow/nmt, 2017
[9] Ashish Vaswani, et. al., Attention Is All You Need, 2017.12
[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018.10
[11] Papineni, Kishore, et al., BLEU: a method for automatic evaluation of machine translation, 2002, 7