Automated Essay Scoring with BERT based NLP Model

코랩사조(Team4)
허윤빈(응용통계학과), 강다연(응용통계학과), 김명빈(응용통계학과), 나영은(역사학과)

1. 서 론

윌리엄과 플로라 휴렛 재단은 지난 2012년부터 현재까지 'ASAP(Automated Student Assessment Prize)'를 후원해 왔다. 딥러닝 분야가 최근에서야 유망해진 것을 보면 'ASAP'는 꽤 오래전부터 명맥을 유지해 온 대회임이 분명하다. 그중에서도 'Automated Essay Scoring 2.0'은 소외된 지역 사회에서 교육자들의 올바른 지도를 받지 못하는 학생들을 위해 시작된 대회이다. 그 때문에 에세이 채점 알고리즘을 증진하는 데에 초점을 두고 있다.

알고리즘 평가는 QWK(Quadratic Weighted Kappa)로 이루어진다. QWK 는 관찰자 간의 측정 범주 값에 대한 일치도를 관측하는 방법으로, 0(무작위 일치)부터 1(완전 일치) 사이의 값으로 측정된다. QWK 는 실제 점수 i와 예측 점수 j로 이루어진 n x n 행렬 O와 가중 행렬 W로 계산되며, 식은 다음과 같다.

$$\kappa = 1 - \frac{\sum_{i,j} w_{j,j} o_{i,j}}{\sum_{i,j} w_{j,j} E_{i,j}}$$

Figure 1 Quadratic Weighted Kappa

본 는 자연어 처리에 특화된 BERT 알고리즘을 활용하여 해당 문제에 접근하고자 했다. BERT 알고리즘은 2018년 구글에서 고안된 오픈 소스 모델로, 레이블이 없는 텍스트의 양방향 표현이 특징이다. BERT 기반의 여러 모델 중에서도 Albert, Electra, DeBERTa 모델을 통해 점수를 예측하고자 한다.

2. 본 론

서론에서 언급하였듯, 학생들의 essay 자료를 input으로 입력받아 자동화된 채점 알고리즘을 학습하는 것이 목표이다. Essay를 채점하는 기준이 모호하다고 판단하여, 직접 Feature Extracting을 진행하여 머신러닝 모델을 학습하기 보다는, 해당 과정을 모델이 직접 판단할 수 있는 Pretrained Language Model을 활용하여 score를 예측하고자 한다.

1) Dataset

Kaggle Competition에서 제공한 Learing Agency Lab Automated Essay Scoring 2.0 데이터셋을 사용하였다.

약 24000명의 학생들이 작성한 에세이가 데이터프레임로 구성되어 있다. 각 에세이는 1에서 6까지의 척도로 점수가 매겨졌다.

- essay_id : The unique ID of the essay
- full_text : The full essay response
- score : Holistic score of the essay on a 1-6 scale

essay_id	text	score
ea9f1f5	New Planet, New Home₩n₩n"It is hard to leave s...	5
1845963	Electoral college unfair, outdated, and irrati...	2
d69dd40	Fourty-five minutes and counting wait for the ...	2
f3c5fdd	To:Mr. State senator.₩n₩nFrom: PROPER_NAME₩n₩n...	3
ced362b	From my opinion I think this is one of the bes...	3

Figure 2 Essay Scoring Dataset

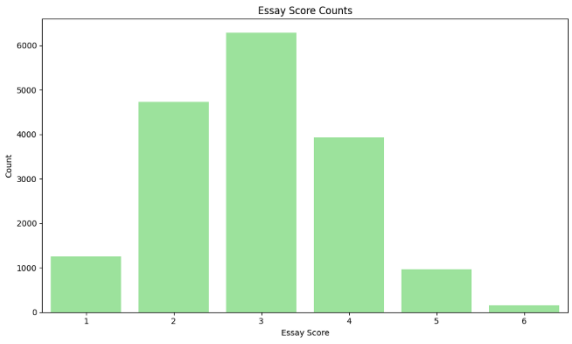

Figure 3 Target Distribution

2) Metric

Real data와 predicted data의 측정 범주 간 값에 대한 일치도를 측정할 수 있는 방법인, Quadratic Weighted Kappa(이하 qwk)를 평가지표로 활용하였다.

보통 다중 클래스 간에 순서 관계가 있을 경우 사용할 수 있는 지표로, 완전한 예측에 가까울수록 1, 랜덤한 예측에 가까울수록 0에 가까워진다.

순서 관계가 존재하는 데이터는 실제 값과 예측 값의 차이가 작을수록, 즉 두 측정값이 가까울수록 좋은 예측임을 나타내고 Quadratic Weighted Kappa는 이러한 특성을 반영하는 지표이다.

아래 식을 보면, 실제 값과 예측 값이 가까울수록 평가지표가 좋아짐을 확인할 수 있다. 실제 점수는 i, 예측 점수는 j로 표현된다.

$$w_{i,j} = \frac{(i-j)^2}{(N-1)^2} \qquad \kappa = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}}.$$

Figure 4 Quadratic Weighted Kappa expression

3) Model

BERT 기반의 파생모델을 활용하여, downstream task로 score를 예측하는 fine tuning을 진행하였다.

Bidirectional Encoder Representations from Transformers의 약자인 BERT는 트랜스포머의 encoder를 쌓아 올린 구조이다.

BERT는 MLM(Masked Language Model)과 NSP (Next Sentence Prediction)을 통해 사전학습된다. MLM은 일련의 단어가 주어지면 무작위 토큰에 마스킹한 뒤, 주변 단어의 맥락을 통해 마스킹된 토큰을 예측하는 pretrain 방식이다. 따라서, 이전 토큰들을 참고하여 다음 토큰을 예측하는 GPT와 같은 decoder 기반의 모델보다 문맥을 파악하는 능력이 더 뛰어나다는 특징을 갖는다. 추가적으로 NSP를 통해 두 문장의 관계를 이해하는 pretrain이 진행된다. 따라서 BERT 기반의 모델은 질문 답변과 같은 문장간 관계를 이해하는 task에도 뛰어나다는 특징을 갖는다. Essay를 이해하고 그에 따른 score를 예측하는 task에 적합하다고 판단하여 BERT기반 파생모델을 학습 모델로 채택하였다.

프로젝트에 사용된 BERT기반의 파생모델은 다음과 같다.

1. ALBert

ALBert(A Lite version of BERT)는 인코더 구조를 더욱 효율적으로 만들기 위해 몇 가지 방안을 도입한 모델이다.
첫째로, cross-layer parameter sharing 기법을 도입하여 첫 번째 인코더 레이어의 파라미터를 다른 레이어와 공유한다. 이를 통해 변수의 개수와 메모리 사용량을 줄이는 효과를 얻었다.
둘째로, factorized embedding layer parameterization을 통해 hidden layer에서 토큰 임베딩 차원을 분리, 임베딩 차원을 크게 줄였다. 매개변수의 효율성이 증대되는 효과를 얻었다.
마지막으로, NSP 목표를 문장 순서 예측으로 바꾸었다. 단순히 두 문장이 함께 속해 있는지가 아니라, 연속된 두 문장의 순서가 바뀌었는지를 예측함으로써 문맥의 이해도를 높일 수 있었다. 모델 구조의 큰 변화 없이 몇 가지 detail을 수정하는 것 만으로도 모델의 성능과 효율성을 증대시킨 모델이다.

2. ELECTRA

ELECTRA(Efficiently Learning an Encoder that Classifies Token Replacements Accurately)는 기존의 MLM task를 사용하는 BERT의 사전학습 방식을 RTD(Replaced Token Detection)으로 대체한 모델이다.

RTD는 마스킹할 대상이 될 토큰들을 다른 토큰으로 변경한 뒤 이 토큰이 실제 토큰인지 교체된 토큰인지 판별하는 형태로 학습을 진행한다. MLM을 사용하는 BERT모델은 마스킹된 토큰이 무엇인지를 예측하기에, 마스킹된 15%의 토큰만을 학습하지만 ELECTRA는 각 토큰의 원본 여부를 판단해야 하므로 모든 토큰을 대상으로 학습이 이루어진다는 점에서 더 효율적이다.

3. DeBERTa

DeBERTa(Decoding-enhanced BERT with Disentangled Attention)은 BERT모델의 두 가지 구조를 변경하였다.

우선, 각 토큰이 두 개의 벡터로 표현된다. 기존의 토큰들은 임베딩벡터에 위치임베딩 값을 추가하여 하나의 벡터로 활용하였다. 이를 토큰임베딩 위치임베딩 각각의 벡터로 분리하여 인코딩함으로써, 셀프 어텐션 단계에서 인접한 토큰 쌍의 의존성을 더 잘 모델링하도록 하였다. 또한 단어의 구문론적 역할을 반영할 수 있는 절대 위치를 파악하기 위해, softmax layer이전에 absolute word position embedding을 통합하였다.

4) Classification to Regression

Essay dataset의 종속변수인 score값이 1-6 까지의 scale을 갖는 범주형 데이터이기 때문에 가장먼저 6개의 class를 분류하는 다중분류 문제로 파인튜닝을 진행하였다. 이를 위해 num_labels 파라미터는 6으로, loss는 cross entropy를 활용하였고 dropout_prob는 0.5로 설정하였다. 하지만 qwk 자체가 순서가 있는 범주형 변수를 반영하는 지표이기 때문에 성능이 좋지 않았다. 종속형 변수의 순서관계를 고려하기 위해 Regression문제로 변환하여 파인튜닝을 진행하였다. Classification모델을 Regression모델로 변환하기 위해

num_label을 1로, loss는 MSE를 사용, dropout은 고려하지 않았다.

5) Data Augmentation

본론 1)Dataset – [그림2]와 같이, 종속변수 class간 데이터 불균형이 존재한다. 특히 class1, 5, 6의 경우 다른 클래스에 비해 매우 희소한 것을 알 수 있는데, 이로 인해 모델이 class1, 5, 6을 제대로 예측하지 못할 것이라 판단하였다. 모델이 적절한 예측을 할 수 있도록 EDA(Easy Data Augmentation)기법을 적용하여 class 1, 5, 6 데이터를 증강하였다.

EDA기법 중 Randsom Insetion을 적용한 결과 예시는 다음과 같다.

```
# Example usage
sentence = "This is an example sentence for random insertion."
augmented_sentence = random_insert(sentence, 3)
print("Original Sentence: ", sentence)
print("Augmented Sentence: ", augmented_sentence)
```

```
Original Sentence:  This is an example sentence for random insertion.
Augmented Sentence:  example This is an example random sentence for random insertion insertion.
```

Figure 5 Random Insertion example

그림을 보면, 'This is an example sentence for random insertion'이라는 문장이, 'example This is an example random sentence for random insertion' 이라는 문장으로 증강되었다. 문법 혹은 문맥적 의미가 중요한 essay scoring에서, 이처럼 문법적 문맥적 오류가 분명한 문장은 scoring에 영향을 준다는 문제가 있다. 따라서, EDA기법 대신 단어나 구문을 동의어로 대체해주는 WordNet과 재귀적 번역 방식인 Round-Trip Translation기법을 활용하여 데이터를 증강하였다.

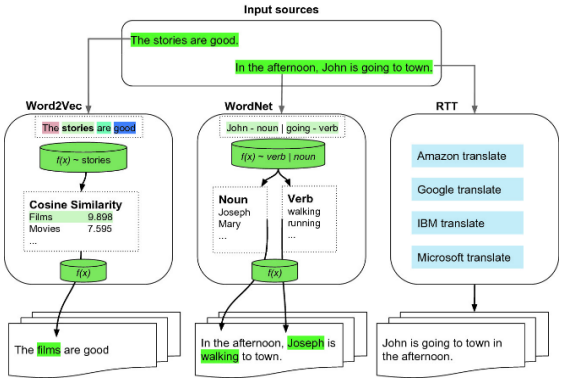

Figure 6 Text Augment

Wordnet은 영어에 대한 오픈 소스 어휘 데이터베이스이다. Wordnet은 단어를 의미별로 그룹화하여 표현하는 신셋(synset)을 통해 동의어 집합을 제공한다. 동의어 외에도 반의어, 유의어, 상위어, 하위어 등의 다양한 어휘적 관계를 제공하기 때문에 단어의 의미와 형식을 상호 연결하여 의미가 명확해진 네트워크에서 서로 가까이 있는 단어를 제공하기 때문에 동의어 대치에 효율적인 증강 방법이다.

Round-Trip Translation을 의미하는 RTT는, 문장을 다른 언어로 번역한 뒤 결과를 다시 원래 언어로 번역하는 프로세스를 거쳐 텍스트 데이터를 증강한다.

두 방법을 이용해 텍스트 데이터를 증강할 수 있는 Augmenter 클래스를 아래와 같이 정의하여 불균형 클래스를 기존 데이터 수의 3배로 증강하였다.

```
class Augmenter:
    def __init__(self, method='wordnet', **kwargs):
        """
        Augmenter 클래스 초기화
        method: 사용할 증강 기법 ('wordnet', 'word2vec', 'translate')
        kwargs: 각 증강 기법의 인스턴스 생성 시 필요한 추가 매개변수
        """
        self.method = method.lower()
        if self.method == 'wordnet':
            self.augmenter = Wordnet(v=kwargs.get('v', True),
                                     n=kwargs.get('n', False),
                                     runs=kwargs.get('runs', 1),
                                     p=kwargs.get('p', 0.5))

        elif self.method == 'translate':
            self.augmenter = Translate(src=kwargs.get('src', 'en'),
                                       to=kwargs.get('to', 'fr'))

        else:
            raise ValueError("지원하지 않는 증강 기법입니다. 'wordnet', 'translate' 중 하나를 선택하세요.")

    def augment_text(self, text, **kwargs):
        """
        주어진 텍스트를 증강하는 메소드
        text: 증강할 원본 텍스트
        kwargs: 증강 시 필요한 추가 매개변수
        """
        if self.method == 'wordnet':
            top_n = kwargs.get('top_n', 10)
            augmented_text = self.augmenter.augment(text, top_n=top_n)
        elif self.method == 'translate':
            augmented_text = self.augmenter.augment(text)
        else:
            raise ValueError("지원하지 않는 증강 기법입니다. 'wordnet', 'translate' 중 하나를 선택하세요.")

        # 개행 문자 제거
        augmented_text = re.sub(r'\n', ' ', augmented_text)
        return augmented_text
```

Figure 7 Augmenter Class

Original Text	WordNet	RTT
I will be explaining how the "face" is a land form.	I will comprise explaining how the "face" is a ground form.	I will explain how the "face" is a terrestrial form.

Table 1 Augmented Text example

6) Experiments

Model Configuration(이하 CFG)은 다음과 같이 설정하였다.

Configuration

Model = 'microsoft/deberta-v3-small'

max_length = 1024

batch_size = 8

epoch = 10

learning rate = 2e-5

weight_decay = 0.01 optimizer = adamw early stop patience = 3 dropout = 0.5

CFG는 동일하게 둔 채로, 4), 5)의 적용 유무에 대한 Ablation study를 다음과 같이 진행하였다.

Table 2 Ablation1. Classification vs Regression

	Classification	Regression
ALBert	0.698652	0.788330
ELECTRA	0.505179	0.782262
DeBERTa	0.730117	0.818309

Classification과 Regression의 Best QWK 값을 비교한 표이다. Classification보다 Regression의 경우 결과가 훨씬 좋은 모습을 보인다. 이는 종속형 변수값이 순서 관계가 있는 범주형 값이기 때문인 것으로 파악된다. 세 가지 BERT based Model 중에서도 DeBERTa가 가장 좋은 성능을 보였다. 따라서, Scoring을 위한 NLP 모델로 DeBERTa를 채택하였다.

Table 3 Ablation2. Original vs Augmented

DeBERTa	Loss	Best QWK
Original	0.297137	0.818309
Augmented	0.242523	0.914721

Augmenter 클래스를 이용하여 소수 데이터 클래스를 증강 후 DeBERTa모델을 적용했을 때 약 0.1 더 향상된 QWK값을 얻을 수 있었다. 소수 클래스에 대한 신뢰성이 이전보다 향상됨으로써 가장 최적의 결과를 도출해 낼 수 있었다.

3. 결 론

BERT 알고리즘을 사용하여 분류 모델을 구축했을 때, QWK는 최소 0.50에서 최대 0.73이었다. 평균적인 QWK가 현저하게 낮은 것을 확인할 수 있었으며, 이 문제를 해결하기 위해 회귀 모델을 다시 구성했다. 이 때 QWK가 최소 0.78에서 최대 0.81로 증가했다. 그 중에서도 DeBERTa 가 가장 높은 수행 능력을 드러냈다.

데이터 시각화 과정에서 에세이 점수 데이터의 불균형을 발견했으며, 해결 방안으로 텍스트 데이터 증강을 선택했다. 데이터 증강을 하지 않은 DeBERTa 회귀 모델의 최대 QWK는 0.78인 반면, 데이터 증강을 거친 DeBERTa 회귀 모델의 최대 QWK는 0.91로 괄목할 만한 성능 향상을 보여 주었다.

수행 능력을 더욱 증대시키기 위해 더욱 다양한 LLM 모델을 사용하여 모델을 학습시킨다면 더 나은 결과를 얻을 수 있을 것으로 기대한다. 최근 Llama3와 같은 LLM모델이 계속해서 출시되고 있는데, 다양한 LLM 모델을 활용하여 성능을 더욱 향상시킬 여지가 남아있다.

QWK 값 0.9라는 좋은 결과를 얻었지만 Kaggle Competition내 공식 test set을 활용한 public score를 얻지 못했다는 점이 가장 아쉽다. Validation set에 대한 Score는 높았지만 실제 test set에 적용했을 경우에는 이보다 더 낮은 값이 도출될 것이라는 것을 인지하고 있다. 또한 공식 public score는 실행시간을 함께 고려하기 때문에, public score를 확인하여 성능뿐 아니라 실행속도를 고려하여 더 나은 알고리즘을 제안하는 것이 본 프로젝트가 향후 나아갈 수 있는 방향이라고 생각한다.

2014년, 말랄라 유사프자이는 '한 명의 아이, 한 명의 선생님, 하나의 펜, 하나의 책이 세상을 바꿀 수 있다. 교육이 오직 해결책이며, 교육이 우선 되어야 한다.'고 호소했다. 교육은 우리 삶과 사회에 있어 필수 불가결한 존재이다. 그러나 2014년으로부터 10년이 지난 지금, 전세계 아동 100명 중 63명은 여전히 학습 빈곤 상태에 놓여져 있다. 'ASAP'는 유사프자이가 말한 '하나의 선생님'의 역할을 하고 있다.

'Automated Essay Scoring 2.0'의 모든 참가자들이 63명을 위한 하나의 선생님이 되어 주기를 바라며 보고서를 마친다.

참고 문헌

1) Marivate, V., Sefara, T. (2020). Improving Short Text Classification Through Global Augmentation Methods. In: Holzinger, A., Kieseberg, P., Tjoa, A., Weippl, E. (eds) Machine Learning and Knowledge Extraction. CD-MAKE 2020. Lecture Notes in Computer Science(), vol 12279. Springer, Cham. https://doi.org/10.1007/978-3-030-57321-8_21
Github : https://github.com/dsfsi/textaugment

2) https://link.springer.com/chapter/10.1007/978-3-030-57321-8_21 : Improving Short Text Classification Through Global Augmentation Methods

3) Attention is All You Need,
arXiv:1706.03762 [cs.CL]

4) ALBERT : A Lite BERT for Self-surpervised Learning of Language Representations,
arXiv:1909.11942 [cs.CL]

5) ELECTRA : Pre-training Text Encoders as Discriminators Rather Than Generators,
arXiv:2003.10555 [cs.CL]

6) DeBERTa : Decoding-enhanced BERT with Disentangled Attention,
arXiv:2006.03654 [cs.CL]

7) Afrizal Doewes, Nughthoh Arfawi Kurdhi, and Akrati Saxena. "Evaluating Quadratic Weighted Kappa as the Standard Performance Metric for Automated Essay Scoring." Eindhoven University of Technology, The Netherlands; Universitas Sebelas Maret, Indonesia; Leiden University, The Netherlands.
https://educationaldatamining.org/EDM2023/proceedings/2023.EDM-long-papers.9/index.html