KoBERT를 이용한 사투리 지역, 감정 분류 모델

Korea University COSE461 Final Project

Name
Team 97
2014170231 정준하
2016320114 장세음

Abstract

저희 팀은 KoBERT를 이용하여 사투리의 언어적 특성을 더 잘 내포하고 있는 embedding을 가진 모델을 만들었습니다. AI Hub의 한국어 방언 발화 데이터를 활용하여 사투리 data set을 만들었고, 이를 KoBERT에 학습시켜 사투리 지역분류 모델과 사투리 감정 분류 모델을 만들었습니다. 사투리 지역 분류 모델은 78.7%의 정확도를 보였고 사투리 감정 분류 모델은 60.5%의 정확도를 보였습니다. 또한 사투리 감정 분류 모델이 사투리 감정 예측에서 표준어 감정 분류 모델보다 더 좋은 성능을 보였습니다.

1. Introduction

한국 뿐만 아니라 전 세계에서 지역 및 나라에 따라 다양한 방언, 사투리(dialect)를 사용하며 그에 따른 연구들이 이루어지고 있습니다[1]. 이러한 사투리(dialect)들은 구문, 음운 단어 및 이를 조합하는 방법에서 표준어와 크고 작은 차이를 보이고 있습니다. 사투리(dialect)는 언어를 통해 지리적 및 사회학적 특성을 내포하며 지역 및 연령 계층간 사회적 정체성의 지표로 활용되기도 합니다.

사투리가 내포하는 사회적, 지리적 정체성은 다양한 컨텐츠에서 활용되어 왔습니다. 사투리를 적극적으로 채용한 드라마(응답하라 1988등)가 큰 인기를 끌었으며 이러한 류의 컨텐츠 속에서 사투리는 개인이 어느 지역을 연고로 하는지 쉽게 나타내는 수단입니다. 같은 지역을 연고로 함을 보여주는 지리적 정체성 외에도 사투리를 사용하지 않는 저연령층과 사투리를 주로 사용하는 고연령층 간의 소통 문제도 사투리로 인한 사회적 정체성 갈등을 잘 나타냅니다. 특히 해외에서는 sns에서 영어를 사용하는 사람들 간에 사용하는 dialect를 통해 인종 및 지리, 사회적 특성을 추론할 수 있다는 연구 결과가 존재합니다[2].

한국에서 사투리 연구는 주로 제주도 사투리의 음성 및 텍스트 데이터에 대해 이루어졌습니다. 제주어는 한글 창제 당시의 중세에 사용하던 어휘들이 많이 남아있어 한국어의 원형과 한글의 제작 원리를 보여주는 언어라 평가받고 있기 때문입니다[3]. 하지만 그 외 지역의 사투리들도 사회적, 지리적 요인을 나타내는 척도로써 큰 역할을

차지함에도 연구가 거의 이루어지지 않고 있습니다.

2016년 구글이 방대한 양의 위키피디아 vocab에 대해 pretrain된 BERT를 공개하였습니다. 하지만 기존 BERT가 한국어 데이터 셋에 대해 성능이 좋지 않아 SKT-KoBERT 등 BERT 기반 모델과 konlpy등 한국어 단어의 토큰화를 위한 다양한 형태소 분석기가 등장하였으며 이는 여러 목적을 가진 언어모델 간 성능 비교에서 우위를 보여왔습니다. 특히 KoBERT는 SKT에서 구글의 BERT 모델을 기반으로 한국 위키피디아 vocab를 pre-train시켜 만든 트랜스포머 기반 모델로, 요약(summarization), 번역, 핵심문장 추출 등에서 좋은 성능을 보이고 있습니다.

저희는 KoBERT를 이용하여 사투리의 언어적 특성을 더 잘 내포하고 있는 embedding을 가진 모델을 만들고자 합니다. 저희는 사투리간 지역 분류와 사투리 데이터 감정분석을 통해 사투리 데이터 셋을 통해 학습된 embedding의 성능이 뛰어난지, 그리고 이중 부정이나 비꼬기 등의 어려운 예제들에 대해서도 잘 분류가 될지에 관해 확인해보고자 합니다. 또한 표준어로 학습된 KoBERT 모델보다 사투리로 학습된 KoBERT 모델이 사투리 감정분류에서 더 좋은 성능을 보여줄 수 있는지 확인해보고자 합니다

2. Related Work

아랍어는 수백 가지 언어를 포함하는 포괄적인 용어로, 하나의 언어 뿌리를 가지고 있음에도 아랍어에서 파생된 언어 사용자끼리 잦은 소통의 어려움을 겪는다는 기이한 특성을 가집니다[1]. 기존까지는 표준 아랍어(MSA)를 기준으로 사용하여 아랍어의 다양성을 구어영역으로 개인에 한해 억제해왔지만, SNS 사용이 확대되면서 개인이 표준 아랍어가 아닌 아랍어를 사용하면서 사용자간 소통이 어려운 문제가 더 대두되기 시작했습니다. 특히 다양한 방언 간의 표준 철자법이 명확히 정해져 있지 않고, 형태학적인 풍부함이 커서 방대한 양의 온라인 데이터 주석과 방언사전이 먼저 갖춰져야 한다는 문제점이 존재했습니다.

한국어 사투리는 아랍어와 달리 이중모음 의와 위의 유무, ㄴ탈락 등 음운론적인 사실과 해요체의 존재 등 형태론적인 사실, 그리고 부정형식에서 부사‘못’의 통사적인 사용 등 음운의 형태와 어법, 어휘 등의 차이를 통해 구분이 가능합니다[4]. 이를 통해 사투리를 크게 호남사투리, 영남사투리, 경기도 사투리, 제주 방언으로 나눌 수 있습니다. 또한 800만개의 지역별 사투리 데이터 셋 또한 이미 잘 갖춰져 있습니다.

3. Approach

앞서 방언 구획과정에서 나타난 한국어 사투리간 차이점과, 한글이 교착어이면서 동의어를 context에 따라 다르게 분석해야 한다는 것을 고려하면 사투리 classification model은 형태소 간의 차이 및 단어 주위의 문맥(context) 단어들과의 관계 분석이 중요하다는 것을 알 수 있습니다.

위의 목적을 만족하고 Out-of-vocabulary 문제를 막기 위해 embedding을 만들 때 Sub-word를 적용한 tokenizer를 이용하고자 합니다. 사투리 데이터 작업을 위해 알맞은 tokenizer는 타 논문[5]에서 나온 성능을 참고하여, Morpheme-aware Subword 방식을 채택하였습니다. Morpheme-aware Subword는 data-driven 방식과 linguistically-driven 방식을 혼합하여 사용하며 MeCab-ko와 BPE를 순서대로 적용하여 형태소 정보를 포함하는 subword로 tokenize할 수 있습니다. Mecab 형태소 분석기로 토큰화된 문장을 이어 붙여 처리한 후 sentencepiece로 재토큰화하는 방식으로 진행하였습니다. 코드는 KoBERT 깃허브에 있는 '네이버 영화평 이중분류 예시코드'를 참조하였습니다.

모델의 경우 아랍어 방언 연구[1] 속 classification model에서 BERT를 기반으로 한 모델이 가장 성능이 좋았기 때문에. BERT 를 기반으로 한 KoBERT를 모델로 선정하였습니다. BERT는 Transformer의 인코더를 사용하고 MLM(masked language model)을 통해 원본 token을 대체하는 특수 mask token을 사용하여 손상된 token을 맞추는 방식으로 학습하는 모델입니다. BERT 기반 모델들은 검색된 vector들에 대해 drop out layer에서 시작해 hidden layer을 거쳐 softmax 활성화 출력 함수로 forwarding해 클래스 간 분류를 생성하고, backpropagation을 하는 동안 손실이 BERT 인코더를 포함하여 전체 네트워크에 다시 전파됩니다. 이 때 BERT가 사용하는 tokenizer은 BPE 중에서 AE(AutoEncoding)에 해당되어 양방향 모델이라 mask 앞 뒤 문맥을 고려한 embedding을 만들게 되고 그러한 이유로 BERT는 지금까지 무난히 좋은 성능을 보여주고 있습니다. 저희는 이러한 BERT 모델의 특성을 이어받되, 형태소 분석이 쉽고, 한국어 데이터셋에 어울리는 KoBERT 모델을 이용했습니다.

```
tokenizer = KoBERTTokenizer.from_pretrained('skt/kobert-base-v1', sp_model_kwargs={'nbest_size': -1, 'alpha': 0.6, 'enable_sampling': True})
bertmodel, vocab = get_kobert_model('skt/kobert-base-v1',tokenizer.vocab_file)
```

Tokenizer에서 사용한 parameter로는 enable_sampling=True, alpha=0.6, nbest_size=-1 를 통해 drop out 을 적용하였습니다.

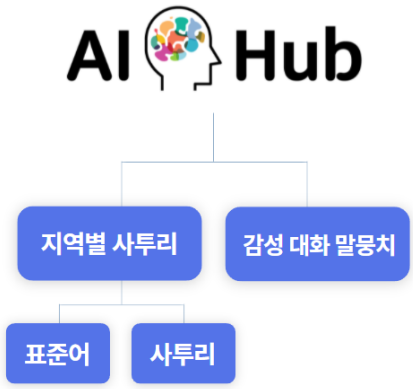
```
#optimizer와 schedule 설정
no_decay = ['bias', 'LayerNorm.weight']
optimizer_grouped_parameters = [
    {'params': [p for n, p in model.named_parameters() if not any(nd in n for nd in no_decay)], 'weight_decay': 0.01},
    {'params': [p for n, p in model.named_parameters() if any(nd in n for nd in no_decay)], 'weight_decay': 0.0}
]
```

Adam optimizer와 weight decay를 함께 포함시켰고 이를 통해 Adam 알고리즘을 거쳐서 막고 기울기가 급작스럽게 줄어들어 부분 극솟값에서 학습을 멈추는 것을 막고자 했습니다.

```
# Setting parameters
max_len = 64
batch_size = 128
warmup_ratio = 0.1
num_epochs = 5
max_grad_norm = 1
log_interval = 200
learning_rate =  5e-5
```

4. Experiments

4.1 Data



데이터는 AI-Hub에 있는 '한국어 방언 발화 데이터' 중 강원도, 경상도, 전라도, 제주도, 충청도 데이터를 사용했습니다. 각 데이터의 양은 약 100만개 ~ 150만개 사이였으며 각 데이터를 모두 합친 Training set으로는 약 800만개의 데이터를 추출하여 학습에 사용했습니다.

추출 방법은 다음과 같습니다. AI-Hub의 데이터 중 txt 파일만을 뽑아내어 그 중 사투리 데이터와 사투리의 표준어 버전 데이터를 각각 텍스트 전처리 과정을 거쳐 추출해 주었습니다. 전처리 과정 중에 지역별로 labeling도 같이 진행해 주며 지역이 labeling된 사투리 데이터셋을 도출해냈습니다.

다음으로 사투리 감정분류 모델을 만들기 위해 표준어 감정분류 모델을 먼저 만들었습니다. 모델을 학습하기 위해 감정이 labeling된 표준어 데이터셋을 사용했습니다. 이는 AI-Hub의 '감성 대화 말뭉치'를 사용했으며, 텍스트에 대한 감정 대분류(기쁨, 슬픔, 당황, 분노, 불안, 상처)를 label로 사용하였습니다.

위와 같이 labeling된 표준어 데이터를 모델에 학습시켜 표준어 감정 분류 모델을 만든 후, 앞서 전처리한 사투리의 표준어 버전 데이터를 넣어 감정을 labeling했습니다. 이를 원본 사투리 데이터와 매칭시켜 사투리 감정분류 모델 학습을 위한 6가지 감정이 labeling된 사투리 데이터를 만들었으며 추가적인 human labeling을 통해 예측이 잘 되지 않은 label들을 수정했습니다.

사투리의 이중 부정, 비꼬기 등의 어려운 예제들까지 분류해보기 위해 각 텍스트에 대한 긍정/부정 유사도를 구해 긍정/부정에 관한 labeling도 진행했습니다. 이는 kakaobrain에서 개발한 pororo api를 이용하여 긍정/부정 labeling을 진행했습니다. 최종적으로 사투리 감정분류 모델 학습에 쓰일 데이터셋으로 6가지의 감정만 labeling되어 총 6가지 label로 분류된 사투리 데이터의 표준어 버전 데이터와, 여기에 긍정/부정을 합쳐 총 12가지 label로 분류된 사투리 데이터의 표준어 버전 데이터 이렇게 두가지 데이터셋을 구축하였습니다.

4.2 Evaluation method

Training 과정에서 train 80%, test 20%로 분할하여 테스트 했으며 model의 성능을 evaluate 하기 위해 test accuracy와 f1 score을 구했습니다.

감성 분류 모델에서는 이중 부정과 비꼬기 등의 어려운 예제 data set을 직접 만들고 labeling을 진행하여 모델에 대입하여 나오는 accuracy를 구해 추가로 검증을 진행했습니다.

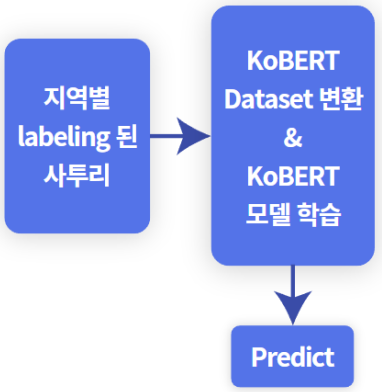
마지막으로 감성 대화 말뭉치로 학습된 KoBERT 모델과 감정 labeling으로 학습된

KoBERT 모델 사이의 성능 비교를 위해 5000개의 감정이 labeling된 사투리 data로 validation set을 만들어 accuracy와 f1 score을 측정하여 성능 비교를 진행했습니다.

4.3 Model

지역별 사투리의 분류를 위해, 그리고 사투리의 감정 분류를 위해 사투리 지역 분류 모델과 사투리 감정 분류 모델을 각각 만들었습니다. 두 모델 모두 KoBERT를 이용했으며, 각 분류의 목적에 맞게 labeling된 데이터를 KoBERT 모델에 학습시켜 분류 모델을 개발했습니다.

I. 사투리 지역 분류 모델

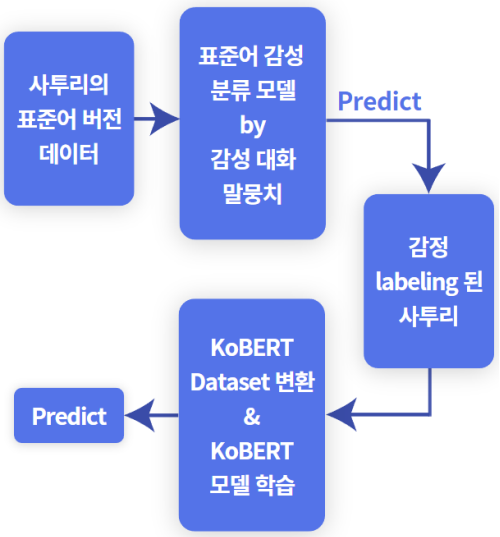


지역별로 labeling된 training set 데이터를 기반으로 지역 분류 모델을 만들었습니다. 800만개의 데이터 중 text의 길이가 30이 넘는 text만을 골라내어 사용했으며 그 중 랜덤으로 15만개를 골라 학습에 사용할 training set으로 만들었습니다.

이를 BERT 모델의 input에 알맞은 형태로 바꿔준 다음, tokenizer을 통하여 text의 토큰화를 진행했습니다. 변환이 완료된 데이터를 pytorch 기반으로 설계한 BERTClassifier에 학습시켜 최종 사투리 지역 분류 모델을 만들었습니다.

총 8번의 epoch를 거쳐 학습을 진행했으며 학습 시간이 오래 걸려 2번씩 나눠서 학습을 진행했습니다.

II. 사투리 감정 분류 모델



감정이 labeling된 표준어 데이터를 이용하여 먼저 표준어 감정 분류 모델을 만들었습니다. 8만개의 데이터 중 랜덤으로 5만개를 추출하여 총 10번의 epoch를 거쳐 KoBERT에 학습시켜 표준어 감정 분류 모델을 만든 후, 이를 이용하여 사투리 데이터의 표준어 버전 데이터의 감정 labelling을 진행했습니다.

이렇게 6가지의 감정만 labeling되어 총 6가지 label로 분류된 사투리 데이터의 표준어 버전 데이터와, 여기에 긍정/부정을 합쳐 총 12가지 label로 분류된 사투리 데이터의 표준어 버전

데이터 set을 각각 KoBERT에 학습시켜 사투리 감정 분류 모델을 만들었습니다.

총 10번의 epoch를 거쳐 학습을 진행했으며 epoch를 5번씩 나눠서 학습을 진행했습니다.

4.4 Results

두 모델 모두 학습 과정에서 epoch 4,5를 넘어서자 training accuracy는 증가하되 test accuracy가 더 이상 증가하지 않는 경향을 보였고, training loss도 oscillate하는 경향을 보였습니다.

I. 사투리 지역 분류 모델

Epoch를 8번 모두 진행한 결과 걸린 시간은 약 40시간 정도이며 test data에 대한 accuracy는 78.7%를 보였으며 f1 score는 0.601를 기록했습니다.

II. 사투리 감정 분류 모델

감정만을 labeling하여 6가지 label로 분류한 모델에서는 epoch를 10번 모두 진행한 결과 test set에 대한 accuracy는 60.5%를 보였으며 f1 score는 0.474를 기록했습니다.

감정에 긍정/부정을 추가하여 12가지 label로 분류한 모델에서는 epoch를 10번 모두 진행한 결과 test set에 대한 accuracy는 47.2%를 보였으며 f1 score는 0.364를 기록했습니다.

마지막으로 이중 부정과 비꼬기 등의 어려운 예제들에 대해서 검증을 진행한 결과, 6가지 label로 분류한 모델에서는 accuracy가 22%를 기록했으며 12가지 label로 분류한 모델에서는 accuracy가 20%를 기록했습니다.

	Accuracy	**F1 score**
사투리 지역 분류 모델	0.787	0.601
사투리 감정 분류 모델(label 6개)	0.605	0.474
사투리 감정 분류 모델(label 12개)	0.472	0.364

III. 표준어 감정 분류 모델과의 성능 비교

표준어 감정 분류 모델을 통해 사투리 감정분류를 진행한 결과 accuracy는 42.7%를 보였으며 f1 score는 0.329를 기록했습니다.

사투리 감정 분류 모델을 통해 사투리 감정분류를 진행한 결과 accuracy는 53.2%를 보였으며 f1 score는 0.41를 기록했습니다.

	Accuracy	F1 score
표준어 감정 분류 모델	0.427	0.329
사투리 감정 분류 모델(label 6개)	0.532	0.41

이렇게 모두 모델을 학습시키고 검증한 결과 지역 분류 모델은 비교적 좋은 결과를 얻은 반면 감정 분류 모델은 6가지 label로 분류한 모델과 12가지 label로 분류한 모델은 좋지 않은 성능을 보였습니다. 또한 직접 만든 이중 부정과 비꼬기 data set에 대해서도 두 모델 모두 좋지 않은 성능을 보이는 것으로 보아 긍정/부정 라벨링을 포함시킨 것이 이중부정과 비꼬기 등의 어려운 예제 분류에 도움이 되지 않았다고 결론 지을 수 있습니다. 마지막으로 표준어 감정 분류 모델과의 성능 비교에서 사투리 감정 분류 모델의 accuracy와 f1 score이 더 높은 것으로 보아 사투리 학습이 사투리 감정 분류에 도움이 되었다 결론 지을 수 있습니다.

감성 분류 모델이 낮은 성능을 보인 것에 대한 이유로 우선 데이터셋 크기의 한계가 있다 생각합니다. 사투리 지역 분류 모델은 15만개의 데이터로 5가지 label을 분류하는 모델을 만들어 비교적 높은 accuracy를 보였던 것에 반해 사투리 감정 분류 모델에서는 5만개의 데이터로 6가지 감정 label 또는 12가지 감정 label로 분류해야 했습니다. 따라서 학습에 충분하지 않은 크기의 학습 데이터셋 사용으로 인해 모델의 성능이 저하됐다고 생각합니다.

사투리 데이터의 감정 labeling 과정에서도 허점이 존재했습니다. 사투리 데이터의 감정 labeling 과정에서 저희 팀은 표준어 감정 데이터셋을 KoBERT 모델에 학습시켜 표준어 감정 분류 모델을 만든 후, 이를 이용해서 사투리의 표준어 버전 데이터에 감정을 labeling 했습니다. 그러나 이 과정에서 표준어 감정 분류 모델의 test accuracy가 58%를 기록하며 60%도 채 되지 않은 낮은 성능으로 인해 labeling 과정에서 정확하지 않은 labeling이 진행되었습니다. 또한 human labeling을 통한 수정 과정에서도 감성 표현의 영역이 주관적인 영역이라 정답을 매기기 애매했기 때문에 이러한 문제들로 예상보다 낮은 test accuracy와 f1 score이 나왔다 생각합니다.

5. Conclusion

저희 팀은 KoBERT 모델을 이용하여 AI Hub의 사투리 데이터를 통한 지역 분류와 감정 분류를 진행해 보았습니다. 사투리의 지역 분류 모델은 labeling되어 있는 AI Hub의 지역별 사투리 데이터를 사용하였으며 이를 KoBERT 모델에 학습시켜 5가지 지역의 사투리를 분류할 수 있는 모델을 개발했습니다. 모델은 test data에 대하여 78.7%의 accuracy를 기록했으며 f1 score은 0.601을 기록하며 비교적 높은 성능을 보였습니다.

사투리 감정 분류 모델은 우선 감정이 labeling 된 표준어 데이터를 KoBERT 모델에 학습시킨 후 이를 이용하여 사투리 데이터에 감정 labeling을 진행, 이를 다시 새로운 KoBERT 모델에 학습시켜 최종 사투리 감정 분류 모델을 개발했습니다. 이 과정에서 6가지 감정 label로 분류하는 모델과, 이중 부정과 비꼬기 등의 사투리까지 효과적으로

분류해보기 위해 텍스트 전체의 긍정/부정 label을 결합하여 12가지 label로 분류하는 모델을 각각 개발했습니다. 6가지 감정으로 분류했을 경우 test data에 대해서 60.5%의 accuracy를 기록했으며, f1 score는 0.474를 기록했습니다. 긍정/부정을 결합하여 12가지 label로 분류했을 경우 test data에 대해서 47.2%의 accuracy를 기록했으며, f1 score는 0.364를 기록했습니다. 마지막으로 직접 만든 이중 부정 및 비꼬기 예제들에 대해서는 두 모델 각각 22%, 20%의 정확도를 보이며 비슷하게 낮은 정확도를 보였습니다.

마지막으로 표준어 감정 분류 모델과의 성능 비교에서 사투리 감정 분류 모델이 사투리 감정 예측에서 더 좋은 성능을 보이면서 사투리 data 학습이 유의미함을 보였습니다.

사투리 감정분류 모델의 성능이 낮게 나온 이유로 학습 데이터셋 크기의 한계, 사투리 데이터의 감정 labeling 과정에서의 허점 등 다양한 이유가 있다 생각합니다. 따라서 학습 데이터셋의 크기를 크게 만들어 학습을 진행시키거나, 사투리 데이터에 대해서 더욱 정확한 labeling이 이루어진다면 더 나은 성능의 모델을 기대해볼 수 있을 것이라 예상합니다.

References

[1] Multi-Dialect Arabic BERT for Country-Level Dialect Identification 2020. Bashar Talafha

[2] Demographic Dialectal Variation in Social Media: A Case Study of African-American English Su Lin Blodgett Jørgensen, et al., 2016

[3] pakr ku byeng, Jejueo Datasets for Machine Translation and Speech Synthesis

[4] 1998, 최명옥, 국어의 방언 구획

[5] An Empirical Study of Tokenization Strategies for Various Korean NLP Tasks, 2020, Kyubyong Park