자기주도프로젝트

AJOU SOFTCON 2020-1

SentencePiece*를 이용한 유튜브 댓글의 감정분류 문제 성능 개선 연구

이름 임윤진 지도교수 손경아 교수님

연구배경

소셜 네트워크 텍스트 데이터의 특징

대체로 짧은 문장 길이를 가진다. 또한, 다양한 종류의 이모티콘, slang 및 약어와 비문을 사용하기 때문에 문자를 이용한 여러가지 의사 표현 방식이 존재한다.

선행 연구 및 개선 방향

구분	선행 연구*	개선 방향
전처리 작업	중복 단어 제거, 불용어 제거, 철자 교정, 숫자 및 특수문자 제거, 지정 문자 교체 등.	작업 속도를 고려하여 이모티콘과 불용어 제거를 제외한 전처리 작업을 최소화.
단어 사전 생성	대용량 단어 사전을 사용. Slang 단어 추출 및 번역 과정을 거침.	단어 사전의 크기를 줄여도 Out Of Vocabulary 문제를 효과적으로 처리할 수 있음.
감정 분류	모든 단어의 Sementic Orientation(SO)를 측정하고 문장의 감정 점수를 계산.	Sequential Data의 특성을 고려한 Recurrent neural network를 이용.

단어 사전 생성 및 토큰화 기능 제공 Library 비교

구분	Keras Tokenizer	SentencePiece
일반적인 입력 데이터	전처리 작업한 데이터	전처리 하지 않은 raw 데이터
문장 토큰화	띄어쓰기 단위로 단어 분리. '.' 사이 띄어쓰기에 대한 정보를 담지 못함.	Byte-pair-encoding, unigram language model 등의 분리 알고리즘을 지원.
문장 복원	문장을 완벽하게 복원하는 데 한계가 있음.	띄어쓰기를 '_'로 변환하여 손실 없이 원본 복원 가능. Decoder 지원.

[1] SentencePiece: Taku, K. &John, R., 2018, SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing, arXiv:1808.06226v1 [cs.CL]
[2] 선행 연구: Fazal M. K., Aurangzeb K., Shakeel A., &Muhammad Z. A., 2014, Lexicon-Based Sentiment Analysis in the Social Web, J. Basic. Appl. Sci. Res., 4(6)238-248

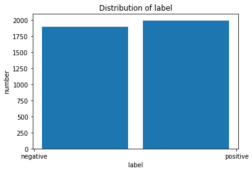
연구 진행 과정

연구 목표

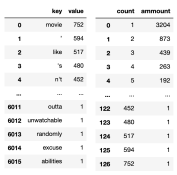
수많은 단어 변칙이 존재하는 유튜브 댓글 데이터를 이용하여 Out Of Vocabulary 문제를 해결하고, 이진 감정 분류의 성능을 개선하고자 함.

1. 데이터

뷰티, 영화, 교육, 음식 등의 카테고리에서 수집한 총 3884개의 댓글 데이터
긍정 : 1992개
부정: 1892개



2. 이모티콘 및 불용어 제거



	key	value
0	movie	752
1	.	594
2	like	517
3	's	480
4	n't	452
...	...	...
6011	outta	1
6012	unwatchable	1
6013	randomly	1
6014	excuse	1
6015	abilities	1

6016 rows × 2 columns

	count	ammount
0	1	3204
1	2	873
2	3	439
3	4	263
4	5	192
...	...	...
122	452	1
123	480	1
124	517	1
125	594	1
126	752	1

127 rows × 2 columns

3. 댓글 평균 길이와 사용 단어 수 계산

총 6016개의 단어가 존재하고, 단어 빈도수를 기준으로 나열한 결과, 전체의 약 70%를 차지하는 단어 4300개를 단어 사전의 크기로 지정함.

4. Kera Tokenizer, SentencePiece를 이용하여 각각 단어사전 생성 후 문장 벡터화

원본 [actually reaaally gooood movie ! 9/10]

토큰화 [_actually, _rea, a, ally, _goo, ood, _movie, _!, _9, /10]

벡터화 [74, 340, 79, 487, 2303, 548, 10, 6, 657, 627]

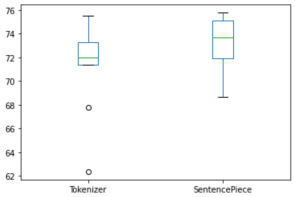
결과 및 분석

학습 모델 요약

```
Model: "sequential_29"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_29 (Embedding)     (None, None, 100)         430000
_________________________________________________________________
lstm_87 (LSTM)               (None, None, 100)         80400
_________________________________________________________________
lstm_88 (LSTM)               (None, None, 100)         80400
_________________________________________________________________
dropout_29 (Dropout)         (None, None, 100)         0
_________________________________________________________________
lstm_89 (LSTM)               (None, 100)               80400
_________________________________________________________________
dense_29 (Dense)             (None, 2)                 202
=================================================================
Total params: 671,402
Trainable params: 671,402
Non-trainable params: 0
```

총 3 층의 LSTM을 쌓고 첫번째 LSTM 층에서 current_dropout = 0.4, 두번째와 세번째 층 사이에 dropout = 0.5로 설정하였다.
10-fold cross validation을 적용하여 학습한다.

학습 결과



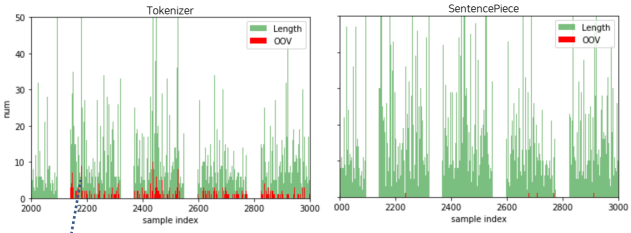
10 validation set의 accuracy 측정 결과

Tokenizer
평균 71.39%, 표준편차 +/- 3.65%

SentencePiece
평균 73.30%, 표준편차 +/- 2.21%

결론: SentencePiece가 Tokenizer보다 성능이 조금 더 좋고, 표준편차가 더 작은 것을 확인할 수 있었다.

결과 분석



위 그래프는 각 문장의 길이와 OOV의 개수를 bar plot으로 시각화한 그래프이다. Keras Tokenizer library을 사용할 때의 OOV의 개수가 훨씬 많이 분포하는 것을 알 수 있었다.

S: Sentencepiece T: Tokenizer

index	Method	Text	Predict
99	S	ok 40 secondish mark sarcastic badd actor	0
	T	ok <UKN> <UKN> mark <UKN> <UKN> actor	1
169	S	4:37 c-c-c-c-combo breaker ! ! ! !	1
	T	4 <UKN> c c c c <UKN> <UKN>	0
347	S	owowowow , smiled :)	1
	T	<UKN> <UKN>	0

위의 표는 Sentencepiece model은 맞게 분류하고 Tokenizer model은 틀리게 분류한 경우를 찾아서 텍스트를 분석한 결과 중 일부이다.
Sentencepiece 방식은 Tokenizer 방식과 달리 물음표, 느낌표, 온점 등과 같은 특수문자를 data로 사용하여 '☺'와 같은 감정 표현을 포함할 수 있다. 또한, 단어 사전의 크기 제한으로 포함되지 못했던 단어를 표현할 수 있다. 예를 들어, <UKN> 대신 'owowowow', 'smiled', 'breaker', 'secondish' 등의 단어를 포함할 수 있다.

결론: SentencePiece가 OOV 처리에 더 효과적이다.