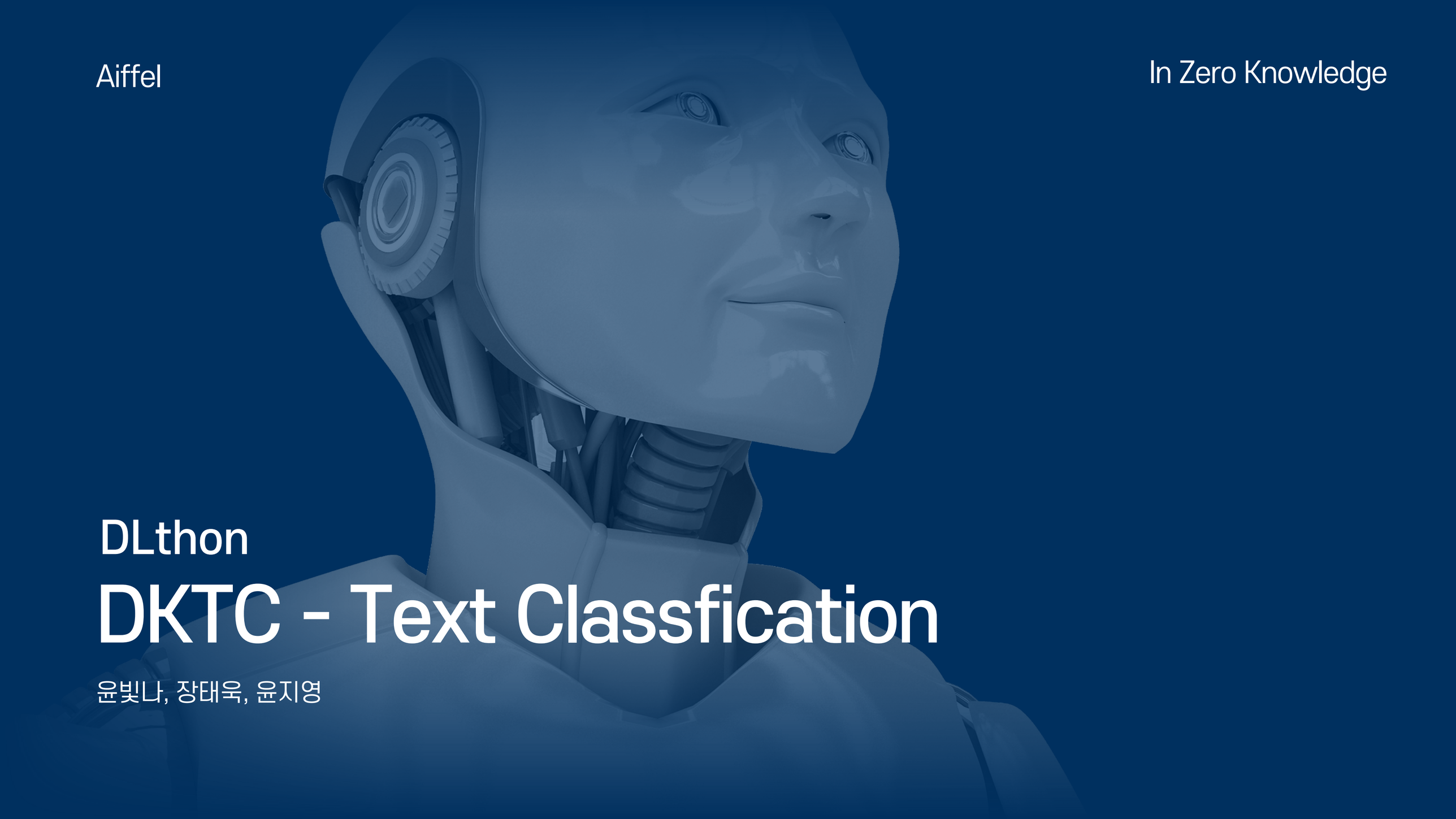
Aiffel

In Zero Knowledge



DLthon

DKTC - Text Classfication

윤빛나, 장태욱, 윤지영

Contents

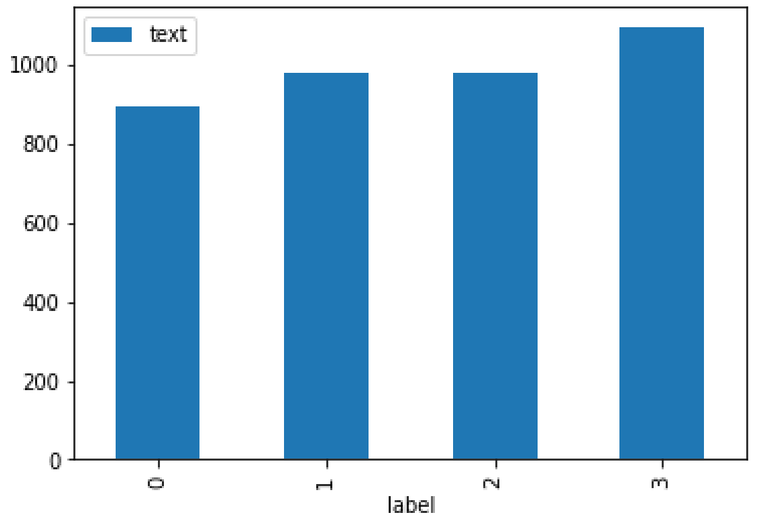
Data - DKTC

Dataset of korean Threatening Conversations

총 3950

class	Class	total
협박	0	896
M갈취	1	981
직장 내 괴힘힘	2	979
기타 괴롭힘	3	1094

	idx	class	conversation
0	0	협박 대화	지금 너 스스로를 죽여달라고 애원하는 것인가?\n 아닙니다. 죄송합니다.\n 죽을 ...
1	1	협박 대화	길동경찰서입니다.\n9시 40분 마트에 폭발물을 설치할거다.\n네?\n똑바로 들어 ...
2	2	기타 괴롭힘 대화	너 되게 귀여운거 알지? 나보다 작은 남자는 첨봤어.\n그만해. 니들 놀리는거 재미...
3	3	갈취 대화	어이 거기\n예??\n너 말이야 너. 이리 오라고\n무슨 일.\n너 옷 좋아보인다?...
4	4	갈취 대화	저기요 혹시 날이 너무 뜨겁잖아요? 저희 회사에서 이 선크림 파는데 한 번 손등에 ...



Data - DKTC

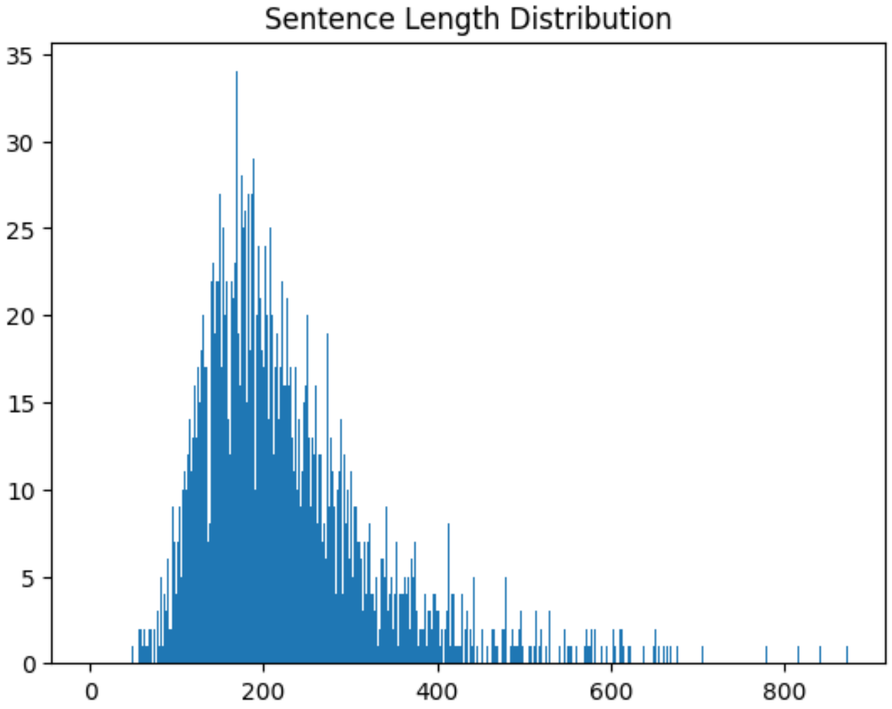
Dataset of korean Threatening Conversations

DKTC 훈련 데이터를 이용해
협박, 갈취, 직장 내 괴롭힘, 기타 괴롭힘 4가지 대화 유형 Class

문장의 최단 길이: 41

문장의 최장 길이: 874

문장의 평균 길이: 226



각 키워드 추출 TF-IDF

Top 10 keywords for label

'0': ['소릴' '제삿날' '서장' '죽여서' '주가' '지나' '진하' '불려' '요구사항' '그러']
'1': ['삼송' '아팠' '날려' '소릴' '진하' '윗사람' '담당' '그럼' '한판' '그래']
'2': ['주가' '드디어' '회색' '사부인' '어치' '과장' '자니까' '식조' '담당' '지나']
'3': ['그리구' '진하' '담당' '보낼게' '못내' '그래' '소릴' '그냥' '잘라서' '한판']

EDA Conclusion

데이터 분석 후 고려해 볼 사항

1. Multi classificaiton
2. Text Classification
3. Long Sequence
4. Korean Language

데이터 전처리

Data Preprocessing

중복 제거

104개의 중복 데이터

	idx	class	conversation
392	392	3	병신이 아이스크림 먹게 돼 있냐?\n난 먹으면 안 돼? 그만 좀 해.\n당연히 안 ...
523	523	2	과장님. 저 이번에 휴가 좀 갔다와도 되겠습니까.?\n휴가? 왜??\n좀 쉬다가 오...
789	789	3	야 얘 이 쪽 손가락 세 개밖에 없다\n엥 손가락이 세개밖에 없을 수가 있어?\n봐...
824	824	3	이야 내동생 여자친구한테 편지쓰네?\n 아 형 돌려줘.\n 어디보자. 사랑하는 여친...
869	869	2	지현씨 나 소개팅 좀 시켜줘봐\n네? 저 주변에 아는 사람이 없어서요\n아 상사라 ...
...	...	...	...
3797	3797	3	너 이번 방학 때 쌍꺼풀 수술 하고왔지?\n아닌데?\n아니긴 뭐가 아니야. 눈이 이...
3798	3798	3	안녕하세요 지금 먹방 촬영중인데 촬영가능할까요?\n안돼요\n한번만 안될까요?\n안돼...
3855	3855	3	그 소문 진짜야? 너가 다른 애들 뒷담화하고 다녔다며?\n응? 나 그런 적 없는데?...
3874	3874	3	야 니 왤캐 못생겼냐?\n뭐라그랬냐?\n으 나 보고 말하지마 니 얼굴보면 토나올거 ...
3928	3928	3	새파랗게 젊은게 어디 여길 앉아있어\n저 임산부에요\n사지 멀쩡한게! 임신이 벼슬이...

104 rows × 3 columns

토큰화

Mecab, SentencePiece, Okt

cleaned_conversation	okt	mecab
지금 너 스스로를 죽여달라고 애원하는 것인가 아닙니다 죄송합니다 죽을 거면 혼자...	[지금, 너, 스스로, 를, 죽여, 달라, 고, 애원, 하는, 것, 인가, 아닙니다...	[지금, 너, 스스로, 를, 죽여, 달, 라고, 애원, 하, 는, 것, 인가, 아닙...
길동경찰서입니다 시 분 마트에 폭발물을 설치할거다 네 똑바로 들어 한번만 더 얘기한...	[길동, 경찰서, 입니다, 시, 분, 마트, 에, 폭발물, 을, 설치, 할거다, 네...	[길동, 경찰서, 입니다, 시, 분, 마트, 에, 폭발물, 을, 설치, 할, 거, ...
너 되게 귀여운거 알지 나보다 작은 남자는 첨봤어 그만해 니들 놀리는거 재미없어 지...	[너, 되게, 귀여운거, 알, 지, 나, 보다, 작은, 남자, 는, 첨, 봤어, 그...	[너, 되게, 귀여운, 거, 알, 지, 나, 보다, 작, 은, 남자, 는, 첨, 봤...
어이 거기 예 너 말이야 너 이리 오라고 무슨 일 너 옷 좋아보인다 얘 돈 좀 있나...	[어이, 거기, 예, 너, 말, 이야, 너, 이리, 오라, 고, 무슨, 일, 너, ...	[어, 이, 거기, 예, 너, 말, 이, 야, 너, 이리, 오, 라고, 무슨, 일,...
저기요 혹시 날이 너무 뜨겁잖아요 저희 회사에서 이 선크림 파는데 한 번 손등에 발...	[저기, 요, 혹시, 날, 이, 너무, 뜨겁잖아요, 저희, 회사, 에서, 이, 선크...	[저기, 요, 혹시, 날, 이, 너무, 뜨겁, 잖아요, 저희, 회사, 에서, 이, ...
...	...	...
준하야 넌 대가리가 왜이렇게 크냐 내 머리가 뭐 밥먹으면 대가리만 크냐 너는 아니 ...	[준, 하야, 넌, 대가리, 가, 왜, 이렇게, 크냐, 내, 머리, 가, 뭐, 밥,...	[준, 하, 야, 넌, 대가리, 가, 왜, 이렇게, 크, 냐, 내, 머리, 가, 뭐...
내가 지금 너 아들 김길준 데리고 있어 살리고 싶으면 계좌에 억만 보내 예 선생님 ...	[내, 가, 지금, 너, 아들, 김길준, 데리, 고, 있어, 살리고, 싶으면, 계좌...	[내, 가, 지금, 너, 아들, 김길준, 데리, 고, 있, 어, 살리, 고, 싶, ...

불용어 제거

['은','는','이','가', ...]

```
# 불용어 리스트 정의
stop_words = [
    "은", "는", "이", "가", "을", "를", "으로", "에", "에서",
    "하다", "하는", "된", "나", "고", "와", "과", "너무", "있는", "같은",
    "하는", "것", "그", "그리고", "하는", "하는", "말", "다", "그런",
    "저", "제", "네", "들", "이다", "입니다", "한다", "합니다", "돼", "이런",
    "저런", "그런", "할", "있는", "또는", "아니", "합니다", "있다", "왜",
    "여기", "저기", "그렇게", "그러니까", "뭐", "하나", "때", "수", "등", "아주",
    "너", "내", "야", "안", "니", "도"
```

모델 선정

DeepLearning Model for text classificaiton

LSTM

장점:
장기 의존성 학습이 가능하므로 긴 시퀀스 데이터에 적합.

단점:
많은 매개변수로 인해 학습 시간이 오래 걸릴 수 있음.
복잡한 모델 구조로 인해 메모리 사용량이 많음.

GRU

장점:
LSTM에 비해 간단한 모델 구조로 빠른 학습 가능.
적은 매개변수로 인해 메모리 효율적.

단점:
데이터의 복잡한 패턴을 학습하는 능력이 부족할 수 있음.

CNN1D

장점:
국부적인 특징 학습으로 인한 효과적인 시퀀스 처리.
병렬 계산으로 인해 학습 및 예측 속도가 빠름.

단점:
장기 의존성 모델링에서는 RNN에 비해 한계가 있음.
시퀀스 데이터의 길이에 대한 제약이 있을 수 있음.

Training & Evaluate

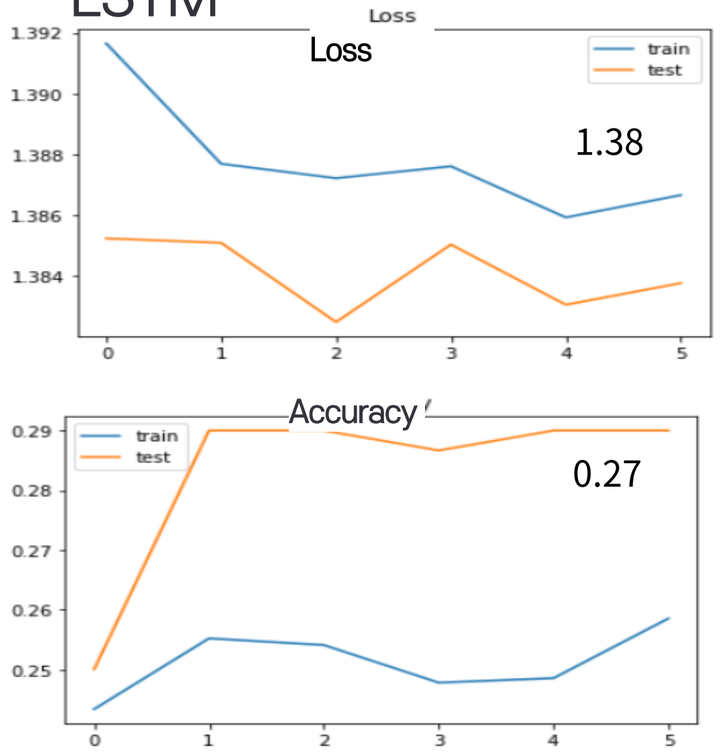
DeepLearning Model for text classificaiton

Option

Tokenizer : Mecab, Optimizer : Adam, Batch_size : 64, 32

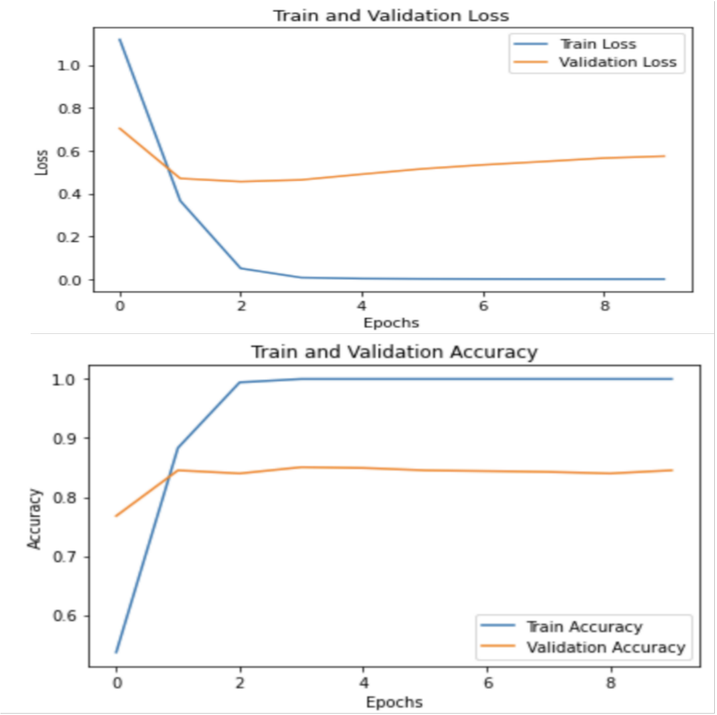
Accuarcy & Loss visualization

LSTM



GRU

CNN1D



Limitation

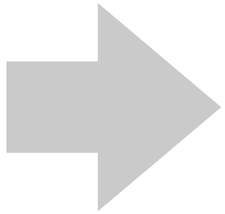
모델의 한계

LSTM

GRU

CNN1D

" 모델의 한계 "



장거리 종속성을 유지하기 어려움

복잡한 패턴이나 의미론적 의미 파악의 한계

전역 문맥을 이해하기 어려움

Limitation

모델의 한계

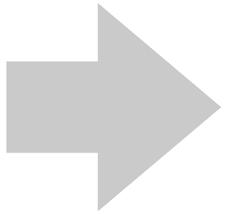
" 사전 훈련되어 풍부한 문맥화된 단어 표현
전체 문장을 고려하여 전체 문장의 맥락에서
단어의 의미를 이해하는 능력 "

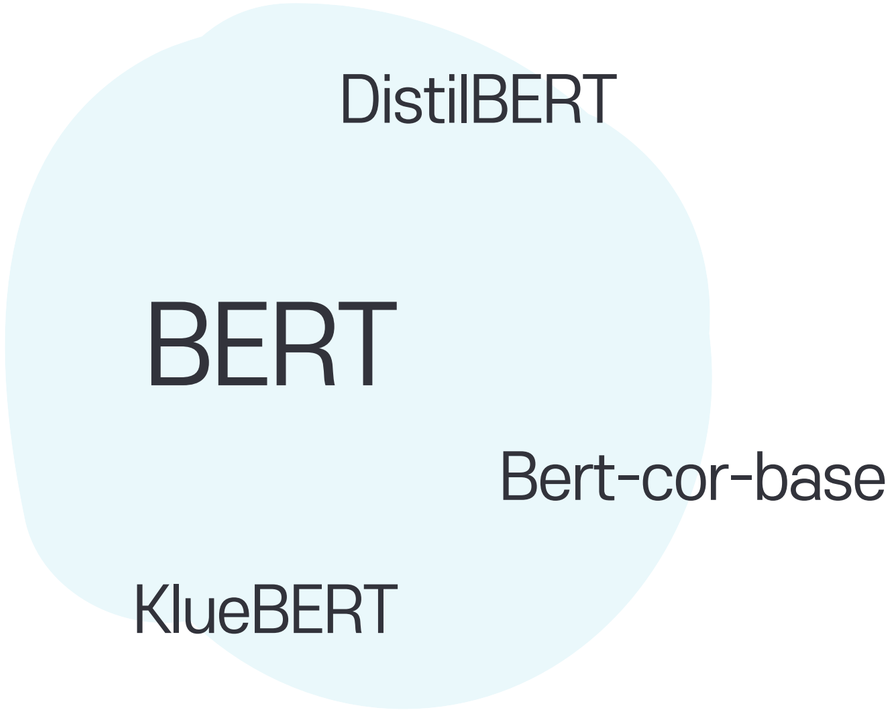
LSTM

GRU

CNN1D

" 모델의 한계 "





Why DistilBERT , KlueBERT?

Pre-trained model - BERT

DistilBert(영어기반)

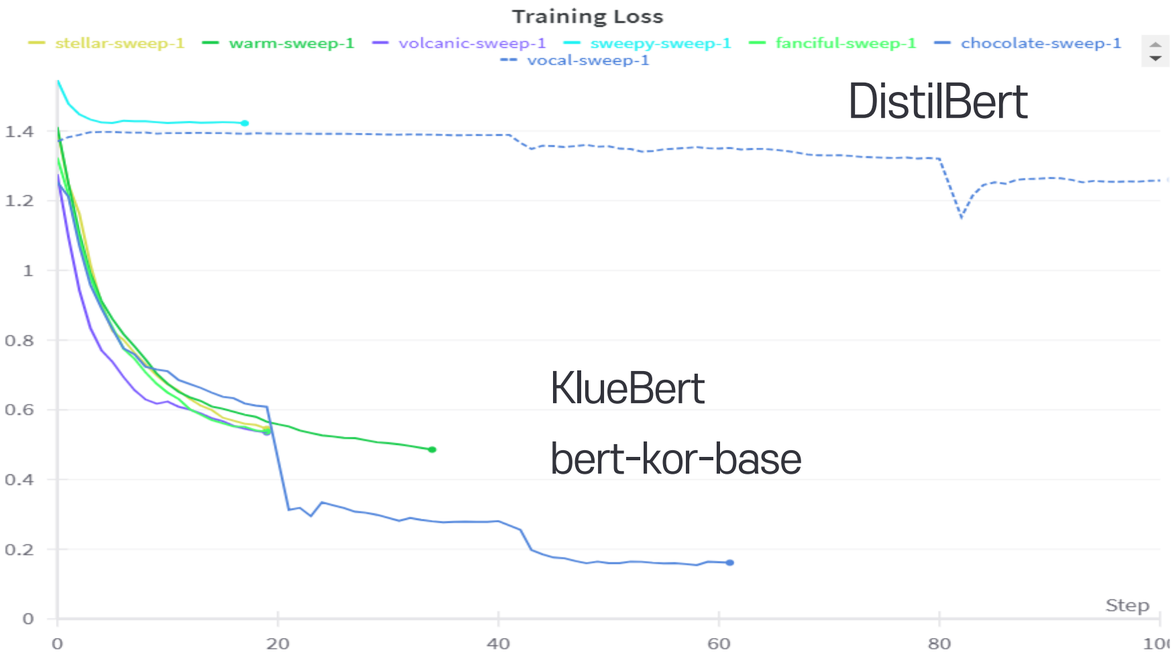
- 간결한 구조
- 작은 모델 크기, 빠른 학습 속도
- 매개변수가 기존 Bert의 절반
- Bert 버금가는 성능

KlueBert

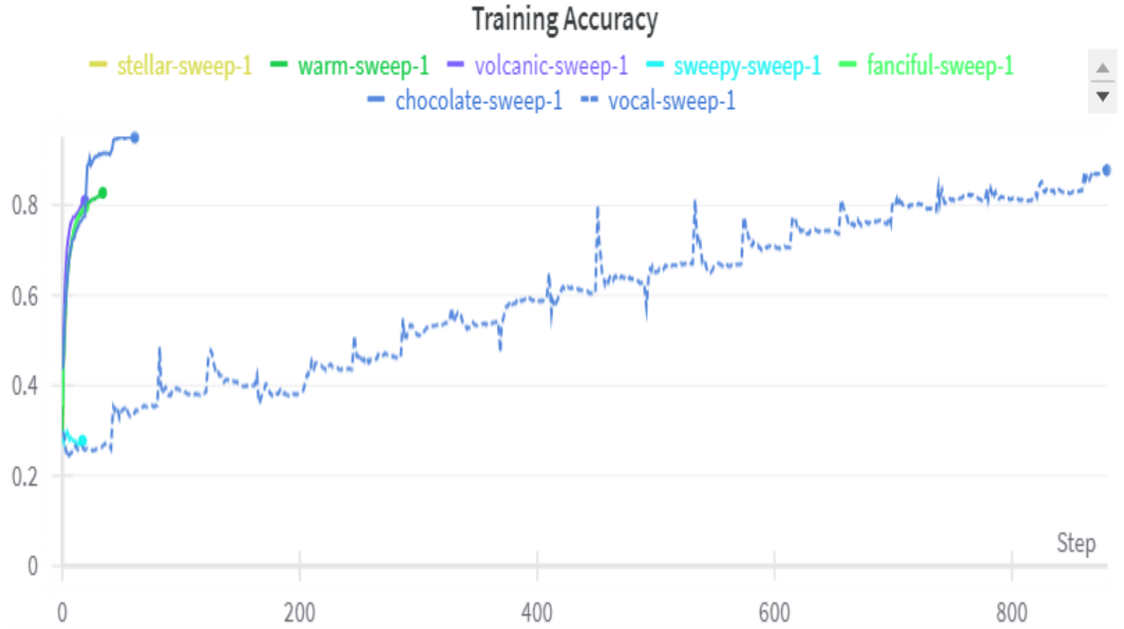
- KLUE(Korean Language Understanding Evaluation) 벤치마크에서 사전 훈련
- 한국어 자연어처리 작업에서 우수한 성능
- 전이학습 효과

BERT - DistilBert , KlueBert, Bert-Kor-base(비슷한 한국어 모델?)

Loss visualization



Accuracy



도메인 특화

- KlueBERT는 한국어 텍스트의 도메인 특성을 고려하여 사전 훈련되었기 때문에 작업에서 우수한 성능을 보여 주는 것으로 예상

- 같은 시간 대비 KlueBERT로 학습하는 것이 훨씬 효율이 높았음
- 한국어 도메인을 가진 Klue-Bert와 Bert-kor-base를 비교했을 때 Bert-kor-base가 더 높은 성능을 보여주었음

BERT - DistilBert , KlueBert, Bert-Kor-base

Pre-trained model Summary

Model	Optimizer	batch_size	Tokenizer	evaluate	accuracy	loss
DistilBert	Adam	8	distil/bert-base	accuracy	0.60	0.603
KlueBert	Adam	8	klue/bert-base	accuracy	0.86	0.60
KlueBert	Adam	16	klue/bert-base	accuracy	0.80	0.53
KlueBert_aug	Adam	16	klue/bert-base	accuracy	0.86	0.55
Bert-kor-base	Adam	16	kykim/bert-kor-base	accuracy	0.87	0.54

Data Augmentation Experimental

가설

Bert 모델은 문맥도 중요하지만 자주 나오거나 주요한 단어에 집중하기도 하니까 그냥 불용어 제거하고 토큰 순서만 조금 바꿔서 데이터를 증강해서 사용해도 도움이 되지 않을까?

- 토큰 순서를 바꾸면 문맥적이 요소 파괴되어 문제가 있을 수 있지만 데이터가 작기 때문에 시도
- 결과 기존 모델 중에 같은 조건으로 돌렸는데 데이터 증강을 한 것이 더 높은 성능을 보여줬다. 그 이유는 데이터가 적은 상황이었고, 문맥은 파괴되더라도 중요한 역할을 하는 토큰은 남아 있기 때문에 자주 나오는 키워드를 통해서 도움이 되지 않았을까 하고 예상해본다.

과정

1. 불용어 제거
2. Mecab으로 토큰화
3. 토큰 위치 조정
4. ' '.join()
5. 기존의 데이터와 Concat

총 데이터 7900개

CNN1D

```
Epoch 1/10
94/94 [==============================] - 2s 16ms/step - loss: 1.0599 - accuracy: 0.5833 - val_loss: 0.7131 - val_ac
curacy: 0.7493
Epoch 2/10
94/94 [==============================] - 1s 14ms/step - loss: 0.4210 - accuracy: 0.8593 - val_loss: 0.4575 - val_ac
curacy: 0.8400
Epoch 3/10
94/94 [==============================] - 1s 14ms/step - loss: 0.1269 - accuracy: 0.9730 - val_loss: 0.4281 - val_ac
curacy: 0.8680
Epoch 4/10
94/94 [==============================] - 1s 14ms/step - loss: 0.0245 - accuracy: 0.9990 - val_loss: 0.4572 - val_ac
curacy: 0.8640
Epoch 5/10
94/94 [==============================] - 1s 14ms/step - loss: 0.0068 - accuracy: 1.0000 - val_loss: 0.4794 - val_ac
curacy: 0.8640
Epoch 6/10
94/94 [==============================] - 1s 14ms/step - loss: 0.0036 - accuracy: 1.0000 - val_loss: 0.4958 - val_ac
curacy: 0.8640
```

모델 학습

W&B 결과

submission_sp

file_name	class
t_000	1
t_001	2
t_002	2
t_004	3
t_005	0
t_006	2
t_007	1
t_009	1
t_010	0

Test Accuracy Rate:	24.93
Test Error Rate:	75.07
accuracy	0.25067
best_epoch	5
best_val_loss	1.38567
epoch	8
loss	1.38892
val_accuracy	0.24933
val_loss	1.39078

Summary

Model	Optimizer	batch_size	Tokenizer	evaluate	accuracy	loss
DistilBert	Adam	8	distil/bert-base	accuracy	0.60	0.603
KlueBert	Adam	8	klue/bert-base	accuracy	0.86	0.60
KlueBert	Adam	16	klue/bert-base	accuracy	0.80	0.53
KlueBert_aug	Adam	16	klue/bert-base	accuracy	0.86	0.55
Bert-kor-base	Adam	16	kykim/bert-kor-base	accuracy	0.87	0.54
LSTM	Adam	64	Sentencepiece/Okt/Mecab	accuracy	0.27	1.38
GRU	Adam	32	Sentencepiece/Okt/Mecab	accuracy	0.27	1.38
CNN1D	Adam	32	Sentencepiece/Okt/Mecab	accuracy	0.25	1.38

Conclusion

LSTM vs. GRU vs. CNN1D:

정확하게 어떤 이유에서인진 알 수 없지만 모델 학습시에는 굉장히 저조한 학습 결과를 보여줬는데 최종적으로 분류 성능을 평가했을 때 CNN에서 Sentencepiece 전처리 한 결과물은 가장 높은 성능을 나타냈으며 Okt, Mecab토크나이저도 70%대의 정확도를 보여주었다. CNN1D는 LSTM에 비해 학습 속도가 상대적으로 빠르지만, 장기 의존성 모델링에서는 한계가 있을 수 있기에 최종적인 결과물에서도 더 좋은 결과를 보였다.

최종 - 리더보드(CNN1D-SP) : 0.85

DistilBERT vs. KlueBERT:

DistilBERT도 우수한 성능을 보였지만, 한국어 텍스트 분류에서는 KlueBERT가 더 높은 성능을 보였다. 이를 통해 도메인에 특화된 데이터로 훈련하는 것이 성능 향상에 중요하다는 점을 확인할 수 있었다.

최종 - 리더보드 : 0.875

회고

GPT를 활용한 문자 데이터 증강 실험

GPT를 활용한 문자 데이터 증강을 시도해보고 싶다. 모델 성능 향상을 위해 새로운 텍스트 생성 방법을 테스트해 보고, 증강된 데이터를 활용하여 데이터 다양성을 높여보고 싶다.

다양한 BERT 모델 시도

다양한 BERT 모델을 활용해 보면서 좀 더 모델 별로 강점과 약점이 무엇인지 고려해서 사용했다면 더 좋았을 것 같다는 아쉬움이 남는다. 이후에는 사용해보면서 모델의 강점과 약점을 탐색하여 특정 태스크에 최적화된 모델을 찾아내는 시도를 해보고 싶다.

LSTM과 CNN 모델 학습 문제

CNN에서 모델 학습은 저조했는데 분류 성능에는 좋은 결과가 있었다는 점이 아직 의문이 생기며 아쉬움도 많이 남는다. 하이퍼파라미터도 바꿔보고 학습 epoch도 100으로 시도해봤지만 급격하게 증가하며 overfitting으로 accuracy가 1로 고정되는 문제가 지속적으로 발생되었다. 어떤 문제가 있었던건지 더 상세하게 공부해보고 확인해봐야 할 것 같다.

감사합니다!
아이펠 그루들 화이팅!!