NH투자증권

AI야, 진짜뉴스를 찾아줘

NH 투 자 증 권

오끼동 팀 옥창원, 윤영주

NH투자증권

CONTENTS

01

팀원 소개

01 팀원 소개

팀원 소개

- 이름 : 옥창원
- 소속 : 한양대학교 산업공학과 대학원 3기
- 닉네임 : 오끼동
- Github : https://github.com/Chuck2Win

- 이름 : 윤영주
- 소속 : 한양대학교 산업공학과 대학원 1기
- 닉네임 : 난브래드
- Github : https://github.com/YoungjuYoon

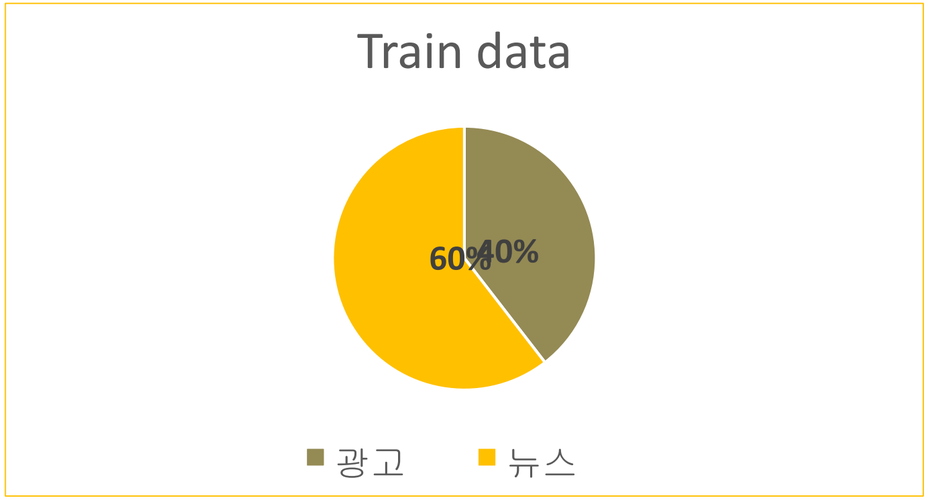
02

아이디어

02 아이디어

EDA

- Feature : n_id, Date, Title, Content, order, info
- Number of data : 118,746
- Number of News : 71,814 (60%)
- Number of AD : 46,932 (40%)



- n_id, Date, Title, order은 info 와 유의미한 관계를 갖지 못함
- Content, Content length 는 유의미함
  - 진짜 뉴스의 경우, 더 길
  - 길이가 512 이상인 경우, 진짜 뉴스임.
- 뉴스와 광고의 비율이 6:4
  -> Imbalance 하므로 이를 Balance하게 맞춰주는 과정이 필요함

	Train data			Test data
	전체	진짜 뉴스	가짜 뉴스	전체
평균 길이	27.54	31.91	20.92	29.36
최대 길이	1,684	1,684	512	1,895
하위 95%의 길이	59	56	36	63

02 아이디어

데이터 전처리 - 불용어 제거

같은 Title임에도 content가 다름 -> Title은 무의미함

n_id	date	title	ord	info
NEWS02033	20200413	에이수스, 인텔 10세대 CPU 장착한 게이밍노트북 7종 출시	14	0
NEWS02033	20200413	에이수스, 인텔 10세대 CPU 장착한 게이밍노트북 7종 출시	15	0
NEWS02033	20200413	에이수스, 인텔 10세대 CPU 장착한 게이밍노트북 7종 출시	16	0
NEWS02033	20200413	에이수스, 인텔 10세대 CPU 장착한 게이밍노트북 7종 출시	17	0
NEWS02033	20200413	에이수스, 인텔 10세대 CPU 장착한 게이밍노트북 7종 출시	18	0
NEWS02033	20200413	에이수스, 인텔 10세대 CPU 장착한 게이밍노트북 7종 출시	19	0
NEWS02033	20200413	에이수스, 인텔 10세대 CPU 장착한 게이밍노트북 7종 출시	20	0
NEWS02033	20200413	에이수스, 인텔 10세대 CPU 장착한 게이밍노트북 7종 출시	21	1
NEWS02033	20200413	에이수스, 인텔 10세대 CPU 장착한 게이밍노트북 7종 출시	22	1
NEWS02033	20200413	에이수스, 인텔 10세대 CPU 장착한 게이밍노트북 7종 출시	23	1
NEWS02033	20200413	에이수스, 인텔 10세대 CPU 장착한 게이밍노트북 7종 출시	24	1
NEWS02033	20200413	에이수스, 인텔 10세대 CPU 장착한 게이밍노트북 7종 출시	25	1

n_id	date	title	ord	info
NEWS05454	20200409	[스탁리포트] 9일, 코스피 개인 순매수 기관·외국인 매도, 코	29	0
NEWS05454	20200409	[스탁리포트] 9일, 코스피 개인 순매수 기관·외국인 매도, 코	30	0
NEWS05454	20200409	[스탁리포트] 9일, 코스피 개인 순매수 기관·외국인 매도, 코	31	0
NEWS05454	20200409	[스탁리포트] 9일, 코스피 개인 순매수 기관·외국인 매도, 코	32	0
NEWS05454	20200409	[스탁리포트] 9일, 코스피 개인 순매수 기관·외국인 매도, 코	33	0
NEWS05454	20200409	[스탁리포트] 9일, 코스피 개인 순매수 기관·외국인 매도, 코	34	0
NEWS05454	20200409	[스탁리포트] 9일, 코스피 개인 순매수 기관·외국인 매도, 코	35	1
NEWS05454	20200409	[스탁리포트] 9일, 코스피 개인 순매수 기관·외국인 매도, 코	36	1
NEWS05454	20200409	[스탁리포트] 9일, 코스피 개인 순매수 기관·외국인 매도, 코	37	1
NEWS05454	20200409	[스탁리포트] 9일, 코스피 개인 순매수 기관·외국인 매도, 코	38	1
NEWS05454	20200409	[스탁리포트] 9일, 코스피 개인 순매수 기관·외국인 매도, 코	39	1

Content에서 불용어만 제거함

[아시아경제 임혜선 기자] 롯데하이마트가 '대한민국 동행세일'에 동참한다

아시아경제 뉴욕=백종민 특파원 미국 국방부 고위 당국자가 18일(현지시간) 북한이 동북아지역에서 지속적으로 특별한 위협이 되고 있다고 우려했다

알테오젠 이외에도 신성이엔지(011930), 두산우(0000155), 파트론(091700) 등에 대해서도 투자자들의 게시물이 급증하고 있다

종합 경제정보 미디어 이데일리 - 무단전재 & 재배포 금지

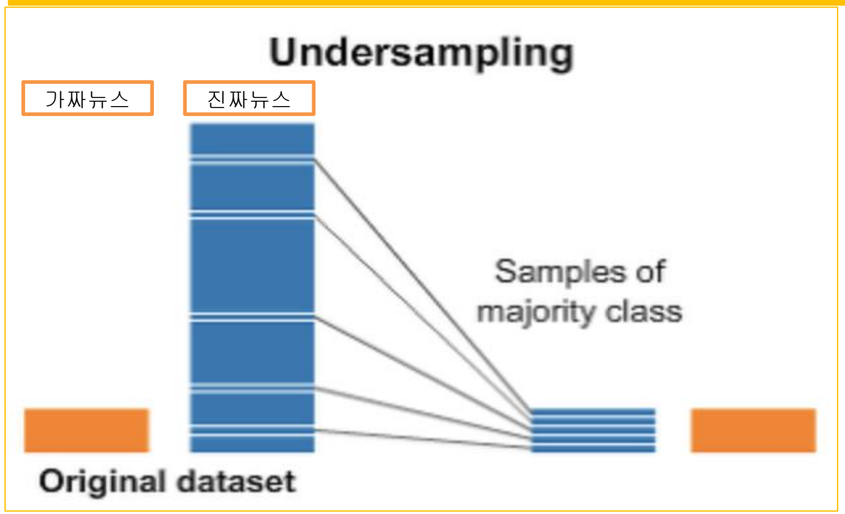
상신이디피(091580), 이엠코리아(095190), 우수AMS(066590)

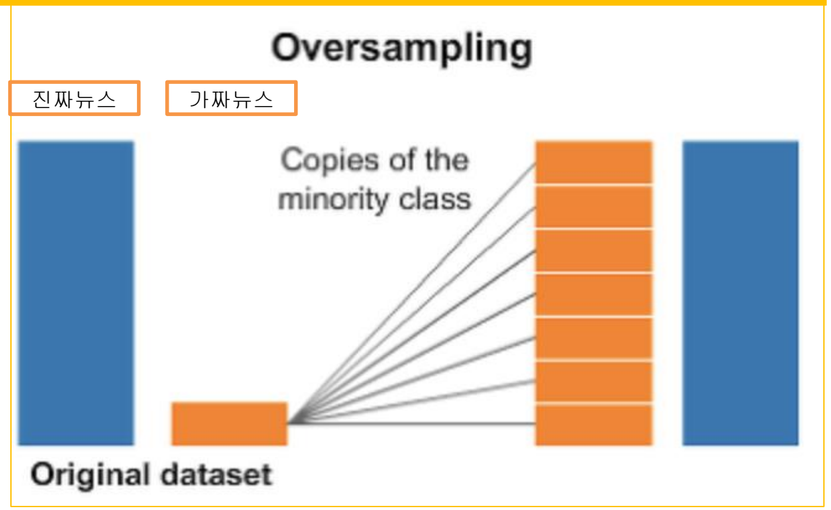
- 불용어 : 재배포 금지, 무단배포, 무단전재
- [],() 괄호 안에 내용

02 아이디어

데이터 전처리 - Oversampling / Undersampling

Under sampling, Over sampling 비교(epoch 10 기준)





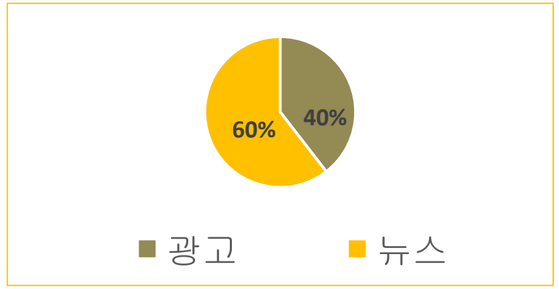
Average test loss: 0.59591

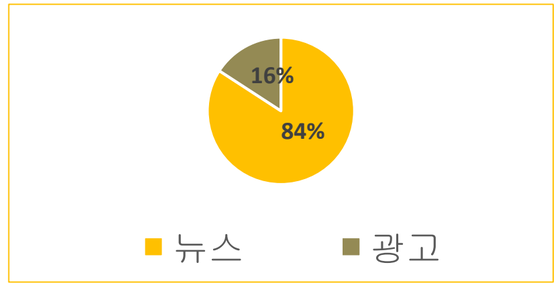
	precision	recall	f1-score	support
0	1.0000	0.8783	0.9352	575
1	0.8427	1.0000	0.9146	375
accuracy			0.9263	950
macro avg	0.9213	0.9391	0.9249	950
weighted avg	0.9379	0.9263	0.9271	950

Average test loss: 0.11590

	precision	recall	f1-score	support
0	1.0000	0.9374	0.9677	575
1	0.9124	1.0000	0.9542	375
accuracy			0.9621	950
macro avg	0.9562	0.9687	0.9609	950
weighted avg	0.9654	0.9621	0.9624	950

Good!





02 아이디어

01. 팀원 소개

02 아이디어

03. 결과 해석

04. 향후 방향

05. 코드

데이터 전처리 - 중복 제거

중복 제거 전

	뉴스	광고	총합
기사 개수	71,814개	46,932개	118,745
비율	60	40	100

중복 제거 후

	뉴스	광고	총합
기사 개수	40,370개	7,591개	46,161
비율	87	13	100

n_id	date	title	content
NEWS04454	20200103	법무부 '머그샷 공개 제동'에...경찰 '주민등록사진' 공개 추진	경찰은 이 법에 근거, 지방경찰청 내 신상공개위원회를 두고 강력범들의 신상공개를 결정해왔다
NEWS04454	20200103	법무부 '머그샷 공개 제동'에...경찰 '주민등록사진' 공개 추진	경찰은 이 법에 근거, 지방경찰청 내 신상공개위원회를 두고 강력범들의 신상공개를 결정해왔다
NEWS04454	20200103	법무부 '머그샷 공개 제동'에...경찰 '주민등록사진' 공개 추진	경찰은 이 법에 근거, 지방경찰청 내 신상공개위원회를 두고 강력범들의 신상공개를 결정해왔다
NEWS04454	20200103	법무부 '머그샷 공개 제동'에...경찰 '주민등록사진' 공개 추진	경찰은 이 법에 근거, 지방경찰청 내 신상공개위원회를 두고 강력범들의 신상공개를 결정해왔다
NEWS07069	20200524	'홍콩 국가보안법' 반대 격렬 시위...시위대 200여명 체포	경찰은 이날 8000여명을 시내 곳곳에 배치, 불법 시위가 벌어지는 즉시 엄중하게 대응하겠다는 방침을 밝혔다
NEWS07707	20200518	성추행 오거돈 곧 소환...휴대폰·블랙박스 압수(종합)	경찰은 이번 압수수색에서 오 전 시장 휴대전화와 차량 블랙박스 영상 등을 압수한 것으로 알려졌다
NEWS03859	20200320	'지오영' 정부 지침 어기고 마스크 수십만장 불법 판매, 경찰 수사 착수	경찰은 자체 조사한 내용과 식약처로부터 들어온 고발 내용을 토대로 정식 수사에 착수할 방침이다
NEWS05297	20200227	'코로나19 업무' 전주시 공무원, 집에서 숨진 채 발견	경찰은 정확한 사망 원인을 파악하기 위해 국립과학수사연구원에 부검을 의뢰할 예정이라고 밝혔다
NEWS05297	20200227	'코로나19 업무' 전주시 공무원, 집에서 숨진 채 발견	경찰은 정확한 사망 원인을 파악하기 위해 국립과학수사연구원에 부검을 의뢰할 예정이라고 밝혔다
NEWS05297	20200227	'코로나19 업무' 전주시 공무원, 집에서 숨진 채 발견	경찰은 정확한 사망 원인을 파악하기 위해 국립과학수사연구원에 부검을 의뢰할 예정이라고 밝혔다
NEWS05297	20200227	'코로나19 업무' 전주시 공무원, 집에서 숨진 채 발견	경찰은 정확한 사망 원인을 파악하기 위해 국립과학수사연구원에 부검을 의뢰할 예정이라고 밝혔다
NEWS05261	20200430	'조혜연 9단 스토킹' 40대 남성, 구속송치	경찰은 조 9단 요청에 따라 그의 주거지와 학원 일대 순찰을 강화하고 신변보호 조치 중이다
NEWS08418	20200324	학보에 성폭력 예방 기사·범죄 중에도 봉사...'두 얼굴' 조주빈	경찰은 조 씨의 성착취물을 관전한 이용자의 신상을 파악하는 데 주력하고 있다
NEWS08418	20200324	학보에 성폭력 예방 기사·범죄 중에도 봉사...'두 얼굴' 조주빈	경찰은 조 씨의 성착취물을 관전한 이용자의 신상을 파악하는 데 주력하고 있다
NEWS08418	20200324	학보에 성폭력 예방 기사·범죄 중에도 봉사...'두 얼굴' 조주빈	경찰은 조 씨의 성착취물을 관전한 이용자의 신상을 파악하는 데 주력하고 있다
NEWS08418	20200324	학보에 성폭력 예방 기사·범죄 중에도 봉사...'두 얼굴' 조주빈	경찰은 조 씨의 성착취물을 관전한 이용자의 신상을 파악하는 데 주력하고 있다
NEWS06763	20200424	휴대폰 둔 채 무단이탈한 30대 자가격리자, 구속영장 기각	경찰은 지난 22일 자가격리 위반에 무관용으로 엄정 대응하겠다는 정부 방침에 따라 A씨에게 구속영장을 신청했고 검찰도 이를 받아들여 법원에 구속영장을 청구했다
NEWS07030	20200325	경찰, 성폭행 혐의 김건모 기소의견 송치	경찰은 지난해 12월부터 김씨에 대한 수사에 착수했다
NEWS03686	20200330	경찰, '프로포폴 투약 의혹' 이부진 지난 22일 소환 조사	경찰은 지난해 3월부터 시작된 수사가 1년을 넘겨 장기화된 만큼 4월 중 수사를 마무리한다는 방침이다
NEWS00026	20200226	'오피스텔 성매매' 적발된 현직 검사, 벌금형 약식기소	경찰은 채팅앱 등에 게시된 성매매 광고 글을 추적하다가 성매매 현장을 잡은 것으로 알려졌다
NEWS00340	20200502	올림픽 은메달리스트, 미성년자 성폭행 혐의로 구속	경찰은 추가로 수사한 뒤 다음주 중 사건을 검찰에 송치할 방침이다
NEWS01106	20200117	[단독] 관사도, 식당도, 인력도 없다...혈세 136억 들인 새 경찰견훈련센터 '무용지물'	경찰은 특수 목적견을 본격적으로 번식훈련해 전국 경찰에 지원하고, 유능한 경찰견 핸들러를 양성하겠다며 2009년 10월 관련 사업에 착수했지만, 주민 항의 등 온갖 잡음 끝에 2018년 4월에야 새 센터를 착공할 수 있었다
NEWS01106	20200117	[단독] 관사도, 식당도, 인력도 없다...혈세 136억 들인 새 경찰견훈련센터 '무용지물'	경찰은 특수 목적견을 본격적으로 번식훈련해 전국 경찰에 지원하고, 유능한 경찰견 핸들러를 양성하겠다며 2009년 10월 관련 사업에 착수했지만, 주민 항의 등 온갖 잡음 끝에 2018년 4월에야 새 센터를 착공할 수 있었다
NEWS01106	20200117	[단독] 관사도, 식당도, 인력도 없다...혈세 136억 들인 새 경찰견훈련센터 '무용지물'	경찰은 특수 목적견을 본격적으로 번식훈련해 전국 경찰에 지원하고, 유능한 경찰견 핸들러를 양성하겠다며 2009년 10월 관련 사업에 착수했지만, 주민 항의 등 온갖 잡음 끝에 2018년 4월에야 새 센터를 착공할 수 있었다
NEWS01106	20200117	[단독] 관사도, 식당도, 인력도 없다...혈세 136억 들인 새 경찰견훈련센터 '무용지물'	경찰은 특수 목적견을 본격적으로 번식훈련해 전국 경찰에 지원하고, 유능한 경찰견 핸들러를 양성하겠다며 2009년 10월 관련 사업에 착수했지만, 주민 항의 등 온갖 잡음 끝에 2018년 4월에야 새 센터를 착공할 수 있었다

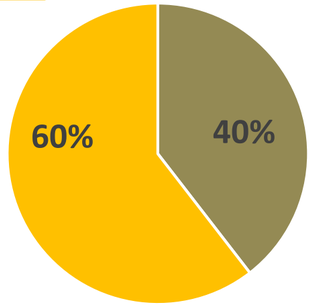
중복 Content!

02 아이디어

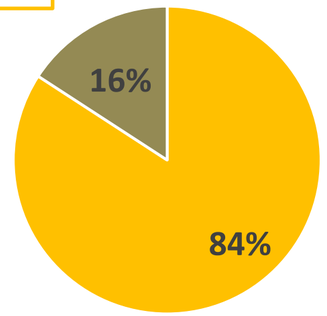
데이터 전처리 - 중복 제거

중복 content를 제거한 데이터와 제거하지 않은것 비교(Oversampling, epoch 10 기준)

중복 제거 전



중복 제거 후



Average test loss: 0.11590

	precision	recall	f1-score	support
0	1.0000	0.9374	0.9677	575
1	0.9124	1.0000	0.9542	375
accuracy			0.9621	950
macro avg	0.9562	0.9687	0.9609	950
weighted avg	0.9654	0.9621	0.9624	950

Good!

Average test loss: 1.17042

	precision	recall	f1-score	support
0	1.0000	0.6633	0.7976	398
1	0.3163	1.0000	0.4806	62
accuracy			0.7087	460
macro avg	0.6582	0.8317	0.6391	460
weighted avg	0.9079	0.7087	0.7549	460

02 아이디어

데이터 전처리 - Summary

데이터 전처리

1. Content length

   - 모든 Content 내용은 길이 64로 잘라서 사용함. KoBERT는 최대 512까지의 문장 길이만을 처리하는데, 데이터를 보니 95% 수준이 56이여서 모델의 연산 속도 향상을 위해 64수준을 기준으로 문장을 잘랐음.
   - Train data 기준 가짜 뉴스의 최대 길이가 512이기 때문에, 512를 기준으로 하는 범주형 변수 longer와, shorter라는 feature를 추가함. (One hot encoding으로 구현)

2. Oversampling

   - Train data의 진짜뉴스, 가짜뉴스의비율이 6:4로 imbalance하기 때문에 학습을 시킬수록 정확도를 제외한 다른 지표들이 높지 않을 수 있음. 따라서 데이터를 5:5로 맞춰주기 위해 Oversampling 또는 Undersampling을 시켜야 함. 이 중 epoch 10 기준 accuracy와 cross entrop가 좋은 Oversampling 작업을 해줌.

3. 중복데이터

   - 기존 train data의 진짜 뉴스, 가짜 뉴스 비율을 6:4이고 Content 기준 중복 제거 한 후 진짜 뉴스, 가짜 뉴스의 비율을 87:13임. 중복 제거 전과 후 Oversampling으로 데이터 비율을 5:5 로 맞춰주고 epoch 10기준 accuracy, cross entropy가 더 좋은 중복 제거 전 데이터를 Oversampling하여 사용함.

4. Stopwords

   - 기사 특성상 내용과 큰 관계가 없는 단어를 제거함.

02 아이디어

Model - BERT

- 단어 표현에 있어서는 분류에서 성능이 더 뛰어난 BERT 모델을 활용하였음.(한국어, NSMC dataset기준)
- BERT는 구글에서 개발한 NLP(자연어처리) 사전 훈련 기술이며, 특정 분야에 국한된 기술이 아니라 **모든 자연어 처리 분야에서 좋은 성능을 내는 범용** Language Model

Naver Sentiment Analysis

- Dataset : https://github.com/e9t/nsmc

Model	Accuracy
BERT base multilingual cased	0.875
KoBERT	0.901
KoGPT2	0.899

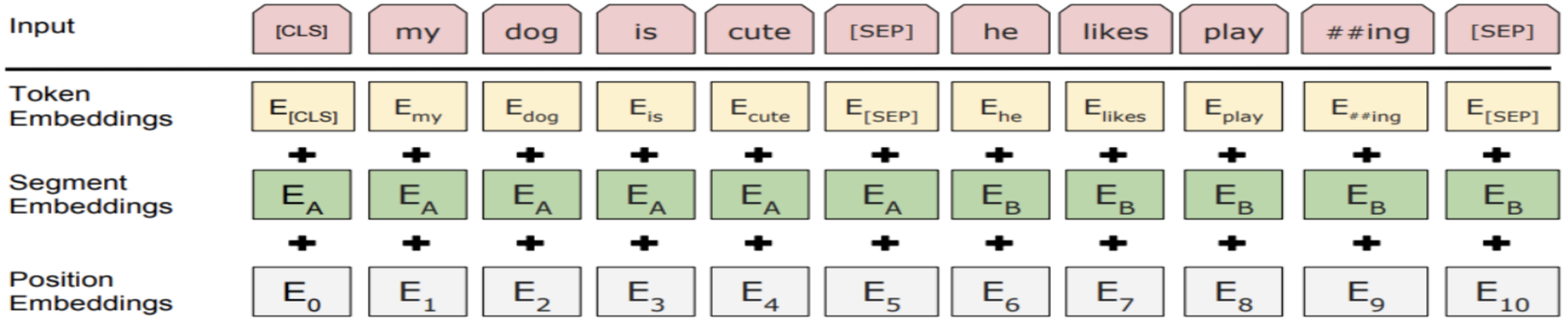
- 한국어 Machine Reading Comprehesion 데이터셋 KorQuard로 QA대회 'KorQuard'의 리더보드의 상위 성적을 기록한 대부분의 참가자들은 이 언어모델 BERT를 사용하였음

Rank	Reg. Date	Model	EM	F1
-	2018.10.17	**Human Performance**	80.17	91.20
1	2020.01.08	SkERT-Large (single model) Skelter Labs	87.66	**95.15**
2	2019.10.25	KorBERT-Large v1.0 ETRI ExoBrain Team	**87.76**	95.02
3	2020.01.07	SkERT-LARGE (single model) Skelter Labs	87.25	94.75
4	2019.06.26	LaRva-Kor-Large+ + CLaF (single) Clova AI LaRva Team	86.84	94.75
5	2020.01.03	SkERT Large (single model) Skelter Labs	87.28	94.66

02 아이디어

Model - BERT

1. Input



- Token Embedding

- Word Piece 임베딩 방식 사용, 각 Char(문자) 단위로 임베딩을 하고, 자주 등장하면서 가장 긴 길이의 sub-word를 하나의 단위로 만듦. 자주 등장하지 않는 단어는 다시 sub-word로 만들어 이는 이전에 자주 등장하지 않았던 단어를 모조리 ' OOV ' 처리하여 모델링의 성능을 저하했던 ' OOV ' 문제도 해결 할 수 있음.

- Segment Embedding

- Sentence Embedding, 토큰 시킨 단어들을 다시 하나의 문장으로 만드는 작업임. BERT에서는 두개의 문장을 구분자([SEP])를 넣어 구분하고 그 두 문장을 하나의 Segment로 지정하여 입력함.
- BERT에서는 이 한 세그먼트를 512 sub-word 길이로 제한하는데, 한국어는 보통 60 sub-word가 넘지 않는다고 하니 BERT를 사용할 때, 하나의 세그먼트에 128로 제한하여도 충분히 학습이 가능함.

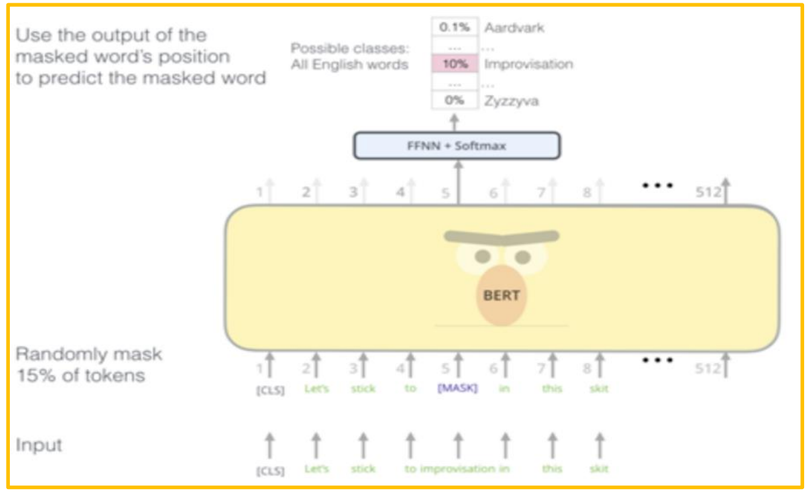
- Position Embedding

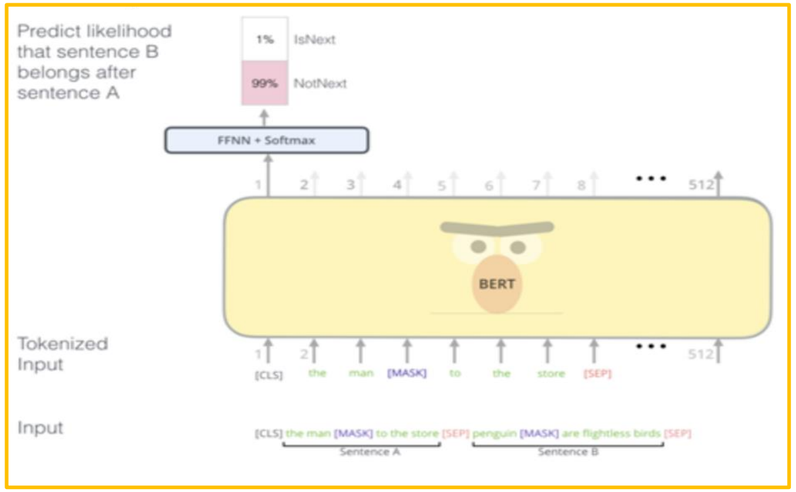
- BERT는 Transformer의 인코더, 디코더 중 인코더만 사용하고 Self Attention은 입력의 위치를 고려하지 않고 입력 토큰의 위치 정보를 고려함. 그래서 Transformer모델에서는 Sinusoid 함수를 이용하여 Positional encoding을 사용하고 BERT는 이를 따서 Position Encoding을 사용함. 간단하게 이해하면 Position encoding은 Token 순대로 인코딩 하는 것을 뜻함.

02 아이디어

Model - BERT

2. Pre-Training





•MLM(Masked Language Model)

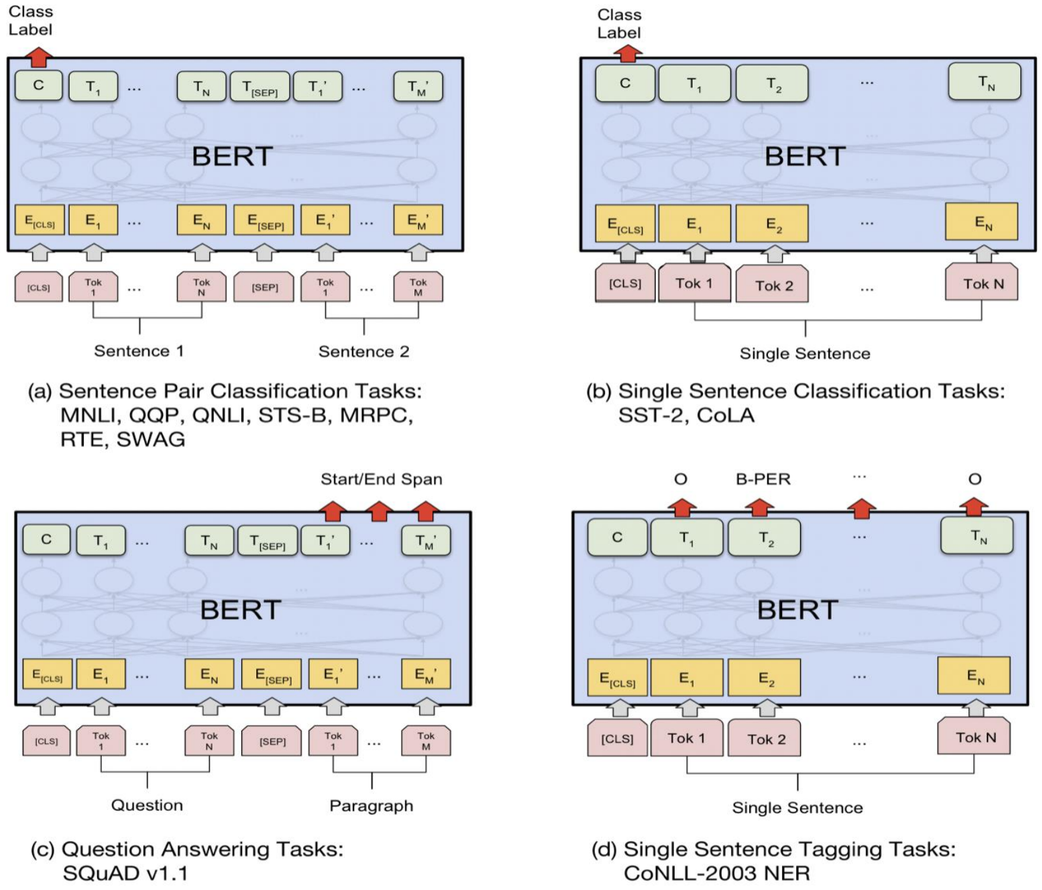
•NSP(Next Sentence Prediction)

- 데이터들을 임베딩하여 훈련시킬 데이터를 모두 인코딩 하였으면, 사전훈련을 시킬 단계임. 기존의 방법들은 보통 문장을 왼쪽에서 오른쪽으로 학습하여 다음 단어를 예측하는 방식이거나, 예측할 단어의 좌우 문맥을 고려하여 예측하는 방식을 사용하지만 BERT는 언어의 특성을 잘 학습하도록, MLM(Masked Language Model), NSP(Next Sentence Prediction) 두가지 방식을 사용함.
- MLM만 사용하거나 NSP만 사용한 경우보다 두가지를 모두 사용한 경우가 성능이 확실히 뛰어남.
- 우리는 SKT 에서 개발한 KoBERT를 활용함.

02 아이디어

Model - BERT

3. Transfer learning



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

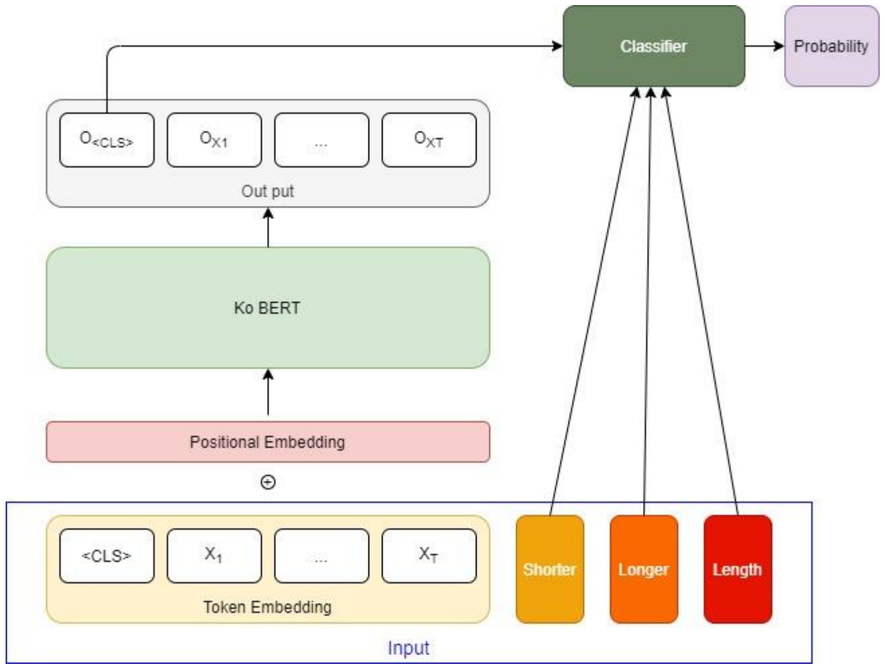
(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

- (a), (b)는 sequence-level task, (c)와 (d)는 token-level task임
- ©의 QA task경우는, **Question에 정답이 되는 Paragraph의 substring을 뽑아내는 것**이므로, [SEP] token 이후의 token들에서 **Start/End Span**을 찾아내는 task를 수행함.
- (d)의 경우는 **Named Entity Recognition(NER)**이나 **형태소 분석**과 같이 single sentence에서 각 토큰이 어떤 class를 갖는지 **모두 classifier** 적용하여 정답을 찾아냄.
- 향후에 모델의 효용성에 따라 **Distilling BERT(Tiny BERT)**로 축소시켜서 사용하기도 용이함.

02 아이디어

Model – Our Model



[feature 설명]

- Input : Tokenized Sentence, Shorter, Longer, Length
- Shorter의 경우 512보다 같거나 짧은 경우 1, 아닌 경우 0인 feature
- Longer의 경우 512보다 길면 1, 아니면 0인 feature
- Length의 경우 Tokenize된 문장의 길이임.

[모델 process 설명]

- Tokenized Sentence는 KoBERT 모델을 통과함. 나온 결과 값 중에서 <CLS>에 해당하는 부분과 Shorter, Longer, Length를 concat해서 classifier 층을 통과시켜 probability 를 계산.

02 아이디어

Model – Optimization - AdamW

Adam VS AdamW

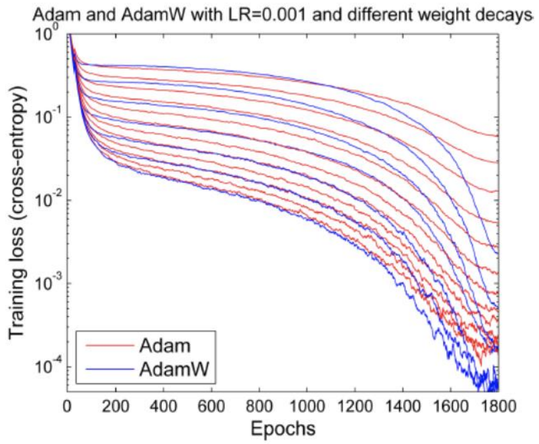
Algorithm 2 Adam with L_2 regularization and Adam with decoupled weight decay (AdamW)

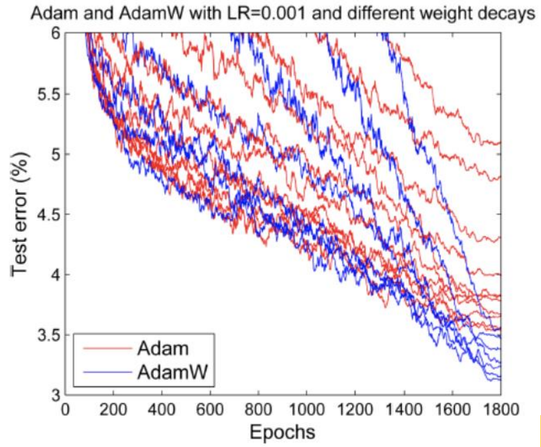
given $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}, \lambda \in \mathbb{R}$
initialize time step $t \leftarrow 0$, parameter vector $\boldsymbol{\theta}_{t=0} \in \mathbb{R}^n$, first moment vector $\boldsymbol{m}_{t=0} \leftarrow \boldsymbol{0}$, second moment vector $\boldsymbol{v}_{t=0} \leftarrow \boldsymbol{0}$, schedule multiplier $\eta_{t=0} \in \mathbb{R}$
repeat
  $t \leftarrow t + 1$
  $\nabla f_t(\boldsymbol{\theta}_{t-1}) \leftarrow \text{SelectBatch}(\boldsymbol{\theta}_{t-1})$ ▷ select batch and return the corresponding gradient
  $\boldsymbol{g}_t \leftarrow \nabla f_t(\boldsymbol{\theta}_{t-1}) + \lambda \boldsymbol{\theta}_{t-1}$
  $\boldsymbol{m}_t \leftarrow \beta_1 \boldsymbol{m}_{t-1} + (1 - \beta_1)\boldsymbol{g}_t$ ▷ here and below all operations are element-wise
  $\boldsymbol{v}_t \leftarrow \beta_2 \boldsymbol{v}_{t-1} + (1 - \beta_2)\boldsymbol{g}_t^2$
  $\hat{\boldsymbol{m}}_t \leftarrow \boldsymbol{m}_t / (1 - \beta_1^t)$ ▷ β_1 is taken to the power of t
  $\hat{\boldsymbol{v}}_t \leftarrow \boldsymbol{v}_t / (1 - \beta_2^t)$ ▷ β_2 is taken to the power of t
  $\eta_t \leftarrow \text{SetScheduleMultiplier}(t)$ ▷ can be fixed, decay, or also be used for warm restarts
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \eta_t \left(\alpha \hat{\boldsymbol{m}}_t / (\sqrt{\hat{\boldsymbol{v}}_t} + \epsilon) + \lambda \boldsymbol{\theta}_{t-1} \right)$
until *stopping criterion is met*
return optimized parameters $\boldsymbol{\theta}_t$

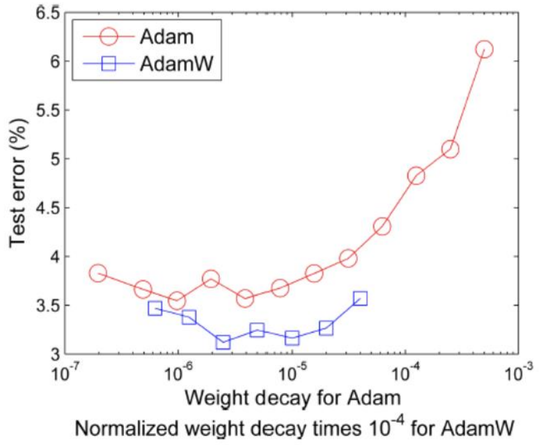
02 아이디어

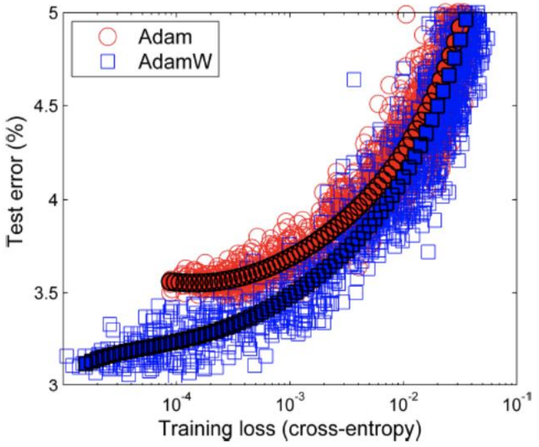
Model – Optimization - AdamW

Adam VS AdamW









- 학습 초기에는 Adam과 AdamW과 비슷한 loss를 보이지만 학습이 진행될 수록 **AdamW의 훈련 손실과 test 에러가 더 낮아짐**.
- 오른쪽 아래 그래프에서 같은 훈련 손실에 대해서 AdamW의 테스트 에러가 더 낮은 것을 볼 수 있음.
- AdamW의 좋은 성능이 단순히 학습 동안 더 좋은 수렴 지점을 찾았기 때문이 아니라, **더 좋은 일반화 능력이 있기 때문**.

02 아이디어

Model - Summary

BERT

- Word representation 방식으로는 성능이 우수한 BERT를 선택하여 사용함.
- KoBERT의 경우 최대 문장의 길이가 512까지만 처리가 가능한데, 좀 더 효율적인 연산을 위해서 데이터의 길이를 전체 데이터의 95% 수준과 비슷한 64로 설정함. 그리고 길이에 대한 정보를 반영하고자 feature에 length를 추가. 또한 마지막으로, train data의 경우 길이가 512 이상인 경우 진짜 뉴스임을 반영해서 512를 기준으로 하는 feature, shorter와 longer를 추가함.

adamW

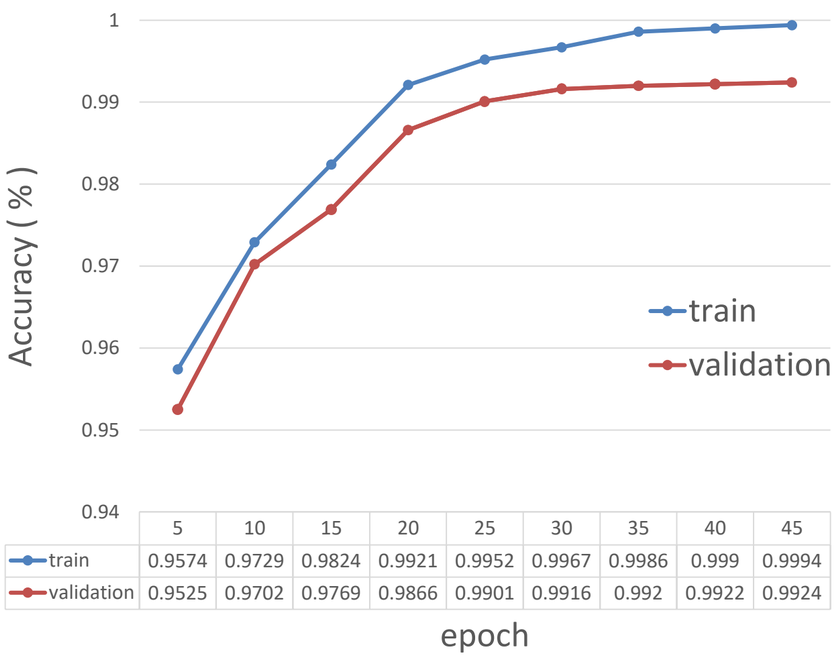
- 모델의 일반화 능력을 증가 시키고자, Adam보다 일반화 능력이 뛰어난 AdamW를 사용함.
- 본 실험에서는 0.1로 설정함.

03

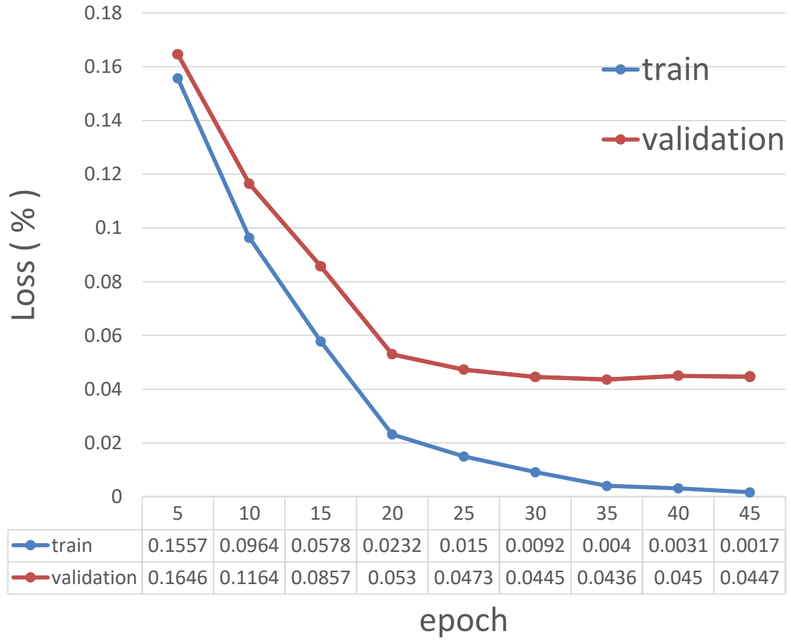
결과 해석

03 결과 해석

Model - 결과



	5	10	15	20	25	30	35	40	45
train	0.9574	0.9729	0.9824	0.9921	0.9952	0.9967	0.9986	0.999	0.9994
validation	0.9525	0.9702	0.9769	0.9866	0.9901	0.9916	0.992	0.9922	0.9924



	5	10	15	20	25	30	35	40	45
train	0.1557	0.0964	0.0578	0.0232	0.015	0.0092	0.004	0.0031	0.0017
validation	0.1646	0.1164	0.0857	0.053	0.0473	0.0445	0.0436	0.045	0.0447

학습

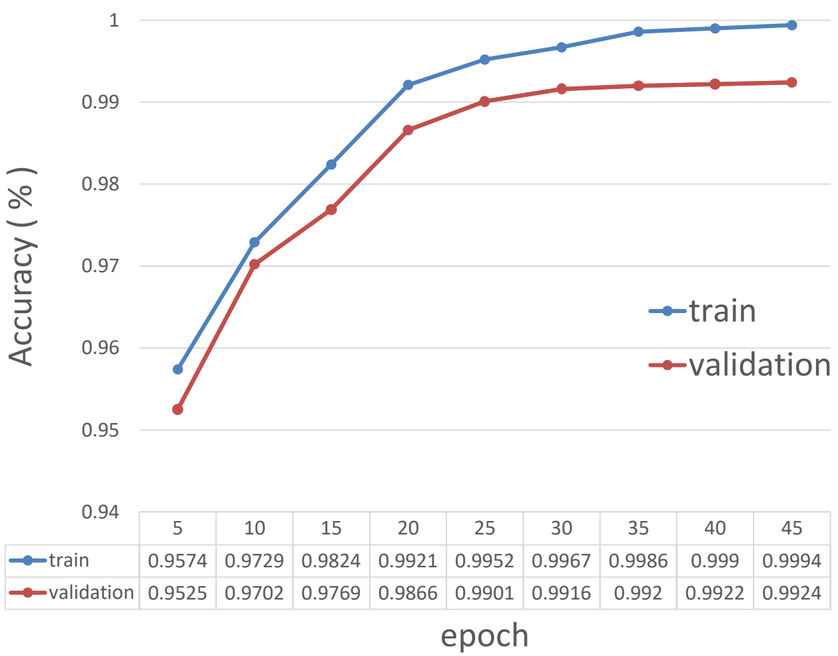
- Google colab pro를 활용하였음. (GPU : Tesla p100(16Giga), RAM : 25Giga)
- 학습 시간은 epoch당 12분 정도 소요됨.

결과 해석

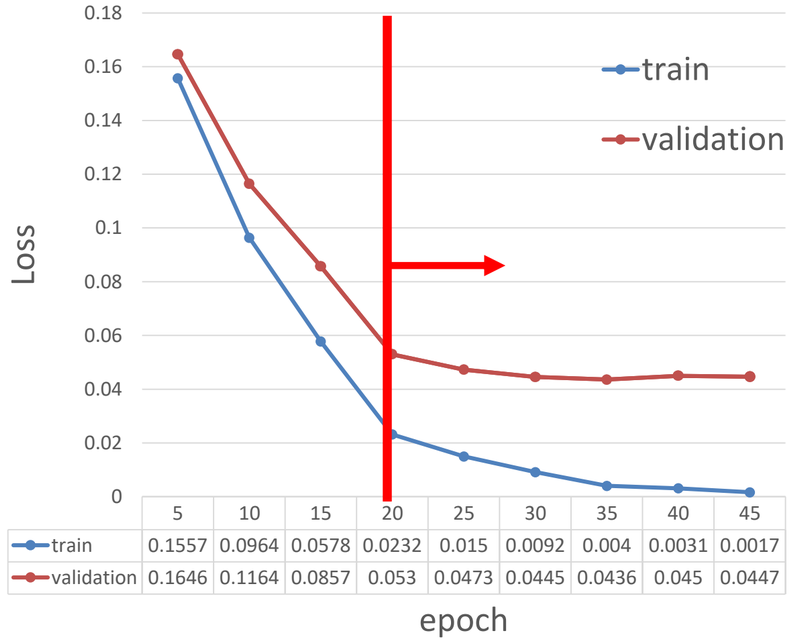
- Epoch 35인 경우 가장 낮은 validation set에서의 cross entropy를 얻음. 이와 두번째로 작은 loss를 갖는 epoch 45와 비교해서 test set에서의 loss 값과 acc를 보았을 때, Epoch 35인 경우 loss : 0.02731, acc : 0.9937., epoch 45인 경우 loss : 0.02461, acc : 0.9958로 나오게 되어서. Epoch 45일 때 모델을 선택함.

03 결과 해석

Model - 결과



	5	10	15	20	25	30	35	40	45
train	0.9574	0.9729	0.9824	0.9921	0.9952	0.9967	0.9986	0.999	0.9994
validation	0.9525	0.9702	0.9769	0.9866	0.9901	0.9916	0.992	0.9922	0.9924



	5	10	15	20	25	30	35	40	45
train	0.1557	0.0964	0.0578	0.0232	0.015	0.0092	0.004	0.0031	0.0017
validation	0.1646	0.1164	0.0857	0.053	0.0473	0.0445	0.0436	0.045	0.0447

결과 해석

- Epoch 20을 기점으로 점차 validation과 train의 loss 차이가 증가함. Train set에 대한 over fitting 경향이 강해지는 것을 알 수 있음.
- 데이터에 중복된 부분이 많아서 이러한 현상이 발생하는 것으로 해석됨
- 이를 해결하기 위해선 cross validation을 실시, Adam에서 weight decay를 증가하거나, 중복을 제거 후 anomaly detection 방식으로 변경하는 것이 합리적으로 생각됨.

03 결과 해석

Model - 결과

Average train loss: 0.00166

	precision	recall	f1-score	support
0	0.9998	0.9989	0.9994	68238
1	0.9989	0.9998	0.9994	68238
accuracy			0.9994	136476
macro avg	0.9994	0.9994	0.9994	136476
weighted avg	0.9994	0.9994	0.9994	136476

Average val loss: 0.04466

	precision	recall	f1-score	support
0	0.9961	0.9888	0.9924	2315
1	0.9889	0.9961	0.9925	2315
accuracy			0.9924	4630
macro avg	0.9925	0.9924	0.9924	4630
weighted avg	0.9925	0.9924	0.9924	4630

Average test loss: 0.02461

	precision	recall	f1-score	support
0	1.0000	0.9930	0.9965	575
1	0.9894	1.0000	0.9947	375
accuracy			0.9958	950
macro avg	0.9947	0.9965	0.9956	950
weighted avg	0.9958	0.9958	0.9958	950

- train set과 validation set은 동일하게 over sampling을 해서 비교적 좋은 loss 값과 accuracy, recall, f1 score 값을 낳았음. 하지만 test set은 imbalance하므로 정확성은 0.9958로 높지만, precision, recall 값이 좋지 않았음(특히 1에 대한 precision) 실제로도 데이콘에서 제공한 test set에서의 public acc 값은 0.987수준으로 train set에서의 acc와 차이가 많았음.
  -> 즉, overfitting 됐다는 것을 알 수 있음
- 이를 보완하기 위해, cross validation을 실시해서 보다 다양한 데이터셋에서 높은 acc과 낮은 loss 값을 갖는 모델을 도출하는 방법이 있음.(hardware에 대한 capacity가 필요)
- 또한 weight decay 값을 증가하고, drop out 비율을 증가시킨 모델과의 비교를 통해 더 좋은 일반화 능력을 갖는 모델을 도출할 수 있음. (hardware에 대한 capacity가 필요)

03 결과 해석

Model - 결과

```
In [32]: data.loc[data.result==0,['content','info','predict','length','longer','shorter']].tail(30)
```

Out[32]:

	content	info	predict	length	longer	shorter
39384	데이터에서 찾은 AI 속보	0	1	9	0	1
39421	데이터에서 찾은 AI 속보	0	1	9	0	1
39473	데이터에서 찾은 AI 속보	0	1	9	0	1
39675	데이터에서 찾은 AI 속보	0	1	9	0	1
40089	데이터에서 찾은 AI 속보	0	1	9	0	1
40694	데이터에서 찾은 AI 속보	0	1	9	0	1
40755	데이터에서 찾은 AI 속보	0	1	9	0	1
41157	데이터에서 찾은 AI 속보	0	1	9	0	1
41221	데이터에서 찾은 AI 속보	0	1	9	0	1
41565	데이터에서 찾은 AI 속보	0	1	9	0	1
41945	데이터에서 찾은 AI 속보	0	1	9	0	1
42226	데이터에서 찾은 AI 속보	0	1	9	0	1
42266	데이터에서 찾은 AI 속보	0	1	9	0	1
42666	1일에서 11일사이 각 지역별 주요 부동산 실거래가는 다음과 같다.서울지역 아파트 ...	0	1	976	1	0
42912	EPL, 코로나19 검사에서 748명 중 6명 양성 반응	1	0	20	0	1
43072	"설득의 리더십으로 일하는 국회 만들 것"	1	0	14	0	1
44174	한국TV 주식카톡방의 수익률을 믿기 어렵다면 무료로 이용해보도록 하자	0	1	21	0	1
45250	미국 FDA가 코오롱티슈진이 이전까지 제출한 임상시험 데이터의 유효성을 인정, 이를...	1	0	66	0	1
48108	쭉스바겐, 20일부터 단계적으로 생산 재개	1	0	13	0	1
48133	무료종목상담 일정: 4월 20일~ 4월 24일, 08:00~18:00	1	0	23	0	1
48169	우리나라 먼저 발 빠른 대응 "이종목" 갑니다.	0	1	15	0	1
48202	삼성중공업 빅텍 일양약품 조아제약 씨젠	0	1	14	0	1

```
data.loc[data.length>64,['result']]
```

	result
19	True
40	True
41	True
47	True
54	True
...	...
118602	True
118649	True
118650	True
118651	True
118652	True

4257 rows × 1 columns

```
(data.loc[data.length>=64,['result']]==False).sum()
```

```
result    7
dtype: int64
```

```
(data.loc[data.length>64,['result']]==False).sum()/len(data.loc[data.length>=64,['result']])
```

```
result    0.001542
dtype: float64
```

결과 해석

- 주어진 train data에서 잘 못 분류된 경우를 보게 되면, content의 이름이 같은 경우인 경우가 많았음.
- 이를 해결하기 위해선 중복 제거를 해야된다고 생각됨
- 이런 방식으로 접근하게 되면 anomaly detection 방식으로 접근하면 됨.

- Max length가 64를 넘는 경우가 4,257 개 중, 잘 못 분류한 경우는 0.1542%(7건)으로 매우 낮았음. 이를 볼 때 길이를 64수준으로 자른 것은 적절한 결정이라고 생각됨.

03 결과 해석

Model - 결과

```
[15]  now = time.time()

      with torch.no_grad():
          model.eval()
          Predicted=[]
          for batch in test_dataloader:
              batch = tuple(t.to(device) for t in batch)
              input_ids, attention_mask, length, longer, shorter = batch
              outputs = model.forward(input_ids, attention_mask, length, longer, shorter)
              predicted = outputs.argmax(-1).tolist()
              Predicted.extend(predicted)
      print(time.time()-now)

      145.84559082984924
```

- Test set에 대한 시뮬레이션 결과
  전처리까진 45초 소요
  모델 불러오는 데 10초 소요
  inference에 **145초** 소요
  (Batch size ; 32, 64, 128, 245, 512, 1024 중 128이 가장 좋은 성능을 발휘함)

- 속도를 빠르게 하기 위해선, **모델 경량화**가 필요할 것으로 생각됨

03 결과 해석

Model - parameters

- 1. Model parameters (Total parameters : **92,188,424**)
  batch size : 256
  1.1. Kobert
  - num layers : 12
  - embedding dim : 768
  - hidden_size : 3072
  - max_length : 512
  - num_heads : 12
  - dropout : 0.1
  - embed drop out : 0.1
  - token type vocab size : 2(<CLS>,<SEP>)
  - n_vocab : 8002

  1.2. Linear classifier
  - input_dim : 771
  - output_dim : 2

```
predefined_args = {
        'attention_cell': 'multi_head',
        'num_layers': 12,
        'units': 768,
        'hidden_size': 3072,
        'max_length': 512,
        'num_heads': 12,
        'scaled': True,
        'dropout': 0.1,
        'use_residual': True,
        'embed_size': 768,
        'embed_dropout': 0.1,
        'token_type_vocab_size': 2,
        'word_embed': None,
    }
```

- 2. Optimization
  2.1. learning rate : linear decreasing
  1e-6에서 시작해서 epoch 100인 경우 0이 되게끔
  2.2 AdamW
  weight decay : 0.1

- 3. Early stop
  epoch5마다 validation set에서의 cross entropy loss 를 계산
  만약 10 epoch 동안 해당 값이 최소값이면 학습 중단
  그리고 가장 낮은 validation set에서의 cross entropy를 가지는 2개의 모델 중 가장 낮은 test set에서의 cross entropy를 가지는 모델을 최종 모델로 선택

03 결과 해석

Model - 결과 Summary

정확성

- Model의 경우, train, validation data에서는 우수한 성능을 발휘함
- 하지만 test set에서는 정확성 외 다른 성능이 다소 떨어짐
- 이는 train, validation의 경우 oversampling을 했으나 test set의 경우 imbalanced data여서 발생하는 문제임
- 이를 해결하기 위해선, cross validation을 실시하거나, 중복을 제거한 data에서 anomaly detection 방식으로 접근해야 함

속도

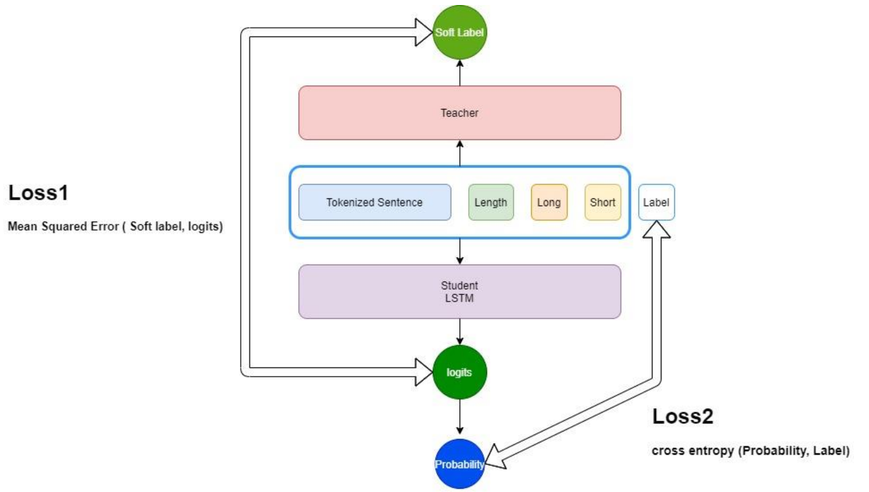
- Inference를 할 때 가장 많은 시간이 소요됨, 이는 모델이 매우 크기 때문임(모델 para : **92,188,424개,** size **: 350Mb**)
- 이를 해결하기 위해, 모델 경량화가 필요함.

04

향후 방향

04 향후 방향

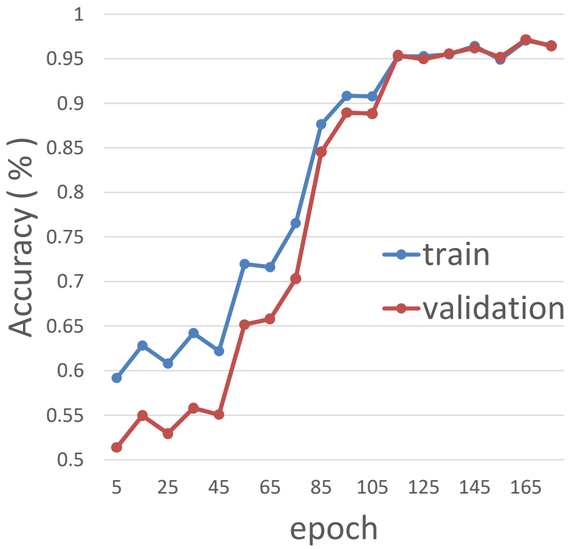
Model - 모델 경량화

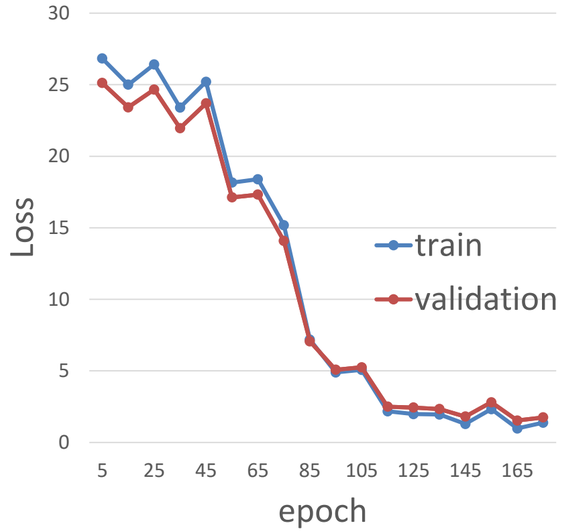


- Teacher : 기존 KoBERT Base Model
- Student : LSTM + Linear
- Teacher가 예측한 Logit(Soft Label)과 Student가 예측한 Logit과의 Mean Squared Error (Loss 1)
- Student가 예측한 Probability와 Label과의 Cross Entropy Loss (Loss 2)
- 이 방식을 활용하면 Teacher(Large Model)이 학습한 능력을 Student(Smaller Model)이 배울 수 있음(Distill)
- 통상적으로 a*Loss 1 + (1-a)*Loss 2를 쓰나, 본 실험에서는 a=1로 설정함.

04 향후 방향

Model - 모델 경량화





160

Average train loss: 0.07048

	precision	recall	f1-score	support
0	0.9992	0.9984	0.9988	68238
1	0.9984	0.9992	0.9988	68238
accuracy			0.9988	136476
macro avg	0.9988	0.9988	0.9988	136476
weighted avg	0.9988	0.9988	0.9988	136476

Average val loss: 0.53365

	precision	recall	f1-score	support
0	0.9862	0.9883	0.9873	2315
1	0.9883	0.9862	0.9872	2315
accuracy			0.9873	4630
macro avg	0.9873	0.9873	0.9873	4630
weighted avg	0.9873	0.9873	0.9873	4630

Average test loss: 0.45929

	precision	recall	f1-score	support
0	0.9913	0.9896	0.9904	575
1	0.9840	0.9867	0.9854	375
accuracy			0.9884	950
macro avg	0.9877	0.9881	0.9879	950
weighted avg	0.9884	0.9884	0.9884	950

- 학습 결과, validation set에서 loss가 가장 낮은 경우(early stopping)는 epoch이 160일 때 였고, 그 때의 train, validation, test set에서의 loss와 accuracy는 다음과 같음
- Train set에서는 acc가 99% 수준이였으나, validation, test set에서는 98% 수준으로 떨어짐
- 이에 대해선, a(가중치) 값을 조절함으로서 더 높은 수준을 달성할 것으로 예상됨

04 향후 방향

Model - 모델 경량화

```
[26]    now = time.time()

[27]    with torch.no_grad():
            model.eval()
            Predicted=[]
            for batch in test_dataloader:
                batch = tuple(t.to(device) for t in batch)
                input_ids, attention_mask, length, longer, shorter = batch
                outputs = model.forward(input_ids, attention_mask, length, longer, shorter)
                predicted = outputs.argmax(-1).tolist()
                Predicted.extend(predicted)
        print(time.time()-now)

        6.597807884216309
```

- Test set에 대한 시뮬레이션 결과
  전처리까진 45초 소요
  모델 불러오는 데 10초 소요
  inference에 **6.5초** 소요
- 기존 Teacher model : 145.8초에 비해서 **32.4배 빠른** 수준임
- 모델의 parameter 개수 역시, **1,288,968**개로, Teacher Model (92,188,424개)의 **1.3%** 수준임.

04 향후 방향

Model – 모델 경량화 Summary

모델 경량화

- Distill 기법을 활용해, Inference 속도는 Teacher Model에 비해서 32배 빠르고, Model para 개수는 1.3 % 수준을 달성함
- 하지만, 정확도가 Teacher의 경우 99% 수준임에 반해 Student는 98% 수준으로 하락함.
- 이에 대해선 a(가중치)를 변경해서 실험하던가, Teacher Model의 정확성을 올려야 될 것으로 생각됨.

정확성

- Model의 정확성을 높이는 실험에 대해선 진행하지 않았음.
- 중복 제거하고 Anomaly Detection을 하는 것이 좋을 것으로 생각됨

05

코드

05 코드

Code

Github : https://github.com/Chuck2Win/NH_project

- Preprocess : https://github.com/Chuck2Win/NH_project/blob/main/preprocess.py
  데이터 전처리(불용어 제거 등), Over sampling, Train Test Split, BERT에 필요한 Tokenizing, Mask 생성, Length, Longer, Short 생성
- Train : https://github.com/Chuck2Win/NH_project/blob/main/train.py
  전처리된 데이터를 학습시킴, 여러가지 parameter를 설정할 수 있음
- Inference : https://github.com/Chuck2Win/NH_project/blob/main/Inference.py
  학습된 모델을 가지고, 데이터를 전처리 시킨 후 예측을 함
- TEST 예시 : https://github.com/Chuck2Win/NH_project/blob/main/%5BTEST%5D.ipynb
  학습된 모델을 불러오고, 데이터를 전처리 시킨 후 예측을 하는 과정을 보여줌
- Distilling 학습 예시 :
  https://github.com/Chuck2Win/NH_project/blob/main/%5B%EB%8D%B0%EC%9D%B4%EC%BD%98%5D%5BBERT%5D%5B%EC%A4%91%EB%B3%B5%EC%A0%9C%EA%B1%B0%EC%97%86%EC%9D%B4%5D%5Boversampling%5D%5Bdistill%5D.ipynb

  Teacher 모델의 Logit값이 저장된 데이터를 불러와서, Student 를 학습시키는 과정을 보여줌

Reference

- Bert: Pre-training of deep bidirectional transformers for language understanding. Devlin et al, 2018
- Attention Is All You Need. Vaswani, A. et al, 2017.
- Distilling Task-Specific Knowledge from BERT into Simple Neural Networks, Raphael Tang et al, 2019
- Distilling the Knowledge in a Neural Network, Geoffrey Hinton et al, 2015

감사합니다

옥창원, 윤영주