Wrap-up meeting

2022 통계데이터 인공지능 활용대회
: 자연어 기반 인공지능 산업분류 자동화

2022년 4월 26일

목차

대회 개요

통계청이 주관하는 <통계 데이터 인공지능 활용 대회>로, 산업 설명 텍스트로 “산업 분류 자동화 모델”을 개발하는 대회입니다.

- 대회 일정 : 2022년 3월 21일 (월) ~ 4월 13일 (수) **약 3주간**
- 심사 방법 :
  - 1차 평가 : Accuracy, F1-score (Accuracy 동점일 때 F1-score로 평가) / 부분 점수 : 대분류 예측(0.1점), 중분류 예측(0.2점), 소분류 예측 (0.7점)
    - 평가 지표

Accuracy	F1- Score
정확히 분류된 표본 / 전체 표본	(2 * Precision * Recall) / (Precision + Recall) 각 카테고리 별 F1-Score를 산술평균 계산(Macro average) 최종 F1-Score 산출

  - 2차 평가 : 코드의 적정성 전문가 평가 (코드 작성 설명자료, F1-Score 점수 종합평가)
- 데이터 분류 체계
  - “10차 개정 한국 표준 산업 분류” 를 따름
  - 대분류 21종 / 중분류 77종 / 소분류 232종

대분류		중분류		소분류	
코드	항목명	코드	항목명	코드	항목명
A	농업,임업 및 어업	01	농업	011	작물재배업
				012	축산업
		02	임업	020	임업
B	광업	05	석탄, 원유 및 천연가스 광업	051	석탄광업
				052	원유 및 천연가스 채굴업

대회 개요

<통계 데이터 인공지능 활용 대회>에서 주어진 학습데이터는 대분류, 중분류, 소분류로 이뤄진 “산업 코드”와 text_obj, text_mthd, text_deal 로 구분된 “산업 설명 텍스트”로 구성되어있습니다.

- 데이터 구성
  - 학습데이터 : 1,000,000 개 / 추론 데이터 : 100,000 개
  - **Target data : digit_1, digit_2, digit_3 (대분류, 중분류, 소분류)**
  - Input data : text_obj, text_mthd, text_deal
  - 특징
    - Digit_3(소분류)가 Unique 하여, 소분류 만으로도 카테고리를 구별할 수 있음
    - 실 데이터에는 text_obj, text_mthd, text_deal 의 구분이 의미 없이 구성되어 있는 경우도 있음
    - 학습 데이터는 모든 산업분류가 포함되지 않음
      - ✓ 대분류 21종 中 19종 / 중분류 77종 中 74종 / 소분류 232종 中 225종

AI_id	digit_1 (대분류)	digit_2 (중분류)	digit_3 (소분류)	text_obj 무엇을 가지고 (원재료, 영업장소 등)	text_mthd 어떤 방법으로 (주요 영업,생산 활동)	text_deal 생산·제공 하였는가 (최종 재화,용역)
id_0000001	S	95	952	카센터에서	자동차부분정비	타이어오일교환
id_0000002	G	46	464	주문에의해	영업장에서	배드민턴용품도매
id_0000003	G	46	463	산업사용자에게	사업장에서	냉동만두,당면등
id_0000004	M	73	733	광고회사,출판업자를 상대로하는 사진관 운영		

결과 개요

최종적으로 7종 9개의 모델을 선정하여 Voting 앙상블 학습을 진행하였으며, 제출한 결과는 Accuracy 91.03 % , F1-Score 77.84 % 입니다.

- Input data 구성

Label	text
S-95-952	카센터에서 자동차부분정비 타이어오일교환
G-47-472	상점내에서 일반인을 대상으로 과일판매

- 10종의 모델 선정 : **bert, kobert, kobart mlbert, albert, asbart, kogpt2, kogpt3, electra, funnel**

- 베스트 모델 선정 :
  - 선정 기준 : Accuracy,F1-score 및, 동일 손실함수 일시엔 train loss와 validation loss의 차이가 적은 모델 선정

- 베스트 앙상블 조합 추출
  - 앙상블 방법 : Voting
  - albert, asbart, asbart_2, funnel, kobert, kogpt2, kogpt3, mlbert, mlbert_2
  - 성능 : Acc - 91.2 / f1 - 79.9122

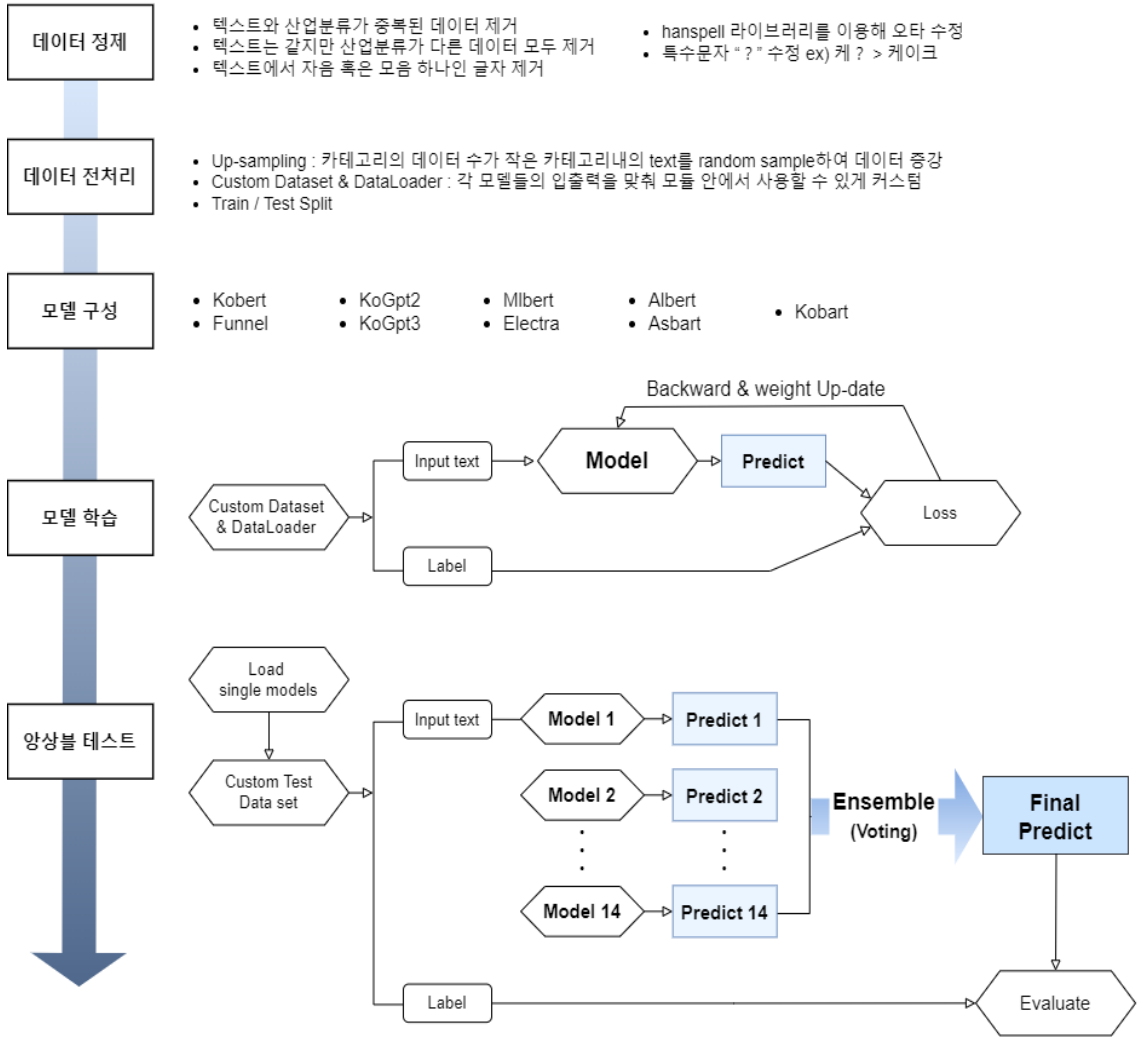
- 최종 점수

Accuracy (%)	F1-score (%)
91.03	77.84

목차

모델 학습 과정 아키텍쳐

모델 학습을 위해 데이터 정제, 전처리를 거쳐 학습에 적합한 데이터를 구성합니다. pretrained된 언어모델 10종을 학습한 다음 앙상블해 최종 결과를 도출했습니다.



성능향상을 위한 시도

성능향상을 위해 데이터 가공, 데이터 불균형 해소, 모델링, 하이퍼 파라미터 튜닝을 시도하였습니다.

데이터 가공

적합하지 않은
학습용 데이터를 수정하여
데이터의 질을 향상시킴

- 데이터 제거
- 맞춤법 및 띄어쓰기 교정
- 결측치 처리
- 오탈자 수정 및 제거

데이터 불균형 해소

카테고리 별 데이터의 불균형
으로 인해 올바른 학습이
방해되는 점을 해소

- UP-Sampling
- FOCAL Loss

모델링

다양한 모델과 앙상블 학습 등
시도하여 성능 개선

- Voting
- Bagging
- 대 중 소 분류별
  모델 구성

하이퍼 파라미터 튜닝

학습 모델이 해당 데이터에
최적화 할 수 있는
모델 **파라미터 탐색**

- Weight Decay
- Learning rate
- Lr-scheduler
- Optimizer

1. 데이터 가공 : 데이터 제거

중복 데이터 및 분류 이상 데이터를 제거함으로써 데이터의 품질을 개선하였습니다.

중복데이터 제거

- 산업 분류(digit_1, digit_2, digit_3)와 텍스트(text_obj, text_mthd, text_deal)가 중복된 데이터를 제거하였음
- 약 30만건

digit_1	digit_2	digit_3	Text_obj	Text_mthd	Text_deal
I	56	561	음식점에서	접객시설을 갖추고	회
I	56	561	음식점에서	접객시설을 갖추고	회

분류 이상 데이터 제거

- 텍스트(text_obj, text_mthd, text_deal)는 같지만 산업 분류(digit_1, digit_2, digit_3)가 다른 데이터 모두 제거하였음
- 약 3천건

digit_1	digit_2	digit_3	Text_obj	Text_mthd	Text_deal
J	62	620	사무실에서	고객의뢰로	소프트웨어개발
J	58	582	사무실에서	고객의뢰로	소프트웨어개발

1. 데이터 가공 : 결측 값 처리

결측 값은 공백으로 변환하여 학습을 진행하였습니다.

‘ ’ (빈 string) 처리

- Input data 형식으로 변환할 때 결측 값을 ‘ ’(빈string)으로 채워주었음

digit_1	digit_2	digit_3	Text_obj	Text_mthd	Text_deal
C	10	102	멸치	입고,가공	
I	56	561		접객시설을 갖추고	회

같은 산업분류 내의 텍스트로 채움

- 소분류 카테고리(‘digit_3’) 내 랜덤하게 채움
  -> 시도하였지만, 답안 제출용 데이터에 적용할 수 없어 최종적으로 채택하지는 않음.

digit_1	digit_2	digit_3	Text_obj	Text_mthd	Text_deal
I	56	561		**매장 에서**	

digit_1	digit_2	digit_3	Text_obj	Text_mthd	Text_deal
I	56	561	접객시설을 갖추고	즉석 조리하여	김밥,라면
I	56	561	**접객시설을 갖추고**	**매장 에서**	**과일 도시락 제공**
I	56	561	음식점에서	접객시설 없이	과일 도시락 제공

1. 데이터 가공 : 오탈자 수정

오타 및 불필요한 문자를 수정하여 데이터 품질을 개선하였습니다.

자음·모음만 있는 글자 수정

- 자음·모음만 있는 데이터 중 유추 가능한 데이터는 단어로 수정하였음
- 자음·모음이 불필요한 경우는 제거 하였음

digit_1	digit_2	digit_3	Text_obj	Text_mthd	Text_deal
I	55	551	펜션에서	일반인 대상으로	ㅅㄱ박 서비스

digit_1	digit_2	digit_3	Text_obj	Text_mthd	Text_deal
I	55	551	펜션에서	일반인 대상으로	숙박 서비스

digit_1	digit_2	digit_3	Text_obj	Text_mthd	Text_deal
I	56	561	접객시석을 갖추고	매장에서ㅏ	타이요리 제공

digit_1	digit_2	digit_3	Text_obj	Text_mthd	Text_deal
I	56	561	접객시석을 갖추고	매장에서	타이요리 제공

특수문자가 포함된 데이터 수정

- 특수문자가 포함된 데이터는 유추 가능한 단어로 수정하였음
- Ex) 피부마사지?에서 -> 피부마사지샵에서
  m^2 -> 제곱미터

digit_1	digit_2	digit_3	Text_obj	Text_mthd	Text_deal
S	96	961	피부 마사지?에서	고객대상	피부마사지

digit_1	digit_2	digit_3	Text_obj	Text_mthd	Text_deal
S	96	961	피부 마사지샵에서	고객대상	피부마사지

digit_1	digit_2	digit_3	Text_obj	Text_mthd	Text_deal
G	47	471	상점에서(165m^2미만)	소비자를 대상으로	음식료품판매

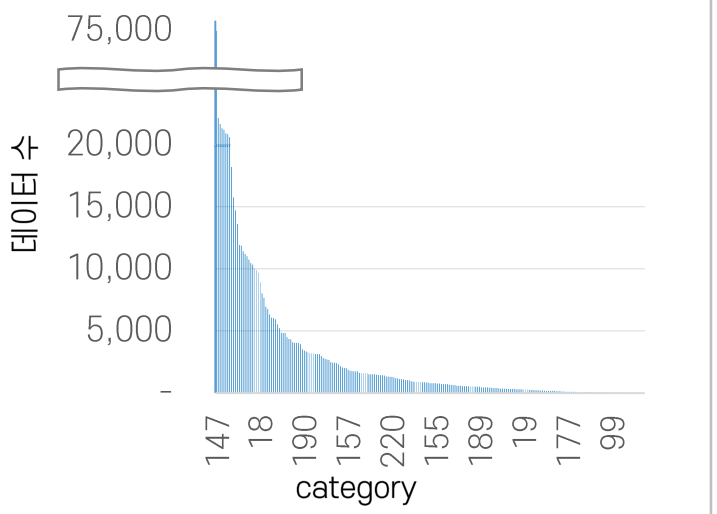
digit_1	digit_2	digit_3	Text_obj	Text_mthd	Text_deal
G	47	471	상점에서(165제곱미터 미만)	소비자를 대상으로	음식료품판매

2. 데이터 불균형 해소 : 데이터 카테고리 분포 현황

전체 데이터의 카테고리 별 분포가 굉장히 편향되어 있어, 이 분포에 따라 Train set 과 test set을 구성하였습니다.

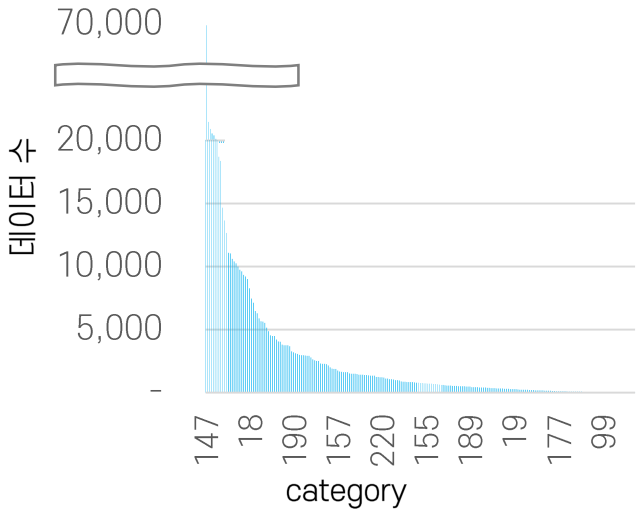
Total_distirbution

- 전처리 후 전체 데이터 수 : 702,685



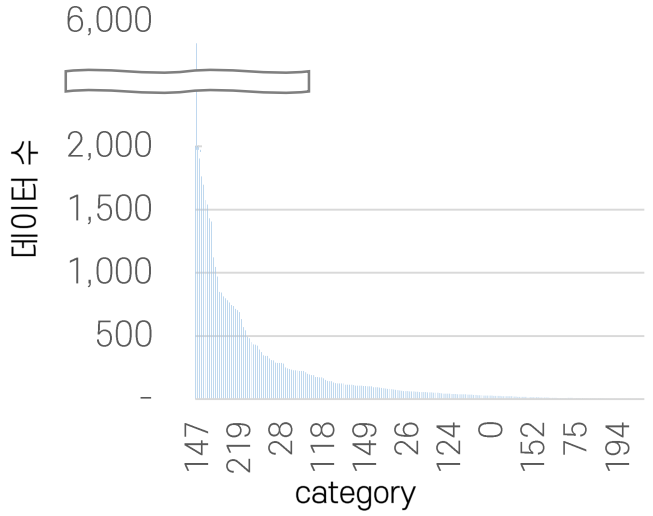
Train_distirbution

- Train set : 652,724
- 소분류 별 데이터가 3개 미만인 경우는 모두 훈련 데이터로 분류하였음



Test_distribution

- Test set : 49,961



2. 데이터 불균형 해소 : Up - sampling

카테고리 별 데이터의 불균형으로 인해 모델 예측이 편향되는 문제를 해소하기 위해 데이터가 적은 카테고리에 대해 Up-sampling을 시도하여 데이터를 보다 균형있게 구성하였습니다.

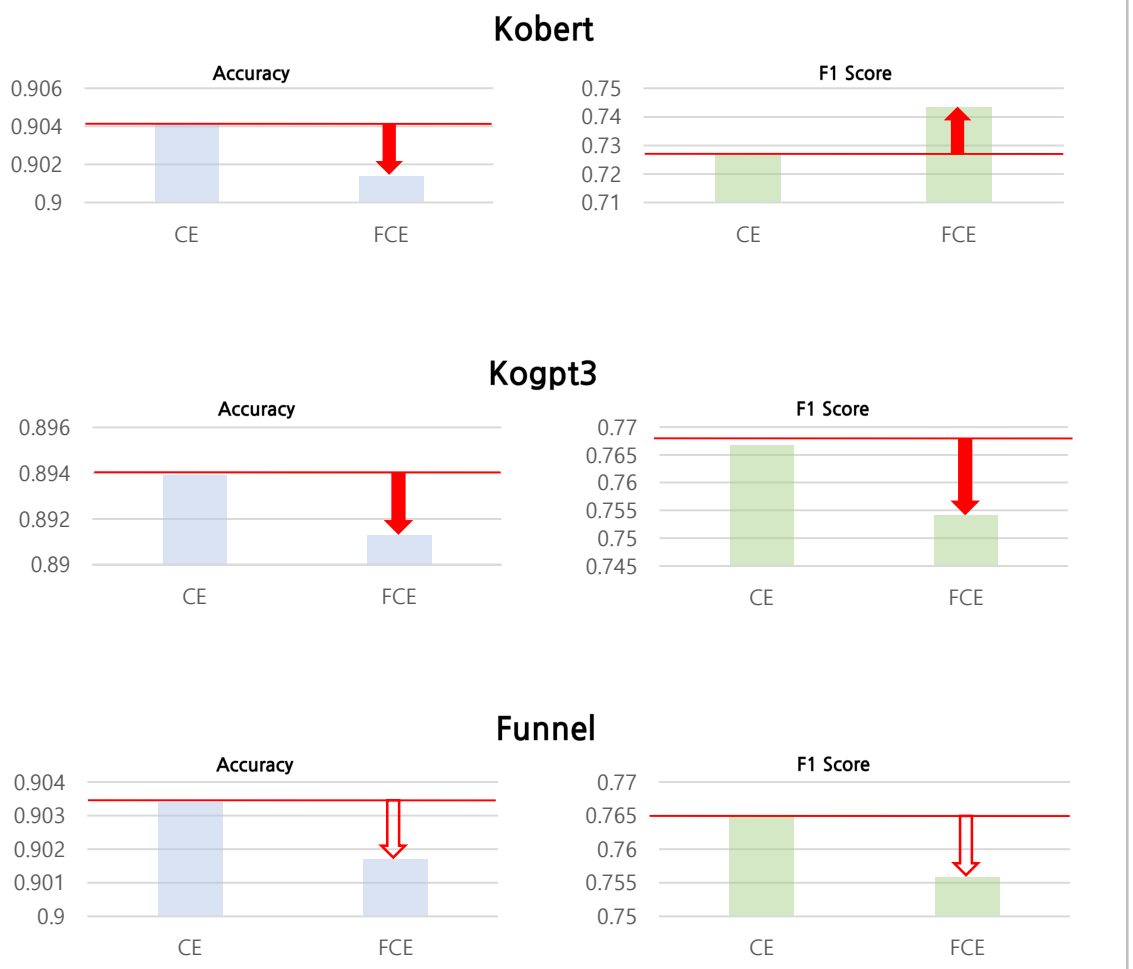
방법

- Up-sample 대상 카테고리를 선정할 최소 데이터 수(minimum) 기준을 정하고, 기준에 부합하지 않는 카테고리에 대해 기준 만큼 Up-sample 합니다.
- 해당 카테고리의 text_obj, text_mthd, text_deal을 정해진 Up-sample 수만큼 random sample하고 이어붙여 새로운 input text를 생성합니다.

Ai_id	Digit_1	Digit_2	Digit_3	Text_obj	Text_mthd	Text_deal
Id_01	F	41	412	현장에서	**고객의 요구에 의해**	조경공사
Id_02	F	41	412	**건설현장에서**	토목공사	도로개설공사
Id_03	F	41	412	태양광	주문에 의한	**태양광 설치 공사**
데이터 생성	**F**	**41**	**412**	**건설현장에서**	**고객의 요구에 의해**	**태양광 설치 공사**

2. 데이터 불균형 해소 : Loss function

Focal Cross Entropy는 정답에 대한 예측 확률이 낮을 수록 높은 가중치를 주는 손실 함수로, 데이터 불균형으로 정답이 편향되는 문제를 개선하기 위해 시도하였습니다.



기대 효과

- **데이터가 적은 카테고리일수록 정답일 때 예측 확률이 낮을 것**이기 때문에 Focal Cross Entropy를 적용해 데이터가 적은 카테고리에 대한 예측 성능을 향상시킬 수 있을 것이다.

검증 방법

- Focal Cross Entropy와 Cross Entropy를 적용해 학습한 모델의 F1 score를 비교한다.

결론

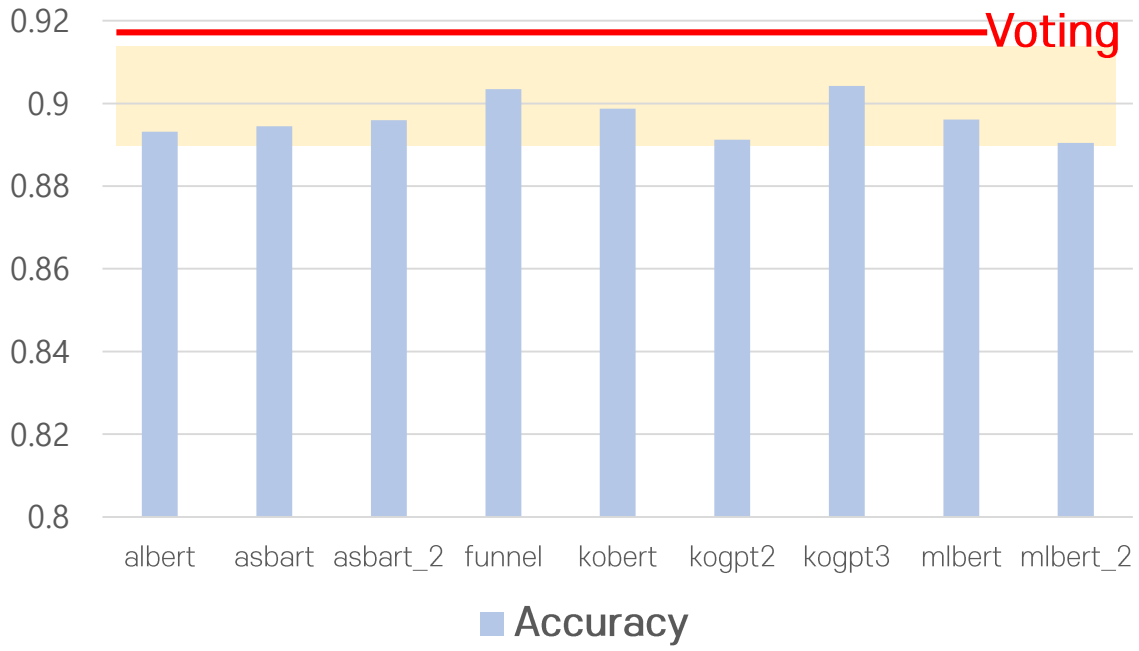
- Kogpt3, Funnel의 경우 Cross Entropy가 Focal Cross Entropy보다 F1 score가 높다.
- 따라서 모델에 따라 더 높은 성능을 보이는 Loss function을 적용하였다.

3. 모델링 : Voting

10개의 싱글 모델들을 여러 번 학습 시켜 기준에 부합하는 베스트 모델을 선정하고, 베스트 모델의 예측 결과를 다양한 조합으로 Voting 했습니다.

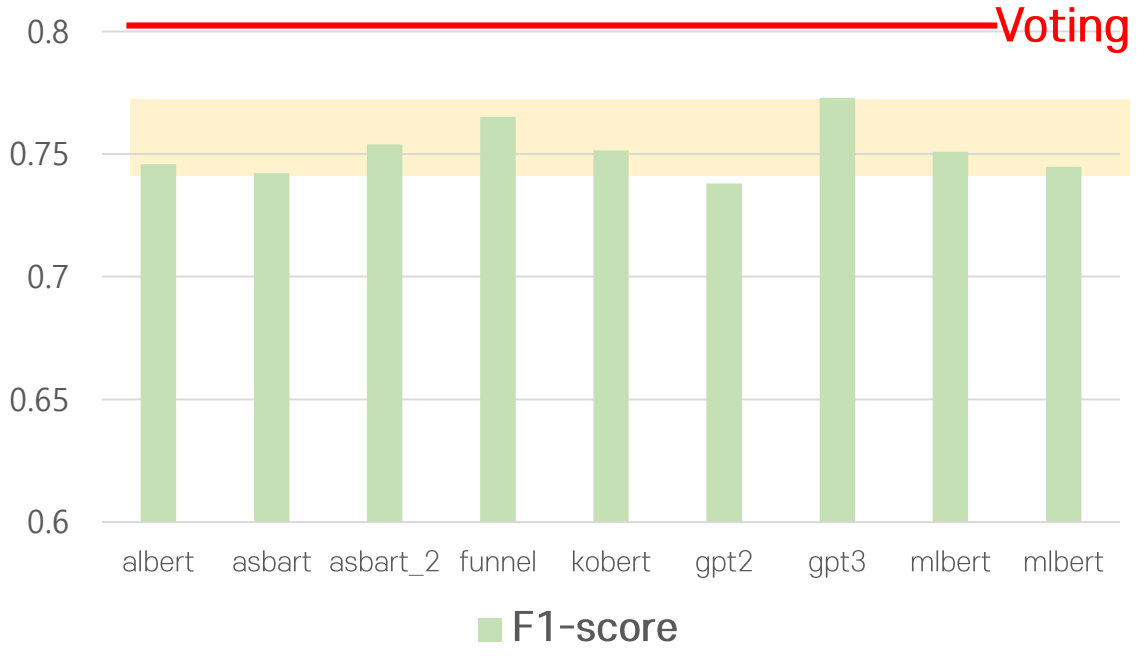
Score

앙상블 모델	Accuracy	F1-score
Voting	0.912	0.79912



베스트 모델 선정 기준

- Accuracy,F1-score를 우선으로 평가하며, 동일 손실함수 일시엔 train loss와 validation loss의 차이가 적은 모델 선정

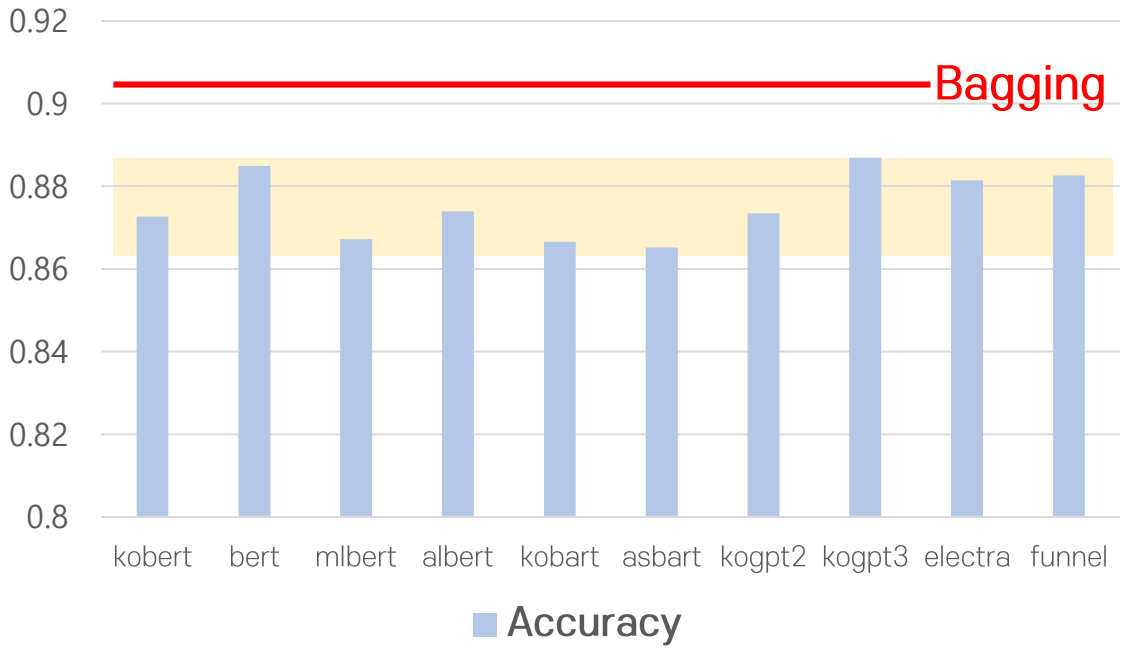


3. 모델링 : Bagging

10개의 싱글 모델 타입에 따라 4개씩, 총 40개 베이스 모델을 학습해 Voting하는 방법으로 Bagging 학습을 진행

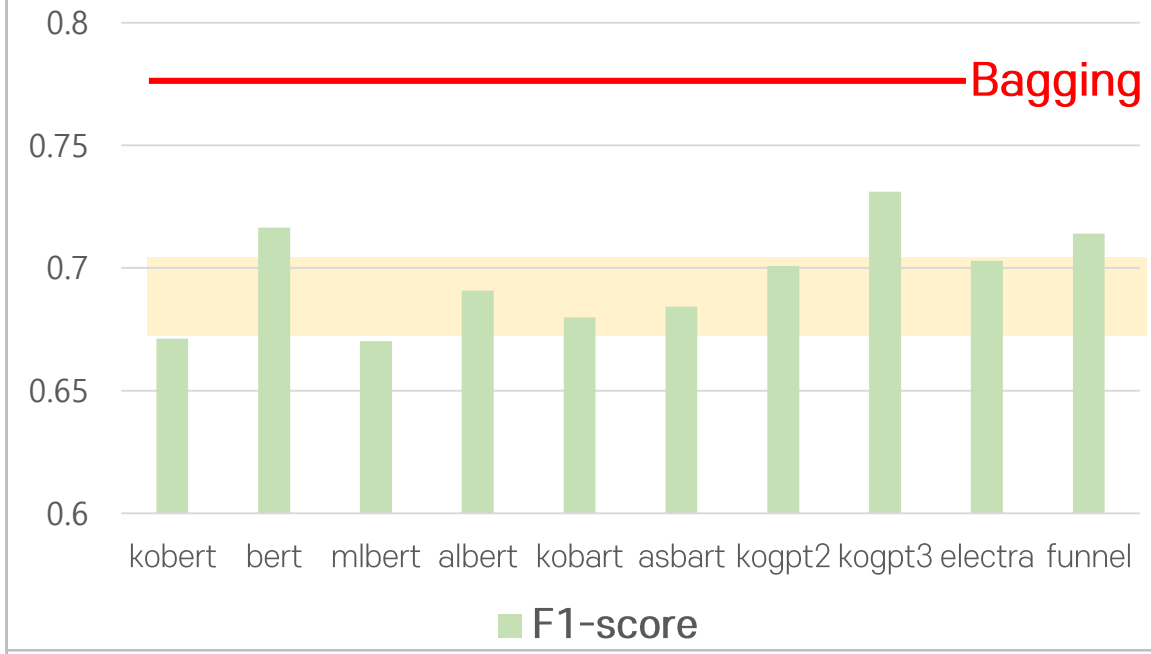
Score

앙상블 모델	Accuracy	F1-score
Bagging	0.903688	0.772277



진행 방법 및 결과

- 싱글 모델 타입에 따라 4개의 베이스 모델 을 만들고, 이렇게 만든 40개의 모델을 Voting 하는 방법으로 Bagging 학습을 진행 하였음.
- 개별 베이스 모델의 학습 성능이 저조하여 최종 성능이 기대에 미치지 못하였다. 학습 데이터 수가 작았기 때문이라고 추측한다. (Sub sampling : 200,000)

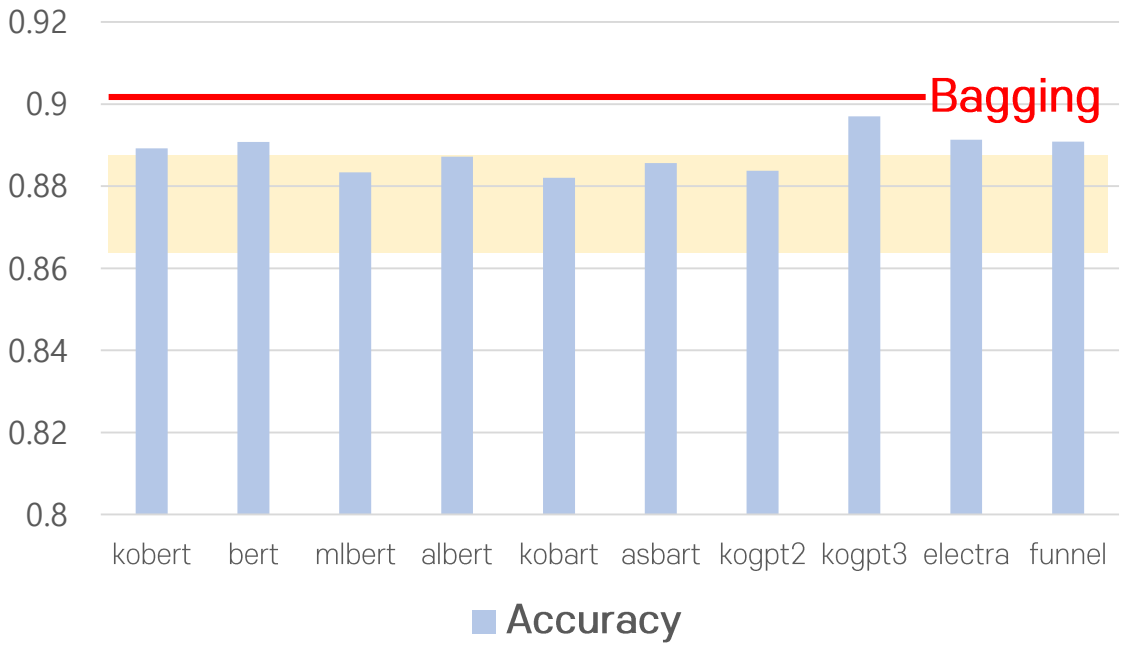


3. 모델링 : Bagging

가설 검증을 위해 학습 데이터 수를 2배인 400,000개로 늘려서 Bagging 학습을 진행

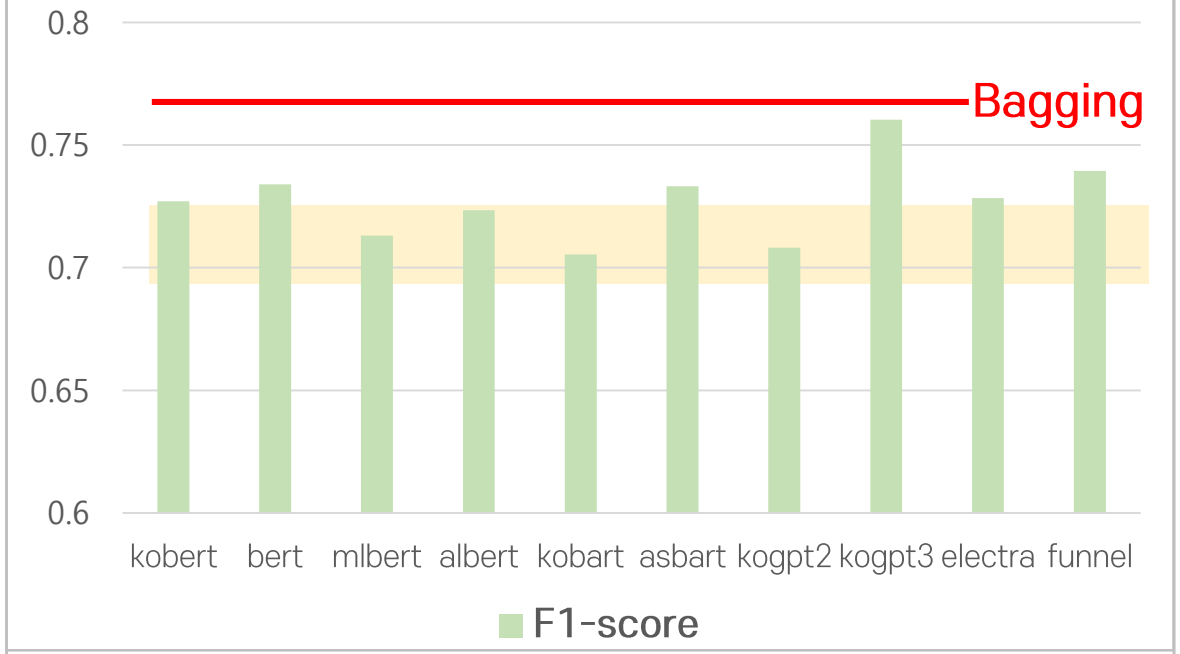
Score

앙상블 모델	Accuracy	F1-score
Bagging	0.907972	0.776981



결과

- 싱글 모델 타입에 따라 4개의 베이스 모델 을 만들고, 이렇게 만든 40개의 모델을 Voting 하는 방법으로 Bagging 학습을 진행 하였음.



3. 모델링 : 카테고리별 개별 모델 구성

심사 기준을 참고해 대·중·소분류를 개별적으로 예측하는 모델을 학습한 결과 Accuracy는 큰 변화가 없었지만, F1 score는 유의미한 변화를 보였습니다.

카테고리별 개별 모델의 필요성

통계데이터 인공지능 활용대회 심사 방법

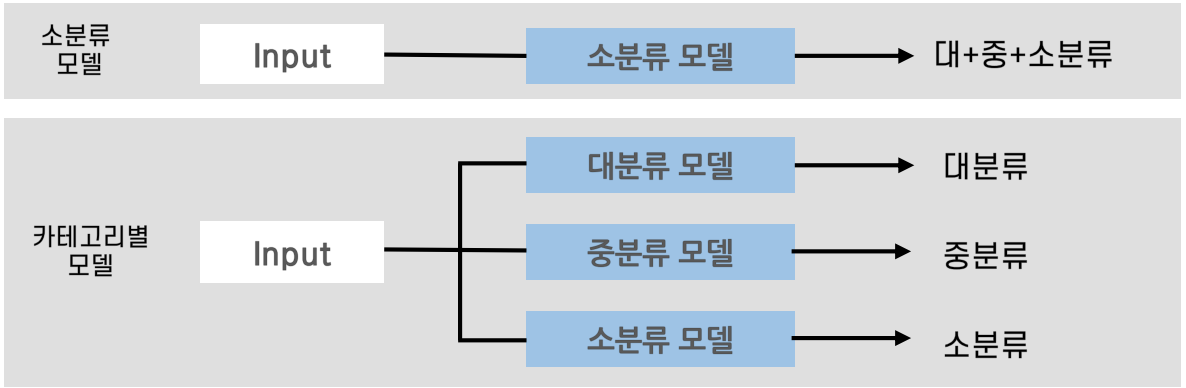
□ 1차 심사 : 다중분류 혼동행렬 모형의 Accuracy*, F1-score

- **부분 점수 : 대분류 예측(0.1점), 중분류 예측(0.2점), 소분류 예측(0.7점)**

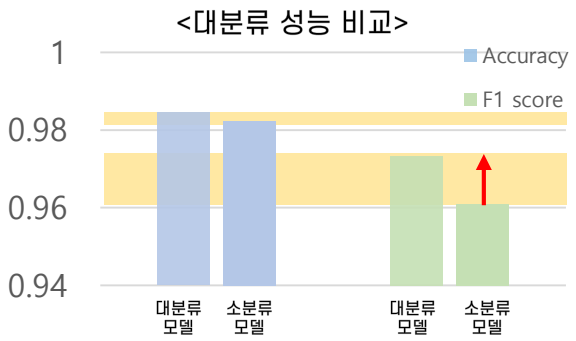
…

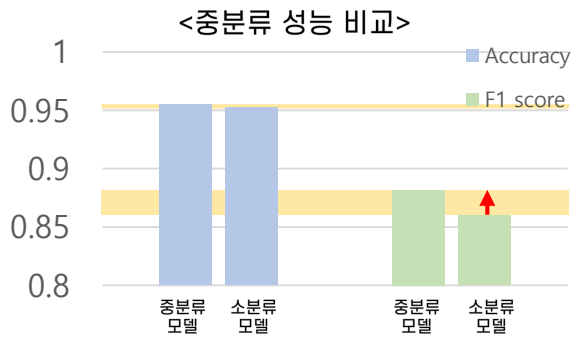
- 대회의 채점기준은 대·중·소분류에 대해 각각 부분 점수를 주고 있음
- 소분류로만 예측하는 경우 소분류가 오답일 때 대분류, 중분류 모두 오답일 확률이 클 것
- 대·중·소분류 별 모델을 학습해 대분류, 중분류를 소분류와 독립적으로 예측한다면 소분류가 오답이어도 대분류, 중분류를 정확히 예측해 Accuracy와 F1 Score가 향상될 것을 기대

카테고리별 개별 모델



❖ Input은 text_obj, text_mthd, text_deal 세 개의 변수를 합친 값을 의미한다.



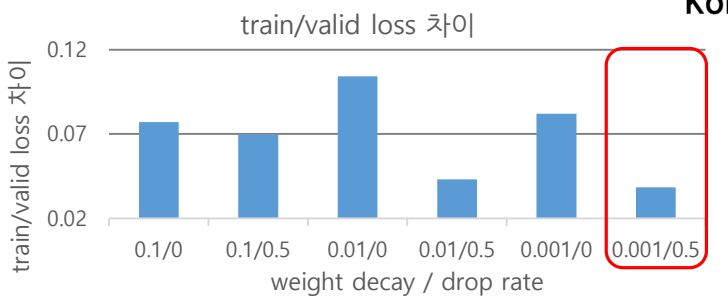


✓ Accuracy의 차이는 미미하지만 F1 score의 경우 소분류 모델 보다 카테고리별 개별 모델이 더 높게 나타남

4. 하이퍼 파라미터 튜닝 : Weight Decay / Drop out

Weight Decay와 Drop out의 값을 증가시킬수록 과적합을 해결하여 모델의 일반화 성능을 향상시킬 것으로 예상했지만, 모델에 따라서 적절한 Weight decay와 drop out의 값이 달랐습니다.

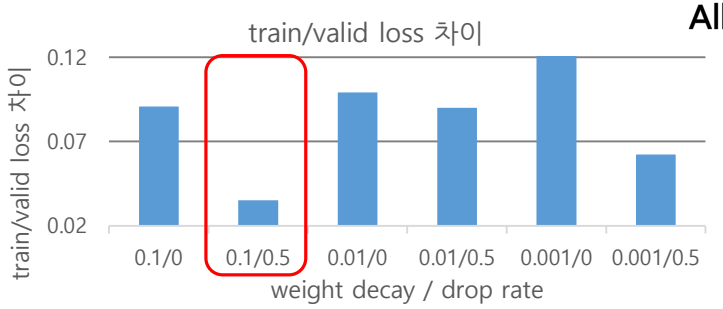
Kobert



Weight Decay, Drop rate 별 valid 성능

Weight Decay	Drop rate	Accuracy	F1 score
0.1	0	0.8971	0.7471
0.1	0.5	0.8988	0.7507
0.01	0	0.8984	**0.7587**
0.01	0.5	0.8983	0.7389
0.001	0	0.8965	0.7345
0.001	0.5	**0.8999**	0.7442

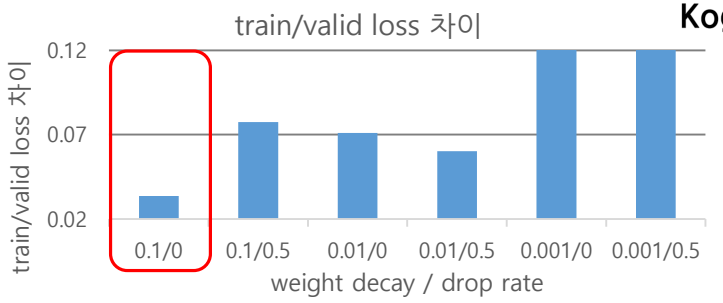
Albert



Weight Decay, Drop rate 별 valid 성능

Weight Decay	Drop rate	Accuracy	F1 score
0.1	0	0.8929	0.7343
0.1	0.5	**0.8937**	**0.7402**
0.01	0	0.8920	0.7364
0.01	0.5	0.8953	0.7380
0.001	0	0.8897	0.7298
0.001	0.5	0.8935	0.7308

Kogpt2



Weight Decay, Drop rate 별 valid 성능

Weight Decay	Drop rate	Accuracy	F1 score
0.1	0	0.8922	0.7462
0.1	0.5	**0.8970**	0.7451
0.01	0	0.8922	0.7473
0.01	0.5	0.8950	0.7409
0.001	0	0.8941	0.7471
0.001	0.5	0.8969	**0.7485**

기대 효과

- Weight decay와 Drop out rate의 값을 증가시켜 모델의 일반화 성능을 향상시킬 것이다.

검증 방법

- Weight Decay 0.1, 0.01, 0.001, drop out을 0.5로 적용하거나 적용하지 않는 경우의 수를 조합해 train loss와 valid loss의 차이가 얼마나 작아지는지 확인한다.

결론

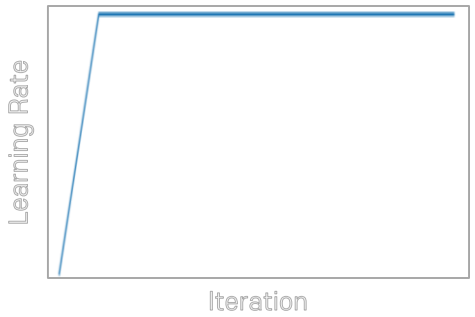
- 모델에 따라 좋은 성능을 내는 Weight decay와 Drop out rate의 값이 모두 다르다.
- 따라서, 모델마다 가장 좋은 일반화 성능을 보이는 값으로 설정해서 학습한다.

4. 하이퍼 파라미터 튜닝 : Lr scheduler

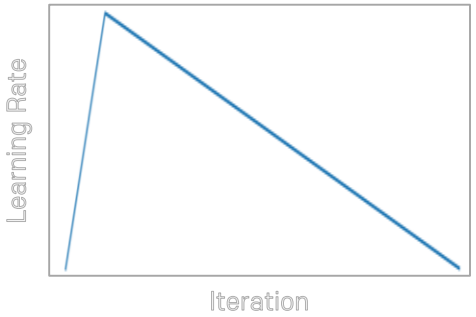
Lr scheduler는 학습 경과에 따라 Learning rate를 변경합니다. 총 네 가지 Lr sheduler를 비교해 모델마다 좋은 성능을 도출한 Lr sheduler를 적용했습니다.

다양한 LR Scheduler의 사용

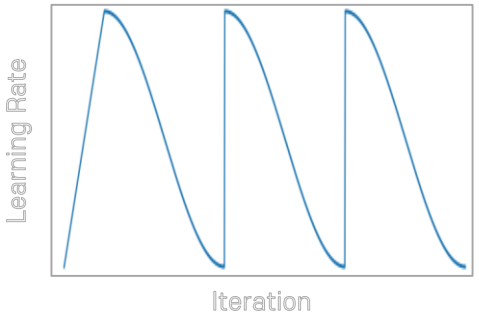
<Constant with warmup>



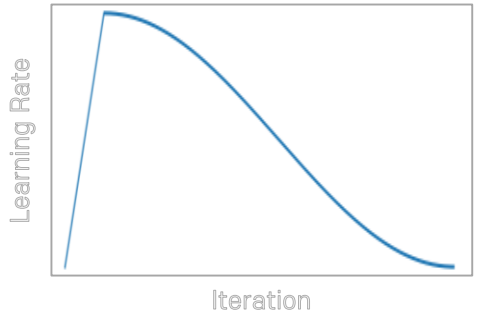
<Cosine with warmup>



<Cosine with restarts>

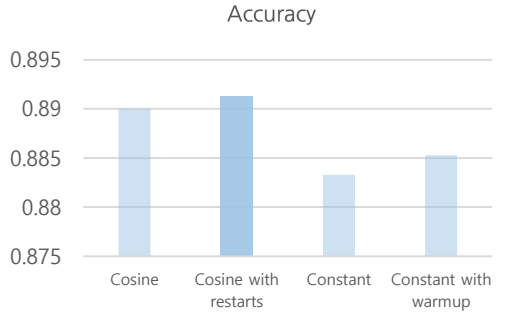


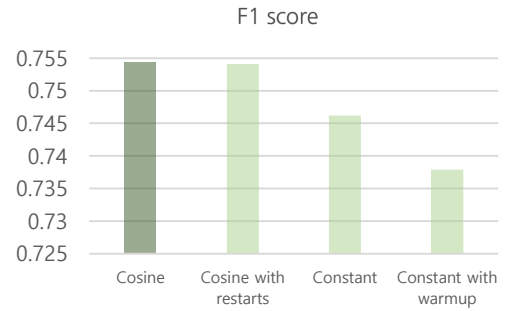
<linear>



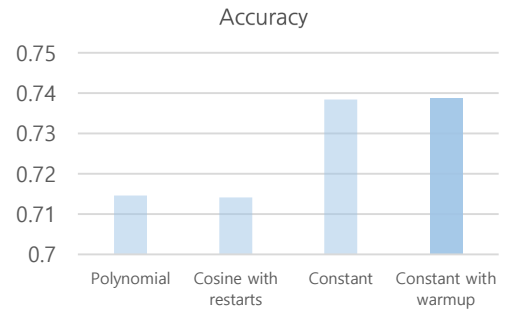
모델 별 Scheduler에 따른 성능

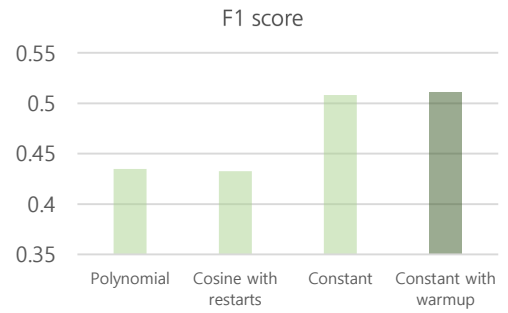
Kogpt3





Albert



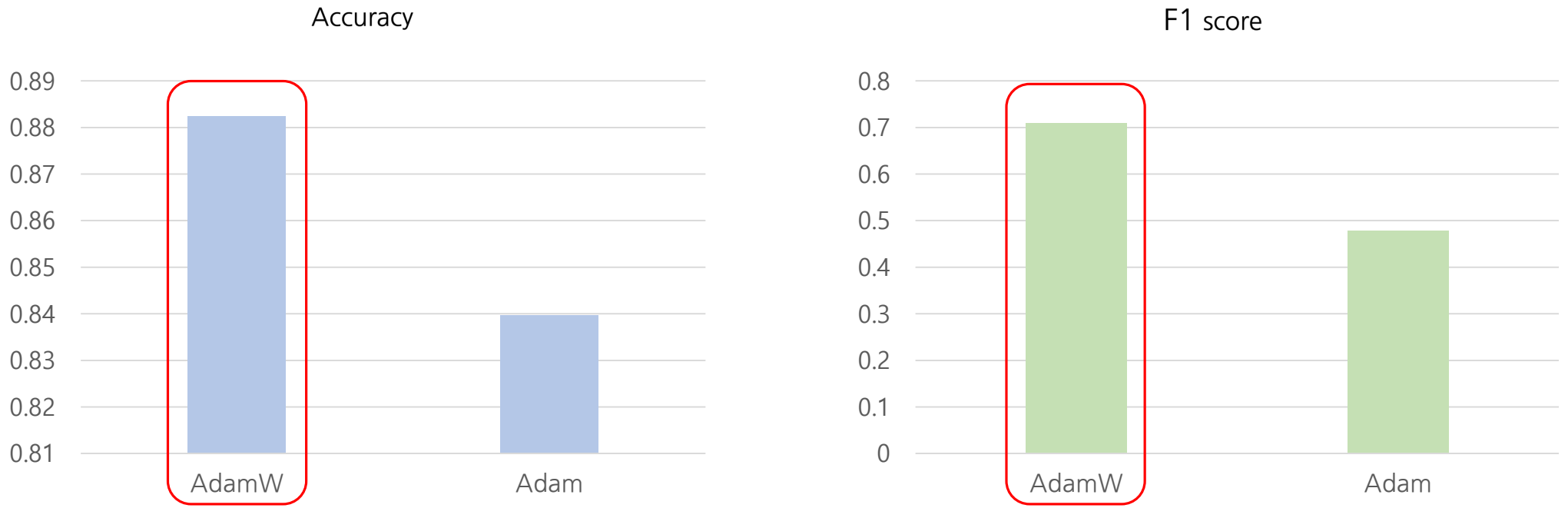


4. 하이퍼 파라미터 튜닝 : Optimizer

중복 데이터 및 분류 이상 데이터를 제거함으로써 데이터의 품질을 개선하였습니다.

Optimizer별 성능 비교

- Kogpt2를 기준으로 AdamW가 Adam보다 우수한 성능을 보임



목차

소감 및 마무리

산업분류코드 분류 AI대회를 통해 배운 점과 아쉬운 점, 그리고 한 줄 소감

<배운 점>
- 언어 모델(Language Model)의 텍스트 분류 Fine-Tuning 구현(PyTorch)
- Bagging 앙상블 구현(PyTorch)

<아쉬운 점(개선방안)>
- Github을 통한 협업을 시도했지만, Github을 익히는데 많은 시간이 소요될 것으로 예상되어 포기함.
- 데이터 가공, 전처리, 하이퍼파라미터 튜닝 등 다양한 시도에 대한 계획적인 실험을 시도했지만, 제대로 이뤄지지 않음.

<한 줄 소감>
- 시간에 쫓기더라도, 차분히 계획하고 기록하면서 실험해야 한다고 느꼈습니다. AI 대회 등 머신러닝 학습을 많이 경험해보면서 계획적인 실험의 노하우를 쌓아가야 할 것 같습니다.