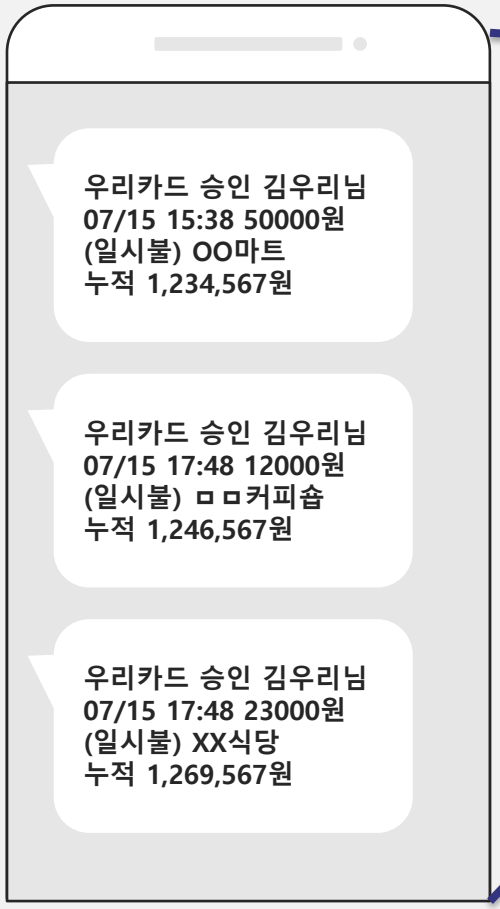
가맹점 분류 Project

Index

프로젝트 소개

가맹점 업종 분류 Project : 카드 결제 내역을 자연어 처리를 이용해 알맞은 업종으로 분류



자연어 처리

업체명 (Input)	업종 (Output)
OO마트	대형마트
ㅁㅁ커피숍	카페
XX식당	외식

고객 개인화 서비스 제공

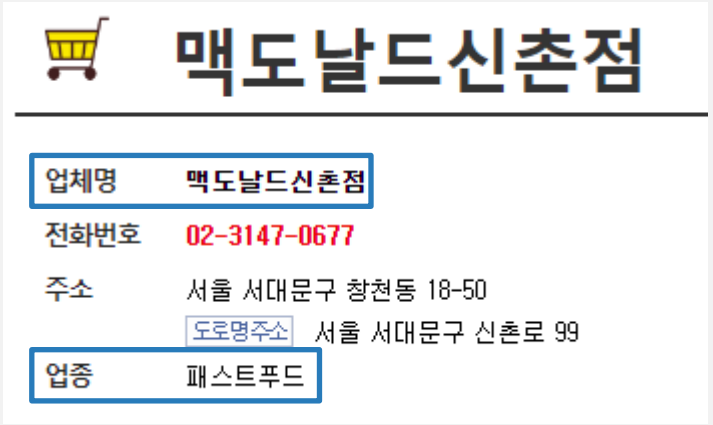
데이터 수집 및 전처리

- 데이터 설명
- 라벨링 방법
- 전처리 방법

데이터 설명

우리은행 카드 데이터 사용 불가로 인해 '한국 전화번호부' 사이트의 업체명, 업종 데이터 사용

- **한국전화번호부 웹사이트에서 크롤링 (업체명 / 업종)**



업체명	업종
맥도날드신촌점	패스트푸드
투썸플레이스신촌기차역점	커피전문점
…	…

→ **2,649,975개**의 업체 / **1,998개** 업종

Issue : 너무 다양한 업종 분류

- **절대적인 업종의 수가 많음** → **업체가 1000개 이상 존재하는 업종만 고려**
- **일반적으로 소비하지 않는 업종 존재 (ex. 건설자재)** → **일반적이지 않은 소비 업종 제외**
- **한국전화번호부 업종 분류 ≠ 우리은행 업종 분류** → **자체적 Labeling 방법 고려**

→ 업체 수 : **2,649,975개 → 1,621,210**

데이터 Labeling

‘한국 전화번호부’의 업종 분류를 우리은행의 업종 분류에 맞추어 Labeling : 업종 별 키워드 사용

1. 우리은행 업종별로 키워드 선정

[빵, **커피**, **카페**, **제과**, **베이커리**, 다방] → 카페
['사회복지', '**양탄자**', '장례', '**결혼정보**', '사주, '**회계**', '세무' ...] → 기타소비

2. 한국 전화번호부의 업종명에 키워드가 해당된다면 매칭

3. 키워드가 두 개 이상의 업종에 해당하거나

4. 어디에도 해당되지 않는다면

Hand Labeling 으로 분류

데이터 전처리

1. 업체명과 업종이 같은 데이터 제거

```
df[df['업체명'] == '스타벅스']
```

	업체명	업종
128037	스타벅스	카페
169968	스타벅스	카페
169969	스타벅스	카페
169970	스타벅스	카페
169971	스타벅스	카페
...	...	...
248404	스타벅스	카페
254540	스타벅스	카페
814844	스타벅스	카페
891320	스타벅스	카페
891321	스타벅스	카페

131 rows × 2 columns

데이터 개수

1,621,210 → 1,007,984

```
df_ = df.drop_duplicates()
df_[df_['업체명'] == '스타벅스']
```

	업체명	업종
128037	스타벅스	카페

데이터 전처리

2. 업체명은 같으나 업종이 다른 데이터는 업종 분포 확인 후 가장 많은 업종으로 분류

```
df_[df_['업체명'] == '토마토']
```

	업체명	업종
101975	토마토	생필품
209014	토마토	통신
209015	토마토	의료/건강
274266	토마토	편의점
285106	토마토	외식
389656	토마토	취미/여가
416794	토마토	미용
425397	토마토	카페
431328	토마토	주류/펍
467921	토마토	대형마트
773981	토마토	여행/숙박
986427	토마토	생활서비스

```
df[df['업체명'] == '토마토']['업종'].value_counts()
```

```
주류/펍       33
생필품        26
외식         25
미용          5
카페          3
취미/여가       2
여행/숙박       2
통신          1
의료/건강       1
편의점         1
대형마트        1
생활서비스       1
Name: 업종, dtype: int64
```

데이터 개수
1,007,984 → 969,168

```
df_1 = df_1.drop_duplicates()
df_1[df_1['업체명'] == '토마토']
```

	업체명	업종
101975	토마토	주류/펍

'토마토'라는 업체명은 주류/펍으로 가장 많이 분류되었으므로 모두 주류/펍으로 분류

데이터 전처리

3. 정규표현식으로 기호 없앤 후 중복되는 업체명은 제거

```
data['업체명_r'] = data['업체명']
data['업체명_r'] = data['업체명_r'].str.replace('[-=+,#/₩?:^$#.@+₩"※~&%ㆍ!』₩₩‘|₩(₩)₩[₩]₩<₩>`₩'…》]',' ')
data['업체명_r'] = data['업체명_r'].str.strip()
```

	업체명	업체명_r	업종
2961	햇빛병원(산부인과)	햇빛병원 산부인과	의료/건강
3485	인정병원(산부인과)	인정병원 산부인과	의료/건강
3528	3S(에스)한의원	3S 에스 한의원	의료/건강
5319	한빛아동병원(소아과)	한빛아동병원 소아과	의료/건강
5426	(재)한마음행복나눔한의원	재 한마음행복나눔한의원	의료/건강
...	...	...	...
969145	메디팜태평양약국(전민)	메디팜태평양약국 전민	의료/건강
969155	(주) 유프리즘	주 유프리즘	기타소비
969159	(주)잉카엔트웍스	주 잉카엔트웍스	기타소비
969160	엔에스 (NS) 정보통신	엔에스 NS 정보통신	통신
969162	(재)인천교구천주교회유지재단	재 인천교구천주교회유지재단	생활서비스

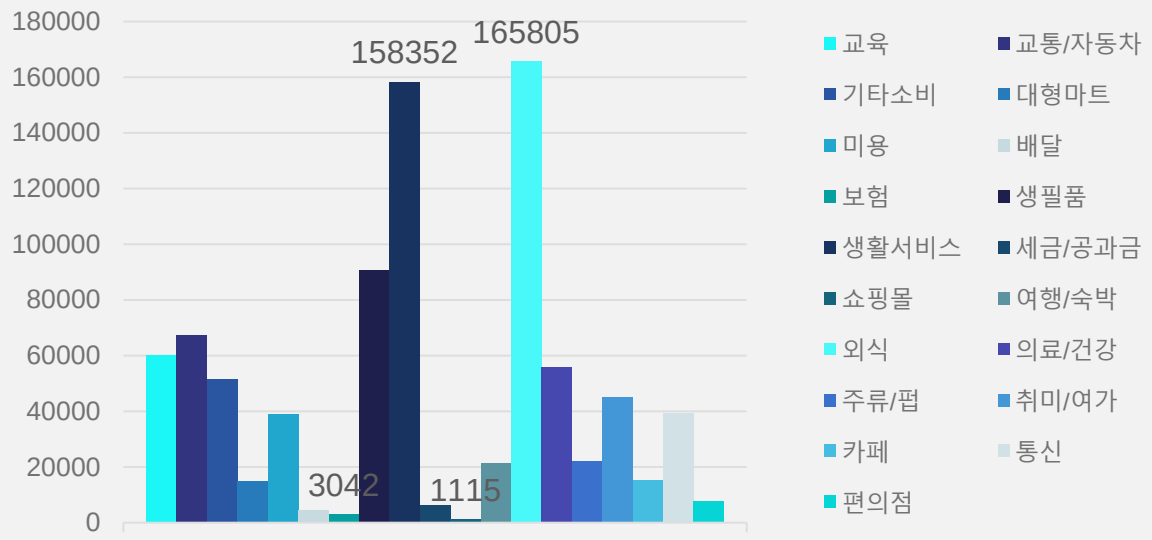
	업체명	업체명_r	업종
1133	베이징 덕	베이징 덕	외식
520358	베이징.덕	베이징 덕	외식

중복 행 제거

	업체명	업체명_r	업종
1133	베이징 덕	베이징 덕	외식

최종 데이터 개수 965,404개

쇼핑몰 1115개, 보험 3042개, 외식 165805개 등 클래스 분포가 불균형함



실험 및 결과

- **실험 구성**
- **모델 설명**
  - KoBERT
  - KoELECTRA
  - RoBERTa
  - 1D CNN
- **실험 결과**
  - 성능 지표 설명
  - 각 모델 별 스코어
  - 오분류 Case

실험 구성

- **Train, Valid, Test dataset**

전체 96만개의 데이터 중 Test 10만개로 분리

86만개 중 8:2의 비율로 Train 69만개, Valid 17만개로 분리

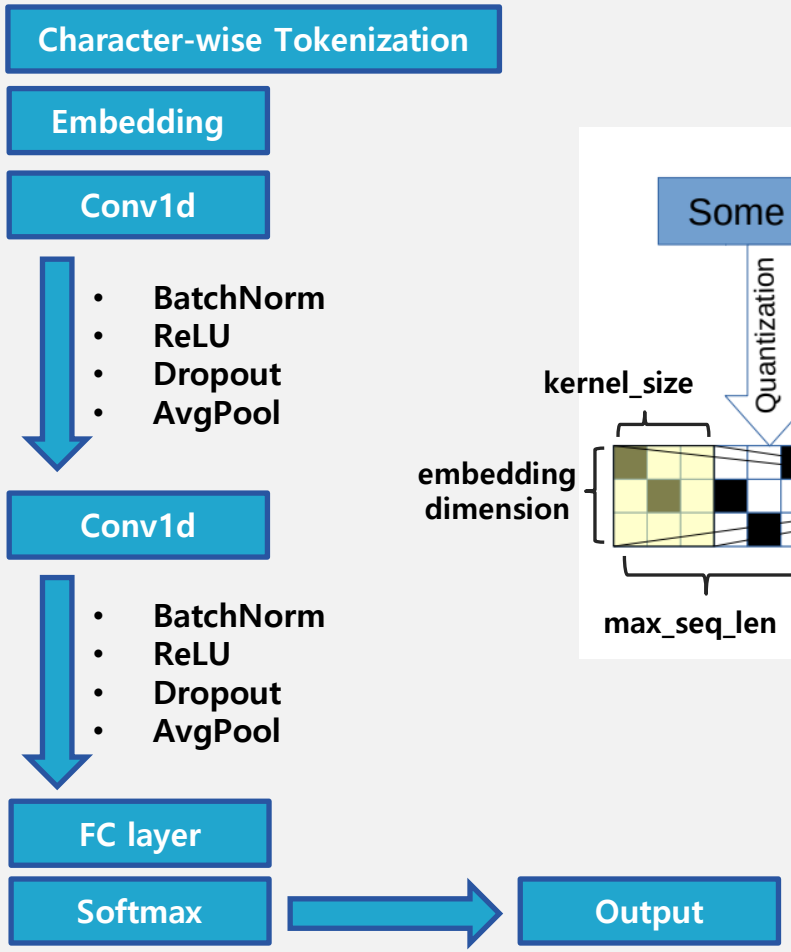
영어 uncased(모두 소문자로 변환한 버전) / cased(대문자 살린 버전)/nonumber(숫자 제거한 버전) 세 종류의 데이터셋

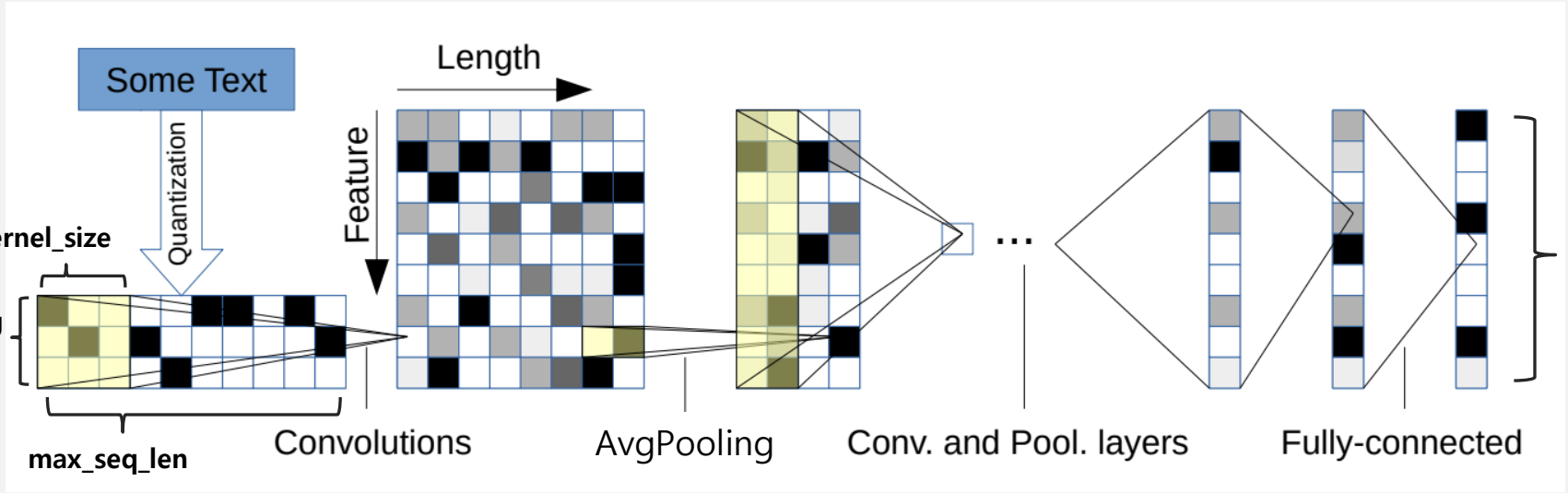
- **사용한 모델**
  - 1D CNN
  - KoBERT
  - KoELECTRA
  - RoBERTa

- **실험 변수**
  - 시퀀스의 최대 길이 (Max Seq Len / 26, 32)
  - 손실함수 (Loss Function / CrossEntropy, FocalLoss, WeightedCrossEntropy)

모델 설명

1D CNN – Model Structure





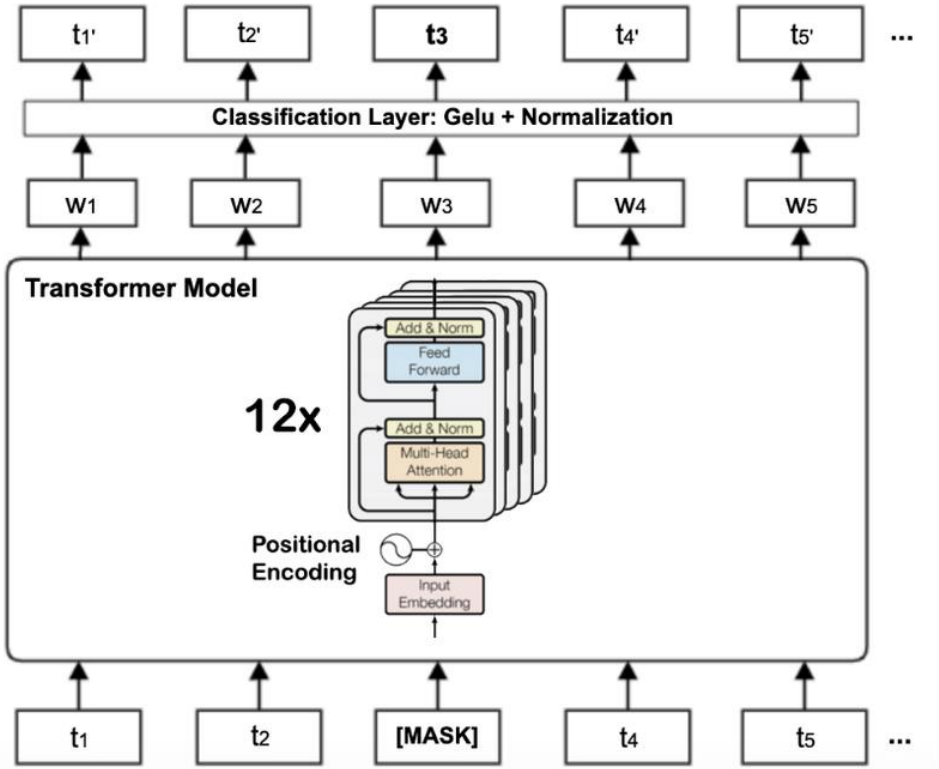
- 음절 단위 토큰화
- Sequential Data 에 용이함

Zhang, Xiang, Junbo Zhao, and Yann LeCun. "Character-level convolutional networks for text classification." *Advances in neural information processing systems* 28 (2015).

모델 설명

KoBERT

Korean **B**idirectional **E**ncoder **R**epresentations for **T**ransformers



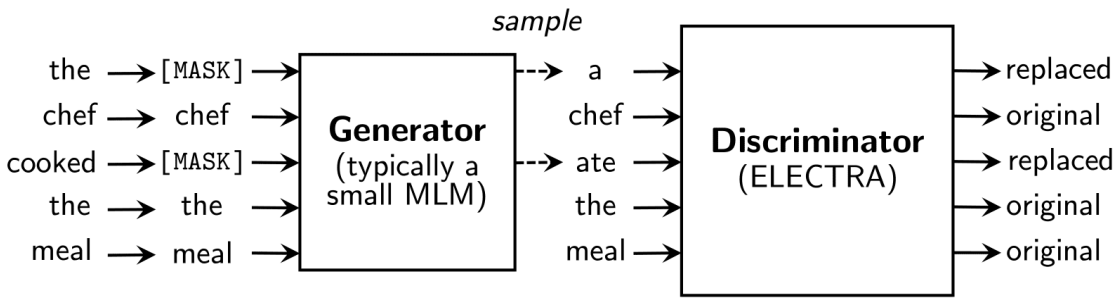
BERT는 구글이 발표한 트랜스포머 기반의 사전학습 모델로, 한국어에 대한 성능의 한계를 극복하기 위해 KoBERT가 개발됨
위키피디아, 뉴스 등에서 수집한 500만 여개의 문장으로 이루어진 코퍼스에 대해 학습했으며, 데이터로 훈련된 SentencePiece 토크나이저를 사용하여 기존 대비 27%의 토큰으로 2.6% 이상의 성능 향상을 기록

KoBERT의 베이스라인 모델인 BERT base 모델은 12개의 트랜스포머 층, 768차원의 hidden size, 12개의 어텐션 헤드로 구성되어있음.

Devlin, Jacob, et al.
"Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).
https://sktelecom.github.io/project/kobert/

모델 설명

KoELECTRA



Efficiently **L**earning **E**ncoder that **C**lassifies **T**oken **R**eplacements **A**ccurately

BERT는 하나의 데이터 중 15%에 대해서만 학습하기 때문에 손실이 발생하기 쉽고 비용이 많이 들며, 학습 시에는 마스킹 토큰을 모델이 학습하지만 추론 과정에서는 마스킹 토큰이 존재하지 않는다는 한계가 존재.

generator 모델을 이용해 일부 토큰을 가짜 토큰으로 바꾸고, discriminator 모델이 해당 토큰이 가짜인지 진짜인지 분류하는 Replaced Token Detection 태스크 적용

Clark, Kevin, et al. "Electra: Pre-training text encoders as discriminators rather than generators." *arXiv preprint arXiv:2003.10555* (2020).

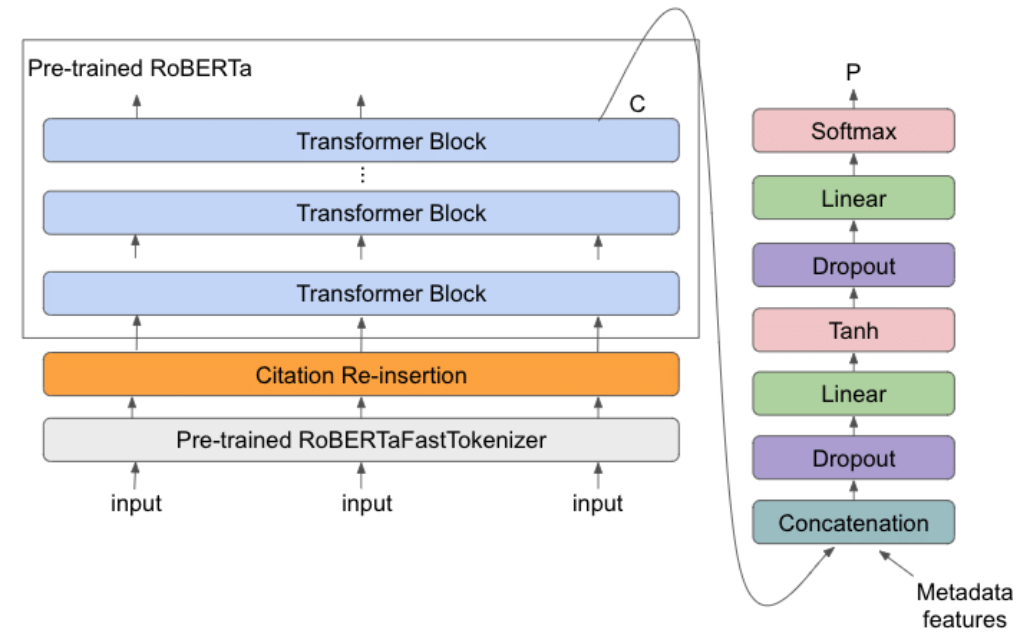
모델 설명

RoBERTa

Robustly **o**ptimized **BERT** pretraining **a**pproach

RoBERTa는 BERT가 underfitted된 점을 극복하기 위해, BERT와 비교해서 더 큰 batch size와 방대한 데이터로 학습되었고, masking하는 토큰의 위치를 계속 바꾸는 Dynamic Masking 적용한 모델

Klue/RoBERTa는 KLUE Benchmark에서 공개한 한국어 자연어 이해 데이터셋으로 사전학습되었음.



Liu, Yinhan, et al. "Roberta: A robustly optimized bert pretraining approach." *arXiv preprint arXiv:1907.11692* (2019).
https://github.com/KLUE-benchmark/KLUE

지표 설명

모델 별 성능을 평가하기 위해 세 가지의 지표 사용

Confusion Matrix

		Predicted: Negative	Predicted: Positive
Actual	Negative	TN	FP
	Positive	FN	TP

Accuracy (정확도)

실제 데이터에서 예측 데이터가 얼마나 같은지 판단

불균형한 데이터에 대해서는 적합한 평가를 내리기 어려움

$$Accuracy = \frac{\text{올바르게 예측한 데이터 개수}}{\text{전체 데이터 개수}} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision (정밀도), Recall (재현율)

$$\text{Precision} = \frac{\text{실제로 } True\text{인 샘플 개수}}{True\text{라고 예측한 샘플 개수}} = \frac{TP}{TP+FP}$$

$$Recall = \frac{True\text{라고 예측한 샘플 개수}}{\text{실제로 } True\text{인 샘플 개수}} = \frac{TP}{TP+FN}$$

F1 Score

Precision과 Recall의 조화평균

$$F1\ Score = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

Macro , Micro, Weighted F1 Score

Multi class classification을 위한 F1 Score

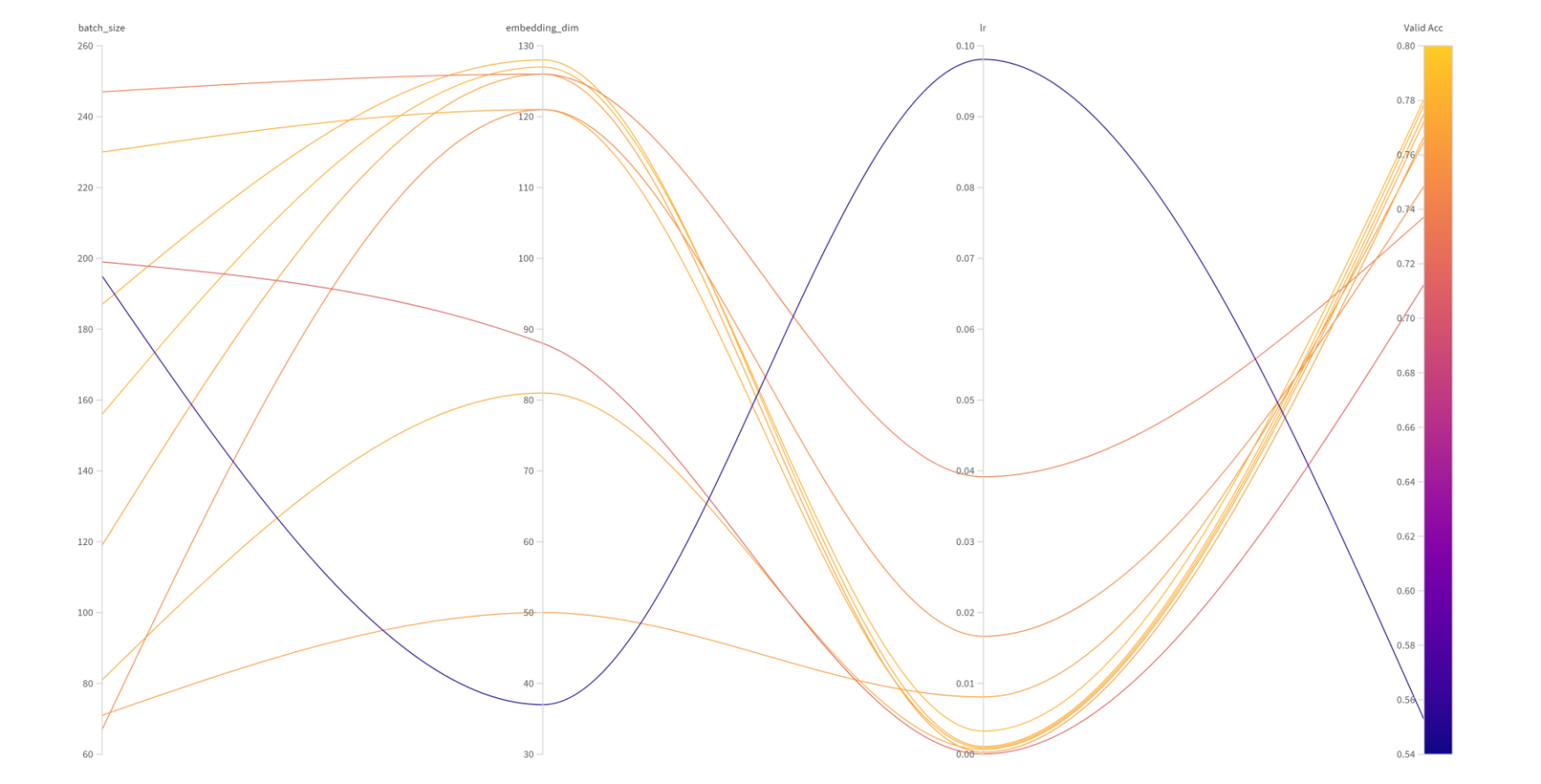
$$Macro\ F1\ Score = \ Average(F1\ score\ of\ each\ class)$$

$$Micro\ F1\ Score = \ Accuracy$$

$$Weighted\ F1\ Score = Weighted\ Average(F1\ score\ of\ each\ class)$$

실험 결과 (1d CNN_1)

Hyperparameter setting에 민감함. 다양한 Hyperparameter 사용하여 실험함.



Sweep Config :

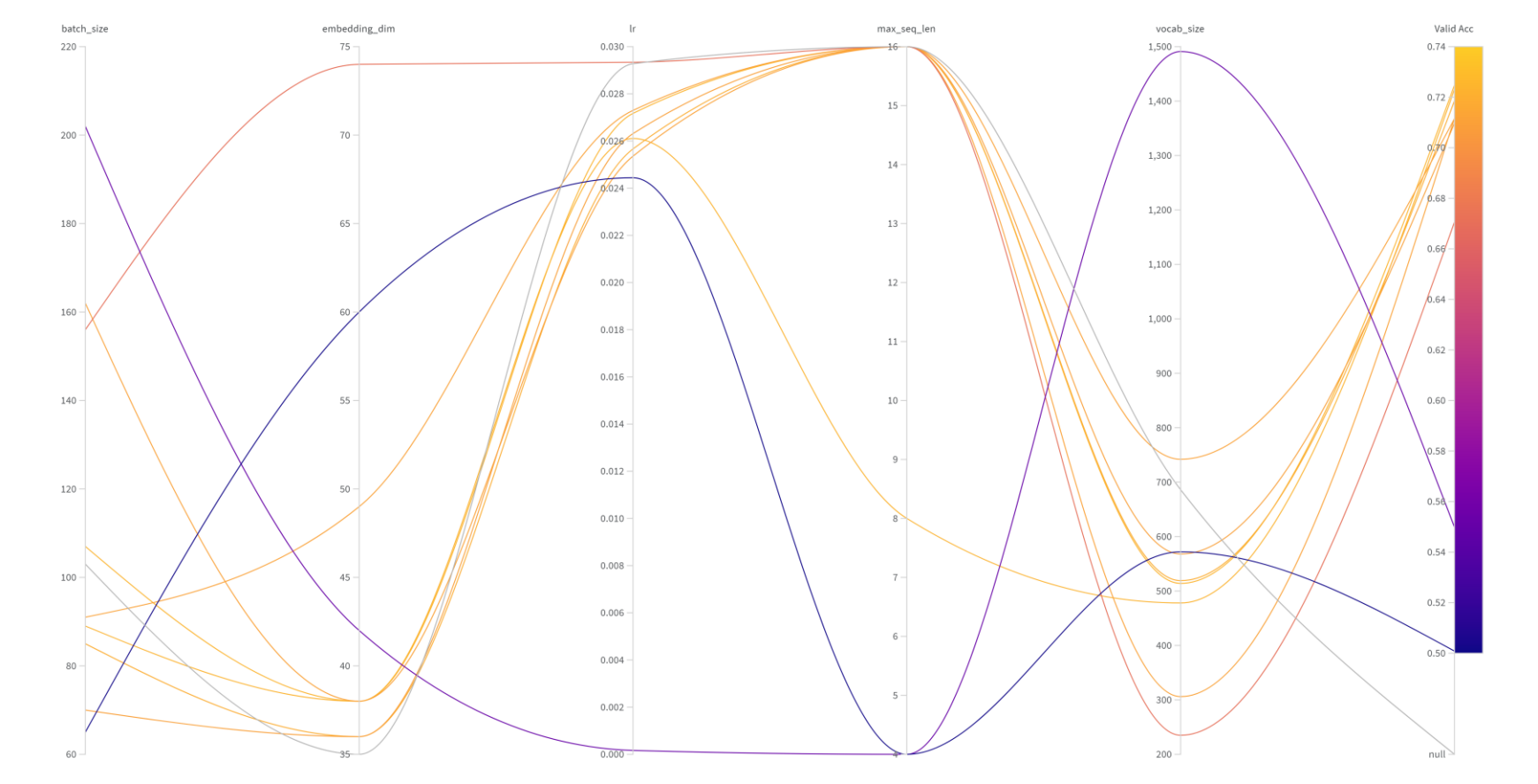
- **Goal : Maximize Valid Acc**
- **batch size : 64 ~ 256**
- **embedding dim : 32 ~ 128**
- **Learning rate : 0.00001 ~ 0.1**

Result :

- **batch size, embedding dim : 모델 성능에 큰 영향 없음**
- **learning rate : 0.05보다 작은 값으로 설정**

실험 결과 (1d CNN_2)

Hyperparameter setting에 민감함. 다양한 Hyperparameter 사용하여 실험함.



Sweep Config :

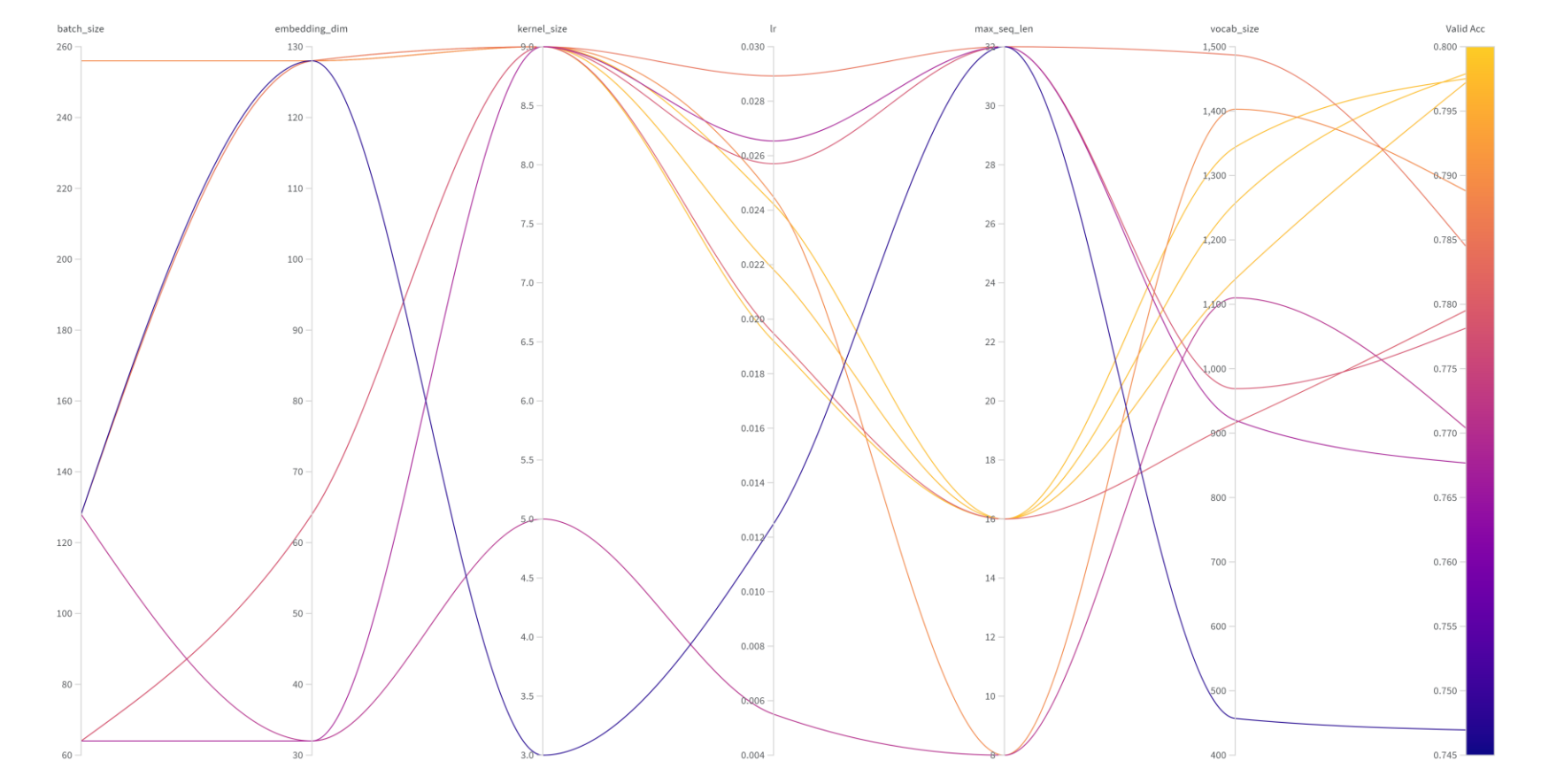
- Goal : Maximize Valid Acc
- batch size : 64 ~ 256
- embedding dim : 32 ~ 128
- Learning rate : 0.00001 ~ **0.05**
- **max sequence length : 4 ~ 16**
- **vocab size : 200 ~ 1500**

Result :

- max seq len : 8 이상이 적절함.
- vocab size :
  모델 성능에 큰 영향 없음

실험 결과 (1d CNN_3)

Hyperparameter setting에 민감함. 다양한 Hyperparameter 사용하여 실험함.



Sweep Config :

- **Goal : Maximize Valid Acc**
- **batch size : 64 ~ 256**
- **embedding dim : 32 ~ 128**
- **Learning rate : 0.00001 ~ 0.05**
- **max sequence length : 4 ~ 16**
- **vocab size : 200 ~ 1500**
- **kernel size : 3 ~ 9**

Result :

- **kernel size :**
  max seq len에 비해 너무 작으면
  모델 성능에 악영향
- **vocab size :**
  너무 작으면 성능에 악영향

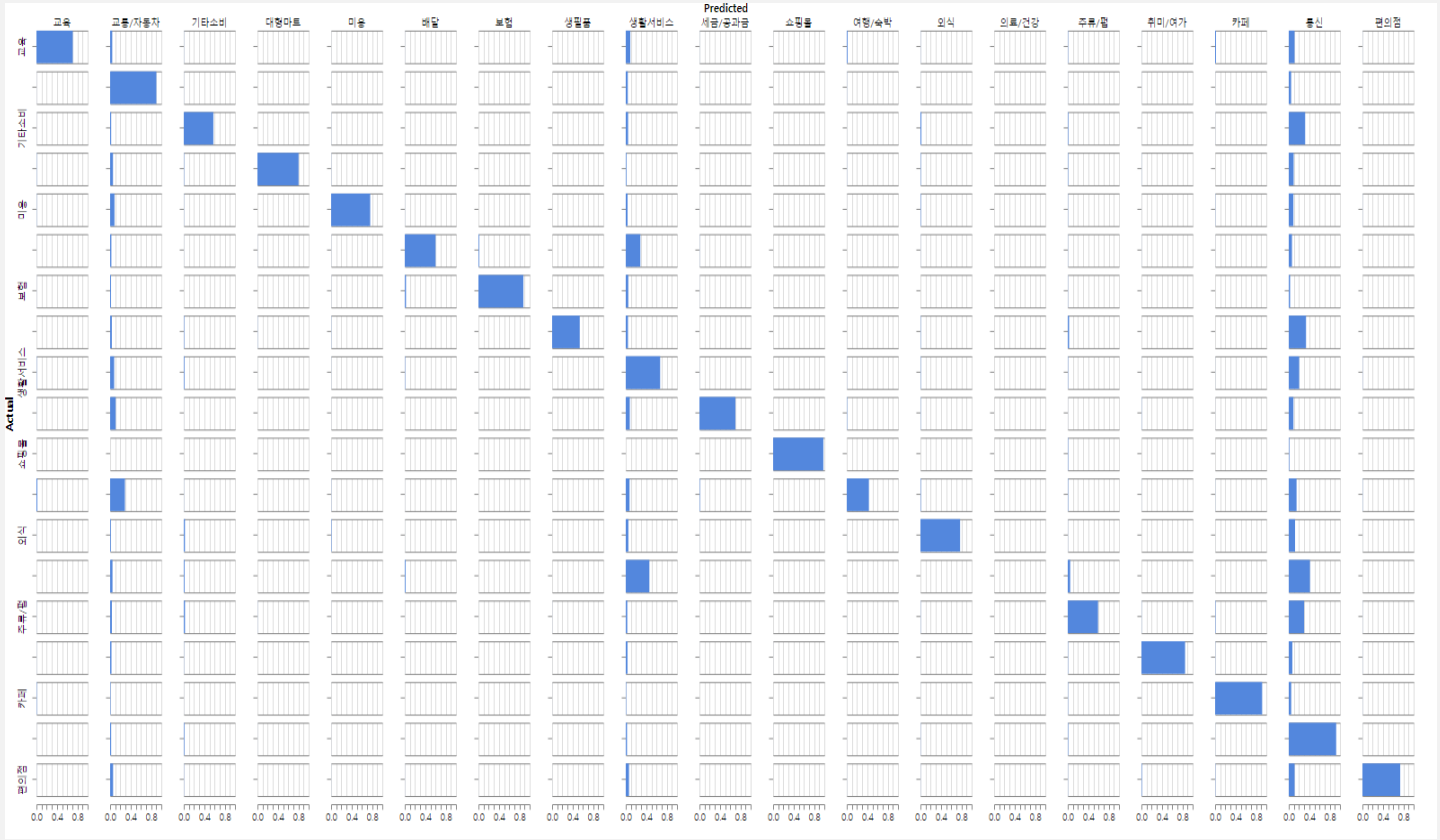
실험 결과 (1d CNN_4)

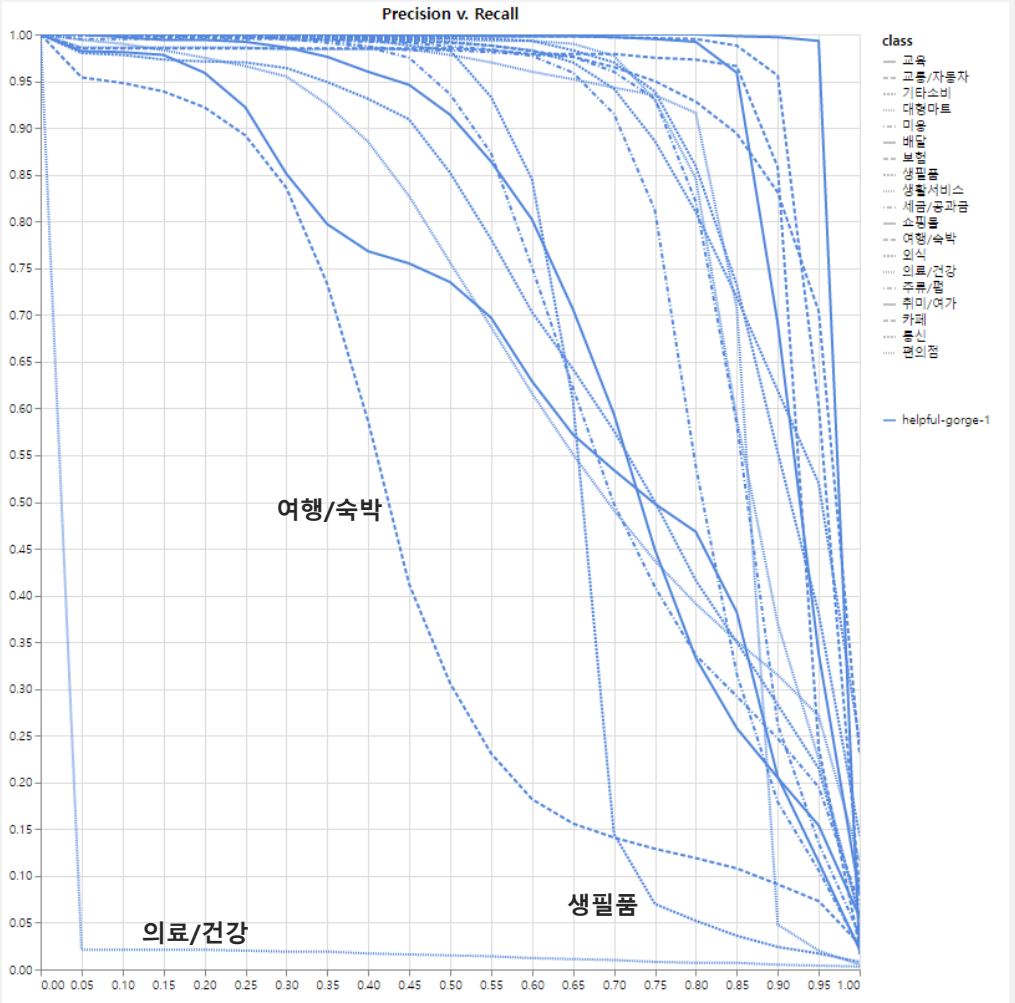
Pooling method : AvgPool vs MaxPool → hyperparameter setting이 더 중요함

실험 결과 (1d CNN)

Best Model : learning rate – 0.025 / batch size – 256 / embedding dim - 128
max seq len – 16 / kernel size – 9 / vocab size – 1300 / pooling - AvgPool

모델	Accuracy	F1	Loss
1d CNN	0.7935	0.7560	0.7017





실험 결과 (Pretrained Model)

시퀀스 최대 길이(Max Seq Len)와 손실함수(Loss Function)를 바꾼 실험 결과 (uncased dataset)

Model	Max Seq Len	Loss Function	Loss	Accuracy	Macro F1	Micro F1	Weighted F1
KoBERT	26	**CrossEntropy**	**0.4423**	**0.8621**	**0.8463**	**0.8621**	**0.8623**
		FocalLoss	0.4887	0.8609	0.8445	0.8609	0.8608
		WeightedCE	0.5180	0.8382	0.7904	0.8382	0.8438
	32	CrossEntropy	0.4476	0.8614	0.8455	0.8614	0.8623
		FocalLoss	0.5026	0.8593	0.8405	0.8593	0.8602
KoELECTRA	26	**CrossEntropy**	**0.4574**	**0.8597**	**0.8438**	**0.8597**	**0.8605**
		FocalLoss	0.4978	0.8593	0.8440	0.8593	0.8596
		WeightedCE	0.4961	0.8585	0.8409	0.8585	0.8588
	32	CrossEntropy	0.4649	0.8584	0.8378	0.8584	0.8588
		FocalLoss	0.5007	0.8569	0.8375	0.8569	0.8569
RoBERTa	26	CrossEntropy	0.4738	0.8532	0.8169	0.8532	0.8532
		FocalLoss	0.5222	0.8526	0.8194	0.8526	0.8537
		WeightedCE	0.6165	0.8143	0.7333	0.8143	0.8269
	32	**CrossEntropy**	**0.4753**	**0.8535**	**0.8228**	**0.8535**	**0.8557**
		FocalLoss	0.5218	0.8509	0.8181	0.8509	0.8529

공통 config :

- **data: uncased**
- **Batch size = 256**
- **Epoch = 5**
- **Dropout rate = 0.5**
- **Learning rate = 5e-5**
- **Max grad norm = 1**
- **Warm up ratio = 0.1**

Result :

적은 수의 class에 더 많은 가중치를 주는

***Weighted CE는 성능 향상에 효과가 없었음**

*Weight = 업종 별 비율의 역수

- 성능 측정은 가장 낮은 validation loss를 기록한 checkpoint model 기준으로 함.

실험 결과 (Pretrained Model)

데이터셋을 바꾼 실험 결과

모델은 KoBERT, 파라미터는 Max Seq Len = 26, Loss Fundtion = CrossEntropy로 고정

모두 소문자로 변환(Uncased), 대문자 그대로(Cased), 숫자 제거(NoNumber)한 데이터셋으로 실험

Model	data	Accuracy	Macro F1	Micro F1	Weigthed F1	Loss
KoBERT	**Uncased**	**0.8621**	**0.8463**	**0.8621**	**0.8623**	**0.4423**
	Cased	0.8559	0.8432	0.8599	0.8602	0.4457
	NoNumber	0.8600	0.8415	0.8600	0.8602	0.4522

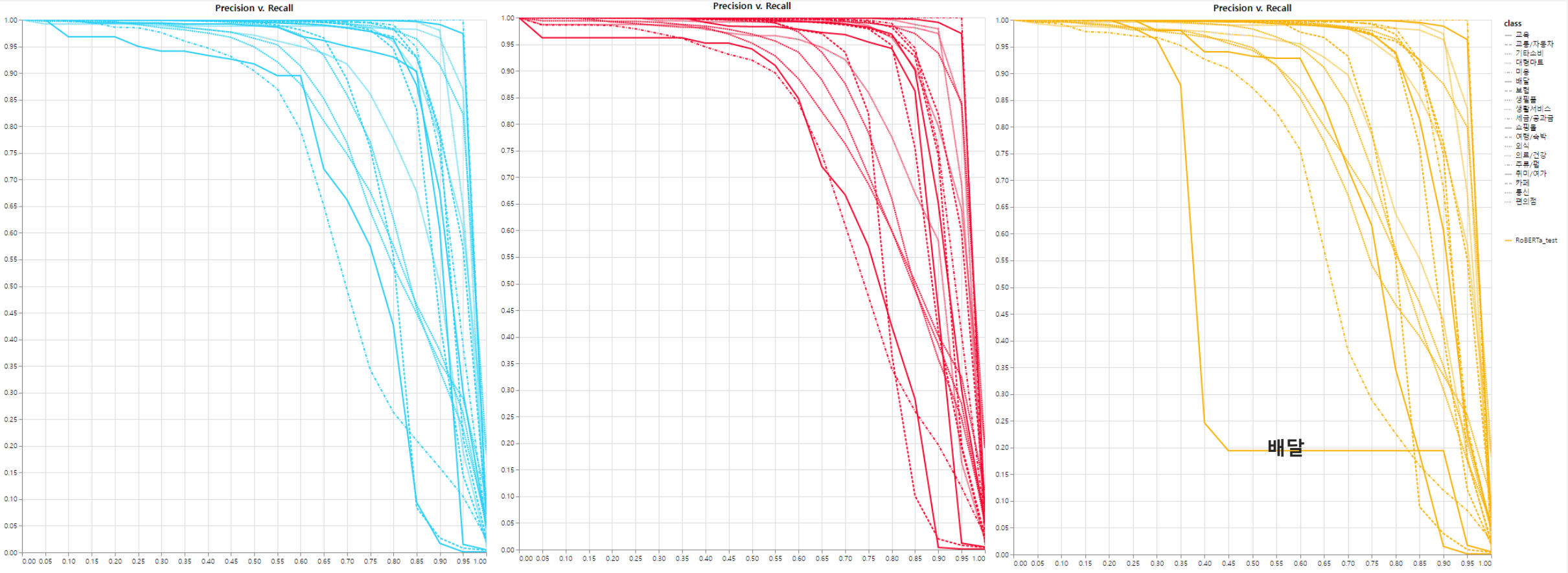
Result :

숫자는 그대로 두고 영어를 모두 소문자로 변환한 데이터셋이 가장 성능이 좋았음.

KoBERT-1 KoELECTA-1 RoBERTa-4 Precision-Recall Curve

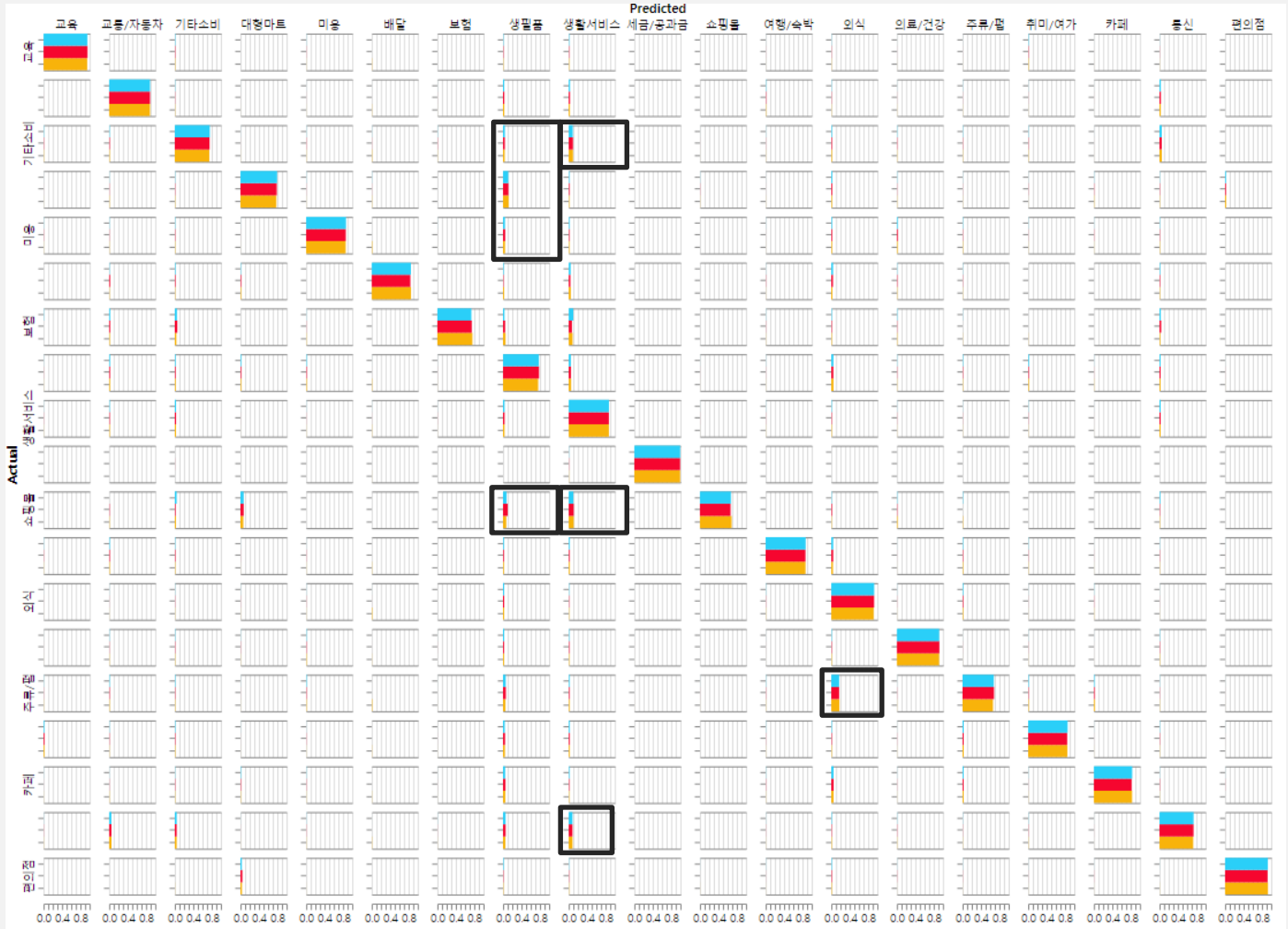
(26, CrossEntropy) (26, CrossEntropy) (32, FocalLoss)

KoBERT와 KoELECTRA는 모든 클래스에 대해 robust하게 분류했으나, RoBERTa의 경우 배달 class의 F1 score가 유의미하게 낮음



KoBERT-1 KoELECTA-1 RoBERTa-4 Confusion Matrix (normalized)

(26, CrossEntropy) (26, CrossEntropy) (32, FocalLoss)



Check!!

- Actual: 주류/펍
  Predicted: 외식
- Actual: 기타소비, 대형마트, 미용, 쇼핑몰
  Predicted: 생필품
- Actual: 기타소비, 쇼핑몰, 통신
  Predicted: 생활서비스
- Actual: 쇼핑몰, 편의점
  Predicted: 대형마트

→ 주류/펍, 외식의 유사성
→ 생필품, 생활서비스의 모호한 labeling 기준
→ 생필품, 생활서비스의 절대적인 data 수
→ 쇼핑몰, 대형마트, 편의점의 유사성

오분류 Case

Model : KoBERT / Max Seq Len = 26 / Loss Function = CrossEntropyLoss / Data = Uncased dataset

Case 1. 사람이 봐도 업종을 구분하기 힘든 업체명

업체명	업종	업종_prediction
이씨스	기타소비	통신
홍종국	생활서비스	기타소비
선부	미용	외식
인터라켄	카페	생활서비스

Case 2. 최초 Labeling 오류

업체명	업종	업종_prediction
GS양구정림점	생필품	편의점
스마트관광㈜	교통/자동차	여행/숙박
비치타운리조텔	생활서비스	여행/숙박
우리생협메트로시티점	의료/건강	대형마트

Case 3. 업체명에 특정 키워드가 들어있는 경우

업체명	업종	업종_prediction
카페뮤토	주류/펍	카페
참숯굴건강랜드	생활서비스	의료/건강
한국앵무새학교	여행/숙박	교육
거기마트	편의점	대형마트

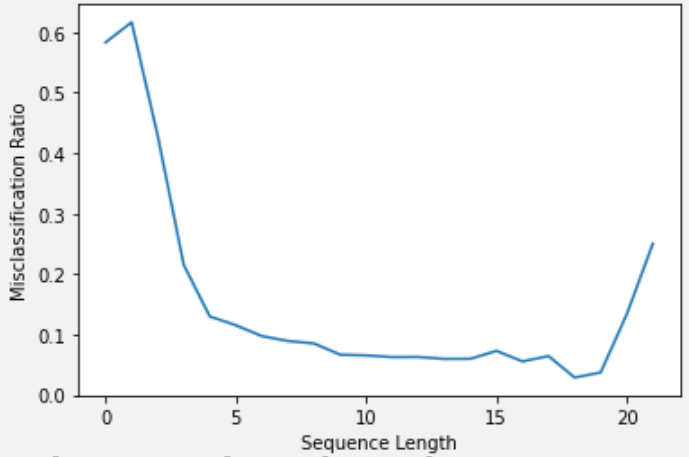
오분류 Case

Model : KoBERT / Max Seq Len = 26 / Loss Function = CrossEntropyLoss / Data = Uncased dataset

- 길이가 너무 짧거나 긴 업체명에 대한 오분류율 높음.

  길이가 너무 짧으면 주요 token을 찾지 못하거나

  길이가 너무 길면 너무 많은 주요 token으로 인해 오분류 발생



- 의미가 모호한 Labeling 간의 오분류율이 높음 (ex. 생활서비스-생필품, 기타소비 / 외식 – 주류/펍)