
WYF
What's your feeling?
당신의 감정을 아는 음악 추천 서비스
WYF

목차

WYF

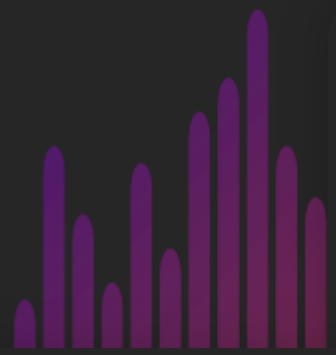
1. 기획의도

노래를 듣는 이유는 무엇인가요?
잠이 오지 않아서?
슬픈 일이 있어서?
신나는 파티를 즐기고 있어서?

음악은 모든 순간 우리와 함께합니다.
그럴 때마다 우리는 여러분의 상황과 감정에 맞게
노래를 추천할 것입니다.

01. Recommend System

- 대중적 인기도를 반영한 추천이라 효율적
- 비슷한 느낌의 노래만 추천하는 필터버블 현상 발생

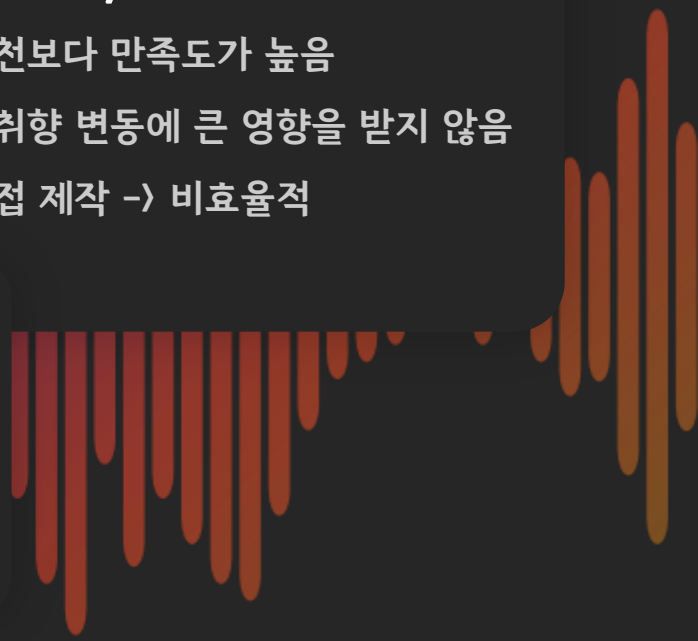


02. YouTube Playlist

- 기계의 추천보다 만족도가 높음
- 사용자의 취향 변동에 큰 영향을 받지 않음
- 사람이 직접 제작 -> 비효율적

03. What we made....

- 필터 버블 문제에서 비교적 자유로움
- 사용자의 현재 감정을 인식
- 노래 자체의 대중적 인기도에 영향받지 않는 추천

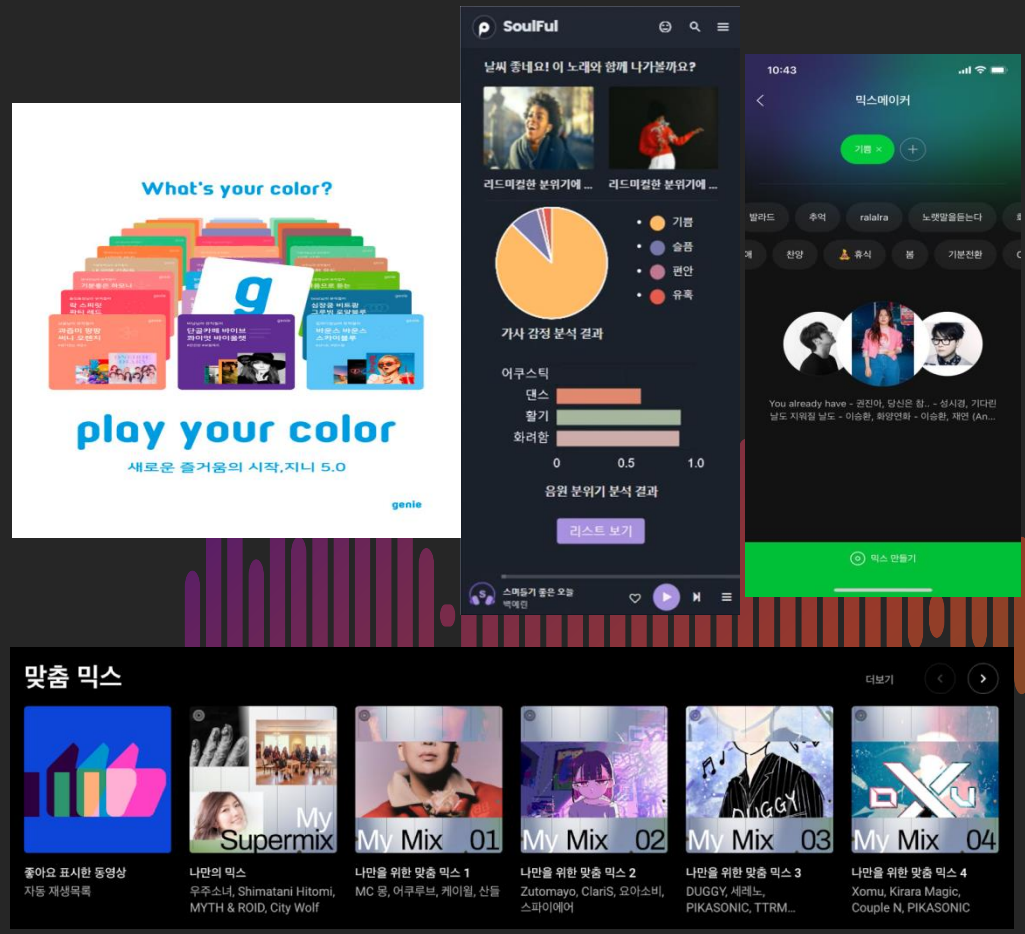


WYF

2. 타 서비스 현황

컬러나 날씨 등에 음악 추천 서비스를 접목하는 케이스 多

→ 사용자의 **현재 감성을 정확하게 알기 위한** 시도가 증가



WYF

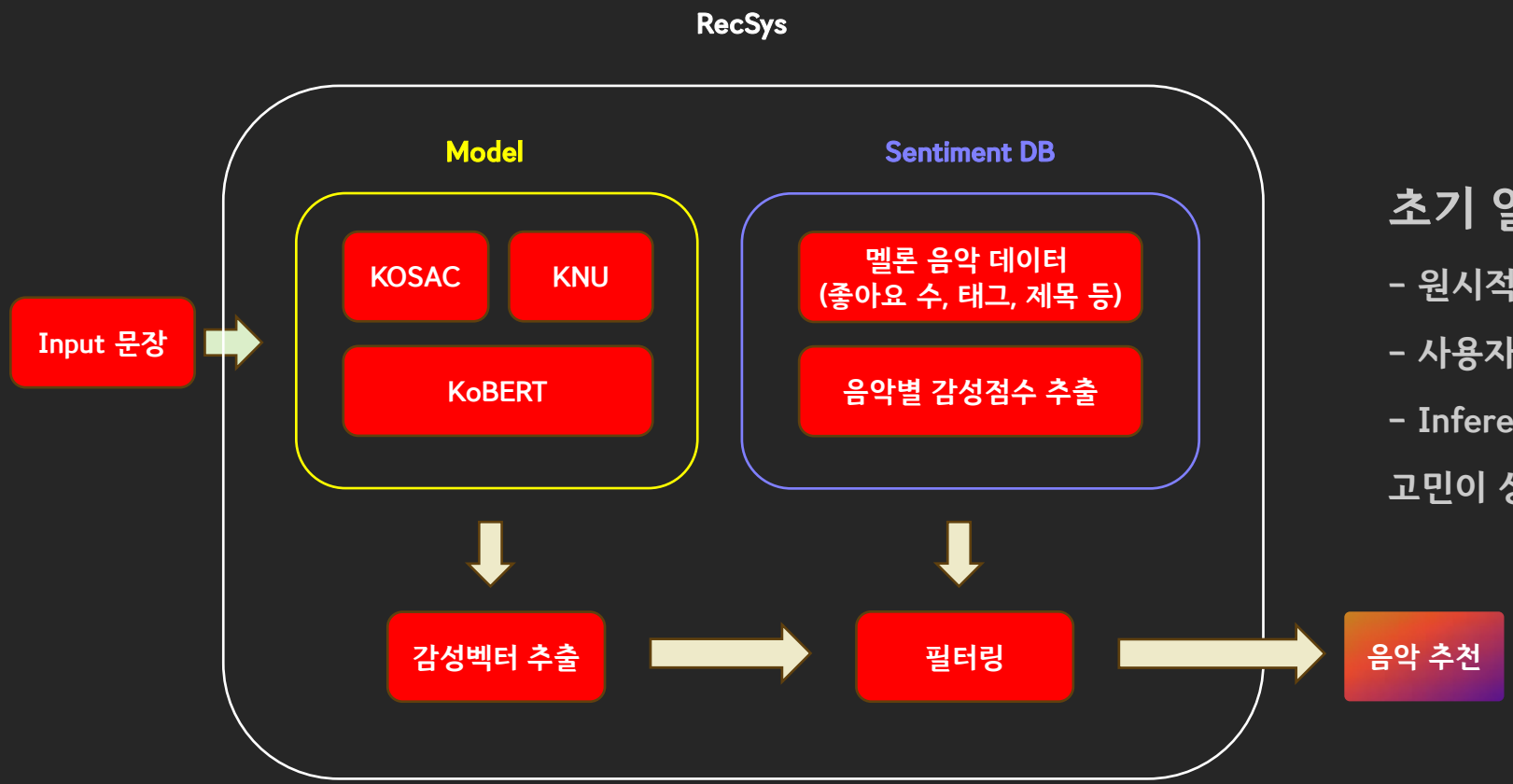
3. 기획내용

- 사용자는 자신의 기분을 자유롭게 입력 가능
- 입력받은 문장의 감정을 다각도로 분석
(Mel Spectrogram, NLP)
- 사용자의 현재 감정과 유사한 감성을 가진 노래를 추천

〈시연 영상〉

WYF

3. 기획내용 (초기 설계)

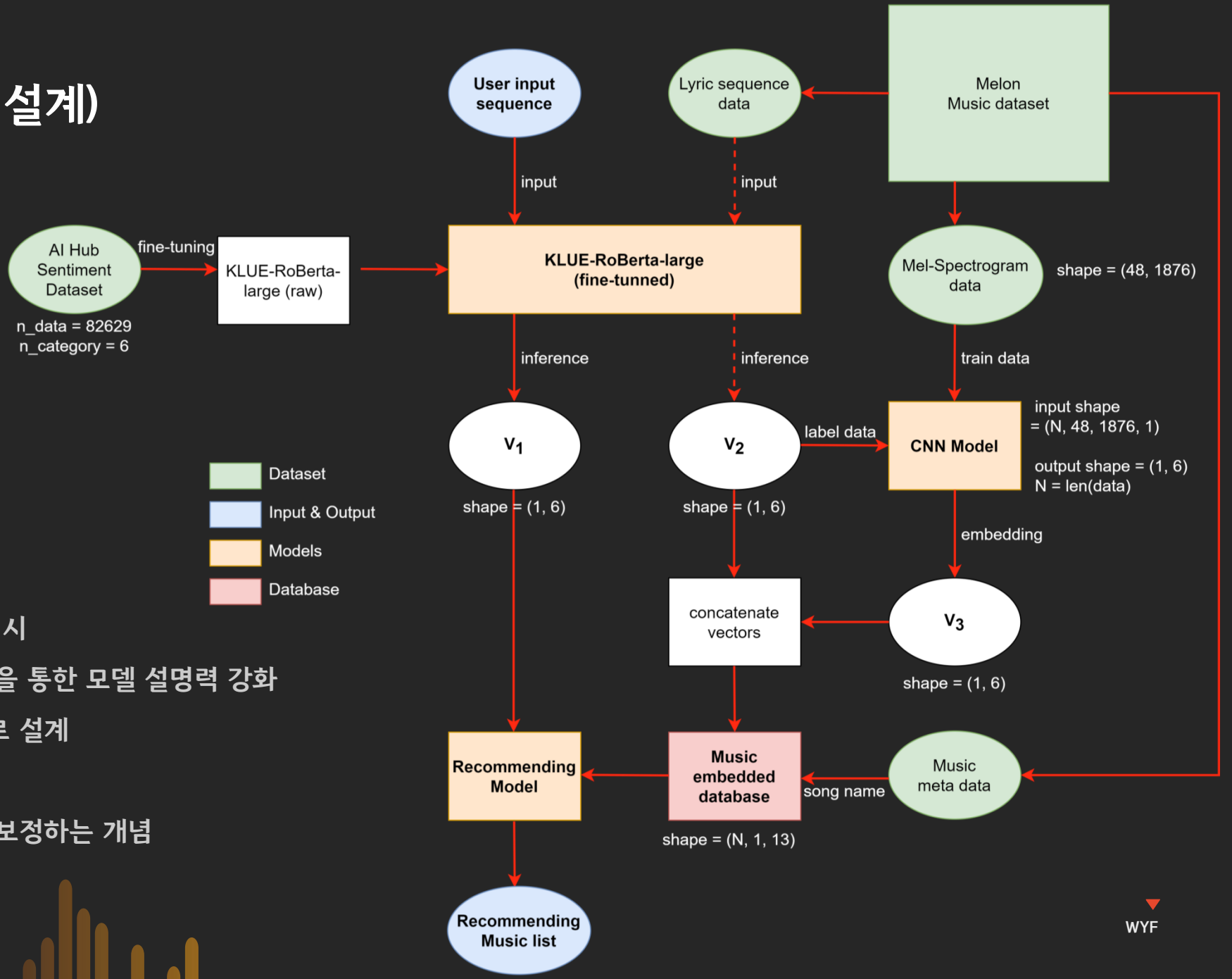


초기 알고리즘

- 원시적 구조
- 사용자의 입력 문장 분석을 위한 감성어 사전 사용
- Inference 속도, 모델 성능, 데이터 정제 등에 대한 고민이 상대적으로 부족

WYF

3. 기획내용 (최종 설계)



모델 아키텍처

- 전체 모델 구조 고도화 및 입출력 명시
- 감성어 사전 대신 추가 데이터 학습을 통한 모델 설명력 강화
- 데이터 구조에 대한 이해를 기반으로 설계
- 데이터베이스 기능 추가
- 가사 분석 결과를 Mel 분석 결과로 보정하는 개념

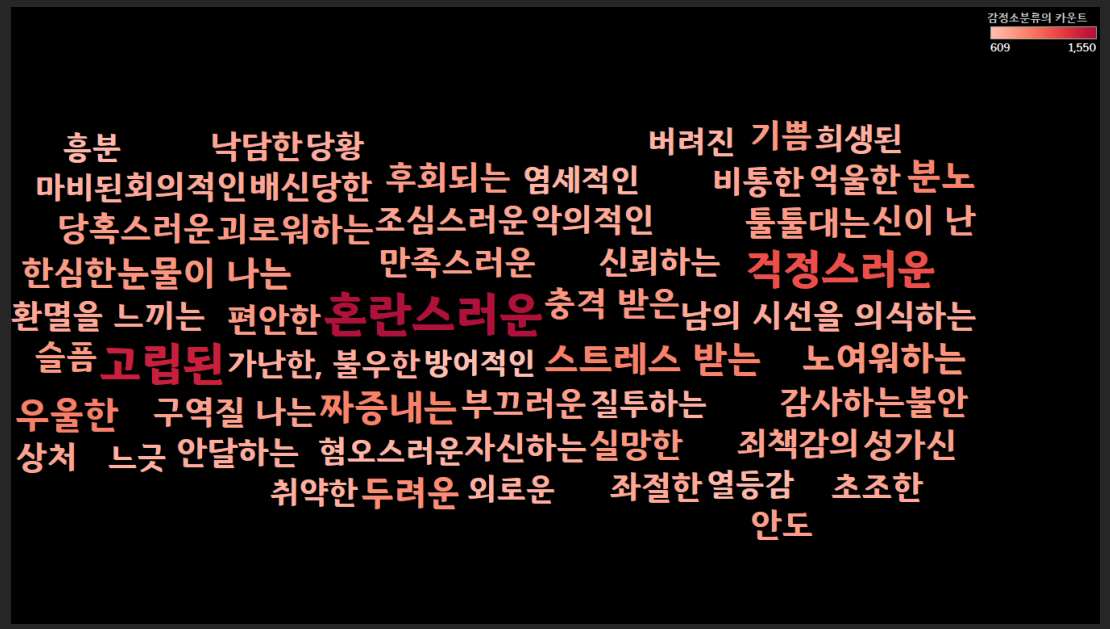
WYF

4. 데이터 (NLP)

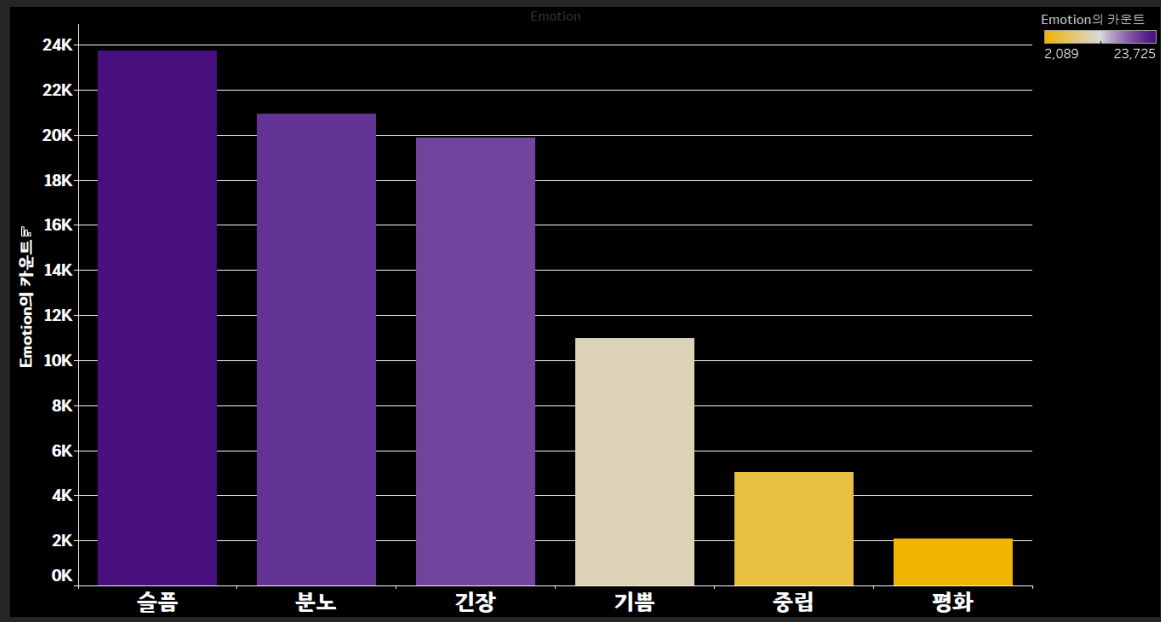

데이터 수집
84,603건
1차 정제
감정 소분류 기준
2차 정제
핵심 단어 기준
슬픔
분노
긴장
기쁨
중립
평화
감성 대화 말뭉치
46,009건, label=6
감정 대분류 / 소분류 존재
단발성 대화 데이터셋
38,594건, label=7
라벨링 불일치 다수 발견
노이즈 최소화 위한 정제
핵심 감정 단어 기준
6가지 감정 Re-labeling

4. 데이터 (NLP)

감정 소분류 Word Cloud



최종 감정 데이터 분포



WYF

4. 데이터 (Mel & Lyric)

kakaoarena

가사 데이터

크롤링
메타 데이터 기반
707,988건

필터링
가사 없는 장르 등

데이터 1차 정제
Bridge, Hook 등
불필요 단어 제거

데이터 2차 정제
Token Length 512개

가사 데이터
112,566건

Mel Spectrogram

데이터 구조
원본 음원의
20-50초

데이터 정제
Shape Unmatch,
Bias 보정 등

Mel 데이터
112,566건

WYF

5. 라이브러리 & 프레임워크

Hugging Face

Model Version Managing

Google Cloud Platform

Resource Supplement

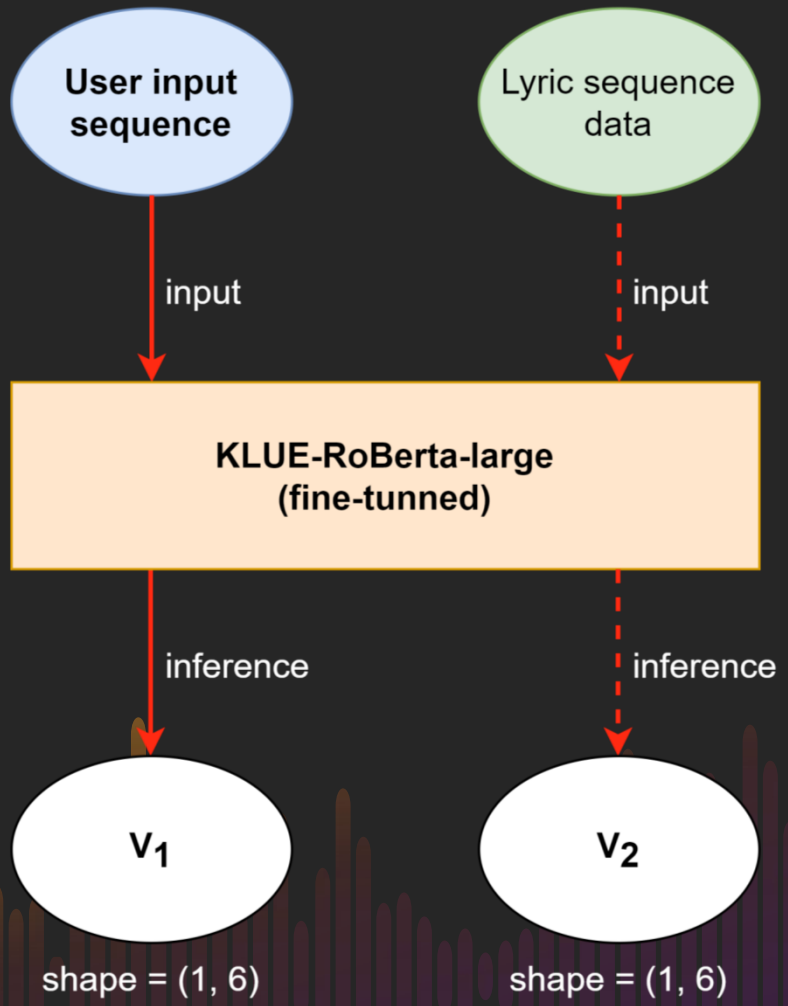
Weight & Biases

Monitoring & Visualize

Streamlit

Model Serving

6. 개발이력 (NLP)



개발 과정

1. 모델 선정
2. 데이터 사용 방향에 대한 고민
3. DAPT 방법론 채택
4. 파라미터 최적화 및 성능 개선

WYF

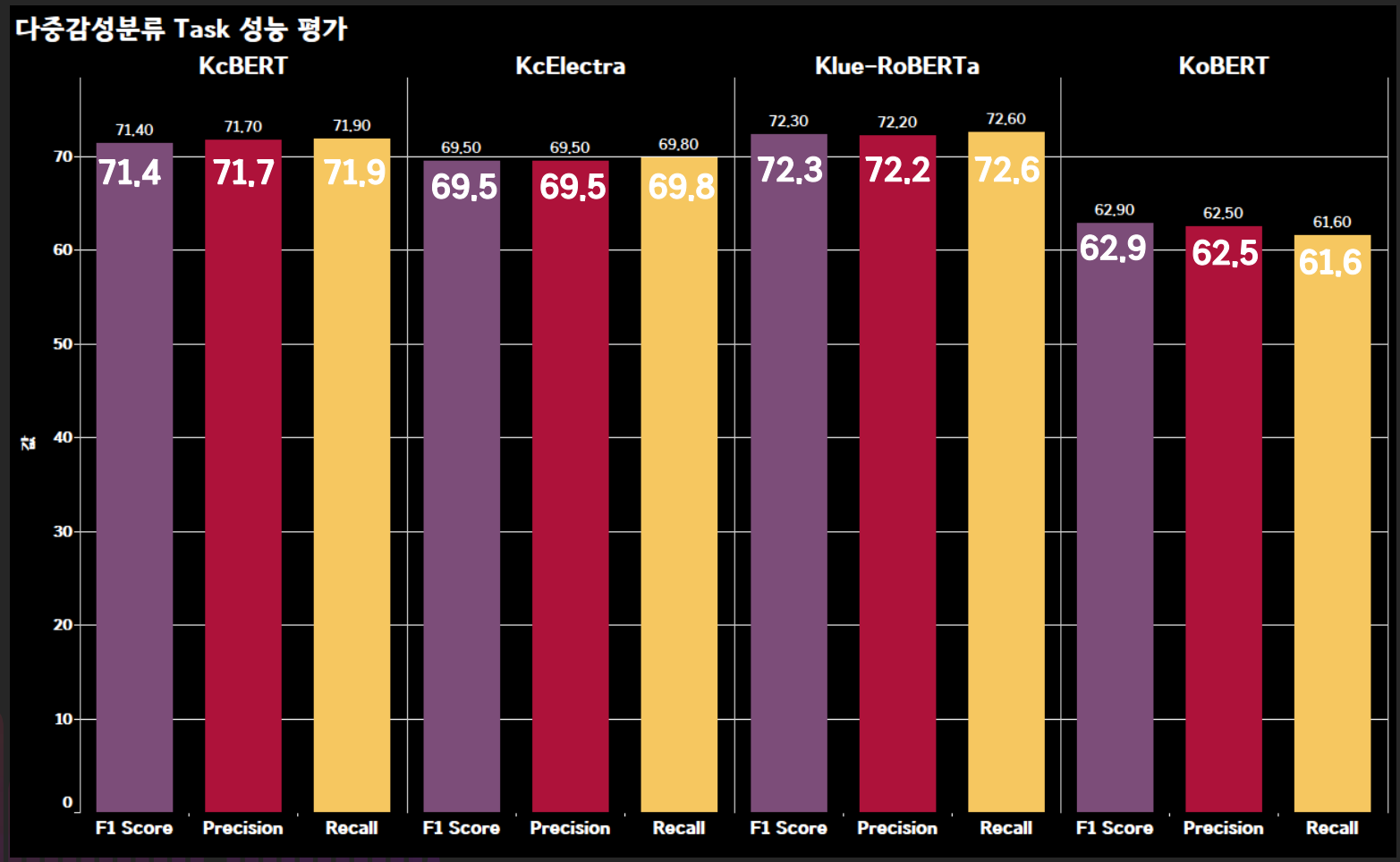
6. 개발이력 (NLP)

1. 사용한 모델 선정

- KcBERT, KcElectra, Klue-RoBERTa, KoBERT
- 감성어 데이터셋 다중분류 Task 평가
- 최종 선정 : Klue-RoBERTa-large

Why Klue-RoBERTa-large ?

- 가장 많은 사전학습 데이터
- NSP가 아닌 SOP 방식의 Pretrain
- Dynamic Masking (동일 데이터 대비 고성능)



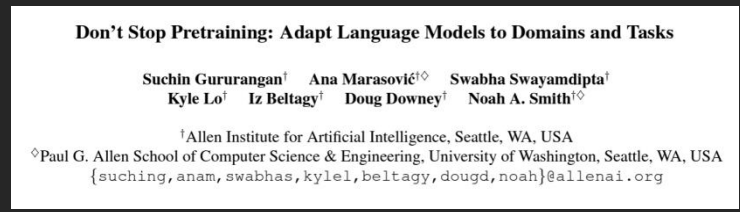
WYF

6. 개발이력 (NLP)

2. 적합한 데이터 사용 방향과 성능 향상에 대한 고민

- "데이터는 많이 모았는데 이걸 어떻게 사용해야 성능이 향상될 것인가?"

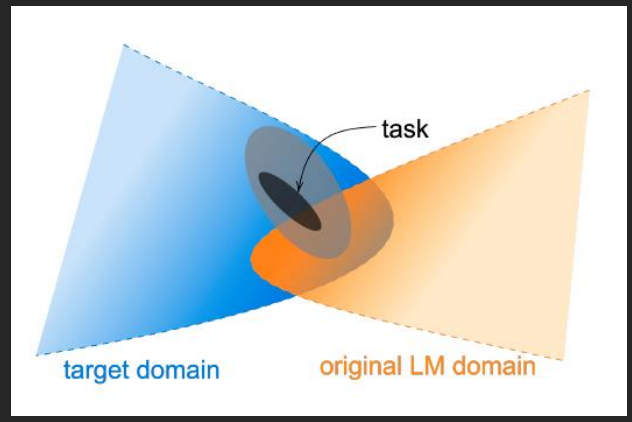
- DAPT (Domain Adaptive PreTraining) 방법론 채택
- Pretrain Data와 Downstream Task Data의 분포 차이로 인한 성능 저하 문제
- 해결책으로 Downstream Task Data를 사용하여 추가 Pretrain 작업할 것을 제안
- 영어를 아는 아이에게 프랑스어 읽는 법을 가르쳐주고 프랑스어 메뉴를 읽게 하는 것과 비슷


Don't Stop Pretraining: Adapt Language Models to Domains and Tasks

Suchin Gururangan† Ana Marasović†◇ Swabha Swayamdipta†
Kyle Lo† Iz Beltagy† Doug Downey† Noah A. Smith†◇

†Allen Institute for Artificial Intelligence, Seattle, WA, USA
◇Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, WA, USA
{suching,anam,swabhas,kylel,beltagy,dougd,noah}@allenai.org

〈Don't Stop Pretraining〉 (2020)



〈사전학습 데이터와 Task 데이터의 도메인 분포 차이〉

6. 개발이력 (NLP)

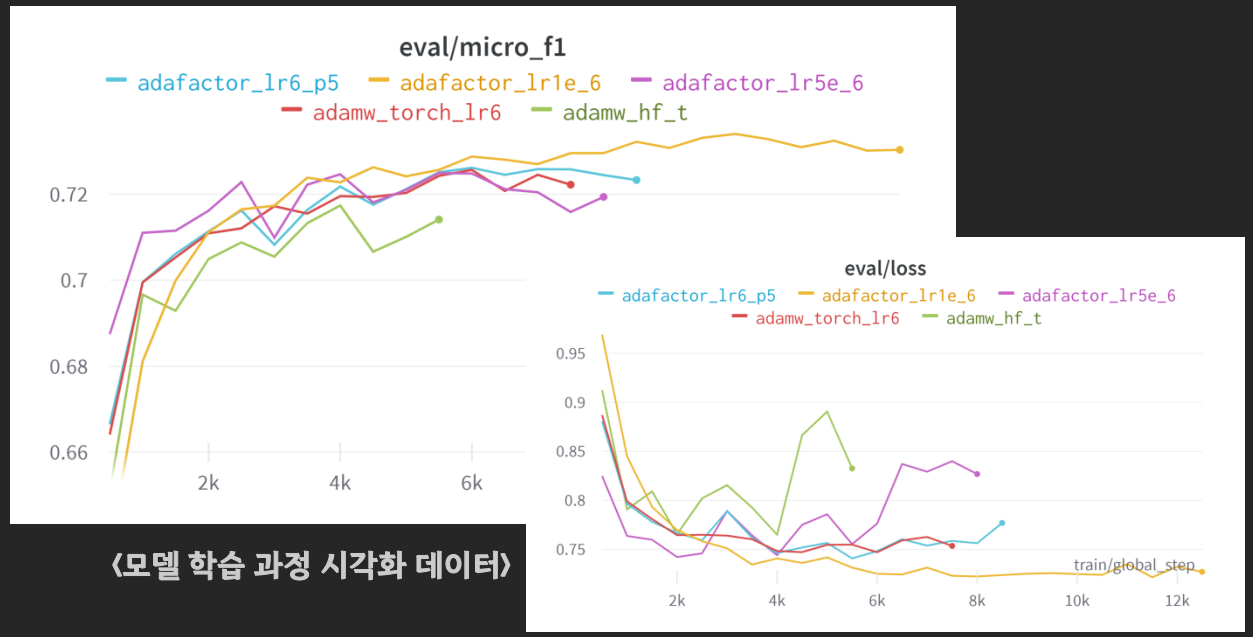
3. 어떻게 하면 효율적인 Pretrain이 가능할까?

- 논문에서는 도메인 크기 순서대로 학습했을 때 BEST 성능

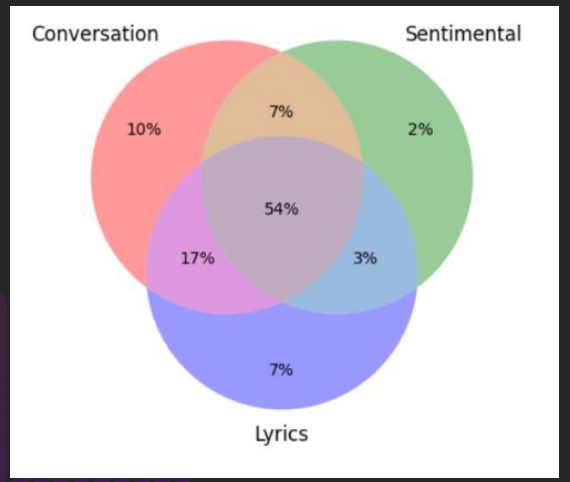
- 보유 데이터에 대한 EDA 진행
- 가사, 감성어, 일상어 데이터셋 내 감성 단어 분포 분석
- 주요 감성 단어가 상당수 중복되고 있음을 확인
- 데이터 분포 확인 후 Pretrain 및 Fine-tuning 진행

4. 파라미터 최적화 및 성능 개선

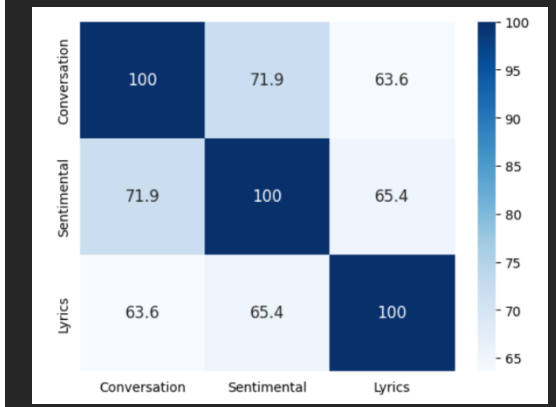
- Optimizer, Learning rate, Early-stop 등에 따른 성능 변화 관찰/평가
- Adafactor +Learning rate 1e-6 파라미터 사용
- 최종 성능 : Micro F1 Score 0.734



〈모델 학습 과정 시각화 데이터〉



〈데이터별 감성 단어 분포 1〉

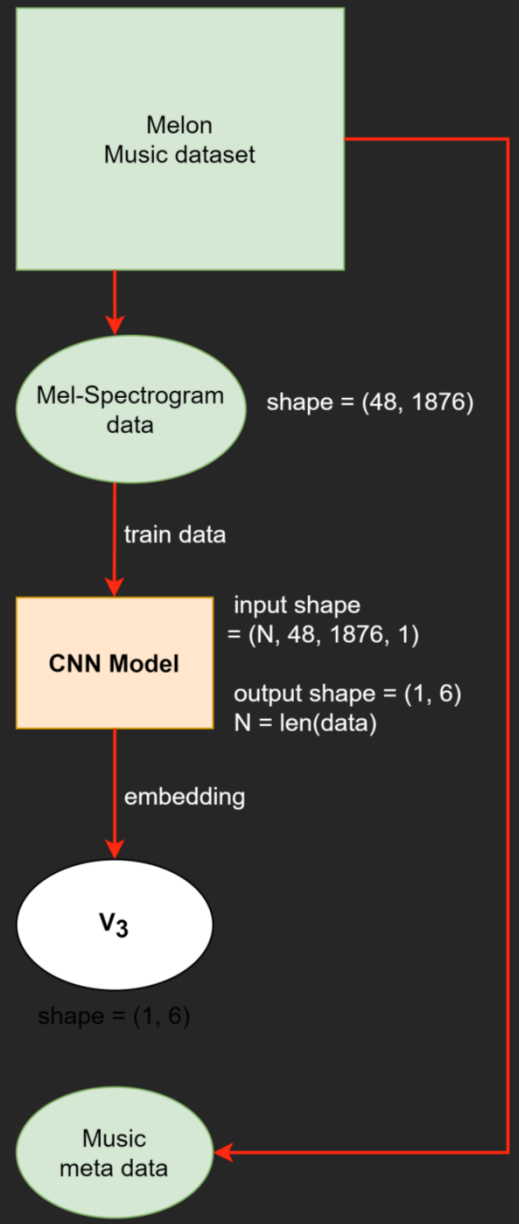


〈데이터별 감성단어 분포 2〉

WYF

6. 개발이력 (Mel)

1. 실험 설계 및 평가 지표 제작
2. 데이터 EDA, 실험 방향 설정
3. 1차 시도 및 문제 발생
4. 문제에 대한 가설 수립
5. 2차 시도



WYF

6. 개발이력 (Mel)

1. 실험 설계 및 평가 지표 제작

- Mel Data 분석 결과와 가사 분석 결과 간 데이터의 감정 분포와 다양성을 볼 수 있는 평가 지표가 필요
- 4개의 평가 지표를 새로 제작
- Top1 Acc.를 중심으로 평가하되 나머지 지표를 참고하여 다양성 판단

Top2 Acc

순서 상관없이 둘 다 맞추는 경우

Top3 Acc

순서 상관없이 셋 다 맞추는 경우

Less-strict Top1 Acc

둘 중 하나 이상 맞추는 경우

Less-strict Top2 Acc

순서 상관없이 셋중 둘 이상 맞추는 경우

(e.g.)

Top1 Acc. + Top2 Acc. => 메인 감정은 동일하지만 보조 감정이 다른 경우

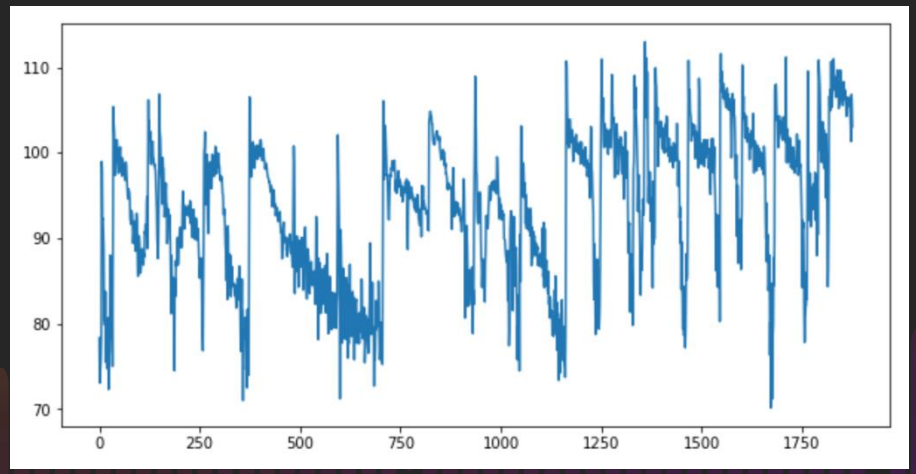
Top1 Acc. + Less-strict Top2 Acc. => 주 감정과 유사한 감정 종류 확인 가능

WYF

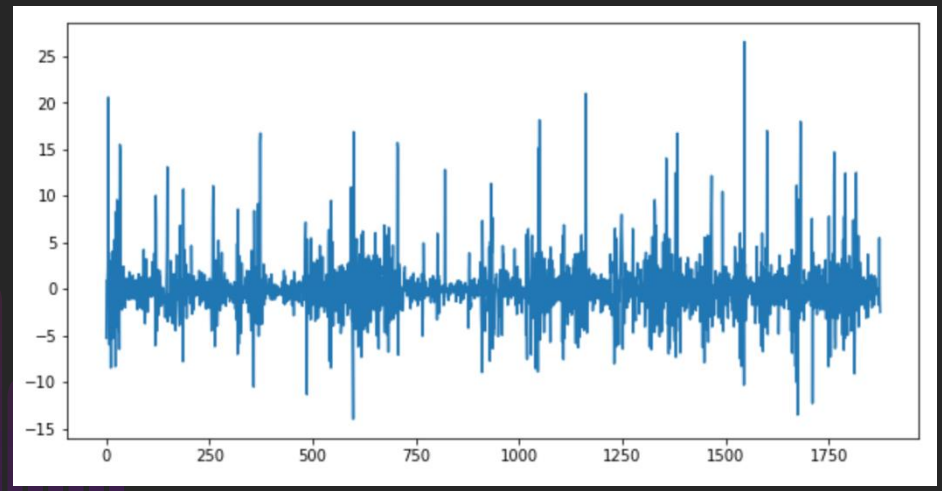
6. 개발이력 (Mel)

2. 데이터 EDA 및 실험 방향 설정

- Mel Spectrogram은 노래별 20~50초 구간 db 값의 분포 데이터
- 고속 푸리에 변환 등의 과정을 거쳐 wav 파일을 Numpy Array로 변환
- 사람이 알아보기 쉬운 형태로 바꾼 데이터 -> 컴퓨터도 알아보기 쉬울까?
- Mel Data는 Time-series적인 측면도 존재
- 다수의 논문에서 CNN 기반 모델을 사용
- 타 팀에서 유사한 프로젝트 진행 -> 상호 협력 및 연계



〈1개 Hz 데이터 시각화〉



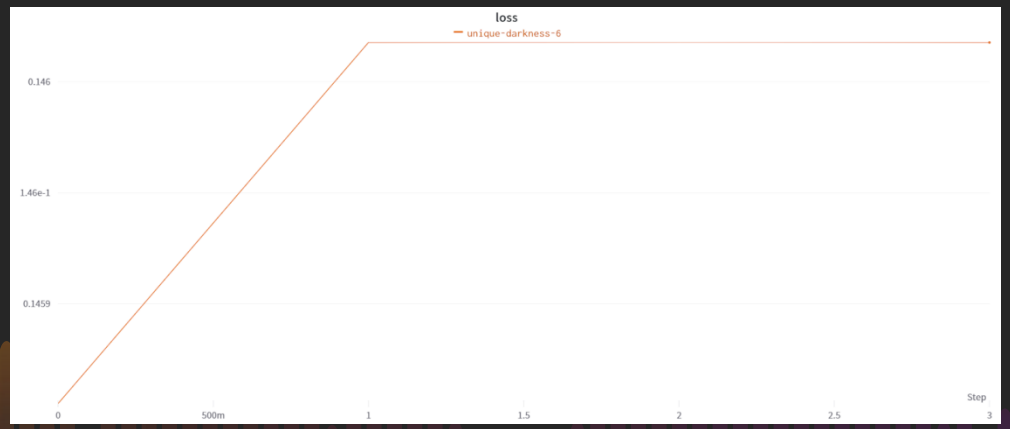
〈보기 쉽게 파형 형태로 변환했을 때〉

WYF

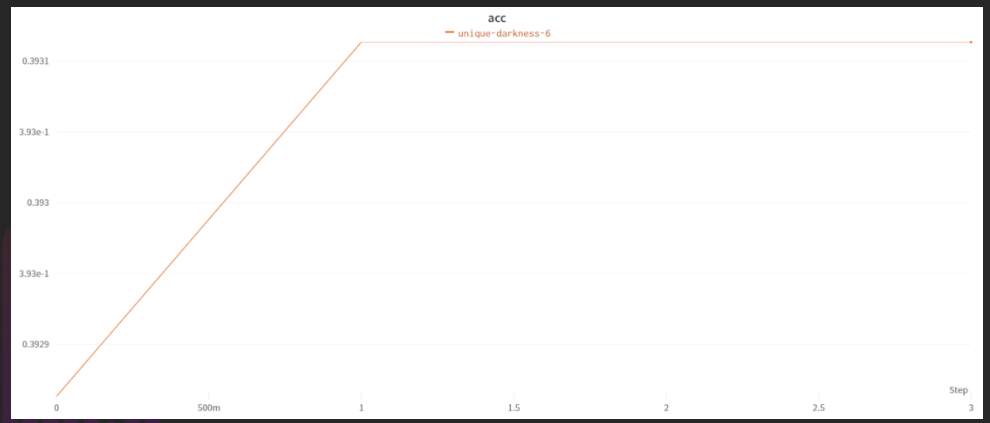
6. 개발이력 (Mel)

3. 1차 실험

- CNN, ResNet50 기반 2개 모델 실험
- 하이퍼파라미터, 모델 아키텍처에 상관 없이 학습이 제대로 되지 않는 문제 발생
- Receptive field 크기 문제도 아니었음
- 모델의 문제가 아니었으므로 데이터를 다시 살펴볼 필요가 있다고 판단



⟨Loss⟩



⟨Accuracy⟩

WYF

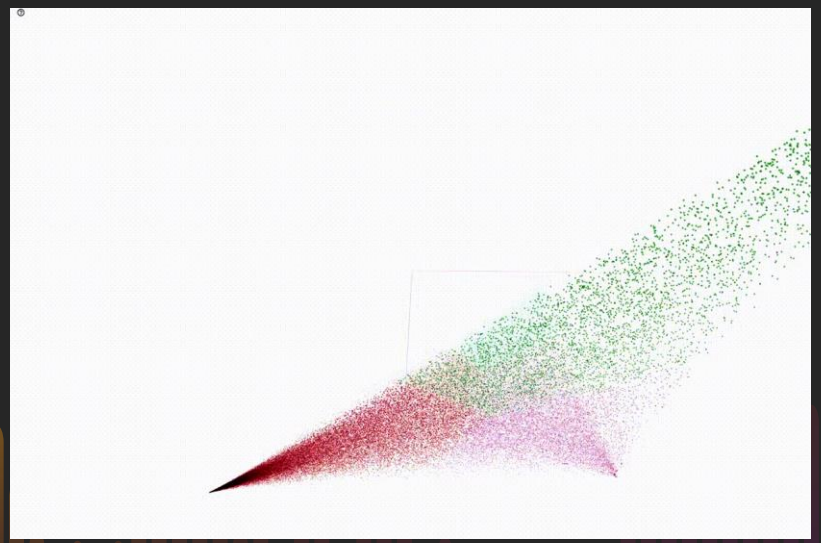
6. 개발이력 (Mel)

4. 문제에 대한 가설 수립

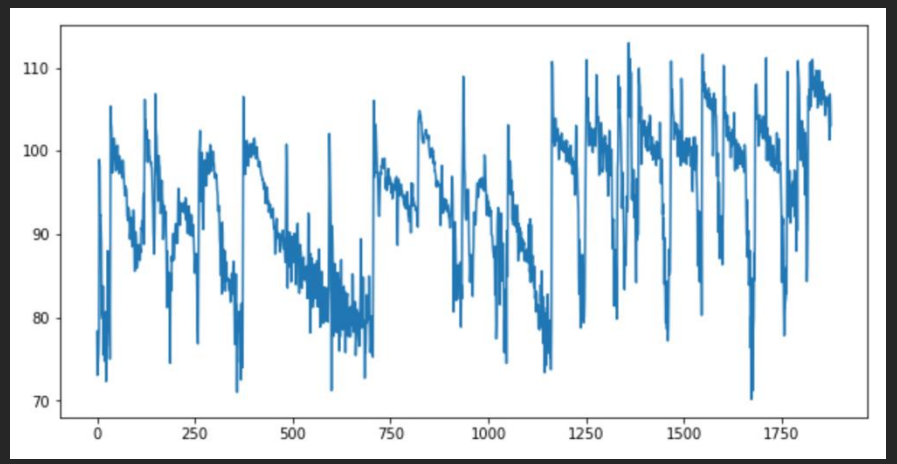
(1) Mel 데이터 자체의 Noise 가능성

(2) 라벨 자체가 잘못 추출되었을 가능성

(3) Mel 데이터의 피처 추출이 충분하지 않을 가능성



〈가사 Embedding 시각화 데이터〉



사람이 알아보기 쉬운 데이터 ≠ 컴퓨터가 이해할 수 있는 데이터

Raw Data를 사용하기보다 추가적인 Feature Extracting 작업이 필요하다고 판단

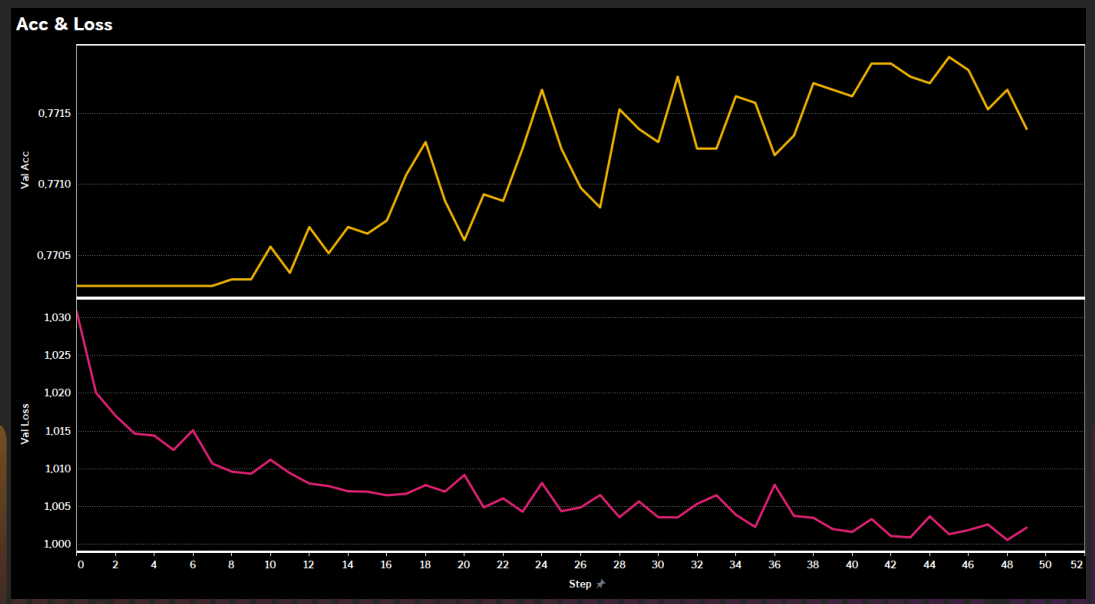
WYF

*SimCLR: 데이터의 고유한 특성을 증폭하여 추출하는 임베딩 방법

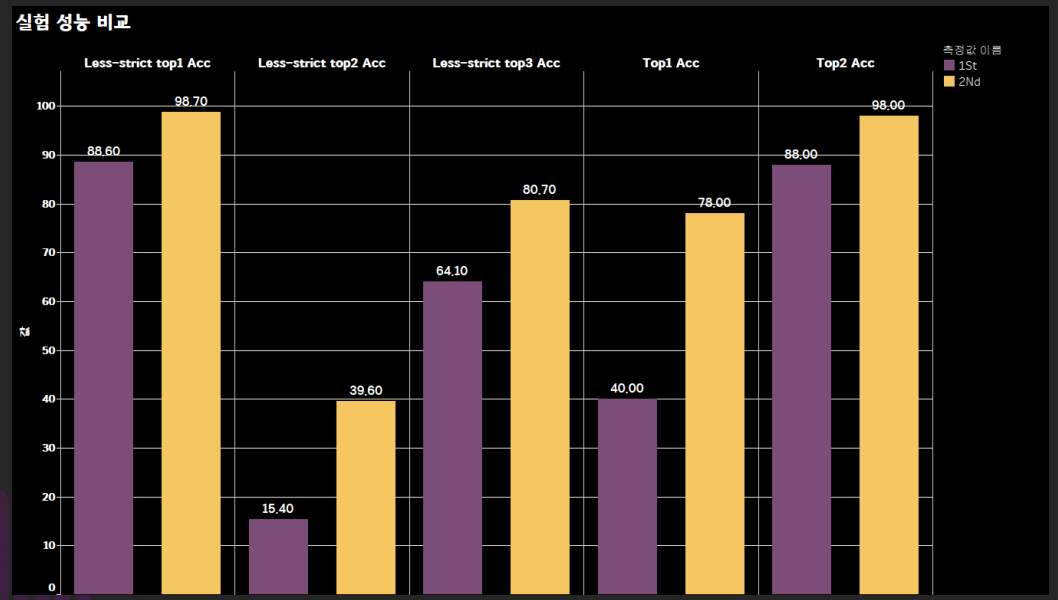
6. 개발이력 (Mel)

5. 2차 실험

- 뮤직임베딩 팀의 *SimCLR 임베딩 모델을 사용하여 추출한 임베딩 벡터를 학습 데이터로 사용
- 사전 임베딩 과정을 거쳤으므로 Deep Model을 사용하지 않음
- 전반적인 추론 성능 향상을 확인
- 최종 성능 : Top-1 Acc. : 40% → 78%, Top-2 Acc. : 88% → 98%



〈임베딩 이후 모델 학습 추이〉

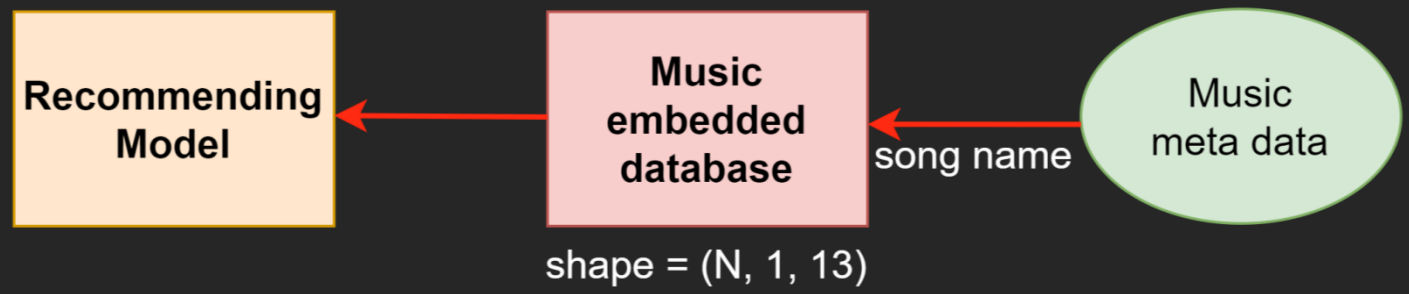


〈임베딩 전후 성능 비교〉

WYF

6. 개발이력 (Recsys)

1. 코사인 유사도 사용
2. 편향 데이터 제거
3. 유저 피드백 반영
4. 유사도 분석 소요시간 문제
5. 사용자 피드백 수집 문제



6. 개발이력 (Recsys)

1. 코사인 유사도 사용

- 사용자의 감정벡터(1,6)와 Music Database의 감정벡터(1,6)간 코사인 유사도 측정을 통해 추천리스트 작성

2. 편향 데이터 제거

- 중복 콘텐츠 제거

(e.g. 축하사절단의 9월 20일 축하노래, 9월 21일 축하 노래 etc..)

-아티스트가 명시되지 않는 (Various Artist)의 노래 삭제

-최종 : 42,839개

3. 유저 피드백 반영

- 발매일이 너무 광범위해서 트렌드에 맞지 않은 노래가 추천 되는 경우 존재

-2015년 이후 발매된 노래만 추천되는 대상으로 함

WYF

6. 개발이력 (Recsys)

4. 유사도 분석 소요시간 문제

- 기존 : 사이킷런 API 코사인 유사도 함수 사용
- 유사도 분석 소요 시간 : 20초 이상
- For Loop 사용 : O(N)

- 변경 후 : 자체 구현 함수 사용
- 유사도 분석 : 1초 내 완료
- 단일 함수 호출 : O(1)

5. 사용자 피드백 수집 문제

- 시간적 한계로 사용자의 피드백 데이터 수집 부족
- 향후 KDT 해커톤 진행 시 사용자의 피드백 데이터까지 포함시킬 계획

7. 추후 발전 방향

01. 추천시스템 고도화

- 사용자 피드백 수집 및 반영
- 협업 필터링 기능 추가

02. 음성인식 기능 추가

- 음성인식을 통한 입력 모듈 추가

03. Mel2Text, Text2Mel

멜로디를 주면 가사를 생성하거나, 가사를 주면 멜로디를 생성하는 모델 구현

04. KDT 해커톤 참여

- 경진대회 참여를 통한 지속 개발

WYF

8. 팀 소개

유제준

Team Leader

Mel Data Modeling

지정용

Crawling & Serving

NLP Data EDA

그림

전형진

Recsys

NLP Data Modeling

오준엽

NLP Data EDA & Preprocessing

Visualization

김도현

NLP Model Finetuning

NLP Data Modeling

WYF

모두의 연구소 강남 WYF팀

서울특별시 강남구 강남대로 324

역삼디오슈페리움 2F

https://github.com/What-s-Your-Feeling/main_project

https://wandb.ai/ethan_wyf

THANK YOU FOR WATHCING US

Appendix

〈트랜스포머 기반 Mel Spectrogram 다중감성분류 task 성능표〉

Test set	Sentiment		Emotions											
	2-class	7-class	Happy		Sad		Angry		Fear		Disgust		Surprise	
	A	*A*	*A*	F1	*A*	F1	*A*	F1	*A*	F1	*A*	F1	*A*	F1
L+ A + V	81.5	44.4	65.0	64.0	72.0	67.9	81.6	74.7	89.1	84.0	85.9	83.6	90.5	86.1
L + A	**82.4**	**45.5**	**66.0**	65.5	**73.9**	67.9	**81.9**	76.0	**89.2**	**87.2**	**86.5**	84.5	**90.6**	**86.1**
L	81.9	44.2	64.5	63.4	72.9	65.8	81.4	75.3	89.1	84.0	86.6	84.5	90.5	81.4
Mu-Net	82.1	-	-	**68.4**	-	**74.5**	-	**80.9**	-	87.0	-	**87.3**	-	80.9
G-MFN	76.9	45.0	-	66.3	-	66.9	-	72.8	-	89.9	-	76.6	-	85.5

A Transformer-based joint-encoding for Emotion Recognition and Sentiment Analysis(2020)

Appendix

TABLE 1. EMOTION COUNT OF DATASETS

Emotions	TESS	RAVDESS	Custom Dataset
ANGRY	400	344	170
SAD	400	344	170
HAPPY	400	344	170
NEUTRAL	400	172	165
DISGUST	400	344	170
SURPRISED	400	344	-
FEAR	400	344	170
CALM	-	344	-
Total	2800	2580	1015

〈논문 데이터셋 분포〉

TABLE 2. ACCURACY FOR BASELINE MODELS

	TESS	RAVDESS	Custom DS
KNN	77.34%	60.13%	59.36%
SVM	89.66%	76.41%	62.58%
MLP	96.76%	82.39%	72.57%
Decision Trees	94.56%	80.35%	65.71%

TABLE 3. ACCURACY FOR AGGREGATOR MODELS

	TESS	RAVDESS	Custom DS
Random Forest	99.01%	83.05%	82.28%
Max Voting	99.12%	85.89%	75.28%
XG Boost	99.46%	89.62%	78.28%

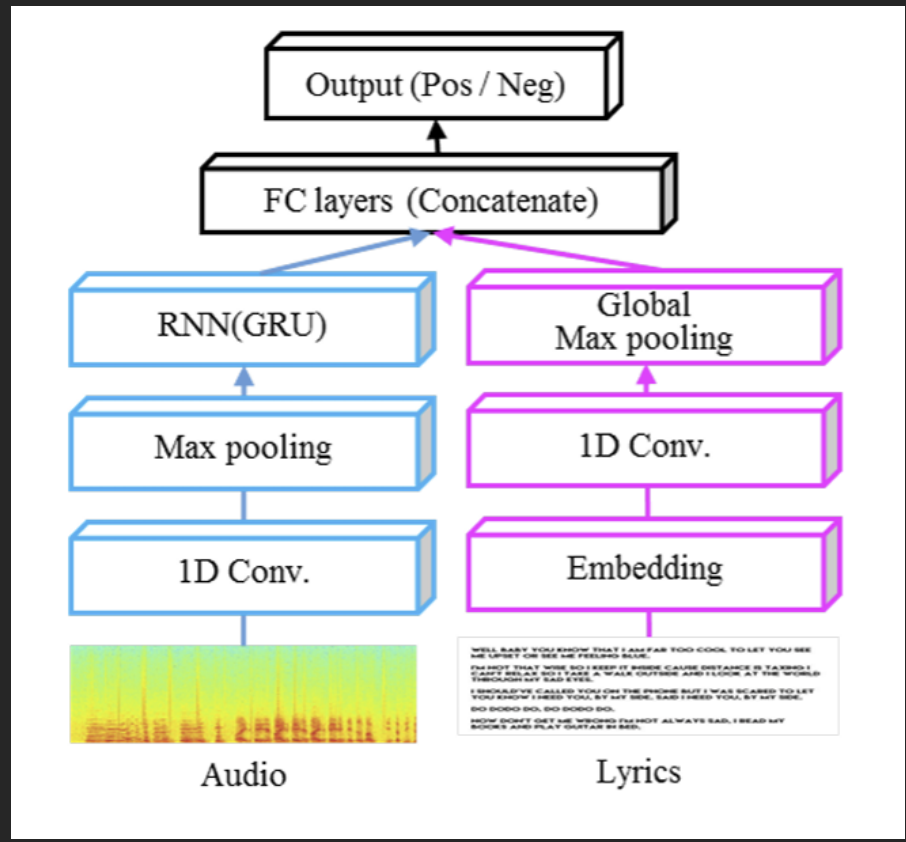
〈논문 실험 결과〉

A Language-Independent Speech Sentiment Analysis Using Prosodic Features(2021, IEEE)

Appendix

Data	Model	Accuracy
Audio	CNN	0.6479
	RNN	0.6303
	CNN+RNN	**0.6619**
Lyrics	**CNN**	**0.7815**
	RNN	0.7716
Both	**CNN+RNN, CNN**	**0.8046**

〈논문 성능 결과표〉

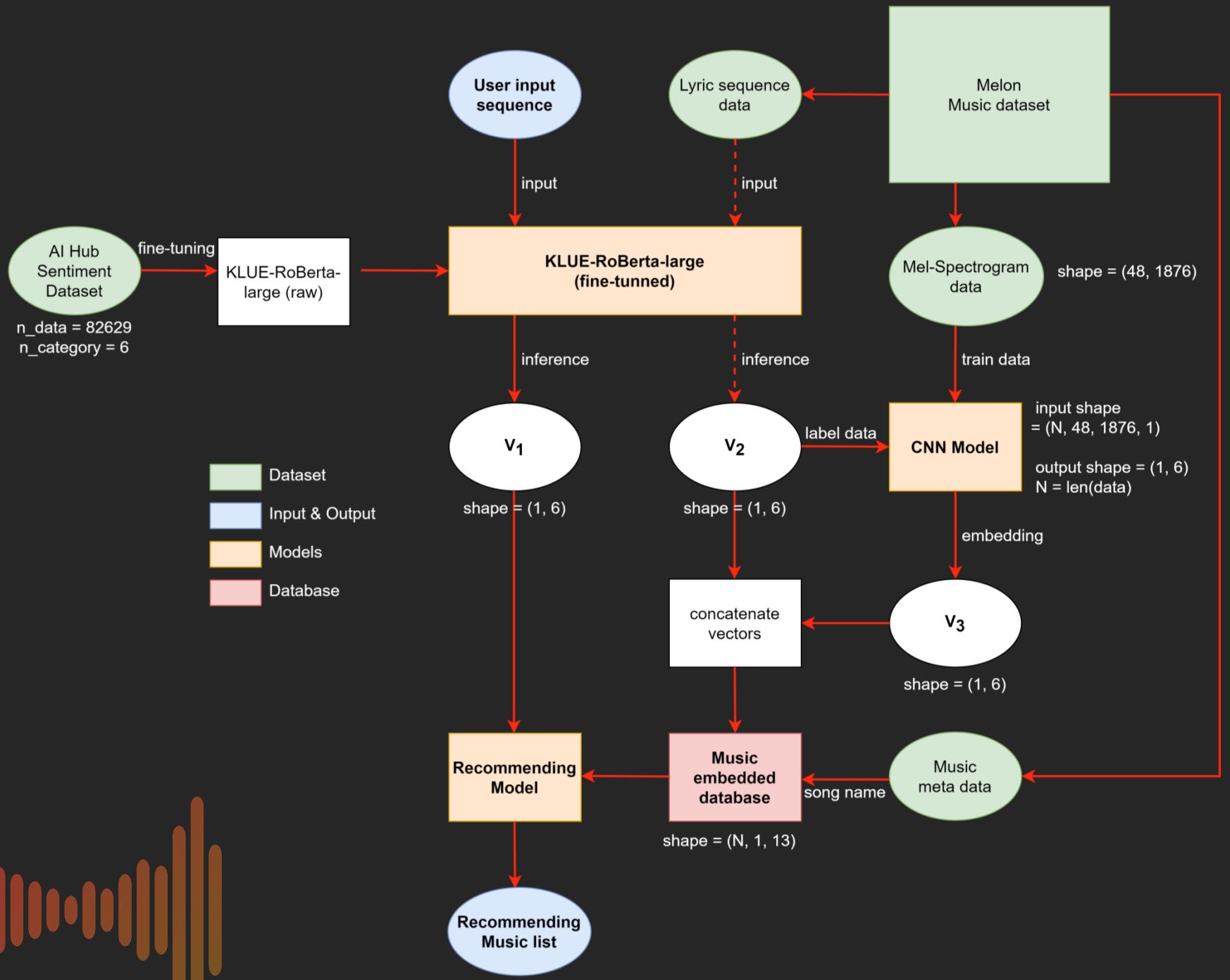


〈모델 아키텍처〉

Music Emotion Recognition via End-to-End Multimodal Neural Networks(2017 Recsys)

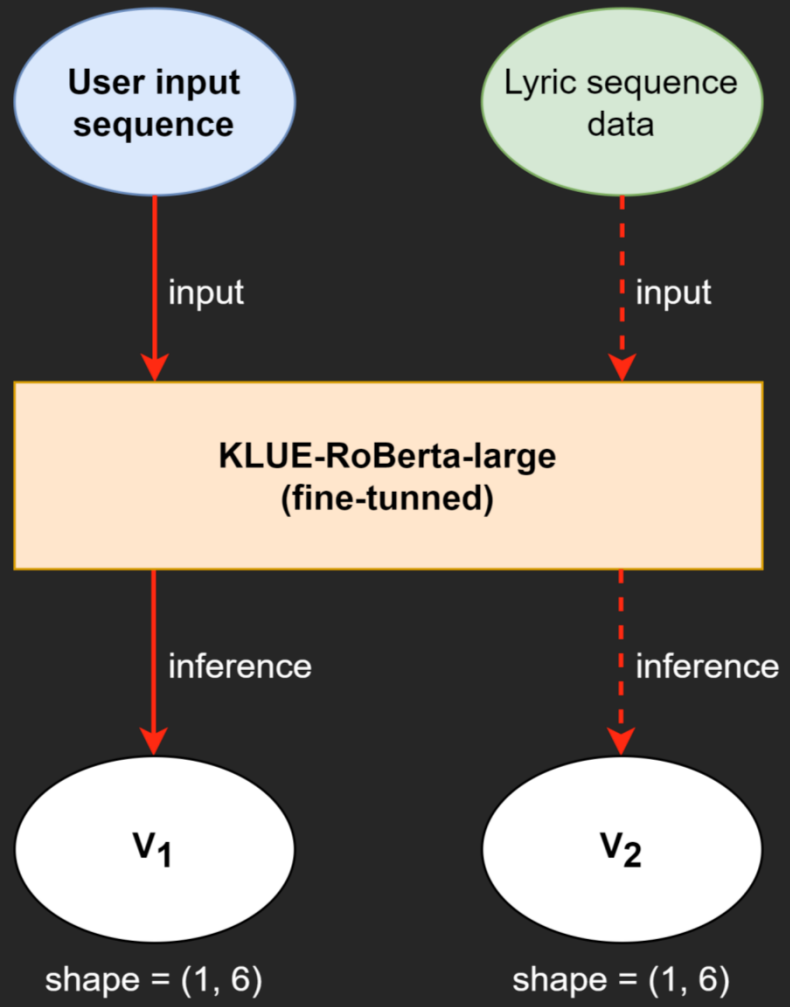
Appendix

전체 모델 아키텍처



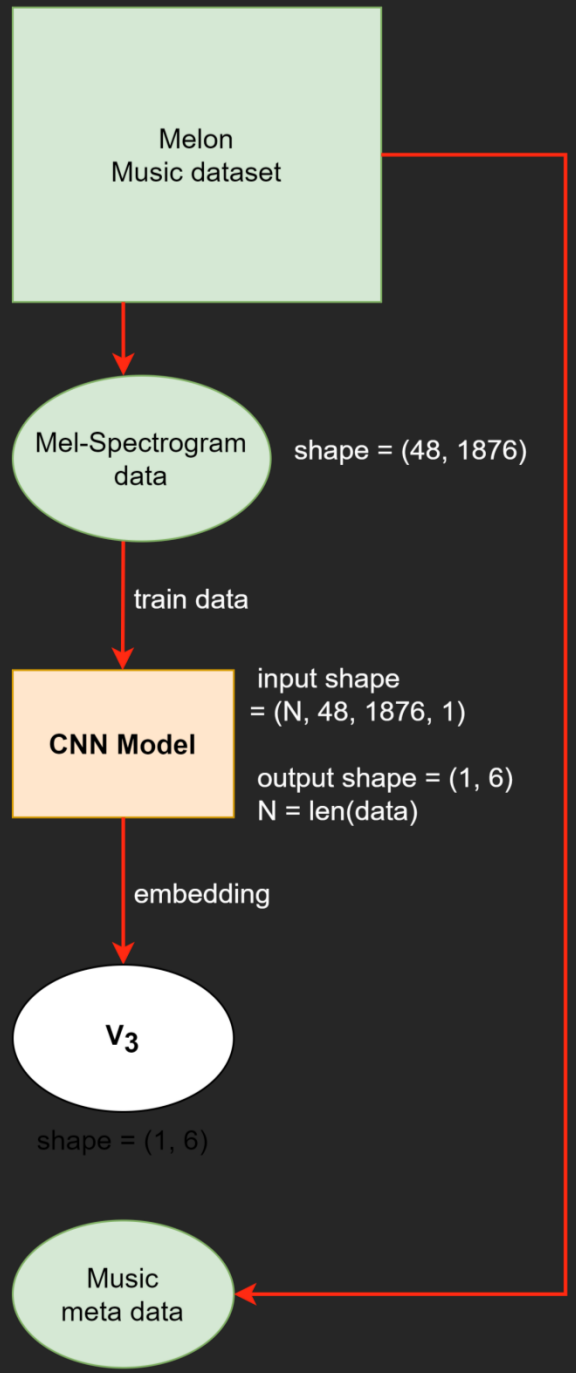
Appendix

NLP 파트 아키텍처



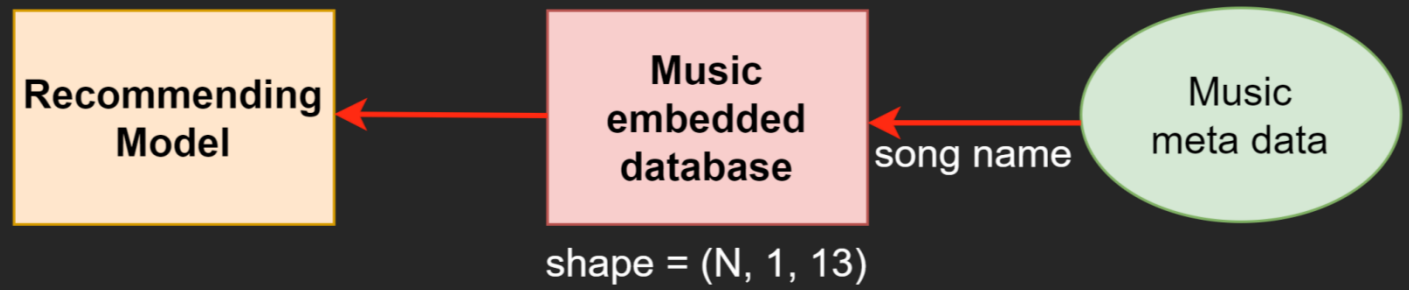
Appendix

Mel 파트 아키텍처



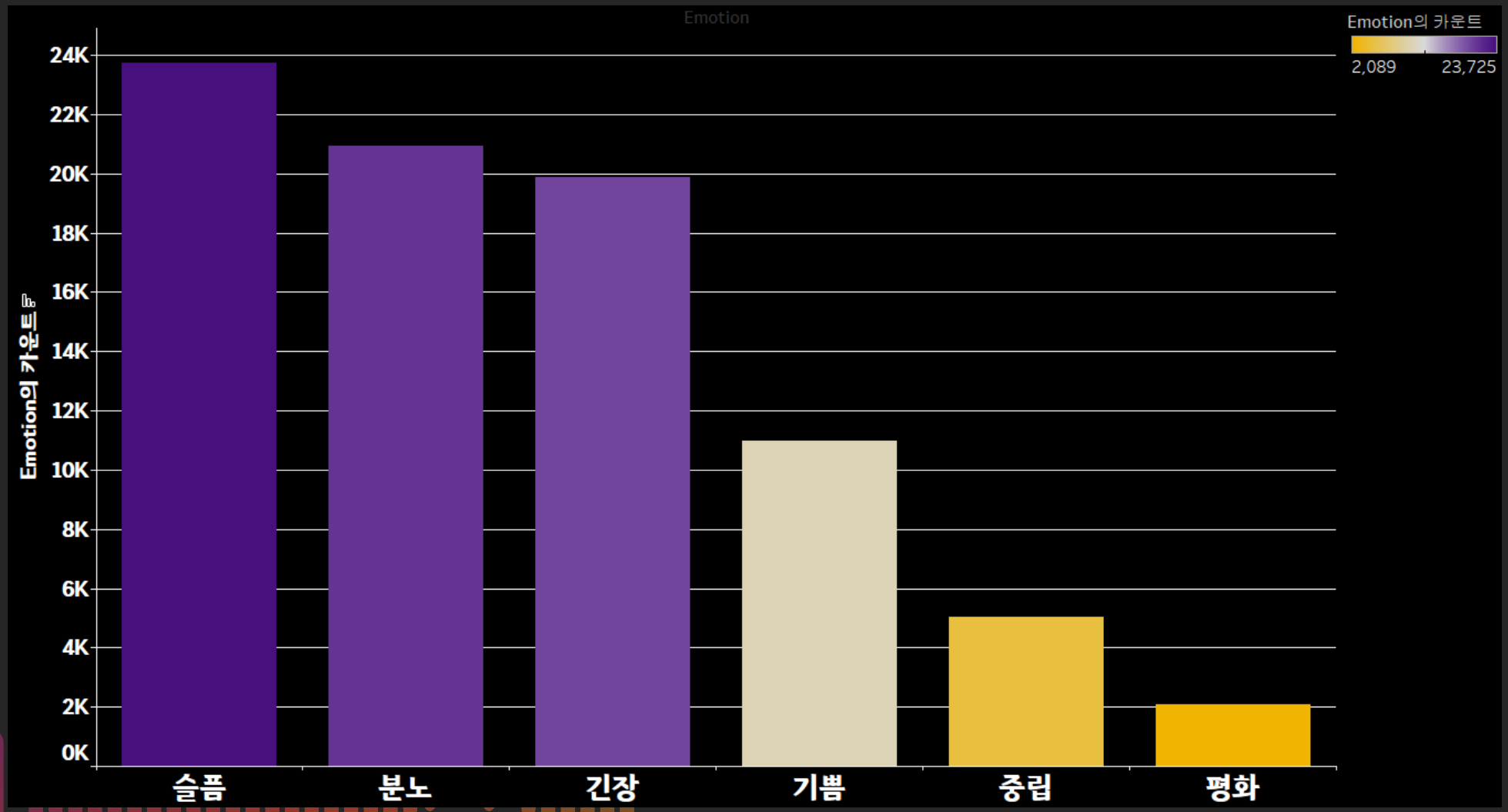
Appendix

추천 파트 아키텍처

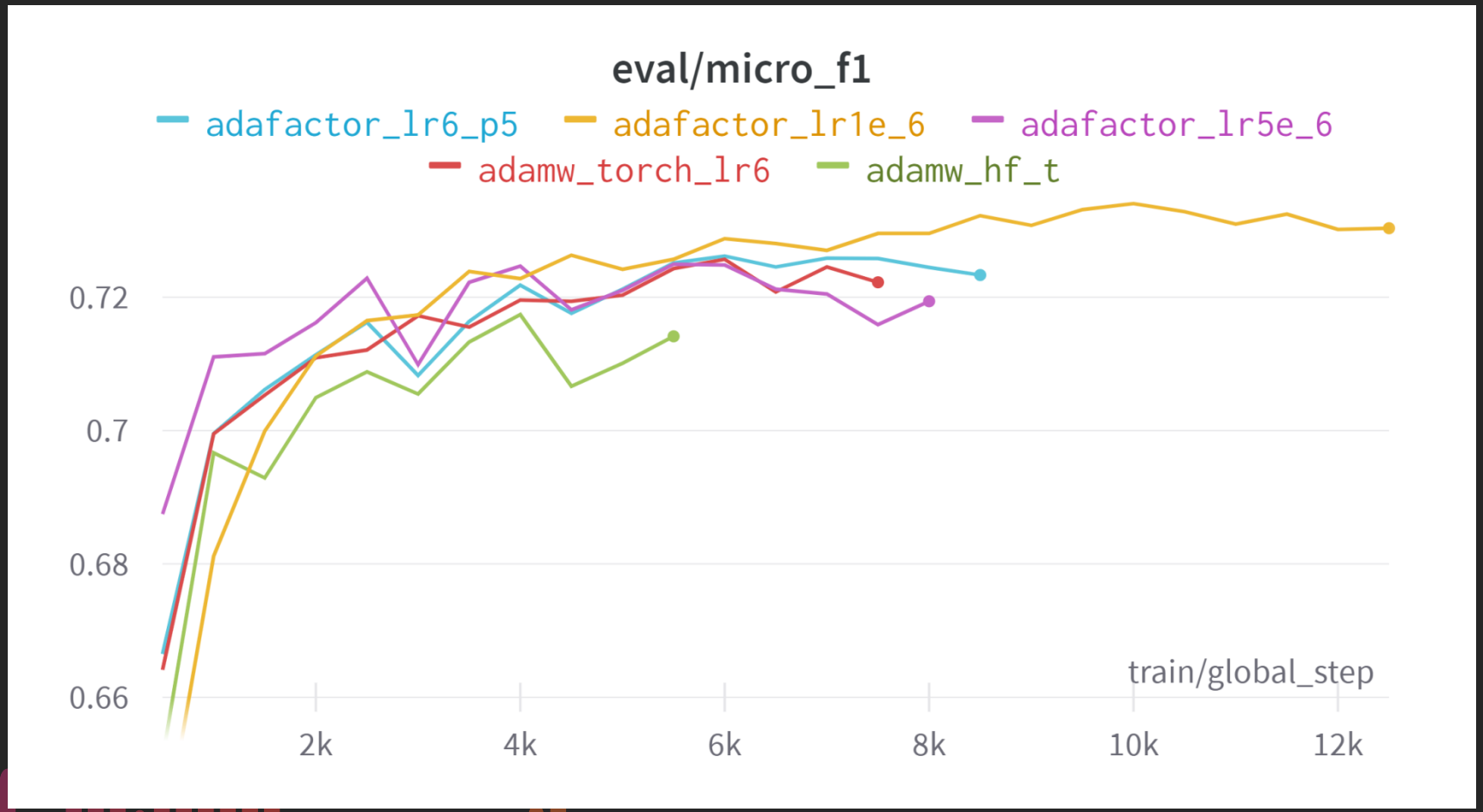


Appendix

최종 감정 데이터 분포

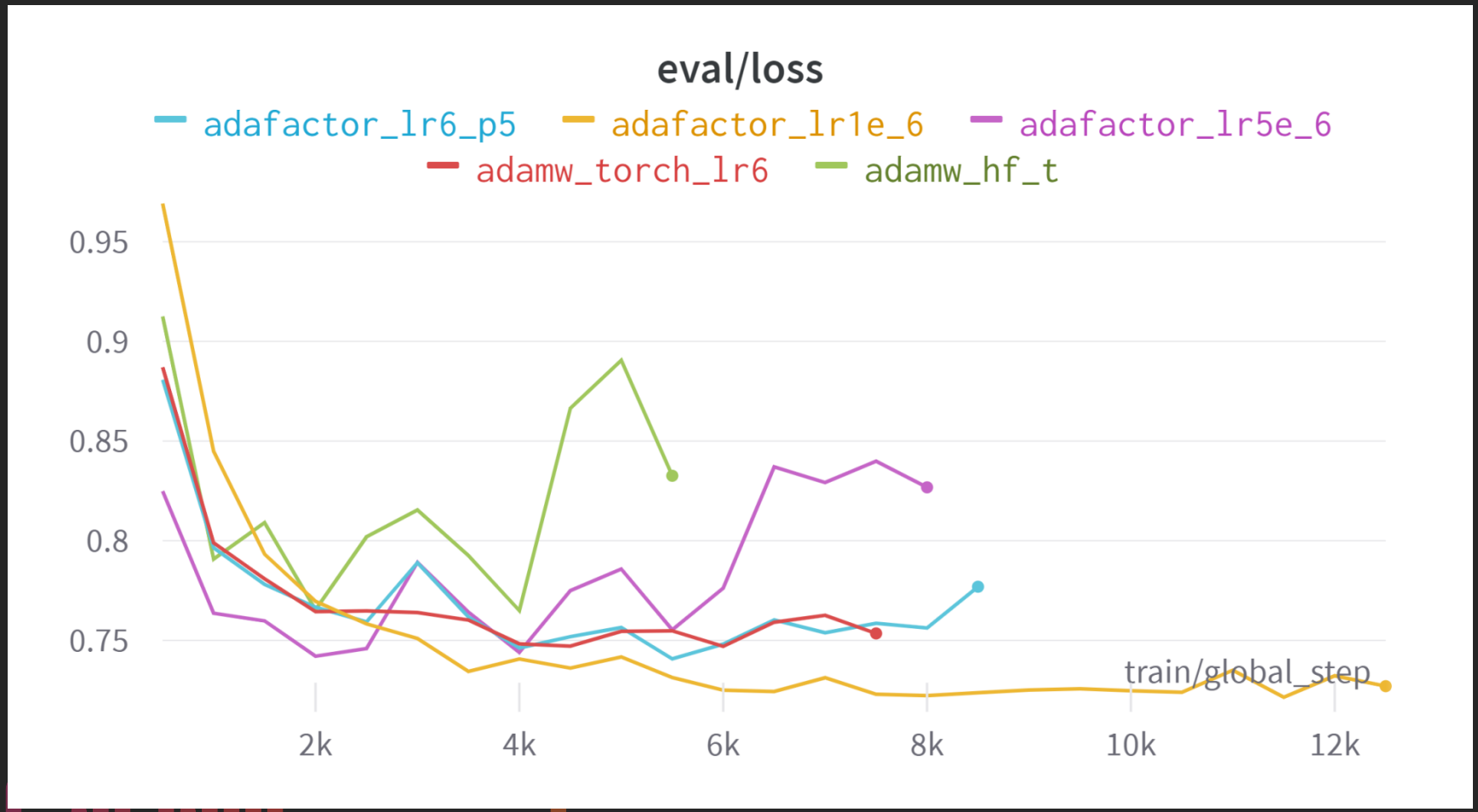


Appendix



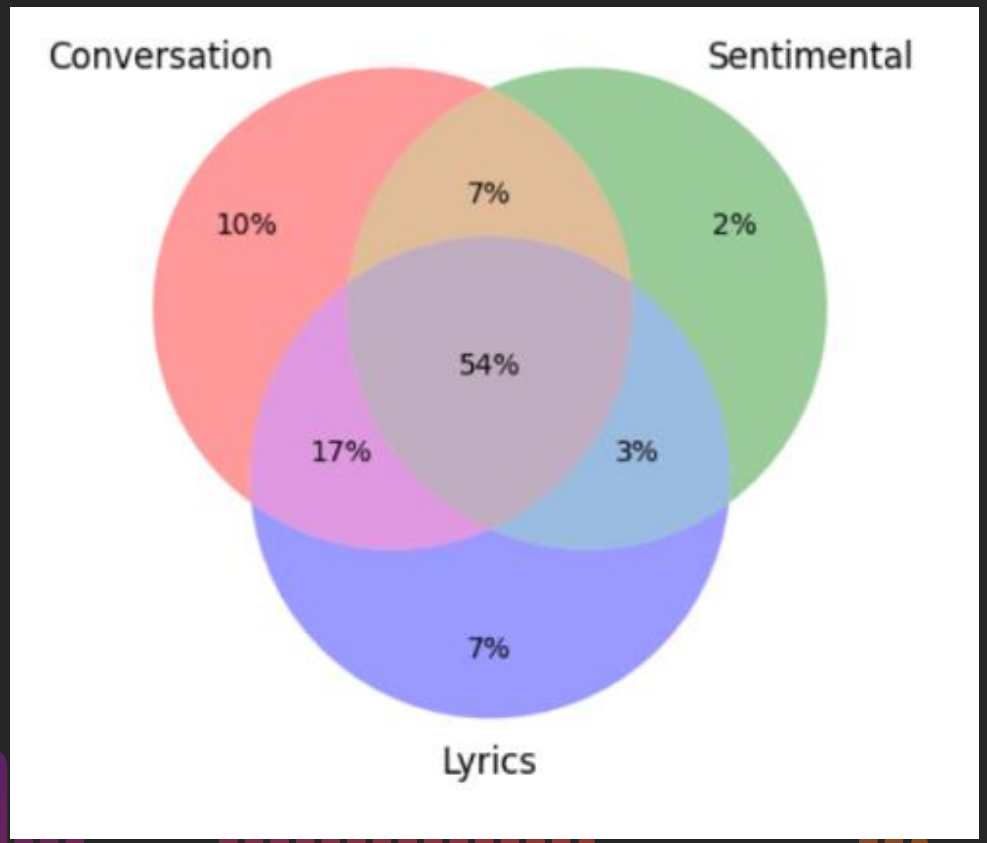
〈모델 학습 과정〉

Appendix

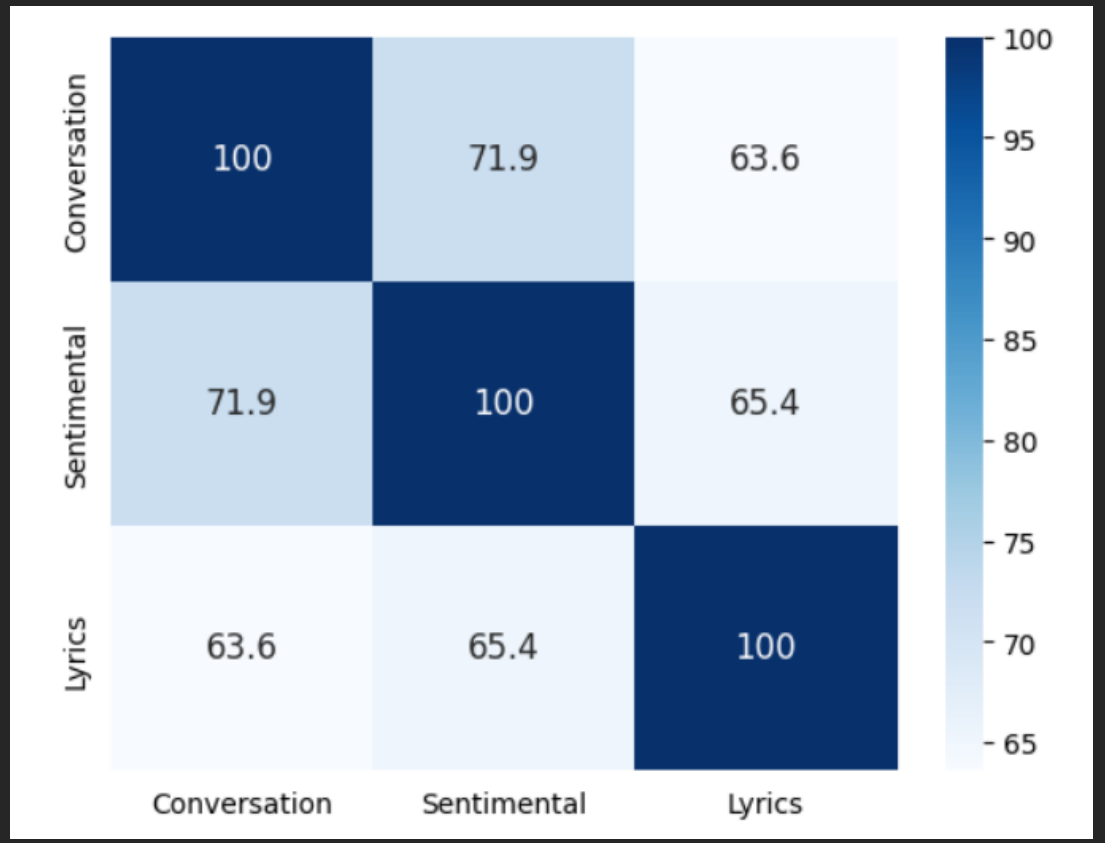


〈모델 학습 과정〉

Appendix

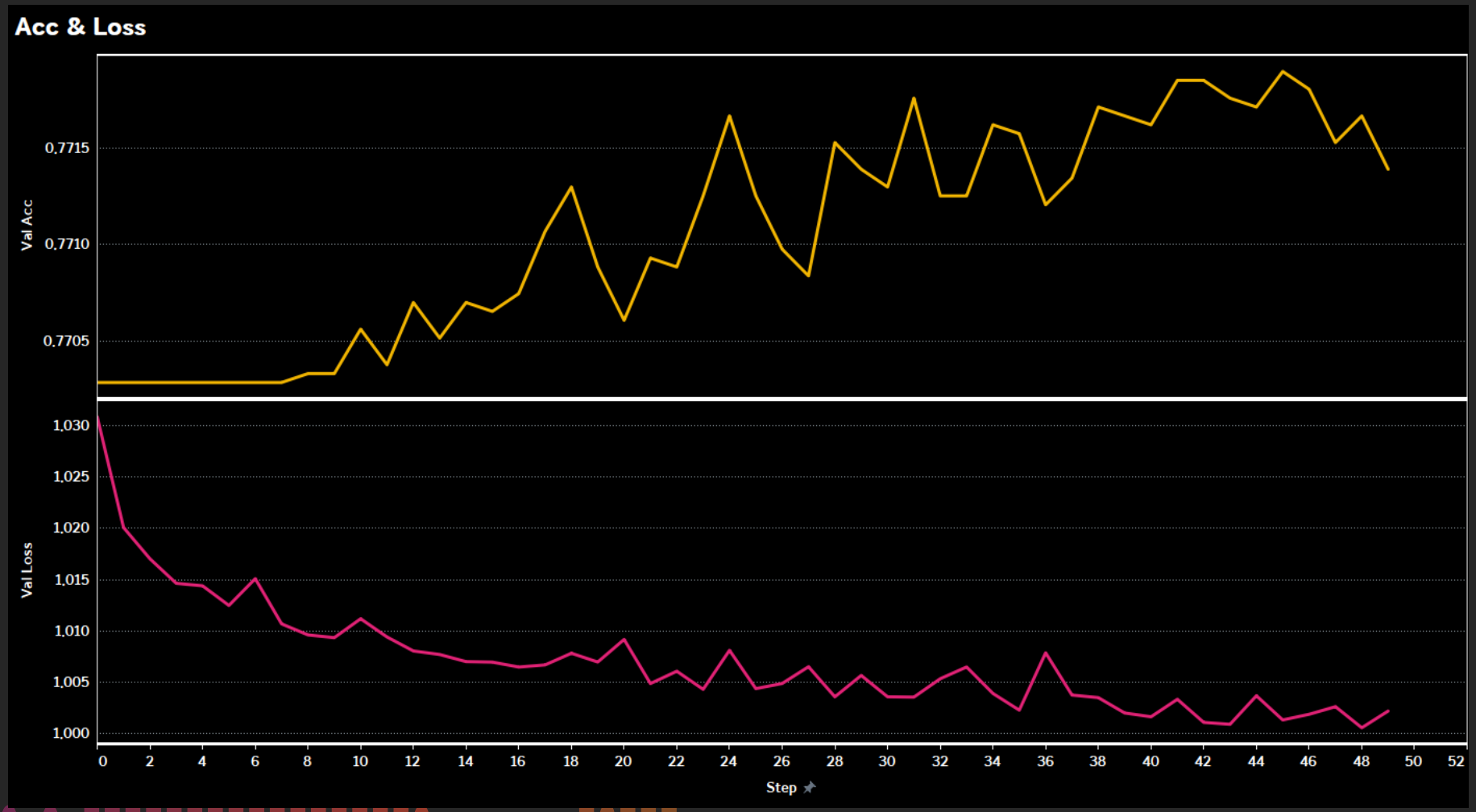


〈데이터별 감성 단어 분포 1〉



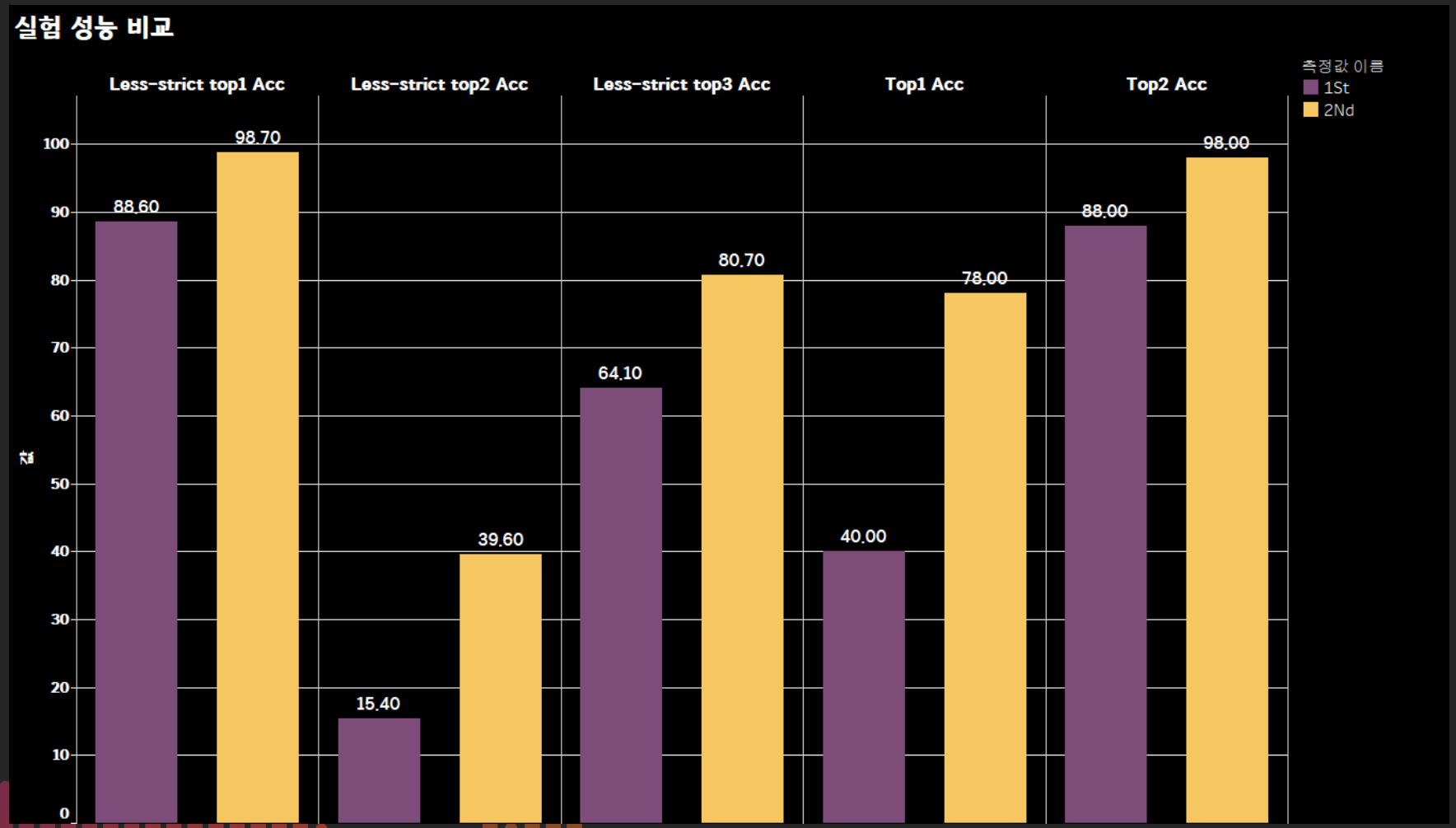
〈데이터별 감성 단어 분포 2〉

Appendix



〈임베딩 이후 모델 학습 추이〉

Appendix



〈임베딩 전후 성능 비교〉

Appendix

Less-strict Top1 Acc

둘 중 하나 이상 맞추는 경우

Less-strict Top2 Acc

순서 상관없이 셋중 둘 이상 맞추는 경우

Top2 Acc

순서 상관없이 둘 다 맞추는 경우

Top3 Acc

순서 상관없이 셋 다 맞추는 경우