Professional Portfolio

Jason Heesang Lee

List of Contents

Timeline

GLION INSTITUTE OF HIGHER EDUCATION
2015.08 ~ 2021.02
Bachelor of Business Administration
Hospitality Management
With Luxury Specialization

FOUR SEASONS HOTEL SEOUL
2021.04 ~ 2021.09
People & Culture
Coordinator

Atheneum
2021.09 ~ 2023.02
Associate
Sr. Associate
Team Lead

Fast campus × KOSME
이어드림스쿨 YEAR-DREAM SCHOOL
2023.04 ~ 2024.02

First
Hello World!
First
Kaggle Submit
Achieved
Notebook Expert
Achieved
Notebook Master
Released
DanGam

2020
2021
2022
2023

Apr
Sep
Jun
Aug
Mar
Apr
Aug
Oct
Dec

Previous Jobs & Career Transition

FOUR SEASONS HOTEL SEOUL

스위스 Glion Institute of Higher Education 졸업 후,
포시즌스 호텔 서울 인재문화부의 코디네이터로 입사하였습니다.

인사행정, 신규 입사자 및 퇴사자 인터뷰, 내부 직원행사, 취업 박람회 개최
등의 업무를 담당했습니다.

Atheneum

프로젝트 매니저로서 매월 약 15개의 프로젝트를 담당했습니다.
평균 250%의 성과를 달성했고, 최고 425%의 성과를 달성하였습니다.

아테네움 파트너스 전 지사 중 가장 빠르게 Team Lead로 승진하였습니다.

AI, 반도체, 은행 등의 산업군을 주로 담당하였으며,
Digital Transformation 관련 프로젝트를 주로 진행했습니다.

fast campus × KOSME 중소벤처기업진흥공단

이어드림스쿨

스타트업-청년인재 YEAR-DREAM SCHOOL

아테네움 파트너스에서 담당하던 프로젝트인 "반도체 산업 벤치마킹 프로젝트", "금융권 기업 디지털 전환 프로젝트"를 통해
데이터와 인공지능에 대해 흥미를 가지게 되었고 관련 소식을 주기적으로 찾아 보았습니다.

인공지능 기술이라면 제가 가진 후각 손실을 해결할 수 있을 것이란 생각으로, 직접 만들어 보고자 직무변경을 결심하게 되었습니다.

2023년 3월 Python을 처음 배우기 시작한 후, 4월부터 중소벤처기업 진흥공단과 패스트캠퍼스 주관의 AI부트캠프인
이어드림스쿨에 참여하며 데이터 사이언스 분야를 학습 중이며, 대회 및 기업연계 프로젝트 등으로 실력을 키워 나가고 있습니다.
다양한 언어를 구사할 수 있다는 점과 언어별 구조적 특성에 관심이 많아 자연어처리 분야에 관심이 깊어지게 되었고,
이와 관련한 다양한 대회 또한 참여하였습니다.

Competitions

AI CONNECT

Generating Seamless En-Ko Translation

- **Date :** 2023-10-27 ~ 2023-11-08
- **Type of Data :** NLP / LLM / 기계번역
- **Rank :** 2nd
- **Used Models :** NLLB, LLaMa2 based pretrained models
- **Focus :** LLM을 통한 번역문 생성 경험

AIFactory

- **Predicting Particular Matter Pollution Degree**
  - **Date :** 2023-04-24 ~ 2023-05-11
  - **Type of Data :** Tabular / 시계열 / 특정 지역의 미세먼지 예측
  - **Rank :** Unranked
  - **Used Models :** None (EDA only)
  - **Focus :** Pandas 활용에 익숙해지는 것을 목표로 진행했습니다.

DACON

- **Judicial Precedent Prediction**
  - **Date :** 2023-06-05 ~ 2023-07-03
  - **Type :** NLP / Classification
  - **Rank :** Public : 15% / Private 18%
  - **Used Models :** Sentence-BERT / Legal-BERT
  - **Focus :** 텍스트 데이터 처리 방법 공부
- **Sound Emotion Recognition**
  - **Date :** 2023-05-07 ~ 2023-06-05
  - **Type :** Acoustic / 감정 분류
  - **Rank :** Public : 43% / Private 40%
  - **Used Models :** Librosa / XGBoost / LightGBM
  - **Focus :** 음성 데이터 처리 방법 공부

kaggle

CAFA 5 Protein Function Prediction

- **Date :** 2023-04-18 ~ 2023-12-21
- **Type of Data :** Tabular / Biology / 단백질 기능 예측
- **Rank :** Public 4% / Private 4% / Top 63rd
- **Used Models : ProtBERT, Prot-T5, ESM2.**
- **Focus :** Protein Language Model 활용하여 문제 해결

- **Kaggle – LLM Science Exam**
  - **Date :** 2023-07-12 ~ 2023-10-11
  - **Type of Data :** NLP / LLM / 위키피디아 정보 기반 Open Book 시험
  - **Rank :** Public : 15% / Private 15%
  - **Used Models :** T5, DeBERTa, LLaMA2, Platypus2, Alpaca
  - **Focus :** RAG 등 LLM 특화 기법 공부
- **CommonLit – Evaluate Student Summaries**
  - **Date :** 2023-07-13 ~ 2023-10-12
  - **Type of Data :** NLP / LLM / 요약문 평가
  - **Rank :** Public : 7% / Private 30%
  - **Used Models :** MobileBERT, DeBERTa, 기타 BERT 계열 모델
  - **Focus :** 트랜스포머 기반 모델의 경량화
- **ICR – Identifying Age-Related Conditions**
  - **Date :** 2023-05-12 ~ 2023-08-11
  - **Type :** Tabular / 건강상태 기반 나이 관련 질병 보유 여부 판단
  - **Rank :** Public : 7% / Private 48%
  - **Used Models :** Rule-Based / TabPFN / XGBoost / LightGBM
  - **Focus :**Meta Data인 Greeks와 Train/Test Data의 관계를 파악

DANGAM

Word-Level Sentiment Analysis

Date : 2023. 12. ~ Current
Contribution : 100%
Link : Github / PyPi

Purpose

이어드림스쿨의 파이널 프로젝트로
아보카도랜드와 기업 연계 프로젝트를 진행하였습니다.

기업측에서 요구한 사항은 “모모리” 앱 사용자들의 일일 기록에 대해
명사형 키워드를 찾은 후, 이를 긍정 /부정으로 나누는 것이었습니다.

기업 측에서 제공한 베이스라인 코드에서는
감정 추출을 위해 TweetNLP를 사용하지만,
TweetNLP는 문장에 대한 전반적인 감정 추출을 할 뿐,
문장을 구성하는 각각의 단어에 대해 감정 추출은 하지 않았습니다.

기업측에서 이르기를 단어 수준의 감정 추출이 반드시 필요한 것은 아니며,
해당 작업을 수행할 수 있는 모델이 없는 경우
문장 수준의 감정 분석으로 전환하여도 괜찮다고 하였으나,
모델이 없으면 만들면 되지! 라는 생각으로 도전을 하게 되었습니다.

DanGam Overview

Compared with other existing research

- **TweetNLP**
  - TweetNLP는 한국어를 포함한 여러 언어를 지원하며 문장 수준에서 잘 작동하나, 단어 수준의 감정 추출은 지원하지 않습니다.
- **HuggingFace Text Classification Models**
  - TweetNLP와 유사하게 문장 수준에서의 감정을 추출하며, 더욱 세분화된 감정 표현을 제공합니다.
- Word-Level Sentiment Analysis with Reinforcement Learning
  - 해당 연구는 DanGam과 유사하나, 문장 내 모든 단어에 대해 감정추출을 진행하지 않습니다.
- Word-Level Contextual Sentiment Analysis with Interpretability
  - 해당 연구는 DanGam과 유사하나, 해당 연구의 결과물은 Training이 필요한 모델인 반면, DanGam은 Inference Tool로서의 역할을 합니다.

How does it work

- DanGam은 문장을 입력으로 받아 해당 문장 내의 전반적인 감정 (긍정, 부정, 중립)과 세부 감정 (행복, 슬픔, 분노, 평온 등)을 식별합니다.
- DanGam은 문장과 감정, 문장과 세부 감정 간의 코사인 유사도를 계산하여 embedding combining 시 가중치로 활용합니다.
- Combined embedding과 단어별 embedding간의 코사인 유사도를 계산합니다.
- 유사도가 높으면 해당 단어가 combined embedding과 유사한 감정을 갖고 있음을 나타냅니다.

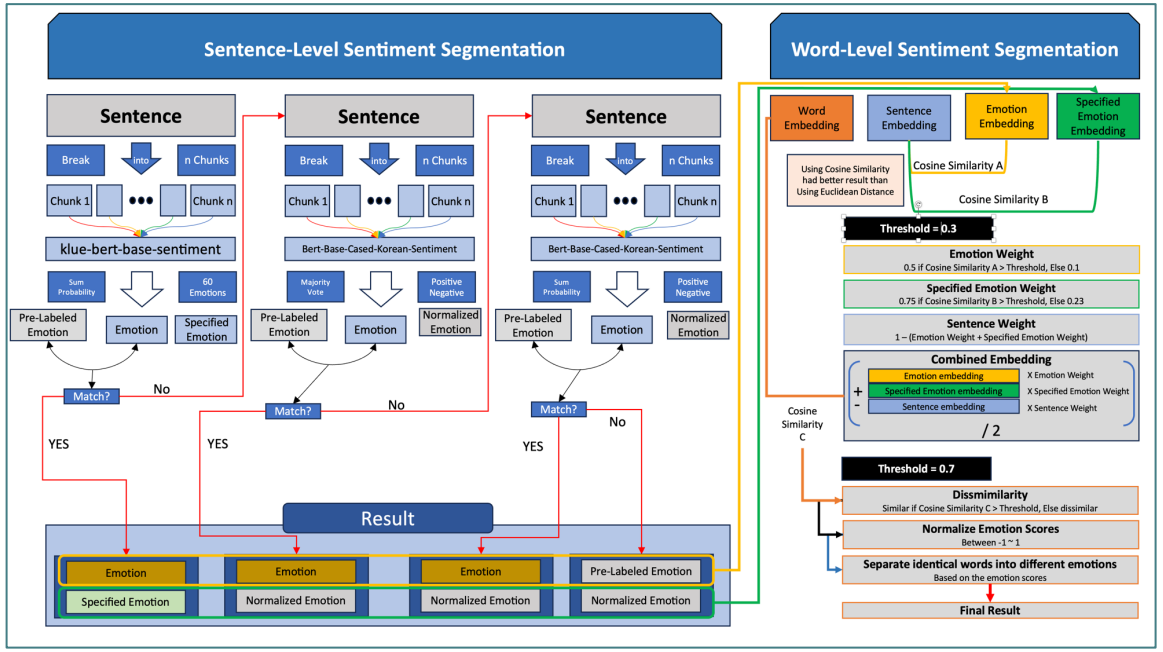
Output Example

- Example Sentence :
  "나는 방금 먹은 마라탕이 너무 좋다.
  적당한 양념에 알싸한 마라향이 미쳤다.
  그런데 고수는 진짜 싫다"
- 단어별 결과값은 아래의 범위 내에서 표현됩니다.
  긍정 (1) ~ 부정 (-1)

```
# {'나는': 1.0,
# '방금': 0.8419228076866834,
# '먹은': 1.0,
# '마라탕이': 0.8522973110543406,
# '너무': 1.0,
# '좋다': 1.0,
# '적당한': 0.965806179144829,
# '양념에': 0.7151325862316465,
# '알싸한': 0.4678710873322536,
# '마라향이': 0.328179239525493,
# '미쳤다': 0.34263925379014165,
# '그런데': -0.07491504014905744,
# '고수는': -0.7992964009024587,
# '진짜': -0.9295882226863167,
# '싫다': -0.9120299268217638}
```

Logic and Key Features

Visualized Diagram



Key Features

get_emotion

- 주어진 문장을 일정한 범위로 분리하여 감정을 추출합니다.
  정확도 향상을 위해 pre-labeled 된 감정과 세부 감정이 있다면 이를 활용합니다.
- 문장의 감정과 세부 감정을 반환합니다.

```
def get_emotion(self,
                sentence: str,
                original_emotion: str = None,
                default_specific_emotion: str = None,
                normalized_emotion: str = None,
                language: str = None,
                ):
```

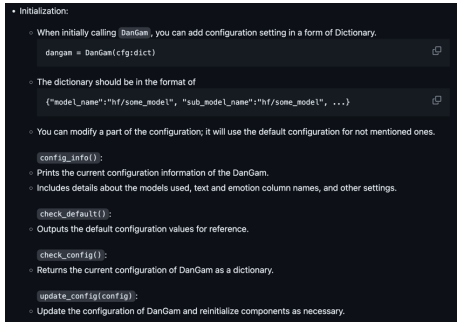
word_emotions

- 문장을 분할하고 문장의 감정과 세부 감정에 따라 각 단어에 감정 라벨을 부여합니다.
- 단어와 감정 점수 (-1 ~ 1)이 매핑된 딕셔너리를 반환합니다.

```
def word_emotions(self,
                  sentence: str,
                  emotion: str = None,
                  specific_emotion: str = None,
                  language=None):
```

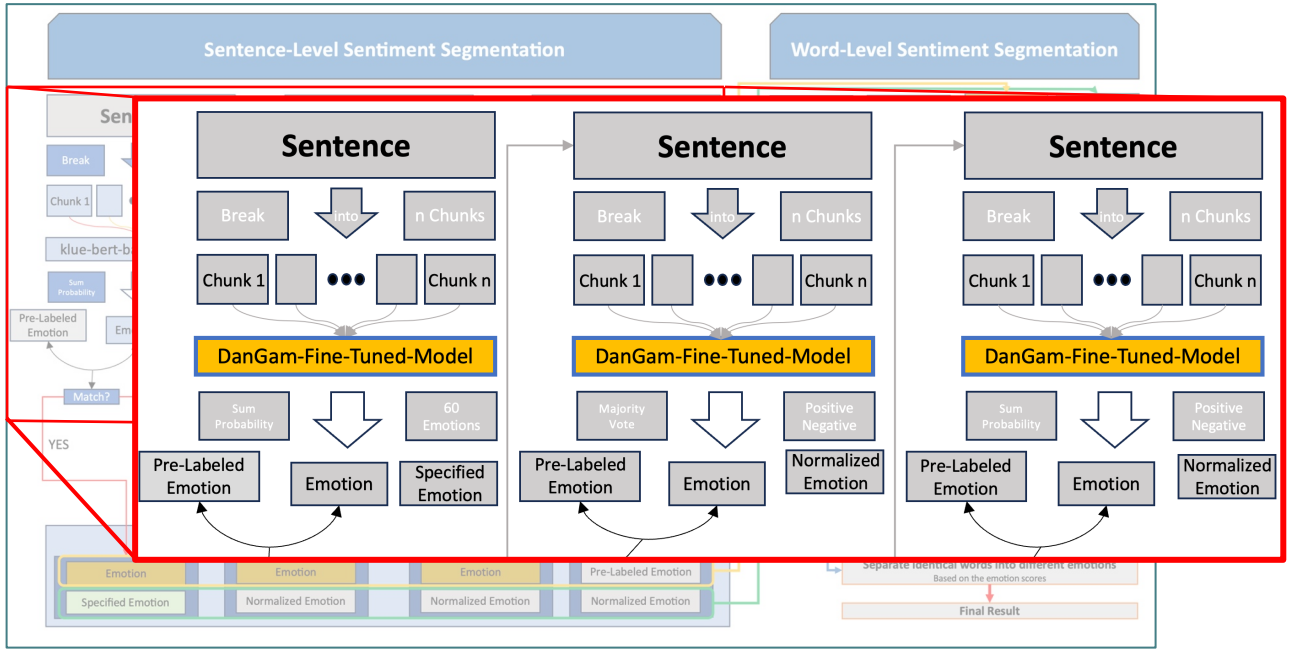
DanGamConfig

- DanGam은 사용자들에게 높은 자유도를 제공합니다.
- 사용자들은 필요에 따라 threshold 등의 세부 설정을 조정할 수 있습니다.



Reflections

What could be improved



- **DanGam 전용의 Fine-Tuned 모델 개발**
  - DanGam은 현재 HuggingFace에서 가져온 Fine-Tuned 모델을 활용합니다. DanGam 전용의 Fine-Tuned 모델은 현재 개발 중이며, 개발이 완료된 후 해당 모델로 대체할 예정입니다.
- **신뢰 가능한 평가지표 구현**
  - 현재 DanGam의 성능은 Human Evaluation 방식으로 평가되고 있습니다.
  - 투입 자원의 효율성을 높이기 위해 신뢰할 만한 평가지표를 구현하여 도입할 예정입니다.
- **Peer Evaluation을 통한 발전**
  - 개인 작업의 한계로 인해 DanGam에는 미처 알아차리지 못한 오류나 모델의 오용이 있을 수 있습니다.
  - GitHub Issue 및 Direct Feedback들은 면밀한 검토 후 DanGam에 반영될 예정입니다.

COMMONLIT

Evaluate Student Summaries

Purpose

CommonLit – Evaluating Student Summary라는 Kaggle 대회에 참여하였습니다.
해당 대회의 목표는 3학년~12학년 학생들의 요약문의 Quality를
평가하는 모델을 개발하는 것입니다.

학생들의 요약문이 명확, 정확, 유창한 지와,
Prompt의 주요 아이디어와 세부 사항을 얼마나 잘 파악하였는지를
평가하는 모델을 구축합니다.

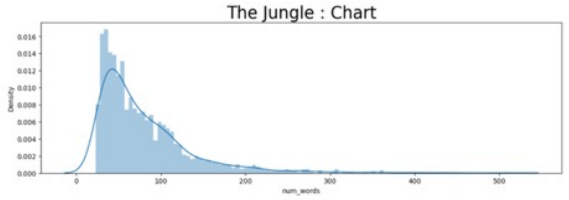
정제되지 않은 학생들의 요약문 데이터가 제공되며,
현업에서 마주칠 수 있는 형식의 실 데이터를 기반으로
모델을 Training 및 개선 시킬 수 있는 대회입니다.

Competition Review

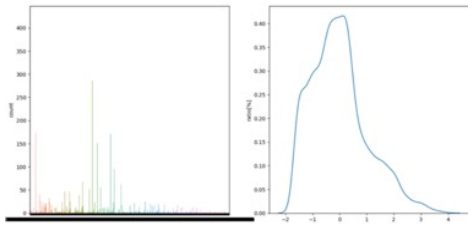
Date : 2023. 07. ~ 2023. 10.
Contribution : 75%
Link : Github / PyPi

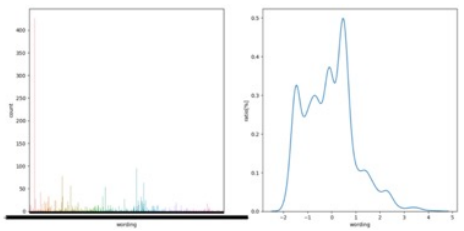
Data Overview

Exploratory Data Analysis



	num_words	num_unique_words	num_chars	num_stopwords	num_punctuations	num_words_upper	num_words_title	mean_word_len
count	1996.0000	1996.0000	1996.0000	1996.0000	1996.0000	1996.0000	1996.0000	1996.0000
mean	79.4509	54.5040	420.7560	42.1242	10.3472	0.0521	3.6844	4.2630
std	55.6905	30.2666	302.9265	29.4349	9.2133	0.3155	3.3914	0.3110
min	23.0000	18.0000	115.0000	8.0000	0.0000	0.0000	0.0000	3.2581
25%	41.0000	32.0000	214.0000	22.0000	4.0000	0.0000	2.0000	4.0522
50%	63.0000	47.0000	330.0000	33.0000	8.0000	0.0000	3.0000	4.2500
75%	100.0000	68.0000	527.2500	53.2500	14.0000	0.0000	5.0000	4.4458
max	509.0000	252.0000	2775.0000	257.0000	88.0000	8.0000	30.0000	5.8947






```
Train content description:
count    7165.000000
mean       -0.014853
std         1.043569
min        -1.729859
25%        -0.799545
50%        -0.093814
75%         0.499660
max         3.900326
Name: content, dtype: float64
```

```
Train wording description:
count    7165.000000
mean       -0.063072
std         1.036048
min        -1.962614
25%        -0.872720
50%        -0.081769
75%         0.503833
max         4.310693
Name: wording, dtype: float64
```

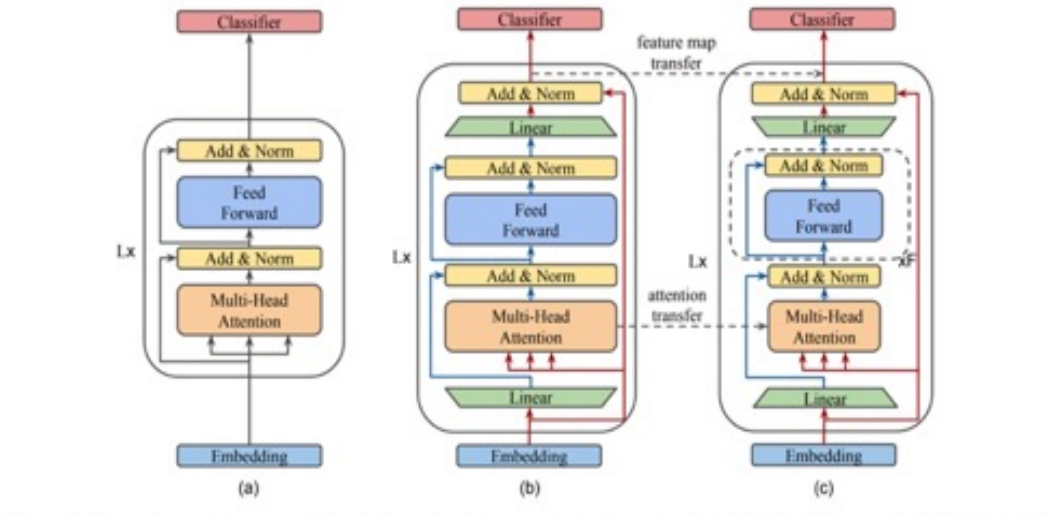
Preprocessing & Spellcheck

- NLP 모델의 최대한의 성능을 내기위해 데이터 전처리는 필수적으로 필요합니다.
- 정제되지 않은 실 데이터인 만큼 아래와 같은 오타가 빈번히 발생하여 기본적인 전처리에 앞서 오타처리를 진행하였습니다.
  ex) “Pharaoh” -> “Pharoh” or “Pharoah”.
- 정확하고 빠른 오타처리를 위해 하단 표에 기재된 네 가지의 라이브러리를 Grammarly 결과물과 비교하며 성능평가를 진행하였으며, 결과는 아래와 같습니다.
- 정확도 : SymSpellPy > PySpellChecker > TextBlob > AutoCorrect
- 속도 : AutoCorrect > PySpellChecker > SymSpellPy > TextBlob

Tool	Detected	Correctly Replaced	Misjudged	Time
Grammarly	16 words	16 words	Less 1 word	-
TextBlob	20 words	14 words	2 words 2 names	4 sec
PySpellChecker	18 words	16 words	2 words	1 sec
SymSpellPy	18 words	17 words	1 name	2 sec
AutoCorrect	15 words	12 words	2 words 1 name	0 sec

Model Application

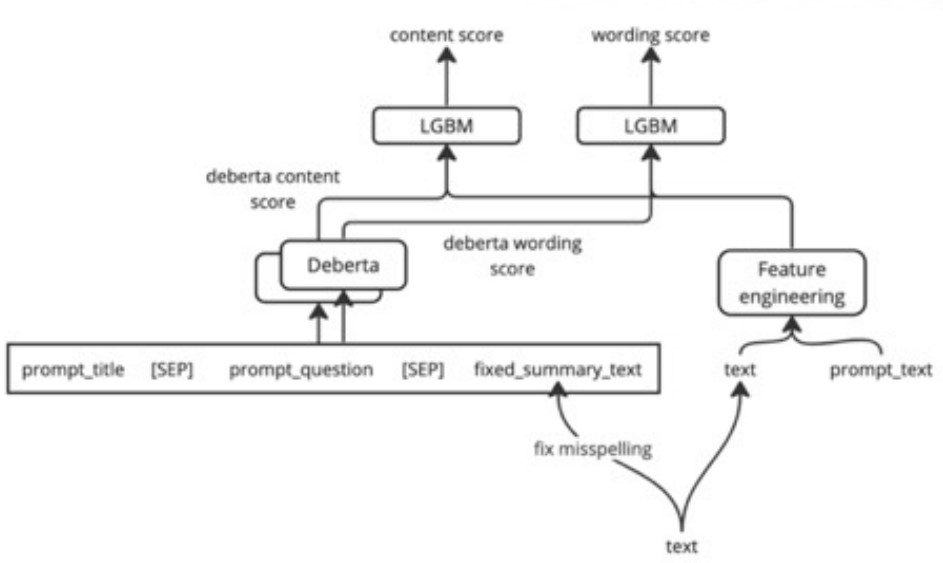
Solving a Real-World Problem with MobileBERT



MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices (Sun et al., 2020)

- 이 대회의 목적은 교사들의 업무 효율도를 높이는 것이 목적이기 때문에 모델을 가볍게 만드는 것이 핵심이라고 생각하였습니다.
- MobileBERT를 도입하면 휴대용 기기에서도 활용 가능하여, 컴퓨터 사용이 어려운 환경에 있는 교사들도 활용 할 수 있습니다.

LGBM – Post-Processing

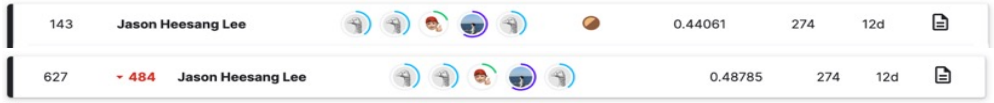


- DeepLearning 모델을 활용하여 얻은 결과에 이 notebook 에 소개된 것처럼 LightGBM을 추가하여 모델의 결과물의 정확도를 향상시켰습니다.
- 해당 노트북의 원작자는 실험을 통해 Hyperparameter을 찾아내었지만, Hyperparameter 최적화 모듈인 Optuna를 활용하면 더 좋은 성능을 내는 Hyperparameter 를 찾을 수 있을 것이란 생각으로 Optuna를 마지막 단계로 추가하였습니다.

Result & Reflection

Result

- 대회 마감일에 Public 리더보드에서 143위를 기록하여 Private에서도 좋은 기록을 가질 것이라 기대하였지만,
  Private 리더보드에서는 627위에 그쳤습니다.



What could be improved

- **First** : 공개된 데이터의 비율.
  주최 측에서 Test에 활용하는 데이터는 전체 Test 데이터의 13%에 불과하다고 하였습니다.
  이후 소개할 1위 팀이 접근한 방식처럼 LLM을 통해 데이터 증강을 했다면 더욱 좋은 성과를 낼 수 있었을 것입니다.

- **Second** : 오타 교정에 SymspellPy 활용.
  대부분의 메달권 참가자들은 PySpellChekcer 와 AutoCorrect를 혼용하여 활용하였습니다.
  오타 교정 라이브러리 성능 평가 시 더욱 많은 데이터를 활용하였다면 더욱 객관적으로 성능 평가를 진행하였을 것입니다.

- **Third** : MobileBERT에 장기간 집중.
  더욱 높은 순위를 목표로 하였다면 MobileBERT는 고려 대상이 되면 안됐습니다.
  다만, 모델 경량화라는 목표가 있어 시도를 했던 것이기 때문에 후회는 없습니다.

- **Fourth** : 기본적인 이해도와 모델링 실력 부족.
  아직 모델링 실력이 부족하여 시간을 효율적이게 활용하지 못하였습니다.
  이해도와 실력 자체를 키운 후 도전을 했다면 높은 리더보드 순위를 기록함과 동시에, T5, BART 등의 다른 모델들도 시도해 볼 기회가 있었을 것입니다.

1st Place Solution

- 놀랍게도, 1위 참가팀은 저희가 사용한 기준과 동일한 Baseline Code를 활용하였습니다.

- 그들의 접근방식과 저희의 접근방식의 차이는 다음과 같습니다.
  1. 해당 참가팀은 Model 활용 로직은 거의 수정하지 않았고, DeBERTa-v3-Large를 활용하였습니다.
  2. 해당 참가팀은 참가자 모두에게 제공된 네 가지 Prompt와 LLM 모델을 활용하여 CommonLit 웹페이지에 게재된 더욱 많은 Prompt로 요약문을 생성하여 Training 과정에 추가시켰습니다.
  3. 해당 팀은 LightGBM 을 활용하지 않고 DeBERTa-v3-Large 만을 활용하여 문제를 해결하였습니다.

- 그들의 discussion 포스트에 따르면, DeBERTa 모델 Tuning보다 LLM으로 새로운 요약문을 생성하는데 더욱 많은 시간을 사용했다고 합니다.
  - 대회 개요에서 이미 LLM 활용에 대해 언급하여, 대회 주최측이 이 방안을 의도한 것이 아닌가 생각이 들었습니다.