

1st Implementation

# 이건 뭐 사람 아니냐고

---

BERT 기반의 영화평 감상 분석

201310737 김주언

201611829 최보라

201712845 류한길

201411283 유병찬

# I. 프로젝트 개요

## 프로젝트 명: 이건뭐사람아니냐고

- 사람 수준의 판단력을 가진 프로그램을 만들자는 목표로 지은 이름

## 프로젝트 설명

- 영화평 텍스트 데이터를 분석하여, 긍정적인 의미를 담은 문장인지 부정적인 의미의 문장인지 판단하는 프로그램을 제작.
- BERT 기반의 딥러닝을 사용하여 제작.

## 딥러닝 언어 모델

- 1) 형태소 분석기를 사용한 BERT와 형태소 분석을 하지 않는 Electra를 모두 사용
- 2) 둘의 결과를 앙상블로 결합
- 3) 긍정어와 부정어를 구분하는 사전을 학습 과정에서 틀린 문제를 바탕으로 업데이트

## 사용 데이터셋

- 4) 네이버 영화평 <https://github.com/e9t/nsmc/>
- 5) KNU 한국어 감성사전 <https://github.com/park1200656/KnuSentiLex>

## II. 요구사항

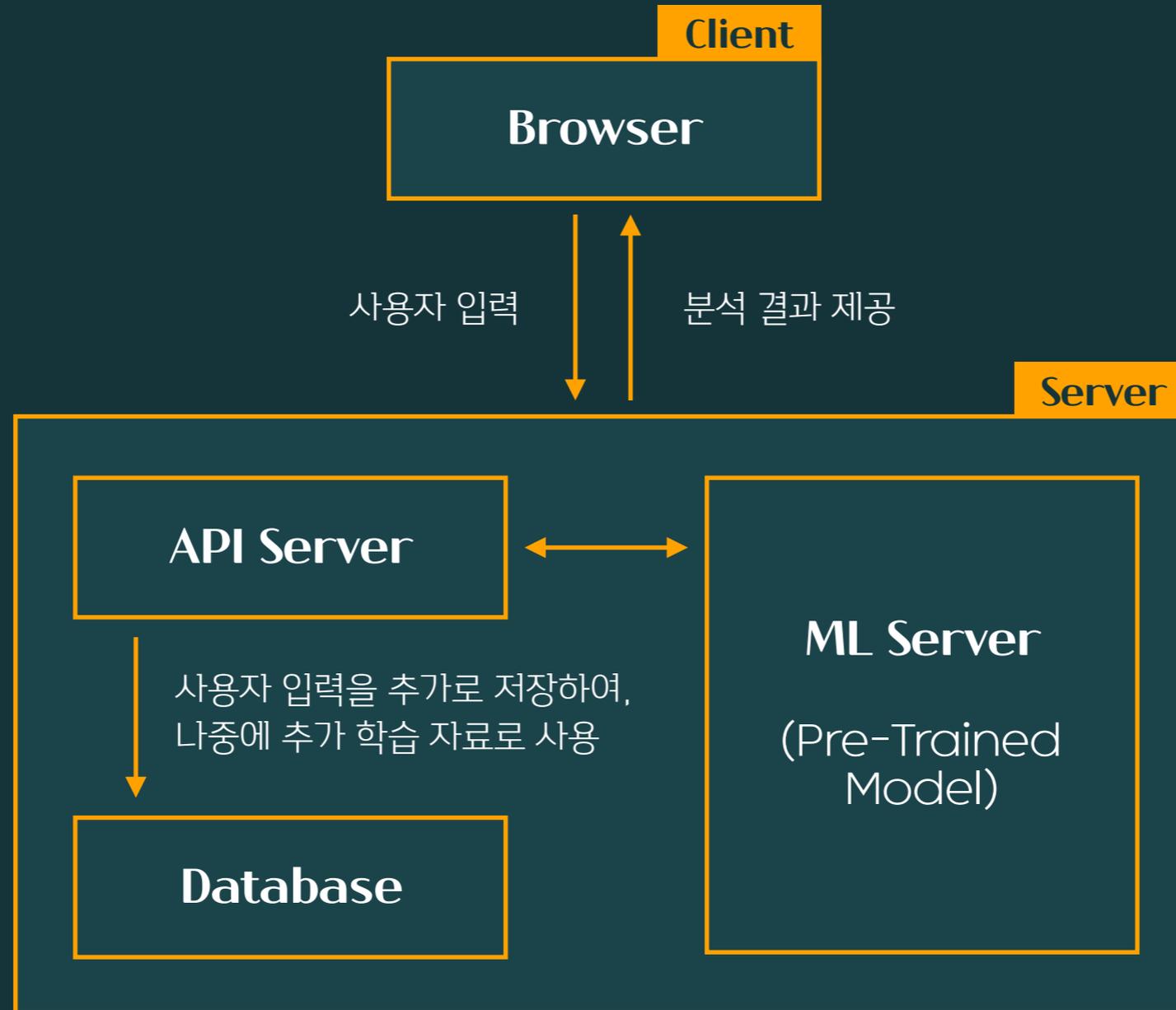
### 기능적 요구사항

- 1) 사용자로부터 감상평을 입력받는다
- 2) 감성 분석 결과를 출력한다
  - 감성 판단 결과를 출력 (긍정/부정)
  - 판단 과정에서 분석된 정보를 시각적으로 제공
    - 단어 판단 결과를 시각적으로 표시 (긍정/부정/중립어)
    - 각 단어의 통계적 발견 비율 표시 (감상평의 경우 한정)
- 3) 사용자가 감성 분석 결과가 맞는지 틀린지 정보를 입력한다

### 비기능적 요구사항

- 1) 감성 분석 정확도가 91% 이상이 되도록 한다
- 2) 사용자의 입력을 받은 순간부터 응답까지 3초 이내가 되도록 한다

# III. 아키텍처 다이어그램



# IV. 프로토타입

## 이건 뭐 사람 아니냐고?

BERT 기반 영화평 긍/부정 판단기

최근에 본 영화에 대한 평가를 적어주세요!

사용자 입력

배우는 별로인데 스토리가 탄탄해서 추천하고 싶다.

긍/부정 판단 결과

결과

배우//	0.000
는//	0.000
별로//	-2.000 ☹️
인데//	0.000
스토리//	0.000
가//	0.000
탄탄//	1.000 😊
해서//	0.000
추천//	2.000 😊
하고//	0.000
싶다//	0.000

최종결과  $[-1.1823558 \ 1.5893534] \Rightarrow 1 \Rightarrow$  긍정적인 영화평 입니다!



최종 결과

정답 여부 사용자 입력

제 생각이 맞았나요??

맞았어요!

틀렸어요!

## V. 딥러닝 모델 설명 - 1

### Transfer Learning (전이 학습)

과거의 머신러닝은 각 Task를 별도로 학습하여, Task 별 많은 데이터가 필요하였을 뿐 아니라, 새로운 Task를 해결하기 위해선 새로 학습을 시켜야 하는 어려움이 있었다. 이를 해결하기 위해 전이학습이라는 방법론이 제시되었는데, 이는 대용량의 일반적 데이터를 사전 학습하고, 그 정보를 새로운 Task에 적용하는 것을 의미한다.

### Transfer Learning의 장점

- 1) 성능이 향상되어 사람보다도 높은 성능을 내기도 한다
- 2) 수렴 속도가 향상되어, 학습 속도가 빨라진다
- 3) 사전 작업(Source Task)에 대한 충분한 데이터만 있다면, 목표작업에대한 데이터가 부족하더라도 비교적 높은 성능을 보이는 모델을 개발할 수 있다. (예전 방식의 머신러닝도 사람보다 빠른 경우가 있었으나, 데이터가 부족한 도메인에서는 어려웠다. 이 부분이 해결된 것)

## V. 딥러닝 모델 설명 - 2

### BERT

이러한 전이 학습 기술을 사용한 모델이 BERT 이다. 처음 등장할 때, NLP 11 개 분야에서 최고 성능을 달성함으로써 유명해졌다.

Masking Prediction 과 Is Next Prediction 으로 사전 학습한 Language Model 의 일종이다. BERT 에서 파생된 여러 모델이 있는데, **우리는 그 중 Electra 를 사용한다.**

### ELECTRA

BERT 의 학습 방식을 바꾸어서, 좀 더 가벼우면서도 성능을 개선한 모델 중 하나이다:

- 가중치 공유를 통해 파라미터를 감소시켜서 가볍다
- Dynamic Masking 으로 학습 효과를 증가
- Is Next Prediction 을 Sentence Order Prediction 으로 변경하여 학습 효과 증가
- Generator - Discriminator 로 기존처럼 사전학습 후에, 옳게 추측했는지 여부를 바탕으로 한 번 더 학습시킨다. 학습 효과가 증가한다

## V. 딥러닝 모델 설명 - 3

### koELECTRA

Electra 는 영어 데이터로 사전학습이 되어 있기 때문에, 한국어 감성 분석을 위해서는 한국어 데이터로 학습시킨 모델이 필요하다. 그 중 하나가 koElectra 이다. 우리 팀은 이 모델을 베이스로, fine-tuning 을 통해 성능을 향상시킬 계획이다.

### Fine-Tuning method

- Bi-LSTM layer 추가
- 군산대 한국어 감성사전 Feature 추가

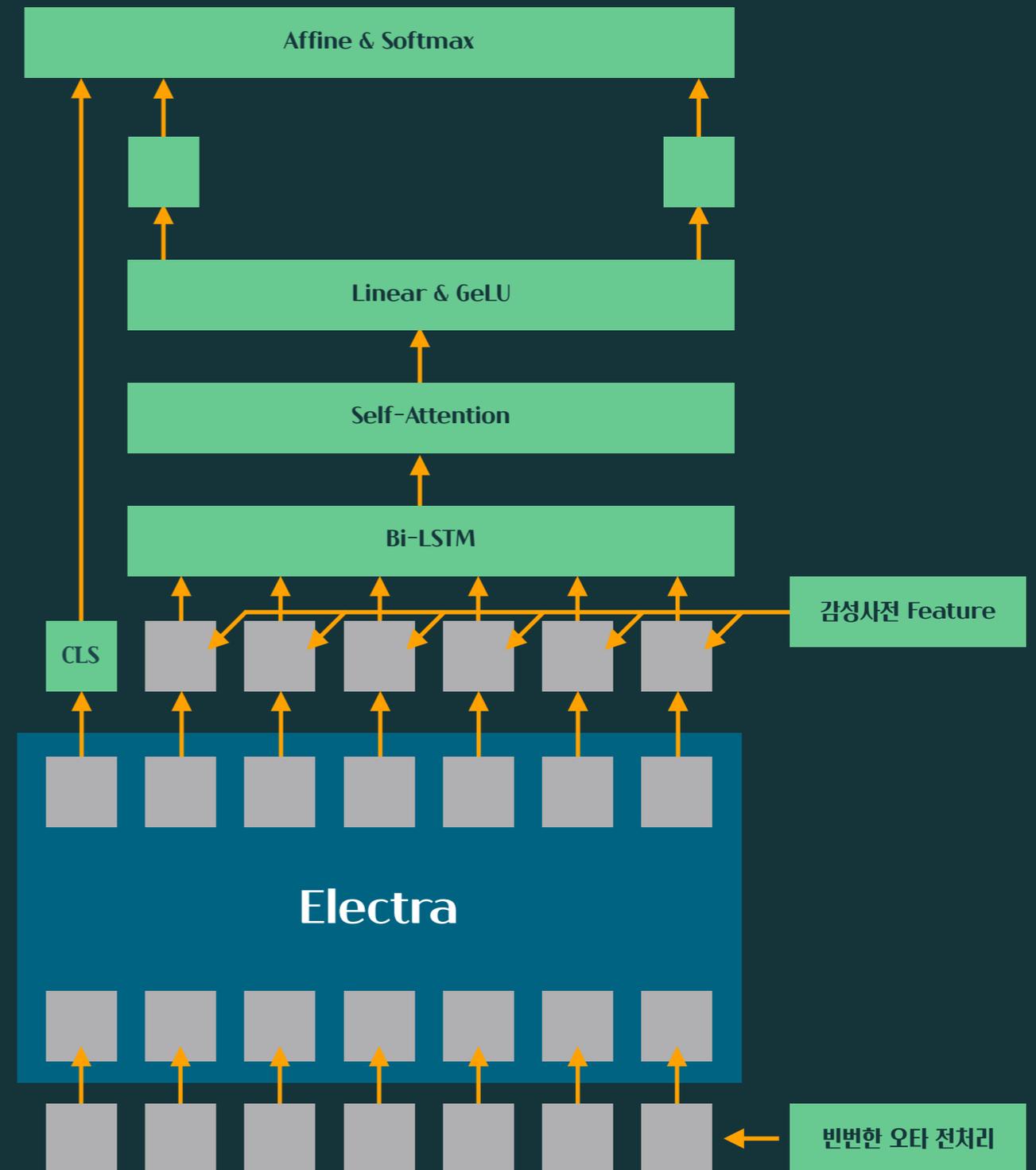
### 학습 데이터

- 네이버 영화평 데이터 15만 건
- 네이버 영화평 자체 크롤 데이터 110만 건

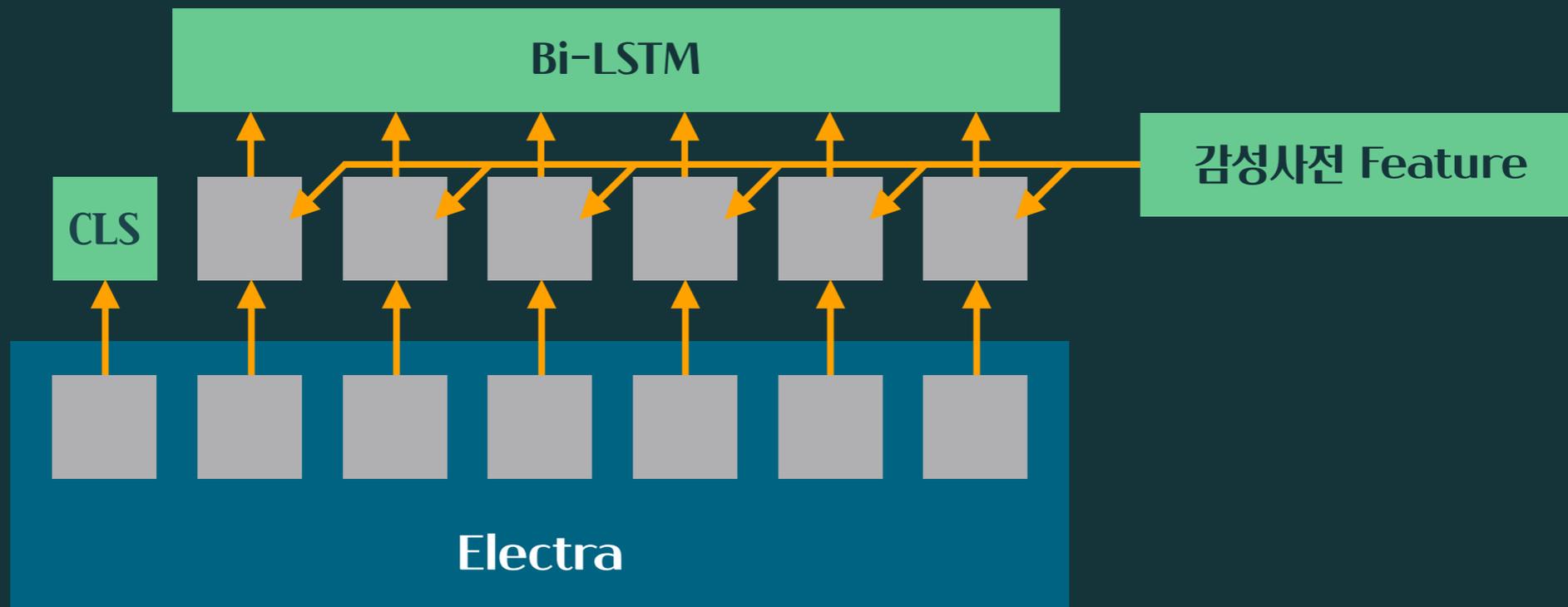
# VI. Fine Tuning Layer - 1

## Fine Tuning Layer 개괄

- Electra 의 출력 word-piece vector 에 한국어 감성사전 데이터를 feature 로 추가한다
- 문장 전체의 데이터를 담고 있는 CLS 값은 별도로 분리하고, 나머지 벡터를 Bi-LSTM -> Self-Attention 으로 학습한다
- Linear Layer 에서 Bi-LSTM 의 결과 벡터와 CLS 의 벡터 차원을 맞춘다
- Bi-LSTM 을 통과한 word-piece 중 맨 앞과 맨 뒤 벡터, 그리고 CLS 를 Affine 한다



## VI. Fine Tuning Layer - 2



### 감성사전 Feature

군산대에서 개발한 한국어 감성사전 데이터와, word-piece 를 연결한다. 감성을 -2, -1, 0, 1, 2 의 5 단계 값으로 구분하고, 감성 정도에 따라 [0, 1] 값으로 변환하여 Electra 의 출력으로 얻은 word-piece vector 에 concatenate 하는 식으로 feature 를 추가한다.

## VI. Fine Tuning Layer - 3



### Bi-LSTM Layer

Electra 의 출력 벡터 중, 문장 전체의 정보를 갖고 있는 CLS 를 따로 분리하고 나머지를 Bi-LSTM 을 거쳐 학습시킨다. 한 단어의 감성 정보는 앞 단어 뿐 아니라 뒤의 단어에도 영향을 받을 것이므로 Bi-LSTM 을 선택하였다.

Self-Attention Layer 를 통해 한 문장 안의 정보들의 관계에 관한 정보를 학습한다.

## VI. Fine Tuning Layer - 4



### CLS Affine

Bi-LSTM 을 통과한 word-piece 벡터들 중 맨 앞과 맨 뒤 값은 각각 뒤에서 앞으로, 앞에서 뒤로 학습한 문장 전체의 의미 정보를 갖고 있다. 이 벡터와 Electra 를 통해 학습된 CLS 벡터를 Affine 계층을 통해 병합한다.

## Ⅶ. 시스템 테스트 케이스

### 기능적 요구사항 테스트

- 1) 사용자 감상평 입력 기능
  - 입력한 댓글이 입력한 대로 서버에 정상적으로 전송되는지 확인
- 2) 감성 분석 결과 출력 기능
  - 해당 영화평에 대한 결과가 올바르게 출력되는지 확인한다
  - 영화평의 긍정어, 부정어, 중립어가 시각적으로 올바르게 출력되는지 확인한다
- 3) 사용자가 감성 분석 결과가 맞는지 틀린지 정보를 입력한다.
  - 사용자가 입력한 정보가 database에 정확하게 입력되는지 확인한다

### 비기능적 요구사항 테스트

- 4) 감성 분석 정확도가 91% 이상 요구
  - 테스트 케이스의 분석 성공 비율이 91% 를 넘기는지 확인
- 5) 사용자의 입력을 받은 순간부터 응답까지 3초 이내 요구
  - 입력된 영화평에 대한 결과가 3초 이내에 출력되는지 확인