종합 설계

- LSTM 모델

- 답변 분류 모델

2020305039 유하영

23.09.19

Dataset

write & create

Question Data

Answer Data

Label

Data Augmentation

MBTI prediction Model

LSTM

Bi-LSTM + Attention

Transformer - Encoder

Answer Analysis

KoNLPy
(Morpheme Analysis)

Word Embedding Vector values for each Key-word

Sentiment Analysis

LSTM

Frequency

Modifier
(adverbs)

UI

Bot

User

E/I Question

User's Answer

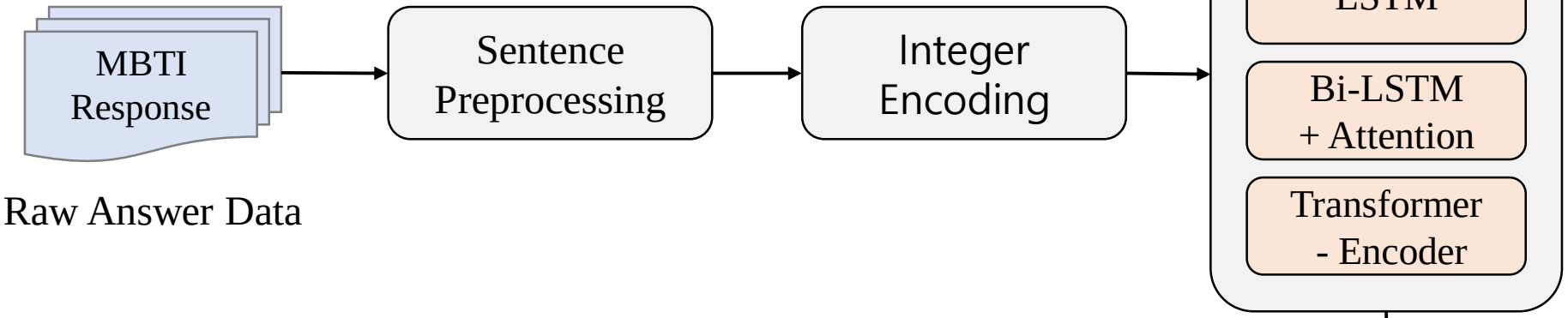
Answer Analysis **Model**

Identifying **a single MBTI type**

Output

: User's **MBTI**

MBTI Feature Learning

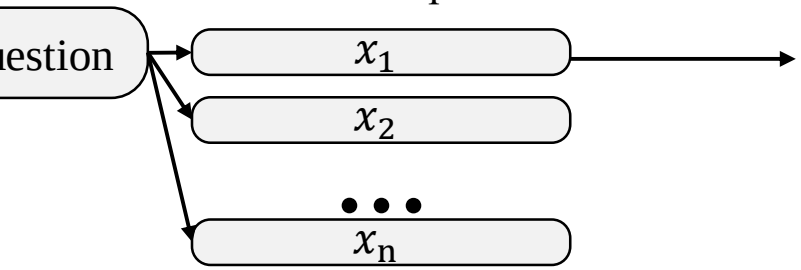


MBTI Response Analysis

UI

Input Data

User's Response



Input Data

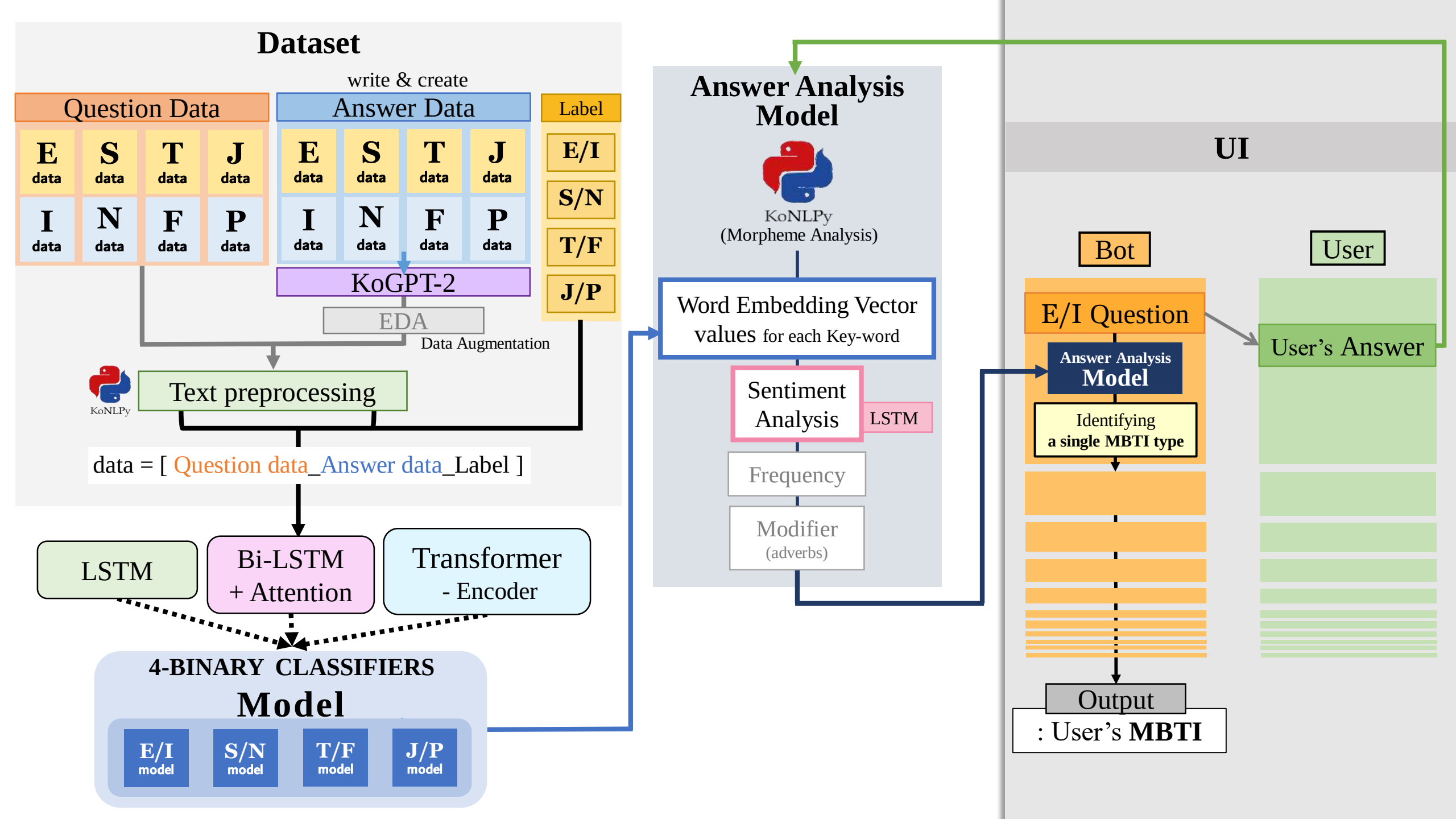
Sentiment Analysis

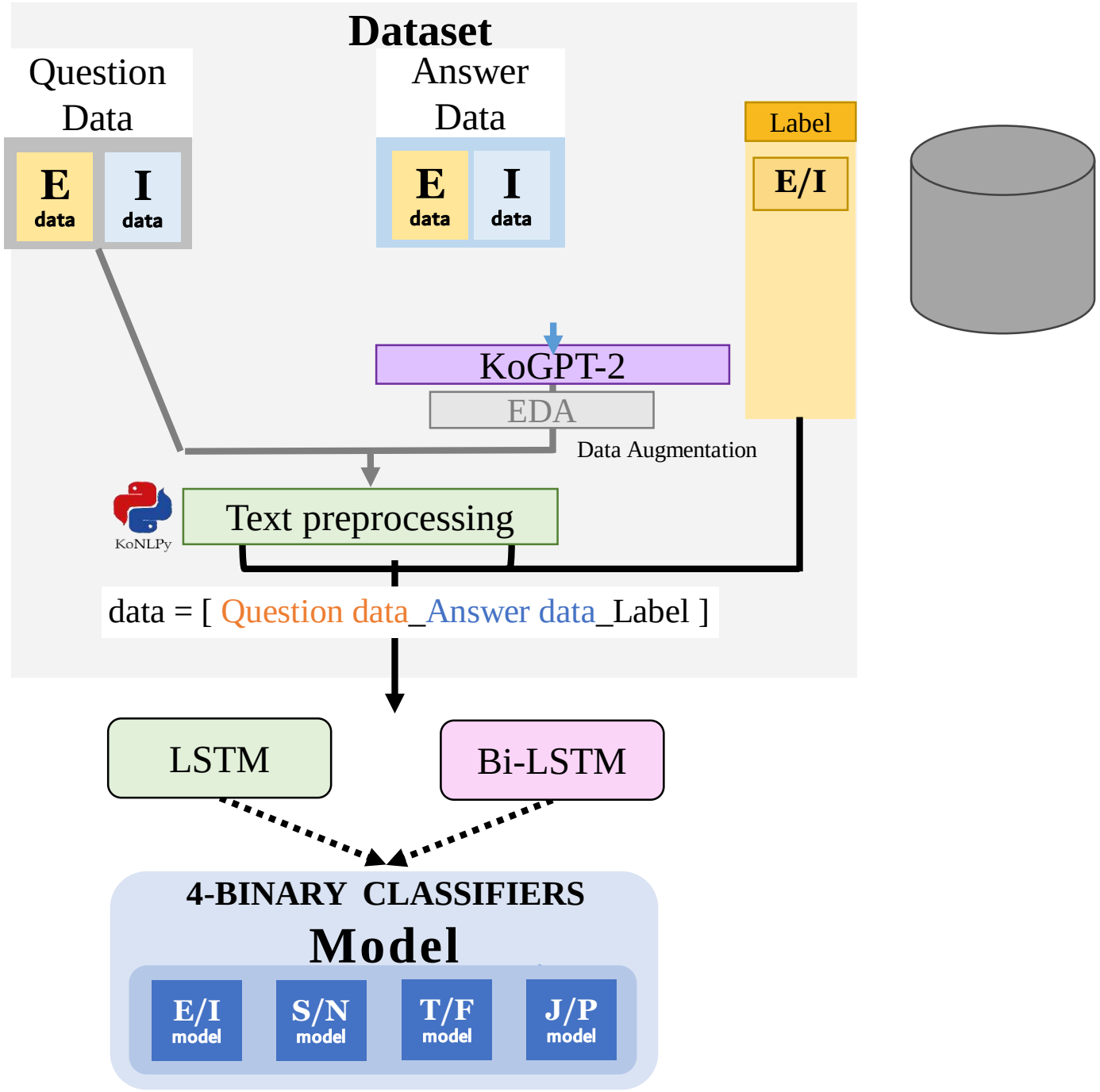
Frequency

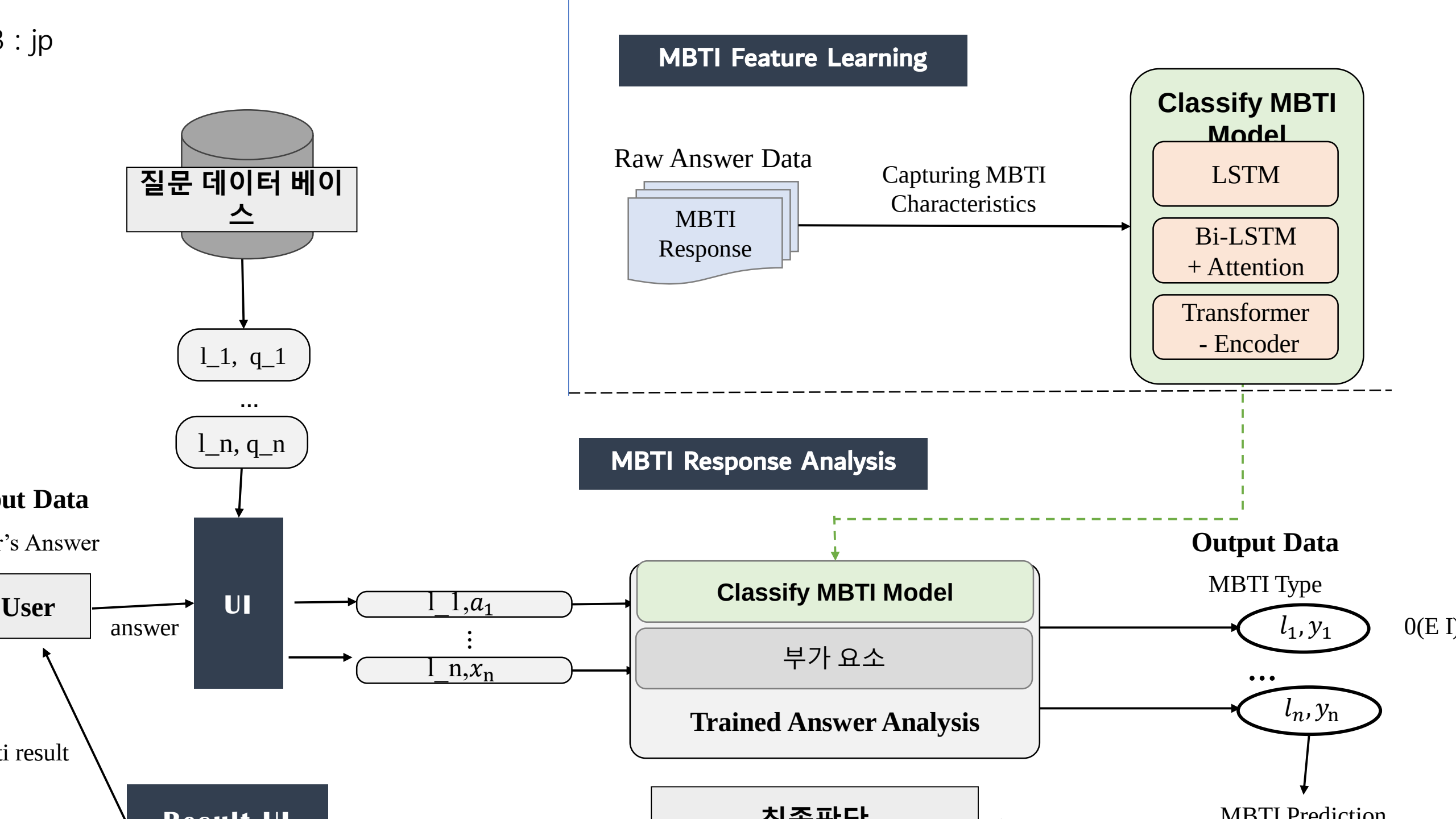
Modifier (adverbs)

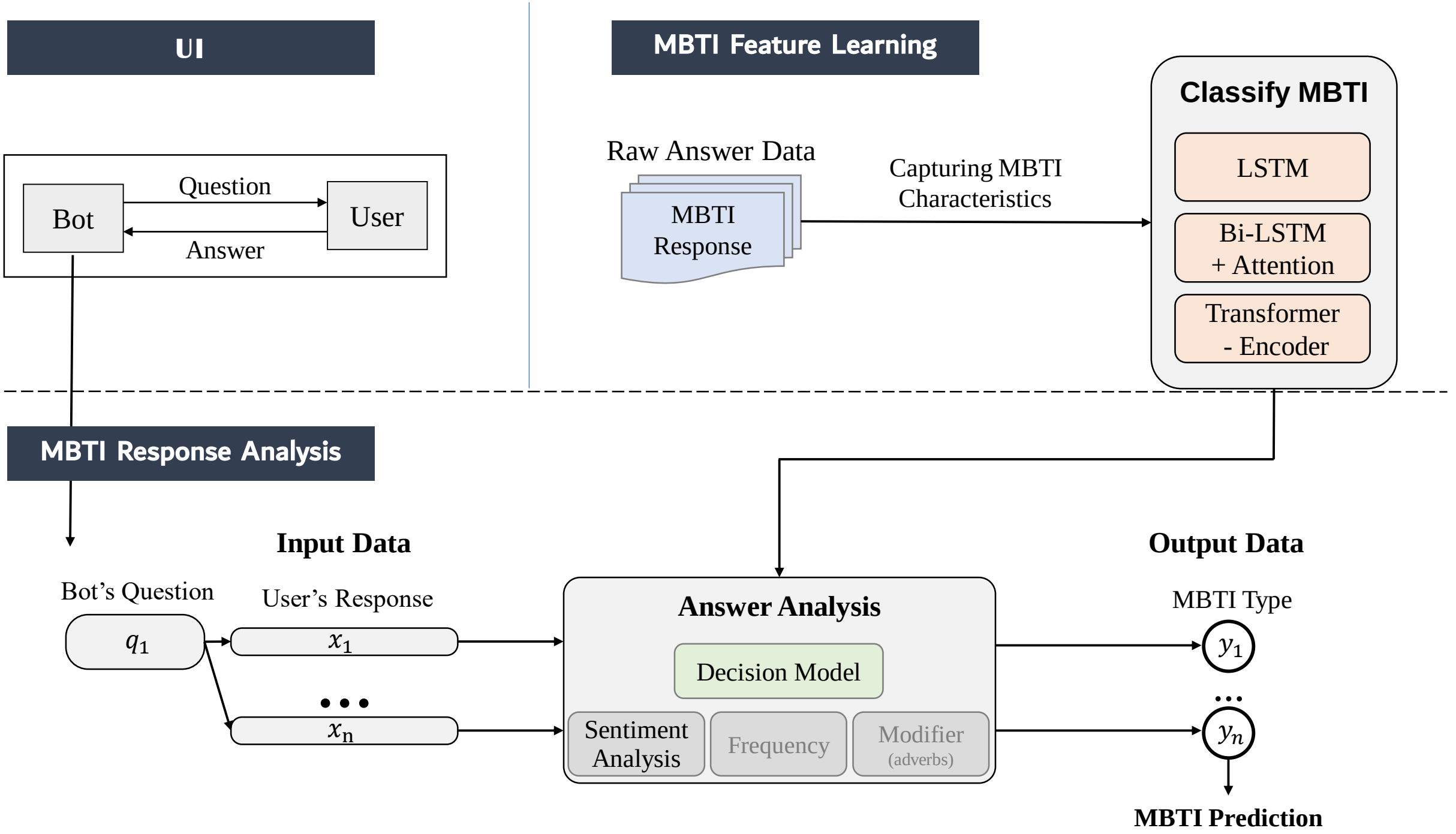
Output Data

MBTI Type

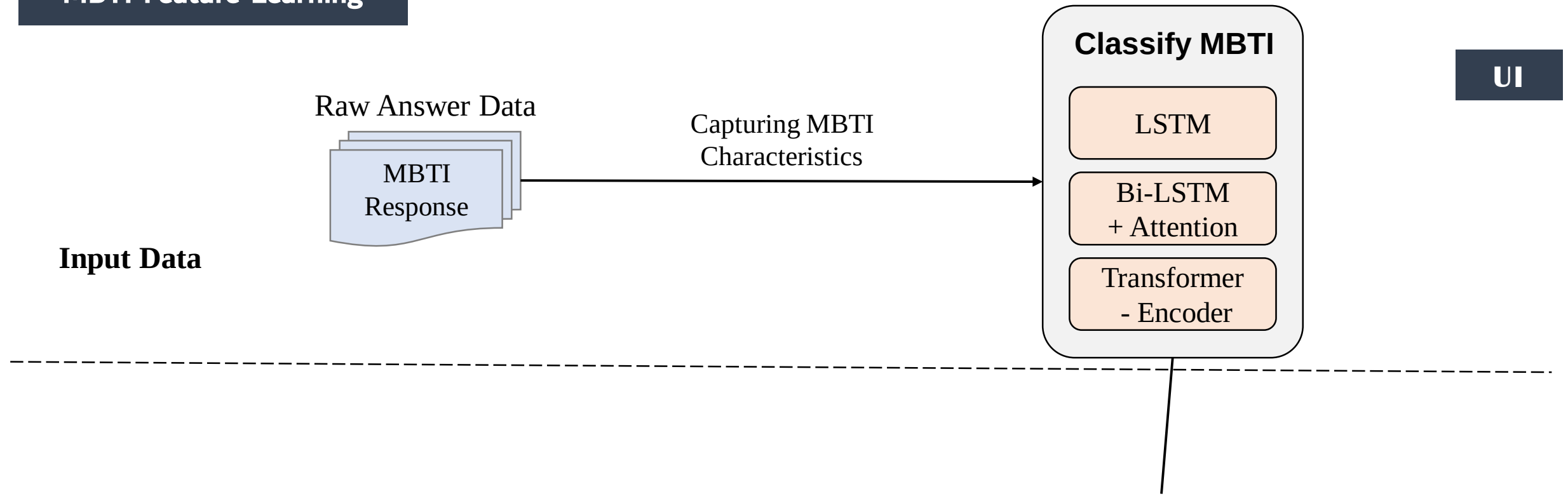

Dataset
write & create
Question Data
Answer Data
Label
E data
S data
T data
J data
I data
N data
F data
P data
E/I
S/N
T/F
J/P
KoGPT-2
EDA
Data Augmentation
KoNLPy
Text preprocessing
data = [Question data_Answer data_Label]
LSTM
Bi-LSTM + Attention
Transformer - Encoder
4-BINARY CLASSIFIERS
Model
E/I model
S/N model
T/F model
J/P model
Answer Analysis Model
KoNLPy
(Morpheme Analysis)
Word Embedding Vector values for each Key-word
Sentiment Analysis
LSTM
Frequency
Modifier
(adverbs)
UI
Bot
User
E/I Question
Answer Analysis Model
Identifying a single MBTI type
User's Answer
Output
: User's MBTI


Dataset
Question Data
E data
I data
Answer Data
E data
I data
Label
E/I
KoGPT-2
EDA
Data Augmentation
KoNLPy
Text preprocessing
data = [Question data_Answer data_Label]
LSTM
Bi-LSTM
4-BINARY CLASSIFIERS
Model
E/I model
S/N model
T/F model
J/P model


MBTI Feature Learning
Raw Answer Data
MBTI Response
Capturing MBTI Characteristics
Classify MBTI Model
LSTM
Bi-LSTM + Attention
Transformer - Encoder
질문 데이터 베이스
l_1, q_1
...
l_n, q_n
MBTI Response Analysis
ut Data
r's Answer
User
answer
UI
l_1,a_1
l_n,x_n
Classify MBTI Model
부가 요소
Trained Answer Analysis
Output Data
MBTI Type
l_1, y_1
l_n, y_n
0(E I
ti result
MBTI Prediction


UI
MBTI Feature Learning
Bot
Question
User
Answer
Raw Answer Data
MBTI Response
Capturing MBTI Characteristics
Classify MBTI
LSTM
Bi-LSTM + Attention
Transformer - Encoder
MBTI Response Analysis
Input Data
Bot's Question
User's Response
q_1
x_1
x_n
Answer Analysis
Decision Model
Sentiment Analysis
Frequency
Modifier (adverbs)
Output Data
MBTI Type
y_1
y_n
MBTI Prediction

MBTI Feature Learning



UI

색깔 조합

질문!!!!

그러면 좀 큰 틀의 구조도를 보여주고

저 위에 만들어 놓은거 각부분 설명할때 써도 되나?

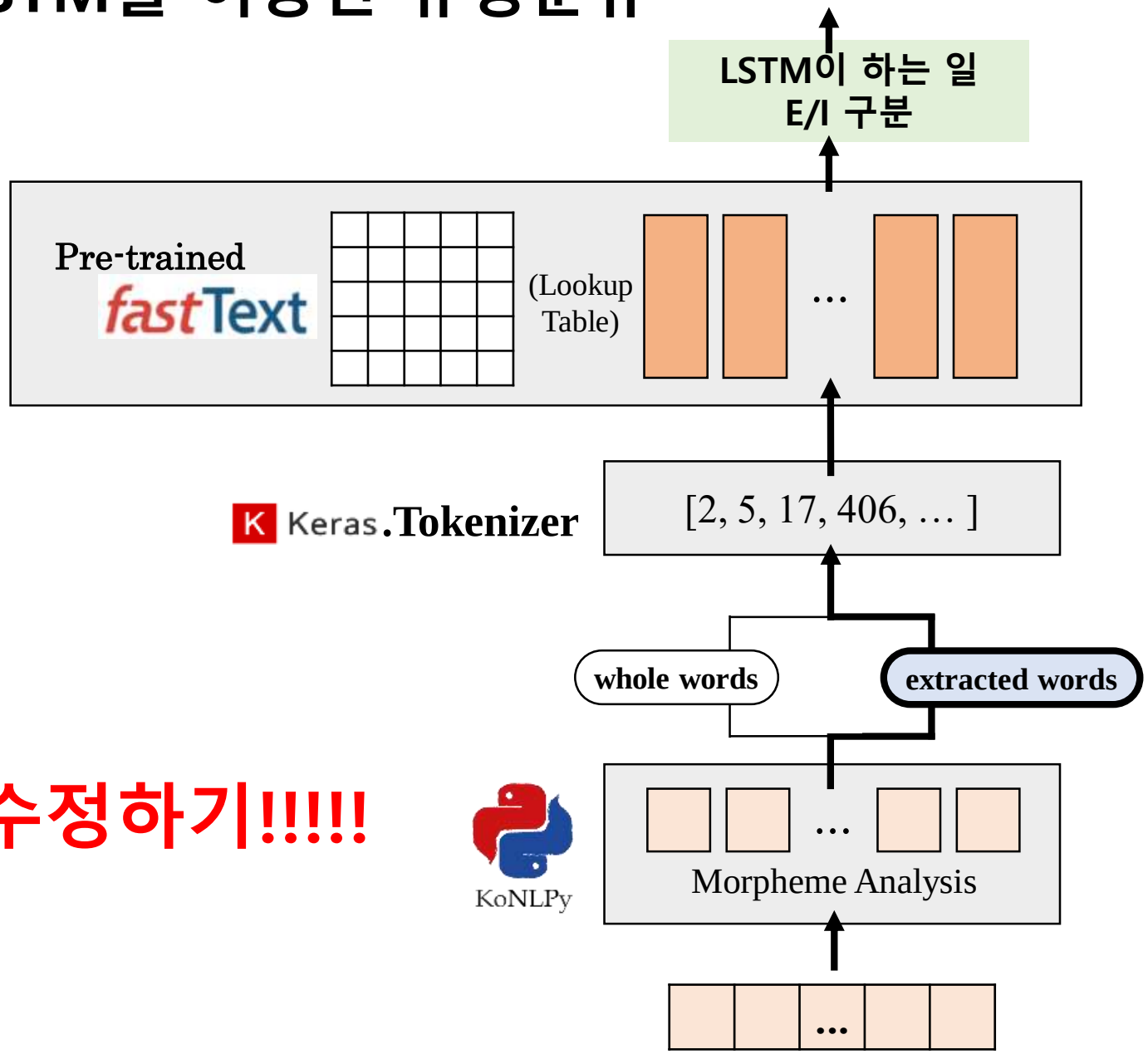
++ ??? 우키오빠한테 물어볼거!!
그럼 개괄식으로 이렇게 작성하면 세부적인 부분은 어떻게 설명을 하는가?

[교수님 질문]
fasttext 의 입력은 word일텐데 나는 정수 인코딩을 넣어버리면 ,, 그게 정확히 사전학습된 모델의 벡터를 잘 가져오는 게 맞는것인가?

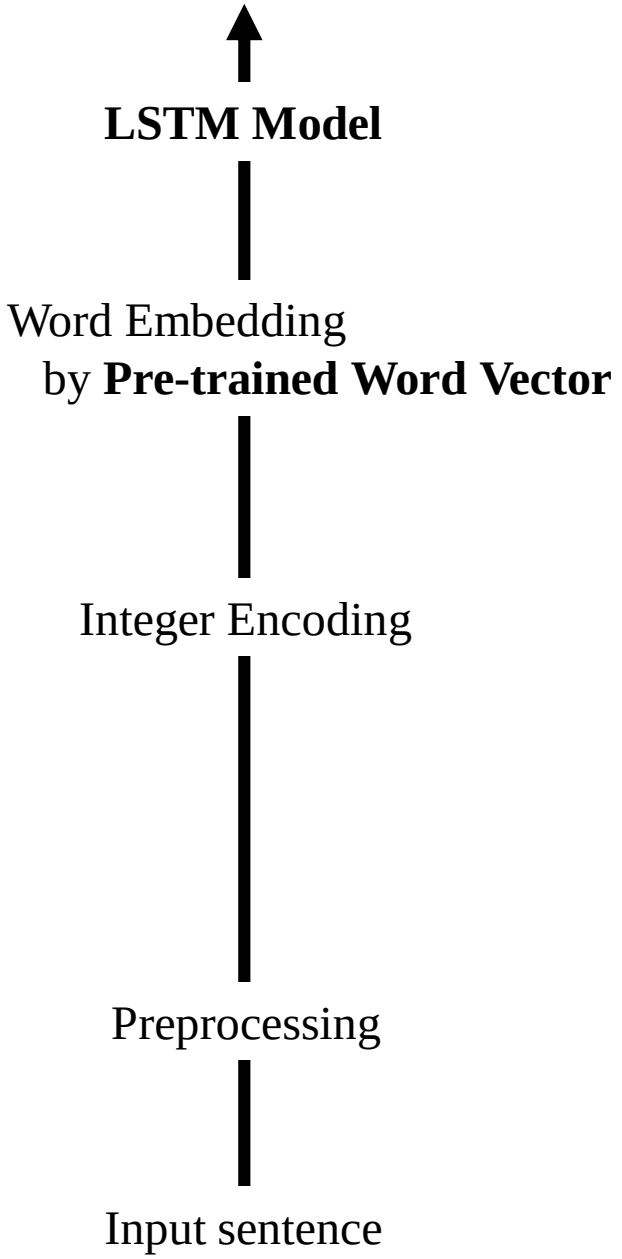
2. 구조도 수정
더 개괄식으로 작성하기

3. Weighted Sum
: 가중치의 곱을 더하여
일정 비율만큼 각 모델들을 사용해주는 것

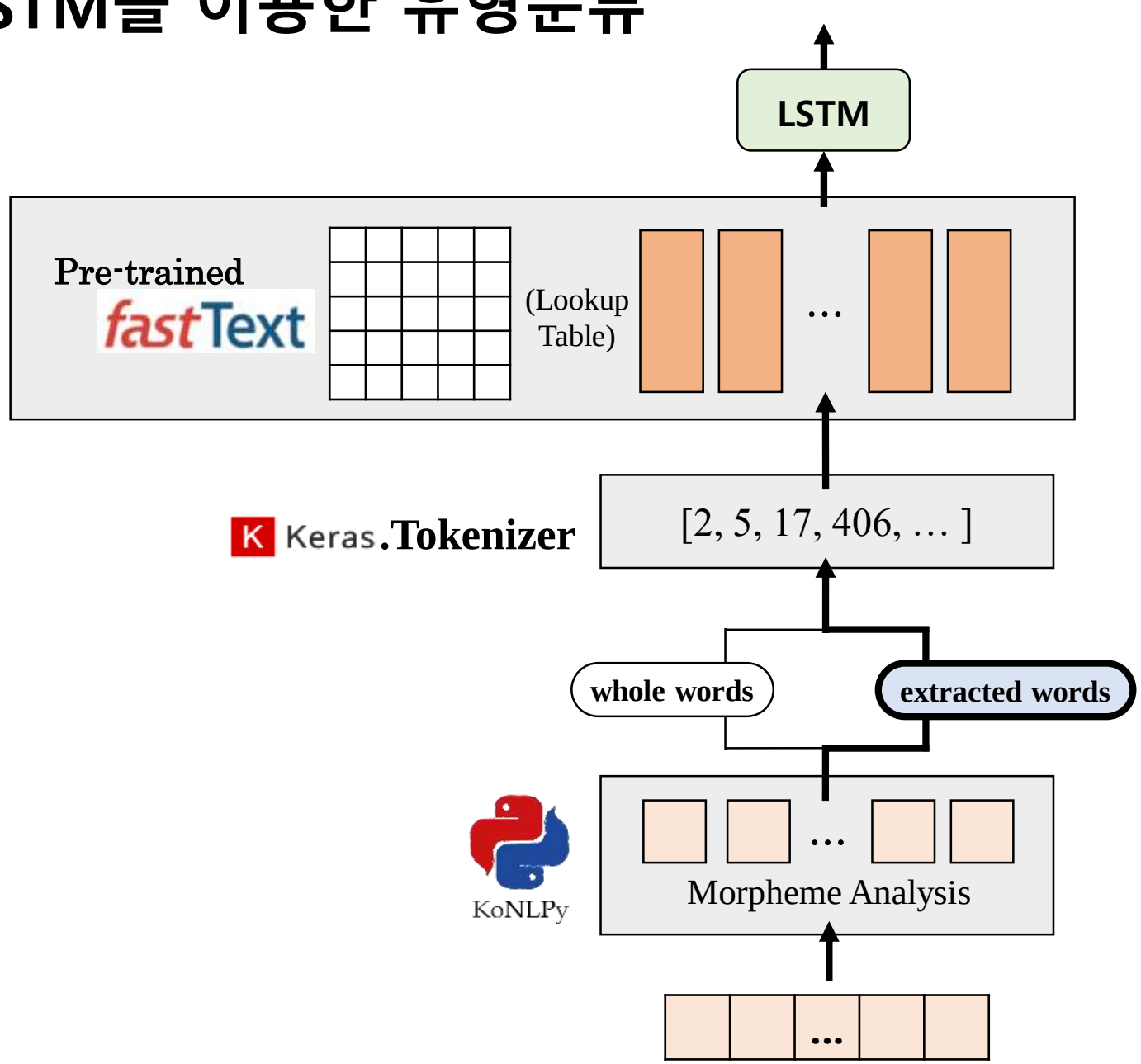
LSTM을 이용한 유형분류

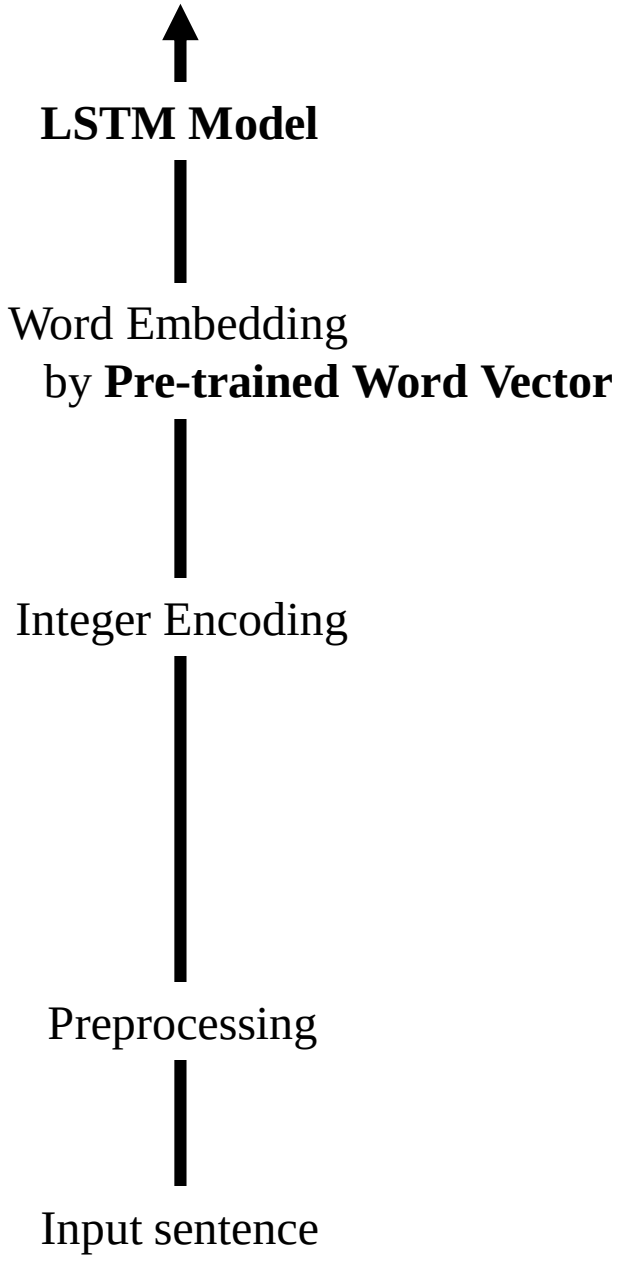


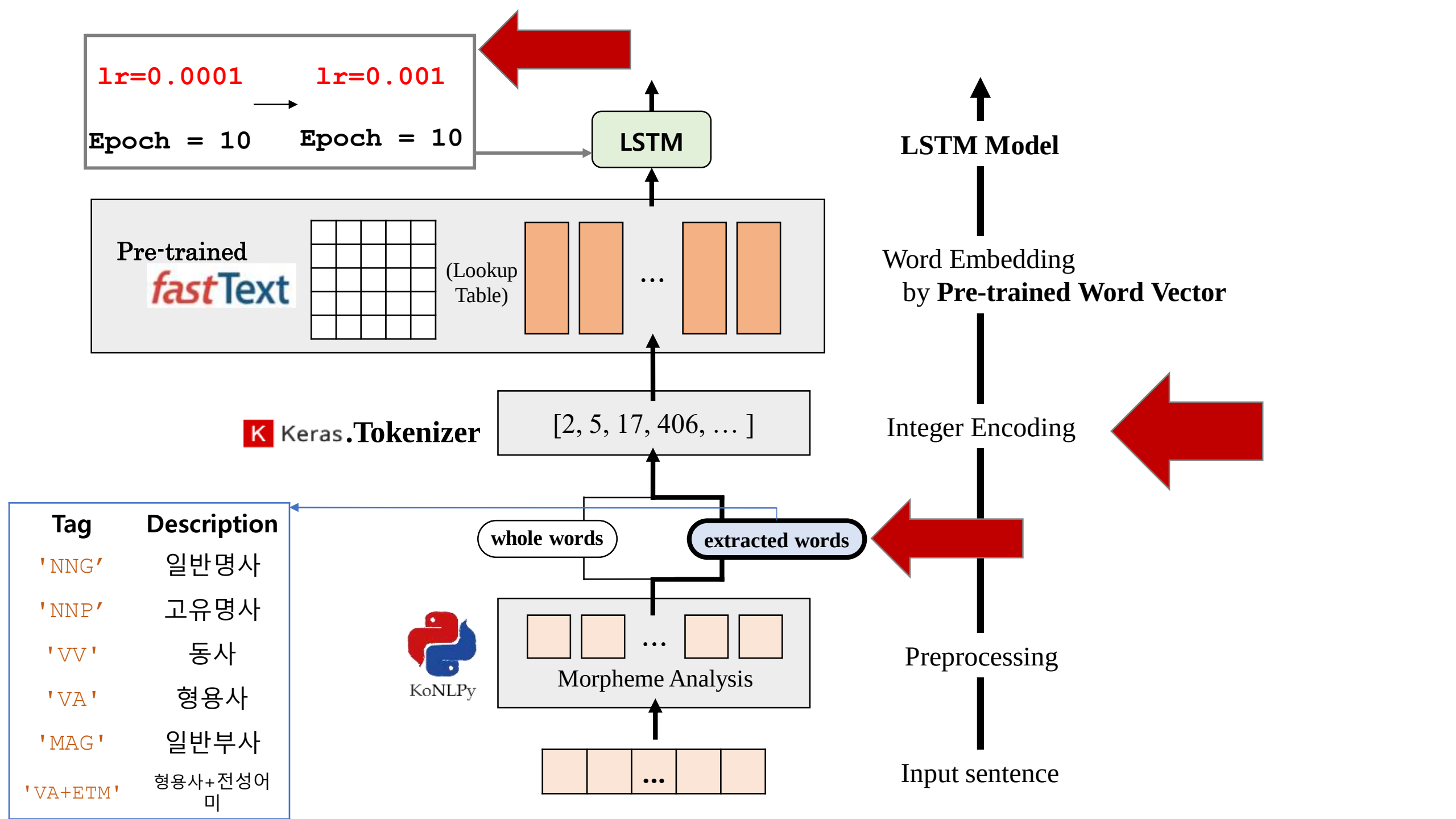
수정하기!!!!!



LSTM을 이용한 유형분류



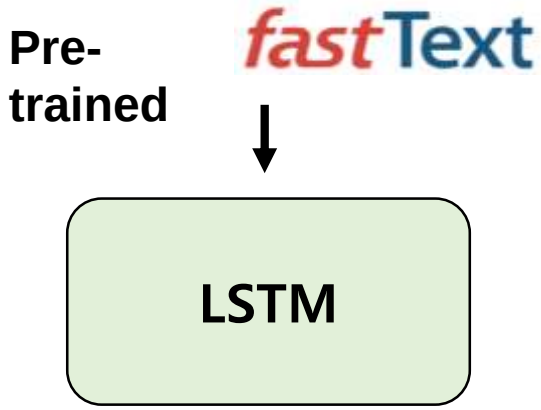



lr=0.0001 → lr=0.001
Epoch = 10 Epoch = 10
LSTM
LSTM Model
Pre-trained
fastText
(Lookup Table)
Word Embedding by Pre-trained Word Vector
Keras.Tokenizer
[2, 5, 17, 406, …]
Integer Encoding
whole words
extracted words
KoNLPy
Morpheme Analysis
Preprocessing
Input sentence
Tag Description
'NNG' 일반명사
'NNP' 고유명사
'VV' 동사
'VA' 형용사
'MAG' 일반부사
'VA+ETM' 형용사+전성어미

성능 비교

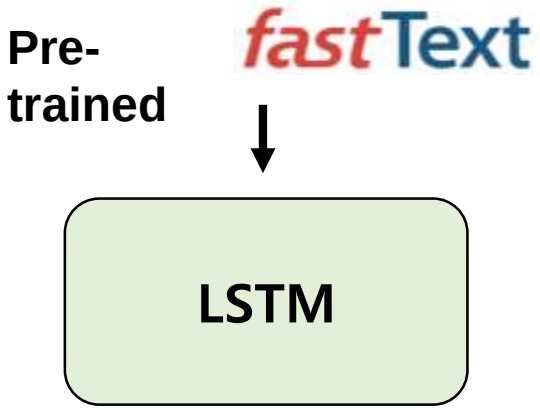
1.

Data	약 1000개
Embedding vector	사전학습 모델
학습 모델	LSTM



2.

Data + EDA	약 3000개
Embedding vector	사전학습 모델
학습 모델	LSTM



(EDA Data)

EDA 기법 적용

num_aug = 3

유의어 교체	랜덤 삽입	**랜덤 교체**	**랜덤 삭제**
alpha_sr = 0	alpha_ri = 0	**alpha_rs = 0.1**	**alpha_rd = 0.1**

sentence = 새로운 사람을 만나면 항상 두근거린다

새로운 사람을 만나면 항상 두근거린다
새로운 만나면 항상

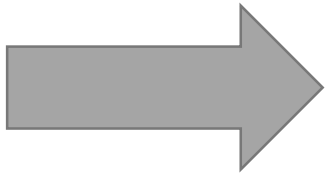
sentence = 대부분의 경우 먼저 말을 걸려고 해요

대부분의 경우 먼저 말을 걸려고 해요
대부분의 우 먼저 말을 걸려고 해요
먼저 경우 대부분의 말을 걸려고 해요

EDA 기법 적용

기존 data 개수

개수: 1,111

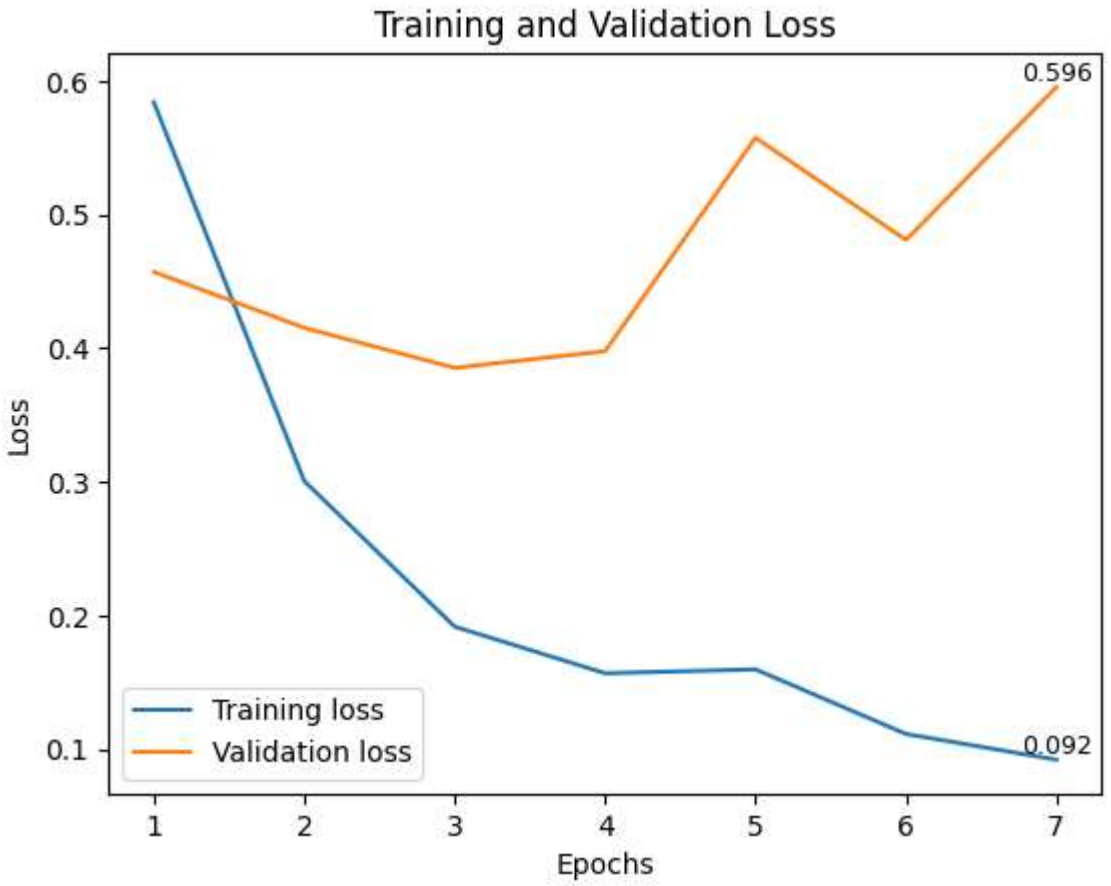


EDA 적용

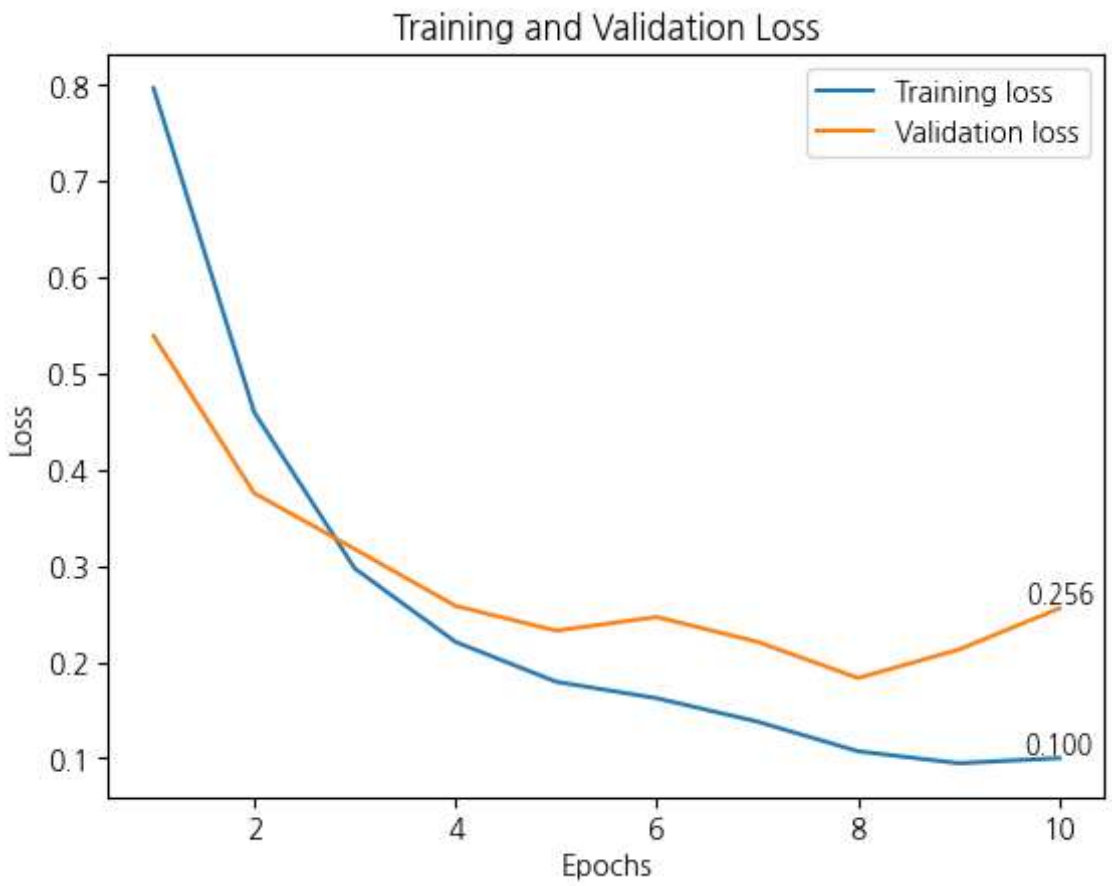
개수: 3,155

Loss

Base Data

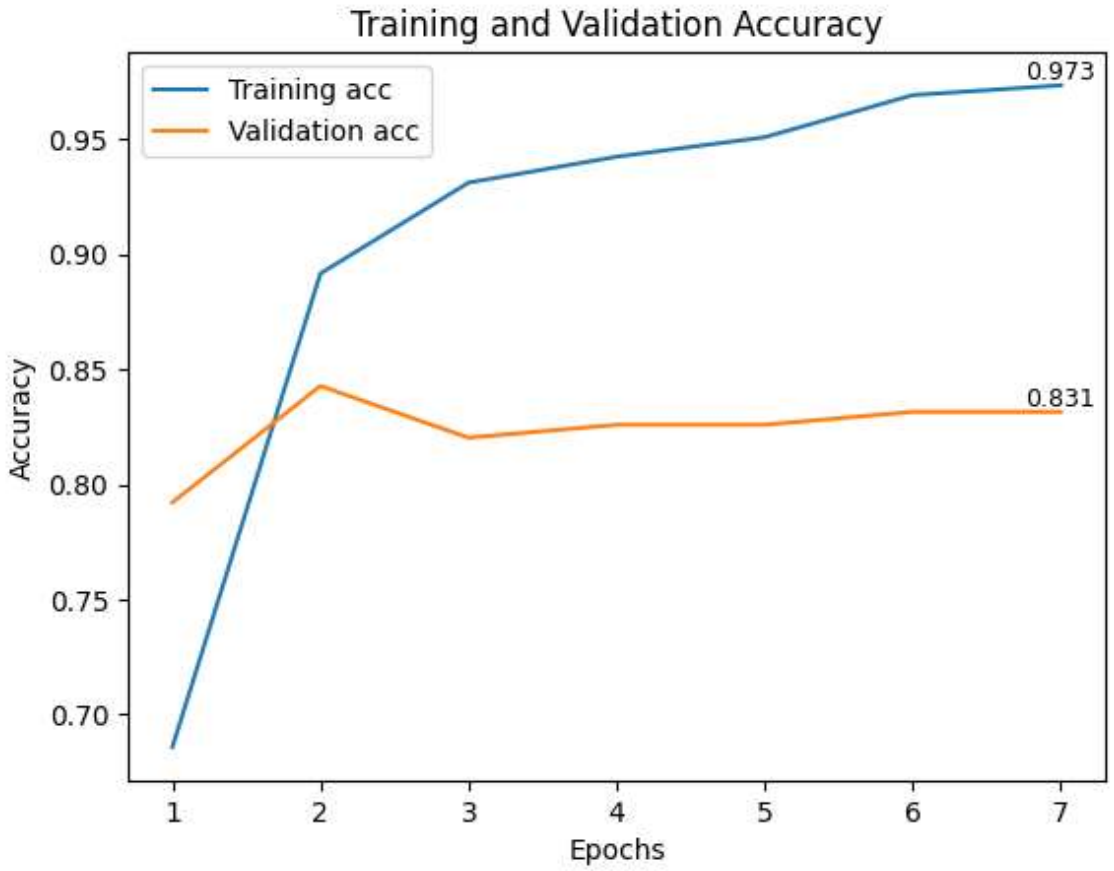


Base Data
+ EDA

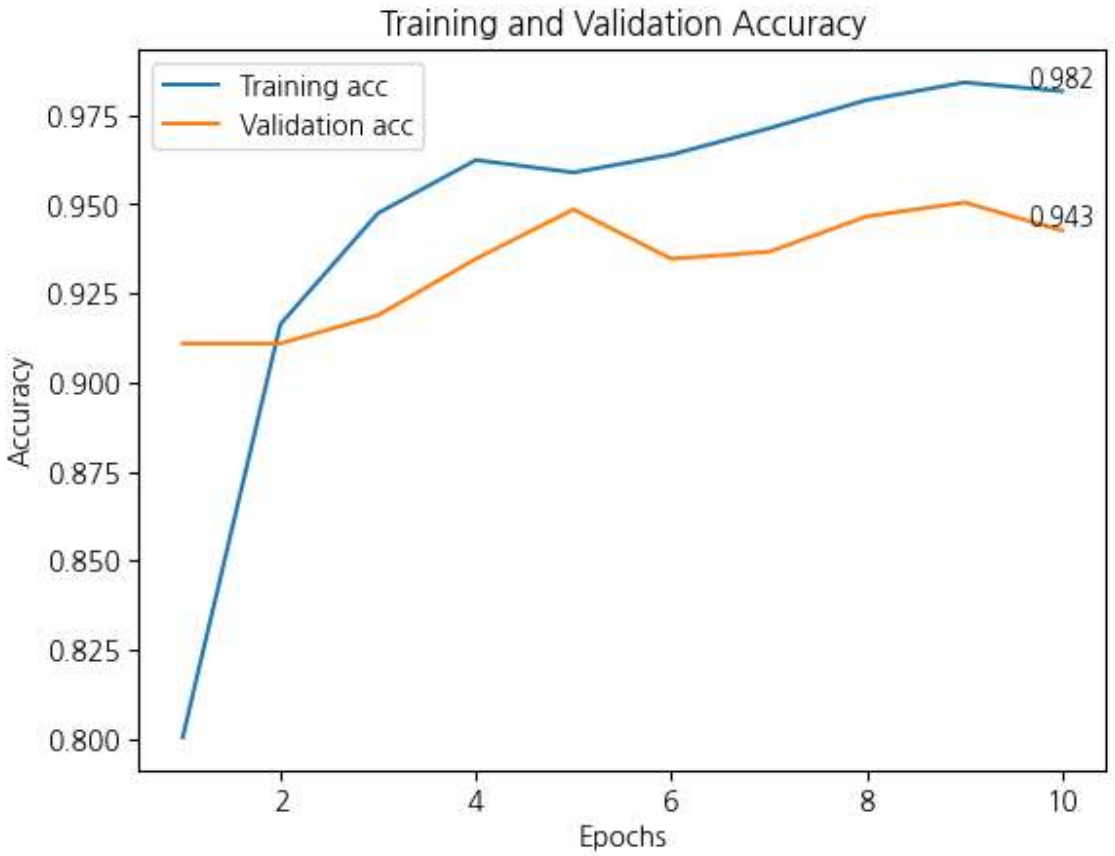


Acc

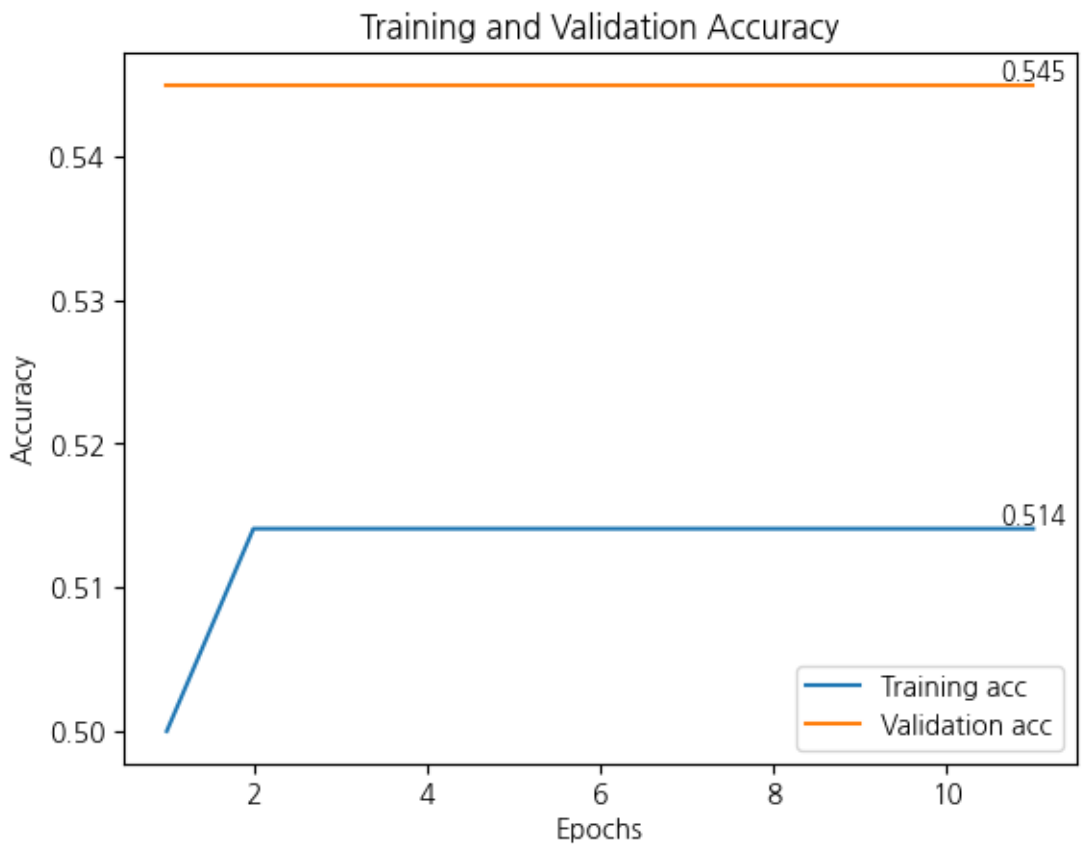
Base Data



Base Data
+ EDA



lr 미조절

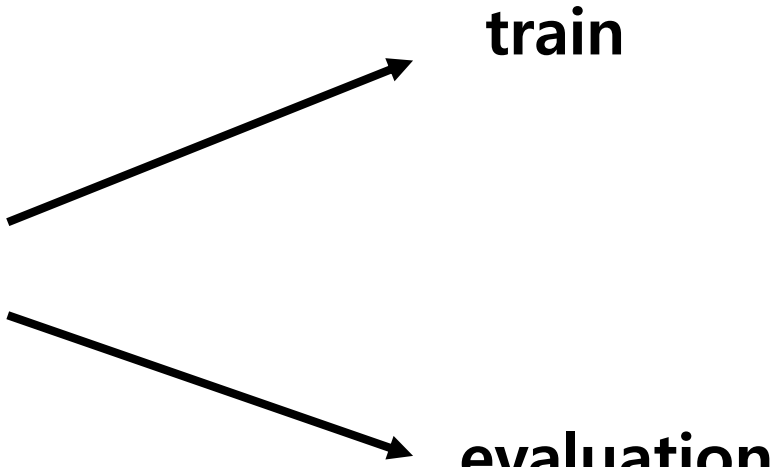

Training and Validation Accuracy
0.545
0.514
0.54
0.53
0.52
0.51
0.50
Accuracy
2
4
6
8
10
Epochs
Training acc
Validation acc


Training and Validation Loss
0.6940
0.6935
0.6930
0.6925
0.6920
0.6915
0.6910
Loss
0.693
0.691
2
4
6
8
10
Epochs
Training loss
Validation loss

높은 성능의 이유?

1. Extract words
   (명사, 어간, 동사, 형용사 등)

로



ex) 좋다, 좋아, 좋아하다, 좋고 -> “좋(다)”

나는 사람 만나는 것을 좋아한다.

~~2. 사전 학습된 모델 사용~~

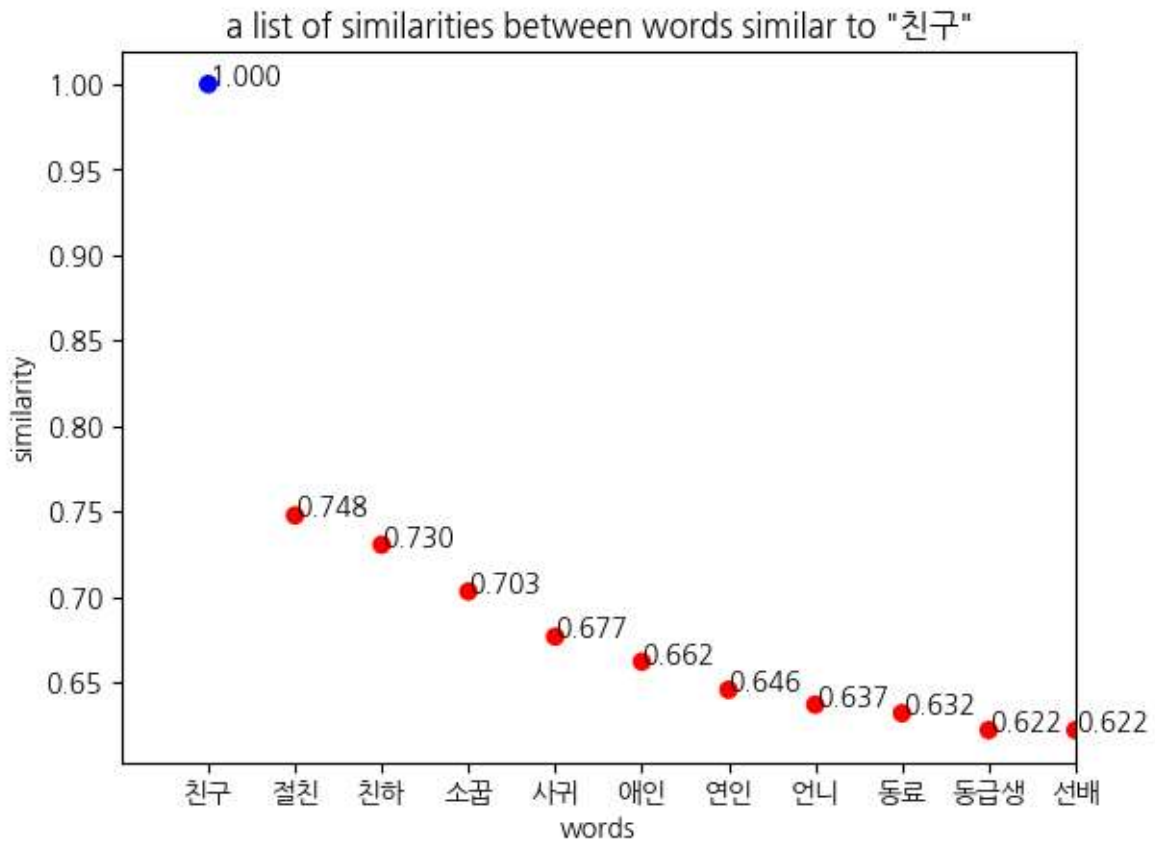
3. EDA 기법 때문..

문장 A –(EDA기법 적용)-> A1, A2, A3
A1-train에 들어감
A2-test에 들어감
A3- validation에 들어감..

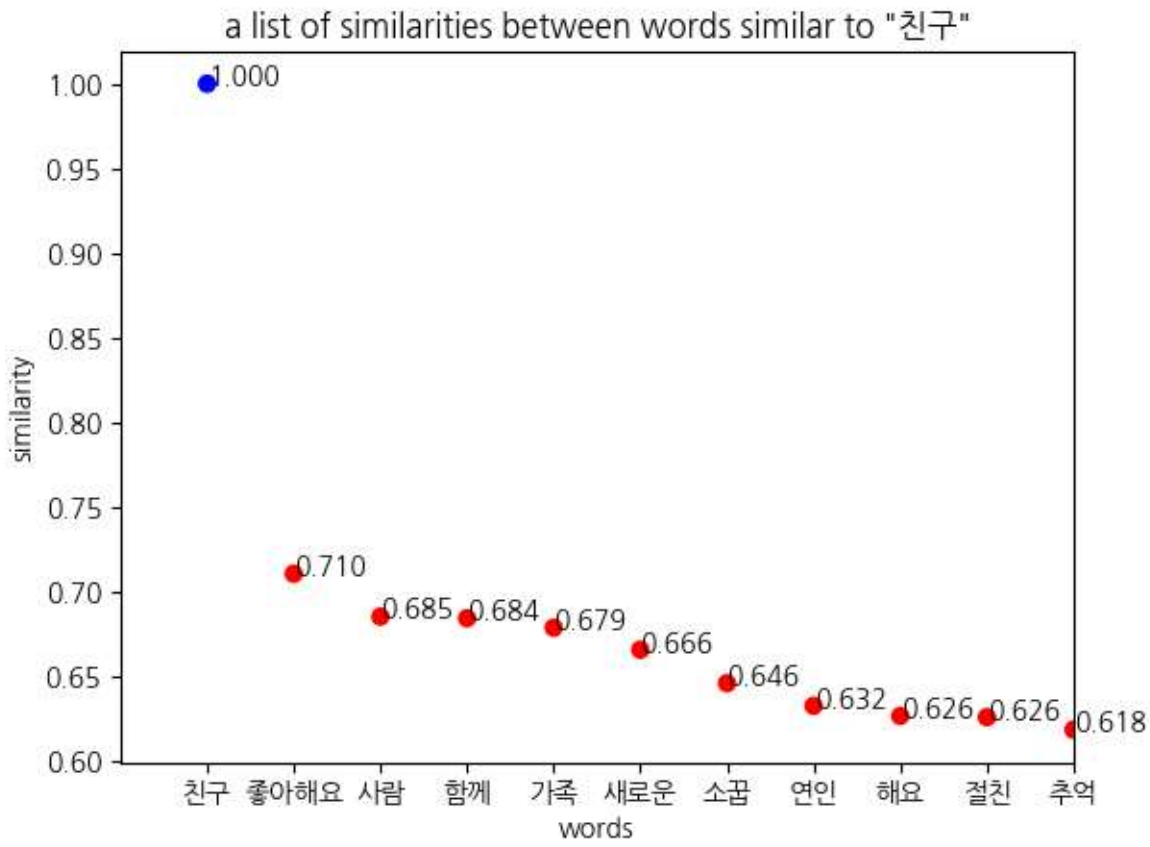
그럼 비슷한 문장이 거의 들어간거니까
아무래도 성능이 좋을 수 밖에..

Similarity

Pre-trained FastText - KR

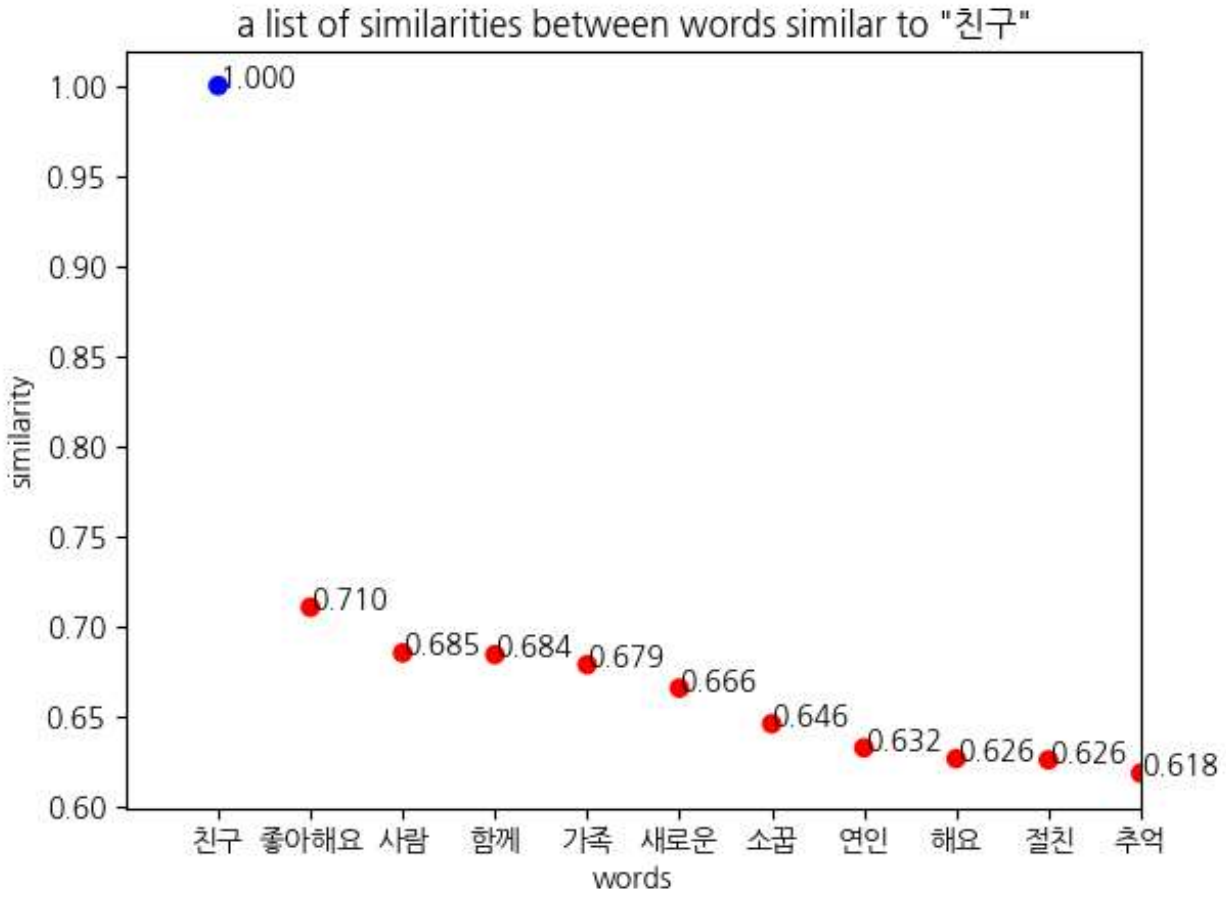


new_FastText model

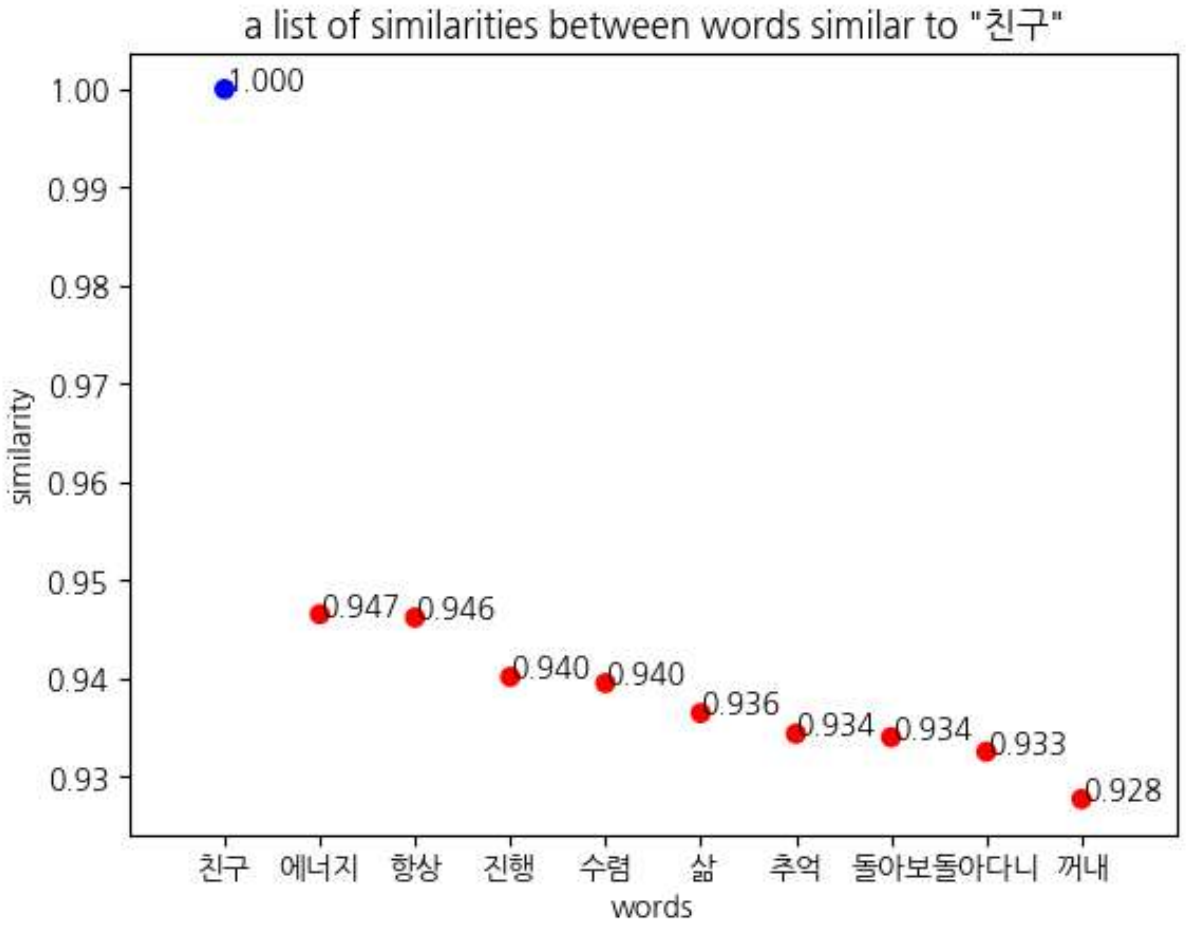


Similarity

09.05



09.19



Prediction for **a single word**

어느정도 예측이 나오기는 하는데
거의 고정된 예측만 나옴!

단일 단어이기 때문에 40~50사이의 어중간한 값
나오는 듯 함!

```
Input: 친구
Predicted label: 0.5188979506492615
  51.89% 정도로 E입니다.
```

```
Input: 혼자
Predicted label: 0.4132554233074188
  40.33% 정도로 I입니다.
```

Prediction for **a single word**

단일 토큰에 대한 예측 어려움...

-> LSTM 모델은 문장 전체의 정보를 사용하여 학습함

따라서 단일 토큰만 제공된다면
해당 **토큰의 문맥적, 구조적 정보가 부족하여**
모델의 예측이 어려워짐

Prediction for **a sentence**

문장입력

```
입력문장: <그래도 주말에는 집에 있는 게 좋아요..>
```

out of voca 그래도 에 는 에 는 아요

토큰화

```
OOV 토큰 확인: OOV 주말 OOV OOV 집 OOV 있 OOV 게 좋 OOV
```

정수 인코딩

```
[[1, 32, 1, 1, 24, 1, 4, 1, 641, 9, 1]]
```

모델 예측값

```
예측값 :  0.02986094
```

MBTI 판별

```
>> 97.01% 정도로 I 유형입니다.
```

pre ≥ 0.5 -> E
pre < 0.5 -> I

Prediction for **a sentence**

```
예측값 :  0.996568

입력문장: <생일에는 주변 친구들과 신나게 파티를 열어서 놀아요~>
   >> 99.66% 정도로 E 유형입니다.
```

```
OOV 토큰 확인: 생일 OOV OOV 주변 친구 들 OOV 신나 게 파티 OOV 열 OOV 놀 OOV
[[234, 1, 1, 87, 14, 61, 1, 421, 641, 83, 1, 275, 1, 198, 1]]
```

Prediction for a sentence – OOV 문제

FastText 성질 없어짐
-> oov처리 x

```
예측값 :  0.12785016

입력문장: <엠티에서는 그 누구보다도 앞장선다.>
   >> 87.21% 정도로 1 유형입니다.
```

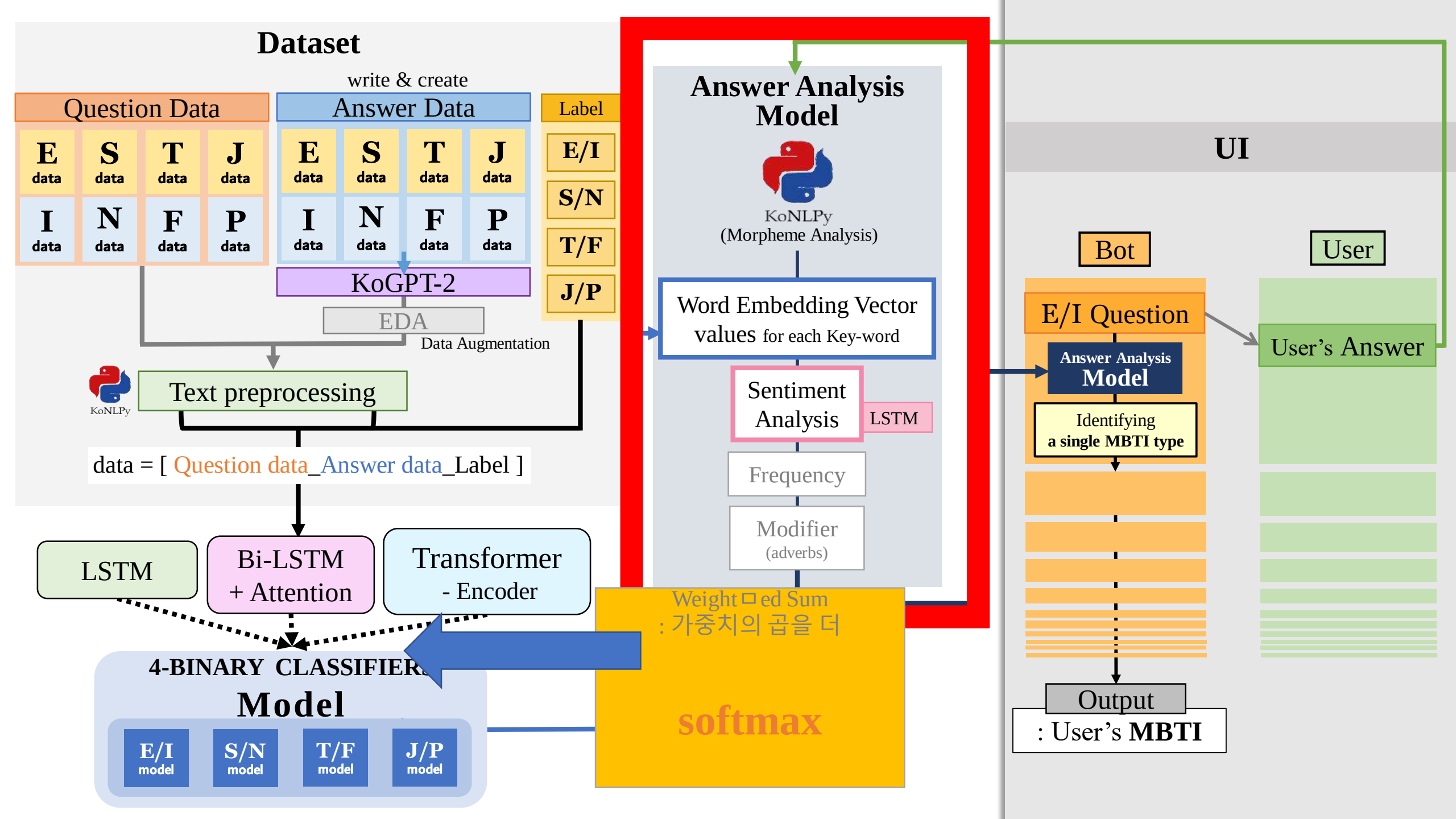
```
OOV 토큰 확인: OOV OOV OOV OOV OOV OOV OOV OOV
[[1, 1, 1, 1, 1, 1, 1, 1]]
```

훈련 데이터의 OOV 비율에
따른 예측 값

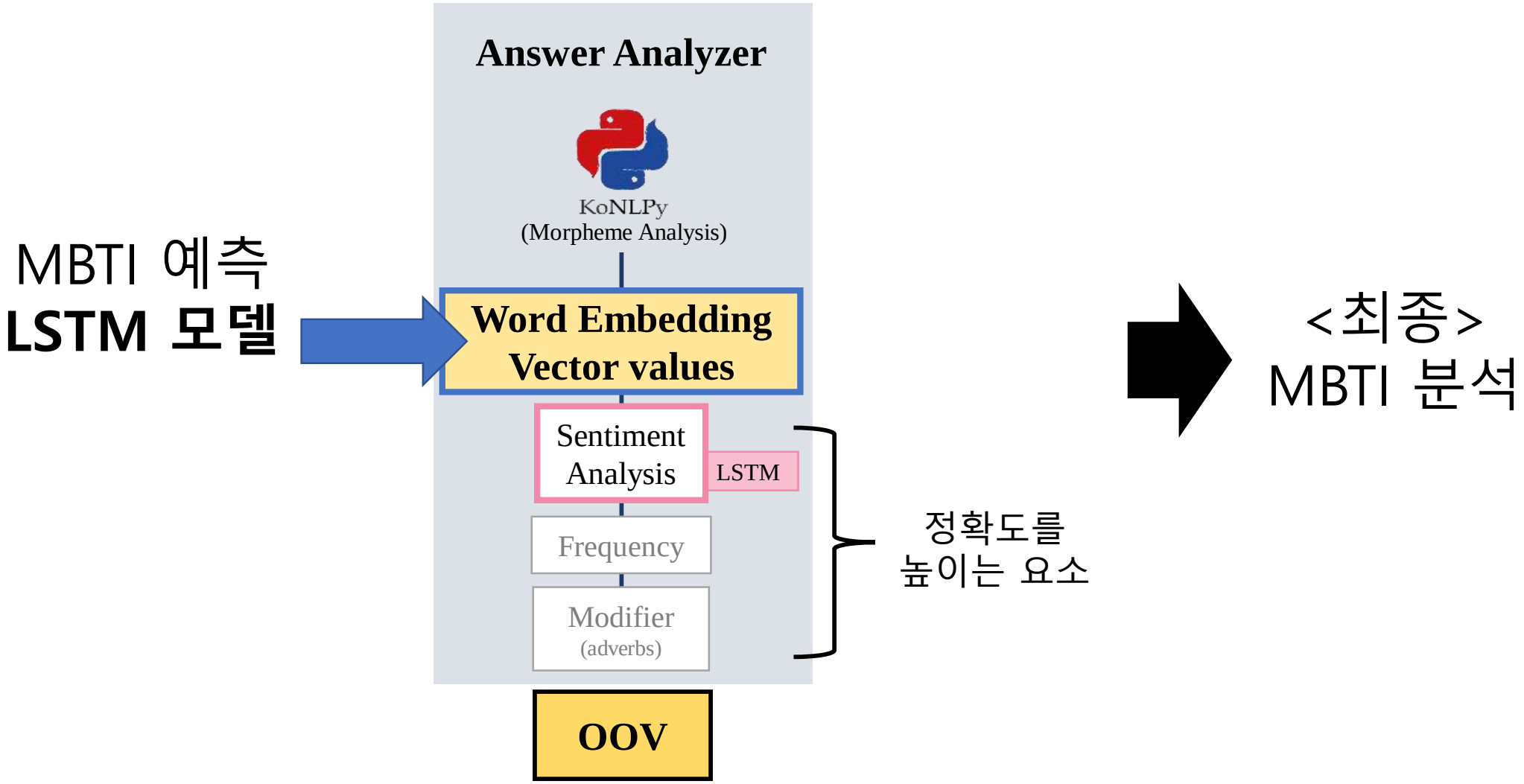
Prediction for a sentence (using extracted words)

```
입력문장: <생일에도 집에서 혼자 시간을 보내곤 해요.>
   >> 98.84% 정도로 I 유형입니다.

OOV 토큰 확인: 생일 집 혼자 시간 보내
[[222, 24, 8, 6, 42]]
```

```
입력문장: <먼저 말을 거는 편이다.>
   >> 98.64% 정도로 E 유형입니다.

OOV 토큰 확인: 먼저 말 OOV
[[20, 49, 1]]
```


Dataset
write & create
Question Data
Answer Data
Label
E data
S data
T data
J data
I data
N data
F data
P data
E/I
S/N
T/F
J/P
KoGPT-2
EDA
Data Augmentation
KoNLPy
Text preprocessing
data = [Question data_Answer data_Label]
LSTM
Bi-LSTM + Attention
Transformer - Encoder
4-BINARY CLASSIFIER
Model
E/I model
S/N model
T/F model
J/P model
Answer Analysis Model
KoNLPy
(Morpheme Analysis)
Word Embedding Vector values for each Key-word
Sentiment Analysis
LSTM
Frequency
Modifier
(adverbs)
Weighted Sum
: 가중치의 곱을 더
softmax
UI
Bot
User
E/I Question
Answer Analysis Model
Identifying a single MBTI type
User's Answer
Output
: User's MBTI

Answer Analyzer



Answer Analyzer – **OOV**

입력 문장 = "나는 친구를 만날 때 가장 행복하다."

형태소 단위로 토큰분리

문장

=

OOV

+

OOV가 아닌
일반 토큰

1

제거

Tokenizer

양의 정수로 인코딩

2

pre-trained
FastText로 벡터화

고차원 벡터

Tokenizer

양의 정수로 인코딩

3

pre-trained
FastText로 벡터화

고차원 벡터

pre-trained
FastText로 벡터화

고차원 벡터

4

avg 적용

Tokenizer

양의 정수로 인코딩

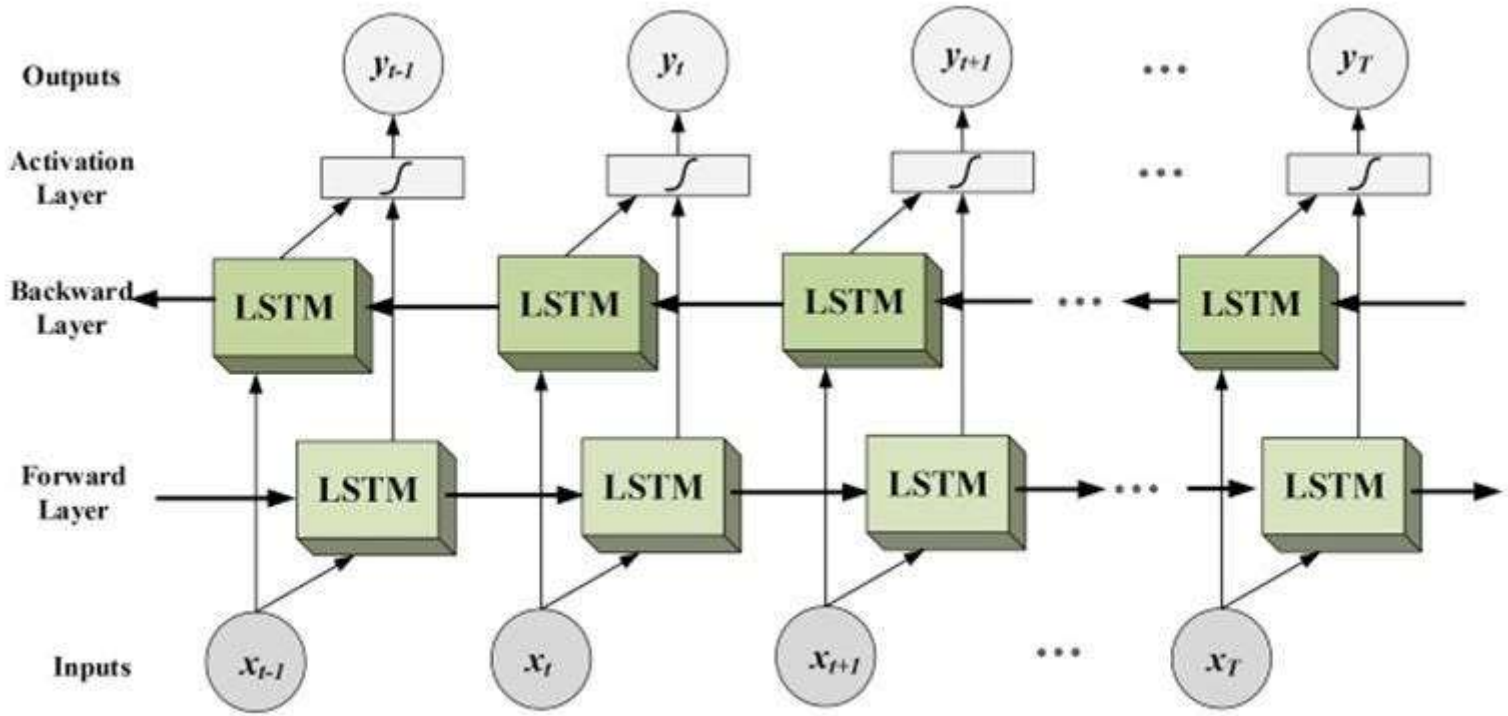
LSTM 모델 학습의 한계

1. 문맥 학습의 한계 -> 예측이 어느정도만 되고 그렇게 잘 되지는 않는다!

2. EDA 기법의 분산 문제 미해결
(문장과 문장 aug는 (train,val,test)중 같은 곳에 위치해야한다!

3.

Bi-LSTM


Outputs
Activation Layer
Backward Layer
Forward Layer
Inputs
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM