2022 K-ium 의료 인공지능 경진대회

급성 허혈성 뇌졸중 판독문 분류

참가접수
9월 19일(월) ~ 10월 30일(일)

1차 평가
10월 31일(월) ~ 11월 4일(금)

2차 접수
11월 5일(토) ~ 11월 13일(일)

2차 평가
11월 14일(월) ~ 11월 18일(금)

최종발표
11월 23일(수)

주관 기관에서 제공한 데이터를 바탕으로 모델 구현

	Findings	Conclusion	AcuteInfarction
2	Clinical information : 두부외상 후 후유증 평가 Axial T1WI, sagittal T1WI, axial T2WI, axial FLAIR, axial T2* GRE image 획득하였으며 조영증강은 시행하지 않았음.	1. Encephalomalacic change in both frontal lobes, left temporal lobe. 2. Old infarctions at both BG. 3. Microangiopathy. 4. Microbleeds in both BG, thalami and pons. 5. Right maxillary sinusitis.	0
3	Clinical information : lung cancer Axial T1WI, sagittal T1WI, axial T2WI, axial FLAIR, axial T2* GRE image 획득하였으며 조영증강을 시행함.	1. No change of focal enhancing lesion in left cerebellum. --> Metastasis. 2. Nonvisualization of enhancing lesion in left parietal bone on MR	0
4	Clinical information : Multiple Sclerosis Axial T1WI, sagittal T1WI, axial T2WI, axial FLAIR, axial T2* GRE image 획득하였으며 조영증강을 시행함.	No significant interval change of abnormal hyperintense lesion on T2WI and FLAIR - both periventricular white matter, corona radiata, pons.	0
5	Clinical information : patient with DLBCL. Axial T1WI, sagittal T1WI, axial T2WI, axial FLAIR, axial T2* GRE image 획득하였으며 조영증강을 시행함.	1. Decreased extent of enhancing mass in the left BG. -> Lymphoma involvement 2. ICH at left basal ganglia and corona radiata. 3. No change of others.	0

- **급성 허혈성 뇌졸중 판독문 (= 8843)**
  1차 제출용으로 제공된 데이터 = 6190
- **한글 + 영어 혼용 자연어 문장 데이터**
- **카테고리 형 분류 (뇌졸중 여부 = [1, 0])**

★ **Findings : 영상 소견 (MRA, MRI 등)**

★ **Conclusion : 진단 소견 요약**

★ **Acute Infarction : 여부 판단 (1 = 존재, 0 = 없음)**

※ 과제 수행 과정

1. 원본 데이터 내용 확인 및 처리에 문제가 발생할 수 있는 요소 확인.

Cl, Lt. breast cancer, under ctx, follow-up, s/p GKRS.
Axial T1WI, sagittal T1WI, axial T2WI, axial FLAIR, axial T2* GRE image 획득하였으며 조영증강을 시행함.

1. Increased size of a metastasis at the Rt. parietal lobe.
   - 24mm --> 34mm.
   --> probable tumor progression (DDx. radiation-induced change).
Rec) MR perfusion.

2. No significant change in size of hemorrhagic metastasis at midbrain
-- with slightly decreased extent of enhancing portion.

3. No definite new enhancing lesions.

- **여러 의학용어들이 축약된 표현으로 사용** (Cl, s/p, f/u 등).

Ex.) follow-up과 f/u를 혼용하면 두 내용이 같음을 판단시켜야 하는 문제

- **다양한 종류의 항목 번호 사용** (1. 2. 1) 2)).

Ex.) 항목 번호는 뇌경색 유무 판별에 중요한 역할을 하지 않음.

- **문장 내 특수기호 포함** (- <> () 등).

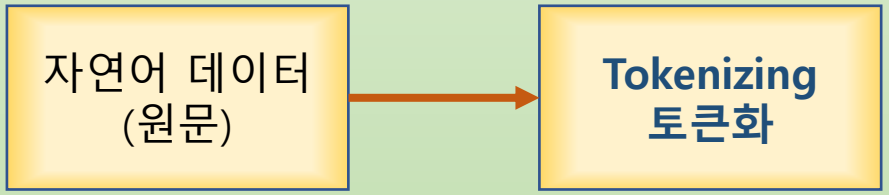
Ex.) 내용 강조(<MRA>), 항목 구분(-, --),
특정 의미 함축(24mm --> 34mm : 24mm에서 34mm로 길이 변화)
등 불특정한 역할로 사용.

- **아직 발견하지 못한 잠재적 문제 존재.**

※ 과제 수행 과정

2. 학습 모델을 만들기 위한 자연어 전처리 과정.

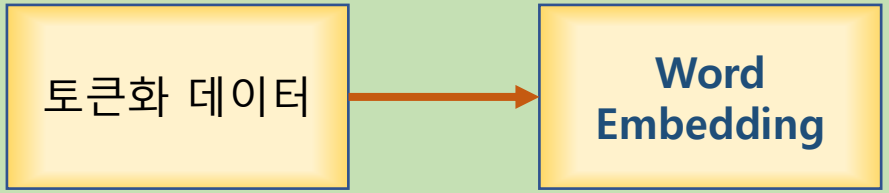
- 인공지능은 자연어를 그대로 읽어서 처리하는 고도화된 능력을 가지진 않는다.
- 자연어 문장 구성에 대해 특정 의미를 부여할 수 있도록 '수치화'하는 과정이 필요하다.



Normal brain MRI. → ['Normal', 'brain', 'MRI', '.']

영문의 경우 높은 확률로 띄어쓰기 기준 구분 용이.

★ 토큰화 과정에서 의미 있는 문장/단어 구성을 위해 불필요한 글자나 분석 목적에 어긋나는 내용 처리.



['Normal', 'brain', 'MRI', '.'] → **2153 1550 6620**

Normal = [1.3, 0.2, 0.11 ...]

brain = [0.3, 0.7, 0.13 ...]

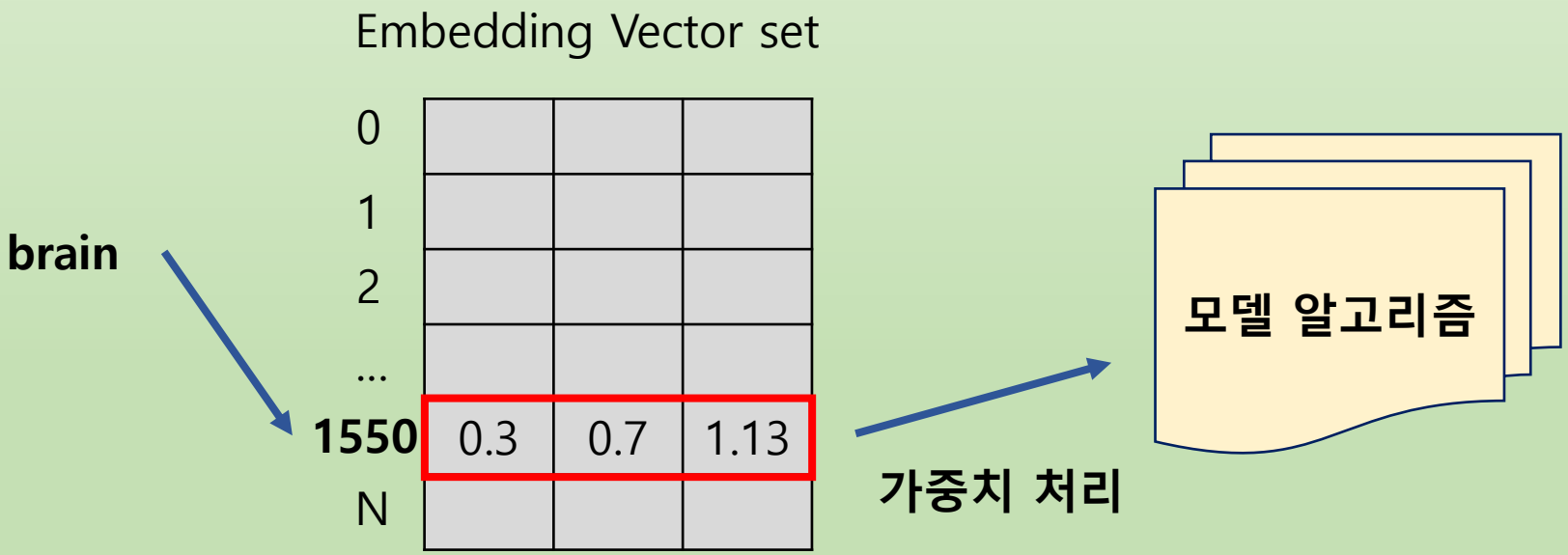
MRI = [0.89, 1.2, 0.5 ...]

각각의 토큰을 의미를 가지는 밀집 벡터 구조로 표현.

※ 과제 수행 과정

3. 전처리된 데이터를 활용 (사전 모델, 학습 모델 등).

- 최종적으로 만들어진 Embedding Vector 리스트를 이용해 머신러닝, 딥러닝 등의 방법 적용.



- 토큰으로 분리된 단어/문자는 특정한 정수 인덱스를 가지며 고유해야 한다.
  (brain은 1550과 같으며 1개만 존재)

- Word Embedding 과정에서 정수 **1550**에 대한 실수형 Vector 리스트를 생성한다.

※ 토큰화 예시

CI, F/U for cerebral metastases, s/p GKRS. Axial T1WI, sagittal T1WI, axial T2WI, axial FLAIR, axial T2* GRE image 획득하였으며 조영증강을 시행함. Brain, CSF space, and related findings Multiple cerebral metastases. Rt frontal lobe 6 lesions A. Middle frontal gyrus 9 mm 8mm. B. Other smaller lesions all slightly decreased or no change in size Rt occipital lobe all slightly decreased in size. Rt parietal lobe slightly decreased in size. Lt cerebellum slightly decreased in size. Slightly decreased extent of an indeterminate enhancement at the left subinsular area. Rec F/U to exclude metastasis. New appearance of an indeterminate enhancing lesion at the right frontal lobe Skull, PNS, orbits, and temporal Unremarkable.

Keras 모델 사용

```
['ci', 'f', 'u', 'cerebral', 'metastases', 'p', 'gkrs', 'axial', 't1wi', 'sagittal', 't1wi',
'axial', 't2wi', 'axial', 'flair', 'axial', 't2', 'gre', 'image', '획득하였으며', '조영증강을',
'시행함', 'brain', 'csf', 'space', 'related', 'findings', 'multiple', 'cerebral', 'metastases',
'rt', 'frontal', 'lobe', '6', 'lesions', 'middle', 'frontal', 'gyrus', '9', 'mm', '8mm', 'b',
'smaller', 'lesions', 'slightly', 'decreased', 'change', 'size', 'rt', 'occipital', 'lobe', 's
lightly', 'decreased', 'size', 'rt', 'parietal', 'lobe', 'slightly', 'decreased', 'size', 'l
t', 'cerebellum', 'slightly', 'decreased', 'size', 'slightly', 'decreased', 'extent', 'indeter
minate', 'enhancement', 'left', 'subinsular', 'area', 'rec', 'f', 'u', 'exclude', 'metastasi
s', 'new', 'appearance', 'indeterminate', 'enhancing', 'lesion', 'right', 'frontal', 'lobe',
'skull', 'pns', 'orbits', 'temporal', 'unremarkable']
```

BERT 모델 사용

```
['ci', ',', 'f', '/', 'u', 'for', 'cerebral', 'meta', '##sta', '##ses', ',', 's', '/', 'p',
'g', '##kr', '##s', '.', 'axial', 't', '##1', '##wi', ',', 'sa', '##git', '##tal', 't', '##1',
'##wi', ',', 'axial', 't', '##2', '##wi', ',', 'axial', 'flair', ',', 'axial', 't', '##2',
'*', 'gr', '##e', 'image', '[UNK]', 'ㅈ', '##ㅗ', '##ㅇ', '##ㅐ', '##ㅇ', '##ㅈ', '##ㅡ', '##
ㅇ', '##ㄱ', '##ㅏ', '##ㅇ', '##ㅇ', '##ㅡ', '##ㄹ', 'ㅅ', '##ㅣ', '##ㅎ', '##ㅐ', '##ㅇ', '##ㅎ',
'##ㅏ', '##ㅁ', '.', 'brain', ',', 'cs', '##f', 'space', ',', 'and', 'related', 'findings', 'mu
ltiple', 'cerebral', 'meta', '##sta', '##ses', '.', 'rt', 'frontal', 'lobe', '6', 'lesions',
'a', '.', 'middle', 'frontal', 'g', '##yr', '##us', '9', 'mm', '8', '##mm', '.', 'b', '.', 'ot
her', 'smaller', 'lesions', 'all', 'slightly', 'decreased', 'or', 'no', 'change', 'in', 'siz
e', 'rt', 'o', '##cci', '##pit', '##al', 'lobe', 'all', 'slightly', 'decreased', 'in', 'size',
'.', 'rt', 'par', '##ie', '##tal', 'lobe', 'slightly', 'decreased', 'in', 'size', '.', 'lt',
'ce', '##re', '##bell', '##um', 'slightly', 'decreased', 'in', 'size', '.', 'slightly', 'decre
ased', 'extent', 'of', 'an', 'ind', '##eter', '##minate', 'enhancement', 'at', 'the', 'left',
'sub', '##ins', '##ular', 'area', '.', 'rec', 'f', '/', 'u', 'to', 'exclude', 'meta', '##sta',
'##sis', '.', 'new', 'appearance', 'of', 'an', 'ind', '##eter', '##minate', 'enhancing', 'le
s', '##ion', 'at', 'the', 'right', 'frontal', 'lobe', 'skull', ',', 'p', '##ns', ',', 'orbit
s', ',', 'and', 'temporal', 'un', '##rem', '##ark', '##able', '.']
```

개발 과정 정리

※ 1. 원본 데이터를 Pandas DataFrame으로 가져오기

- Pandas에서 데이터 파일(csv, excel, table 등)을 Python에서 분석할 목적으로 변환 가능.

- **{Index : value}** (dict 같은) 구조의 Series 자료구조,
  Row, Column 기반의 2차원 데이터 구조인 DataFrame을 활용할 수 있다.

```
import pandas as pd

kiumSet = pd.read_csv('상대/절대파일 경로.csv')
df = pd.DataFrame(kiumSet)
```

kiumSet.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6190 entries, 0 to 6189
Data columns (total 3 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Findings         4814 non-null   object
 1   Conclusion       6156 non-null   object
 2   AcuteInfarction  6190 non-null   int64
dtypes: int64(1), object(2)
memory usage: 145.2+ KB
```

총 3개의 Column이 있으며 행(데이터 수)는 총 6190개

'Findings' Column에서 발견된 결측값(NaN)은 1376개

'Conclusion'은 34개

'AcuteInfarction'은 0개

※ 2. 데이터 전처리 작업

- 데이터 처리에 있어서 문제를 발생시키거나 인공지능 학습에 반드시 필요하지 않은 요소를 제거.

  - Column에서 결측값이 존재하는 데이터 추출

    -> **df['Column명'].isnull()**

    ☆ **df.fillna('', inplace=True)** # 결측값(NaN)을 빈 문자열('')로 변환

	Findings	Conclusion	AcuteInfarction
266	MRI for radiosurgery ₩r₩n	NaN	0
446	MRI for radiosurgery₩r₩n	NaN	0
482	MRI for radiosurgery₩r₩n	NaN	0
537	MRI for radiosurgery₩r₩n	NaN	0
716	MRI for radiosurgery ₩r₩n	NaN	0

- 결측값이 존재하는 위치에 적절한 데이터를 대체할 수 있다면 값의 변경을 시도할 수 있다.
- 이 때, 예측하는 값을 두기 때문에 값의 변경의 경우 신중.

※ 2. 데이터 전처리 작업

- CSV 파일에서 DataFrame으로 변환한 결과, 불필요한 이스케이프 문자(₩n, ₩r, ₩t 등)가 많이 산재.

  -> Escape Character는 DataFrame에서 특정 문자로 취급하기 때문에 데이터 처리에 문제 발생 가능.

- 사람의 육안으로 판단하기 쉽도록 작성된 문항 번호나 특수문자가 존재. -> 인공지능의 입장에선 불필요.

-> 언급된 문자들을 적절히 처리(삭제 등)하는 과정이 필요하며 모든 데이터에 일관되게 적용할 수 있도록 코드를 작성할 필요가 있다. → 반복문과 정규표현식의 조합.

```
for i in range(df.shape[0]):
    row = df.iloc[i]
    Ftext = ' '.join(map(str, row['Findings'].split('₩n'))).strip()
    Ftext = Ftext.replace('₩r', '')
    Ctext = ' '.join(map(str, row['Conclusion'].split('₩n'))).strip()
    Ctext = Ctext.replace('₩r', '')

    Ftext = re.sub('[1-9]₩.[^0-9]|[1-9]₩)|[₩-₩<₩>₩(₩)₩:]', "", Ftext)
    Ctext = re.sub('[1-9]₩.[^0-9]|[1-9]₩)|[₩-₩<₩>₩(₩)₩:]', "", Ctext)

    Atext = int(str(row['AcuteInfarction']).strip())

    df.iloc[i] = [Ftext, Ctext, Atext]
```

※ 2. 데이터 전처리 작업

불필요한 문자에 대한 전처리 수행 결과

1. New appearance of small enhancing lesion at the right basal ganglia(<5mm)
--> Probably metastasis
DDx.)
1) normal vessel structure
2) subacute lacunar infarct
2. No change of small enhancing lesion at the left occipital lobe, probable metastasis.
3. Old infarction at left parietal lobe and both basal ganglia.
4. Non specific microbleeds at right temporal lobe, both corona radiata and left frontal lobe.
5. Microangiopathy.

변경 전

New appearance of small enhancing lesion at the right basal ganglia 5mm Probably metastasis DDx. normal vessel structure subacute lacunar infarct No change of small enhancing lesion at the left occipital lobe, probable metastasis. Old infarction at left parietal lobe and both basal ganglia. Non specific microbleeds at right temporal lobe, both corona radiata and left frontal lobe. Microangiopathy.

변경 후

※ 2. 데이터 전처리 작업

- BERT의 사전학습 모델인 'bert-base-multilingual-cased'에 적용하기 위해 데이터 구조 변경

1) 불용어(문장에서 수식 역할만 하는 의미 없는 단어 – the, a(an), is(are) 등)처리 및 구두점 분리.

-> 대량의 데이터로 자연어 처리 모델을 연구한 nltk 라이브러리의 'punkt', 'stopwords' 사전 활용.

2) 자연어 데이터는 ['Findings', 'Conclusion'] 2개의 Column에 존재 → 하나의 문장 데이터 Column으로 변환.

Ex. 'Findings' = "Hello NLP.", 'Conclusion' = "Oh, Hi bro?" 2개로 분리되어 있는 자연어 데이터를

1개의 Column = "Hello NLP. Oh, Hi bro?" 문장 데이터로 변환.

3) BERT 모델이 이해할 수 있게 BERT에서 사용하는 특수토큰을 문장 데이터에 추가.

-> [CLS] : 첫 문장의 시작을 의미하는 토큰. 문장의 가장 처음에 한 번 추가.

-> [SEP] : 문장의 마침표를 의미하는 토큰. 각 문장의 마지막이라면 계속 추가.

※ 2. 데이터 전처리 작업

BERT 모델에 적용하기 위한 문장 데이터

[CLS] Clinical information Lung cancer patient. [SEP] Axial T1WI, sagittal T1WI, axial T2WI, axial FLAIR, axial T2* GRE image 획득하였으며 조영증강을 시행함. [SEP] Right basal ganglia에 5mm 미만의 새롭게 enhancing되는 병변이 있고 probably metastasis로 생각됨. [SEP] 그러나 DDx.로 normal vessel structure와 subacute lacunar infarct을 고려해야 함. [SEP] 그 외 이전과 큰 변화 없음. [SEP] New appearance of small enhancing lesion at the right basal ganglia 5mm Probably metastasis DDx. [SEP] normal vessel structure subacute lacunar infarct No change of small enhancing lesion at the left occipital lobe, probable metastasis. [SEP] Old infarction at left parietal lobe and both basal ganglia. [SEP] Non specific microbleeds at right temporal lobe, both corona radiata and left frontal lobe. [SEP] Microangiopathy. [SEP]

※ 3. 문장 데이터 토큰화

1) BERT의 사전학습 모델인 'bert-base-multilingual-cased'에 문장 데이터를 입력으로 넣어 단어 토큰 추출.

-> BERT 의 사전학습 모델에서 등록된 단어를 찾는다.
만약, 사전에 포함된 단어가 아니라면 사전에 포함된 단어까지 인식한 다음,
뒤에 추가로 붙는 단어를 '##'으로 구분해 이해한다.

['[CLS]', 'Clinical', 'information', 'Lu', '##ng', 'cancer',
'patient', '.', '[SEP]', 'A', '##xia', '##l', 'T1', '##W', '##I', ',',
'sa', '##gitt', '##al', 'T1', '##W', '##I', ',', 'a', '##xia', '##l',
'T2', '##W', '##I', ',', 'a', '##xia', '##l', 'FL', '##A', '##IR', ',',
'a', '##xia', '##l', 'T2', '*', 'GR', '##E', 'image', '획', '##득',
'##하였으며', '조', '##영', '##증', '##강', '##을', '시', '##행',
'##함', '.', '[SEP]', 'Right', 'basal', 'gang', '##lia', '##에', '5',
'##mm', '미', '##만', '##의', '새', '##롭', '##게', 'en', '##han',
'##cing', '##되는', '병', '##변', '##이', '있고', 'probably',

※ 3. 문장 데이터 토큰화

2) 추출된 단어 토큰에 고유한 번호를 Mapping. -> 단어 자체를 학습하는 것보다 대응하는 수치로 학습.

-> 단어가 같은 토큰은 모두 같은 번호를 가진다. 서로 다른 단어는 무조건 다른 번호.

-> [CLS] 토큰은 101, [SEP]는 102의 고정된 번호로 Mapping된다.

3) 학습에 필요한 하나의 Input Data를 구성하기 위해 고정된 길이의 배열에 담는다.

-> BERT 모델은 최대 512 크기의 배열을 학습에 사용한다.

-> 단어 토큰의 개수가 512만큼 있지 않다면 값의 공백이 발생한다. -> 0의 Padding을 적용한다.

4) Padding 값은 공간을 채워주는 목적이지 학습에 직접 사용할 필요가 없는 값이다.

-> Attention Masking으로 실제 학습에 필요한 데이터 구간을 직접 명시하는 방법이 있다.

-> Input Data와 같은 크기(512)의 배열이며 Padding이 있는 구간은 0으로,
실제 데이터가 있는 구간은 1로 값을 넣어둔다.

※ 3. 문장 데이터 토큰화

학습에 사용될 512 크기의 배열 데이터

Padding

```
[     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0
      0     0   101   140 11281   117 11649 12396   119   102
    138 30967 10161 94459 13034 11281   117 10148 78108 10415
  94459 13034 11281   117   169 30967 10161 102172 13034 11281
    117   169 30967 10161 83243 10738 73522   117   169 30967
  10161 102172   115 58787 11259 18170   117   169 30967 10161
    141 13034 11281   117 69617 40333 74116 58573 11565 52157
  10738  9999 118813 53529 63938 52157 10738 10530 33378  9678
  30858 119230 47181 10622  9485 25549 48533   119   102 40586
    117 42818 11565 16199   117 10111 16382 79441   138 103411
  10114 30156 103411 11649 12396 10160 10105 45684 24289 23178
  45846 10406 16216   119   102 11982 17437 73995 10161 23773
```

배열 데이터에 대응하는 Attention Mask

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
```

※ 4. Input Data 학습/검증

☆ 학습/검증에 사용할 언어모델, Library

1. 자연어 (사전)학습 모델**: BERT (bert-base-multilingual-cased)**
2. 딥러닝 연산 보조 Library**: Pytorch**

BERT (Bidirectional Encoding Representations from Transformers)

- 대표적인 자연어 처리 모델. (미국 만화의 동명 캐릭터가 있으며 실제로 사용...)
- 사전 모델로 Input Data를 만들기 전 전처리 단계에서 토큰화에 사용할 수 있다.
- 사전 모델에 Fine-Tuning하면서 목적에 맞는 모델을 개발할 수 있는 특장점.

Pytorch

- Pytorch는 대표적인 머신러닝 프레임워크.
- BERT 모델의 Fine-Tuning 또는 Validation 용도의 Input Data 생성 목적.
- Tensor 자료구조를 사용하며 복잡한 연산 처리 과정에 장점.

※ 4. Input Data 학습/검증

- **Pytorch DataLoader**: DataFrame 구조를 Tensor로 변환하고, 여러 Tensor를 묶어 하나의 Set로 만든 구조.
  → 학습에 사용될 512 사이즈의 배열 데이터가 하나의 Tensor가 되는 것.
  → 학습에 필요한 Attention Mask 배열과, 정답지인 label 데이터를 함께 묶는다.
  → 그래서 총 3개의 Tensor로 구성된 하나의 DataLoader가 만들어진다.

```
torch.Size([512])
tensor([[    0,     0,     0,  ..., 13890,   119,   102],
        [    0,     0,     0,  ..., 10157,   119,   102],
        [    0,     0,     0,  ..., 29731, 17530,   102],
        ...,
        [    0,     0,     0,  ..., 15851,   119,   102],
        [    0,     0,     0,  ..., 10251,   119,   102],
        [    0,     0,     0,  ..., 43977,   119,   102]], dtype=torch.int32)
tensor([[0, 0, 0,  ..., 1, 1, 1],
        [0, 0, 0,  ..., 1, 1, 1],
        [0, 0, 0,  ..., 1, 1, 1],
        ...,
        [0, 0, 0,  ..., 1, 1, 1],
        [0, 0, 0,  ..., 1, 1, 1],
        [0, 0, 0,  ..., 1, 1, 1]])
tensor([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        1, 0, 0, 0, 0, 0, 0, 0])
```

Input Data

Attention Mask

Label

※ 4. Input Data 학습/검증

- **BERT 처리 과정**

Data = '[CLS] DCU is the University located in Gyeongsan. [SEP] Do you want to come in? [SEP]'

Input Data = '0 0 0 0 ... 101 289 120 230 217 330 516 920 ... 102' - 512 size

Attention Mask = '0 0 0 0 ... 1 1 1 1 1 1' - 512 size