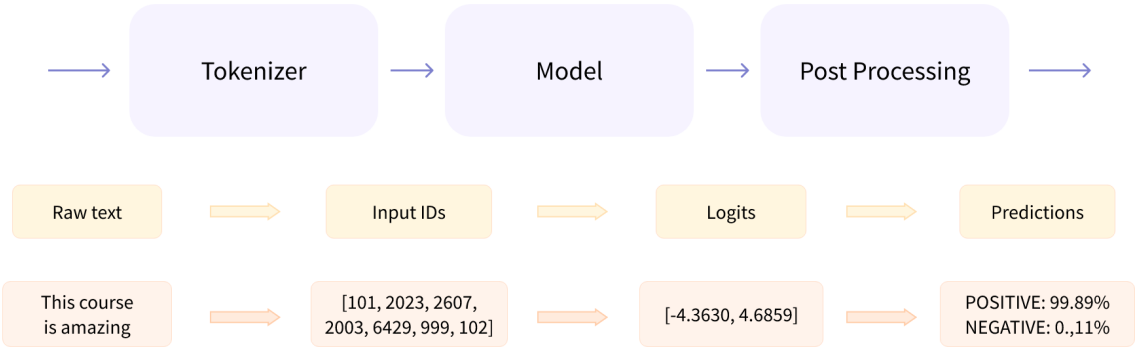
토크나이저(Tokenizer)

출처 : https://wikidocs.net/166796

토크나이저란?(Tokenizer)



- 토크나이저(Tokenizer)
  - NLP 파이프라인의 핵심 구성 요소 중 하나
  - 입력된 텍스트를 모델이 처리할 수 있는 데이터로 : 텍스트 → 숫자

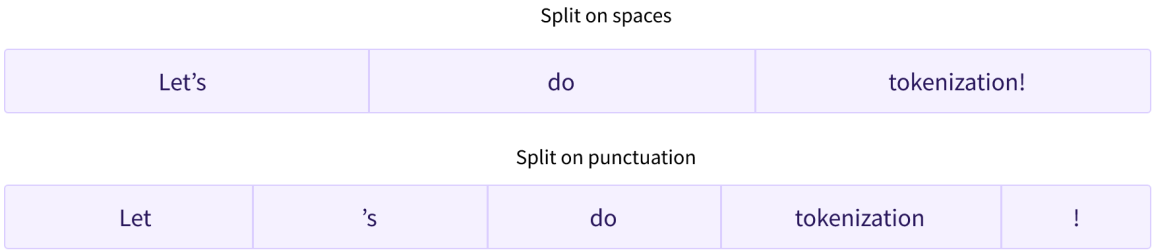
ex) 원시 텍스트(raw text)

```
Let's do tokenization!
```

  - 위와 텍스트가 들어 왔을 때 숫자로 변환해주기 위한 규칙이 필요
  - 모델에 가장 적합한 가장 의미있는 표현을 단위로 토큰화하여 숫자와 매치

단어 기반 토큰화(Word-based Tokenization)

- 단어를 기반으로 토큰화(공백을 기준으로 토큰화 하는 방법)



위 그림과 같이 구두점에 대한 추가 규칙 있는 단어기반 토크나이저의 변형도 존재

- 어휘집(Vocabulary)
  - 토큰화된 단어마다 0부터 어휘집(Vocabulary) 크기(개수) 사이의 ID(식별자)를 할당

    ex) dog : 100, cat :300
  - 모델은 ID를 사용하여 각 단어를 식별함
- 단점
  - 단어기반 토크나이저가 특정 언어를 완전히 커버하려면, 해당 언어의 **모든 단어**에 대한 식별자(ID)가 필요
  - dog - dogs, run - running 과 같은 단어들이 다른 식별자로 다르게 표현되어 모델이 처음에는 유사한 단어인지 파악하기 어려움
- Unknown 토큰
  - 어휘집에 없는 단어를 표현하기 위한 사용자 정의 토큰
  - “[UNK]”, “” 로 많이 표시
  - unknown 토큰을 많이 생성 = 해당 단어의 합당한 표현 찾을 수 없음을 의미 → 어휘집을 만들 때 토크나이저가 unknown 토큰을 최대한 적게 출력하게끔 하는 것이 목표

  예시)

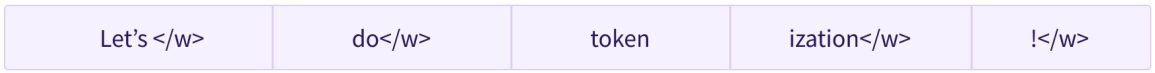
문자 기반 토큰화(Character-based Tokenization)

- unknown 토큰의 양을 줄이는 방법 중 한가지
- 문자(character)를 기반으로 토큰화(공백을 기준으로 토큰화 하는 방법)
- 장점
  - 어휘집의 크기가 매우 작음
  - 모든 단어들이 문자를 가지고 만들어질 수 있기에 unknown 토큰이 훨씬 적어짐
- 단점
  - 토큰화된 표현 자체가 단어가 아닌 문자 기반 → 직관적으로 각 토큰의 의미 파악 어려움
    - 언어마다 달라짐 : 중국어의 문자는 라틴 언어의 문자보다 더 많은 정보 전달함
  - 모델이 처리해야할 토큰의 수가 많아짐

ex) 위 그림처럼 단어기반 토큰화한 문장보다 훨씬 많은 토큰으로 변환됨

하위 단어 토큰화(Subword Tokenization)

- 문자기반, 단어기반 방식의 장점을 최대한 활용하기 위해 결합된 방식
  - 빈번하게 사용되는 단어는 더 분할 하지 않음
  - 희귀한 단어는 의미있는 하위 단어로 분할



- 예시

  "tokenization"는 "token"과 "iztion"으로 분리됨

  두 개의 토큰이 각각의 의미를 가지면서도 공간 효율적임(긴 단어를 두개의 토큰으로만 표현)

  문자기반의 경우, 11개의 토큰으로 표한됨

  단어기반의 경우, 1개의 토큰이지만 token, tokenizer tokenization 등등이 유사하지만 다른 토큰으로 표현됨

⇒ 어휘집을 너무 크게 만들지 않더라도 충분히 많은 수의 토큰을 표현할 수 있음, unknown 토큰이 거의 없어짐

- 해당 방법은 하위 단어를 연결하여 길이가 긴 복잡한 단어를 임의로 만들 수 있는 터키어(Turkish), 한국어 같은 교착 언어(agglutinative languages)에서 특히 유용함

세부적인 기법들

하위 단어 토큰화와 관련된 더 많은 기법들이 존재

- Byte-level BPE (GPT-2에 사용됨)
- WordPiece (BERT에 사용됨)
- SentencePiece, Unigram (몇몇 다국어 모델에 사용됨)