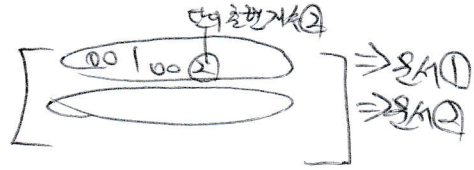
KoNLPy (한국어 처리 Package) ("코엔엘파이")

i) Kkma : 꼬꼬마 (서울대)
ii) Hannanum : 한나눔 (kaist)
iii) Twitter
iv) Mecab : 일본어용 형태소 분석기를 한글에서 사용
v) Komoran : 코모란

Sakit-Learn의 문서 전처리 기능

- BoW (Bag of Words) : 문서를 숫자 Vector로 변환하는 기본적인 방법.

$x_{i,j}$ = 문서 d_i 내의 단어 t_j의 출현빈도



$$x_{i,j} = \begin{cases} 0, & \text{만약 단어 } t_j \text{가 문서 } d_i \text{ 안에 없으면} \\ 1, & \text{만약 단어 } t_j \text{가 문서 } d_i \text{ 안에 있으면} \end{cases}$$

- Sakit Learn의 문서 전처리 기능.

i) DictVectorizer : 각 단어의 수를 세어 놓은 사전에서 BoW 벡터를 만든다. ("단어의 사용 빈도")
ii) CountVectorizer : 문서 집합에서 단어 토큰을 생성하고, 각 단어의 수를 세어 BoW 인코딩한 Vector를 만든다. (토큰화 + 카운팅)
ⓐ문서를 토큰 리스트로 변환 ⓑ각 문서를 BoW 인코딩 벡터로 변환 ⓒ각 문서에서 토큰의 출현 빈도를 센다
iii) TfidfVectorizer : TF-IDF 방식으로 단어의 가중치를 조절한 BoW 벡터를 만든다 (Term Frequency - Inverse document frequency) ⇒단어의 수를 그대로 쓰지 않고, 모든 문서에 공통적으로 나오는 단어는 가중치 축소
iv) HashingVectorizer : (해시 함수) hash function을 사용해 적은 메모리와 빠른 속도로 BoW 벡터를 만든다.

iii) CounterVectorizer 인수
ⓐ stop_words ⇒무시할 단어 목록
ⓑ Analyzer, tokenizer, token_pattern ⇒사용할 토큰 생성자를 선택
ⓒ n-gram ⇒토큰의 크기를 결정. (1-그램 : 토큰 하나만 단어로 사용
2-그램 : 연속적인 토큰을 하나의 단어로 사용)
ⓓ max_df, min_df ⇒빈도수 (너무 자주 나오거나, 너무 안나오면 무시)