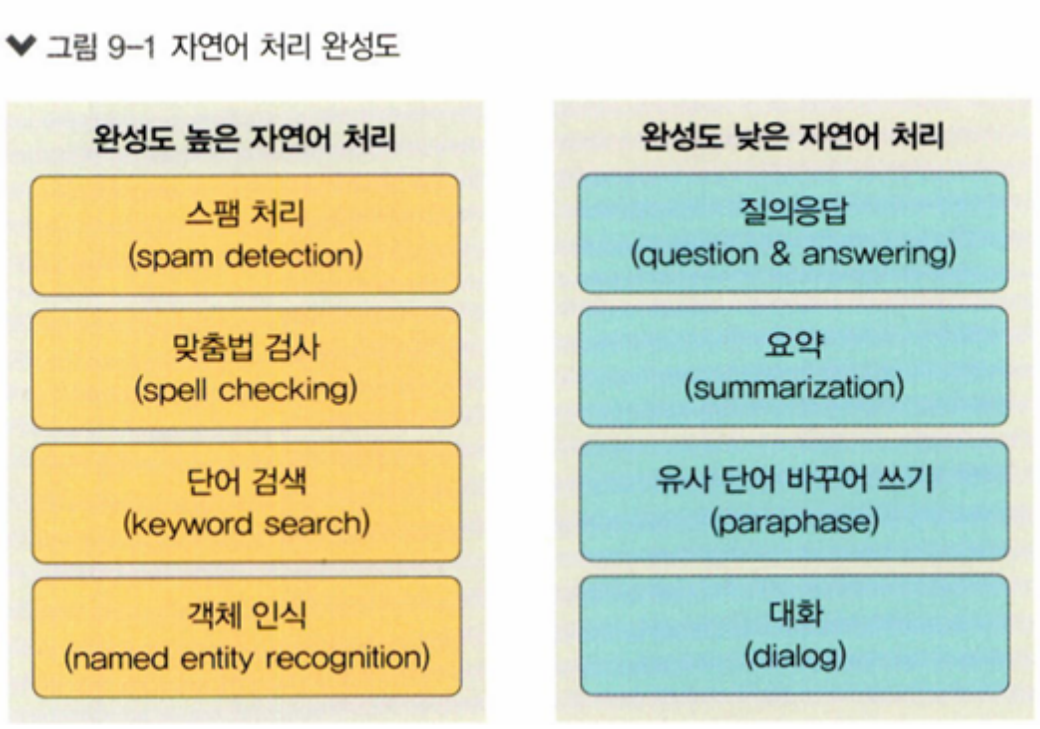
9장 자연어 전처리

생성일	@2024년 9월 22일 오후 4:06
주차	Week14
완료여부	☐

9.1 자연어 처리란

우리가 일상생활에서 사용하는 언어 의미를 분석하여 컴퓨터가 처리할 수 있도록 하는 과정으로 인간 언어에 대한 이해 + 딥러닝에 대한 이해가 필요한 분야

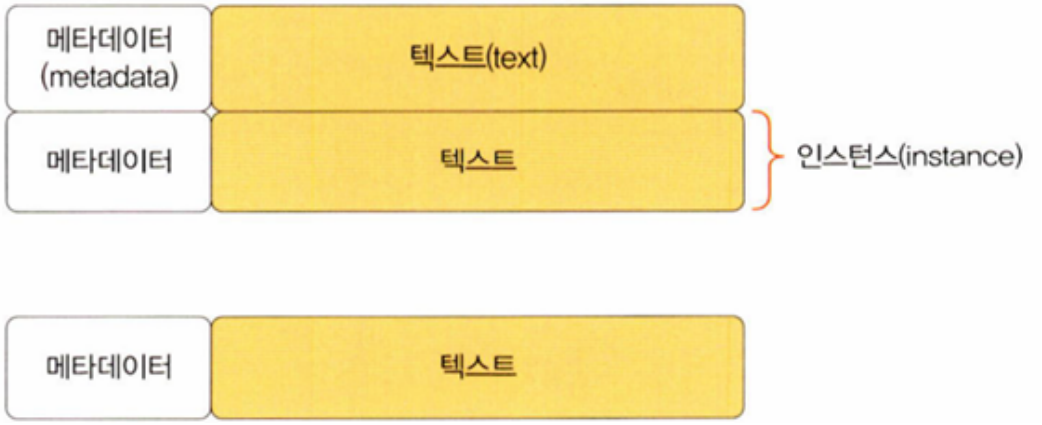


9.1.1 자연어 처리 용어 및 과정

[관련 용어]

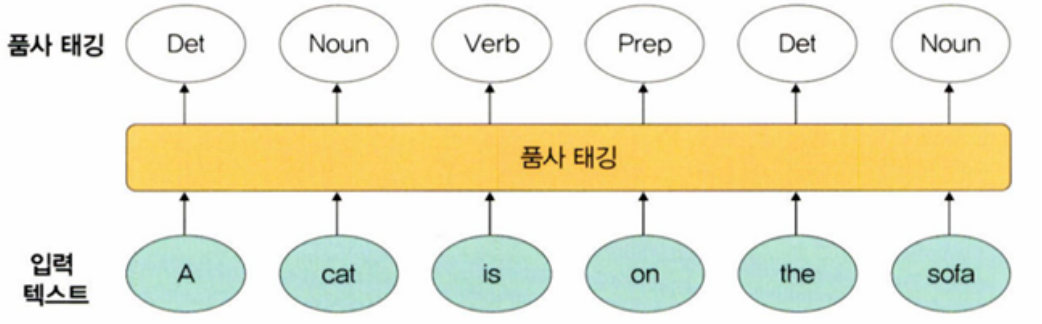
- corpus: 모델을 학습시키기 위한 데이터로 특정 목적에서 표본을 추출한 집합 (말뭉치)

❯ 그림 9-2 말뭉치(corpus)



- token: 자연어 처리를 위해 문서를 나누는 단어로 문자열을 토큰으로 나누는 과정을 tokenizing, 이때 사용하는 함수를 토큰 생성 함수라고 함
- tokenization: 텍스트를 문장이나 단어 단위로 분리하는 과정
- stop words: 문장 내에서 많이 등장하는 단어로 분석과 관계 없으므로 사전에 제거해 줘야 함 ex) a, the, she, he
- stemming: 어간추출로 단어를 기본 형태로 만드는 작업을 의미함 ex) cosigned, cosigning → cosign
- part of speech tagging: 품사를 직별하기 위해 붙여주는 태그를 의미함

❯ 그림 9-4 품사 태깅



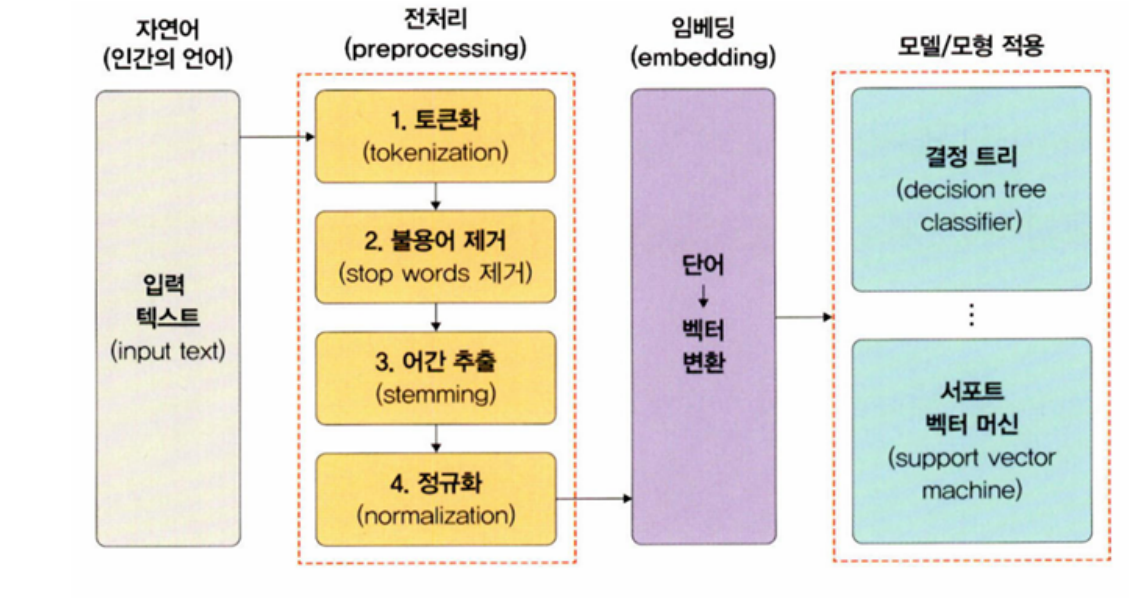
det은 한정사, prep은 전치사

ㄴ NLTK 를 사용

[자연어 처리 과정]

1. 자연어가 입력 텍스트로 들엉옮
2. 입력된 텍스트에 대한 전처리 과정

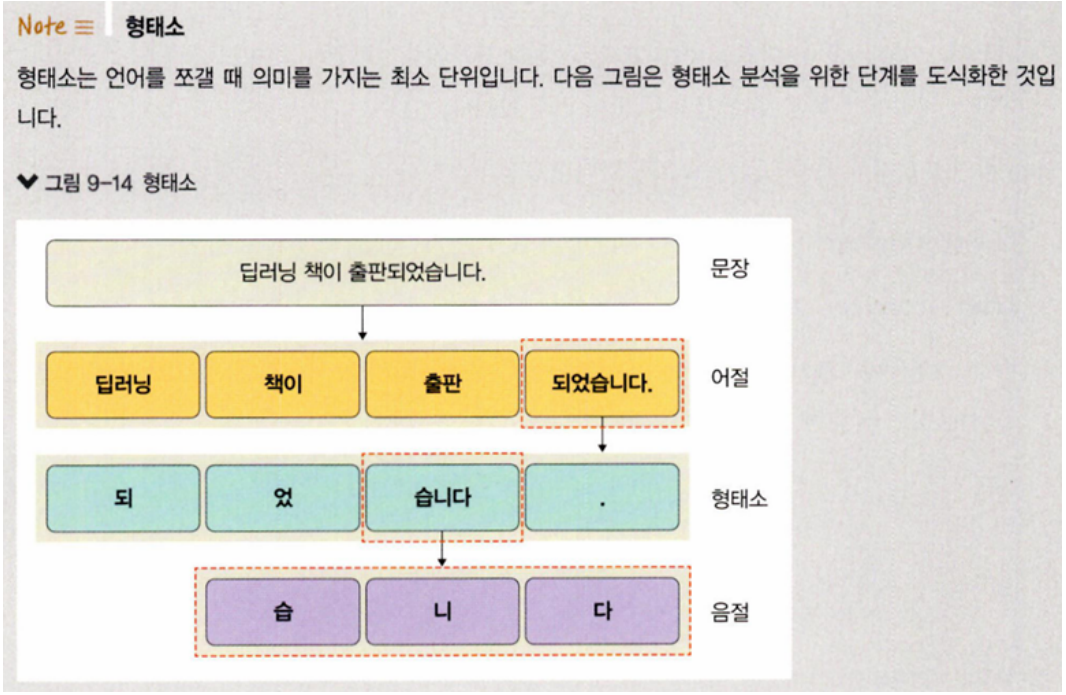
3. 전처리가 끝난 단어들을 임베딩 (벡터로 변환)
4. 모델 이용하여 데이터에 대한 분류/예측을 수행



9.1.2 자연어 처리를 위한 라이브러리

NLTK - corpus, tokenization, 형태소 분석, 품사 태깅을 제공함

KoNLPy - 한국어 처리를 위한 파이썬 라이브러리로 형태소 분석, 품사 태깅을 제공함



Genism - Word2Vec 라이브러리로 임베딩, 토픽 모델링, LDA 제공

사이킷런 - 문서 전처리 할 수 있는 라이브러리로 CountVecotrizer (단어 등장 횟수를 기준으로 특성 추출), Tfidvectorizer (TF-IDF값을 이용해 텍스트에서 특성 추출), HashingVectorizer (해시 함수를 사용해서 횟수를 기준으로 특성 추출) 를 제공함

9.2 전처리

텍스트 데이터에 대한 전처리 작업을 위해 토큰화, 불용어(stop words) 제거, 어간 추출등의 작업을 하는 과정

9.2.1 결측치 확인

결측치 처리 방법 - 데이터에 하나라도 nan값이 있을 때 행 전체를 삭제, 데이터가 거의 없는 열은 열 자체를 삭제, 최빈값 혹은 평균값으로 대체

9.2.2 토큰화

주어진 텍스트를 단어/문자 단위로 자르는 것

문장 토큰화: 문장의 마지막을 뜻하는 기호에 따라 분리하는 것

단어 토큰화: 띄어쓰기를 기준으로 문장을 구분함

한국어는 띄어쓰기 만으로 토큰을 구분하기 어렵기 때문에 KoNLPy를 이용

9.2.3 불용어 제거

문장 내에서 빈번하게 발생하여 의미를 부여하기 어려운 단어들을 제거

9.2.4 어간 추출

어간 추출(stemming)과 표제어 추출(lemmatization)은 단어 원형을 찾아주는 것으로 어간추출의 경우 단어 그 자체만 고려하기 때문에 품사가 달라도 사용이 가능한 반면, 표제어 추출은 단어가 어떤 문장 속에서 어떤 품사로 스였는지 고려하기 때문에 품사가 같아야 사용이 가능함

두 추출의 목적은 모두 어근 추출이지만 표제어 추출과 달리 어간 추출은 사전에 없는 단어도 추출할 수 있음

9.2.5 정규화

모든 데이터가 동일한 정도의 스케일을 갖도록 하는 것