02. 한글 형태소 분석 #src: 9_자연어처리/ch02_한글형태소분석 참고

1) 자연어 처리; 자연어(일상적으로 사용하는 언어, NLP) 처리 #맞춤법, 번역기, 검색 엔진 등

2) 자연어 처리 절차

(1) 전처리: 단어, 어절 추출

(2) 분석 후보 생성: 형태소(의미를 가진 최소 단위) 분리, 원형 복원, 품사 태깅

(3) 제약조건(불용어 처리) 규칙 확인

(4) 분석

3) 한글 형태소 분석 엔진

(1) KoNLPy: 파이썬용 자연어 처리기(JPype1 패키지 의존) #https://pypi.org/project/JPype1/#files

(2) KOMORAN: 자바로 만든 형태소 분석기(JAVA_HOME 시스템 변수)

(3) HanNanum: 자바로 만든 형태소 분석기(JAVA_HOME 시스템 변수)

(4) Kkma: 서울대학교 연구실

(5) KoNLP: R용 자연어 처리기

4) 말뭉치(Corpus); 컴퓨터를 이용해 자연어 분석작업을 할 수 있도록 만든 문서 집합

5) 워드 클라우드; 단어를 출현 빈도에 비례하는 크기로 단어 빈도수를 시각화 하는 기법

```
wordcloud = WordCloud(background_color='white',
                      max_words=300,
                      font_path='c:/Windows/Fonts/H2PORM.TTF',
                      relative_scaling=0.2,
                      mask=mask) #마스크 된 부분에만 글자 나오게
                      stopwords=불용어) #워드 클라우드에서 불용어 제외시키기
```

(1) 불용어 처리(불용어 사전+불용어 추가); 워드 클라우드 생성시 제외시킬 단어 지정

불용어 = STOPWORDS | ENGLISH_STOP_WORDS | set(['단어1', '단어2'])

(2) 마스킹; 워드 클라우드를 지정된 마스크 이미지에 맞도록 처리

(3) 워드 클라우드 저장: wordcloud.to_file('파일명')

6) nltk.Text(단어리스트).plot(num); 상위 num만큼 단어 빈도수 그래프

7) 워드 임베딩(Word Embedding); 단어간 유사성 도출

Word2Vec(문장, size=100, window=5, min_count=2, workers=3) #size; 벡터의 차원, workers=쓰레드 수
#window; 문장 내 현재 단어와 예측 단어 사이의 최대 거리, min_count=n; n번 이상 나온 단어