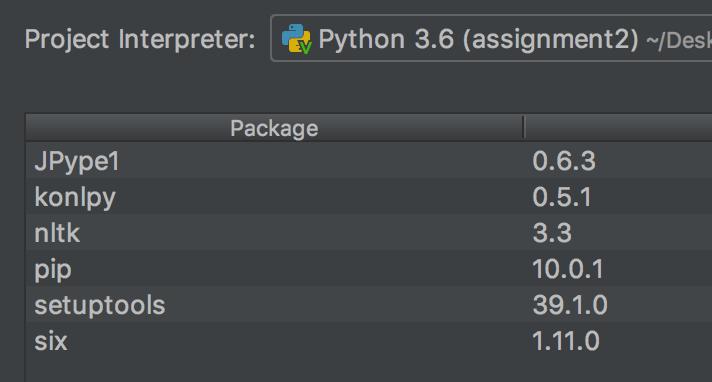
인공지능 #2

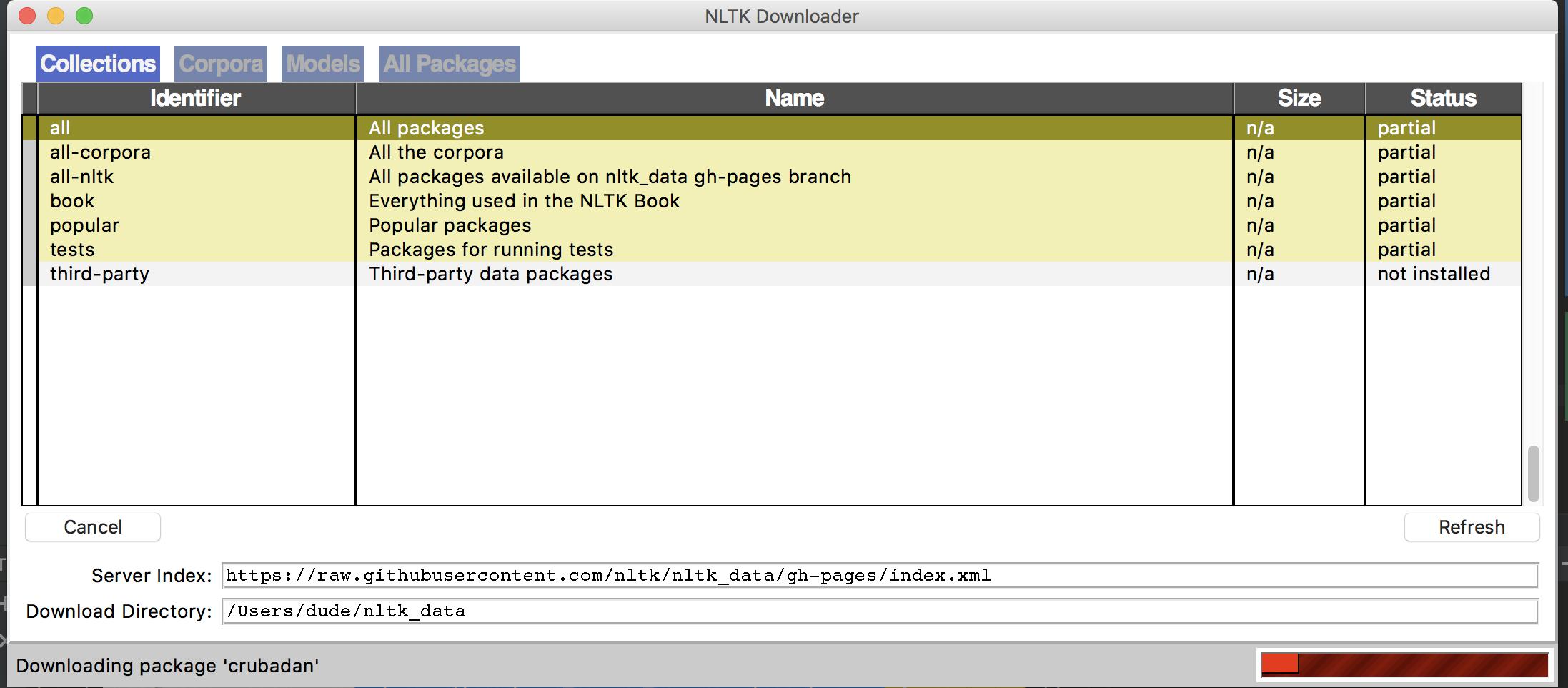
영화 리뷰 긍정 부정 판별하기

1. Environment

konlpy, nltk 로 형태소 분석 사용

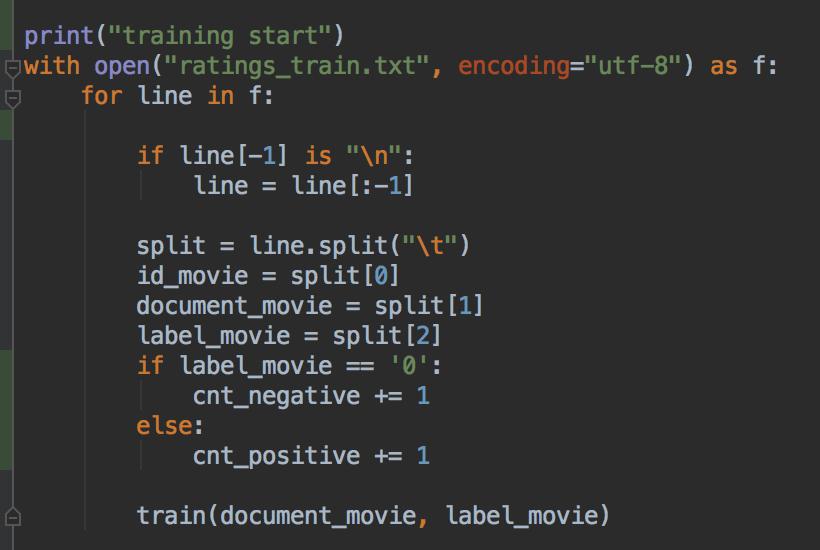


nltk.download() 로 nltk 관련 형태소 tag 다운



2. 코드 설명 - 더 좋은 결과를 얻기 전 가장 기본 코드

2-1 파일 읽기



line 별로 읽는다

id , document, label 로 분류한다 negative, positive 전체 확률위해 cnt 를 1씩 더한다

train으로 단어에 대한 확률을 update한다

2-2 훈련시키기



github.com/chandong83/python\_hangul\_check\_function 한글 문장인지 체크하는 regular expression

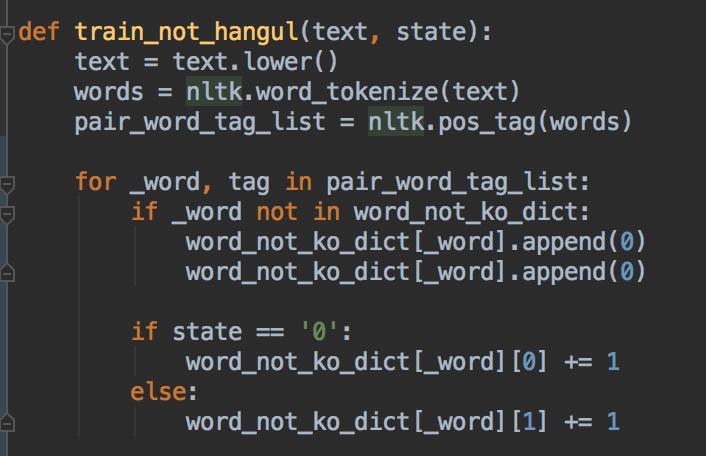
한글(한글 + 영어) 문장이라면 한글 train으로 영어 문장이라면 영어 train으로 훈련시킨다

2-3 훈련시키기 - 한글



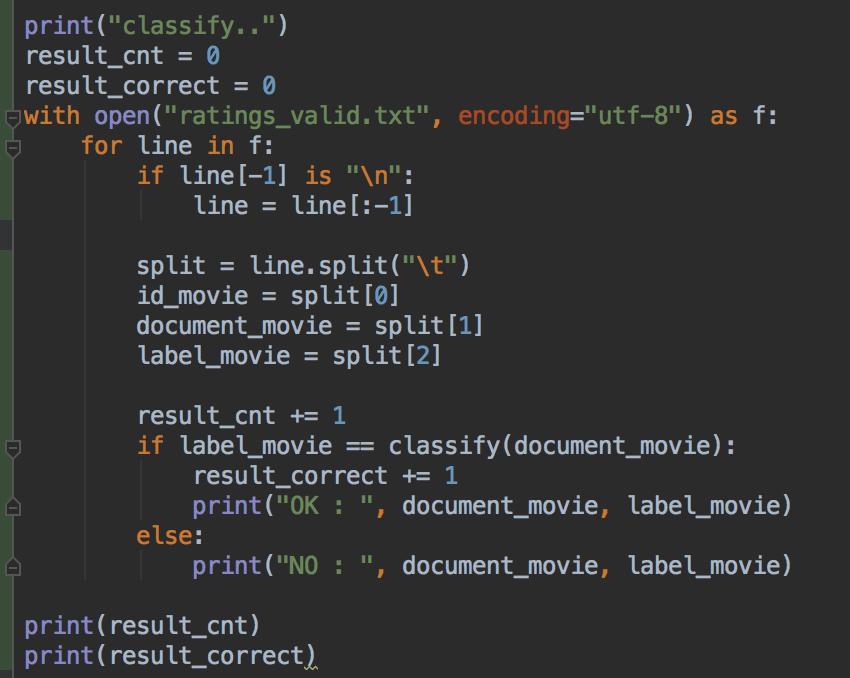
아직 tag값은 사용하지 않고 work token만 가지고 word\_ko\_dictionary 에 긍정값인지 부정값인지 확률을 측정하기 위 해 카운팅

2-4 훈련시키기 - 영어



아직 tag값은 사용하지 않고 work token만 가지고 word\_not\_ko\_dictionary 에 긍정값인지 부정값인지 확률을 측정 하기 위해 카운팅

2-5 분류시키기



한줄 씩 읽으면서 classify로 긍정인지 부정인지 판단한다 아래 소스는 valid를 check하는 것으로 정확도를 측정하기 위한 코드

2-6 분류시키기 2



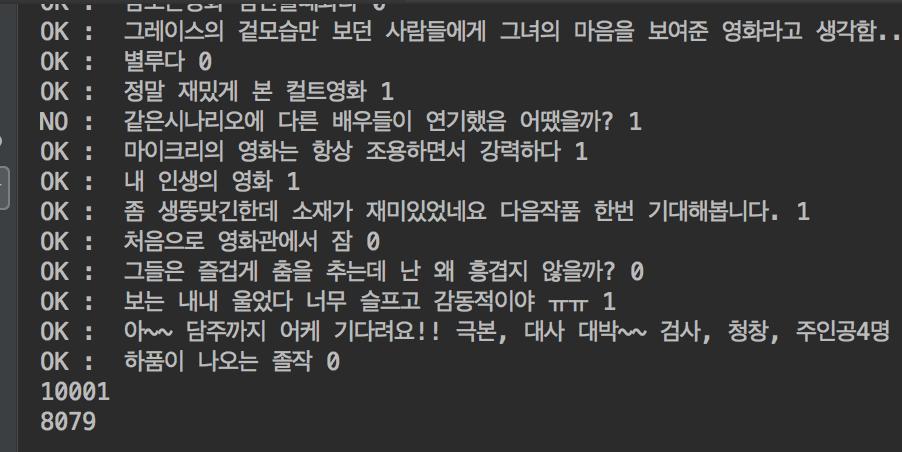
한글 인지 아닌지는 dictionary를 check하는데만 사용

p(pos|words) 랑 p(neg|words)를 위해 p(pos) \* p(word1|pos) \* p(word2|neg) …. p(neg) \* p(word1|neg) \* p(word2|neg) ….

를 위해서 log를 씌운 값을 더해서 pos가 큰지 neg가 큰지 return한다

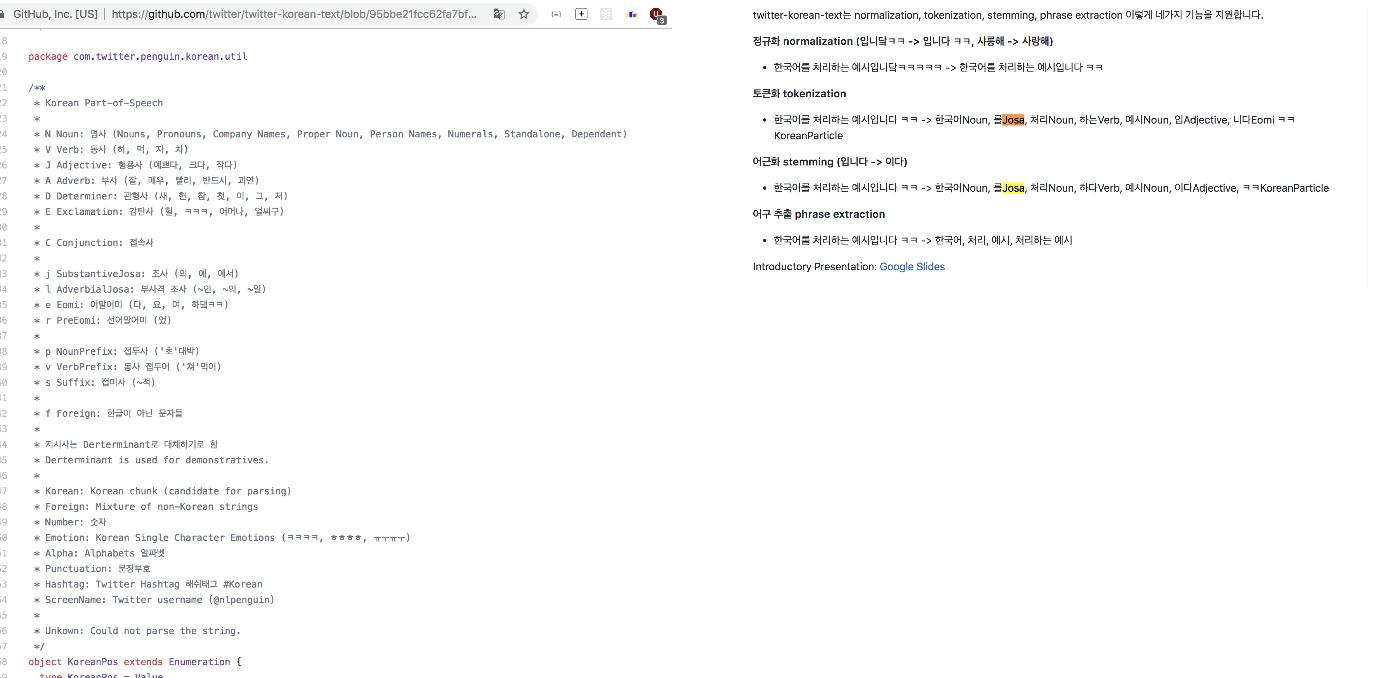
3 실행결과

3-1 형태소에 따로 기능을 추가하지 않고 token화만 한 후 분류한 결과

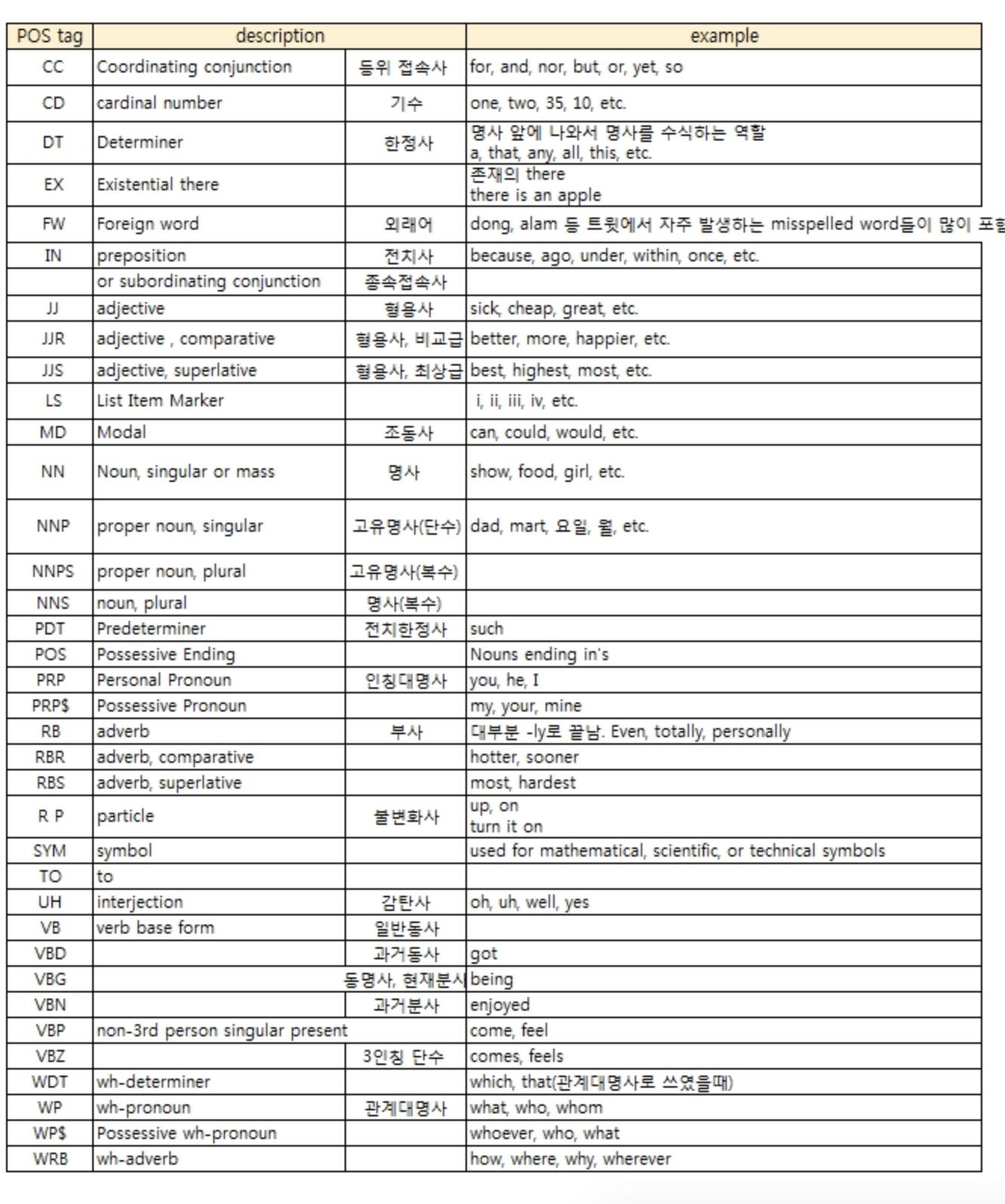


앞으로 할 것을 위한 참고자료

- 한글

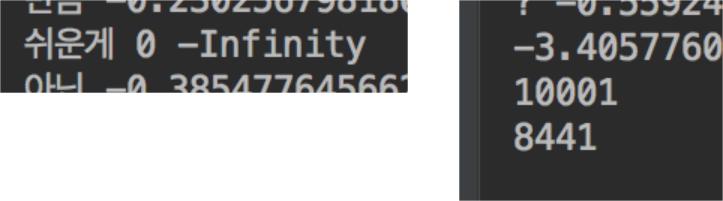


- 영어



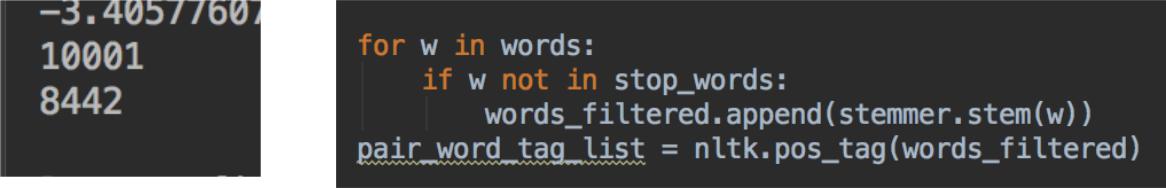
4 추가 작업

1. Infinity 가 되는 것을 막기 위해 양측에 threshold 값 추가 8079 -> 8441



2. 한글이 아닌것 training에 stemmer 추가

8441 -> 8442



1. 한글에 training에 stemmer 추가 8442 -> 7856
2. 한글 classify에도 stemmer추가 7856 -> 8337
3. 한글 training 에 classify에 Su x, Josa, Punctuation 제거 8337 -> 8336
4. 한글 문장이 뒤쪽에 영향을 많이 받는 것 같아서 뒤쪽에 가중치를 둠

8336 -> 8315

1. 그 외에도 여러 변수에 대한 가중치를 추가해서 (1의 threshold값, 6의 power값 등) 8315 -> 8370대

