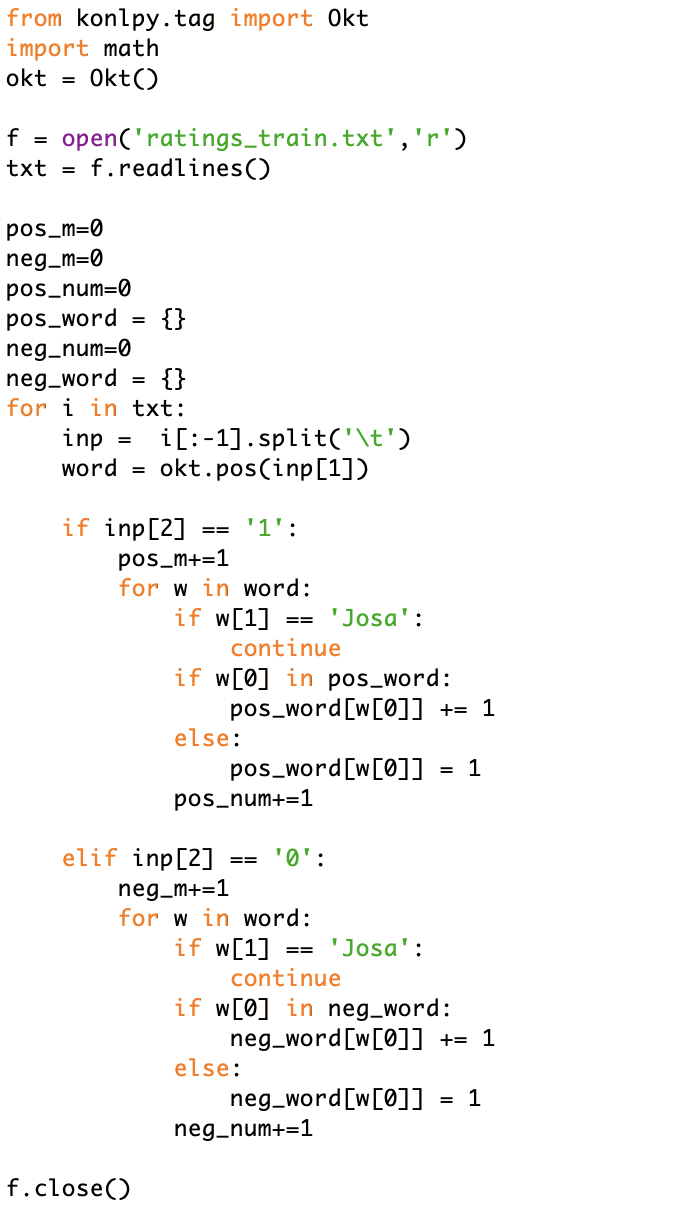
Assignment 2

2015004448 김정욱

# 코드설명

1. 나이브-베이즈 모델 학습



txt변수에 한 줄씩 입력 받은 후 \t기준으로 구분하여 inp에 저장한다.(Inp[0]는 id, inp[1]은 댓글, inp[2]는 1,0이 저장된다)

inp[1](댓글)을 Okt.pos(inp[1])함수를 사용하여 형태소로 잘라서 word에 저장한다.

(norm은 정규화 stem은 근어로 표시하기를 나타낸다. 데이터의 갯수가 적어서 인지 valid에서 확인한 정답률이 더 낮앗져서 제외시킴.)

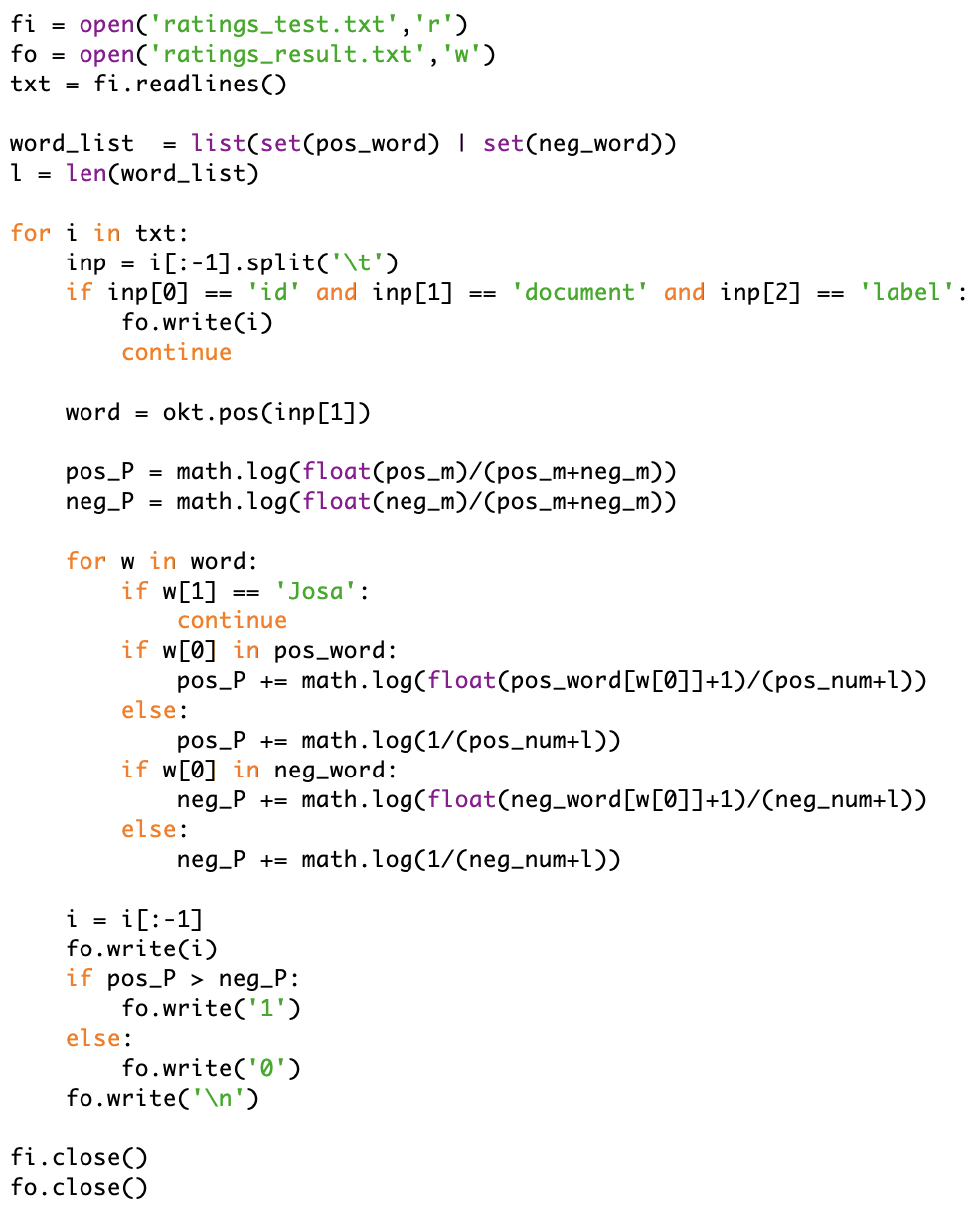
긍정문장일 경우 pos\_m(긍정문장의 갯수)을 증가시키고 형태소로 자른 하나씩 탐색한다. w[1]에는 형태소의 구분이, w[0]에는 잘려진 형태소가 저장되어있다.

조사인 경우는 문장의 긍정 부정에 영향이 없으므로 제외시키고 pos\_word[w[0]](해당형태소의 갯수)와 pos\_num(긍정 형태소의 갯수)를 증가시킨다.

부정문장일 경우 그정문장과 동일하게 동작하고, 변수는 neg\_m, neg\_word[w[0]], neg\_num으로 갯수를 세준다.

pos\_word와 neg\_word는 딕셔너리 형태로 사용되어서 각각의 인덱스를 형태소로 사용하고 인덱스마다 갯수를 저장해 준다.

2. 결과예측



ratings\_test.txt 에서 입력 받은 값을 예측하여 ratings\_result.txt에 출력해준다. 입력받은 한 줄 마다 split을 이용해서 댓글을 구분해주고, 댓글은 okt.pos()함수를 사용하여 형태소별로 구분해준다.

Pos\_P(긍정일 확률)과 neg\_P(부정일 확률)을 계산해야 하는데 크기 비교가 목적이기에 log를 취해서 under flow를 방지해준다.

트레이닝 한 데이터에 없는 형태소가 나올 경우 분자가 0이 되어서 log값을 구할 수 없고, 정확도가 떨어지기 때문에 분자에 1을 더해주고, 분모에는 중복을 제외한 형태소 갯수를 더해준다.(laplace smoothing기법)

word\_list는 pos\_word와 neg\_word를 set형태로 바꾼 후 합집합을 만들어 갯수를 세주면 중복을 제외하고 셀 수 있다.

확률을 구한 후 크기를 비교하여 긍정일 확률이 더 크면 1로 아니면 0으로 예측한다.

3. 정답률 예측



결과 예측과 같은 방법으로 ratings\_valid.txt 결과를 예측 후 정답과 비교하여 같은 갯수 를 세어서 총 문장수로 나눠주면 정답률을 확인 할 수 있다. 정확한 정답률 이라고 할 수는 없고, 대략적으로 어느정도인지 확인 할 수 있다.

# 결과

Ratings\_valid.txt를 예측하여 결과와 비교한 결과 85.16% 정답률

