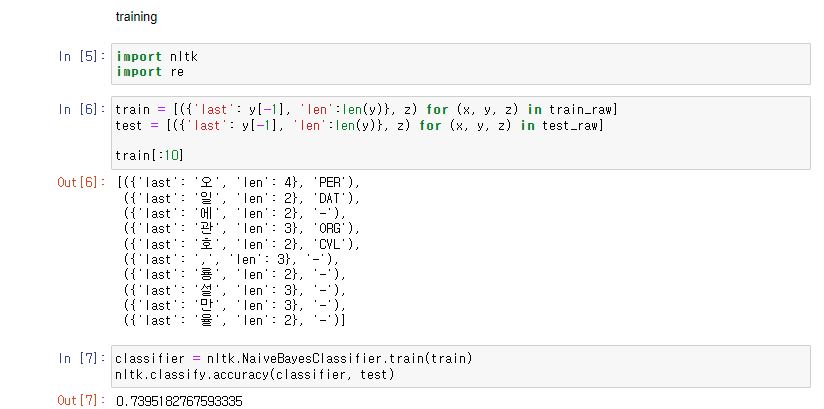
정보처리및자연언어처리 기말 과제

20180071 오서영

우선 train.txt 파일을 읽어와 f 변수에 저장한 뒤, split()하여 (문장내어절번호, 어절, 개체명) 형태로 튜플화하여 train\_raw에 저장하였습니다.

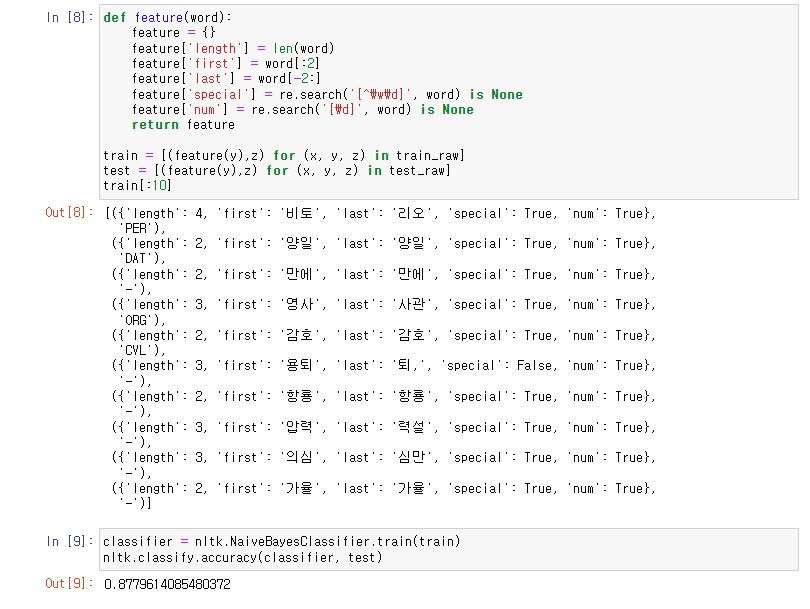


마찬가지로 test.txt에 대해서도 동일한 과정을 진행하여 test\_raw에 저장하였습니다.



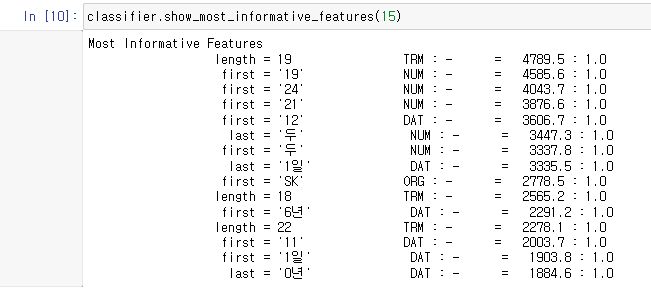
우선 필요한 nltk, re 모듈을 불러왔습니다. 그리고 train set에 대하여 어절의 마지막 문자와 어절의 길이를 feature extraction 하였고, 마찬가지로 test set에 대해서도 동일한 과정을 진행하였습니다. 그 다음 nltk 모듈의 NaiveBayesClassifier 모델을 불러와 train set에 대해 학습을 시키고, test set으로 accuracy를 측정하였습니다. 이는 0.73 수준으로 개선이 필요한 모델입니다.

이렇게 성능이 낮게 학습된 이유에 대해 고민해 보았습니다. 이는 해당 데이터의 라벨과 관련없는 feature를 사용했기 때문이라는 추측을 할 수 있었습니다. 예를 들어, 품사 분류 모델의 경우 문장의 구조와 관련이 있으므로 목적격 조사나 접미사처럼 어절의 맨 마지막 글자가 큰 영향을 끼치지만, 이 모델은 개체명 분류 모델이므로 어절의 마지막 문자가 크게 중요하지 않습니다. 그보다는 해당 단어가 속한 분야를 대표할 수 있는 feature를 필요로 합니다.



그래서 feature 생성을 위한 feature()함수를 만들었습니다.

**length**: 개체명마다 대략적인 어휘의 길이가 있을 것이라 생각하여 추출한 feature입니다. 예를 들어, 어휘가 과도하게 길고 복잡한 경우 전문적인 분야(의학, it 등)의 어휘일 것이라고 추측할 수 있습니다. 실제로 모델 트레이닝 이후 변수 중 가장 영향력이 높은 length=19는 TRM일 확률이 높았습니다.

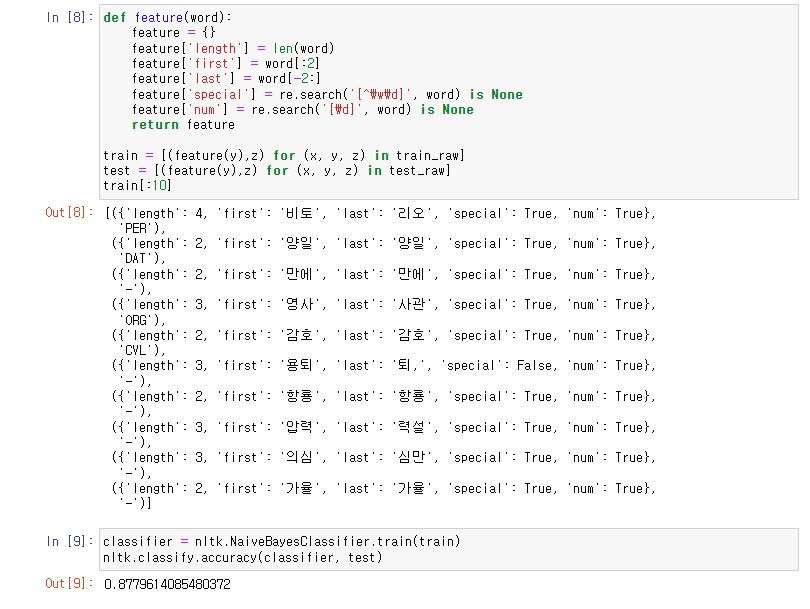


**first**: 어절의 앞 두자리 문자열을 추출한 feature입니다. 한자리, 세자리 문자열을 추출해보았으나 해당 데이터셋에서는 두자리 문자열이 가장 성능이 좋아 이를 택하게 되었습니다.

**last**: 어절의 뒤 두자리 문자열을 추출한 feature입니다. 마찬가지로 한자리, 세자리 문자열보다 성능이 좋았습니다. 직접 모델을 돌려봤을 때 어절 전체를 활용한 경우보다 first와 last로 나눈 경우 모델의 성능이 개선되는 것을 확인할 수 있었습니다. 따라서 어절 전체가 아닌 first와 last를 채택하였습니다.

**special**: 특수문자가 포함되었는지 여부를 체크하는 feature입니다. 특수문자가 포함되었으면 False, 포함되지 않았으면 True입니다. 데이터셋에서 어절에 특수문자가 포함되거나 어절이 특수문자로만 구성된 경우 대부분 개체명이 '-'으로 분류되어 있으므로, 이를 반영할 수 있는 feature를 생성하였습니다.

**num**: 숫자가 포함되었는지 여부를 체크하는 feature입니다. 숫자가 포함되었으면 False, 포함되지 않았으면 True입니다. 데이터셋에서 어절에 숫자가 포함된 경우 대부분 개체명이 'NUM'으로 분류된 것을 확인하고 이를 반영할 수 있는 feature를 생성하였습니다.



위의 feature를 활용해서 만든 train set으로 나이브베이즈 모델을 학습시켰습니다. 그리고 동일한 feature를 가진 test set으로 accuracy를 측정해보니, 0.877에 해당하는 accuracy가 나왔습니다.