자연어 처리 – AI 전문가 양성(기본) Day 2

01 Preprocessing & Classification

Jeong Minsu omicro03@gmail.com

INGLab

실습 소개

- **실습 목표**
  - 한국어 텍스트 전처리
  - Scikit-learn을 이용한 classification

- **실습 내용**
  - 네이버 영화 리뷰 데이터를 활용하여 텍스트 데이터 전처리
  - Sentence Embedding
  - Scikit-learn classifier를 이용한 분류 모델 학습 및 평가

Word Embedding

- 한국어 텍스트 전처리

(1) 특수 문자 처리

"으아아~~~ 흠좀너무재미있는ㄴㄴㄴㄴ영화였다*^_^*,,,,,별점9999"

(2) 띄어쓰기 교정

"으아 아 흠 좀 너무 재미있는 영화였다 별점 9999 "

(3) POS tag 활용하기

"흠/NNG 좀/MAG 너무/MAG 재미있/VA 영화/NNG 별점/NNG"

(4) 불용어 제거

"너무/MAG 재미있/VA 영화/NNG 별점/NNG"

Word Embedding

- Scikit-learn을 이용한 classification
  - 네이버 영화 리뷰 데이터 (https://github.com/e9t/nsmc)

Naver sentiment movie corpus v1.0

This is a movie review dataset in the Korean language. Reviews were scraped from Naver Movies.

The dataset construction is based on the method noted in Large movie review dataset from Maas et al., 2011.

Data description

- Each file is consisted of three columns: `id`, `document`, `label`
  - `id` : The review id, provieded by Naver
  - `document` : The actual review
  - `label` : The sentiment class of the review. (0: negative, 1: positive)
  - Columns are delimited with tabs (i.e., `.tsv` format; but the file extension is `.txt` for easy access for novices)
- 200K reviews in total
  - `ratings.txt` : All 200K reviews
  - `ratings_test.txt` : 50K reviews held out for testing
  - `ratings_train.txt` : 150K reviews for training

	id	document	label
0	9976970	아 더빙.. 진짜 짜증나네요 목소리	0
1	3819312	흠...포스터보고 초딩영화줄....오버연기조차 가볍지 않구나	1
2	10265843	너무재밋었다그래서보는것을추천한다	0
3	9045019	교도소 이야기구먼 ..솔직히 재미는 없다..평점 조정	0
4	6483659	사이몬페그의 익살스런 연기가 돋보였던 영화!스파이더맨에서 늙어보이기만 했던 커스틴 ...	1

Word Embedding

- Scikit-learn을 이용한 classification
  - 네이버 영화 리뷰 데이터 이진 분류

(1) 데이터 전처리

(2) 문장 임베딩

(3) Scikit-learn classifier 학습

(4) 성능 평가

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidfv = TfidfVectorizer(max_features=2000).fit(corpus)
train_x = tfidfv.transform(train['document']).toarray()

from sklearn.naive_bayes import BernoulliNB

nb = BernoulliNB()
nb.fit(train_x, train['label’])

y_pred = nb.predict(test_x)

classification_report(test['label'], y_pred)
```