CH8

텍스트 분석

- 비정형 텍스트 데이터를 어떻게 피처 형태로 추출하고 의미있는 값을 부여
- 수행 프로세스 : 전처리 → 피처 벡터화/추출 → 모델 수립/학습/평가
- 패키지

1. NLTK : 대표적이고 큰 영향을 끼친 패키지이나 속도 측면에서 아쉬운 결과
2. Gensim : 토픽 모델링에 우수, Word2Vec
3. SpaCy : 뛰어난 수행 성능

- 텍스트 정규화 과정

1. 클렌징 : 불필요한 문자, 기호 사전에 제거
2. 텍스트 토큰화
    a. 문장 토큰화
    - 문장의 마지막 기준으로 분리
    - 시맨틱적 의미 중요할 때
    - from nltk import sent_tokenize

    b. 단어 토큰화
    - 순서가 중요하지 않을 경우 사용
    - 정규표현식 사용
    - from nltk import word_tokenize

    c. n-gram : 연속된 n개의 단어를 하나의 토큰화 단위로
3. 필터링/스톱워드 제거/철자 수정
    - 큰 의미가 없는 단어 제거 (조사, 대명사 등등)
4. Stemming
    - 원형 단어를 원래 단어에서의 어근 단어를 추출하는 방식 위주로 찾아냄

- Porter, Lancaster, Snowball Stemmer

5. Lemmatization
   - 의미론적 기반에서 원형 찾아냄
   - 시간 더 오래 걸림
   - WordNetLemmatizer

BOW (Bag of Words)

- 문맥이나 순서 무시하고 일괄적으로 단어에 대한 빈도값 부여해 피처 추출
- 문장들에서 중복 제거하고 개별 문장에서 해당 단어가 나타나는 횟수를 기재
- 장점 : 쉽고 빠른 구축
- 단점
  1. 문맥 의미 반영 부족 : 문맥적 해석 처리 불가
  2. 희소 행렬 문제 : 수행 시간과 예측 성능 저하

BOW 피처 벡터화

1. 카운트 기반 벡터화

- 카운트 값이 높을수록 중요한 단어로 인식

2. TF-IDF 기반 벡터화

- 자주 나타나는 단어에 높은 가중치 + 모든 문서에 전반적으로 나타내는 단어 패널티
  ⇒ 가중치 균형

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

tf_{ij} = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

- 사이킷런에서 구현하기

CountVectorizer/TfidfVectorizer

텍스트 피처 단위로 벡터화 + 변환하고 CSR 형태의 희소 행렬 반환

희소 행렬 → 대부분 값이 0인 행렬 → 메모리 낭비 문제

1. COO 형식

- (행, 열, 값) 형식으로 저장
- sparse.coo_matrix((data, (row_pos, col_pos)))

2. CSR 형식

- COO의 행/열 위치를 나타내기 위해 반복적인 위치 데이터를 사용해야하는 문제점 해결

CSR(Compressed Sparse Row)

$$A_{IJ} = \begin{pmatrix} 10_{(0,0)} & 0 & 0 & 12_{(0,3)} & 0 \\ 0 & 0 & 11_{(1,2)} & 0 & 13_{(1,4)} \\ 0 & 16_{(2,1)} & 0 & 0 & 0 \\ 0 & 0 & 11_{(3,2)} & 0 & 13_{(3,4)} \end{pmatrix}$$

$$\text{데이타}(A) = \begin{pmatrix} 10_{(0,0)} & 12_{(0,3)} & 11_{(1,2)} & 13_{(1,4)} & 16_{(2,1)} & 11_{(3,2)} & 13_{(3,4)} \end{pmatrix}$$

$$\text{열 인덱스 값}(JA) = \begin{pmatrix} 0_{(0} & 3_{)} & 2_{(1} & 4_{)} & 1_{(2)} & 2_{(3} & 4_{)} \end{pmatrix}$$

$$\text{행 압축 정보}(IA) = \begin{pmatrix} 0_{(0)} & 2_{(1)} & 4_{(2)} & 5_{(3)} & 7_{(4)} \end{pmatrix}$$

- 데이터를 행(가로)의 순서대로 정리 압축하는 방법이다.
- 구성요소
  - 행 순서대로 데이터 배열(A)
  - 행 순서대로 데이터의 열 인덱스 배열(JA)
  - 행 압축 정보 배열(IA)
    - **행 압축 정보 배열**은 [최초 시작 행번호, 시작 행에서의 데이터 누적 개수, 두번째 행에서의 데이터 누적 개수,..., 마지막 행에서의 데이터 누적개수]이다.

- 고유값의 시작 위치만 알고 있으면 되므로 메모리 적게 들고 빠른 연산 가능
- sparse.csr_matrix((data2, col_pos, row_pos_ind))

20 뉴스그룹 분류

감정 분석

- 주관적인 감성/의견/감정/기분 등 파악
- 감성 수치 계산 → 긍부정

1. 지도학습 : 학습 데이터와 타겟 레이블 값 기반 학습 후 예측

IMDB 영화평_지도학습

2. 비지도학습 : ‘Lexicon’이라는 감성 어휘 사전을 이용해서 판단

- 감정 지수 : 긍정, 부정 감성 정도를 의미하는 수치
- Synset : 단어가 가진 문맥, 시맨틱 정보를 제공하는 WordNet 핵심 개념
- SentiWordNet, VADER, Pattern 등 다양한 감성 사전 존재

IMDB 영화평_SentiWordNet Lexicon

IMDB 영화평_VADER

토픽 모델링

- 문서 집합에 숨어있는 주제 찾기
- 중심 단어를 함축적으로 추출

1. LSA (Latent Semantic Analysis)
2. LDA (Latent Dirichlet Allocation)

20 뉴스그룹_LDA