8장 텍스트 분석

- **NLP(National Language Processing)** : 머신이 인간의 언어를 이해하고 해석하는 데 더 중점을 두고 기술이 발전
- **텍스트 마이닝(Text Mining)** : 비정형 텍스트에서 의미 있는 정보를 추출하는 것에 좀 더 중점을 두고 기술이 발전

  ex)텍스트 분류(Text Classification), 감성 분석(Sentiment Analysis), 텍스트 요약(Summarization), 텍스트 군집화(Clustering)

01.텍스트 분석 이해

- 비정형 텍스트 데이터를 어떻게 피처 형태(숫자)로 추출하고 추출된 피처에 의미 있는 값을 부여할까?
- **피처 벡터화(Feature Vectorization)**

  BOW(Bag of WORDS), Word2Vec 두 가지 방법이 존재한다.
- **피처 추출(Feature Extraction)**
- **텍스트 분석 수행 프로세스**
  1. 텍스트 사전 준비작업(텍스트 전처리)
  2. 피처 벡터화/추출
  3. ML 모델 수립 및 학습/예측/평가
- **파이썬 기반의 NLP, 텍스트 분석 패키지**
  - NLTK(Natural Language Toolkit for Python) - 가장 대표적인 NLP 패키지(영어)
  - Gensim - 토픽 모델링 분야에서 가장 두각을 나타내는 패키지
  - SpaCy - 뛰어난 수행 능력으로 최근 가장 주목을 받는 NLP 패키지

02. 텍스트 사전 준비 작업(텍스트 전처리)

텍스트 정규화는 텍스트 데이터의 사전작업을 수행하는 것을 의미하며, 아래와 같은 작업들이 있다.

- **클렌징(Cleansing)**

  텍스트 분석시 방해가 되는 불필요한 문자, 기호 등을 사전에 제거하는 작업

  ex) HTML, XML 태그나 특정 기호 등을 사전에 제거한다.

- **토큰화(Tokenization)**

  문서에서 문장을 분리하는 문장 토큰화와 문장에서 단어를 토큰으로 분리하는 단어 토큰화로 나눌 수 있다.

- 필터링/스톱 워드 제거/ 철자 수정

  스톱 워드 - 분석에 큰 의미가 없는 단어를 지칭한다.

  많은 언어에서 문법적인 요소에 따라 단어가 다양하게 변한다. → 따라서 단어의 원형을 찾는 것이 의미가 있을 것이다. 이러한 방법론 중 대표적인 것이 Stemming, Lemmatization입니다.

- **Stemming**

  Lemmatization보다는 더 단순화된 방법을 적용해 원래 단어에서 일부 철자가 훼손된 어근 단어를 추출하는 경향이 있다.

- **Lemmatization**

  품사와 같은 문법적인 요소와 더 의미적인 부분을 감안해 정확한 철자로 된 어근 단어를 찾아준다.

03. Bag of Words - BOW

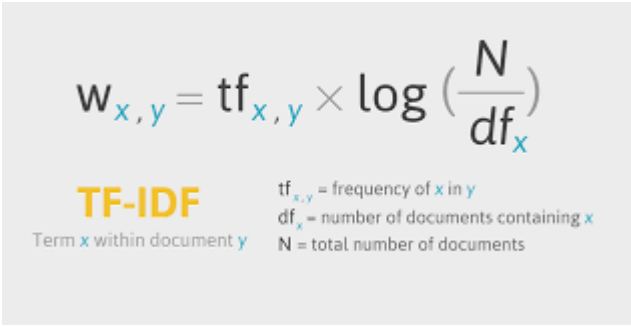
- 방법 : 문서가 가지는 모든 단어(Words)를 문맥이나 순서를 무시하고 일괄적으로 단어에 대해 빈도 값을 부여해 피처 값을 추출합니다.
- 장점 : 쉽고 빠른 구축이 가능하다, 단순히 단어의 발생 횟수에 기반하지만 생각보다 문서의 특징을 잘 반영하여 활용도가 높다.
- 단점 : 문맥 의미(Semantic Context) 반영 부족, 희소 행렬 문제(희소성, 희소 행렬)
- **BOW의 피처 벡터화**

  1. 카운트 기반의 벡터화

     각 문서에서 해당 단어가 나타는 횟수, 즉 Count를 부여하는 식으로 진행된다.

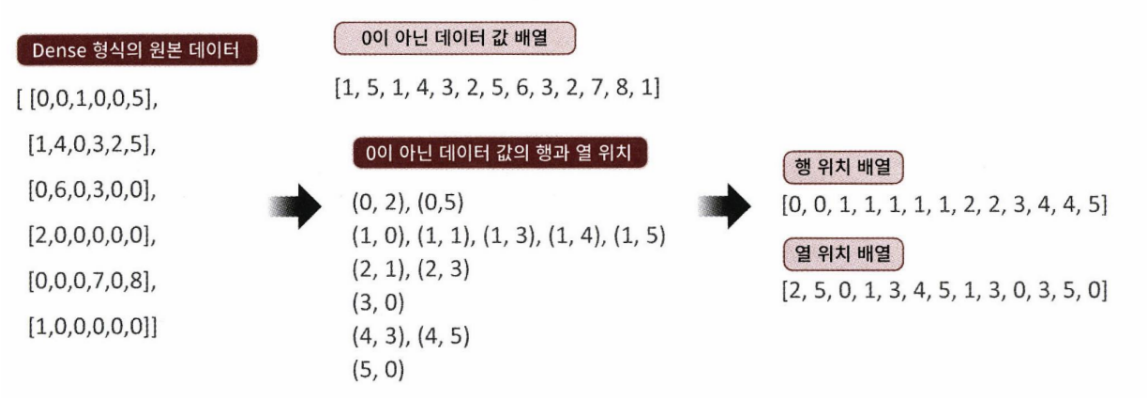
  2. TF-IDF(Term Frequency - Inverse Document Frequency)

     개별 문서에서 자주 나타는 단어에 높은 가중치를 주되, 모든 문서에서 전반적으로 자주 나타는 단어에 대해서는 패널티를 주는 방식으로 값을 부여한다.



대부분의 BOW 형태를 가진 언어 모델은 희소행렬일 수 밖에 없다. 즉 메모리 공간을 많이 차지한다. → COO, CSR 형식을 통해 메모리 문제 해결함

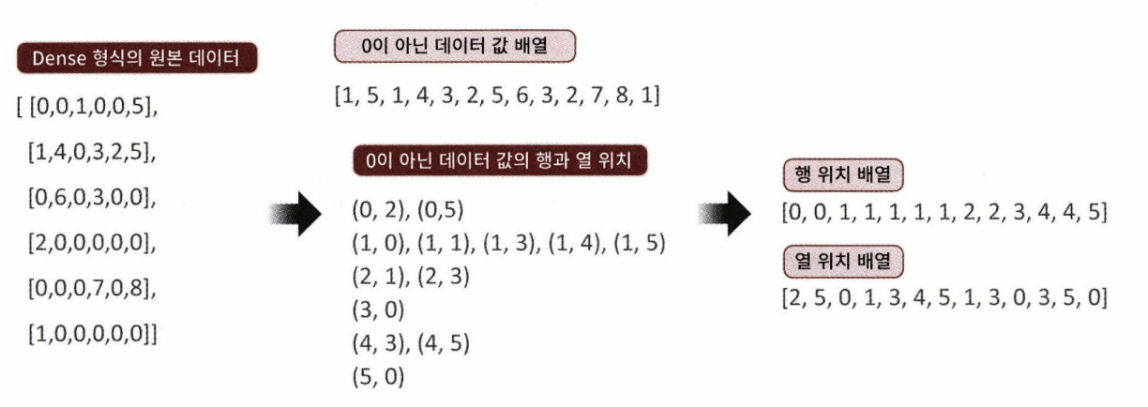
희소 행렬 - COO(Coordinate) 형식



- 0이 아닌 데이터만 별도의 배열(Array)에 저장하고, 그 데이터가 가리키는 행과 열의 위치를 별도의 배열로 저장하는 방식

희소 행렬 - CSR(Compressed Sparse Row) 형식

- COO 형식이 행과 열의 위치를 나타내기 위해서 반복적인 위치 데이터를 사용해야 하는 문제점을 해결한 방식 - 행과 열의 위치에서 반복적으로 나타는 숫자들을 위치의 위치를 표기하는 방식으로 극복함.
- → 사이킷런의 CountVectorizer나 TfidfVectorizer클래스로 변환된 피처 벡터화 행렬은 모두 사이파이의 CSR 형태의 희소 행렬입니다! (어쩐지....ㅎㅎ)



04. 텍스트 분류 실습 - 20 뉴스그룹 분류

- 코드 참고

05. 감성분석

- 문서의 주관적인 감성/의견/감정/기분 등을 파악하기 위한 방법이다.
- 주관적인 단어와 문맥을 기반으로 감성 수치를 계산하는 방법을 이용한다.
- **지도학습** : 학습 데이터와 타깃 레이블 값을 기반으로 감성 분석 학습을 수행한 뒤 이를 기반으로 다른 데이터의 감성분석을 예측하는 방식으로 진행된다. → 일반적인 텍스트 기반의 분류와 동일
- **비지도학습** : 'Lexicon'이라는 일종의 감성 어휘 사전을 이용한다. 이를 이용해 문서의 긍정적, 부정적 감성 여부를 판단한다.
  1. SentiWordNet : NLTK 패키지의 WordNet과 유사하게 감성 단어 전용의 WordNet을 구현한 것
  2. VADER : 소셜 미디어의 텍스트에 대한 감성 분석을 제공하기 위한 패키지
  3. Pattern : 예측 성능 측면에서 가장 주목받는 패키지(감성분석에 관심이 많다면 사용해보기)
- 구체적인 예시는 코드를 통해 참고하기로 한다.

06. 토픽 모델링(Topic Modeling)

- 토픽 모델링 : 문서 집합에 숨어 있는 주제를 찾아내는 것

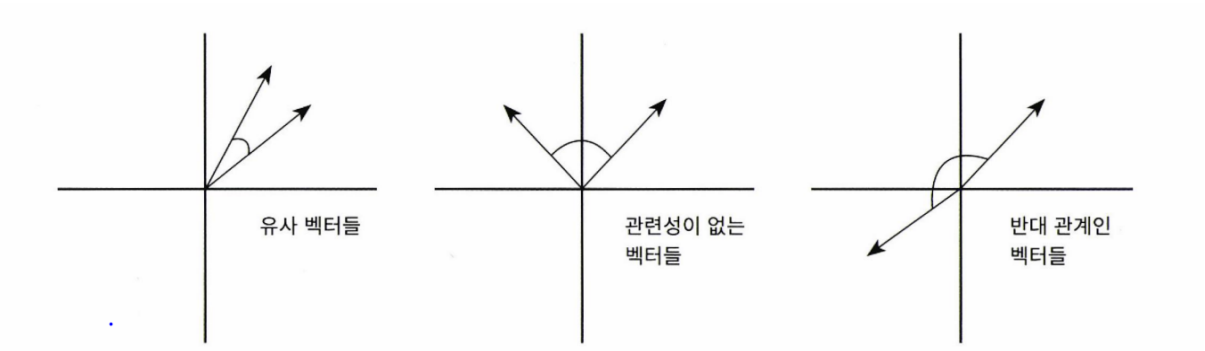
  ex) LSA(Latent Semantic Analysis), LDA(Latent Dirichlet Allocation)

07. 문서 군집화 소개와 실습(Opinion Review 데이터 세트)

- **문서 군집화(document clustering)** : 비슷한 텍스트 구성의 문서를 군집화(Clustering)하는 것
- 동일한 군집에 속하는 문서를 같은 카테고리 소속으로 분류할 수 있으므로 앞에서 소개한 텍스트 기반의 문서 분류와 유사하다. → 하지만 문서 군집화는 사전 학습데이터가 필요없는 비지도학습인게 차이점이다.

08. 문서 유사도 - 코사인 유사도

- **코사인 유사도(Cosine Similarity)** : 벡터와 벡터 간의 유사도를 비교할 때 벡터의 크기보다는 벡터의 상호 방향성이 얼마나 유사한지에 기반한다.
- → **단순 카운트 기반 비교**는 불확실하다. ex) 빈도가 높다 하여 그 단어가 문서에서 차지하는 비중 모른다.



$$\text{similarity} = cos\theta = \frac{A \cdot B}{\| A \| \| B \|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

09. 한글 텍스트 처리 - 네이버 영화 평점 감성 분석

- 코드 참고

10. 텍스트 분석 실습 - 캐글 Mercari Price Suggestion Challenge

- 코드 참고

11. 정리