

# 텍스트 분석

## 1. 텍스트 분석

### 1.1 NLP vs. 텍스트 분석

#### NLP

- 머신이 인간의 언어를 이해하고 해석하는 데 중점
- 기계 번역, 자동으로 질문으로 해석하고 답을 해주는 질의응답 시스템
- 텍스트 분석을 향상하게 하는 기반 기술

#### Text Mining

- 비정형 텍스트에서 의미 있는 정보를 추출하는 것에 중점
- 머신러닝, 언어 이해, 통계 등을 활용해 모델 수립하고 정보 추출
- 활용
  - **텍스트 분류**: 문서가 특정 카테고리에 속하는 것을 예측  
→ 지도학습
  - **감정 분석**: 텍스트에서 나타내는 감정, 의견 등 주관적인 요소를 분석하는 기법  
→ SNS 감정 분석, 영화 리뷰 분석  
→ 지도학습 & 비지도학습
  - **텍스트 요약**: 중요한 주제 추출  
→ 토픽 모델링
  - **텍스트 군집화와 유사도 측정**: 비슷한 유형의 문서에 대해 군집화  
→ 텍스트 분류를 비지도 학습으로 수행

### 1.2 텍스트 분석 이해

- 비정형 데이터인 텍스트를 분석하는 것

- 피처 벡터화 (피처 추출)
  - 텍스트를 word 기반의 다수의 피처로 추출하고 이 피처에 단어 빈도수와 같은 숫자 값을 부여
  - 텍스트가 단어의 조합인 벡터값으로 표현됨
  - BOW (Bag of Words), Word2Vec

## 1.3 텍스트 분석 수행 프로세스

### 1 텍스트 전처리

- 클렌징 작업
- 토큰화 작업
- stop word (의미 없는 단어) 제거 작업
- stemming, lemmatization 어근 추출

### 2 피처 벡터화/추출

- 텍스트에서 피처 추출하고 벡터값 할당
- BOW → Count 기반 또는 TF-IDF 기반 벡터화

### 3 ML 모델 수립 및 학습, 예측, 평가

## 1.4 파이썬 기반 NLP, 텍스트 분석 패키지

- NLTK
- Gensim
- SpaCy

# 2. 텍스트

## 2.1 클렌징

불필요한 문자, 기호 등 제거

## 2.2 토큰화

### 문장 토큰화

문장의 마침표, 개행문자 등의 기호에 따라 분리

```
sent_tokenize
```

### 단어 토큰화

공백, 콤마 등으로 단어 분리

```
word_tokenize
```

## 2.3 스톱워드 제거

- 관사 등 분석에 큰 의미가 없는 단어 지칭
- 언어별로 스톱워드가 목록화 되어 있음



## 2.4 Stemming / Lemmatization

- 시제, 단/복수를 고려하여 단어의 원형 찾기

## 3. BOW

- 문서가 가지는 모든 단어를 문맥이나 순서를 무시하고 일괄적으로 단어에 대해 빈도 값을 부여해 피쳐 값을 추출하는 모델

### 3.1 단점

- 문맥 의미 반영 부족
- 희소 행렬 문제
  -  희소 행렬 : 대부분의 값이 0으로 채워짐
  -  밀집 행렬 : 대부분의 값이 0이 아닌 의미 있는 값으로 채워짐

## 3.2 BOW 피쳐 벡터화

- 텍스트를 특정 의미를 가지는 숫자형 값인 벡터 값으로 변환
- 모든 문서에서 모든 단어를 칼럼 형태로 나열하고 각 문서에서 해당 단어의 횟수나 정규화된 빈도를 값으로 부여하는 데이터 세트 모델로 변경

M개의 텍스트 문서

N개의 값이 할당된 피쳐의 벡터 세트

M x N 개의 단어 피쳐로 이뤄진 행렬

### TF-IDF

Term Frequency - Inverse Document Frequency

- 개별 문서에서 자주 나타나는 단어에 높은 가중치를 주되, 모든 문서에서 전반적으로 자주 나타나는 단어에 대해서는 페널티를 주는 방식

### CountVectorizer

- 카운트 기반의 벡터화를 구현한 클래스
- 소문자 일괄 변환, 토큰화, 스톱 워드 필터링 등의 텍스트 전처리도 함께 수행

파라미터	
max_df	전체 문서에 걸쳐서 너무 높은 빈도수를 가지는 단어 피쳐를 제외
min_df	전체 문서에 걸쳐서 너무 낮은 빈도수를 가지는 단어 피쳐를 제외
max_features	추출하는 피쳐의 개수를 제한하며 정수로 값을 지정
stop_words	english 로 지정하면 영어의 스톱 워드로 지정된 단어는 추출에서 제외
n_gram_range	Bag of Words 모델의 단어 순서를 보강하기 위해 범위 최솟값과 최댓값을 지정
analyzer	피쳐 추출을 수행한 단위로 지정

token_pattern	토큰화를 수행하는 정규 표현식 패턴을 지정
tokenizer	토큰화를 별도의 커스텀 함수로 이용시 적용

#### 1 사전 데이터 가공

모든 문자를 소문자로 변환하는 등 사전 작업 수행

#### 2 토큰화

n\_gram\_range 반영하여 토큰화 수행

#### 3 텍스트 정규화

Stop words 필터링 수행

#### 4 피쳐 벡터화

max\_df, min\_df, max\_features 등의 파라미터를 반영하여 Token 된 단어들을 feature extraction 후 vectorization 적용

### BOW 벡터화를 위한 희소 행렬

CountVectorizer / TfidfVectorizer 이용해 텍스트를 피쳐 단위로 벡터화

## 3.3 희소행렬

### COO 형식

0이 아닌 데이터만 별도의 데이터 배열에 저장하고, 그 데이터가 가리키는 행과 열의 위치를 별도의 배열로 저장하는 방식

Scipy 의 coo\_matrix

### CSR 형식

COO 형식이 행과 열의 위치를 나타내기 위해서 반복적인 위치 데이터를 사용해야 하는 문제점을 해결

## 5. 감성 분석

문서 내 텍스트가 나타내는 여러 가지 주관적인 단어와 문맥을 기반으로 감성 수치를 계산하는 방법 이용

- 지도 학습 : 학습 데이터와 타깃 레이블 값을 기반으로 감성 분석 학습을 수행한 뒤 이를 기반으로 다른 데이터의 감성 분석을 예측하는 방법
- 비지도학습: Lexicon 이라는 감성 어휘 사전 이용하여 문서의 긍정적, 부정적 감성 여부 판단