머신러닝 - 텍스트분석

2022.04

1. 개요

❖ NLP(Natural Language Processing)

- 머신이 인간의 언어를 이해하고 해석하는 데 중점을 두고 발전
- 기계 번역, 질의 응답 시스템
- 텍스트 분석을 향상시켜주는 기반 기술

❖ 텍스트 분석(Text Analysis)

- 비정형 텍스트에서 의미있는 정보를 추출하는 것에 중점을 두고 발전
- 비즈니스 인텔리전스(BI)나 예측 분석 등의 분석 작업을 수행
- 텍스트 분류
- 감성 분석
- 텍스트 요약
- 텍스트 군집화와 유사도 측정

2. 텍스트 분석

❖ 텍스트 분석 프로세스

Text 문서

데이터 사전 가공 후
Feature Vectorization 수행

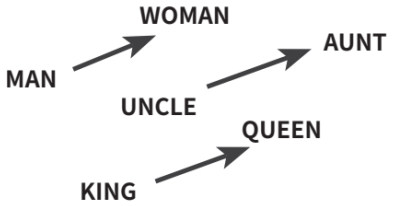
Feature Vectorization

Bag of Words

단어 #1	단어 #2	.....	단어 #n
3	4	0	1

또는

Word2Vec



Feature 기반의
데이터 세트 제공

ML 학습/예측/평가

gensim
topic modelling for humans

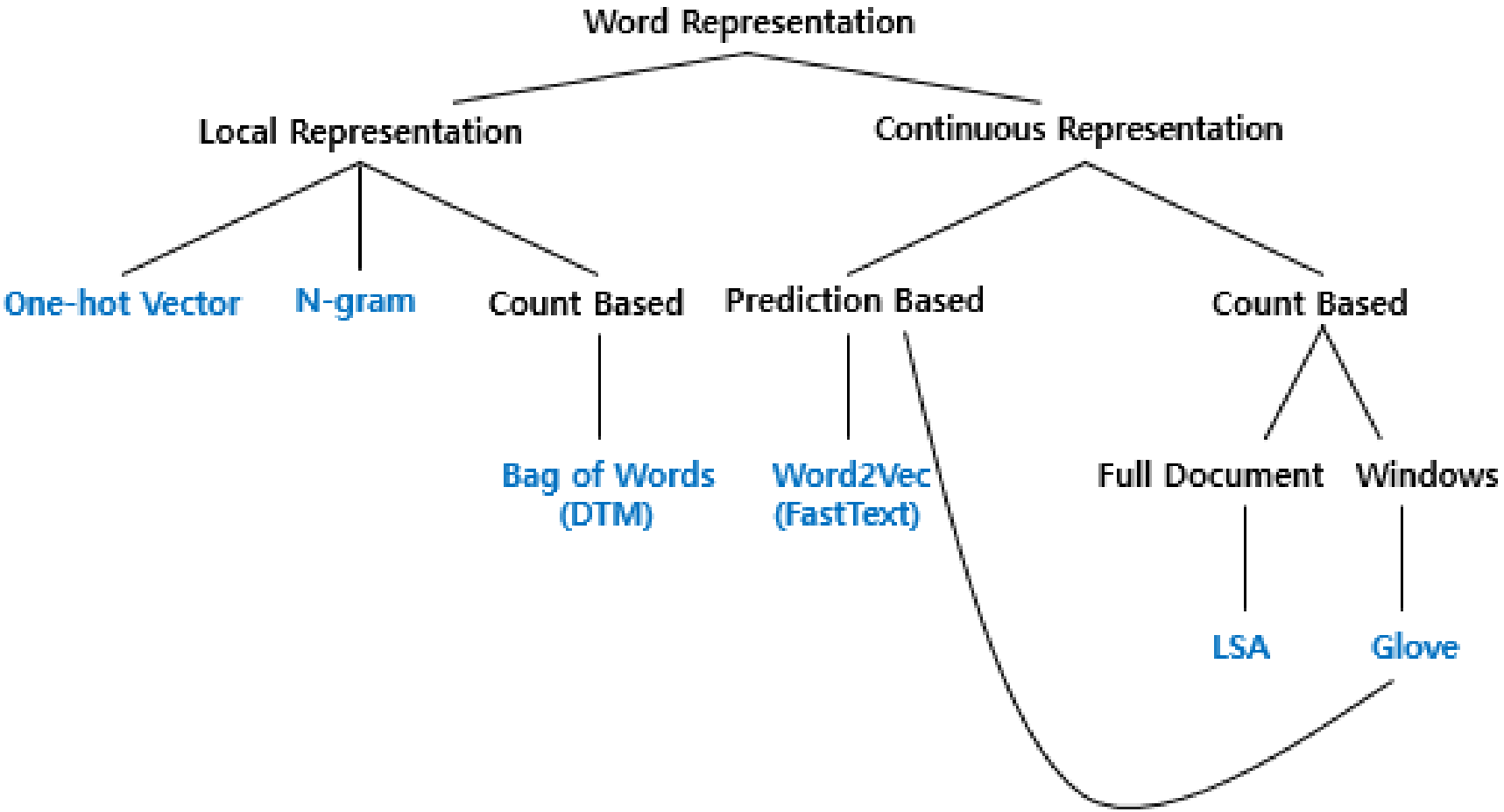
2. 텍스트 분석

❖ 텍스트 전처리

- 클렌징(Cleansing)
- 토큰화(Tokenization)
  - 문장 토큰화
  - 단어 토큰화
- 필터링, 스톱 워드(불용어) 제거, 철자 수정
- Stemming
- Lemmatization

3. Bag of Words - BOW

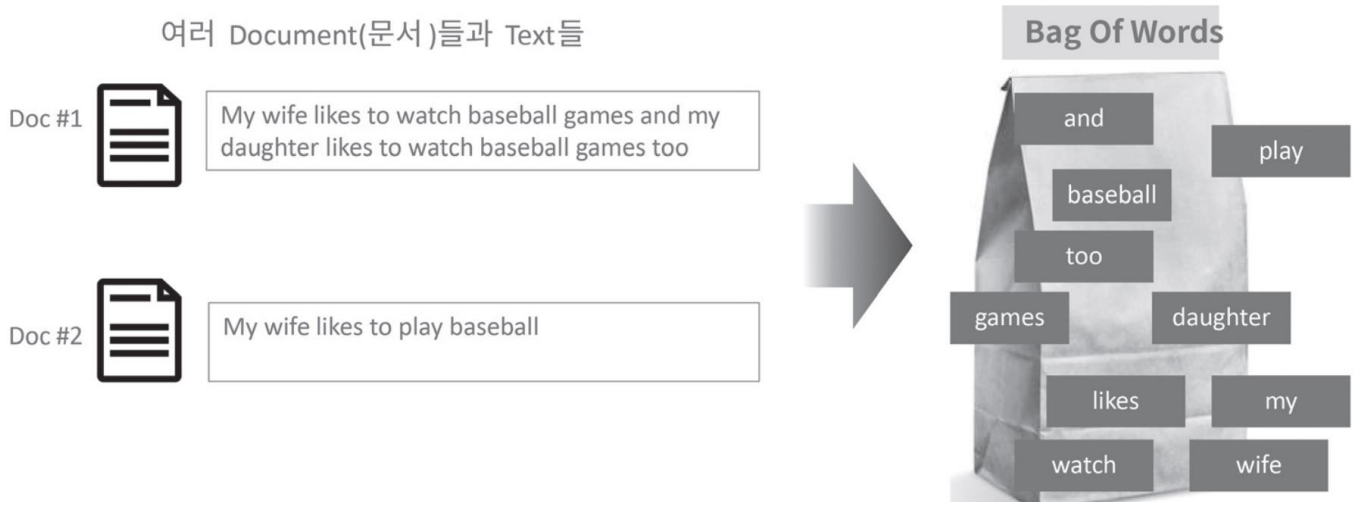
❖ 단어의 표현 방법



3. Bag of Words - BOW

❖ 개요

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



	Index 0	Index 1	Index 2	Index 3	Index 4	Index 5	Index 6	Index 7	Index 8	Index 9	Index 10
	and	baseball	daughter	games	likes	my	play	to	too	watch	wife
문장 1	1	2	1	2	2	2		2	1	2	1
문장 2		1			1	1	1	1			1

문장 1에서 baseball은 2회 나타남

3. Bag of Words - BOW

❖ 장단점

- 장점
  - 쉽고 빠른 구축
  - 의외로 높은 효율

- 단점
  - 문맥 의미(Semantic Context) 반영 부족
  - 문장의 순서가 무시됨
    - I work at google.
    - I google at work.
  - 오타, 줄임말에 취약함
  - 희소 행렬 문제

3. Bag of Words - BOW

❖ 피쳐 벡터화

- 카운트 기반의 벡터화
- TF-IDF(Term Frequency – Inverse Document Frequency) 기반의 벡터화

3. Bag of Words - BOW

❖ TF-IDF

- 개별 문서에서 자주 나타나는 단어에 높은 가중치 부여
- 모든 문서에서 자주 나타나는 단어에 대해서는 페널티를 주는 방식

한 개의 문서(Document)

모든 문서들(Corpus)

Term Frequency

The	Matrix	is	nothing	but	an	advertising	gimmick
40	5	50	12	20	45	3	2

Document Frequency

The	Matrix	is	nothing	but	an	advertising	gimmick
2000	190	2300	500	1200	3000	52	12

$$TFIDF_i = TF_i * \log\frac{N}{DF_i}$$

TF_i = 개별 문서에서의 단어 i 빈도

DF_i = 단어 i를 가지고 있는 문서 개수

N = 전체 문서 개수

3. Bag of Words - BOW

❖ Scikit-Learn CountVectorizer class

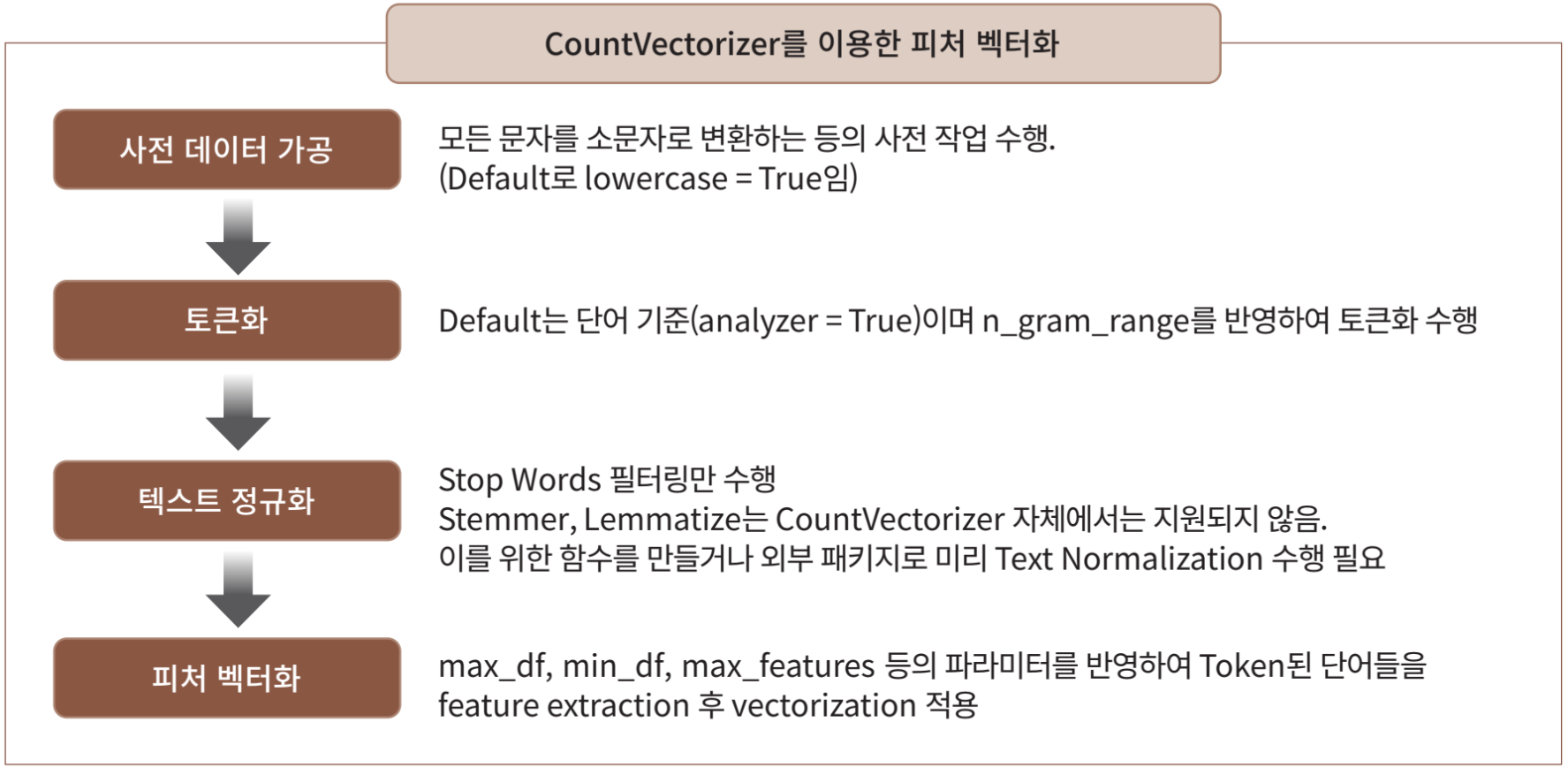
파라미터 명	파라미터 설명
max_df	전체 문서에 걸쳐서 너무 높은 빈도수를 가지는 단어 피처를 제외하기 위한 파라미터입니다. 너무 높은 빈도수를 가지는 단어는 스톱 워드와 비슷한 문법적인 특성으로 반복적인 단어일 가능성이 높기에 이를 제거하기 위해 사용됩니다. max_df = 100과 같이 정수 값을 가지면 전체 문서에 걸쳐 100개 이하로 나타나는 단어만 피처로 추출합니다. Max_df = 0.95와 같이 부동소수점 값(0.0 ~ 1.0)을 가지면 전체 문서에 걸쳐 빈도수 0~95%까지의 단어만 피처로 추출하고 나머지 상위 5%는 피처로 추출하지 않습니다.
min_df	전체 문서에 걸쳐서 너무 낮은 빈도수를 가지는 단어 피처를 제외하기 위한 파라미터입니다. 수백~수천 개의 전체 문서에서 특정 단어가 min_df에 설정된 값보다 적은 빈도수를 가진다면 이 단어는 크게 중요하지 않거나 가비지(garbage)성 단어일 확률이 높습니다. min_df = 2와 같이 정수 값을 가지면 전체 문서에 걸쳐서 2번 이하로 나타나는 단어는 피처로 추출하지 않습니다. min_df = 0.02와 같이 부동소수점 값(0.0 ~ 1.0)을 가지면 전체 문서에 걸쳐서 하위 2% 이하의 빈도수를 가지는 단어는 피처로 추출하지 않습니다.
max_features	추출하는 피처의 개수를 제한하며 정수로 값을 지정합니다. 가령 max_features = 2000으로 지정할 경우 가장 높은 빈도를 가지는 단어 순으로 정렬해 2000개까지만 피처로 추출합니다.
stop_words	'english'로 지정하면 영어의 스톱 워드로 지정된 단어는 추출에서 제외합니다.

3. Bag of Words - BOW

❖ Scikit-Learn CountVectorizer class

n_gram_range	Bag of Words 모델의 단어 순서를 어느 정도 보강하기 위한 n_gram 범위를 설정합니다. 튜플 형태로 (범위 최솟값, 범위 최댓값)을 지정합니다. 예를 들어 (1, 1)로 지정하면 토큰화된 단어를 1개씩 피처로 추출합니다. (1, 2)로 지정하면 토큰화된 단어를 1개씩(minimum 1), 그리고 순서대로 2개씩(maximum 2) 묶어서 피처로 추출합니다.
analyzer	피처 추출을 수행한 단위를 지정합니다. 당연히 디폴트는 'word'입니다. Word가 아니라 character의 특정 범위를 피처로 만드는 특정한 경우 등을 적용할 때 사용됩니다.
token_pattern	토큰화를 수행하는 정규 표현식 패턴을 지정합니다. 디폴트 값은 '\b\w\w+\b로, 공백 또는 개행 문자 등으로 구분된 단어 분리자(\b) 사이의 2문자(문자 또는 숫자, 즉 영숫자) 이상의 단어(word)를 토큰으로 분리합니다. analyzer= 'word'로 설정했을 때만 변경 가능하나 디폴트 값을 변경할 경우는 거의 발생하지 않습니다.
tokenizer	토큰화를 별도의 커스텀 함수로 이용시 적용합니다. 일반적으로 CountTokenizer 클래스에서 어근 변환 시 이를 수행하는 별도의 함수를 tokenizer 파라미터에 적용하면 됩니다.

❖ CountVectorizer를 이용한 피처 벡터화



3. Bag of Words - BOW

❖ BOW 벡터화를 위한 희소 행렬

- COO(Coordinate: 좌표) 형식
- CSR(Compressed Sparse Row) 형식

수십만 개의 칼럼 ⟷

수천 ~ 수만개 레코드

	단어 1	단어 2	단어 3	....	단어 1000	....	단어 2000	....	단어 10000	....	단어 20000	....	단어 100000
문서1	1	2	2	0	0	0	0	0	0	0	0	0	0
문서2	0	0	1	0	0	1	0	0	1	0	0	0	1
문서 ...	....	....	....	....	....	....	....	....	....	....	....	....	....
문서 10000	0	1	3	0	0	0	0	0	0	0	0	0	0

BOW의 Vectorization 모델은 너무 많은 0값이 메모리 공간에 할당되어 많은 메모리 공간이 필요하며
연산 시에도 데이터 액세스를 위한 많은 시간이 소모됩니다.

❖ n-그램

- 연속적인 n 개의 토큰으로 구성된 단어, 문자

fine thank you

- 1-gram (Unigram)
  - Word level: [fine, thank, you]
  - Character level: [f, i, n, e, , t, h, a, n, k, , y, o, u]
- 2-gram (Bigram)
  - Word level: [fine thank, thank you]
  - Character level: [fi, in, ne, e , t, th, ha, an, nk, k , y, yo, ou]
- 3-gram (Trigram)
  - Word level: [fine thank you]
  - Character level: [fin, ine, ne , e t, th, tha, han, ank, nk , k y, yo, you]

4. n-그램

❖ 사용 이유

- Bag of Words 의 약점 극복
- 다음 단어 예측
- 오타 발견
- 단어 추천

❖ Bag of Words 약점

- machine learning is fun and is not boring

machine	fun	is	learning	and	not	boring	. . .
1	1	2	1	1	1	1	. . .

  - machine learning 조합을 찾기 어려움
  - not 이 어디에 위치하는지 모름
  - "machine is boring and learning is not fun" 과 구분이 안됨

- Bag of Bigram

machine learning	learning is	is fun	fun and	and is	is not	not boring	. . .
1	1	1	1	1	1	1	. . .

❖ Naïve next word prediction

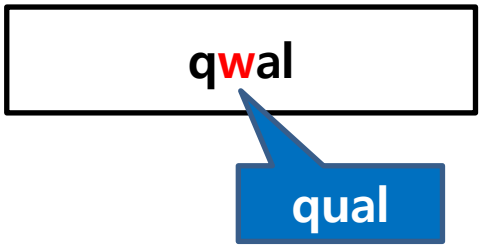
- how are you doing
- how are you
- how are they

Trigram	횟수
how are you	2
are you doing	1
how are they	1

Input box **how are** you

❖ Naïve spell checker

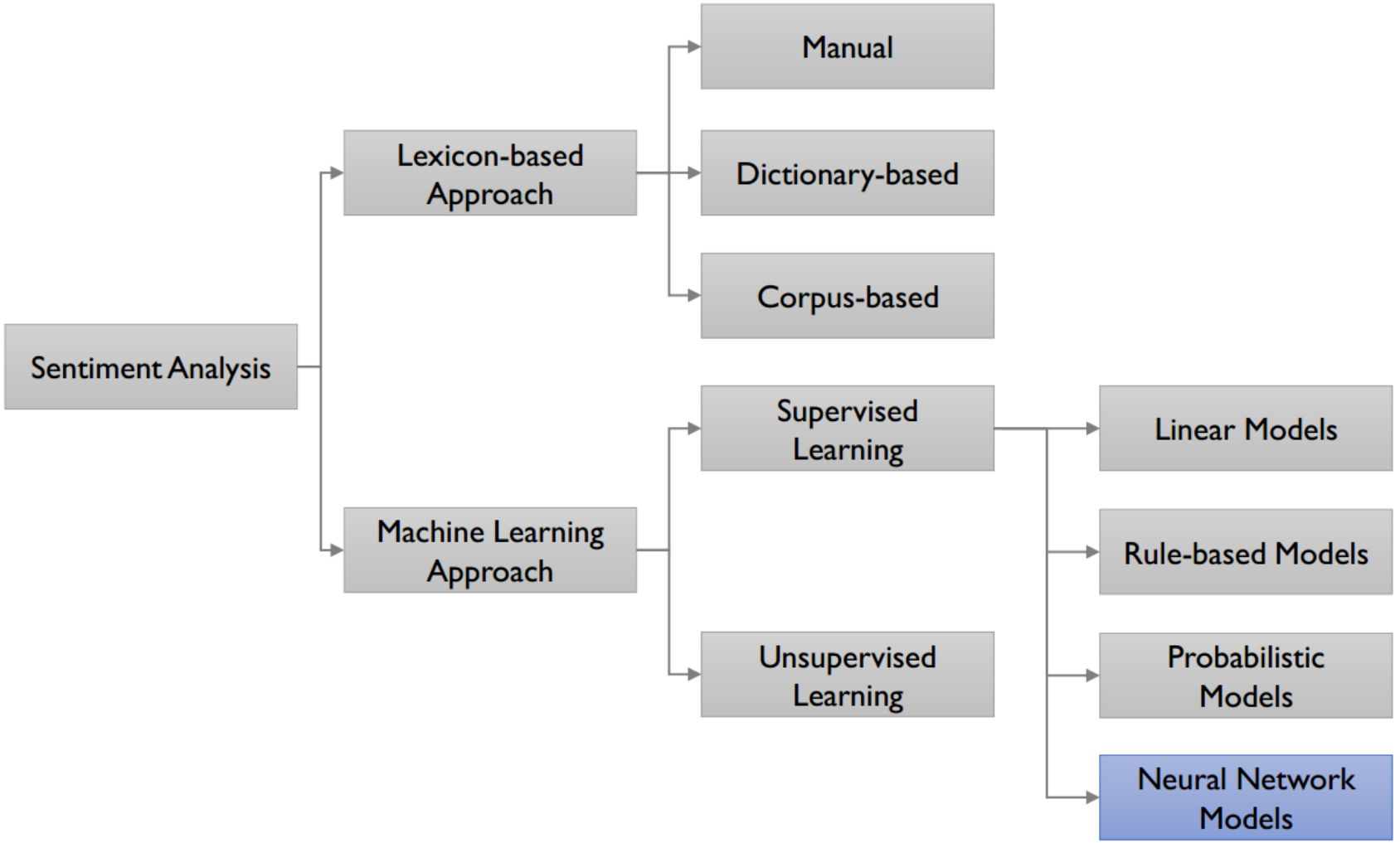
- quality
- quarter
- quit



Bigram	횟수
qu	3
ua	2
al	1
li	1
it	2
ty	1
ar	1
rt	1
te	1
er	1
ui	1

5. 감성 분석

❖ 종류


Sentiment Analysis
Lexicon-based Approach
Manual
Dictionary-based
Corpus-based
Machine Learning Approach
Supervised Learning
Linear Models
Rule-based Models
Unsupervised Learning
Probabilistic Models
Neural Network Models

❖ 지도학습 기반 감성분석 - IMDB 영화평

Bag of Words Meets Bags of Popcorn

Use Google's Word2Vec for movie reviews

578 teams · 3 years ago

Overview Data Kernels Discussion Leaderboard Rules

❖ 비지도학습 기반 감성분석 - Lexicon(어휘집) 기반

- SentiWordNet을 이용한 영화 감상평 감성 분석(비지도)
  1. 문서를 문장 단위로 분해
  2. 다시 문장을 단어 단위로 토큰화하고 품사 태깅
  3. 품사 태깅된 단어 기반으로 synset 객체와 senti_synset 객체를 생성
  4. senti_synset에서 긍정 감성/부정 감성 지수를 구하고 이를 모두 합산해 특정 임계치 값 기준으로 긍정/부정 감성을 결정

- 어휘 간의 유사도

	tree	lion	tiger	cat	dog
tree	1.00	0.07	0.07	0.08	0.12
lion	0.07	1.00	0.33	0.25	0.17
tiger	0.07	0.33	1.00	0.25	0.17
cat	0.08	0.25	0.25	1.00	0.20
dog	0.12	0.17	0.17	0.20	1.00

lion은 tree와의 유사도가 0.07로 가장 적고,
tiger와는 유사도가 0.33으로 가장 큼.

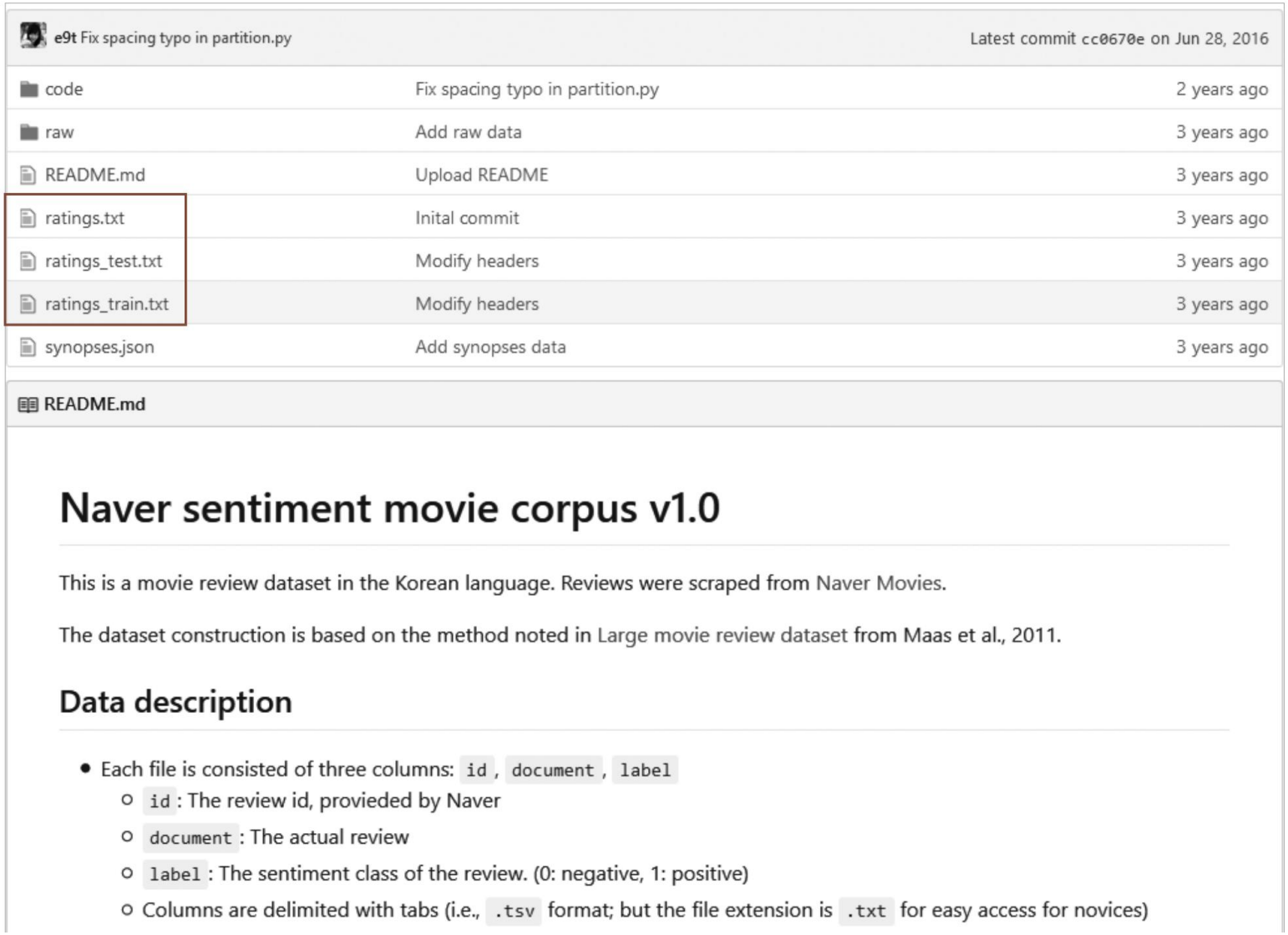
❖ 비지도학습 기반 감성분석 - VADER

- 소셜 미디어의 감성분석 용도로 만들어진 룰 기반의 Lexicon
- 성능 비교

평가 지표	정확도	정밀도	재현율
SentiWordNet	0.6613	0.6472	0.7091
VADER	0.6948	0.6485	0.8506

5. 감성 분석

❖ 네이버 영화 평점 감성 분석



e9t Fix spacing typo in partition.py — Latest commit cc0670e on Jun 28, 2016

code	Fix spacing typo in partition.py	2 years ago
raw	Add raw data	3 years ago
README.md	Upload README	3 years ago
ratings.txt	Inital commit	3 years ago
ratings_test.txt	Modify headers	3 years ago
ratings_train.txt	Modify headers	3 years ago
synopses.json	Add synopses data	3 years ago

README.md

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)

❖ 나이브 베이즈 분류기(Naïve Bayes Classifier)

▪ 장점

- 간단하고, 빠르며, 정확한 모델
- computation cost가 작음 (따라서 빠름)
- 큰 데이터셋에 적합
- 연속형보다 이산형 데이터에서 성능이 좋음
- Multiple class 예측을 위해서도 사용 가능

▪ 단점

- feature 간의 독립성이 있어야 함
  하지만 실제 데이터에서 모든 feature가 독립인 경우는 희박함

❖ 베이즈의 정리

▪ 용어 정리

- 사전 확률 : 이미 알고 있는 사건(들)의 확률
- 우도(Likelihood Probability)

  이미 알고 있는 사건(들)이 발생했다는 조건하에

  다른 사건이 발생할 확률
- 사후 확률 : 사전 확률과 우도 확률을 통해서 알게 되는 조건부 확률

▪ 베이즈 정리(Bayes Theorem)

우도 **사전확률**

사후확률

$$P(A_k|B) = \frac{P(B|A_k)P(A_k)}{P(B)}$$

주변우도(Marginal Likelihood)

❖ 한글 감성 사전 - KNU 한국어 감성사전

- 개요
  - 특정 도메인에서 사용되는 긍부정어보다는 인간의 보편적인 기본 감정 표현을 나타내는 긍부정어로 구성됨.
  - 각 도메인의 감성사전을 빠르게 구축하기 위한 기초 자료로 활용하기 위해 개발되었음
  - 본 한국어 감성사전은 다음과 같은 소스로부터 통합되어 개발되었음
    1. 국립국어원 표준국어대사전의 뜻풀이(glosses) 분석을 통한 긍부정 추출(이 방법을 통해 대부분의 긍부정어 추출)
    2. 김은영(2004)의 긍부정어 목록
    3. SentiWordNet 및 SenticNet-5.0에서 주로 사용되는 긍부정어 번역
    4. 최근 온라인에서 많이 사용되는 축약어 및 긍부정 이모티콘 목록
  - 총 14,843개의 1-gram, 2-gram, 관용구, 문형, 축약어, 이모티콘 등에 대한 긍정, 중립, 부정 판별 및 정도(degree)값 계산
- 시연: http://dilab.kunsan.ac.kr/knu/knu.html
- 다운로드: https://github.com/park1200656/KnuSentiLex

6. 토픽 모델링

❖ 20 뉴스그룹 분류

❖ 문서 군집화

▪ 개념

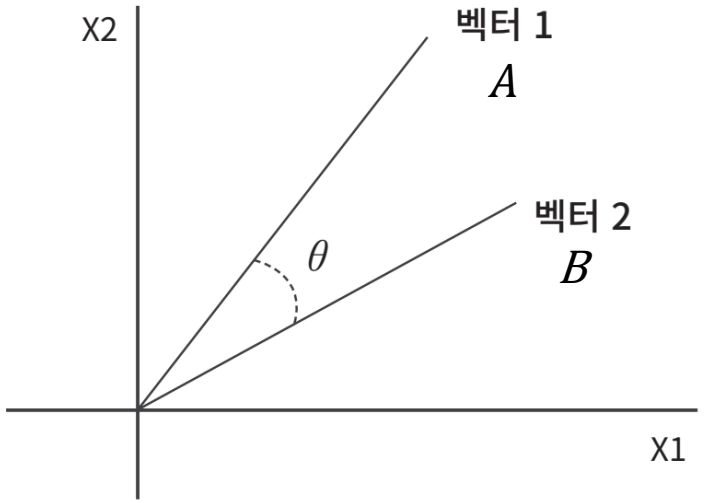
- 비슷한 텍스트 구성의 문서를 군집화하는 것
- 텍스트 분류 기반의 문서 분류와 유사
- 비지도 학습

▪ Opinion Review 데이터 세트를 이용한 문서 군집화

https://archive.ics.uci.edu/ml/datasets/Opinosis+Opinion+%26frasl%3B+Review

8. 문서 유사도

❖ 코사인 유사도



$$A \cdot B = \|A\|\|B\| \cos\theta$$