텍스트 분석

■ 텍스트 분석 기술 활용 영역

- 텍스트 분류 (Text Classification): 문서가 특정 분류 또는 카테고리에 속하는 것을 예측하는 기법

  ex) 신문기사 카테고리 분류, 스팸 메일 검출

- 감성분석 (Sentiment Analysis): 텍스트에서 나타나는 김정/판단/믿음/의견/기분 등의 주관적인 요소를 분석하는 기법

  ex) 소셜 미디어 감성 분석, 영화/제품에 대한 리뷰 분석

- 텍스트 요약 (Summarization): 텍스트 내에서 중요한 주제나 중심 사상을 추출하는 기법

  ex) 토픽 모델링

텍스트 분석

■ 파이썬 기반의 텍스트 분석 패키지

- 영어 기반 패키지
  - NLTK(Natural Language Toolkit for Python): 파이썬의 가장 대표적인 NLP 패키지로 NLP의 거의 모든 영역을 커버하고 있다. 단, 수행속도에 있어서 다소 아쉬움이 있음
  - Gensim: 토픽 모델링 분야에서 가장 두각을 나타내는 패키지
  - SpaCy: 뛰어난 수행 성능으로 최근 가장 주목을 받는 NLP 패키지, 영어를 포함한 8개 국어에 대한 자연어 전처리 모듈 제공 (상업용 패키지)

- 한글 기반 패키지
  - KoNLPy: 파이썬의 대표적인 한글 형태소 패키지

텍스트 분석

■ 텍스트 데이터 처리 절차

- 전처리 (정형화 과정)

✓ 텍스트 토큰화: 토큰화의 유형은 문서에서 문장을 분리하는 **문장 토큰화**와 문장에서 단어를 토큰으로 분리하는 **단어 토큰화**로 나뉜다.

- 문장 토큰화는 문자의 마침표(.), 개행문자(\n) 등 문장의 마지막을 뜻하는 기호에 따라 분리한다.
- 문장 토큰화는 각 문장이 가지는 의미가 중요한 요소로 사용될 때 사용한다.
- 단어 토큰화는 일반적으로 공백, 콤마(,), 마침표(.), 개행문자(\n) 등으로 문장을 단어로 토큰화 하는 것이다.

✓ 한글의 토큰화: 영어는 띄어쓰기를 기준으로 토큰화를 시도해도 단어 토큰화가 잘 작동한다. 하지만 한국어는 영어와는 달리 띄어쓰기만으로 토큰화를 하기에는 문제가 있다.

- 한글은 조사, 어미 등을 붙여서 말을 만드는 교착어이다.
- '그'라는 단어 하나에도 '그가', '그에게', '그를', '그와' 같이 다양한 조사가 '그'라는 글자 뒤에 붙게 된다. 따라서 한국어 NLP에서는 조사를 분리해줄 필요가 있다.
- 한국어 토큰화는 **형태소분석**을 통해 이뤄진다. (형태소란 뜻을 가진 가장 작은 말의 단위를 의미)

텍스트 분석

■ 텍스트 데이터 처리 절차

- 전처리 (정형화 과정)
  - ✓ 대소문자 통일: 한국어 텍스트에는 해당되지 않음
    - 파이썬의 lower(), upper() 함수 이용
  - ✓ 숫자, 문장부호, 특수문자 제거
    - 정규표현식 이용
  - ✓ 불용어 제거: ‘the’, ‘a’, ‘an’과 같은 관사 (구체적인 의미를 찾기 어려운 단어)
    - nltk 패키지의 stopwords 리스트 이용 (한국어는 지원하지 않아 별도의 불용어 리스트를 제작하여 이용)

텍스트 분석

■ 텍스트 데이터 처리 절차

- 품사 분석
  - ✓ 어근 동일화: 한국어는 어미, 조사에 따라 단어의 형태가 바뀌고, 영어는 주어의 형태, 시제에 따라 동사의 형태가 바뀌며 단수, 복수에 따라서도 단어의 형태가 바뀐다. 즉, **동일한 의미의 단어들을 같은 형태로 통일**한다.
    - NLTK 패키지의 PorterStemmer 라이브러리 이용
    - 한국어에는 별도 라이브러리가 존재하지 않아서 형태소 분석기를 사용해야 함
  - ✓ 품사 분석은 Part-Of-Speech의 앞 글자를 따서 **POS 태깅**이라고 부른다.
  - ✓ 동일한 단어라도 서로 다른 품사 분석을 통해 더욱 심층적인 의미를 발견해낼 수 있다.
  - ✓ Ex) Love나 Presents 라는 단어는 동사로 쓰인 경우와 명사로 쓰인 경우로 나눌 수 있다.
  - ✓ 한국어 품사 분석을 위해서는 KoNLPy 패키지를 사용한다.

텍스트 분석

■ Bag of Words (BoW)

- Bag of Words란 단어들의 문맥이나 순서를 고려하지 않고, 일괄적으로 단어에 대한 빈도 값을 부여해 피처 값을 추출하는 모델이다.
- BoW를 만드는 과정
  ① 각 단어에 고유한 정수 인덱스를 부여한다.
  ② 각 인덱스의 위치에 단어 토큰의 등장 횟수를 기록한 벡터를 만든다.
- BoW의 장단점
  ✓ 장점: 쉽고 빠른 구축, 문서의 특징을 잘 나타낼 수 있어 여러 분야에서 활용도가 높음
  ✓ 단점: 문맥 의미 반영 부족 (단어의 순서를 고려하지 않기 때문에 문장 내에서 단어의 문맥적인 의미가 무시됨), 희소행렬 문제
- BoW의 피처 벡터화 방식
  ✓ 카운트 기반의 벡터화: 문서에서 해당 단어가 나타나는 횟수 벡터화. 언어의 특성상 문장에서 자주 사용될 수밖에 없는 단어까지 높은 값을 부여하게 되는 단점
  ✓ TF-IDF(Term Frequency-Inverse Document Frequency): 개별 문서에서 자주 나타나는 단어에 높은 가중치를 주되, 모든 문서에서 전반적으로 자주 나타나는 단어에 대해서는 페널티를 부여하는 방식

텍스트 분석

■ Bag of Words (BoW)

- DTM(Document-Term Matrix, 문서 단어 행렬)
  - 다수의 문서에서 등장하는 각 단어들의 빈도를 행렬로 표현한 것
  - 사이킷런의 CounterVectorizer를 사용하여 생성할 수 있다.

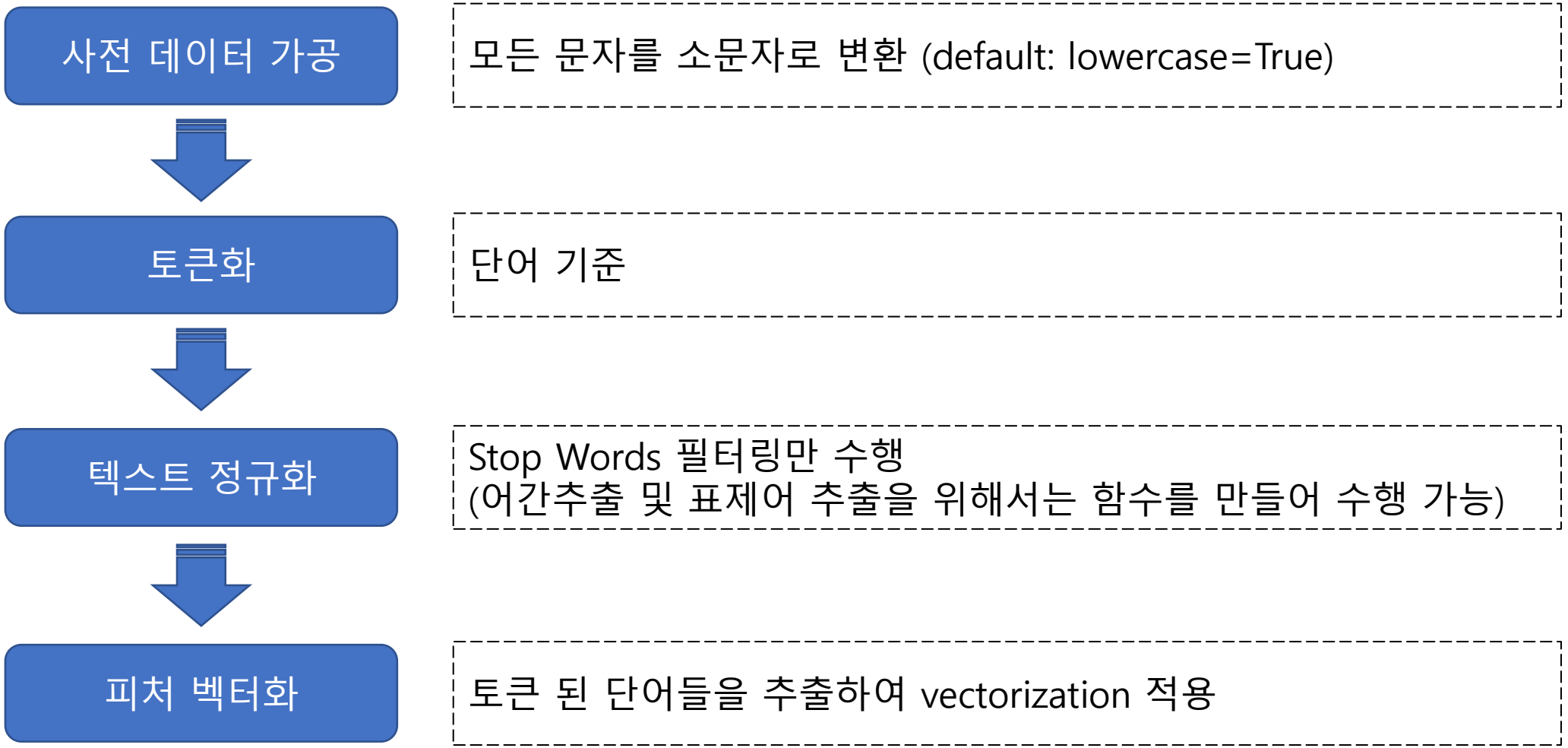
문서1: 먹고 싶은 사과
문서2: 먹고 싶은 바나나
문서3: 길고 노란 바나나 바나나
문서4: 저는 과일이 좋아요

	과일이	길고	노란	먹고	바나나	사과	싶은	저는	좋아요
문서1	0	0	0	1	0	1	1	0	0
문서2	0	0	0	1	1	0	1	0	0
문서3	0	1	1	0	2	0	0	0	0
문서4	1	0	0	0	0	0	0	1	1

텍스트 분석

■ Bag of Words (BoW)

- CounterVectorizer를 이용한 피처 벡터화



텍스트 분석

■ Bag of Words (BoW)

- TF-IDF(Term Frequency-Inverse Document Frequency)
  - TF-IDF는 TF와 IDF를 곱한 값을 의미한다.
  - TF-IDF는 단어의 빈도와 역 문서 빈도를 사용하여 DTM 내의 각 단어들마다 중요한 정도를 가중치로 주는 방식
  - 즉, 개별 문서에서 자주 나타나는 단어에 높은 가중치를 주되, **모든 문서에서 전반적으로 자주 등장하는 단어에 대해서는 페널티**를 준다.
  - 주로 문서의 유사도를 구하거나, 검색 시스템에서 검색 결과의 중요도를 정하는 작업에 활용된다.

※ d: 문서, t:단어, n:문서의 총 개수

(1) tf(d,t): 특정 문서 d에서 특정 단어 t의 등장 횟수

(2) df(t): 특정 단어 t가 등장한 문서의 수

(3) idf(d,t): df(t)에 반비례하는 수 ⟶ $idf(d,t) = log_e\left(\frac{1+n}{1+df(t)}\right)$

※ 총 문서의 수(n)가 커질수록 IDF의 값이 기하급수적으로 커지는 것을 방지하기 위해 log를 취한다.

$$TF - IDF = TF \times IDF$$

텍스트 분석

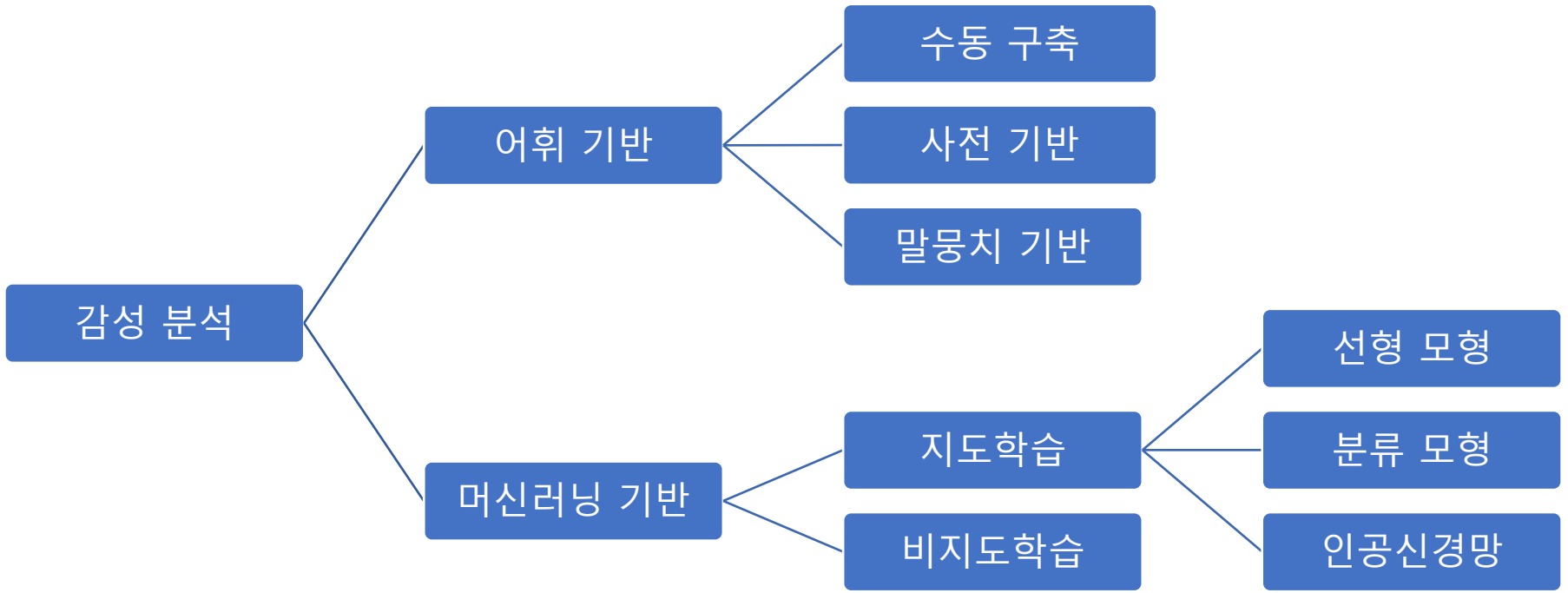
■ Bag Of Words (BOW)

- 피처 벡터와 / 추출
  - ✓ N-gram: n개의 연속적인 단어 나열을 의미
    - N이 1일 때는 유니그램, 2일 때는 바이그램, 3일 때는 트라이그램이라고 명명한다.
    - 복합 단어로서 한 단어의 의미를 갖는 단어를 찾기 위해 사용
    - Ex) ‘Republic of Korea’ 같은 경우에는 N-gram을 이용해야 제대로 된 단어 객체로 인지할 수 있다.
    - N-gram을 과도하게 적용하면 의미 없는 단어 뭉치가 많이 발생하여 불필요한 작업이 될 수 있다.

텍스트 분석

■ 감성 분석 (Sentiment Analysis)

- 감성 분석이란 텍스트에 들어있는 의견이나 감성, 의견, 감정 등의 주관적인 정보를 분석하는 방법이다.
- 여론조사, 온라인 리뷰, 피드백 분석 등의 분야에 활용된다.
- 감성 분석의 방법론은 크게 어휘 기반(Lexicon-based)의 감성 분석과 머신러닝 기반(ML-based)의 감성 분석으로 나눌 수 있다.

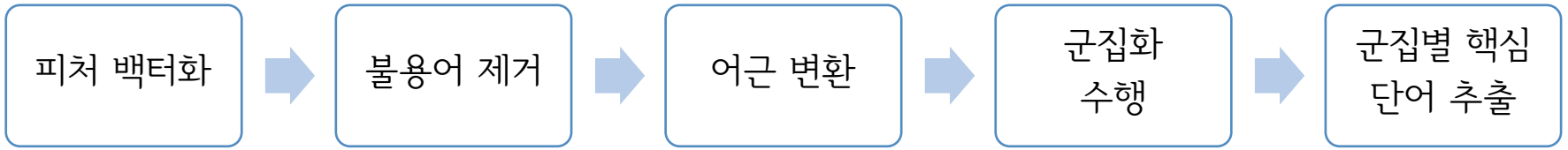


텍스트 분석

■ 문서 군집화(Document Clustering)

- 문서 군집화는 비슷한 텍스트 구성의 문서를 군집화(Clustering)하는 것이다.
- 문서 군집화는 학습 데이터가 필요 없는 비지도학습 기반의 텍스트 분류이다.

■ 문서 군집화 절차



텍스트 분석

■ 한글 텍스트 처리

- 한글의 자연어 처리는 띄어쓰기나 다양한 조사 사용으로 영어 자연어 처리보다 어렵다

■ KoNLPy

- 파이썬의 대표적인 한글 형태소 패키지 (※ 형태소: 단어로서 의미를 가지는 최소 단위)
- 공식 홈페이지: https://konlpy.org/ko/latest/
- 형태소분석: https://konlpy.org/ko/v0.5.2/morph/

■ 설치 방법 (Windows 기준)

① 윈도우 비트 수와 일치한 파이썬 설치

② 윈도우와 비트 수가 일치하고, 버전 1.7 이상의 JDK 설치 → JAVA_HOME 환경변수 설정

③ 윈도우와 비트 수가 일치하는 JPype 설치

: https://www.lfd.uci.edu/~gohlke/pythonlibs/#jpype 에서 JPype 라이브러리 다운로드

> pip install --upgrade pip

> pip install JPype1-1.4.0-cp310-cp310-win_amd64.whl

④ KoNLPy 설치 → >pip install konlpy

텍스트 분석

■ KNU 한국어 감성 사전을 이용한 감성 분석

- 군산대학교에서 구축한 한국어 감성 사전
- 특정 도메인에서 사용되는 긍부정어보다는 인간의 보편적인 기본 감정 표현을 나타내는 긍부정어로 구성
- 감성 사전 데이터: https://github.com/park1200656/KnuSentiLex

■ KNU 한국어 감성 사전 특징

- 표준국어대사전을 구성하는 각 단어의 뜻풀이를 분석하여 긍부정어를 추출하였음
- 1-gram, 2-gram, n-gram(어구, 문형), 축약어, 이모티콘 등의 다양한 종류의 긍부정어 포함

  ※ 축약어, 이모티콘까지 사전으로 구축되어 있어 텍스트 정제 및 표준화를 하지 않는 것이 더 정확한 감성을 계산할 수 있다.
- 영화, 음악, 자동차 등 어떤 도메인에도 사용될 수 있는 보편적인 긍부정어로 구성

텍스트 분석

■ Mecab 형태소 분석기 설치

① Mecab 설치 폴더 생성 : C:/mecab (윈도우에 설치하기 위한 기본 path가 고정되어 있음)

② Mecab msvc 설치 (c 기반인 mecab이 윈도우에서 실행 될 수 있도록 해주는 프로그램)

: 운영체제 버전에 맞는 패키지 다운로드 (https://github.com/Pusnow/mecab-ko-msvc/releases)

: 다운 받은 zip 파일을 mecab 폴더에 폴더를 생성하지 않고 곧바로 압축 해제

③ Mecab dic 설치

: mecab-ko-dic-msvc.zip 다운로드 (https://github.com/Pusnow/mecab-ko-dic-msvc/releases)

: 다운 받은 zip 파일을 mecab 폴더에 폴더를 생성하지 않고 곧바로 압축 해제

④ Mecab wheel 패키지 설치

: 파이썬과 운영체제 버전에 맞는 wheel 패키지 다운로드 (https://github.com/Pusnow/mecab-python-msvc/releases)

: 설치 > pip install .\mecab_python-0.996_ko_0.9.2_msvc-cp39-cp39-win_amd64.whl