Chapter 08. 텍스트 분석

NLP이냐 텍스트 분석이냐?

- NLP

머신이 인간의 언어를 이해하고 해석 (ex. 기계 번역, 질의응답 시스템)

텍스트 분석을 향상하게 하는 기반 기술

- 텍스트 분석 / 텍스트 마이닝(Text Mining)

비정형 텍스트에서 의미 있는 정보를 추출

머신러닝, 언어 이해, 통계 등을 활용해 모델을 수립하고 정보를 추출해 비즈니스 인텔리전스(Business Intelligence)나 예측 분석 등의 분석

  - 텍스트 분류(Text Classification / Text Categorization)
    - 문서가 특정 분류 또는 카테고리에 속하는 것을 예측하는 기법 (ex. 기사 유형 분류, 스팸 메일 검출)
    - 지도학습 적용
  - 감성 분석(Sentiment Analysis)
    - 텍스트에서 주관적인 요소를 분석하는 기법 (ex. 소셜 미디어 감정 분석, 영화나 제품에 대한 리뷰, 여론조사 의견)
    - 텍스트 분석에서 가장 활발하게 사용되고 있는 분야
    - 지도학습, 비지도학습 적용
  - 텍스트 요약(Summarization)
    - 텍스트 내에서 중요한 주제나 중심 사상을 추출하는 기법 (ex. 토픽 모델링(Topic Modeling))
  - 텍스트 군집화(Clustering)와 유사도 측정
    - 비슷한 유형의 문서에 대해 군집화를 수행하는 기법
    - 텍스트 분류를 비지도학습으로 수행

    - 문서들간의 유사도를 측정해 비슷한 문서끼리 모을 수 있는 방법

8.1 텍스트 분석 이해

- 텍스트 분석: 비정형 데이터(텍스트)를 분석하는 것
- ML 모델: 정형 데이터 기반에서 모델 수립 및 예측 수행

→ 피처 벡터화(Feature Vectorization) or 피처 추출(Feature Extraction)

- 머신러닝 알고리즘은 숫자형의 피처 기반 데이터만 입력 받을 수 있음
- 텍스트를 머신러닝에 적용하기 위해서는
    - 비정형 텍스트 데이터를 어떻게 피처 형태로 추출하고,
    - 추출된 피처에 의미 있는 값을 부여하는가

  하는 것이 매우 중요한 요소
- 머신러닝 모델 적용 전 수행해야 할 요소
    - 텍스트를 word 기반의 다수의 피처로 추출
    - 이 피처에 단어 빈도수와 같은 숫자 값 부여
    - 텍스트는 단어의 조합인 벡터값으로 표현 가능
- 대표적인 티처 벡터화 변환 방법
    - BOW(Bag of Words)
    - Word2Vec

텍스트 분석 수행 프로세스

1. 텍스트 사전 준비작업(텍스트 전처리)
    - 텍스트를 피처로 만들기 전 단계
    - 예시
        - 대/소문자 변경, 특수문제 삭제 등의 클렌징

    - 단어(Word) 등의 토큰화 작업
    - 의미 없는 단어(Stop word) 제거
    - 어근 추출(Stemming/Lemmatization) 등의 텍스트 정규화 작업
2. 피처 벡터화/추출
    - 사전 준비 작업. 가공된 텍스트에서 피처를 추출하고 벡터 값 할당
    - BOW와 Word2Vec 방법이 있으며, BOW는 대표적으로 Count 기반과 TF-IDF 기반 벡터화가 있음
3. ML 모델 수립 및 학습/예측/평가
    - 피처 벡터화된 데이터 세트에 ML 모델을 적용해 학습/예측 및 평가를 수행

파이썬 기반의 NLP, 텍스트 분석 패키지

텍스트 사전 정제 작업, 피처 벡터화/추출, ML 모델을 지원하는 대표적인 라이브러리

- NTLK(Natural Language Toolkit for Python)
    - 파이썬의 가장 대표적인 NLP 패키지
    - 방대한 데이터 세트와 서브 모듈을 가지고 있으며 NLP의 거의 모든 영역을 커버함
    - 많은 NLP 패키지가 NLTK의 영향을 받아 작성되고 있음.
    - 수행속도 측면에서 아쉬운 부분이 있어 실제 대량의 데이터 기반에서는 제대로 활용되지 못하고 있음
- Gensim
    - 토픽 모델링 분야에서 가장 두각을 나타내는 패키지.
    - 오래전부터 토픽 모델링을 쉽게 구현할 수 있는 기능 제공.
    - SpaCy와 함께 가장 많이 사용되는 NLP 패키지
- SpaCy
    - 뛰어난 수행 성능으로 최근 가장 주목을 받는 NLP 패키지.
    - NLP 애플리케이션에서 SpaCy를 사용하는 사례가 늘고 있음

사이킷런은 머신러닝 위주의 라이브러리여서 NLP 패키지에 특화된 라이브러리는 가지고 있지 않지만 텍스트 데이터를 피처로 처리하기 위한 편리한 기능을 제공해 충분히 텍스트 분석 기능을 수행 할 수 있음

다양한 텍스트 분석이 적용돼야 하는 경우, 보통 NLTK/Gensim/SpaCy와 같은 NLP 전용 패키지와 함께 결합해 애플리케이션을 작성하는 경우가 많음

8.2 텍스트 사전 준비 작업(텍스트 전처리) - 텍스트 정규화

텍스트 정규화

텍스트를 머신러닝 알고리즘이나 NLP 애플리케이션에 입력 데이터로 사용하기 위해 다양한 텍스트 데이터의 사전 작업을 수행하는 것

- 클렌징(Cleansing)
- 토큰화(Tokenization)
- 필터링/스톱 워드 제거/철자 수정
- Stemming
- Lemmatization

클렌징

텍스트에서 분석에 방해가 되는 불필요한 문자, 기호 등을 사전에 제거하는 작업. (ex. HTML, XML 태그나 특정 기호 등)

텍스트 토큰화

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

문장 토큰화

- 문장의 마침표(.), 개행문자(\n) 등 문장의 마지막을 뜻하는 기호에 따라 분리
- 정규 표현식에 따른 문장 토큰화도 가능
- `sent_tokenize()` : 각각의 문장으로 구성된 list 객체 반환

단어 토큰화

- 문장을 단어로 토큰화하는 것
- 기본적으로 공백, 콤마(,), 마침표(.), 개행문자 등으로 단어 분리
- 정규 표현식을 이용하여 다양한 유형으로 토큰화 수행 가능
- `word_tokenize()`

n-gram

- 연속된 n개의 단어를 하나의 토큰화 단위로 분리해내는 것
- n개 단어 크기 윈도우를 만들어 문장의 처음부터 오른쪽으로 움직이면서 토큰화 수행

스톱 워드 제거

- 스톱 워드: 분석에 큰 의미가 없는 단어 (ex. is, the, a, will 등 필수 문법 요소지만 문맥적으로 큰 의미가 없는 단어)
- 문법적인 특성으로 인해 스톱 워드가 텍스트에 빈번하게 나타나 사전에 제거하지 않으면 중요한 단어로 인지될 수 있음
- NLTK 에서 다양한 언어의 스톱 워드 목록을 제공

Stemming과 Lemmatization

많은 언어에서 문법적인 요소에 따라 단어가 다양하게 변하기 때문에 문법적 또는 의미적으로 변화하는 단어의 원형을 찾는 과정이 필요함

- Stemming
  - 원형 단어로 변환 시 일반적인 방법을 적용하거나 더 단순화된 방법을 적용
  - 원래 단어에서 일부 철자가 훼손된 어근 단어를 추출하는 경향
  - `LancasterStemmer()`
- Lemmatization
  - Stemming보다 정교하며, 의미론적인 기반에서 단어의 원형을 찾음
  - 품사와 같은 문법적인 효소와 더 의미적인 부분을 감안해 정확한 철자로 된 어근 단어를 찾아줌
  - 변환에 더 오랜 시간이 걸림
  - `WordNetLemmatizer()` : 단어의 품사를 함께 입력해줘야 함

8.3 Bag of Words - BOW

Bag of Words(BOW) 모델

- 문서가 가지는 모든 단어를 문맥이나 순서를 무시하고 일괄적으로 단어에 대해 빈도 값을 부여해 피처 값을 추출하는 모델
- 프로세스
  1. 2개의 문장(문장 1, 문장 2)이 있다고 가정
  2. 두 문장에 있는 모든 단어에서 중복 제거, 각 단어를 칼럼 형태로 나열
  3. 각 단어에 고유의 인덱스 부여
  4. 개별 문장에서 해당 단어가 나타나는 횟수를 각 단어(단어 인덱스)에 기재
- 장점
  - 쉽고 빠른 구축
  - 단순히 단어의 발생 횟수에 기반하지만, 예상보다 문서의 특징을 잘 나타낼 수 있는 모델 → 여러 분야에서 활용도가 높음
- 단점
  - 문맥 의미(Semantic Context) 반영 부족
  - 희소 행렬 문제(희소성, 희소 행렬)

BOW 피처 벡터화

1. 각 문서의 텍스트를 단어로 추출해 피처로 할당
2. 각 단어의 발생 빈도와 같은 값을 해당 피처에 부여해 벡터로 생성

<u>카운트 기반의 벡터화</u>

- 단어 피처에 값을 부여할 때 각 문서에서 해당 단어가 나타나는 횟수(Count)를 부여하는 경우
- 카운트 값이 높을수록 중요한 단어로 인식

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

- 빈도수 - 중요도 문제를 보완하고자 도입
- 개별 문서에서 자주 나타나는 단어에 높은 가중치를 주되, 모든 문서에서 전반적으로 자주 나타나는 단어에 대해서는 패널티를 주는 방식으로 값 부여
- 텍스트가 길고 문서의 개수가 많은 경우 카운트 방식보다는 TF-IDF 방식을 사용하는 것이 더 좋은 예측 성능을 보장

사이킷런의 Count 및 TF-IDF 벡터화 구현: CountVectorizer, TfidfVectorizer

- `CountVectorizer`
- `TfidfVectorizer`

입력 파라미터 p. 502, 503 참고

BOW 벡터화를 위한 희소 행렬

희소행렬

- 대규모 행렬의 대부분의 값을 0이 차지하는 행렬
- BOW 형태를 가진 언어 모델의 피처 벡터화는 대부분 희소 행렬
- 메모리 공간이 많이 필요, 행렬의 크기가 커서 연산 시 데이터 액세스를 위한 시간이 많이 소모됨

희소 행렬 - COO(Coordinate: 좌표) 형식

- 0이 아닌 데이터만 별도의 데이터 배열에 저장하고, 그 데이터가 가리키는 행과 열의 위치를 별도의 배열로 저장하는 방식
- 사이파이(Scipy)의 sparse 패키지 `coo_matrix`

희소 행렬 - CSR(Compressed Sparse Row) 형식

- COO 형식이 행과 열의 위치를 나타내기 위해 반복적인 위치 데이터를 사용해야 하는 문제점을 해결한 방식
- 사이파이(Scipy)의 sparse 패키지 `csr_matrix`

8.4 텍스트 분류 실습 - 20 뉴스그룹 분류

사이킷런 내부 예제 데이터 fetch_20newsgroups() 를 이용해 텍스트 분류를 적용

텍스트 정규화

피처 벡터화 변환과 머신러닝 모델 학습/예측/평가

사이킷런 파이프라인(Pipeline) 사용 및 GridSearchCV와의 결합

실습 코드 참고

8.5 감성 분석

감성 분석(Sentiment Analysis) 소개

- 문서의 주관적인 감성/의견/감정/기분 등을 파악하기 위한 방법
- 문서 내 텍스트가 나타내는 여러가지 주관적인 단어와 문맥을 기반으로 감성 수치를 계산하는 방법 이용
    - 감성 지수는 긍정 감성 지수와 부정 감성 지수로 구성
    - 이 지수를 합산해 긍정 감성 또는 부정 감성을 결정
- 지도학습과 비지도학습 방식으로 나눌 수 있음
    - 지도학습: 학습 데이터와 타깃 레이블 값을 기반으로 감성 분석 학습을 수행한 뒤 이를 기반으로 다른 데이터의 감성분석을 예측하는 방법
    - 비지도학습: ‘Lexicon’이라는 일종의 감성 어휘 사전(감성 분석을 위한 용어와 문맥에 대한 다양한 정보)을 이용하여 문서의 긍정적, 부정적 감성 여부를 판단

지도학습 기반 감성 분석 실습 - IMDB 영화평

영화평의 텍스트를 분석해 감성 분석 결과가 긍정 또는 부정인지를 예측하는 모델 만들기

실습 코드 참고

비지도학습 기반 감성 분석 소개

- NLP 패키지의 WordNet 모듈: 방대한 영어 어휘 사전. 단순한 어휘 사전이 아닌 시맨틱(semantic, 문맥상 의미) 분석을 제공하는 어휘 사전
- 비지도 감성 분석은 Lexicon을 기반으로 함
- NLTK를 포함한 대표적인 감성 사전
    - SentiWordNet: NLTK 패키지 WordNet의 Synset 개념을 감성 분석에 적용한 것으로 3가지 감성 점수(긍정 감성 지수, 부정 감성 지수, 객관성 지수)를 할당
    - VADER: 주로 소셜 미디어의 텍스트에 대한 감성 분석을 제공하기 위한 패키지로 뛰어난 감성 분석 결과를 제공하고 비교적 빠른 수행 시간을 보장해 대용량 텍스트 데이터에 잘 사용됨
    - Pattern: 예측 성능 측면에서 가장 주목받는 패키지

SentiWordNet을 이용한 감성 분석

WordNet Synset과 SentiWordNet SentiSynset 클래스의 이해

SentiWordNet을 이용한 영화 감상평 감성 분석

실습 코드 참고

VADER를 이용한 감성 분석

실습 코드 참고