NLP : 머신이 인간의 언어를 이해하고 해석하는데 더 중점을 두고 기술이 발전.

텍스트 마이닝 : 비정형 텍스트에서 의미있는 정보를 추출하는 것에 더 중점을 두고 기술이 발전.

ex, 기계 번역, 질의응답 시스템
NLP : 텍스트 분석을 향상학 p하는 기반 기술.
NLP 기술이 발전함에 따라 텍스트 분석도 더욱 정교하게 발전할 수 있었음.
NLP와 텍스트 분석의 발전 근간에는 머신러닝이 존재함.
예전의 텍스트를 구성하는 언어적인 룰이나 업무의 룰에 따라 텍스트를 분석하는 룰 기반 시스템에서 머신러닝의 텍스트 데이터를 기반으로 모델을 학습하고 예측하는 기반으로 변경되면서 많은 기술적 발전이 가능해짐.

텍스트 분석은 머신러닝, 언어 이해 ,통계 등을 활용해 모델을 수립하고 정보를 추출해 ㅂ즈니스 인텔리전스나 예측 분석 등의 분석 작업을 주로 수행함.

● 텍스트 분류
: 문서가 특정 분류 또는 카테고리에 속하는 것을 예측하는 기법을 통칭함. 지도학습을 적요함.

● 감성 분석
: 텍스트에서 나타나는 감정/ 판단/ 믿음/ 의견/ 기분 등의 주관적인 요소를 분석하는 기법을 총칭함. 지도학습 방법뿐만 아니라 비지도학습을 이용해 적용할 수 있음.

● 텍스트 요약
: 텍스트 내에서 중요한 주제나 중심 사상을 추출하는 기법을 말함.

● 텍스트 군집화와 유사도 측정
: 비슷한 유형의 문서에 대해 군집화를 수행하는 기법을 말함. 텍스트 분류를 비지도학습으로 수행한ssm 방법의 일환으로 사용될 수 있음. 유사도 측정 역시 문서들간의 유사도를 측정해 비슷한 문서끼리 모을 수 있는 방법임.

[1] 텍스트 분석 이해

텍스트 분석은 비정형 데이터인 텍스트를 분석하는 것.
머신러닝 알고리즘은 숫자형의 피처 기반 데이터만 입력받을 수 있기 때문에 텍스트를 머신러닝에 적용하기 위해서는 비정형 텍스트 데이터를 어떻게 피처 형태로 추출하고 추출된 피처에 의미 있는 값을 부여하는가 하는 것이 매우 중요한 요소임

피처 벡터화/피처 추출 : 텍스트를 word 기반의 다수의 피처로 추출하고 이 피처에 단어 빈도수와 같은 숫자 값을 부여하면 텍스트는 단어의 조합인 벡터값으로 표현될 수 있음.

ex. Bow, Word2Vec 방법

** 텍스트를 벡터 값을 가지는 피처로 변환하는 것은 ㅓ신러닝 모델을적용하기 전에 수행해야 할 매우 중요한 요소임.

<텍스트 분석 프로세스>
1. 텍스트 사전 준비 작업(텍스트 전처리) :
   텍스트를 피처로 만들기 전에 미리 클렌징, 대/소문자 변경, 특수문자 삭제 등의 클렌징 작업, 단어 등의 토큰화 작업, 의미 없는 단어 제거, 어근 추출 등으 ㅣ텍스트 정규화 작업을 수행하는 것을 통칭함.

2. 피처 벡터화/추출 :
   사전 준비 작업으로 가공된 텍스트에서 피처를 추출하고 여기에 벡터 갓을 할당함. 대표적으로는 Bow와 Word2Vec이 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 수행함.

<텍스트 분석 패키지>
● NLTK
가장 대표적인 NLP 패키지. 방대한 데이터 세트와 서브 모듈을 가지고 있으며 NLP의 거의 모든 영역을 커버하고 있음. 많은 NLP 패키지가 NLTK의 영향을 받아 작성되고 있음. 수행 속도 측면에서 아쉬움.

● Gensim
토픽 모델링 분야에서 가장 두각을 나타내는 패키지.

● SpaCy

[2] 텍스트 사전 준비 작업(텍스트 전처리) - 텍스트 정규화
텍스트 정규화?
- 머신러닝 알고리즘이나 NLP 애클리케이션에 입력 데이터로 사용하기 위해 클렌징, 정제, 토큰화, 어근화 등의 다양한 텍스트 데이터의 사전 작업을 수행하는 것을 의미함.

(1) 클렌징
(2) 토큰화
(3) 필터링/스톱 워드 제거/ 철자 수정
(4) Stemming
(5) Lemmatization

(1) 클렌징
텍스트에서 분석에 방해가 되는 불필요한 문자, 기호 등을 사전에 제거하는 작업.

(2) 텍스트 토큰화
- 문장 토큰화
: 문장의 마침표(.), 개행문자(\n) 등 문자의 마지막을 뜻하는 기호에 따라 분리하는 것이 일반적임. 또한 정규 표현식에 따른 문장 토큰화도 가능함.
es. NTLK에서 많이 사용하는 sent_tokenize

- 단어 토큰화
문장을 단어로 토큰화하는 것임. 기본적으로는 공백, 콤마(,) , 마침표(.), rogod 문자 등으로 단어를 분리하laks, 정규 표현식을 이용해 다양한 유형으로 토큰화를 수행할 수 있음.

(3) 스톱 워드 제거
스톱 워드는 분석에 큰 의미가 없는 단어를 지칭함.
ex. is, the a will 등 문장을 구성하는 필수 문법 요소지만 문맥적으로 큰 의미가 없는 단어

(4) Stemming과 Lemmatization
많은 언어에서 문법적인요소에 따라 단어가 다양하게 변함.
ex. 영어 : 과거/현재, 3인칭 단수 여부, 진행형 등 매우 많은 조건에 따라 원래 단어가 변화함.
Stemming과 Lemmatization은 문법적 똔느 의미적으로 변화하는 단어의 원형을 찾는 것임.

두 기능 모두 원형 단어를찾는다는목적은 유사하지만, Lemmatization이 Stemming보다 정교하며 의미론적인 기반에서 단어의 원형을 찾음.
Stemming은 원형 단어로 변환 시 일반적인 방법을 적용하거나 더 단순화된 방법을적용해 원래 단어에서 일부 철자가 훼손된 어근 단어를 추출하는 경향이 있음.
Lemmatization은 품사와 같은 문법적인 요소와 더 의미적인 부분을 감안해 정확한 철자로 된 어근 단어를 찾아줌. 따라서 Lemmatization이 Stemming보다 변환에 더 오랜 시간이 필요함.
NTLK
[Stemmer] : Porter, Lancaster, Snowball Stemmer
[Lemmatization] : WordNetLemmatizer
*일반적으로 Lemmatization은 보다 정확한원형 단어 추출을 위해 단어의 품사를 입력해줘야 함.

[3] Bag of Words = Bow
BAg of Words 모델은 문서가 가지는 모든 단어를 문맥이나 순서를 무시하고 일괄적으로 단어에 대해 빈도 값을 부여해 피처 값을 추출하는 모델임.

2개으 lanswkd이 있다고 가정하고 이 문장을 Bag of words의 단어 수 기반으로 피처를 추

출 .
문장 1) ‘My wife likes to watch baseball gams and my daughter lkis to watch baseball games too’

문장 2) : ‘My wife likes to play baseball’

(1) 문장 q과 문장 2에 있는 단어에서 중복을 제거하고 각 단어를 칼럼 형태로 나열함. 그리고나서 각 단어에 고유의 인덱스를 북여함.
(2)개별 문장에서 해당 단어가 나타나는 횟수를 각 단어에 기재함.

** Bow 모델의 장점 : 쉽고 빠른 구축. (단순히 단어의 발생 횟수에 기반하고 있지만 예상보다 문서의 특징을 잘 나타낼 수 이쓴 모델이어서 전통적으로 여러 분야에서 활용도가 높음)

** 단점 : (1) 문맥 의미 반영 부족, (2) 희소 행렬 문제

<Bow 피처 벡터화>
머신러닝 알고리즘은 일반적으로 숫자형 피처를 데이터로 입력받아 동작하기 때문에 텍스트와 같은 데이터는 머신러닝 알고리즘에 바로 입력할 수 없음. 따라서 텍스트는 특정 의미를 가지는 숫자형 값인 긕터 값으로 변혼해야 하는데, 이러한 변환을 피처 벡터화라고 함 .

각 문서의 텍스트를 단어로 추출해 피처로 할당하고 각 단어의 발생 빈도와 같은 값을 이 피처에 값으로 부여해 각 문서를 이 단어의 발생 빈도와 같은 값이 피처에 값으로 부여해 각 문서를 이 단어 피처의 발생 빈도 값으로 구성된 벡터로 만드는 기법. 피처 벡터화는 기존 텍스트 데이터를 또 다른 형태의 피처의 조합으로 변경하기 때문에 넓은 범위의 피처 추출에 포함함.

Bow 모델이서 피처 벡터화를 수행한다는 것은 모든 문서에서 모든 단어를 칼럼 형태로 나열하고 각 문서에서 해당 단어으 ㅣ횟수나 정규화된 빈도를 값으로 부여하는 데이터 세트 모델로 변경하는 것임.

(1) 카운트 기반의 벡터화
: 카운트 값이 높을수록 중요한 단어로 인식함. 그러나 카운트만 부여할 경우 그 문서으 ㅣ특징을 나타내기 보다는 언어의 특성상 문장에서 자주 사용될 수밖에 없는 단어까지 높은값을 부여하게 됨.
(2) TF-IDF
: 개별 문서에서 자주 나타나는 단어에 높은 가중치를 주되, 모든문서에서 전반적으로 자주 나타나는단어에 대해서는 패털티를 주는 방식으로 값을 부여함

<CountVectorizer, TfidfVectorizer>
CountVectorizer : 단지 피처 벡터화만 수행하지는 않으며 소문자 일괄 변환, 토큰화, 스톱 워드 필터링 등의 텍스트 전처리도 함께 수행함.

ContVectorizer클래스를 이용해 카운트 기반의 피처 여러 개의 문서로 구성된 텍스트의 피처 벡터화 방법
1) 영어의 경우 모든 문자를 소문자로 변경하는 등의 전처리 작업을 수행함.
2) 디폴트로 단어 기준으로 n_gram_range를 반영해 각 단얼르 토큰화함.
3) 텍스트 정규화를 수행함.

<Bow 벡터화를 위한희소 행렬>\4
대규모 행렬의 대부분의 값을0이 차지하는 행렬을 가리켜 희소 행렬이라고 함.
희소행렬은 너무 많은 불필요한 0값이 메모리 공간에 할당되어 메모리 공간이 많이 필요하며, 행렬의 크기가 커서 연산 시에도 데이터 액세스를 위한 시간이 많이 소ㅗ도미.l
-> 물리적으로 적ㅇ른 메모리 공간을 차지할 수 있도록 변환해야 함. 대표적인 방법으로 COO 형식과 CSR 형식이 있음.

<COO 형식>
0이 아닌 데이터만 별도으 ㅣ데이터 배열에 저장하고, 그 데이터가 가리키는 행과 열의 위치를 별도의 배열로 저장하는방식임.
파이썬에서는 희소 행렬 변환을 위해서 주로 사이파이를 이용함.

<CSR 형식>
COO 형식이 godd과 열의 위치를 나타내기 위해서 반복적인 위치 데이터를 사용해야 하는 문제점을 해결한 방식임.
고유값의 시작 위치만 알고 있으면 얼마든지 행 위치 배열을 다시 만들 수 있기에 COO 방식보다 메모리가 적게 들고 빠른 연산이 가능함.

[4] 텍스트 분류 실습 - 20 뉴스그룹 분류
텍스트 분류는 특정 문서를학습 데이터를통해 학습해 모델을 생성한 뒤 이 학습 모델을 이용해 다른 문서의 분류를 예측하는 거임.
텍스트를 피처 벡터화로 변환하면 일반적으로 희소 행렬 형태가 됨. 그리고 이러한 희소 행렬에 분류를 효과적으로 잘 처리할 수 있는 알고리즘은 로지스틱 회귀, 선형 서포트 벡터 머신, 나이브 베이즈 등임.
텍스트를기반으로 분류를수행할 때는 먼저 텍스트를정규화한 뒤 벡터화를 적용함. 그리고 그 이후에 적합한 머신러닝 알고리즘을 적용해 분류를 학습/예측/평가함.

< 텍스트 정규화 >

< 피처 벡터화 변환과 머신러닝 모델 학습/에측/평가>
-> 테스트 데이터 역시 피처 벡터화를 수행하는데, 한가지 반드시 유의해야 할 점이 있음.
: 테스트 데이터에서 CountVectorizer를 적용할 때는 반드시 학습 데이터를 이용해 fit()dl 수행된 CountVectorizer 객체를 이용해 테스트 데이터를 변환해야 한다는 것임. 그래야만 학습 시 설정된 CountVectorizer의 피처 개수오 ㅏ테스트 데이터를CountVectorizer로 변환할 피처 개수가 같아짐.

테스트 데이터의 피처 벡터화 시 fit_transform()을 사용하면 안된다는 점도 유의하기!!

텍스트 분석에서 머신러닝 모델의 성능을 향상시키는 중요한 2가지 방법
1) 최적의 ML 알고리즘을 선택
2) 최상의 피처 전처리를 수행

<사이킷런 파이프라인 사용 및 GridSearchCV와의 결합>
사이킷런의 Pipeline 클래스를 이용하면 피처 벡터화와 ML 알고리즘 학습/예측을 위한 코드 작성을 한 번에 진행할 수 있음.
=> 데이터의 전처리와 머신러닝 학습 과정을 통일된 API 기반에서 처리할 수 있어 더 직관적인 ML 모델 코드를 생성할 수 있음.
=> 대용량의 데이터으 ㅣ피처 벡터화 결과를 별도 데이터로 저장하지 않고 스트림 기반에서 바로 머신러닝 알고리즘의 데이터로 입력할 수 있기 때문에 수행 시간을 절약할 수 있음.

일반적으로 사이킷런 파이프라인은 텍스트 기반의 피처 벡터화뿐만 아니라 모든 데이터 전처리 작업과 Estimator를 결합할 수 있음.

[5] 감성 분석
감성 분석은 문서의 주관적인 감성/의견/감정/기분 등을 파악하기 위한 방법으로 소셜 미디어, 여론 조사, 온라인 리뷰, 피드백 등 다양한 분야에서 활용되고 있음.
감성 분석은 문서 내 텍스트가 나타내는 여러 가지 주관적인 단어와 문맥을 기반으로 감성 수치를 계산하는 방법을 이용함. 이러한 감성 지수는 긍정 감성 지수와 부정 감성 지수로 구성되며 이들 지수를 합산해 긍정 감성 또는 부정 감성을 결정함
감성분석은 머신러닝 관점에서 지도학습과 비지도학습 방식으로 나눌 수 있다

<지도학습 기반 감성 분석 실습 - IMDB 영화평>

<비지도학습 기반 감성 분석 소개>
-> Lexicon을 기반으로 함.
Lexicon은 일반적으로 어휘집을 의미하지만 여기서는 주로 감성만을 분석하기 위해 지원한느 감성 어휘 사전임. 감성 사전은 긍정 감성 똔느 부정 감성의 정도를 의미하는 수치를 가지고 있으며 이를 감성 지수라고 함.
이 감성 지수는 단어의 위치나 주변 단어, 문맥, POS 등을 참고해 결정됨.
이러한 감성 사전을 구현한 대표격은 NLTK 패키지임. NLTK는 많은 서브 모듈을 가지고 있으며 그중에 감성 사전인 Lexicon 모듈도 포함되어 있음.

[WordNet 모듈]
시맨틱(문맥상 의미) 분석을 제공하는 방대한 영어 어휘 사전.
WordNet은 다양한 상황에서 같은 어휘라도다르게 사용되는 어휘의 시맨틱 정볼르 제공하며, 이를 위해 각각의품사로 구성된 개별 단어를 Synset이란느 개념을 이용해 표현함.
Synsetdsm 단순한 하나의 단어가 아니라 그 단어가 가지는 문맥, 시캔틱 정보를 제공하는

WordNet의 핵심 개념임.

단점 : 예측 성능이 그릿 ᄂᆞᆮ지 못함.

실제 업무 적용은 다른 감성 사전을 적용하는 것이 일반적임.
ex. SentiWordNet, VADER, Pattern

<SentiWordNet을 이용한 감성분석>

감성 분석을 수행하는 개략적인 순서
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 임계ㅣ값 이상일 때 긍정 감성으로, 그렇지 않을 때는 부정 감성으로 결정.

<VADER를 이용한 감성 분석>

[6] 토픽 모델링 - 20 뉴스그룹
토픽모델링이란 문서 집합에 숨어 있는 주제를 찾아내는 것임.
머신러닝 기반의 토픽 모델은 숨겨진 주제를 효고적으로 표현할 수 있는 중심 단어를 함축적으로 추출함.