9️⃣

9. 자연어 처리

9.1 자연어 처리란

자연어 처리는 인간이 일상생활에서 사용하는 언어의 의미를 분석하여 컴퓨터가 처리할 수 있도록 변환하는 과정

언어의 형태와 종류가 매우 다양하고 복잡하기 때문에 처리하기 어려움. 자연어 처리는 이미 다양한 분야에서 활용되고 있지만, 여전히 많은 발전이 필요한 분야도 존재.

9.1.1 자연어 처리 용어 및 과정

말뭉치 (Corpus)

- **정의**: 자연어 연구 및 모델 학습을 위해 특정 목적에서 표본을 추출한 데이터 집합.

토큰 (Token)

- **정의**: 문서를 나누는 최소 단위.
- **토큰 생성 (Tokenizing)**: 문자열을 토큰 단위로 나누는 작업.
  - **토큰화**: 텍스트를 문장 또는 단어 단위로 나누는 과정.
  - **토큰 생성 함수**: 문자열을 분리하여 토큰을 생성하는 함수.

불용어 (Stopwords)

- **정의**: 문장 내에서 자주 등장하지만 분석에 큰 의미가 없는 단어.
- **처리 필요성**: 불용어는 분석 성능을 저하시킬 수 있으므로 제거해야 함.(예: `a`, `the`, `she`, `he` 등)

어간 추출 (Stemming)

- **정의**: 단어를 기본 형태(어간)로 변환하는 작업.

품사 태깅 (Part-of-Speech Tagging)

- **정의**: 문장에서 각 단어의 품사를 식별하여 태그를 부여하는 과정.
- **태그 예시**:
  - `Det` : 한정사
  - `Noun` : 명사
  - `Verb` : 동사
  - `Prep` : 전치사
  - `VBZ` : 동사, 현재형
  - `PRP` : 인칭 대명사
  - `JJ` : 형용사
  - `VBG` : 동명사 또는 현재 분사
  - `NNS` : 복수형 명사
  - `CC` : 등위 접속사

자연어 처리의 과정

1. **입력 텍스트**: 자연어 데이터 입력.
2. **전처리**: 텍스트를 정리하고 필요한 형태로 변환.
3. **단어 임베딩**: 텍스트 데이터를 벡터로 변환.
4. **모델 활용**: 데이터를 분석하거나 예측 수행.

9.1.2 자연어 처리를 위한 라이브러리

NLTK

- **정의**: 영어 기반의 교육용 자연어 처리 및 문서 분석용 파이썬 라이브러리.
- **주요 기능**:
  - 말뭉치 제공
  - 토큰 생성
  - 형태소 분석
  - 품사 태깅

KoNLPy

- **정의**: 한국어 자연어 처리를 위한 파이썬 라이브러리.
- **주요 분석기**: Kkma, Komoran, Hannanum, Twitter(Okt), Mecab 등.
- **형태소**: 언어를 나눌 때 의미를 가지는 최소 단위.

9.2 전처리

텍스트 데이터를 분석하기 전, 데이터를 정리하고 변환하는 과정.

9.2.1 결측치 확인

결측치 처리 방법:

1. NaN 값을 포함하는 행 삭제.
2. 데이터가 거의 없는 열 삭제.
3. NaN 값을 최빈값 또는 평균값으로 대체.

9.2.2 토큰화

토큰화는 텍스트를 단어 또는 문자 단위로 자르는 작업.

- **문장 토큰화**: 마침표(`.`), 느낌표(`!`), 물음표(`?`) 등 문장 구분 기호를 기준으로 분리.
- **단어 토큰화**: 공백을 기준으로 문장을 나눔.예: `A cat is on the sofa` → `['A', 'cat', 'is', 'on', 'the', 'sofa']`

9.2.3 불용어 제거

- **정의**: 문장 내에서 자주 등장하지만 의미를 부여하기 어려운 단어.
- **필요성**: 처리 효율성을 높이고 분석 시간을 줄이기 위해 반드시 제거.

9.2.4 어간 추출 및 표제어 추출

어간 추출 (Stemming)

- **정의**: 단어의 기본 형태(어근)만 추출.
- **특징**: 단어 그 자체만 고려하며, 품사에 관계없이 사용 가능.예: `Automates`, `automatic`, `automation` → `automat`.

표제어 추출 (Lemmatization)

- **정의**: 단어의 원형을 문맥에 맞춰 추출.

- **특징**: 문장에서의 품사를 고려하여 품사가 같아야 사용 가능.
  예:
  `am`, `are`, `is` → `be`; `car`, `cars` → `car`.

> 차이점:
>
> - 어간 추출: 사전에 없는 단어도 처리 가능.
> - 표제어 추출: 사전에 등록된 단어만 처리 가능.
> - 예: Porter 알고리즘은 축소 정도가 적어 비교적 정확도가 높음.

9.2.5 정규화

정규화는 데이터 특성의 범위를 일정하게 조정하는 과정.

- **필요성**: 데이터 크기가 크다고 해서 분석에서 더 중요한 요소라고 단정할 수 없기 때문.

정규화 종류

1. **MinMaxScaler**:
   - 데이터 값을 0~1 사이로 조정.
   - 이상치에 민감.
2. **StandardScaler**:
   - 평균을 0, 분산을 1로 조정.
3. **RobustScaler**:
   - 중간값과 사분위수 범위를 사용.
   - 이상치의 영향을 줄임.
4. **MaxAbsScaler**:
   - 데이터 절댓값을 -1~1 사이로 조정.
   - 큰 이상치에 민감.
   - 주로 희소 행렬에 사용.

사분위수

- **Q1**: 제1사분위수 (25%)

- **Q2**: 제2사분위수 (50%, 중앙값)
- **Q3**: 제3사분위수 (75%)
- **IQR**: 사분위수 범위 = Q3 - Q1