9기_DL_장서연_요약본

5.0 토큰화개요

토큰화

- 자연어 처리 모델을 만들기 위해 말뭉치를 토큰으로 나누어야 함.
  - 말뭉치 : 자연어 모델을 훈련하고 평가하는데 사용되는 대규모의 자연어
  - 토큰 : 개별 단어나 문장 부호와 같은 텍스트
  - 토큰화 : 컴퓨터가 자연어를 이해할 수 있도록 토큰으로 나누는 과정
    → 토크나이저(알고리즘/소프트웨어)를 사용
- 토큰화 예시
  - 입력 : 아버지가 방에 들어가신다.
  - 결과 : 아버지, 가, 방, 에, 들, 어, 가, 시, ㄴ, 다, .
- 토크나이저를 구축하는 다양한 방법이 존재
  - 공백 분할
  - 정규 표현식
  - 어휘 사전
  - 머신러닝

5.1 단어 및 글자 토큰화

단어 토큰화

- 텍스트 데이터를 의미있는 단위인 단어로 분리하는 작업
- 자연어 처리 분야에서 핵심적인 전처리 작업 중 하나
- 품사 태깅, 개체명 인식, 기계 번역 등의 작업에서 널리 사용
- 가장 일반적
- `.split()` 메서드를 사용

글자 토큰화

- 글자 단위로 문장을 나눔
- 비교적 작은 단어 사전 구축 가능
- 언어 모델링과 같은 시퀀스 예측 작업에서 활용
- `list()` 함수를 이용
- 자모 라이브러리를 이용한 토큰화
    - 개별 토큰의 의미가 없으므로 자연어 모델이 각 토큰의 의미를 조합해 결과를 도출해야함.
        - 다의어/동음이의어가 많은 도메인에서 구별 어려움
        - 모델 입력 시퀀스의 길이가 길어질수록 연산량 증가

5.2 형태소 토큰화

형태소 토큰화

- 텍스트를 형태소 단위로 나누는 토큰화 방법

형태소 어휘 사전

- 단어가 어떤 형태소들의 조합으로 이루어져있는지에 대한 정보를 담고 있음
- 각 형태소가 어떤 품사에 속하는지 + 해당 품사의 뜻 등의 정보 제공
- 품사 ㅐ깅 : 텍스트 데이터를 형태소 분석해 해당하는 품사를 태깅하는 작업

KoNLPy 형태소 토큰화 라이브러리

- 명사추출, 형태소 분석, 품사태깅 등의 기능 제공
- 시스템의 목적과 환경에 맞는 적절한 형태소 분석기 사용해야함.

Okt

- 문장을 입력받아 명사, 구, 형태소, 품사 등의 정보 추출하는 메서드 제공
- 대표적 메서드
    - 명사 추출 : okt.nouns
    - 구문 추출 : okt.phrases

- 형태소 추출 : okt.morphs
- 품사 태깅 : okt.pos

꼬꼬마

- 구문 추출 기능 X

NLTK

- 품사태깅, 토큰화 작업을 위해 패키지나 모델을 다운받아야함
- punkt : 통계 기반 모델
- Averaged Perceptron Tagger : 퍼셉트론 기반 모델, 품사태깅

spaCy

- 빠른 속도와 높은 정확도를 목표로 하는 ML 기반 자연어 처리 라이브러리

5.3 하위 단어 토큰화

하위 단어 토큰화

- 기존 OOV(Out of Vocabulary) 문제 해결
- 너무 세밀하지도, 러프하지도 않게 중간 단위로 분해
- 장점 : OOV 감소, 어휘 축소, 형태변형 대응
- 원칙 : 자주 함께 나타나는 문자열을 묶어 서브워드 단위로 모델링

BPE(Byte Pair Encoding)

- 문자 단위 → 가장 자주 출현하는 인접쌍 병합 반복

SentencePiece & Kopora

- SentencePiece
  - 바이트 페어 인코딩과 유사한 알고리즘으로 하위단어 토큰화
  - 언어 비의존 : 공백 비의존, raw text 바로 학습
- Korpora
  - 한국어 공개 말뭉치 수집 패키지

토크나이저 학습 공통 프로세스

1. 데이터 수집 : 도메인 대표성/크기/클린
2. 정규화 : lowercasing, NFD/NFKC, 특수문자 규칙
3. 프리 토큰화 : 공백/ByteLevel/Metaspace 등
4. 모델 학습 : BPE/WordPiece/Unigram, vocab 크기 결정
5. 포스트프로세싱 : [CLS]/[SEP] 삽입 등 템플릿 처리
6. 평가 : 토큰 길이, 커버리지, OOV, Perplexity 대리지표
7. 저장/배포 : 버전 고정, 재현성/호환성 체

WordPiece

- 목표 : BERT 계열에서 채택, 확률 기반 MLE로 어휘 선정
- BPE는 빈도 기반 병합, WordPiece 는 언어 모델 우도/점수 사용
- 접두사 : 보통 ##로 서브워드 표기
- 글자쌍 점수식 : $score(x, y) = \frac{f(x,y)}{f(x)f(y)}$
- f : 빈도, $f(x, y)$: 글자 x와 글자 y의 쌍 빈도
  - BPE는 빈도 최다쌍 채택
  - WordPiece는 우도/점수가 최대가 되는 식 선택

Hugging Face tokenizers 라이브러리

- 리스트 기반 고성능, 파이썬 바인딩
- BPE, Wordpiece, Unigram, ByteLevel 등 지원
- 정규화/프리토큰화/포스트프로세싱 모듈화
- 불필요 공백 제거, 대소문자 변환, 유니코드 정규화, 구두점 처리, 특수문자 처리 등 제공
- 공백 혹은 구두점을 기준으로 입력문장을 나눠 효과적 텍스트 데이터 처리
- 대용량 말뭉치에서도 빠르게 학습
- transformers와 호환됨

WordPiece 토크나이저 모델 학습 과정

1. 말뭉치 수집/정제
2. 정규화 규칙 확정(NFKC, 숫자/기호 규칙)
3. 프리토큰화(Whitespace/ByteLevel)
4. WordPiece 학습(어휘 크기, ## 접두사)
5. 포스트 프로세싱([CLS]/[SEP] 템플릿)
6. 품질 평가(토큰 길이/커버리지/OOV율)
7. 버전 관리 및 배포(JSON 파일 저장)

- 한국어는 공백 의존 낮음 —> ByteLevel/MetaSpace 고려

6.0 임베딩

- 텍스트 벡터화 : 텍스트 → 숫자
  - 원핫 인코딩
  - 빈도 벡터화
- 워드 임베딩
  - 단어를 고정된 길이의 실수 벡터로 표현
  - 단어의 의미를 벡터공간에서 다른 단어와의 상대적 위치로 표현해 단어간의 관계추론
  - Word2Vec, fastText
  - 동적 임베딩 : 고정된 임베딩을 학습해 다의어나 문백정보를 다루기 어렵다는 단점 완화

6.1 언어 모델

- 입력된 문장으로 각 문장을 생성할 수 있는 확률 계산 모델
- 주어진 문장을 바탕으로 문맥을 이해하고, 문장 구성에 대한 예측 수행

자기회귀언어모델

- 입력된 문장들의 조건부 확률을 이용해 다음에 올 단어 예측

통계적 언어모델

- 언어의 통계적 구조를 이용해 문장이나 단어의 시퀀스 생성/분석
- 마르코프 체인을 이용해 구현됨
    - 빈도 기반의 조건부 확률 모델
    - 이전 상태와 현 상태간 전이 확률을 이용해 다음 상태 예측

6.2 N-gram

- 가장 기초적인 통계적 언어모델
- 텍스트에서 N개의 연속된 단어 시퀀스를 하나의 단어로 취급 > 특정 단어 시퀀스가 등장할 확률 추정

6.3 TF-IDF

- 텍스트 문서에서 특정 단어의 중요도를 계산하는 방법
- Bow(Bag-of-Words)에 가중치를 부여
    - 문서나 문장을 단어의 집합으로 표현
    - 중복 허용해 빈도 기록
    - 모든 단어는 동일한 가중치를 가짐.

- TF : 단어 빈도
- DF : 문서 빈도
- IDF : 역 문서 빈도
- TF IDF = TF * IDF