

교육 제목	데이터 기반 인공지능 시스템 엔지니어 양성 과정
교육 일시	2021년 11월26일
교육 장소	YGL C-6 학과장 & 자택(디스코드 이용한 온라인)
교육 내용	
오전	<p>자연어 처리</p> <p>1. NLP Project Workflow</p> <pre> graph LR     A[문제 정의 • 단계를 나누고 simplify • x와 y를 정의] --&gt; B[데이터 수집 • 문제 정의에 따른 수집 • 필요에 따라 레이블링]     B --&gt; C[데이터 전처리 및 분석 • 형태를 가공 • 필요에 따라 EDA수행]     C --&gt; D[알고리즘 적용 (모델 설계) • 가설을 세우고 구현/적용]     D --&gt; E[평가 • 실험 설계 • 테스트셋 구성]     E --&gt; F[배포 • API를 통한 배포 • 상황에 따라 유지/보수]           </pre> <p>2. 전 처리 Workflow</p> <pre> graph LR     A[데이터(코퍼스) 수집 • 구입, 외주 • 크롤링을 통한 수집] --&gt; B[정제 • Task에 따른 노이즈 제거 • 인코딩 변환]     B --&gt; C[레이블링 (Optional) • 문장마다 또는 단어마다 labeling수행]     C --&gt; D[Tokenization • 형태소 분석기를 활용하여 분절수행]     D --&gt; E[Subword Segmentation (Optional) • 단어보다 더 작은 의미 단위 추가 분절수행]     E --&gt; F[Batchify • 사전 생성 및 word2index 맵핑수행 • 효율화를 위한 전/후처리]           </pre> <p>3. 서비스 전체 Pipeline</p> <pre> graph LR     A[데이터(코퍼스) 수집 • 구입, 외주 • 크롤링을 통한 수집] --&gt; B[정제 • 학습 데이터와 같은 방식의 정제수행 • Task에 따른 노이즈 제거 • 인코딩 변환]     B --&gt; C[레이블링 (Optional) • 문장마다 또는 단어마다 labeling수행]     C --&gt; D[Tokenization • 학습 데이터와 같은 방식의 분절수행 • 형태소 분석기를 활용하여 분절수행]     D --&gt; E[Subword Segmentation (Optional) • 학습 데이터로부터 얻은 모델을 활용하여 똑같은 분절수행]     E --&gt; F[Batchify • 학습 데이터로부터 얻은 사전에 따른 word2index 맵핑수행]     F --&gt; G[Prediction • 모델에 넣고 추론수행 • 필요에 따라 search수행 (자연어 생성)]     G --&gt; H[Detokenization (Optional) • 사람이 읽을 수 있는 형태로 변환 (index2word) • 분절 복원]           </pre>

#### 4. Data Cleaning

정규식(Regular expression)을 활용하면 복잡한 규칙의 노이즈도 제거/치환 가능  
코딩 없이 단순히 텍스트 에디터(Sublime Text, VSCode등)도 가능

```
>>> # 주민등록번호 형식을 변경
>>> re.sub("-", "", "901225-1234567")
'90122501234567'
>>> # 필드 구분자를 통일
>>> re.sub(r"[:,|\\s]", " ", "Apple:Orange Banana|Tomato")
'Apple, Orange, Banana, Tomato'
>>> # 문자열의 변경 횟수를 제한
>>> re.sub(r"[:,|\\s]", " ", "Apple:Orange Banana|Tomato", 2)
'Apple, Orange, Banana|Tomato'
```

#### 5. Tokenization

한국어의 경우

- 1) 접사를 분리하여 희소성을 낮추고
- 2) 띄어쓰기를 통일하기 위해 tokenization을 수행

굉장히 많은 POS Tagger가 존재하는데,

- 전형적인 쉬운 문장(표준 문법을 따르며, 구조가 명확한 문장)의 경우, 성능이 비슷함
- 하지만 신조어나 고유명사를 처리하는 능력이 다름
- 따라서, 주어진 문제에 맞는 정책을 가진 tagger를 선택하여 사용해야 함.

#### 6. Tokenization Style의 특성

##### 토큰 길이가 짧을 수록

- Vocabulary 크기 감소
  - 희소성 문제 감소
- OOV가 줄어듦
- Sequence의 길이가 길어짐
  - 모델의 부담 증가
- 극단적 형태
  - Character단위

##### 토큰 길이가 길 수록

- Vocabulary 크기 증가
  - 희소성 문제 증대
- OOV가 늘어남
- Sequence의 길이가 짧아짐
  - 모델의 부담 감소



**토큰 길이에 따른 Trade off가 존재**

- 빈도가 높을 경우, 하나의 token으로 나타내고,
- 빈도가 낮을 경우 더 잘게 쪼개어, 각각 빈도가 높은 token으로 구성한다.



**압축 알고리즘?**

오후	별도 실습 파일 참조