**여름방학 2주차 회의록 (21.07.11)**

4조 김태희, 안세훈, 정영훈

**7월 2주차 계획**

1. 크롤링 자동화 구현 (도메인: 네이버)

크롤링 자동화 함수를 만들어 해당 상품명 또는 URL을 입력하면 크롤링이 되도록 구현

<고려사항>

- 상품 종류: 스터디 카페

- 크롤링 리뷰 개수: 1000개

2. 전처리 스터디

크롤링 데이터에서 유의미한 정보를 가져오는 방법 스터디

전반적인 전처리 과정의 원리 및 사용 라이브러리 스터디

Ex) 단어 추출, 품사 태깅, 긍부정 분석 등

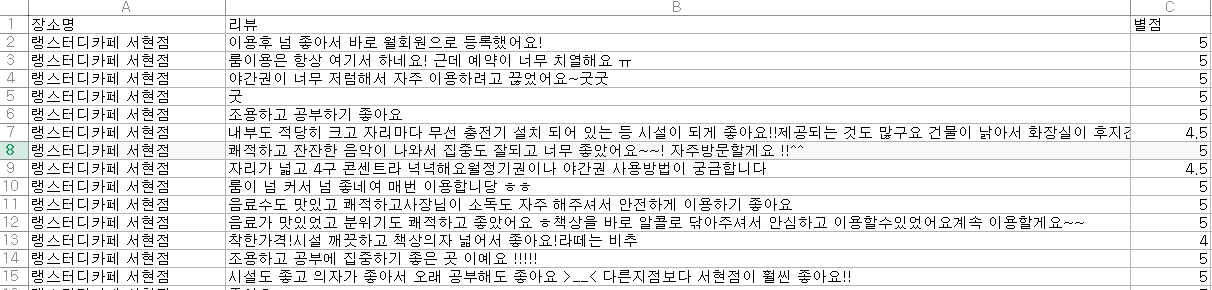
**7월 2주차 활동 결과 요약**

**<전처리 데모 구현 완료>**

- Selenium과 beautifulsoap 을 활용해 자동 스크롤 및 페이지 자동 넘김 구현

- 매장명, 리뷰, 별점 엑셀 파일로 저장

-> 소스코드 : <https://github.com/ashpurple/Graduation-Project/blob/main/Web_Crawling/crawling.py>



**<전처리 전과정 스터디 완료>**

전처리 전과정 스터디 및 스터디 내용 문서화

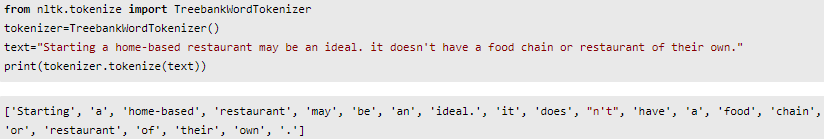
**전처리 스터디**

안세훈

목표: 크롤링된 텍스트 데이터 전처리 스터디

**1. Tokenization**

기준: 띄어쓰기, 구두점, 특수 문자 등



**고려사항**

- 구두점이나 특수 문자를 단순 제외해서는 안된다  
ex) 07/18/21->날짜, 100,123,000->수치

- 줄임말과 단어 내의 띄어쓰기  
ex) we’re->[we] [‘re], New York->[New York]

- 문장 토큰화도 단순 마침표 제외하면 안된다  
ex) ['I am actively looking for Ph.D. students.', 'and you are a Ph.D student.']  
-> 이진 분류기 (1. 마침표가 단어의 일부분일 경우(약어), 2. 마침표가 문장의 구분자일 경우)  
원리: 약어사전 / 사용 사례: NLTK, OpenNLP, splitta ..

**한국어 토큰화**

**한국어 토큰화 고려사항**

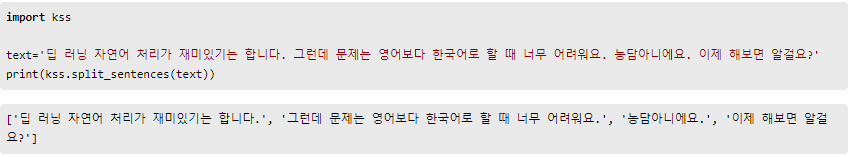
- 한국어 = 교착어 -> 어절 토큰화x, 띄어쓰기로 구분x -> 형태소 토큰화 필요

- 띄어쓰기가 지켜지지 않는 경우가 많다 -> 특히 리뷰와 같은 비정형 데이터에서는 더 심하다

- 품사태깅 필요 : 단어의 표기는 같지만 품사에 따른 의미 구별

**예시 코드**

한국어 문장 tokenizer



토큰화 with Okt 형태소 분석기



토큰화 with 꼬꼬마 형태소 분석기



1) morphs : 형태소 추출

2) pos : 품사 태깅(Part-of-speech tagging)

3) nouns : 명사 추출

**2. Cleaning and Normalization**

정제 및 정규화 Case

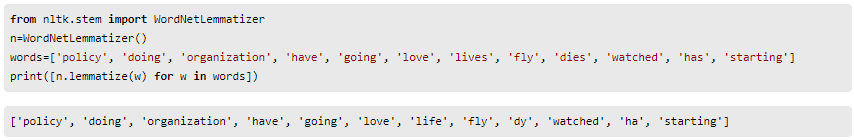
**표기가 다른 단어들의 통합** -> 어간 추출, 표제어 추출  
ex) USA =US, Automobile = automobile

**불필요한 단어의 제거**- 등장 빈도가 적은 단어  
- 길이가 짧은 단어(영어만 해당): 대부분의 단어는 3글자 이상임을 활용

**정규 표현식:** 노이즈 데이터의 특징으로 규칙에 기반하여 제거

어간 추출 및 표제어 추출

**표제어 추출**: 기본사전형 추출 (품사정보 보전)  
ex) am are is -> be, cats -> cat

형태소 = 어간(의미를 담고 있는 핵심), 접사(추가적인 의미)  
형태학적 파싱: 어간과 접사 분리  
  
예시에서 dies->dy, has->ha로 틀린 어간을 츨력한다.   
이때는 를 이용하여 단어의 품사를 알려주면 정확한 출력이 나온다.

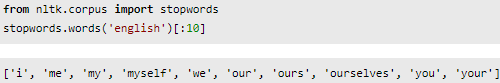
**어간 추출**: 형태학적 분석의 단순화 버전(단순 자르기)  
ex) formalize->formal, organization->organ (추출 단어가 사전에 존재하지 않을 수 있다)

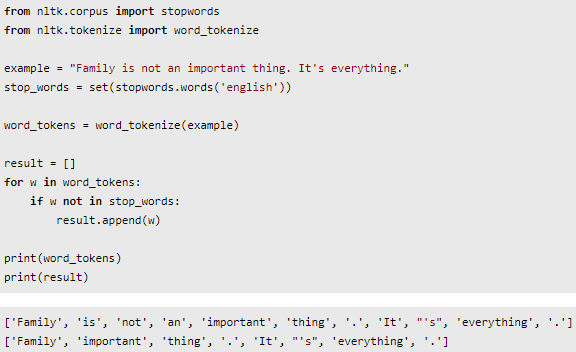
**표제어 추출 vs 어간 추출**  
am -> be / am having-> have / hav

**한국어의 어간 추출**- 용언(동사, 형용사)에 적용  
- 활용(규칙/불규칙): 어간+어미에서 어간의 모습이 일정하다면 규칙 활용 아니라면 불규칙 활용  
ex) 잡다-잡으니(규칙), 듣다-들으니(불규칙)

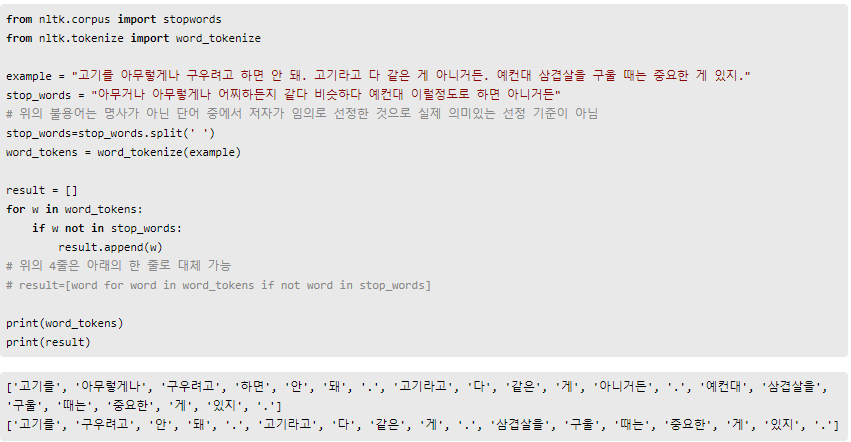
불용어 제거(stop word)

불용어: 주어, 조사, 접미사와 같이 자주 등장하지만 의미 분석에 기여가 적은 단어들

영어에서의 불용어  


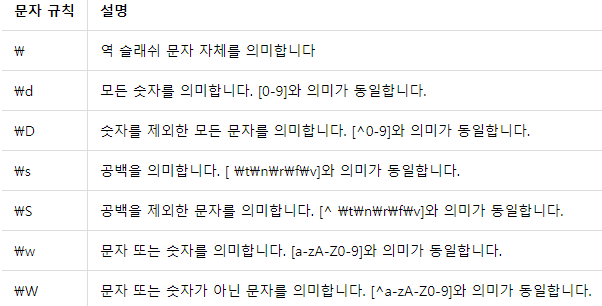
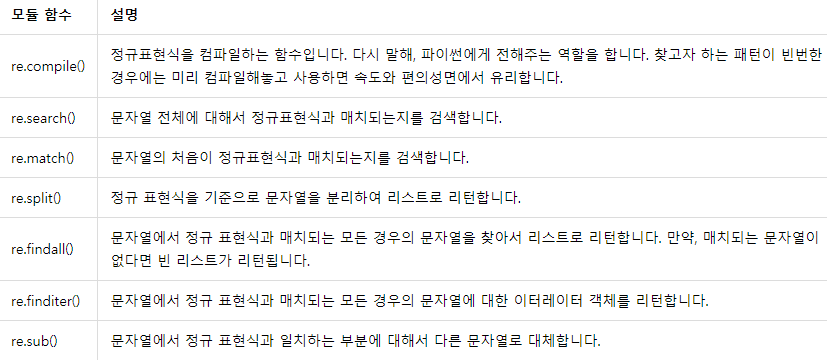
NLTK 불용어 제거  


한국어 불용어 제거



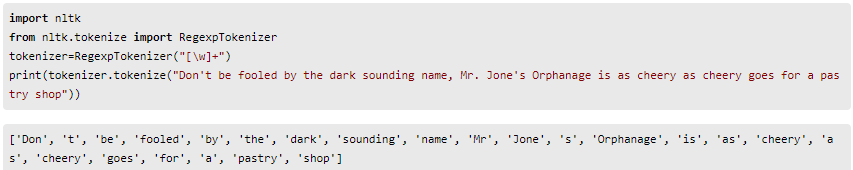
-> 사용자 정의 불용어  
한국어 불용어 제거는 txt파일이나 csv 파일로 불용어를 정리해놓고 불러와서 사용하는 방법이 바람직하다  
보편적 한국어 불용어 리스트 <https://www.ranks.nl/stopwords/korean>

정규 표현식

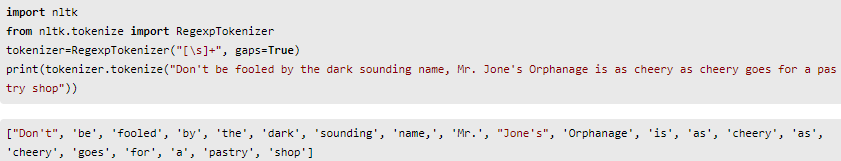
  

**정규 표현식을 이용한 토큰화**

NLTK의 REgexpTokenizer



-> 문장에서 구두점을 제외하고, 단어들을 기준으로 토큰화

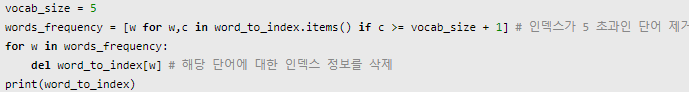
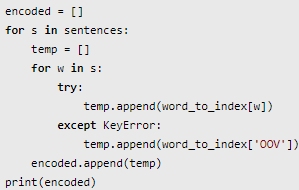


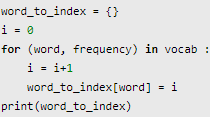
-> 공백을 기준으로 토큰화

**3. Encoding**

Sequence: 문장 토큰화-> 단어 토큰화 with Cleaning -> 정수 인코딩

**정수 인코딩**: 텍스트를 정수화 시키는 과정

**Dictionary 사용**  
1) 빈도수 높은 순으로 정수 인덱스 부여  
  
2) 빈도수가 높은 n개 만 사용  
  
3) 빈도 사전에 없는 단어들을 OOV의 인덱스로 인코딩  
  
4) 문장의 모든 단어들을 맵핑되는 정수로 인코딩  


**Counter 사용**(권장)  
1) 단어들을 하나의 리스트로 만들고 중복제거  
   
  
2) 빈도수가 높은 n개만 저장  
  
3) 빈도수 높은 순으로 인덱스 부여  
  

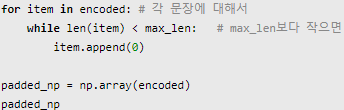
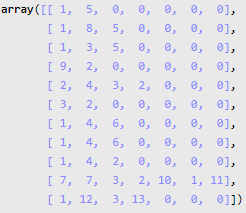

기타 방법: NLTK의 FreqDist, tensorflow의 Keras

순서가 있는 자료형(list, set, tuple, dictionary, string)에서 enumerate 활용

**4. Padding**

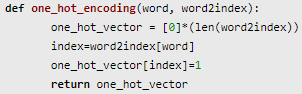
: 병렬 연산을 위해 여러 문장의 길이를 임의로 동일하게 맞춰주는 작업

Numpy로 패딩하기  
1) 정수 인코딩  
  
2) 최대 문장 길이 구하기  
  
3) 가상의 단어(index = 0)로 길이 맞추기

**5. One-Hot Encoding**

**정의**: 집합의 크기 = 벡터의 차원, 각 단어에 대해 고유한 index 부여, 해당 단어만 1 나머지는 0 부여



**한계**  
- 공간 낭비 ex) 1000개의 단어 = 1000개의 차원  
- 단어의 유사도 표현 불가(검색에 치명적)  
-> 단어의 잠재의미를 반영하여 다차원 공간에 벡터화 하는 기법  
1) 카운트 기반 (LSA, HAL) 2) 예측 기반(NNLM, RNNLM, Word2Vec, FastText)

**6. Split Data**

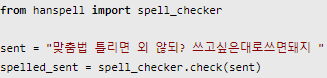
목적: 머신러닝 training/test

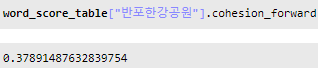
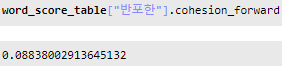
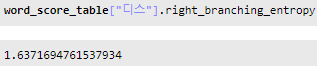
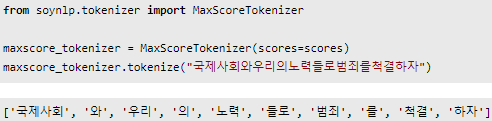
1) X, y 분리: zip 함수, 데이터프레임, numpy  
2) 테스트 데이터 분리

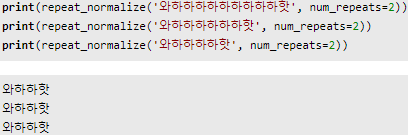
**한국어 전처리 패키지**

KoNLPy와 KSS와 함께 유용하게 사용할 수 있는 패키지

**PykoSpacing**: 띄어쓰기가 돼있지 않은 문장을 띄어쓰기 문장으로 변환  


**Py-Hanspell**: 네이버 맞춤법 검사기를 바탕으로 맞춤법 교정 및 띄어쓰기 보정  


**SOYNLP**: 품사 태깅, 단어 토큰화 등을 지원하는 단어 토큰나이저(비지도 학습)  
-> 신조어와 같이 등록되지 않은 단어를 학습하는 장점이 있다  
- 학습  
  
- 응집확률  
   
-> 응집도 높을수록 완성형 단어  
- 브랜치 엔트로피  
   
-> 엔트로피 낮을수록 다음문자 예측이 명백해짐  
- 최대 점수 토크나이저  


- 반복 문자 정제  


**Customized KoNLPy**: 사용자 사전을 형태소 분석기에 추가  
