

SNS 텍스트 데이터 전처리

2021년 8월

대진대학교 컴퓨터공학과 서혜선교수

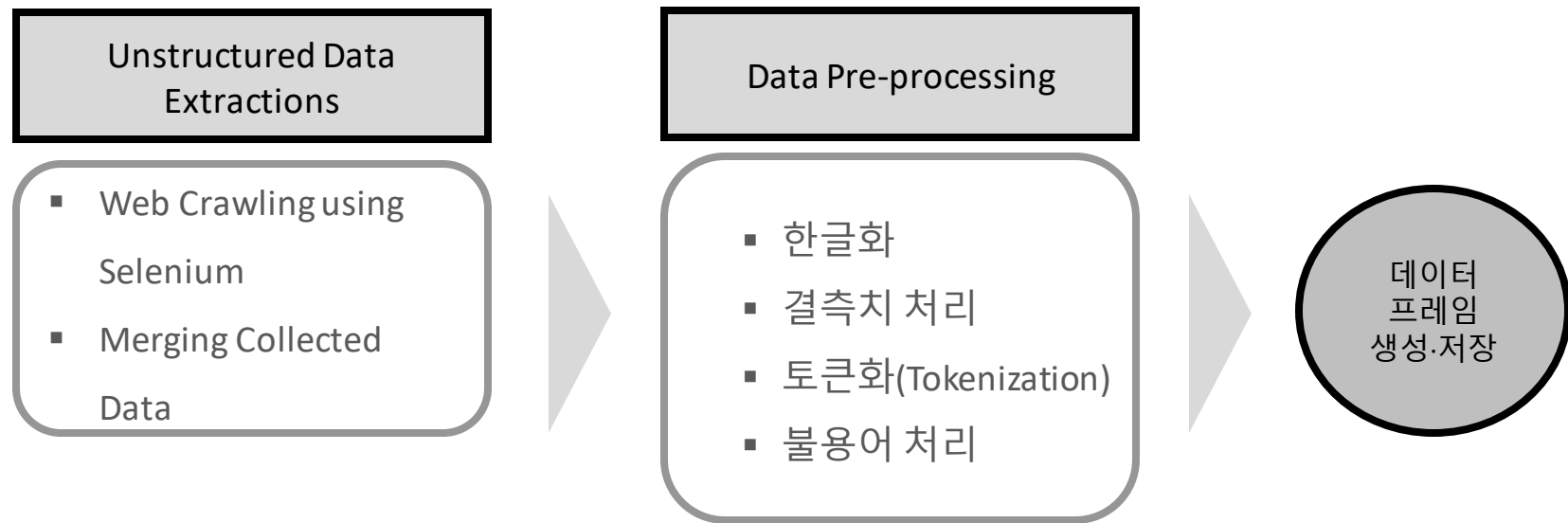
jako403@daejin.ac.kr

SNS 텍스트 데이터 전처리

- 데이터 병합
- 한글화
- 토큰화(형태소 분석)
- 불용어 처리

데이터 전처리 process

자연어 처리에서 크롤링으로 확보한 데이터가 필요에 맞게 전처리되지 않은 상태라면, 사용목적에 맞게 정제(cleaning) & 정규화(normalization) , 토큰화(tokenization) 등을 해야함



1. 데이터 병합

■ 라이브러리 불러오기

```
import pickle
import pandas as pd
```

- Pickle (데이터 저장 라이브러리)
 - 딕셔너리(dictionary)든 데이터프레임(dataframe)이든 어떤 자료형이든지 저장할 때의 그 형태 그대로를 유지하면서 불러올 수 있게 하는 라이브러리
- Pandas (데이터 분석 라이브러리)
 - 쉽고 직관적인 관계형 또는 분류된 데이터로 작업 할 수 있도록 설계된 빠르고 유연하며 표현이 풍부한 데이터 구조를 제공하는 라이브러리

■ 파일을 저장하고 불러올 경로 설정

```
| file_path = 'C:\Users\samsung\서 올핀 테크아카데미\DATA'
```

1. 데이터 병합

■ 크롤링한 데이터를 pkl 형식으로 읽어오기

```
def pklopen(text):  
    f = open(file_path + '{}.pkl'.format(text), "rb") #rb: binary형식을 읽어올때  
    a = pickle.load(f)  
    f.close()  
    return a
```

■ 앞에서 각 채널로부터 크롤링한 데이터들을 pkl로 오픈하여 df1-df4로 지정

```
df1 = pklopen('naver_blog_')  
df2 = pklopen('naver_cafe_')  
df3 = pklopen('다음_웹문서_연금')  
df4 = pklopen('다음_카페_연금')
```

1. 데이터 병합

```
# data merge
data = pd.concat([df1,df2, df3, df4])
data
```

[illegible]

249 rows × 5 columns

1. 데이터 병합

■ 각 채널별 데이터 사이즈 확인

```
#그룹별로 사이즈 확인  
data.groupby(['ch', 'ch2']).size()
```

```
ch    ch2  
daum  cafe    20  
naver blog    99  
      cafe   100  
dtype: int64
```

1. 데이터 병합

■ 병합 데이터 저장 및 확인

```
f = open(file_path + 'total_data.pkl', "wb")
pickle.dump(data, f) #pickle.dump(데이터, 파일)
f.close()
```

data

	title	doc	url	ch	ch2
0	연금계좌관련 문의드립니다.	\n\n\n\n\n\n안녕하세요~ 주린이 6개월차입니다.국민연 금에 퇴직...	https://cafe.naver.com/likeusstock/158991?art=...	n timer naver	cafe
1	국민연금 추납과 소득공제	\n\n\n\n\n\n와이프가 결혼 전 일을 하다가 결혼 후 육아 로 일을...	https://cafe.naver.com/hotellife/1604742?art=Z...	n timer naver	cafe
2	퇴직금계산방법과 지급기준, 퇴직연금제도 확인해보세요!	\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n안녕하세...	https://cafe.naver.com/dokchi/10707550? art=ZXh...	n timer naver	cafe
3	퇴직연금을 잘 굴려 보세요. 대박수익 가능	\n\n\n\n\n\n\n전 제 개인돈은 주식을 하지만 가계자금은 주로 펀드...	https://cafe.naver.com/vilab/176085? art=ZXh0ZX...	n timer naver	cafe
4	연말정산대비 연금지축, IRP	\n\n\n\n\n\n\n미리 준비하고 공부했어야했는데 너무지금 국세청에서...	https://cafe.naver.com/onepieceholichplus/29862...	n timer naver	cafe
...
244	Re: 귀국 준비 중 후생연금	document.write(removeRestrictTag()); 年金還付制度(연금...	http://cafe.daum.net/osakalife/2xAE/89957? q=%E...	n timer daum	cafe
245	주택연금	document.write(removeRestrictTag()); 주택연금 생각해보...	http://cafe.daum.net/20150811/XcwF/2870? q=%EC%...	n timer daum	cafe
246	개인연금 퇴직연금 전부 다 하시나요?	document.write(removeRestrictTag()); 개인연금 퇴직연금...	http://cafe.daum.net/mmnix/EttT/769? q=%EC%97%B...	n timer daum	cafe
247	연금 질문있습니다	document.write(removeRestrictTag()); 회원님들께서 추천...	http://cafe.daum.net/jordan777/CjFR/6497? q=%EC%	n timer daum	cafe
248	연금법에 대해 논하시오1)국민연금법의 필요 성, 2)국민...	document.write(removeRestrictTag()); 과목레포트 주제[...	http://cafe.daum.net/kimhyeongwoo/AOCD/166? q=%...	n timer daum	cafe

249 rows × 5 columns

2. 한글화

■ 데이터 경로 지정 및 파일 불러오기

```
| import pickle  
| import pandas as pd
```

```
| file_path = 'C:\Users\samsung\서올핀테크아카데미\DATA' # 파일 경로 바꿀것
```

```
| f = open(file_path + 'total_data.pkl', "rb")  
| #pickle 모듈을 사용할 때는 파일 형식을 바이트(b) 형식으로 읽고 써야 함  
| docs = pickle.load(f) #변수 = pickle.load(파일)  
| f.close()  
| docs
```

2. 한글화

■ 한글화 : 한글 외 문자 제거

```
# 본문의 한글화
import re
for i in range(len(docs)):
    docs['doc'][i] = re.sub("[^ㄱ-ㅎㅌ-ㅣ가-힣]", "", str(docs['doc'][i]))
docs
#sub():문자열에서 매치된 텍스트를 다른 텍스트로 치환
```

■ 자주 사용되는 정규표현식

한글 : `^[ㄱ-ㅎㅌ-ㅣ가-힣]`

한글 외 : `[^ㄱ-ㅎㅌ-ㅣ가-힣]`

영문 : `^[a-zA-Z]`

영문 외 : `[^a-zA-Z]`

숫자 : `^[0-9]`

숫자 외 : `[^0-9]`

한,영,숫자 : `^[0-9a-zA-Zㄱ-ㅎㅌ-ㅣ가-힣]`

한,영,숫자 외(특수문자) : `[^0-9a-zA-Zㄱ-ㅎㅌ-ㅣ가-힣]`

■ 한글화 진행 결과

12

3. 결측값 처리

- 결측값 제거 : 문서 내용이 두 글자 미만이거나 공백인 경우, 결측값으로 간주해 제외
즉 특정 문서에 해당하는 행을 삭제

```
#결측값 처리
for i in range(len(docs)):
    if (len(docs['doc'][i]) < 2 or docs['doc'][i].isspace() == True):
        #두글자 미만이나 공백으로만 된 경우 결측값 처리
        docs = docs.drop(i, 0)  #(삭제할 인덱스, axis) #axis = 1일때 열 삭제, 0일때 행 삭제
print("ok")
```

ok

- 데이터프레임 인덱스 초기화 (몇 개의 문서가 삭제되었는지 확인 가능)

```
docs = docs.reset_index(drop=True) #인덱스 초기화 #drop=false가 디폴트인데, 이 경우 기존 인덱스가 새 열로 생성됨
docs
```

3. 결측값 처리

■ 결측값 처리 결과

	title	doc	url	c
0	연금계좌관련 문의드립니다.	안녕하세요 주린이 개월차입니다국민연금에 퇴직연금이 있어서 그동안 개인연금은 관심을 ...	https://cafe.naver.com/likeusstock/158991?art=...	naver
1	국민연금 추납과 소득공제	와이프가 결혼 전 일을 하다가 결혼 후 육아로 일을 쉬고 있습니다 한의사라 연제가 ...	https://cafe.naver.com/hotellife/1604742?art=Z...	naver cafe
2	퇴직금계산방법과 지급기준, 퇴직연금제도 확인해보세요!	안녕하세요오늘은 퇴직금계산방법과 지급 기준을 알아보려고요 의외로 많은 분들이 퇴직금계...	https://cafe.naver.com/dokchi/10707550?art=ZXh...	naver cafe
3	퇴직연금을 잘 굴려 보세요. 대박수익 가능	전 제 개인돈은 주식을 하지만 가계자금은 주로 펀드 투자로 운용하는데요 그러다보니 ...	https://cafe.naver.com/vilab/176085?art=ZXh0ZX...	naver cafe
4	연말정산대비 연금저축. IRP	미리 준비하고 공부했어야했는데 ㅜㅜ지금 국세청에서 미리계산해보니 약 만원 환수될것...	https://cafe.naver.com/onepieceholicplus/29862...	naver cafe
...
244	Re: 귀국 준비 중 후생연금	연금환급제도 연금환급제도 탈퇴일시금이란 탈퇴일시금 환급신청이란일본에서 연금을...	http://cafe.daum.net/osakalife/2xAE/89957?q=%E...	daum cafe
↓				
214	개인연금 퇴직연금 전부 다 하시나요?	개인연금 퇴직연금 이 둘중 퇴직연금만 불입하고 있어요제 생각에는 굳이 연금저축을 ...	http://cafe.daum.net/mmnix/EttT/769?q=%EC%97%B...	daum cafe
215	연금 질문있습니다	회원님들께서 추천해주신 마법의 연금굴리기와 카페 내 글들을 보면서 궁금한 것이 생...	http://cafe.daum.net/jordan777/CjFR/6497?q=%EC...	daum cafe
216	연금법에 대해 논하시오1)국민연금법의 필요성, 2)국민...	과목레포트 주제복지법제론국민연금법에대해논하시오국민연금법의필요성국민연금급여의수준및종...	http://cafe.daum.net/kimhyeongwoo/AOCD/166?q=%...	daum cafe

217 rows × 5 columns

4. 토큰화

주어진 문서에서 토큰(token)이라 불리는 단위로 나누는 작업을 토큰화(tokenization)라고 함
토큰의 단위는 보통 의미있는 단위로 정의 (단어, 단어구, 의미있는 문자열 등)

문장 예시 : I want to present the most shining moment!

구두점(마침표, 쉼표, 물음표(?), 세미콜론, 느낌표, 등)을 제외한 토큰화 작업의 결과

결과 : "I", "want", "to", "present", "the", "most", "shining", "moment"

→ 보통 토큰화 작업은 단순히 구두점이나 특수문자를 제거하는 정제(cleaning)

작업만으로 해결되지 않음.

특히 한국어는 조사 등이 존재하여 띄어쓰기만으로는 단어 토큰을 구분하는데 어려움 존재

예) "그" 라는 단어 : '그가', '그에게', '그를', '그와', '그는'과 같이 다양한 조사가 존재

4. 토큰화

형태소 분석

한국어 토큰화에서는 형태소(morpheme)란 개념을 이해해야 함

형태소란 “**뜻을 가진 가장 작은 말의 단위**”를 의미하며 자립 형태소와 의존 형태소로 나뉨

자립 형태소 : 접사, 어미, 조사와 상관없이 자립하여 사용할 수 있는 형태소로 그 자체로 단어가 되는 경우

명사, 대명사, 수사, 관형사, 부사, 감탄사 등

의존 형태소 : 다른 형태소와 결합하여 사용되는 형태소. 접사인 어미와 조사인 어간으로 나뉨

4. 토큰화

형태소 분석 KoNLPy(Korean Natural Language Python)

- 이러한 한국어 자연어 처리(형태소 분석)을 위해서는 KoNLPy(코엔엘파이)라는 파이썬 패키지를 사용해야함
- 즉, 한국어 NLP에서 형태소 분석기를 사용한다는 것은 단어 토큰화가 아니라 정확히는 형태소(morpheme) 단위로 형태소 토큰화(morpheme tokenization)를 수행
- KoNLPy를 통해서 사용할 수 있는 형태소 분석기로는 Okt(Open Korea Text), 메캅(Mecab), 코모란(Komoran), 한나눔(Hannanum), 꼬꼬마(Kkma) 등이 있음

4. 토큰화

■ KoNLPy를 통해 분석할 수 있는 형태소 분석기들

Ex) 아버지가방에들어가신다

1) Kkma: 꼬꼬마

- 서울대학교 IDS(Intelligent Data Systems) 연구실 개발
- 분석 시간이 다소 오래걸림
- 아버지 / 가방 / 에 / 들어가 / 시 / 니다

2) Komoran: 코모란

- Shineware에서 개발
- 오타자에 대한 분석 품질을 가장 보장함
- 아버지 / 가방 / 에 / 들어가 / 시 / 니다

3) Hannanum: 한나눔

- KAIST Semantic Web Research Center 개발
- 띄어쓰기가 올바르지 않은 문장은 제대로 분석하지 못함
- 아버지가방에들어가 / 이 / 시 니다

4) Okt(Twitter): 오픈 소스 한국어 분석기

- 과거 트위터 형태소 분석기
- 품사 태깅 결과를 Noun, Verb 등 알아보기 쉽게 반환해줌
- 아버지 / 가방 / 에 / 들어가신다

5) Mecab: 은전한닢

- 일본어용 형태소 분석기를 한국어를 사용할 수 있도록 수정
- 분석 속도가 가장 빠르면서도 나쁘지 않은 분석 결과를 보여줌
- konlpy에서 제공하는 Mecab은 mac os에서만 이용 가능, 윈도우에서 사용시 복잡한 별도 설치 필요
- 아버지 / 가방 / 에 / 들어가 / 신다

4. 토큰화

■ 라이브러리 불러오기

```
from tqdm import tqdm          #anaconda prompt에서 pip install tqdm으로 설치가능
from konlpy.tag import Komoran #Komoran 형태소 분석기 불러오기 #https://konlpy-ko.readthedocs.io/ko/v0.4.3/install/
komoran=Komoran()
```

- tqdm
 - 진행상황 상태를 사용자에게 바로 피드백해주는 라이브러리
- tqdm 설치방법 : anaconda prompt에서 설치 가능(쥬피터 노트북에서도 설치가능)
!pip install tqdm

Anaconda Prompt (anaconda3)

```
(base) C:\Users\wdlgmd>pip install tqdm
Requirement already satisfied: tqdm in c:\users\wdlgmd\anaconda3\lib\site-packages (4.54.1)
(base) C:\Users\wdlgmd>
```

4. 토큰화

- konlpy
 - 형태소분석기를 하나로 모은, 한글 자연어 처리에 맞춤형 된 라이브러리
- konlpy 설치방법 : <https://konlpy-ko.readthedocs.io/ko/v0.4.3/install/>

- 윈도우의 경우)

- 1) Java가 1.7버전 이상이 설치되어 있는지 확인

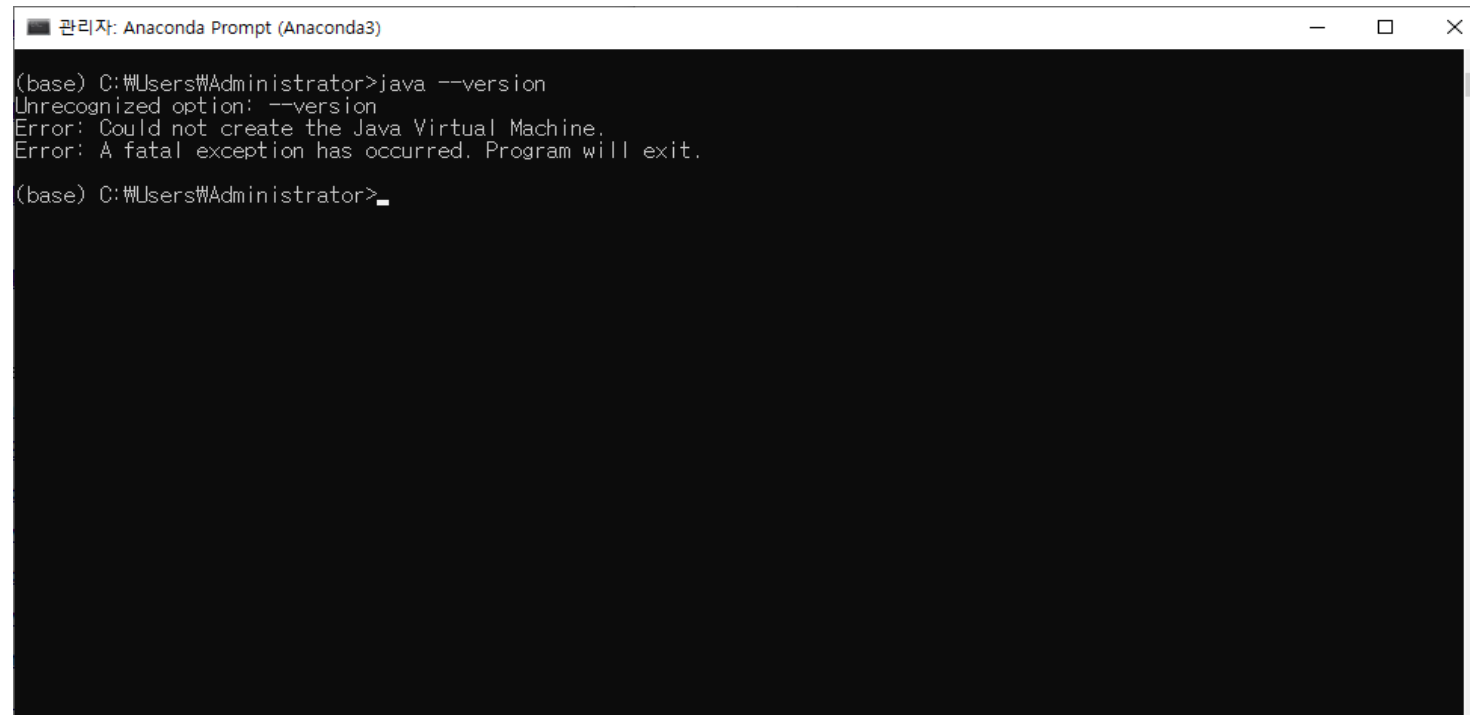
Anaconda Prompt (anaconda3)

```
(base) C:\Users\wdlqmd>java --version
java 15.0.1 2020-10-20
Java(TM) SE Runtime Environment (build 15.0.1+9-18)
Java HotSpot(TM) 64-Bit Server VM (build 15.0.1+9-18, mixed mode, sharing)

(base) C:\Users\wdlqmd>
```

4. 토큰화

- Java 가 설치되어 있지 않은 경우



A screenshot of the Anaconda Prompt (Anaconda3) window. The title bar reads "관리자: Anaconda Prompt (Anaconda3)". The command prompt shows the following text:

```
(base) C:\Users\Administrator>java --version
Unrecognized option: --version
Error: Could not create the Java Virtual Machine.
Error: A fatal exception has occurred. Program will exit.

(base) C:\Users\Administrator>
```

4. 토큰화

1-2) Java가 설치되어 있지 않은 경우

<https://www.oracle.com/kr/java/technologies/javase-downloads.html> 접속

-> JDK 다운로드 클릭

-> 자신의 환경에 맞는 jdk 설치 (설치 경로 기억 !!)

2) JAVA_HOME 설정

제어판 -> 시스템 및 보안 -> 시스템 -> 고급시스템설정 -> 환경변수 접속

시스템 변수 새로 만들기 클릭

변수이름: JAVA_HOME 입력 / 변수값: jdk 설치 경로 입력

사용자변수 - Path 더블클릭 - 새로 만들기 클릭

%JAVA_HOME%\bin 입력

다시 Java 버전 확인

환경변수 설정 후 반드시 Jupyter notebook 을 종료하고 재실행

Anaconda Prompt (anaconda3)

```
(base) C:\Users\wdlqmd>java --version
java 15.0.1 2020-10-20
Java(TM) SE Runtime Environment (build 15.0.1+9-18)
Java HotSpot(TM) 64-Bit Server VM (build 15.0.1+9-18, mixed mode, sharing)

(base) C:\Users\wdlqmd>
```

4. 토큰화

3) JPy1 (>=0.5.7)을 다운받고, pip를 업그레이드한 후 설치

<https://www.lfd.uci.edu/~gohlke/pythonlibs/#jpy1> 접속

-> 자신의 환경에 맞는 JPy1 다운 [Python 3.7버전 : cp37 / Python 3.8버전 : cp38]

(Python 버전을 모르는 경우, anaconda prompt에서 'python -V'로 확인 가능)

-> 다운받은 Jpy1 파일을 anaconda prompt에서 보이는 기본경로로 옮겨주기

-> pip를 업그레이드한 후 Jpy1 설치

```
Anaconda Prompt (anaconda3)
(base) C:\Users\dlqmd>pip install --upgrade pip
Requirement already satisfied: pip in c:\users\dlqmd\anaconda3\lib\site-packages (20.3.3)

(base) C:\Users\dlqmd>pip install JPy1-1.2.0-cp38-cp38-win_amd64.whl
Processing c:\users\dlqmd\jpy1-1.2.0-cp38-cp38-win_amd64.whl
JPy1 is already installed with the same version as the provided wheel. Use --force-reinstall to force an installation of the wheel.

(base) C:\Users\dlqmd>
```

4) Konlpy 설치

```
Anaconda Prompt (anaconda3)
(base) C:\Users\dlqmd>pip install konlpy
Requirement already satisfied: konlpy in c:\users\dlqmd\anaconda3\lib\site-packages (0.5.2)
Requirement already satisfied: colorama in c:\users\dlqmd\anaconda3\lib\site-packages (from konlpy) (0.4.4)
Requirement already satisfied: lxml>=4.1.0 in c:\users\dlqmd\anaconda3\lib\site-packages (from konlpy) (4.6.2)
```

4. 토큰화

3) JPy1 (>=0.5.7)을 다운받고, pip를 업그레이드한 후 설치

<https://www.lfd.uci.edu/~gohlke/pythonlibs/#jpy1> 접속

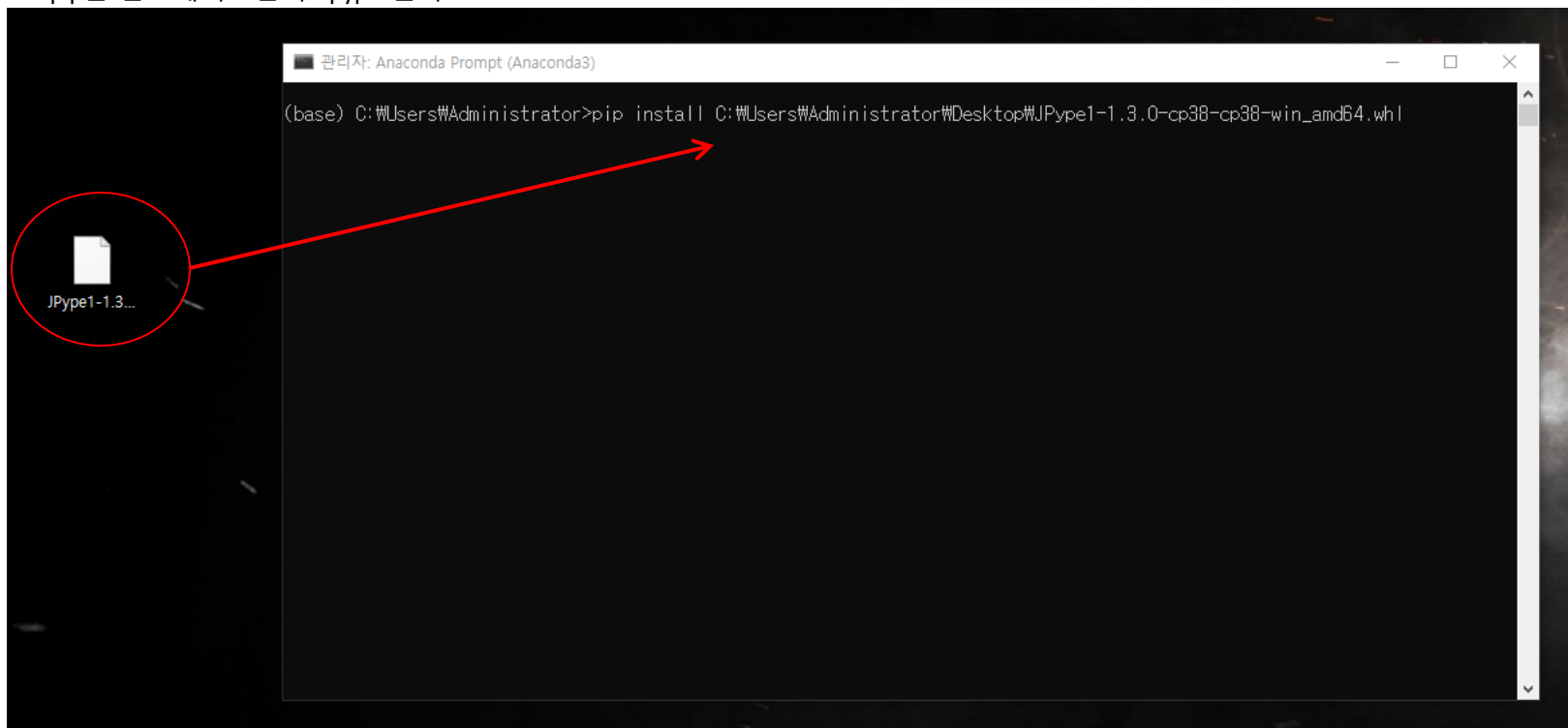
-> 자신의 환경에 맞는 JPy1 다운 [Python 3.7버전 : cp37 / Python 3.8버전 : cp38]

(Python 버전을 모르는 경우, anaconda prompt에서 'python -V'로 확인 가능)

-> anaconda prompt 에 pip install 을 타이핑 후 다운받은 Jpy1 파일을 드래그

anaconda prompt 는 반드시 관리자권한으로 실행

-> pip를 업그레이드한 후 Jpy1 설치



4. 토큰화

- Anaconda prompt 를 관리자 권한으로 실행하지 않은 경우

```
(base) C:\Users\Administrator>pip install C:\Users\Administrator\Desktop\JPype1-1.3.0-cp38-cp38-win_amd64.whl
Processing c:\Users\Administrator\Desktop\jpype1-1.3.0-cp38-cp38-win_amd64.whl
JPype1 is already installed with the same version as the provided wheel. Use --force-reinstall to force an installation
of the wheel.
```

- 정상적인 설치

```
(base) C:\Users\Administrator>pip install C:\Users\Administrator\Desktop\JPype1-1.1.2-cp38-cp38-win_amd64.whl
Processing c:\Users\Administrator\Desktop\jpype1-1.1.2-cp38-cp38-win_amd64.whl
Installing collected packages: JPype1
Successfully installed JPype1-1.1.2
```

4. 토큰화

- 현재 Jpye1 라이브러리에서 오류발생

```
C:\ProgramData\Anaconda3\Lib\site-packages\jpye_core.py in startJVM(*args, **kwargs)
    219
    220     try:
--> 221         _jpye.startup(jvmpath, tuple(args),
    222                       ignoreUnrecognized, convertStrings, interrupt)
    223         initializeResources()

SystemError: java.nio.file.InvalidPathException: Illegal char '<*>' at index 55: C:\ProgramData\Anaconda3\Lib\site-packages\konlpy\java\*
```

- C:\ProgramData\Anaconda3\Lib\site-packages\konlpy 의 경로에 들어간 후 jvm.py 를 메모장에 드래그
- 왼쪽 사진의 * 표시를 제거 후 저장

```
folder_suffix = [
    # JAR
    '{0}',
    # Java sources
    '{0}{1}bin',
    '{0}{1}*', # <- Delete the '*' here
    # Hannanum
    '{0}{1}jhannanum-0.8.4.jar',
    # Kkma
    '{0}{1}kkma-2.0.jar',
    # Komoran3
    '{0}{1}aho-corasick.jar',
    '{0}{1}shineware-common-1.0.jar',
    '{0}{1}shineware-ds-1.0.jar',
    '{0}{1}komoran-3.0.jar',
    # Twitter (Okt)
    '{0}{1}snakeyaml-1.12.jar',
    '{0}{1}scala-library-2.12.3.jar',
    '{0}{1}open-korean-text-2.1.0.jar',
    '{0}{1}twitter-text-1.14.7.jar',
    '{0}{1}*' # <- Delete the '*' here
]
```

```
folder_suffix = [
    # JAR
    '{0}',
    # Java sources
    '{0}{1}bin',
    '{0}{1}',
    # Hannanum
    '{0}{1}jhannanum-0.8.4.jar',
    # Kkma
    '{0}{1}kkma-2.0.jar',
    # Komoran3
    '{0}{1}aho-corasick.jar',
    '{0}{1}shineware-common-1.0.jar',
    '{0}{1}shineware-ds-1.0.jar',
    '{0}{1}komoran-3.0.jar',
    # Twitter (Okt)
    '{0}{1}snakeyaml-1.12.jar',
    '{0}{1}scala-library-2.12.3.jar',
    '{0}{1}open-korean-text-2.1.0.jar',
    '{0}{1}twitter-text-1.14.7.jar',
    '{0}{1}'
]
```

4. 토큰화

■ 토큰화 (실행 진행상황을 확인할 있다 : tqdm)

```
token_list = [] #data의 형태소를 담아낼 리스트
```

```
token_noun = [] #data의 명사만 모아낼 리스트
```

```
for i in tqdm(range(len(docs))):
```

```
pos = komoran.pos(u'{}'.format(docs['doc'][i])) #pos('분석할 문자열') #품사 부착후 추출
```

```
noun = list(term for term in komoran.nouns(u'{}'.format(docs['doc'][i])) if len(term)>1)
```

#for문 안 if문 안 term을 list형식으로 바꿔 noun에 저장하는 구조 #한글자 초과 명사만 추출

```
token_list.append(pos)
```

```
token_noun.append(noun)
```

#형태소 분석기들은 다음과 같은 메서드를 공통적으로 제공합니다

- nouns : 명사 추출

- morphs : 형태소 추출

- pos : 형태소에 품사 부착 후 추출

[illegible]

| 127/217 [00:05<00:15, 5.84it/s]

4. 토큰화

■ 토큰화 결과

형태소에 품사를 부착하여 저장

token_list

```
[('아', 'IC'),  
 ('무', 'XPN'),  
 ('생각', 'NNG'),  
 ('없이', 'MAG'),  
 ('살다가', 'NNP'),  
 ('어느덧', 'MAG'),  
 ('대', 'NNB'),  
 ('중반', 'NNG'),  
 ('이', 'JKS'),  
 ('되', 'VV'),  
 ('엇', 'EP'),  
 ('사오', 'EP'),  
 ('보니다', 'EC'),  
 ('대', 'NNB'),  
 ('첫', 'MM'),  
 ('직장', 'NNG'),  
 ('오', 'VXN')]
```

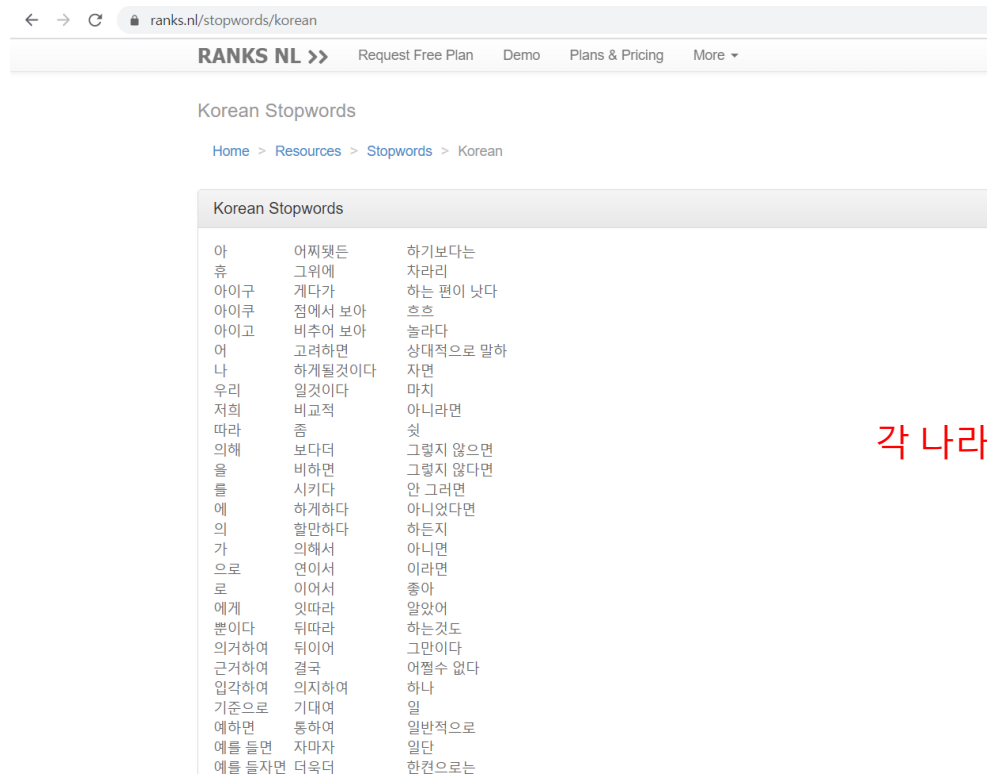
명사 저장

token_noun

```
[('생각',  
 '살다가',  
 '중반',  
 '직장',  
 '그때',  
 '개인연금',  
 '시작',  
 '생각',  
 '연금',  
 '생각',  
 '보험',  
 '회사',  
 '연금',  
 '저축',  
 '이유',  
 '물가',  
 '상승',
```

5. 불용어 처리

- 불용어 사전 : www.ranks.nl/stopwords/korean
복사하여 메모장에 txt파일로 저장



The screenshot shows the RANKS NL website with the URL www.ranks.nl/stopwords/korean in the browser address bar. The page title is "Korean Stopwords". Below the title, there is a navigation menu with links: Home > Resources > Stopwords > Korean. The main content area is titled "Korean Stopwords" and contains a list of 30 Korean stopwords arranged in three columns.

Korean Stopwords		
아	어찌됐든	하기보다는
휴	그위에	차라리
아이구	게다가	하는 편이 낫다
아이쿠	점에서 보아	흐흐
아이고	비추어 보아	놀라다
어	고려하면	상대적으로 말하
나	하게될것이다	자면
우리	일것이다	마치
저희	비교적	아니라면
따라	좀	쉴
의해	보다더	그렇지 않으면
을	비하면	그렇지 않다면
를	시키다	안 그러면
에	하게하다	아니었다면
의	할만하다	하든지
가	의해서	아니면
으로	연이서	이라면
로	이어서	좋아
에게	잇따라	알았어
뿐이다	뒤따라	하는것도
의거하여	뒤이어	그만이다
근거하여	결국	어쩔수 없다
입각하여	의지하여	하나
기준으로	기대여	일
예하면	통하여	일반적으로
예를 들면	자마자	일단
예를 들자면	더욱더	한편으로는

각 나라 언어별로 불용어 사전 등록

5. 불용어 처리

■ 불용어 사전 불러오기 (불용어사전 열어서 불러오기)

```
#불용어 처리
#불용어 처리해주기 위해 불용어 사전 불러오기
f = open(file_path + "stopwords-ko.txt", "r", encoding="UTF-8")
st = f.readlines() #한줄, 한줄이 각각 리스트의 원소로 들어감
f.close()
st
```

수작업으로 불용어 데이터를 복사하여 txt 파일로 저장

```
['.coㄹn',
'coㄹn',
'아ㄹn',
'휴ㄹn',
'아이구ㄹn',
'아이쿠ㄹn',
'순위ㄹn',
'거래ㄹn',
'네이버ㄹn',
'아이고ㄹn',
'어ㄹn',
'나ㄹn',
'포토ㄹn',
'댓글ㄹn',
'우리ㄹn',
'저희ㄹn',
'입니다ㄹn',
'있는ㄹn',
'있습니다ㄹn',
```

readlines : 파일 내용 전체를 가져와 리스트로 출력
readline : 파일의 한 줄을 가져와 문자열로 출력
read : 파일 내용 전체를 가져와 문자열로 출력

5. 불용어 처리

■ 불용어 개행(줄바꿈, \n)문자 제거

```
# rstrip('Str'): 인자로 전달된 문자를 문자열 우측의 가장 첫 번째로 오는 문자열부터 차례대로 삭제를 시도하는 함수  
# lstrip('Str'): 인자로 전달된 문자를 문자열 좌측의 가장 첫 번째로 오는 문자열부터 차례대로 삭제를 시도하는 함수  
# strip('Str'): 인자로 전달된 문자를 문자열 양측의 가장 첫 번째로 오는 문자열부터 차례대로 삭제를 시도하는 함수
```

```
stw = []  
for i in range(0, len(st)):  
    stw.append(st[i].rstrip('\n')) #strip(): 문자열에서 특정 문자를 제거  
stw
```

```
['.co\n',  
 'co\n',  
 '아\n',  
 '휴\n',  
 '아이구\n',  
 '아이쿠\n',  
 '순위\n',  
 '거래\n',  
 '네이버\n',  
 '아이고\n',  
 '어\n',  
 '나\n',  
 '포토\n',  
 '댓글\n',  
 '우리\n',  
 '...']
```



```
['.co',  
 'co',  
 '아',  
 '휴',  
 '아이구',  
 '아이쿠',  
 '순위',  
 '거래',  
 '네이버',  
 '아이고',  
 '어',  
 '나',  
 '포토',  
 '댓글',  
 '우리',  
 '...']
```

5. 불용어 처리

■ 불용어 제거

- 토큰화로 추출한 각 문서의 명사리스트에서 불용어를 제거

```
for word in stw:
    for i in range(0, len(token_noun)):
        while word in token_noun[i]:
            token_noun[i].remove(word)
```

■ 데이터프레임에 추출한 품사리스트와 명사리스트를 추가

```
docs['token_list_pos'] = token_list    # 'token_list_pos' 열로 생성
docs['token_noun'] = token_noun        # 'token_noun' 열로 생성
```


6. 데이터프레임 생성, 저장

■ 데이터프레임 저장 후 불러오기

```
import pickle
f = open(file_path + "total_doc.pkl", "wb")
pickle.dump(docs, f)
f.close()
```

```
import pickle
f = open(file_path + 'total_doc.pkl', "rb")
data = pickle.load(f)
f.close()
```

	title	doc	url	ch	ch2	token_list_pos	token_noun
0	메리츠증권 연금저축펀드 계좌 개설 하기!	아 무 생각없이 살다가 어느덧 대 중반이 되었...	https://blog.naver.com/alice0953?Redirect=Log&...	naver	blog	[(아, IC), (무, XPN), (생각, NNG), (없이, MAG), (살다 가...	[살다가, 중 반, 직장, 개 인연금, 시 작, 보험, 회 사, 저축, 이 유, 물가, 상 승...
1	암환자 장애연금 받기 (1 년 6개월 경과시 / 금액 / 기간)	안녕하세요 여러분 오늘은 좋은 소 식이 있어서 글을 올려봐요 기억하 실지...	https://blog.naver.com/glmss9?Redirect=Log&log...	naver	blog	[(안녕하세요, NNP), (여러분, NNP), (오늘, NNG), (은, JX),...	[소식, 기억, 환자, 장애, 신청, 기억, 개월, 심사, 기간, 장애, 지금, 결...
2	개인연금저축보험 세액공	안녕하세요 웅이의 금융브리핑을 과리하는 김우비지점장 이사드리	https://blog.naver.com/bmw010208?	naver	blog	[(안녕하세요, NNP), (웅이의, NNG), (은, JX),...	[금융, 브리 핑, 관리, 김 웅, 지점장, 어차, 드림...