Chap07.
Text Mining

작성자 : 김진성

Text Mining 4단계

1. 문서 수집(Crawling)
2. 형태소 분석(KONLPY)
3. 시각화(Word Cloud)
4. 희소행렬(Sparse Matrix)

1. 문서 수집

1) Html Parsing
2) BeautifulSoup 패키지
3) 형태소 모듈 테스트

Html Parsing Web Crawling

import urllib.request # url 요청 모듈
from lxml.html import parse # html 양식으로 파싱
from io import StringIO # 문자열 입출력 모듈

# 1. web 문서를 source(text문서) 로 가져오기
url = *"http://media.daum.net/"*
#url = "http://news.naver.com/"

# 1) html source 가져오기
res = urllib.request.urlopen(url) # web 문서 get
# requests.get(url)
data = res.read() # binary 형태로 읽음
#print(data) # b'\n<!doctype html>\n

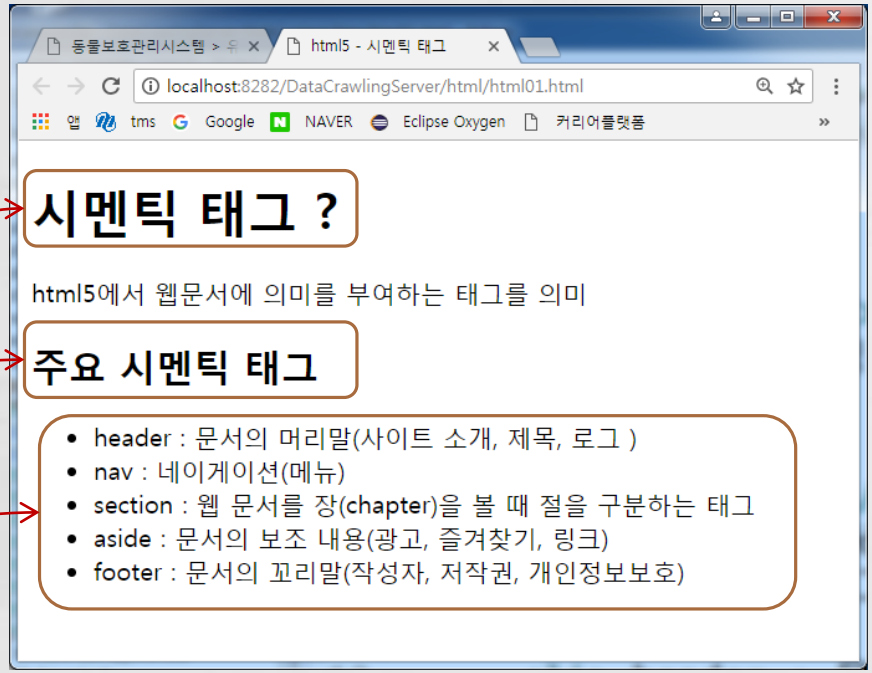
# 2) html 문서열로 변환(파싱)
text = data.decode(*"utf-8"*)
text_source = StringIO(text)
parsed = parse(text_source)
print(parsed)

# 3) root node 찾기
root_node = parsed.getroot()

```
# 2. html의 <a>태그 가져오기
# 형식) root_node.findall(".//태그")
links = root_node.findall(".//a")
print('링크수 : ', len(links)) # 링크수 :  202
print(links) # 202 링크 element object

# 3. 'href' 속성값 가져오기
# 형식) obj.get('속성')
link_url = [] # 속성값을 저장
cnt = 1
for link in links :
    print(cnt, '->', link.get('href'))
    link_url.append(link.get('href')) # 내용 추가
    cnt += 1
print(link_url) # 전체 내용 출력

# 4. <a>태그 내용 가져오기
cnt = 1
centents = []
for link in links :
    print(cnt, '->', link.text_content().strip())
    cnt += 1
    centents.append(link.text_content().strip())
```

BeautifulSoup Web Crawling

```
import urllib.request
from bs4 import BeautifulSoup

url = 'http://localhost:8282/DataCrawlingServer/html/html01.html'

# 1. html source 가져오기
res = urllib.request.urlopen(url) # web 문서 get
data = res.read() # binary 형태로 읽음

# 2. html 파싱
html = data.decode("utf-8") # 디코딩
soup = BeautifulSoup(html, 'html.parser') # html source 파싱

# 3. 태그 내용 가져오기

# 1) 태그 <h1> 가져오기
h1 = soup.html.body.h1
print('h1 :', h1.string) # h1 :  시멘틱 태그?

# 2) find()함수로 찾기
h2 = soup.find("h2")
print("h2 :", h2.string) # h2 :  주요 시멘틱 태그

li = soup.find("li")
print(li.string) # header : 문서의 머리말(사이트 소개, 제목, 로그 )

# 2) find_all() 함수로 여러개 찾기 : list 반환
li2 = soup.find_all("li")
print(li2) # [<li> header : 문서의 머리말(사이트 소개, 제목, 로그 )</li>,, '"]
# print(li2.string) # error 발생

for li in li2 :
    print(li.string)
```



- # url query 이용하기

검색조건 : 2015~2018년도/서울시/강남구/개

http://www.animal.go.kr/portal_rnl/abandonment/public_list.jsp?s_date=2015-01-01&e_date=2018-01-12&s_upr_cd=6110000&s_org_cd=3220000&s_up_kind_cd=417000&s_kind_cd=&s_name=&s_shelter_cd=&s_wrk_cd=&s_state=&s_state_hidden=&pagecnt=48

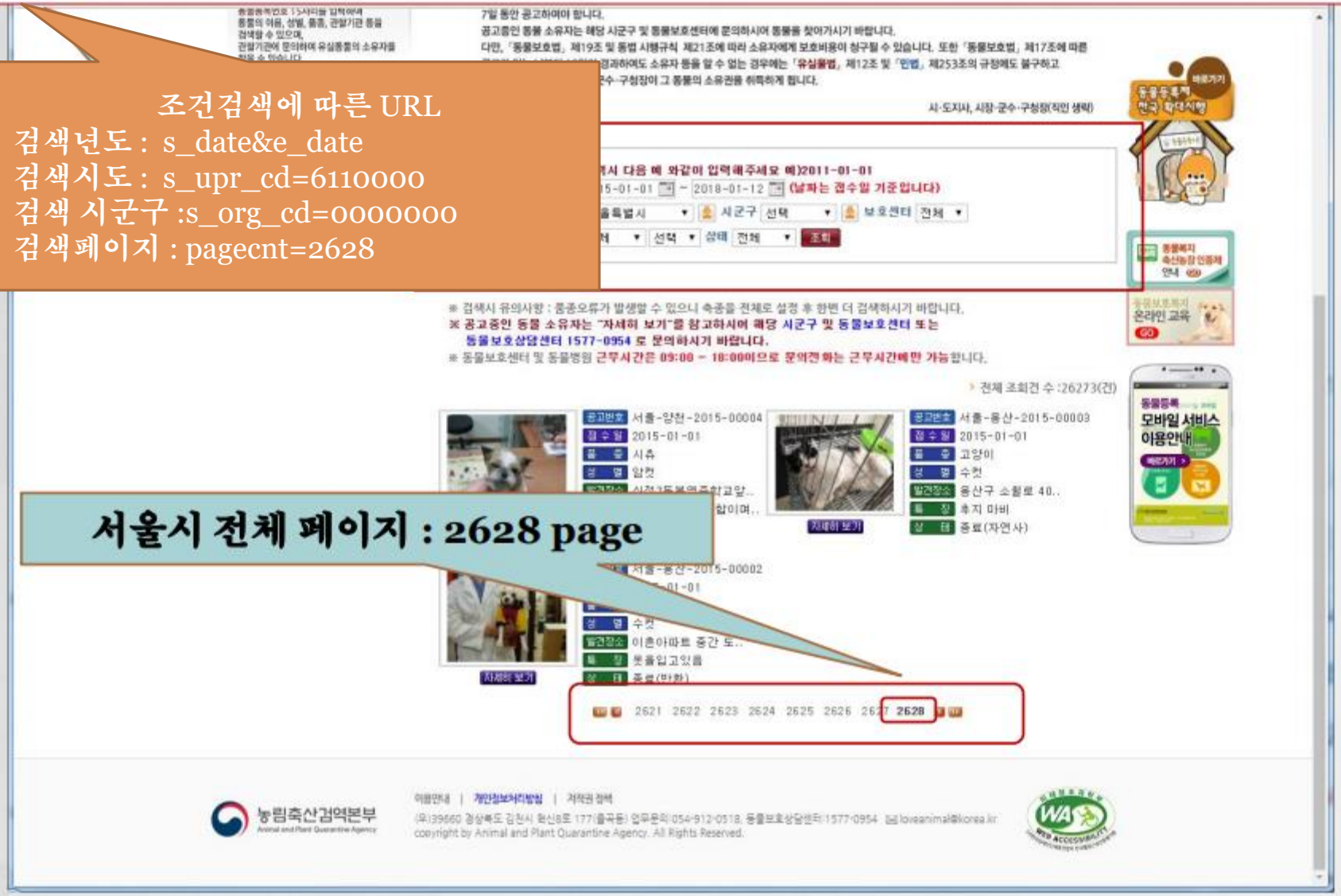
조건검색에 따른 URL
검색년도 : s_date&e_date
검색시도 : s_upr_cd=6110000
검색 시군구 :s_org_cd=3220000
검색페이지 : pagecnt=48

전체 검색 페이지 48 페이지[현재 : 48page]


검색조건 : 2015~2018년도/서울시/전체/전체
SEARCH
날자 입력시 다음 예 와같이 입력해주세요 예)2011-01-01
날자 2015-01-01 ~ 2018-01-12 (날자는 접수일 기준입니다)
시도 서울특별시 시군구 선택 보호센터 전체
축종 전체 선택 상태 전체 조회
※ 검색시 유의사항 : 품종오류가 발생할 수 있으니 축종을 전체로 설정 후 한번 더 검색하시기 바랍니다.
※ 공고중인 동물 소유자는 "자세히 보기"를 참고하시어 해당 시군구 및 동물보호센터 또는 동물보호상담센터 1577-0954 로 문의하시기 바랍니다.
※ 동물보호센터 및 동물병원 근무시간은 09:00 ~ 18:00이므로 문의전화는 근무시간에만 가능합니다.
전체 조회건 수 :26273(건)
공고번호 서울-양천-2015-00004
접수일 2015-01-01
품종 시츄
성별 암컷
발견장소 신정7동봉영중학교앞..
특징 치석있고부정교합이며..
상태 종료(반환)
공고번호 서울-용산-2015-00003
접수일 2015-01-01
품종 고양이
성별 수컷
발견장소 용산구 소월로 40..
특징 후지 마비
상태 종료(자연사)
공고번호 서울-용산-2015-00002
접수일 2015-01-01
품종 말티즈
성별 수컷
발견장소 이촌아파트 중간 도..
특징 옷을입고있음
상태 종료(반환)
2621 2622 2623 2624 2625 2626 2627 2628
농림축산검역본부
개인정보처리방침
copyright by Animal and Plant Quarantine Agency. All Rights Reserved.

http://www.animal.go.kr/portal_rnl/abandonment/public_list.jsp?s_date=2015-01-01&e_date=2018-01-12&s_upr_cd=6110000&s_org_cd=0000000&s_up_kind_cd=&s_kind_cd=&s_name=&s_shelter_cd=&s_wrk_cd=&s_state=&s_state_hidden=&pagecnt=2628

조건검색에 따른 URL
검색년도 : s_date&e_date
검색시도 : s_upr_cd=6110000
검색 시군구 :s_org_cd=0000000
검색페이지 : pagecnt=2628



유기견 자료 Crawling 대상 문서

'div[class=thumb_inner02] > dl[class=thumbnail_table01]'

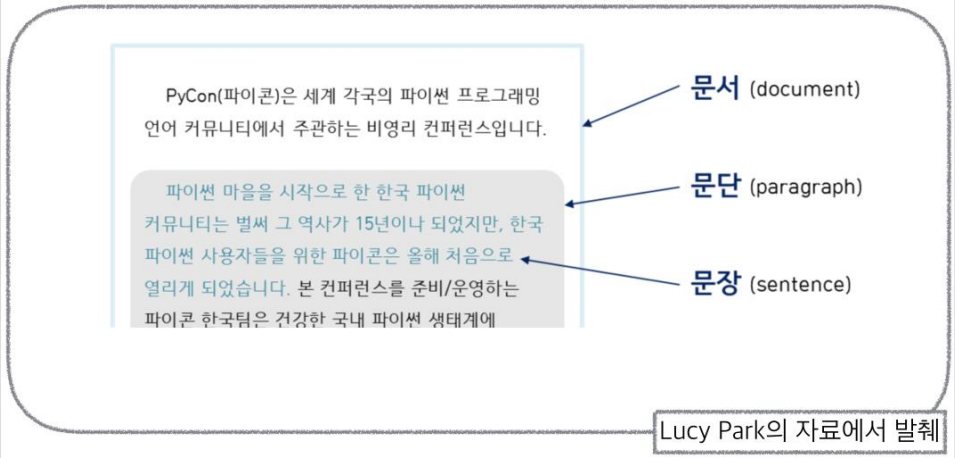
7개 칼럼으로 DataFrame 생성

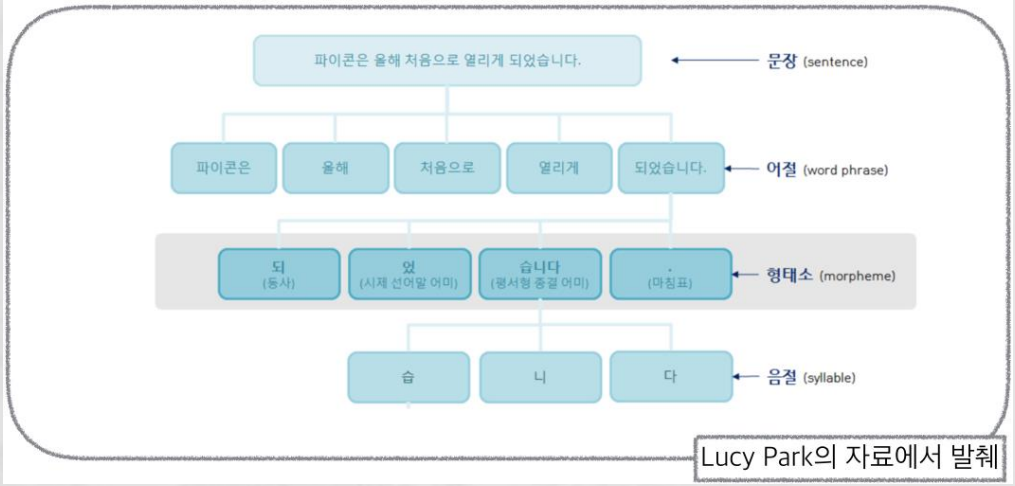
```
<div class="thumb_inner02">
 <dl class="thumbnail_table01">
  <dt class="thumbnail_img02">
    <img src="/portal_rnl/images/board/thumbnail_stitle01.gif" alt="공고번호" /></dt>
  <dd>서울-서초-2017-00092</dd>
  <dt class="thumbnail_img02">
    <img src="/portal_rnl/images/board/thumbnail_stitle02.gif" alt="접수일" /></dt>
  <dd>2017-06-30</dd>
  <dt class="thumbnail_img02"><img src="/portal_rnl/images/board/thumbnail_stitle03.gif" alt="품종" /></dt>
  <dd> 기타축종</dd>
  <dt class="thumbnail_img02"><img src="/portal_rnl/images/board/thumbnail_stitle04.gif" alt="성별" /></dt>
  <dd>미상</dd>
  <dt class="thumbnail_img02"><img src="/portal_rnl/images/board/thumbnail_stitle05.gif" alt="발견장소" /></dt>
  <dd>반포동 두리동물병원..</dd>
  <dt class="thumbnail_img02"><img src="/portal_rnl/images/board/thumbnail_stitle06.gif" alt="특징" /></dt>
  <dd>총19마리리빙박스에..</dd>
  <dt class="thumbnail_img02"><img src="/portal_rnl/images/board/thumbnail_stitle07.gif" alt="상태" /></dt>
  <dd> 종료(입양) </dd>
 </dl>
</div>
```

2. 형태소 분석

1) 형태소 분석 개요
2) 형태소 분석 관련 패키지 설치
3) 형태소 모듈 테스트

1) 형태소 분석 개요


PyCon(파이콘)은 세계 각국의 파이썬 프로그래밍 언어 커뮤니티에서 주관하는 비영리 컨퍼런스입니다.
파이썬 마을을 시작으로 한 한국 파이썬 커뮤니티는 벌써 그 역사가 15년이나 되었지만, 한국 파이썬 사용자들을 위한 파이콘은 올해 처음으로 열리게 되었습니다. 본 컨퍼런스를 준비/운영하는
문서 (document)
문단 (paragraph)
문장 (sentence)
Lucy Park의 자료에서 발췌


파이콘은 올해 처음으로 열리게 되었습니다.
문장 (sentence)
파이콘은
올해
처음으로
열리게
되었습니다.
어절 (word phrase)
되 (동사)
었 (시제 선어말 어미)
습니다 (평서형 종결 어미)
. (마침표)
형태소 (morpheme)
습
니
다
음절 (syllable)
Lucy Park의 자료에서 발췌

2) 형태소 분석 Library 설치

1) Java(JDK)설치

2) Jpype 패키지 설치 : Python에서 java 가상머신 사용(Python+Java)

https://www.lfd.uci.edu/~gohlke/pythonlibs/

3) KoNLPy 패키지 설치

Anaconda prompt> pip install konlpy

1) JDK 설치


Java SE Development Kit 8 Update 71 - Setup
Java
ORACLE
Welcome to the Installation Wizard for Java SE Development Kit 8 Update 71
This wizard will guide you through the installation process for the Java SE Development Kit 8 Update 71.
The Java Mission Control profiling and diagnostics tools suite is now available as part of the JDK.
Next >
Cancel
recipes4dev.tistory.com

2) Jpype 패키지 설치

CTRL + F

안전함 | https://www.lfd.uci.edu/~gohlke/pythonlibs/

jpype 1/12

Unofficial Windows Binaries for Python Extension Packages

by **Christoph Gohlke**, **Laboratory for Fluorescence Dynamics**, **University of California, Irvine**.

This page provides 32- and 64-bit Windows binaries of many scientific open-source extension packages for the official CPython distribution of the Python programming language. A few binaries are available for the PyPy distribution.

The files are unofficial (meaning: informal, unrecognized, personal, unsupported, no warranty, no liability, provided "as is") and made available for testing and evaluation purposes.

Most binaries are built from source code found on PyPI or in the projects public revision control systems. Source code changes, if any, have been submitted to the project maintainers or are included in the packages.

Refer to the documentation of the individual packages for license restrictions and dependencies.

If downloads fail, reload this page, enable JavaScript, disable download managers, disable proxies, clear cache, use Firefox, reduce number and frequency of downloads. Please only download files manually as needed.

Use pip version 9 or newer to install the downloaded .whl files. This page is not a pip package index.

Many binaries depend on numpy-1.14+mkl and the Microsoft Visual C++ 2008 (x64, x86, and SP1 for CPython 2.7), Visual C++ 2010 (x64, x86, for CPython 3.4), or the Visual C++ 2017 (x64 or x86 for CPython 3.5, 3.6, and 3.7) redistributable packages.

Install numpy+mkl before other packages that depend on it.

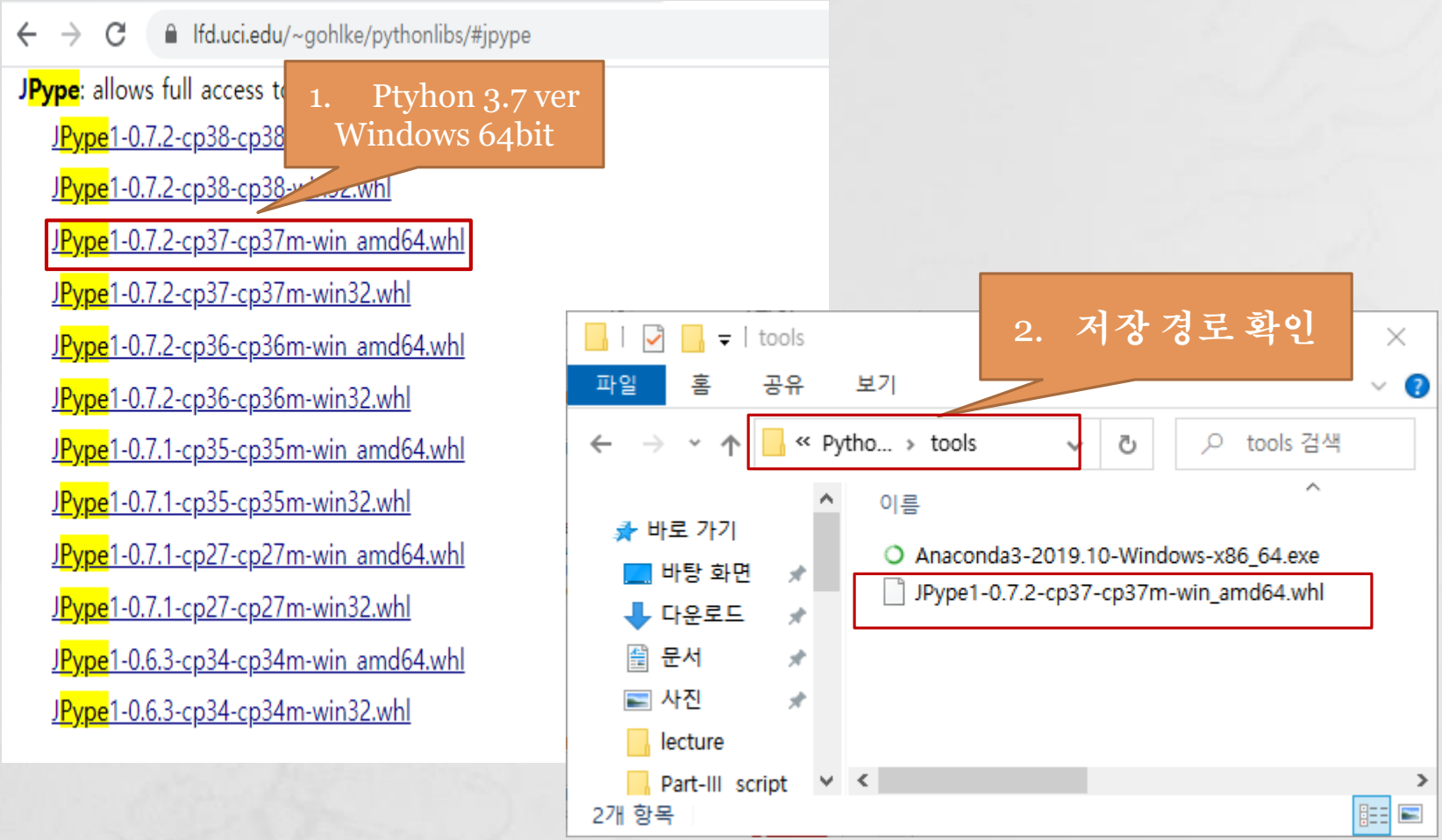
The binaries are compatible with the most recent official CPython distributions on Windows >=6.0. Chances are they do not work with custom Python distributions included with Blender, Maya, ArcGIS, OSGeo4W, ABAQUS, Cygwin, Pythonxy, Canopy, EPD, Anaconda, WinPython etc. Many binaries are not compatible with Windows XP or Wine.

The packages are ZIP or 7z files, which allows for manual or scripted installation or repackaging of the content.

The files are provided "as is" without warranty or support of any kind. The entire risk as to the quality and performance is with you.

The opinions or statements expressed on this page should not be taken as a position or endorsement of the Laboratory for Fluorescence Dynamics or the University of California.

Index by date: mplcairo imread rtree winrandom pywavelets assimulo pyfmi pyfm pymatgen jupyter cobra fast-histogram discretize cairocffi h5py boost.python hmmlearn kiwisolver ruamel.yaml pythonmagick mkl_fft pyamg polylearn pythonnet cellprofiler cvxcanon cupy pycuda scs pygame peewee pytiff blist tensorflow multineat openexr qutip openbabel mahotas noise scikits.vectorplot pyfits reportlab psycopg markupsafe pyrsistent fiona greenlet nitime pymongo zipline gvar matplotlib zs grako lru_dict scikit-misc pyvrml97 pyjnius sparsesvd pyephem seqlearn obspy cyordereddict pystruct gevent apsw numexpr kivy lsqfit scikit-learn open3d sqlalchemy cx_oracle rapidjson numpy iminuit twisted scimath traits chaco enable pyrxp backports.lzma regex xxhash pyzmq dulwich tornado arctic rasterio sympy cython numpy-quaternion lz4 pycairo pandas mpi4py bitarray numba llvmlite gensim fisx mayavi pyyaml slycot pymol orange opencv hyperspy bsddb3 python-snappy pillow protobuf grpcio llist ceodbc liblas holopy qt_graph_helpers guiqwt veusz pyqwt pyqt4 simplejson moderngl rpy2 param pymvpa pymetis fabio mysqlclient lxml indexed_gzip pyasn biopython bokeh scikit-cycling ad3 bsdiff4 multiprocess sfepy aggdraw yarl tifffile pillow-simd cvxpy aiohttp entropy pytables openpiv pywinpty basemap iris stratify psutil pygit2 cf-units pocketsphinx pywin32 xgboost astropy pendulum babel vlfd scikit-image python-ldap python-lzo mlpy milk javabridge vtk osqp ecos zodbpickle trollius cftime statsmodels quickfix spglib zstd sounddevice tatsu multidict brotlipy scipy pgmagick openimageio netifaces netcdf4 dipy pygresql debug-information-files chompack cvxopt vitables pyopencl pytorch curses menpo swiglpk btrees faulthandler thriftpy gmpy zope.interface brotli pip gdal logbook marisa-trie bcolz ets pyodesys ta-lib spacy ujson numcodecs py-lmdb mercurial simpleitk mod_wsgi jpype fastparquet pyodbc pyhdf freesasa pymssql pyldap wordcloud meshpy tomopy cytoolz cheetah xylib-py cyrasterize pyswisseph pulp cantera cchardet pycluster ode salientdetect liblinear libsvm setproctitle cffi cdecimal crcmod crc16 pycld2 planar autopy pyx rtmidi-python pycosat pyflux mkl-service postgresadapter datrie polygon py-earth lightning pystemmer pyqpbo pyopencl pydde x86cpu gpy fisher ffnet fasttext pymc hddm heatmap jsonlib intbitset sasl flann msgpack cartopy scikits.odes louvain-igraph python-igraph pycares pybox2d natgrid pycurl yt bintrees scandir pycifrw coverage lp_solve aspell-python transformations chebyfit vidsrc psf akima pykinsol pyodeint pycvodes fastcache fdint jcc twainmodule triangle scikit-fmm python-levenshtein pyspharm pyminuit pymcubes pylzma pyhook pyeda pyfltk simpleparse nlopt pyaudio thrift pyicu atom pyemd enaml shapely pypmc wrf_python quantlib mkl_random kwant tinyarray udunits spectrum recordclass kapteyn blosc libsbml simpleaudio pylibtiff line_profiler persistent cx_freeze videocapture pyproj fastrlock minepy fann2 mistune lazy_object_proxy wrapt bottleneck scikit-umfpack czifile gr pyarrow pycorrfit pyside vigra imaged11 python-cjson py_gd freeimagedll nipy qimage2ndarray libtfr lfdfiles mathutils yappi pyfftw pyviennacl blz bigfloat cyassimp sima pymca friture pycogent pysqlite blaze scikits.audiolab la bazaar dynd genshi python-sundials glumpy pyamf libxml-python cellcognition pymcmc pyksvd pybluez pygraphviz mxbase libpython re2 pymunk pygtk cgal-bindings bio_formats pysfml pyexiv2 pylibdeconv iocbio pymix umysql lazyflow mmlib scikits.timeseries casuarius wxpython ilastik pywcs scientificpython vpython nmoldyn mmtk pyalembic polymode scikits.delaunay cld py-fcm oursql zfec py2exe pymutt carray llvmpy cgkit pymedia scipy-cluster scikits.scattpy scikits.samplerate scikits.ann pyxml pytst delny mysql-python htseq pyusb-ftdi silvercity steps pysparse pyropes scikits.hydroclimpy sendkeys pydbg pyisapie


lfd.uci.edu/~gohlke/pythonlibs/#jpype
JPype: allows full access t
JPype1-0.7.2-cp38-cp38
JPype1-0.7.2-cp38-cp38-
.whl
JPype1-0.7.2-cp37-cp37m-win_amd64.whl
JPype1-0.7.2-cp37-cp37m-win32.whl
JPype1-0.7.2-cp36-cp36m-win_amd64.whl
JPype1-0.7.2-cp36-cp36m-win32.whl
JPype1-0.7.1-cp35-cp35m-win_amd64.whl
JPype1-0.7.1-cp35-cp35m-win32.whl
JPype1-0.7.1-cp27-cp27m-win_amd64.whl
JPype1-0.7.1-cp27-cp27m-win32.whl
JPype1-0.6.3-cp34-cp34m-win_amd64.whl
JPype1-0.6.3-cp34-cp34m-win32.whl
1. Ptyhon 3.7 ver
Windows 64bit
tools
파일
홈
공유
보기
2. 저장 경로 확인
« Pytho... › tools
tools 검색
바로 가기
바탕 화면
다운로드
문서
사진
lecture
Part-III script
이름
Anaconda3-2019.10-Windows-x86_64.exe
JPype1-0.7.2-cp37-cp37m-win_amd64.whl
2개 항목

- Jpype 패키지 설치

Anaconda Prompt (Anaconda3)

```
(base) C:₩Users₩user>pip install D:₩Mega1T₩Python_ML₩tools₩JPype1-0.7.2-cp37-cp37m-win_amd64.whl
Processing d:₩megait₩python_ml₩tools₩jpype1-0.7.2-cp37-cp37m-win_amd64.whl
Installing collected packages: JPype1
  Found existing installation: JPype1 0.7.0
    Uninstalling JPype1-0.7.0:
      Successfully uninstalled JPype1-0.7.0
Successfully installed JPype1-0.7.2

(base) C:₩Users₩user>
```

3) KoNLPy 패키지 설치

Anaconda Prompt

관리자: Anaconda Prompt

```
(base) C:WUsersWUSER>pip install konlpy
Collecting konlpy
  Downloading https://files.pythonhosted.org/packages/b1/41/73127de031d710fa6fc6
40cc4d4d399977e7a96423131fcd180b9f69627c/konlpy-0.4.4-py2.py3-none-any.whl (22.5
MB)
    100% |                                        | 22.5MB 820kB/s
Installing collected packages: konlpy
Successfully installed konlpy-0.4.4
You are using pip version 9.0.1, however versio
You should consider upgrading via the 'python -
and.

(base) C:WUsersWUSER>
```

C:\Anaconda3\Lib\site-packages 폴더 확인

Anaconda3 ▸ Lib ▸ site-packages ▸

공유 대상 ▾ 새 폴더

이름	수정한 날짜
vboxapi	2018-07-03 오전 10:15
django	2018-07-16 오전 9:14
Django-2.0.7.dist-info	2018-07-16 오전 9:14
JPype1-0.6.3.dist-info	2018-07-23 오전 9:46
jpypex	2018-07-23 오전 9:46
jpype	2018-07-23 오전 9:52
konlpy	2018-07-23 오전 9:56
konlpy-0.4.4.dist-info	2018-07-23 오전 9:56
test	2018-07-23 오전 9:56
heapdict.py	2009-10-17 오전 1:25
singledispatch.py	2014-03-19 오전 4:43
singledispatch_helpers.py	2014-03-19 오전 4:43
itsdangerous.py	2014-03-28 오후 9:08

오전 9:56

3) 형태소 모듈 TEST

```
'''
Jpype1 패키지 설치 테스트
'''

import jpype
p = jpype.getDefaultJVMPath()
print(p)
# C:\Program
Files\Java\jdk1.8.0_161\jre\bin\server\jvm.dll
```

```
from konlpy.tag import Kkma

# Kkma class의 object 생성
kkma = Kkma()

# 문단 -> 문장 추출
para = "형태소 분석을 시작합니다. 나는 홍길동이고 age는 28세 입니다."
ex_sent = kkma.sentences(para) # 문단 -> 문장
print(ex_sent)
'''
['형태소 분석을 시작합니다.', '나는 홍길동이고 age는 28세 입니다.']
'''

# 문단 -> 명사 추출
ex_nouns = kkma.nouns(para) # 문단 -> 명사
print(ex_nouns)
'''
['형태소', '분석', '나', '홍길동', '28', '28세', '세']
'''

# 문단 -> 형태소 추출
ex_pos = kkma.pos(para) # 문단 -> 형태소
print(ex_pos) #  (형태소, 품사)
'''
[('형태소', 'NNG'), ('분석', 'NNG'), ('을', 'JKO'), ('시작하', 'VV'), ('ㅂ니다', 'EFN'), ('.', 'SF'), ('나',
'NP'), ('는', 'JX'), ('홍길동', 'NNG'), ('이', 'VCP'), ('ㅂ니다', 'EFN'), ('.', 'SF')]
'''
```

'''

형태소 : 언어에 있어서 분해 가능한 최소한의 의미를 가진 단위

NNG 일반 명사 NNP 고유 명사 NNB 의존 명사 NR 수사 NP 대명사 VV 동사

VA 형용사 VX 보조 용언 VCP 긍정 지정사 VCN 부정 지정사 MM 관형사

MAG 일반 부사 MAJ 접속 부사 IC 감탄사 JKS 주격 조사 JKC 보격 조사

JKG 관형격 조사 JKO 목적격 조사 JKB 부사격 조사 JKV 호격 조사

JKQ 인용격 조사 JC 접속 조사 JX 보조사 EP 선어말어미 EF 종결 어미

EC 연결 어미 ETN 명사형 전성 어미 ETM 관형형 전성 어미 XPN 체언 접두사

XSN 명사파생 접미사 XSV 동사 파생 접미사 XSA 형용사 파생 접미사 XR 어근

SF 마침표, 물음표, 느낌표 SE 줄임표 SS 따옴표,괄호표,줄표

SP 쉼표,가운뎃점,콜론,빗금 SO 붙임표(물결,숨김,빠짐)

SW 기타기호 (논리수학기호,화폐기호) SH 한자 SL 외국어 SN 숫자

NF <u>명사추정범주 NV 용언추정범주 NA 분석불능범</u>

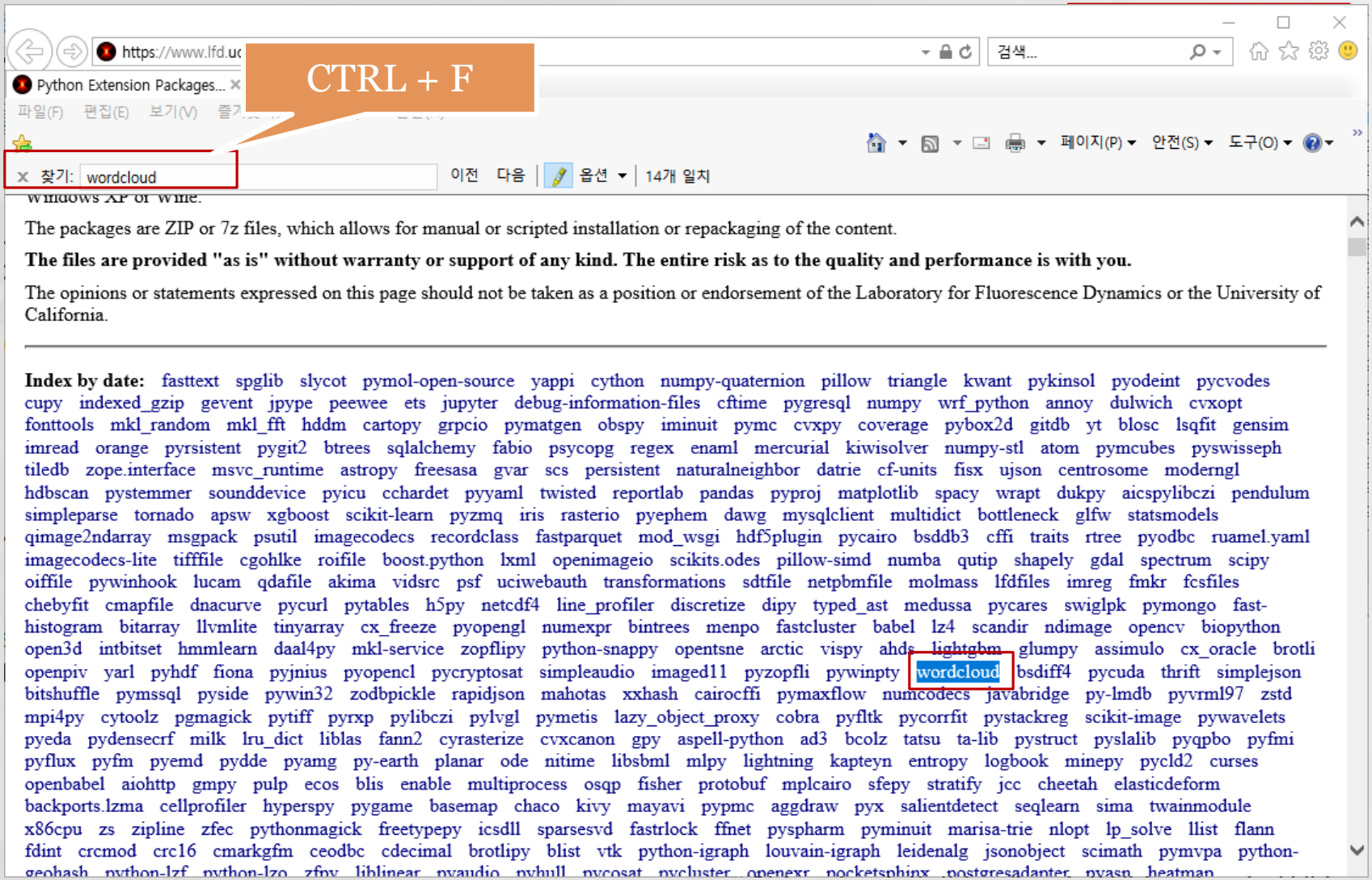
'''

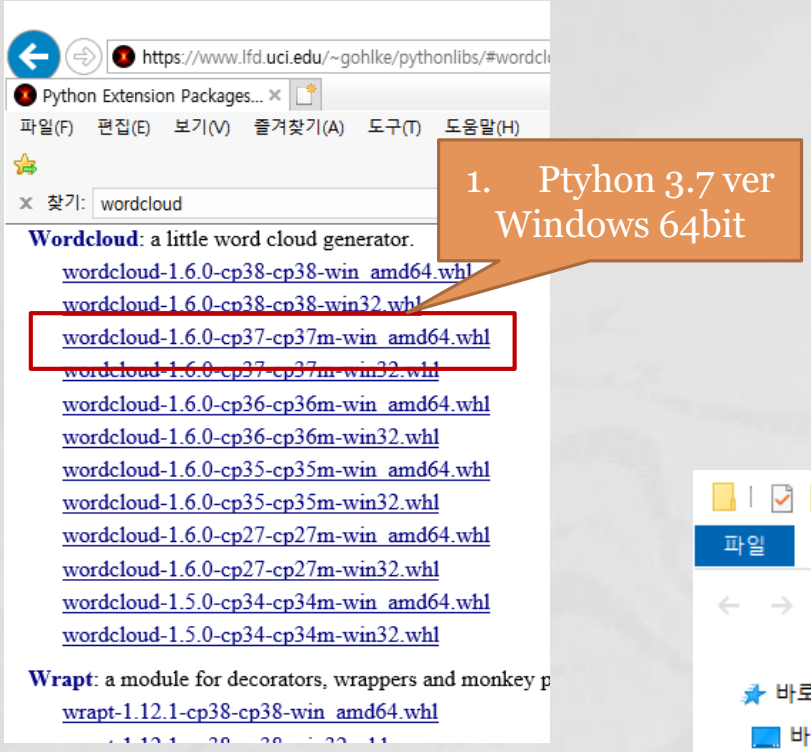
3. 시각화

1) wordCloud 패키지 다운로드 사이트

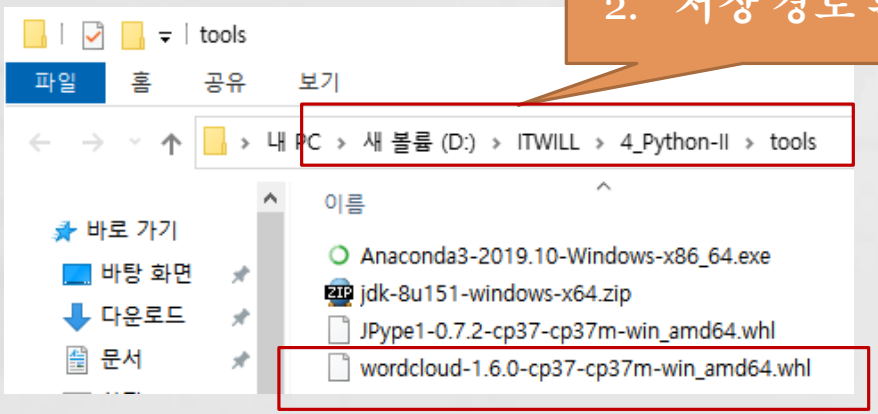
https://www.lfd.uci.edu/~gohlke/pythonlibs/

2) wordCloud 패키지 다운로드





1. Ptyhon 3.7 ver Windows 64bit



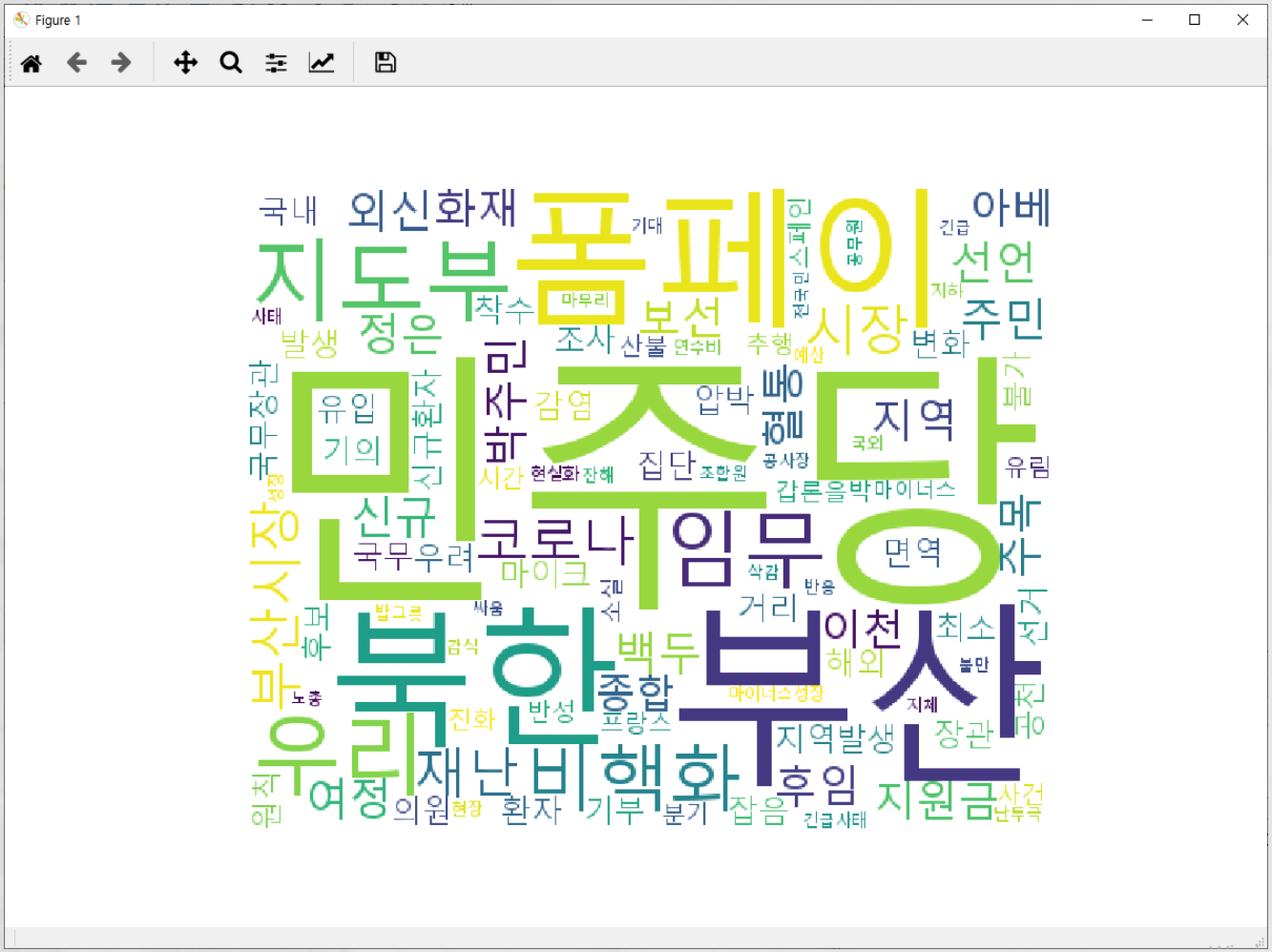
2. 저장 경로 확인

3) wordCloud 패키지 설치

Anaconda Prompt (Anaconda3)

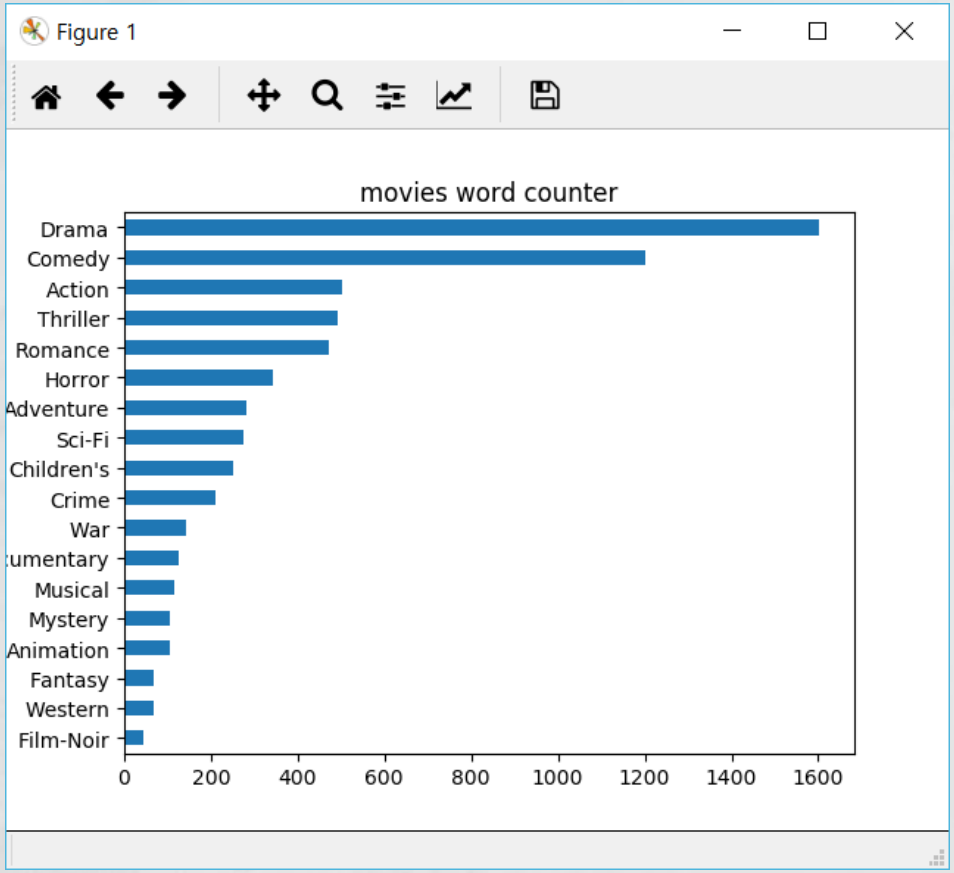
```
(base) C:₩Users₩user>pip install D:₩ITWILL₩4_Python-II₩tools₩wordcloud-1.6.0-cp37-cp37m-win_amd64.whl
Processing d:₩itwill₩4_python-ii₩tools₩wordcloud-1.6.0-cp37-cp37m-win_amd64.whl
Requirement already satisfied: numpy>=1.6.1 in c:₩users₩user₩anaconda3₩lib₩site-packages (from wordcloud=
=1.6.0) (1.17.2)
Requirement already satisfied: pillow in c:₩users₩user₩anaconda3₩lib₩site-packages (from wordcloud==1.6.0
) (6.1.0)
Requirement already satisfied: matplotlib in c:₩users₩user₩anaconda3₩lib₩site-packages (from wordcloud==1
.6.0) (3.1.0)
Requirement already satisfied: cycler>=0.10 in c:₩users₩user₩anaconda3₩lib₩site-packages (from matplotlib
->wordcloud==1.6.0) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:₩users₩user₩anaconda3₩lib₩site-packages (from matpl
otlib->wordcloud==1.6.0) (1.1.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in c:₩users₩user₩anaconda3₩lib₩si
te-packages (from matplotlib->wordcloud==1.6.0) (2.4.0)
Requirement already satisfied: python-dateutil>=2.1 in c:₩users₩user₩anaconda3₩lib₩site-packages (from ma
tplotlib->wordcloud==1.6.0) (2.8.0)
Requirement already satisfied: six in c:₩users₩user₩anaconda3₩lib₩site-packages (from cycler>=0.10->matpl
otlib->wordcloud==1.6.0) (1.12.0)
Requirement already satisfied: setuptools in c:₩users₩user₩anaconda3₩lib₩site-packages (from kiwisolver>=
1.0.1->matplotlib->wordcloud==1.6.0) (41.0.1)
Installing collected packages: wordcloud
Successfully installed wordcloud-1.6.0

(base) C:₩Users₩user>
```

4) wordCloud 시각화



Word Counter

영화 장르별 빈도분석



4. 희소 행렬

	type	text
0	ham	우리나라 대한민국, 우리나라 만세
1	spam	비아그라 500GRAM 정력 최고!
2	ham	나는 대한민국 사람
3	spam	보험료 15000원에 평생 보장 마감 임박
4	ham	나는 홍길동

```
[[ 0.        0.        0.33939315 0.        0.42066906 0.        0.
  0.        0.        0.84133812 0.        0.        0.        0.  0.        0.      ]
 [ 0.5       0.        0.        0.        0.        0.        0.
  0.5       0.        0.        0.        0.        0.5       0.5  0.        0.      ]
 [ 0.        0.53177225 0.53177225 0.        0.        0.        0.
  0.        0.659118   0.        0.        0.        0.        0.  0.        0.      ]
 [ 0.        0.        0.        0.40824829 0.        0.40824829
  0.40824829 0.        0.        0.        0.40824829 0.40824829
  0.        0.        0.40824829 0.      ]
 [ 0.        0.62791376 0.        0.        0.        0.        0.
  0.        0.        0.        0.        0.        0.        0.  0.        0.77828292]]
```

연관 단어

```
# 1. 문서 vs 단어 행렬 만들기
cv = CountVectorizer(max_features=5000)
dtm = cv.fit_transform(texts)

#2. 단어간 상관관계
dtm_corr = np.corrcoef(dtm, rowvar=False)

# 3. 기준 단어 vs 나머지 단어 상관계수
Word_corr = []
For I in range(len(words)) :
   for j in range(i+1, len(words)) :
      word_corr.append( (words[i], words[j], dtm_corr[I, j]))

# 4. 상관계수 기준 오름차순 정렬
word_corr = sorted(word_corr, key=lambda x : x[2])

# 5. 상위 20개 단어 기준 연관어 보기
word_corr[-20:]
```

```
[('가락', '오락', 1.0),
 ('가리기', '옥석', 1.0),
 ('교섭', '교섭단체', 1.0),
 ('국책', '국책은행', 1.0),
 ('노예계약', '대접', 1.0),
 ('대백', '대백병원', 1.0),
 ('대치동', '우등생', 1.0),
 ('동물병원', '목장', 1.0),
 ('동물병원', '제보자', 1.0),
 ('동물병원', '최종목표', 1.0),
 ('대한민국', '서울', 1.0),
 ('매석', '매점', 1.0),
 ('명지', '명지병원', 1.0),
 ('목장', '제보자', 1.0),
 ('목장', '최종목표', 1.0),
 ('버그', '블룸', 1.0),
 ('봉사자', '자원봉사자', 1.0),
 ('브렉', '시트', 1.0)]
```