Week 5: Data Mining

Instructor: Daejin Choi (djchoi@inu.ac.kr)

INCHEON
NATIONAL
UNIVERSITY

Contents

- Introduction to Data Mining
- Practical Tips for Starting Data Mining
- About Term Project

Introduction to Data Mining

Motivation

- The Explosive Growth of Data: from terabytes to petabytes
  - Data collection and data availability
    - Automated data collection tools, Web, computerized society
  - Major sources of abundant data
    - Business: Web, e-commerce, transactions, stocks, …
    - Science: Remote sensing, bioinformatics, …
    - Society and everyone: news, IoT, digital cameras, YouTube
- **We are drowning in data, but starving for knowledge!**
- "Necessity is the mother of invention"—Data mining—Automated analysis of massive data sets

What is Data Mining (DM)?

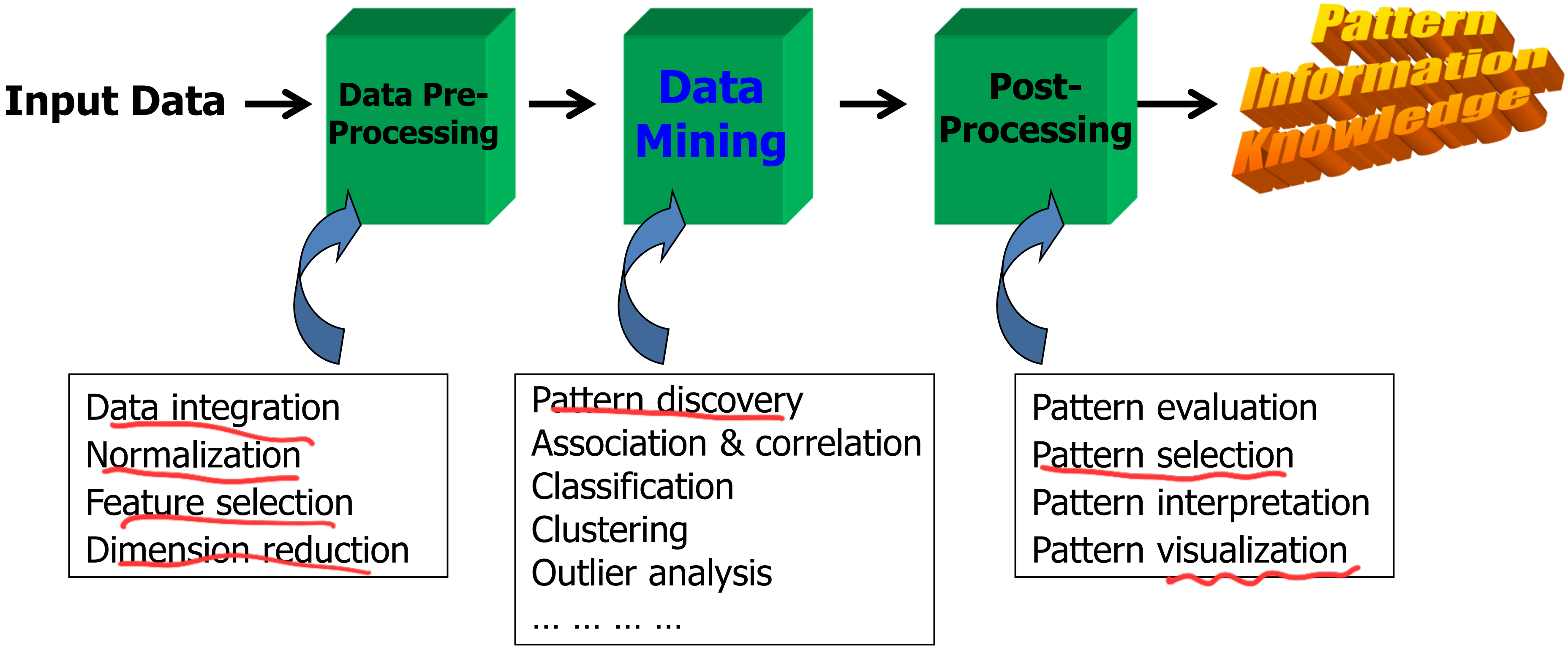
- Data mining (knowledge discovery from data, KDD)
  - Extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) patterns or knowledge from huge amount of data

https://www.rasmussen.edu/degrees/technology/blog/what-is-data-mining/

- Alternative names
  - Knowledge extraction, data analysis, data science, information harvesting, business intelligence, predictive analysis, etc.

KDD Process

KDD Process



Input Data → Data Pre-Processing → Data Mining → Post-Processing → Pattern Information Knowledge

Data integration
Normalization
Feature selection
Dimension reduction

Pattern discovery
Association & correlation
Classification
Clustering
Outlier analysis
… … … …

Pattern evaluation
Pattern selection
Pattern interpretation
Pattern visualization

Key Components in Data Mining

- **Data & Datatypes**
  - Database data (extended-relational, object-oriented, heterogeneous, legacy), data warehouse, transactional data, stream, spatiotemporal, time-series, sequence, text and web, multi-media, graphs & social and information networks

- **Data mining functions** 비교 연관 분석
  - Characterization, discrimination, association, classification, clustering, trend/deviation, outlier analysis, etc.
  - Descriptive vs. predictive data mining
    ↳통계
  - Multiple/integrated functions and mining at multiple levels

→ 어떻게 판단 할 것이냐

Key Components in Data Mining (Cont'd)

- **Techniques utilized**
  - Data-intensive, machine learning, statistics, pattern recognition, visualization, high-performance, etc.

- **Applications adapted**
  - Retail, telecommunication, banking, fraud analysis, bio-data mining, stock market analysis, health care, etc.

DM Component: Data & Datatypes

- Database-oriented data sets and applications
  - Relational database, data warehouse, transactional database
- Advanced data sets and advanced applications
  - Data streams and sensor data
  - Time-series data, temporal data, sequence data (incl. bio-sequences)
  - Structure data, graphs, social networks and multi-linked data
  - Spatial data and spatiotemporal data
  - Multimedia database
  - Text databases
  - The World-Wide Web

DM Component: DM Functions (1/3)

- Generalization (일반화)
  - Multidimensional concept description: Characterization and discrimination
    - Generalize, summarize, and contrast data characteristics, e.g., dry vs. wet region

- Association and Correlation Analysis
  - Frequent patterns (or frequent itemsets)
    - What items are frequently purchased together in your Walmart?
  - How to mine such patterns and rules efficiently in large datasets?
  - How to use such patterns for classification, clustering, and other applications?

DM Component: DM Functions (2/3)

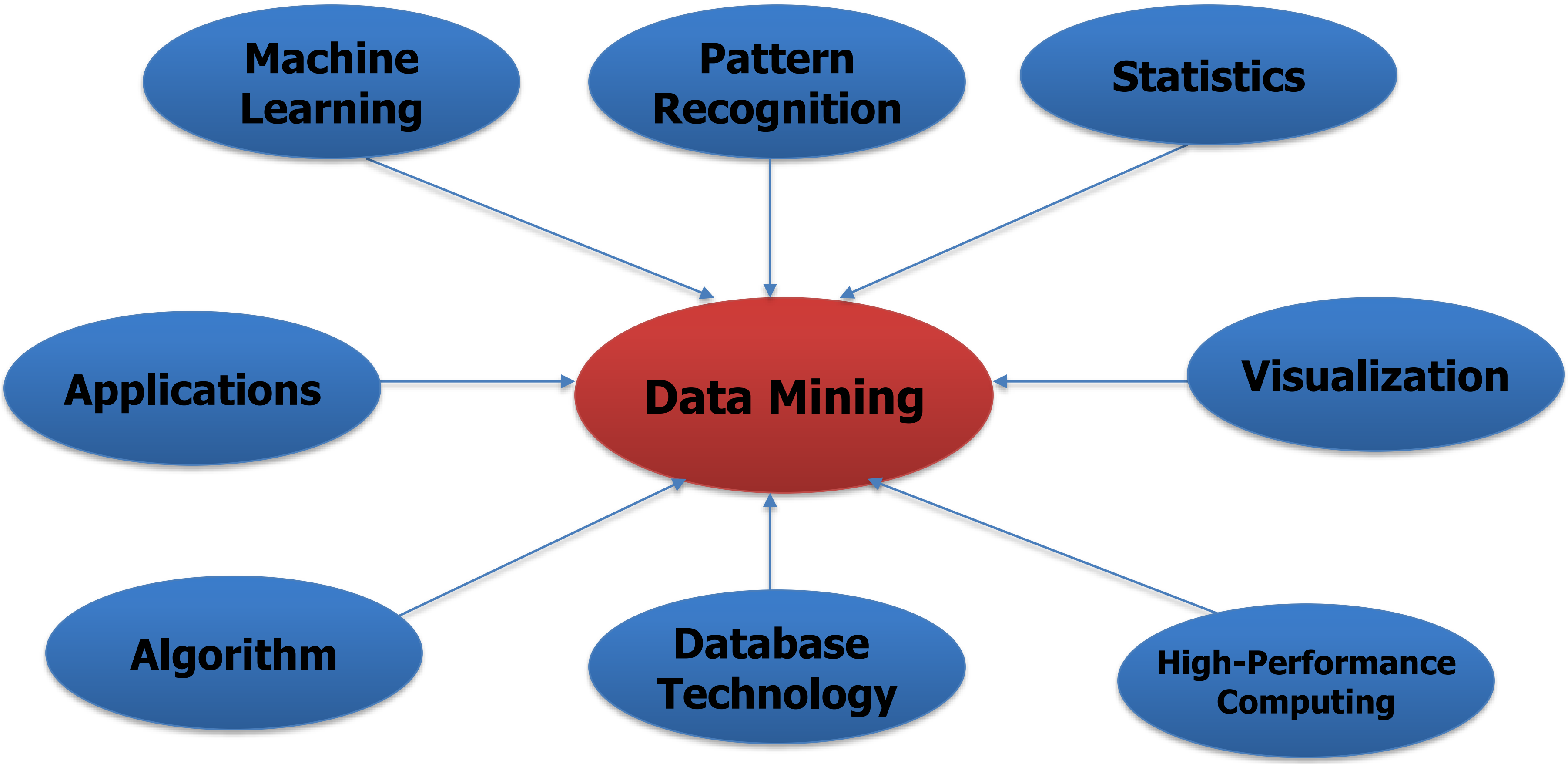
- Classification
  - Construct models (functions) based on some training examples
  - Describe and distinguish classes or concepts for future prediction
    - E.g., classify countries based on (climate), or classify cars based on (gas mileage)
  - Predict some unknown class labels

- Cluster Analysis
  - Unsupervised learning (i.e., Class label is unknown)
  - Group data to form new categories (i.e., clusters), e.g., cluster houses to find distribution patterns
  - Principle: Maximizing intra-class similarity & minimizing interclass similarity

DM Component: DM Functions (3/3)

- Outlier Analysis
  - Outlier: A data object that does not comply with the general behavior of the data
  - Noise or exception? — One person's garbage could be another person's treasure
  - Useful in fraud detection, rare events analysis

DM Component: Techniques to be used



DM Component: Applications adapted

- Web Analysis
  - E.g., from web page classification, clustering to PageRank algorithms
- Collaborative analysis & recommender systems
- Basket data analysis to targeted marketing
- Biological and medical data analysis: classification, cluster analysis (microarray data analysis), biological sequence analysis, biological network analysis
- Social networks
- Journalism
- Culture & Society

Diverse Issues in DM

- Mining Methodology
  - Mining various and new kinds of knowledge
  - Mining knowledge in multi-dimensional space
  - Boosting the power of discovery in a networked environment
  - Handling noise, uncertainty, and incompleteness of data
  - Pattern evaluation and pattern- or constraint-guided mining

평가 방식 선택

- User Interaction
  - Interactive mining
  - Incorporation of background knowledge
  - Presentation and visualization of data mining results

Diverse Issues in DM (Cont'd)

- Efficiency and Scalability
  - Efficiency and scalability of data mining algorithms
  - Parallel, distributed, stream, and incremental mining methods
  - Diversity of data types
- Handling complex types of data
  - Mining dynamic, networked, and global data repositories
- Data mining and society
  - Social impacts of data mining
  - Privacy-preserving data mining

Practical Tips for Starting Data Mining

Let’s Start with an Example

- Let’s assume that

  You are interested in “a specific problem”

  - “중환자실의 산소호흡기에서 이상 현상이 발생하는 문제를 해결하고 싶다”

  You are able to collect and analyze data (I will explain data collection in detail later)

- Let’s think the process first!
  - What is the exact problem?
  - What data do we need? Is the data available?
  - How can we represent the data?
  - What DM functions/techniques will you use?

1st step: Problem Definition

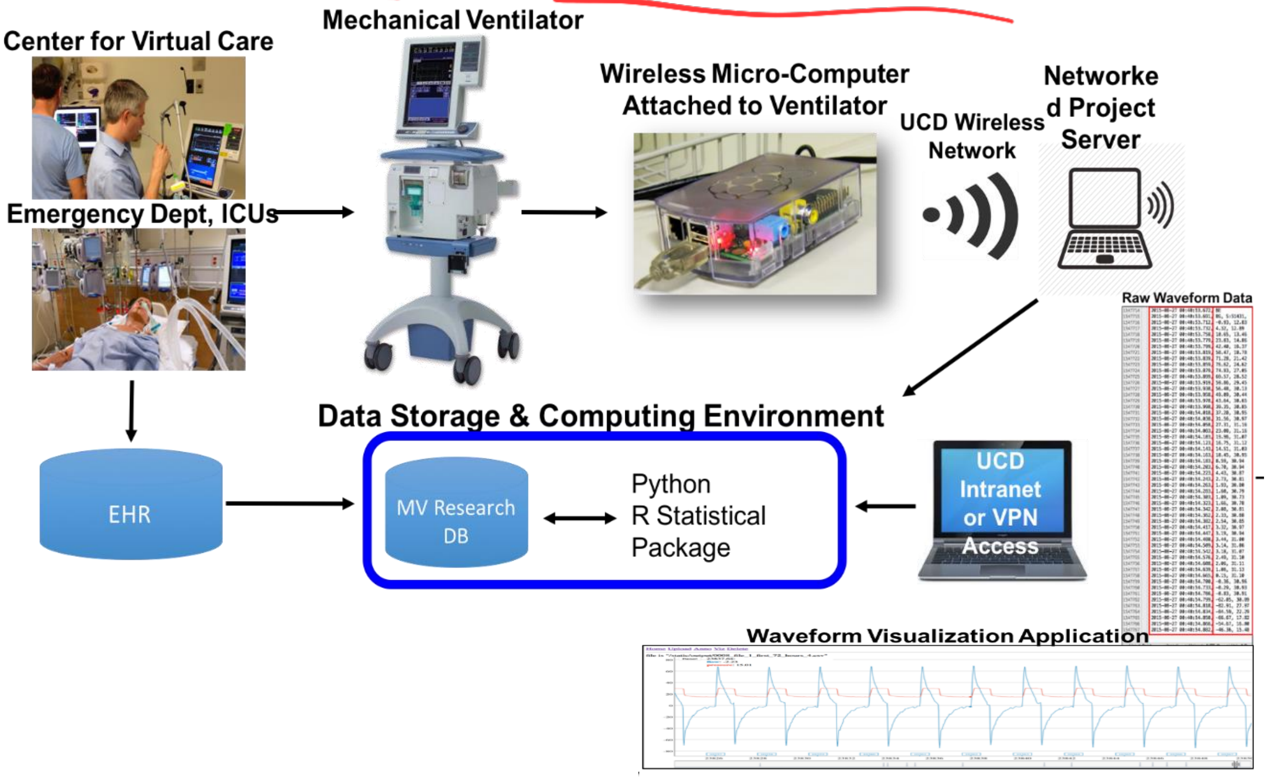
- Define the problem as a scientific form
  - From a rough, uncertain, and undetailed problem to a clear and specific problem
  - E.g., ~~"중환자실의 산소호흡기에서 이상현상이 발생하는 문제를 해결하고 싶다~~" → "**중환자실 산소호흡기에서 발생하는 breath data중에서 normal과 abnormal한 breath를 분류한다**."
  - Normal vs. abnormal의 구별 방법은?
  - …

- Require **domain knowledge** as well as **scientific problem solving capability**

2nd Step: Data Collection

- Collecting data via
  - E.g., devices collection, web crawling, …

- 중환자실 산소호흡기 breath raw data 수집



3rd Step: Data Representation

- Transforming a high-dimensional raw data into **problem-relevant** data
  - DM 기술에 따라 raw data를 그대로 이용하는 경우도 있음 (e.g., Deep learning)

- E.g., breath를 representation 하는 data 생성
  - MV metadata features
  - Waveform-generated features

- **Quantifying** data with various techniques
  - number, number, number! 꼭 숫자로 나와야 됨
  - Based on information theory, algorithm, and domain knowledge

Applying DM Functions/Techniques

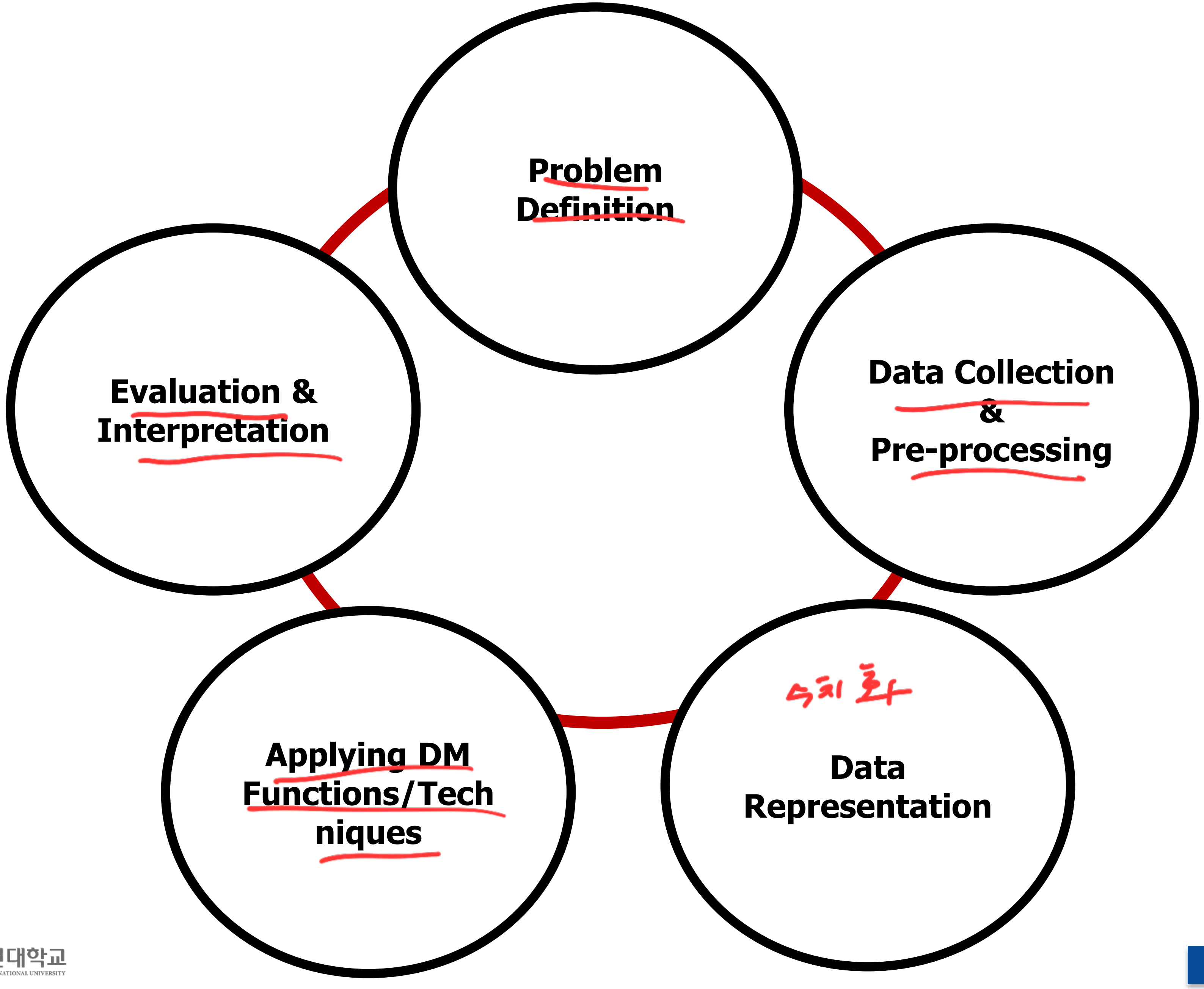
- Applying appropriate DM techniques
  - E.g., machine learning, statistical analysis, time-series analysis
  - Multiple techniques can be applied together

양적

- Quantitative vs. Qualitative
  - Complementary!

질적

Summary



Problem Definition

Data Collection & Pre-processing

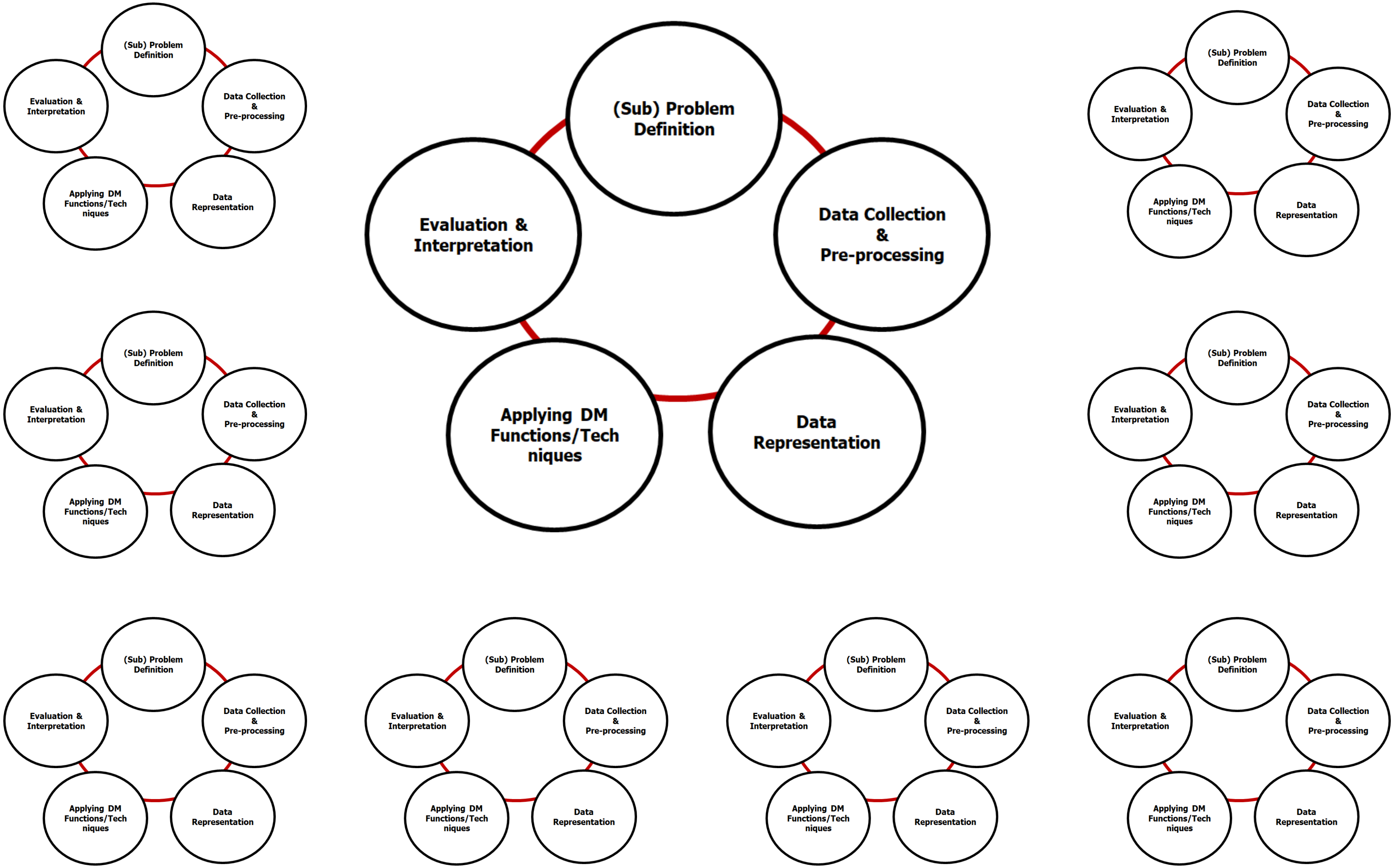
수치화

Data Representation

Applying DM Functions/Techniques

Evaluation & Interpretation

Is One Cycle Enough? NO



(Sub) Problem Definition
Data Collection & Pre-processing
Data Representation
Applying DM Functions/Techniques
Evaluation & Interpretation

(Sub) Problem Definition
Data Collection & Pre-processing
Data Representation
Applying DM Functions/Techniques
Evaluation & Interpretation

(Sub) Problem Definition
Data Collection & Pre-processing
Data Representation
Applying DM Functions/Techniques
Evaluation & Interpretation

(Sub) Problem Definition
Data Collection & Pre-processing
Data Representation
Applying DM Functions/Techniques
Evaluation & Interpretation

(Sub) Problem Definition
Data Collection & Pre-processing
Data Representation
Applying DM Functions/Techniques
Evaluation & Interpretation

(Sub) Problem Definition
Data Collection & Pre-processing
Data Representation
Applying DM Functions/Techniques
Evaluation & Interpretation

(Sub) Problem Definition
Data Collection & Pre-processing
Data Representation
Applying DM Functions/Techniques
Evaluation & Interpretation

(Sub) Problem Definition
Data Collection & Pre-processing
Data Representation
Applying DM Functions/Techniques
Evaluation & Interpretation

(Sub) Problem Definition
Data Collection & Pre-processing
Data Representation
Applying DM Functions/Techniques
Evaluation & Interpretation

After conducting hundreds of cycles, then we can conclude “a claim”

Data Collection & Preprocessing

REMIND: an example in the last section

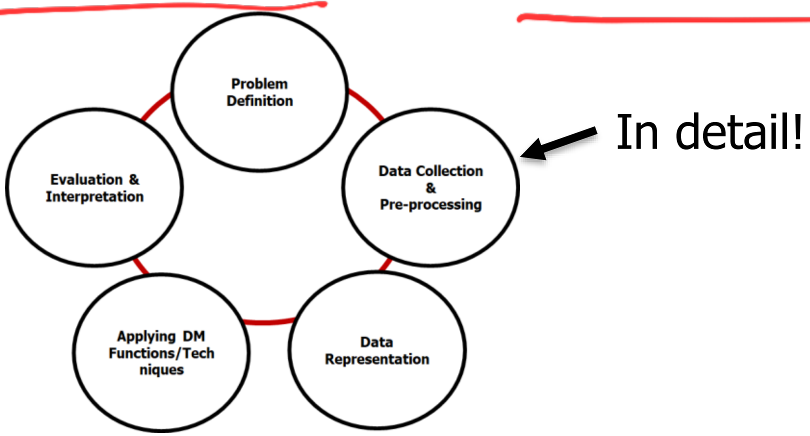
- Let's assume that

  You are interested in "a specific problem"

  - "중환자실의 산소호흡기에서 이상 현상이 발생하는 문제를 해결하고 싶다"

  You are able to collect and analyze data (I will explain data collection in detail later)

What do you have to consider for data collection & preprocessing?


Problem Definition
Data Collection & Pre-processing
Data Representation
Applying DM Functions/Techniques
Evaluation & Interpretation
In detail!

Considerations on Data Collection

- **Does the dataset exist (or can we build the dataset)?**
  - If **NOT**, the DM problem you defined **cannot be conducted**

- **If exists, specify the target data source with considering**
  - Availability (i.e., permission)
  - Access type (API, download, …) → Need to find "limits or warnings"

  (Try to spend much time on searching the previously collected datasets)

Considerations on Data Collection (Cont'd)

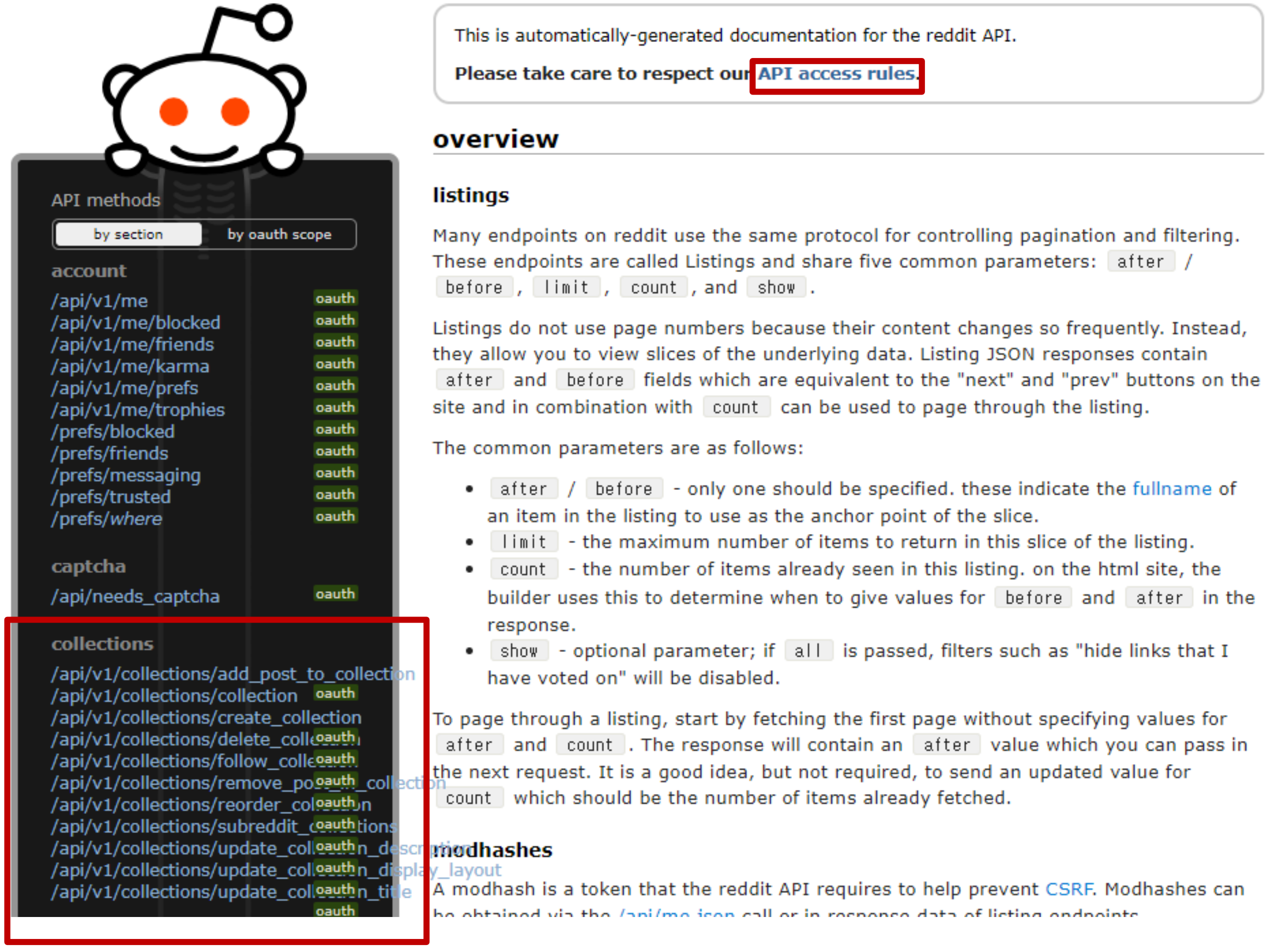
- **Decide the scale of the dataset & what to store**
  - Reasonable amount to solve your problem
  - Storage capability

- **Decide the collection methodology**
  - Crawling web pages or using APIs
  - Multiprocessing vs. single processing

- **TIP: Collect data as much as you can store**
  - Re-collection is too much expensive
  - Nobody (even you) knows what to use
  - During analysis, you may need new data that you ignored

Case Study: Reddit Crawling

- Official API is provided through https://www.reddit.com/dev/api/



API methods

by section | by oauth scope

account

/api/v1/me oauth
/api/v1/me/blocked oauth
/api/v1/me/friends oauth
/api/v1/me/karma oauth
/api/v1/me/prefs oauth
/api/v1/me/trophies oauth
/prefs/blocked oauth
/prefs/friends oauth
/prefs/messaging oauth
/prefs/trusted oauth
/prefs/*where* oauth

captcha

/api/needs_captcha oauth

collections

/api/v1/collections/add_post_to_collection
/api/v1/collections/collection oauth
/api/v1/collections/create_collection
/api/v1/collections/delete_collection oauth
/api/v1/collections/follow_collection oauth
/api/v1/collections/remove_post_in_collection oauth
/api/v1/collections/reorder_collection oauth
/api/v1/collections/subreddit_collections oauth
/api/v1/collections/update_collection_description oauth
/api/v1/collections/update_collection_display_layout oauth
/api/v1/collections/update_collection_title oauth
oauth

This is automatically-generated documentation for the reddit API.

Please take care to respect our API access rules.

overview

listings

Many endpoints on reddit use the same protocol for controlling pagination and filtering. These endpoints are called Listings and share five common parameters: `after` / `before`, `limit`, `count`, and `show`.

Listings do not use page numbers because their content changes so frequently. Instead, they allow you to view slices of the underlying data. Listing JSON responses contain `after` and `before` fields which are equivalent to the "next" and "prev" buttons on the site and in combination with `count` can be used to page through the listing.

The common parameters are as follows:

- `after` / `before` - only one should be specified. these indicate the fullname of an item in the listing to use as the anchor point of the slice.
- `limit` - the maximum number of items to return in this slice of the listing.
- `count` - the number of items already seen in this listing. on the html site, the builder uses this to determine when to give values for `before` and `after` in the response.
- `show` - optional parameter; if `all` is passed, filters such as "hide links that I have voted on" will be disabled.

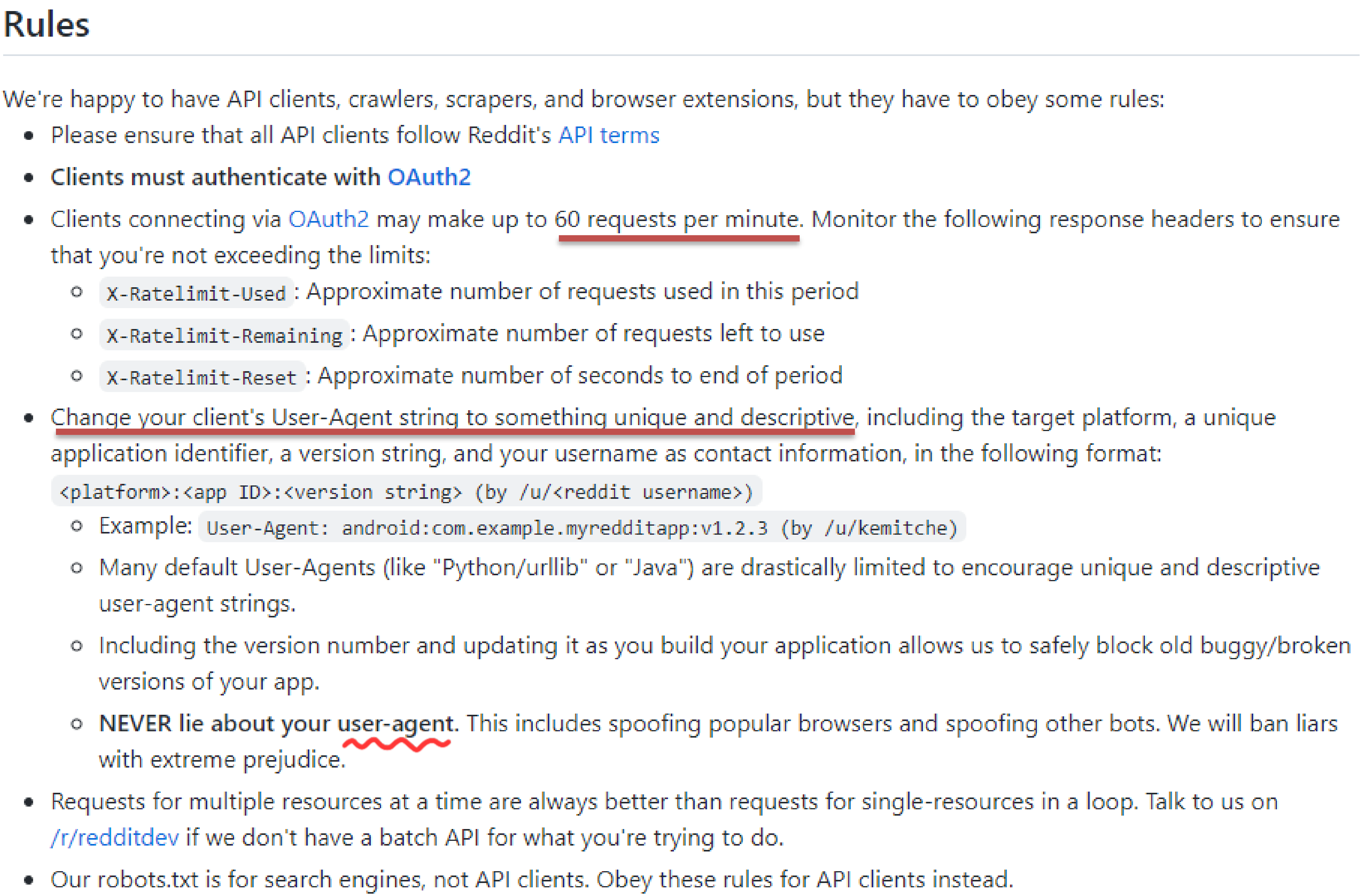
To page through a listing, start by fetching the first page without specifying values for `after` and `count`. The response will contain an `after` value which you can pass in the next request. It is a good idea, but not required, to send an updated value for `count` which should be the number of items already fetched.

modhashes

A modhash is a token that the reddit API requires to help prevent CSRF. Modhashes can be obtained via the /api/me.json call or in response data of listing endpoints.

NOTE: it's written that only last 1000 items (i.e., comments) are accessible

Investigation on Rules



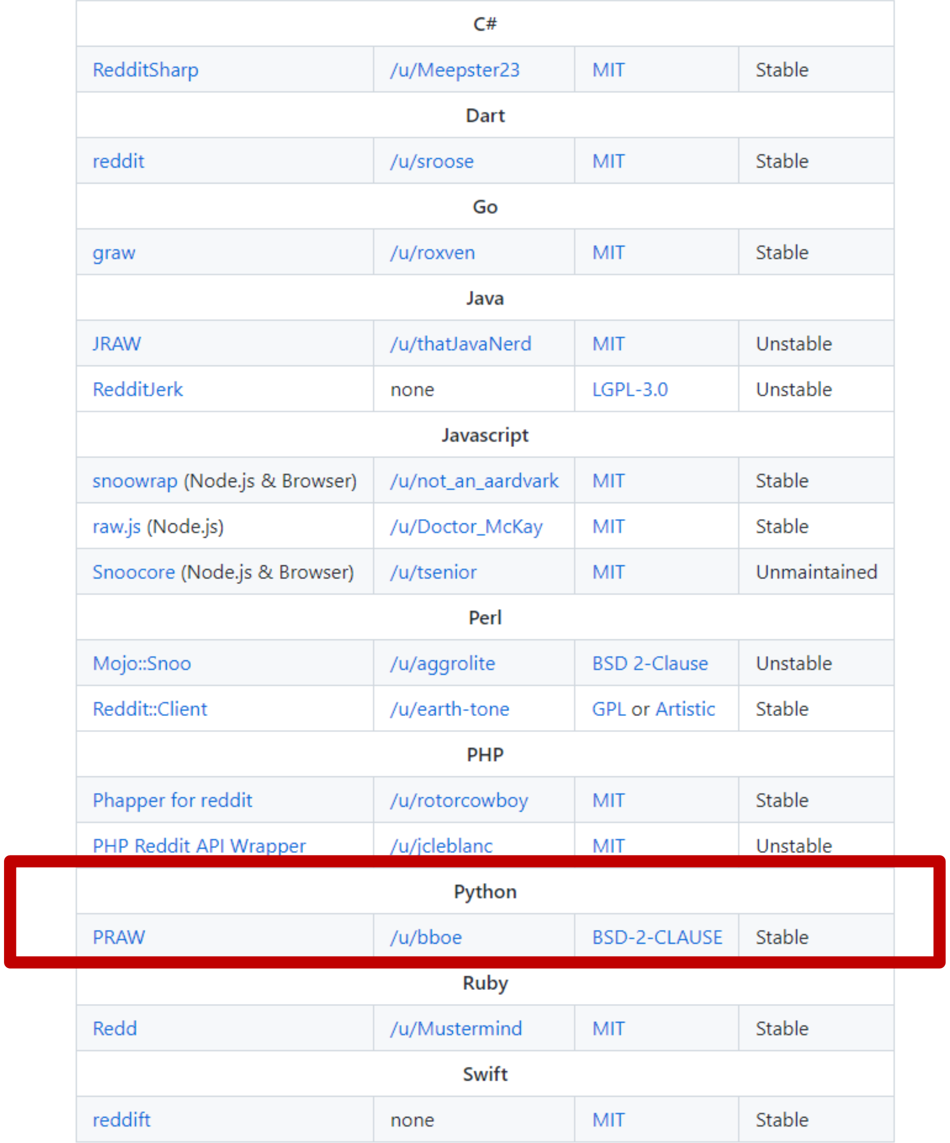
Rules

We're happy to have API clients, crawlers, scrapers, and browser extensions, but they have to obey some rules:

- Please ensure that all API clients follow Reddit's API terms
- **Clients must authenticate with OAuth2**
- Clients connecting via OAuth2 may make up to 60 requests per minute. Monitor the following response headers to ensure that you're not exceeding the limits:
  - `X-Ratelimit-Used`: Approximate number of requests used in this period
  - `X-Ratelimit-Remaining`: Approximate number of requests left to use
  - `X-Ratelimit-Reset`: Approximate number of seconds to end of period
- Change your client's User-Agent string to something unique and descriptive, including the target platform, a unique application identifier, a version string, and your username as contact information, in the following format:
  `<platform>:<app ID>:<version string> (by /u/<reddit username>)`
  - Example: `User-Agent: android:com.example.myredditapp:v1.2.3 (by /u/kemitche)`
  - Many default User-Agents (like "Python/urllib" or "Java") are drastically limited to encourage unique and descriptive user-agent strings.
  - Including the version number and updating it as you build your application allows us to safely block old buggy/broken versions of your app.
  - **NEVER lie about your user-agent**. This includes spoofing popular browsers and spoofing other bots. We will ban liars with extreme prejudice.
- Requests for multiple resources at a time are always better than requests for single-resources in a loop. Talk to us on /r/redditdev if we don't have a batch API for what you're trying to do.
- Our robots.txt is for search engines, not API clients. Obey these rules for API clients instead.

Find OPEN SOURCE (API Wrapper)

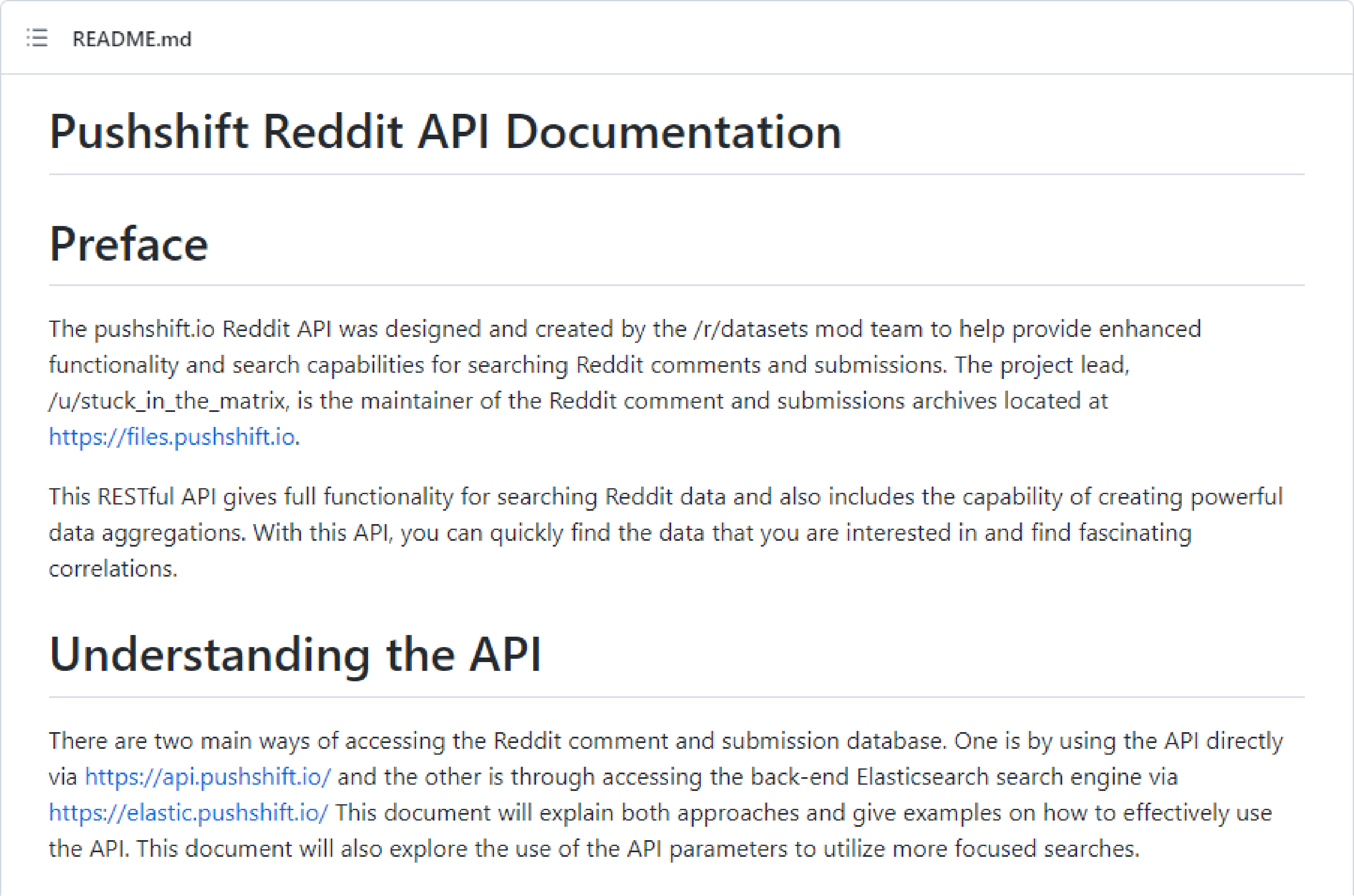
- Don’t start from scratch



C#			
RedditSharp	/u/Meepster23	MIT	Stable
Dart			
reddit	/u/sroose	MIT	Stable
Go			
graw	/u/roxven	MIT	Stable
Java			
JRAW	/u/thatJavaNerd	MIT	Unstable
RedditJerk	none	LGPL-3.0	Unstable
Javascript			
snoowrap (Node.js & Browser)	/u/not_an_aardvark	MIT	Stable
raw.js (Node.js)	/u/Doctor_McKay	MIT	Stable
Snoocore (Node.js & Browser)	/u/tsenior	MIT	Unmaintained
Perl			
Mojo::Snoo	/u/aggrolite	BSD 2-Clause	Unstable
Reddit::Client	/u/earth-tone	GPL or Artistic	Stable
PHP			
Phapper for reddit	/u/rotorcowboy	MIT	Stable
PHP Reddit API Wrapper	/u/jcleblanc	MIT	Unstable
Python			
PRAW	/u/bboe	BSD-2-CLAUSE	Stable
Ruby			
Redd	/u/Mustermind	MIT	Stable
Swift			
reddift	none	MIT	Stable

Pushshift for more availability

- https://github.com/pushshift/api



README.md

Pushshift Reddit API Documentation

Preface

The pushshift.io Reddit API was designed and created by the /r/datasets mod team to help provide enhanced functionality and search capabilities for searching Reddit comments and submissions. The project lead, /u/stuck_in_the_matrix, is the maintainer of the Reddit comment and submissions archives located at https://files.pushshift.io.

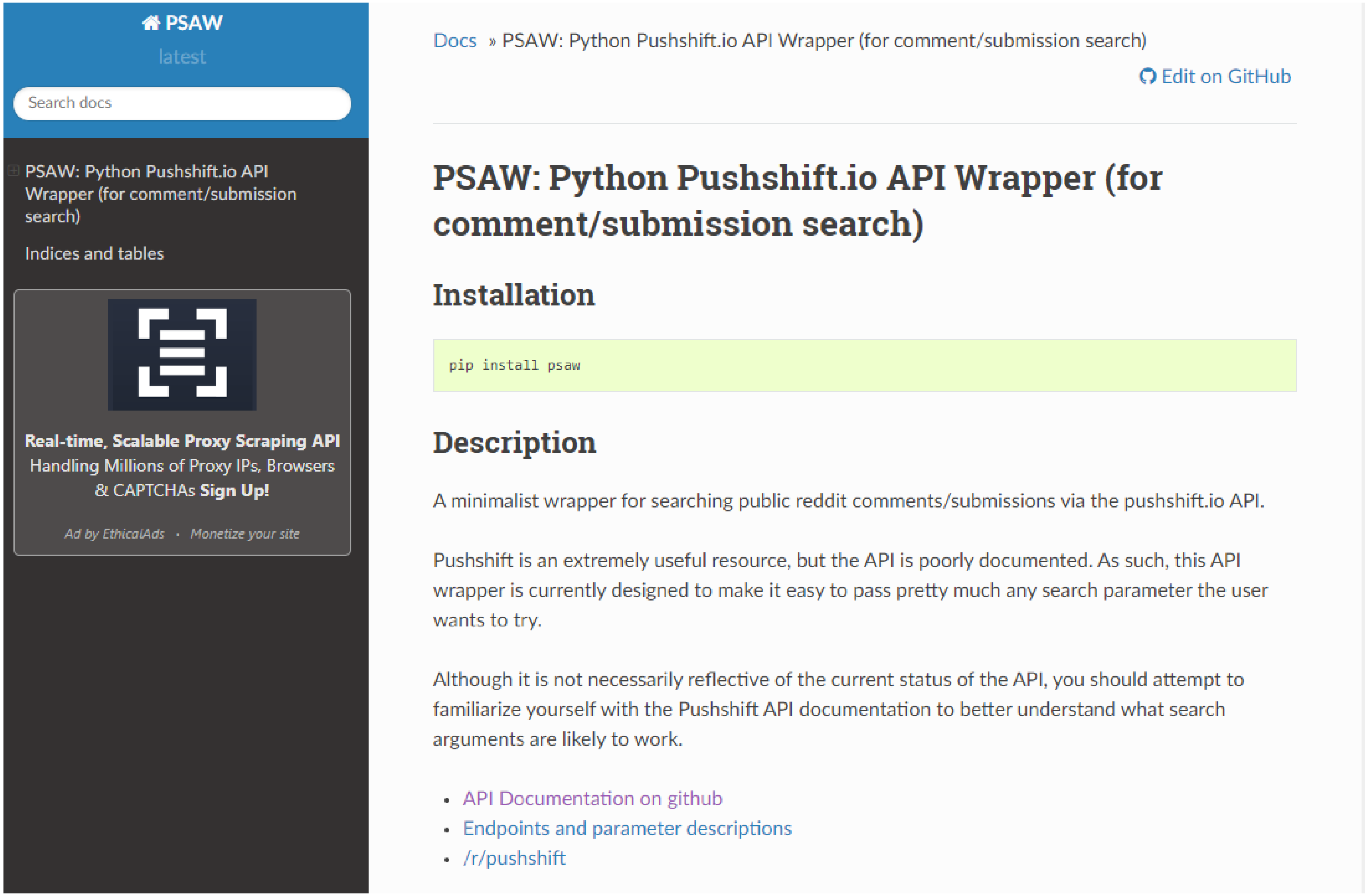
This RESTful API gives full functionality for searching Reddit data and also includes the capability of creating powerful data aggregations. With this API, you can quickly find the data that you are interested in and find fascinating correlations.

Understanding the API

There are two main ways of accessing the Reddit comment and submission database. One is by using the API directly via https://api.pushshift.io/ and the other is through accessing the back-end Elasticsearch search engine via https://elastic.pushshift.io/ This document will explain both approaches and give examples on how to effectively use the API. This document will also explore the use of the API parameters to utilize more focused searches.

Again, Check the existence of OPEN SOURCE

- https://psaw.readthedocs.io/en/latest/
- Also again, please check the API limits



PSAW
latest
Search docs
PSAW: Python Pushshift.io API Wrapper (for comment/submission search)
Indices and tables

Docs » PSAW: Python Pushshift.io API Wrapper (for comment/submission search)
Edit on GitHub

PSAW: Python Pushshift.io API Wrapper (for comment/submission search)

Installation

```
pip install psaw
```

Description

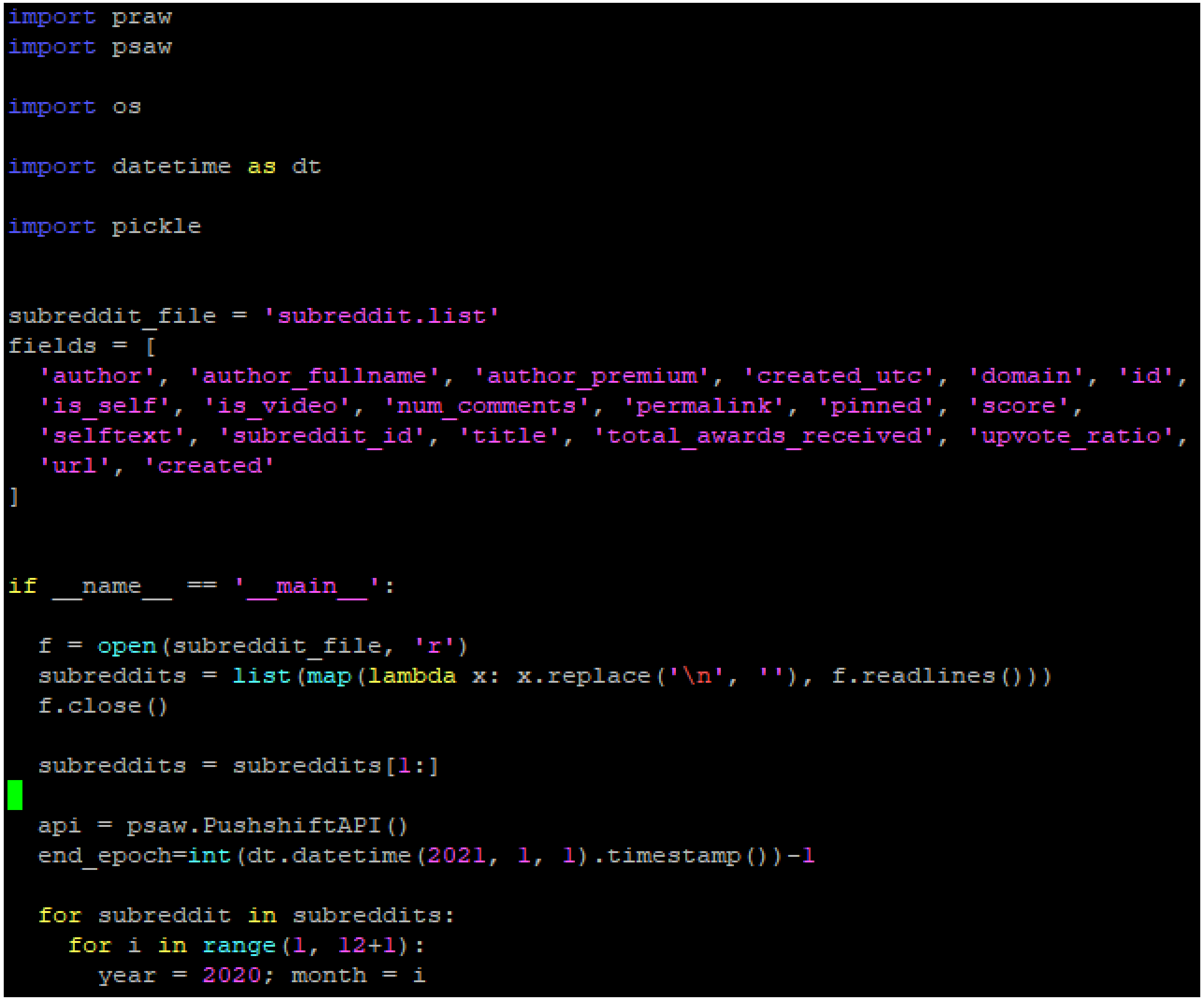
A minimalist wrapper for searching public reddit comments/submissions via the pushshift.io API.

Pushshift is an extremely useful resource, but the API is poorly documented. As such, this API wrapper is currently designed to make it easy to pass pretty much any search parameter the user wants to try.

Although it is not necessarily reflective of the current status of the API, you should attempt to familiarize yourself with the Pushshift API documentation to better understand what search arguments are likely to work.

- API Documentation on github
- Endpoints and parameter descriptions
- /r/pushshift

Then, Implementing Crawler!



```
import praw
import psaw

import os

import datetime as dt

import pickle


subreddit_file = 'subreddit.list'
fields = [
  'author', 'author_fullname', 'author_premium', 'created_utc', 'domain', 'id',
  'is_self', 'is_video', 'num_comments', 'permalink', 'pinned', 'score',
  'selftext', 'subreddit_id', 'title', 'total_awards_received', 'upvote_ratio',
  'url', 'created'
]


if __name__ == '__main__':

  f = open(subreddit_file, 'r')
  subreddits = list(map(lambda x: x.replace('\n', ''), f.readlines()))
  f.close()

  subreddits = subreddits[1:]

  api = psaw.PushshiftAPI()
  end_epoch=int(dt.datetime(2021, 1, 1).timestamp())-1

  for subreddit in subreddits:
    for i in range(1, 12+1):
      year = 2020; month = i
```

About Term Project

Proposal 내용

- Crawling Target?
  - Online Communities: Twitter, Reddit, Instagram, Naver, Inven, DC Inside
  - Media: YouTube, Naver 뉴스, …
  - Specific area: 주식, Game 통계, LOL, 기상청, Google Trends, YouTube Trends, 영화관련 통계, …

- 분석주제들?
  - Covid19 관련 분석 (밀집지역 파악, 정부지지도, 거리두기 효과성, 백신기업주가, 유동인구, 소비패턴, …)
  - Covid19 전후 변화 (구매상품, 학사운영비용, 영화예매율, 관광산업, 책 구매량, 어휘력, …)
  - 주식, 비트코인 관련 분석 (커뮤니티 반응 vs. 주식 가격 등)
  - 게임관련 분석 (playtime vs. 실력/평가도)
  - 기타 전공관련 분석 (항생제 효과, …)

My Impression

- 여러분은 하고싶은 것이 참 많고, 모티베이션도 대부분 훌륭하다
  - Covid-19, 돈, 게임, …

- 하지만, 데이터 수집 관련하여 다음 질문들에 대해서 구체화가 더 필요함
  - 데이터 수집의 target이 불명확한 경우 (e.g., SNS)
  - 데이터 수집 가능 여부에 대한 자료조사가 불명확한 경우 (e.g., Instagram, YouTube API 등)
  - 데이터 수집계획이 불명확한 경우 (e.g., 인스타그램에서 진성팔로워 추출, 네이버 까페에서 드라마 관련 반응 수집)

My Impression (cont'd)

- 유사하게, 분석 계획에 대해서도 모호한 경우가 보임
  - E.g.1, 인기있는 게임들의 공통점 분석
    - 어떤것이 인기 있는 게임이라고 정의될 수 있는지, 어떤 공통점이 있을 것이라고 가설을 세울 수 있는지, …
  - E.g.2, 게임 승패 여부에 영향을 미치는 요인 분석
    - 요인의 가짓수가 엄청나게 많음. 따라서 몇가지 요인을 가설로 만들어 검증하는 것이 더 좋음

- (Advanced) Project Topic의 "의미" or "가치" 에 대한 고찰이 덜 되어있음.

No worries. That means you will learn
“how to design a topic on data mining”

Feedback: Overall

- "Availability"에 대한 조사/고민을 구체적으로 수행할 것
  - 주제/가설 설정 → Availability에 대한 기초조사 수행 → 연구계획 수립 → proposal ~~제출~~ update
- 주제 / 분석계획을 더 명확하게 할 것
  - SNS → Twitter or Facebook
  - 유저의 반응 분석 → 유저의 긍/부정 반응 정도 차이 분석
  - 각종 커뮤니티 분석 → Reddit? DC Inside?
- (Advanced) 해당 데이터 분석 결과를 다른 사람들이 어떻게 활용할 수 있는지 "설득"해보기

REMIND: Tips on Analysis

- What과 How를 동시에 생각
  - 반응분석 → 어떤반응을 분석? → 어떻게?

- 가설을 구체화하는 것이 도움이 될 수 있음
  - 대중들의 긍정적 반응과 주식은 positive correlation이 있을 것이다

- Causality is NOT equal to Correlation
  - Causality usually contains "temporal" factor
  - Community의 반응 ←→ 주식상승/하락

Your Next Step

- Preparing Intermediate Project Presentation & Report
  - The presenters will be selected

- Notes
  - 개별/전체 Feedback을 최대한 많이 반영할 것
  - 주제를 좀 더 명확하게 (ambitious하게 하지 않아도 됨)
  - 가능성 여부를 반드시 검토해볼 것, 다만 부담을 많이 가지지 말 것
  - 최대한 알아보고, 확신이 서지 않는다면 도움을 많이 요청할 것
  - Proposal 이 끝이 아님. 반드시 중간 progress가 있어야 함

Thank you!

Instructor: Daejin Choi (djchoi@inu.ac.kr)

Good Example 1 – Pandemic & Emotion Dynamics

- 팬데믹 진행과정에 따른 사람들의 감정 차이 분석

- 수집 Target data
  - Twitter, 뉴스, 네이버 트렌드, 다음블로그

- 전략: Keyword searching
  - 코로나, Covid, Covid-19, 우한, 신천지, 이태원, 마스크 5부제, …

- 팬데믹 진행과정을 이벤트 단위로 분할
  - 신천지 관련 이슈 발생
  - 이태원 관련 이슈 발생
  - 마스크 5부제 시행
  - 사회적 거리두기 2.5단계

- 각 분할된 시점에 감정의 정도차이가 어떻게 발생하는지 koBERT를 활용하여 분석

주제 및 분석 전략이 명확, 사전 조사가 잘 되어 있음.

Good Example 2: AI에 대한 여론분석

- Artificial Intelligence 키워드에 대해서 시간의 변화에 따라 긍/부정 등의 반응이 어떻게 변해가는지 분석

- 수집계획
  - Target: Naver, Yahoo Japan, Reddit
  - AI keyword로 댓글 검색 및 수집

주제 및 분석 전략이 명확

Good Example 3: Stack Overflow 의 유행 Topic분석

- Stack overflow에 올라오는 질문 수집 후, 일별/월별 등 시간의 구분에 따라 keyword기반 topic을 추출하여 topic의 dynamics를 분석

주제 및 분석 전략이 명확

주제의 가치 및 의미를 구체적으로 서술
(유행 기술의 흐름 분석에 용이 등)