

# 그래프 DB를 이용한 유사 판례 클러스터링

김성현

알파로우 2018

# Overview

---

1. 현재상황
2. 목적
3. 가설
4. 구현
5. 평가

# Overview

---

## 1. 현재상황

- a. 재석님의 은총을 받아 판례DB는 공유되었는데 딱히 할 게 없다(?)
- b. 판례를 분류를 해보려고 해도, 공개된 판례의 정황과 고유한 특성 때문에 분류하기 어렵다(?)
- c. 판례 내용의 전처리가 되어 있지 않다(?)
- d. 게으름(!)

## 2. 목적

## 3. 가설

## 4. 구현

## 5. 평가

# Overview

---

1. 현재상황
2. 목적
  - a. “판례를 좀 신박하게 카테고리를 묶어서 조회해보고 싶다”
  - b. 재운님의 논문 방향성과 Co-Work 가능성
    - i. 김재윤. “대한민국 법률의 인용 네트워크 구조 분석”(석사학위, 서울대학교 대학원. 2015.)
    - ii. 김재윤. “법률의 총량 및 복잡성 증대 추세 분석” 『입법과 정책』 제9권 제1호(국회입법조사처, 2017.4.)
  - c. 기술적 화두 : 온라인 비정형 데이터, NoSQL 및 그래프 데이터베이스
3. 가설
4. 구현
5. 평가

# Overview

---

1. 현재상황
2. 목적
- 3. 가설**
  - a. 인용하고 있는 법률의 조합으로 조희했을 경우, 어느 정도 군집이 형성됨.**
  - b. 자주 인용되는 법률은 제한적임.**
4. 구현
5. 평가

# Overview

---

1. 현재상황
2. 목적
3. 가설
4. 구현
  - a. 판례번호 + 판례와 법률의 인용관계
    - i. SQL : `SELECT casenumber FROM openlawdb.casetbl;`
    - ii. SQL : `SELECT referenceprovision FROM openlawdb.casetbl;`
    - iii. [대한민국 법원 사건구분안내](#)
  - b. Python, Azure COSMOS DB(Graph DB), Amazon RDS(MySQL)
    - i. [rds-insert-case+reflaws.ipynb](#)
    - ii. [cosmosdb-gremlin-insert-case+reflaw.ipynb](#)
  - c. [Gremlin Query](#)
5. 평가

# Overview

---

1. 현재상황
2. 목적
3. 가설
4. 구현
5. 평가
  - a. 법률명 표현의 다양한 형태, 정제한 기준 법률명의 한계
    - i. 공개 법령 API에서 법령ID 값을 기준으로 하면 좋을 것으로 예상
  - b. 기술적인 접근성이 낮지 않은 느낌
  - c. 클러스터링이라고 하기엔 낮은 수준의 그래핀 쿼리
    - i. 또 익혀야 하는 것일까
  - d. 난해한 시각화
    - i. Neo4j에 비해 시각화 라이브러리 및 레퍼런스 양이 적은 것으로 보임.

감사합니다.