**[OOP 10조 13주차 보고서]**

**작성자: 김찬영**

1. **13주차 요약 정리**
   1. **금요일 개발 회의**- 11월 29일 오후 1시부터 저녁 11시까지 약 10시간 정도 개발 회의를 진행하였다. 대시보드 개발에 있어서, 전체 프로젝트 진행 방향 및 할 일에 대해서 헷갈린 부분이 있어서 조정 후에 할 일을 진행하였다. 현재 진행 완료된 부분에 대해서 O 표시를 하였다.
      1. **모든 과정의 단계 별로 문서화하고, 문서화에 있어서 적어야 할 부분**
         1. 하나의 과정에 목표 및 해야하는 의의
         2. 과정에서 사용하는 전처리 단계 및 데이터
         3. 과정에서 사용하는 라이브러리, 알고리즘, 특정 함수 그리고 그들의 특성
         4. 코드에서 중요한 부분
         5. 발생한 문제점들과 해결한 방향
         6. 개선하기 위한 조치 및 방향
      2. **알고리즘 개발 목록**
         1. PDF를 텍스트로 변환하는 과정
            * 논문에서 빼야할 부분
            * 두개의 단으로 나뉘어져 있는 경우 문제
            * 한글이나 특수 기호
            * stopword
         2. 각각의 텍스트 마이닝 알고리즘 적용  
            데이터의 경우, 컴공과 논문 5개, 산경과 논문 5개 총 10개로 진행한다.
            * 전체 논문에 대한 키워드 추출\*(전체 논문의 경우 전체 논문을 상대로 진행한다)\*

Automated Keyword Extraction Using NLP

* + - * + 개별 논문에 대한 키워드 추출

Simple statistics approaches

Word Frequency (X)

Word Collocations and Co-occurrences (O)

TF-IDF (O)

RAKE (O)

Machine learning approaches

Conditional Random Fields(CRF)

Deep Learning

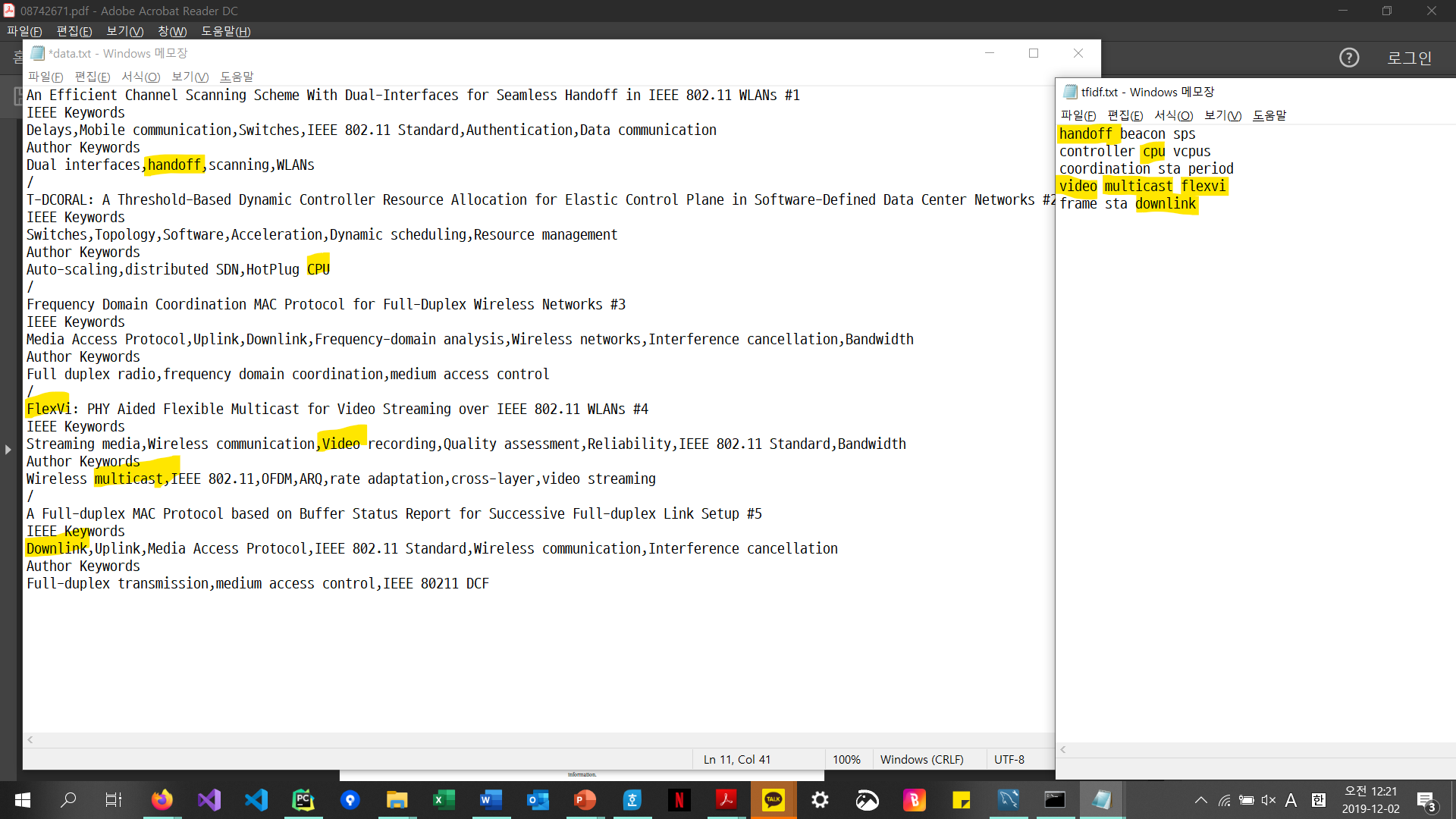
* + 1. **백 프론트 개발 목록**
       1. 대시보드 개발에 필요한 오픈소스 찾기(O)
       2. 오픈소스 코드 해석 및 실행(O)
       3. 백 엔드
          - DataBase — 서버 연결(API 형태로 기능 확인)(O)
          - API 형태로 논문 분석 데이터 제공(학교, 학과, 연구실 별)(O)
          - 세션 및 쿠키 관리(O)
          - 로그인 모듈 개발
       4. 프론트 엔드
          - 프론트 페이지 설계 및 기능 구현

단위별 페이지 구성 요소 설계

학과별 연구실 별 체크 항목 및 페이지 구성

* + - 1. DataBase
         * DataBase 설계
         * 자동 데이터 추가

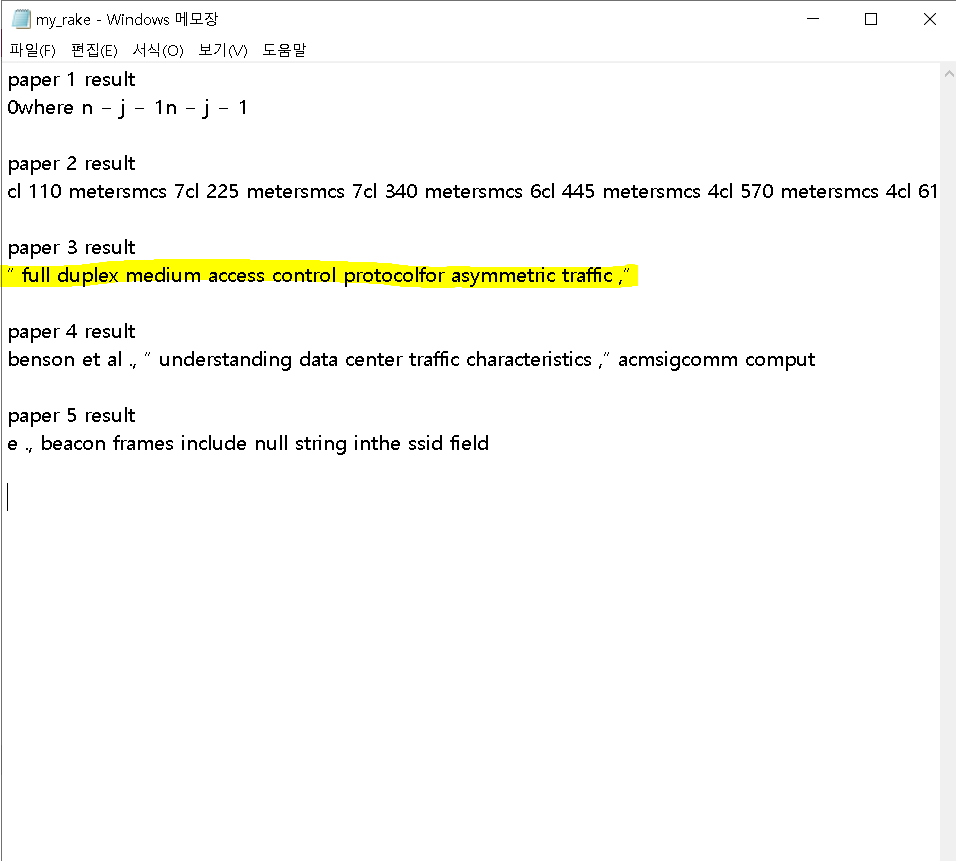
Python을 활용한 데이터 추가 알고리즘

* 1. **일요일 개발 회의**- 일요일 오후 8시부터 5시간 정도 개발을 진행하였다.
     1. **알고리즘 개발**
        1. TF-IDF  
           
           + Exp 조건: 컴퓨터공학과 랩 하나를 선정하여, 그 랩의 5개 논문을 설정하여 저널에서 설정한 키워드와 TF-IDF를 통해 나온 상위 3개 키워드를 비교하였다.
           + 문제점: TF-IDF는 단일 문서에서는 많이 나오고, 다른 대부분의 문서에서는 빈도가 적은 단어를 키워드로 설정한다. 그러나 IEEE에서 설정한 키워드의 경우, 이 단어들은 다른 문서들도 함께 포함될 수 있는 범주에 해당한다. 실제 논문에서는 논문에서 수행한 특정 연구에 직접적으로 관련된 단어들보다 이러한 범주에 해당하는 단어들이 훨씬 적게 나오므로 TF-IDF의 결과와 IEEE에서 설정한 단어를 비교하기에는 어려움이 있다.
           + 개선방안: 현재 조건은 한 개의 랩에서 제출한 연구 성과에 대해서만 비교를 진행하여 실질적인 TF-IDF의 적용 조건이라 보기 어렵다. 다양한 연구실의 다양한 논문들에 대하여 TF-IDF를 진행하면 개선된 결과가 나올 수도 있다. 또한, TF-IDF에서 적용되는 n-gram의 범위를 적절히 지정하면 여러 개의 명사들로 이루어진 전문용어들이 나올 확률이 더 높아질 수 있을 것으로 보인다.
        2. RAKE
           + Rapid Automatic Keyword Extraction(RAKE)는 텍스트에서 가장 관련성이 높은 구(phrase)를 추출할 수 있는 키워드 추출 방법의 일종이다.
           + RAKE이 작동하는 알고리즘은 다음과 같다.

논문을 건네주면, 단어 단위로 토큰화 하고 co-occurrence에 대한 행렬을 만든다. 행렬은 각 단어가 동시 발생하는 횟수를 기록한다.

행렬이 완성되면, score는 행렬에 기록된 co-occurrence 와 frequency를 기준으로, 부여된다.

키워드 구(phrase)와 그에 해당하는 score가 튜플 형태로 추출된다.

* + - * + **결과**각 논문마다 출력된 키워드가 너무 많아서, 스코어가 높은 상위 5개의 키워드를 살펴보았다. 또한 키워드를 추출하는 함수가 구두점을 기준으로 문장으로 쪼갠 후, 구에 대해서 score를 부여하는 방식으로 되어 있었다. 기존의 데이터 전처리에서 구두점을 모두 제거하고 토크나이징 처리를 했기에 이 둘을 하지 않고, 명사형만 추출하도록 한 뒤에 결과를 출력하도록 하였다.  
          **그 결과 논문 3의 경우는 추출된 키워드가 Author keyword와 꽤나 흡사했다. 그러나 나머지 논문들은 전혀 그렇지 않았다. 이유는 분석해내지 못했다. RAKE알고리즘은 논문 같이 아주 긴 text에 대해서는 적절하지 않은 알고리즘 인 것 같다.**
    1. **백 프론트 개발**
       - * 로그인 모듈  
           기존의 오픈 소스의 경우, Session 및 Cookie 관리는 Redis를 통해서 이루어지고, user 데이터는 PostgreSQL을 통해서 이루어지는데, 이를 모두 MySQL로 바꾸기 위한 과정을 거치게 되었다. Session 및 Cookie DB와 User DB의 migration까지 성공하였지만, 웹 페이지를 동작하는데 있어서 DB와 소통하는 과정에서 Redis와 PostgreSQL과 달리 MySQL은 Returning 기능이 없어서 우선 로그인 모듈의 경우, 오픈 소스와 별개로 새롭게 짤 계획이다.
         * 웹 페이지에 DB 데이터 띄우기

기존의 작업에서 MySQL에서 데이터를 받아오는 것까지는 성공하였으나 js파일로 가져온 데이터를 ejs 파일에 넘겨 렌더링 하는 과정을 완료하지 못했다. data라는 이름의 변수를 넘기려 했으나 data is not defined 라는 메시지가 뜸.

여기서 데이터를 넘기는 방법은 2가지 있다는 것을 알게 되었다. (get 함수와 render 함수) 둘 다 사용하려 했으나 좀더 직관적이고 쉽게 이해가 갔던 render 함수를 쓰기로 결정

앞선 과정에서 알게 된 사실을 통해 데이터베이스를 받아오는 동작은 router함수 안에서 해주어야 한다는 것을 깨닫게 되었다. 따라서 dashboard의 router 안에 query 함수를 통해 MySQL 속 모든 데이터를 results에 담는다.

그후 dashboard라는 ejs 파일에 data라는 이름으로 results를 넘기고 dashboard안에서 다른 ejs 파일을 include 할 때 data도 같이 넘겨주며 동작하게 한다.

데이터베이스에서 처음부터 데이터를 골라서 가져오면(예를 들면 year 데이터만 받는다.) 데이터 이름도 같이 저장된다. (year:2019) 따라서 모든 데이터를 받고, 쓸 때만 골라서 쓴다.