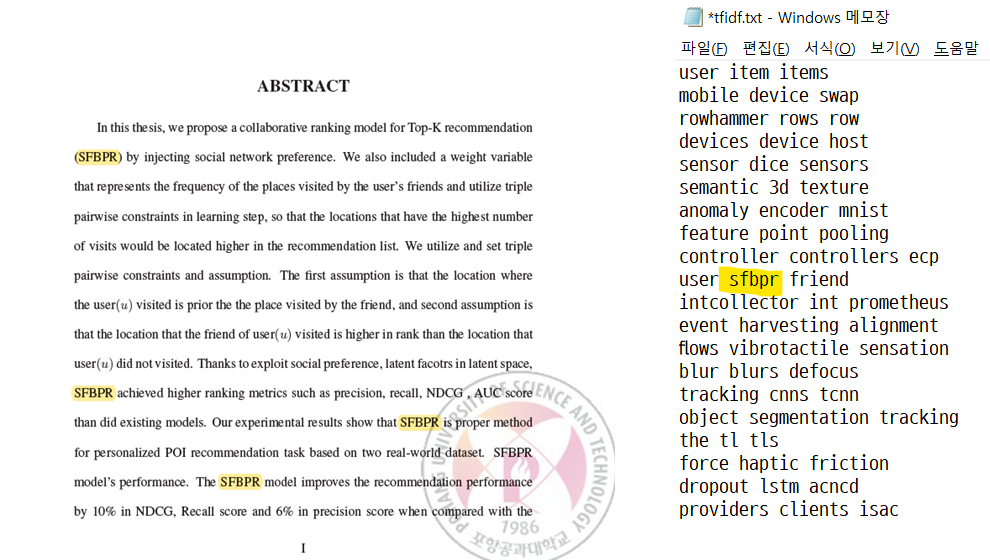
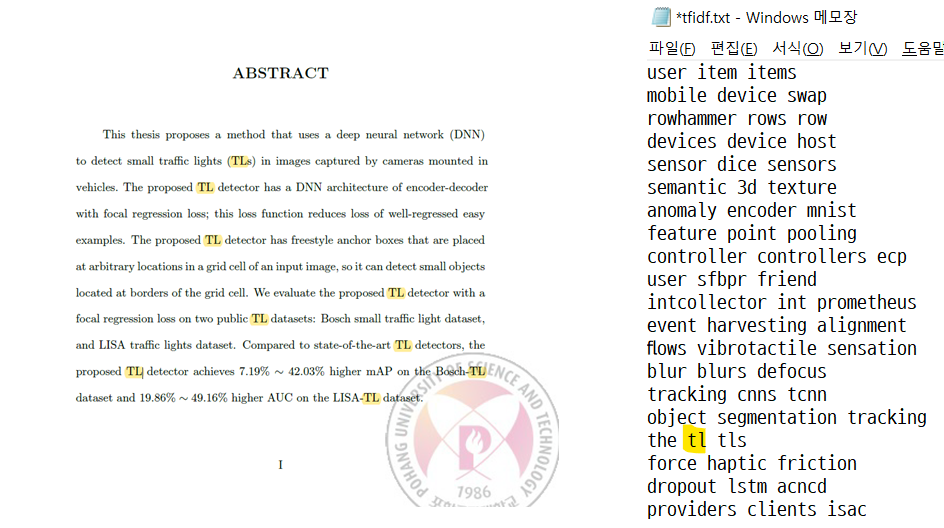
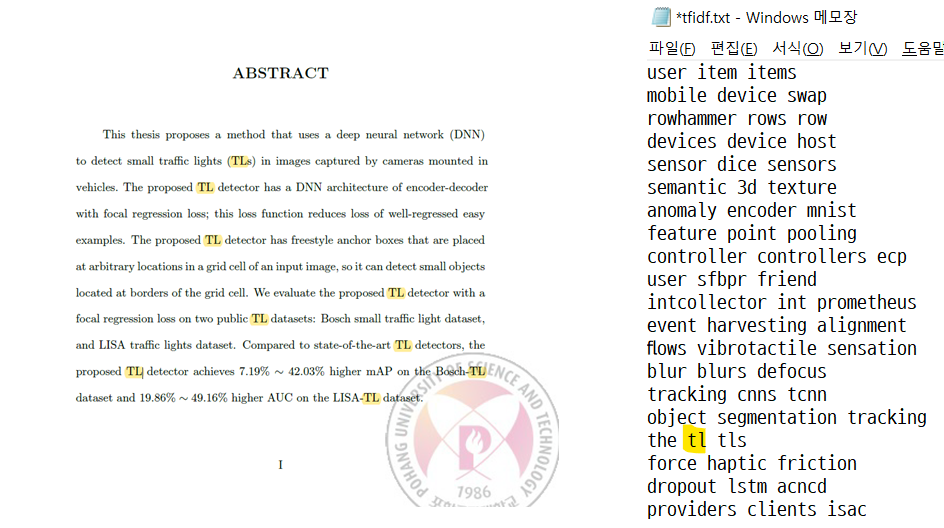
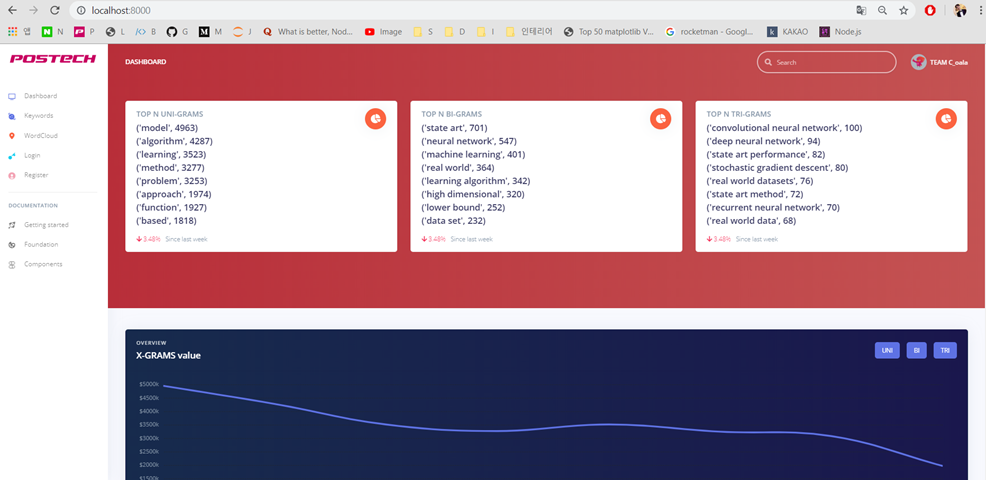
**[OOP 10조 11, 12주차 보고서]**

**작성자: 김찬영**

1. **11주차 요약 정리**
   1. **화요일 주간 회의**- 11월 12일 화요일 점심에 약 1시간 정도 회의를 진행하였다. 알고리즘 팀(상윤, 태규, 진모)과 백 프론트 개발(찬영, 서영, 경민) 팀이 서로 주말 간 했던 것들과 앞으로 진행할 작업들에 대해서 공유하는 자리를 가졌다.
      1. **알고리즘 팀**  
         PDF 형태로 되어있는 논문을 TEXT로 변환하는 작업에 대해서 공유하였다. 오픈 소스 PDFMiner를 사용하여 작업하였는데, 코드를 실행하는데 있어 각각의 단계에서 나오는 Output에 대해서 코드를 실행하면서 공유하였다. 그리고 거쳐간 오류들에 대해서 설명하였다.
      2. **백 프론트 개발 팀**  
         argon-dashboard-nodejs라는 오픈 소스를 활용하였고, 이를 활용하여 대시보드 개발을 하기 위해 오픈 소스 코드를 정리하였다. 우선 기존 오픈 소스의 경우 로그인을 통해서 메인 대시보드를 볼 수 있도록 되어 있는데, 로그인은 필요 없는 기능이라 생각되고, 제대로 작동하지 않아 우선적으로 로그인 관련 코드를 정리하였다. 그리고 서버를 실행해보고 기본적인 UI 및 기능들을 다같이 확인해보았다.
   2. **토요일 – 일요일 포스텍 해카톤 참여**- 토요일 오후 2시부터 일요일 오후 2시까지 포스텍 해카톤을 통해 개발을 진행하였다.
      1. **알고리즘 개발**  
         Text로 변환한 데이터에 대해서 키워드 추출 알고리즘을 적용해보았다. TF-IDF라는 알고리즘을 통해서 각각의 논문에서 키워드를 뽑아보았다. TF-IDF 알고리즘이란 Term frequency-inverse document frequency의 약자로, 영어 의미 그대로 단어가 출현된 수에 전체 문서들에서 얼마나 등장한 횟수의 역수를 곱해준 값을 의미한다. IDF를 곱해주는 것은 그 논문에서만 등장하는 키워드에 대해서 가중치를 주는 방식을 의미한다. 알고리즘 적용 결과, 각 논문의 abstract와 결과를 비교해보니 비교적 좋은 결과를 얻을 수 있었다. 하지만 한계점들도 많았는데, 이에 대해서 계속적으로 해결할 필요를 느꼈다. 아래 결과는 어떤 논문에서 키워드가 되는 것을 왼쪽 사진의 논문 당 세 가지 키워드에 뽑힌 목록에서 볼 수 있는 사진이다.  
           
         
      2. **백 프론트 개발**데모 페이지 시연을 위해서, 페이지 프론트 개발 작업 및 데이터베이스 구현 및 설계하였다. 또한 임시적으로 데이터를 넣어두고 이 데이터가 대시보드 화면에 나올 수 있도록 작업하였다.  
           
           
         하지만, 데이터베이스 설계와 임시 데이터 추가를 완료하였지만, 데모 페이지에 데이터를 뿌리는 것은 시간 내에 완성하지 못했다. 하지만, 서버와 데이터베이스를 연결하여 API 형태로 데이터를 제공하는 것은 구현할 수 있었다.
2. **12주차 요약 정리**
   1. **화요일 주간 회의**- 11월 19일 화요일 점심에 약 1시간 정도 회의를 진행하였다. 알고리즘 팀(상윤, 태규, 진모)과 백 프론트 개발(찬영, 서영, 경민) 팀이 서로 주말 간 했던 것들과 앞으로 진행할 작업들에 대해서 공유하는 자리를 가졌다.
      1. **알고리즘 팀**  
         지금까지 여러 개의 논문을 한번에 텍스트로 바꾸는 작업을 진행하였다. 그리고 NLTK를 사용하여 영어 논문의 텍스트들을 토큰나이징하고 불용어를 처리하였다. 또한 TF-IDF를 scikit-learn 라이브러리를 사용하여 구현하였다. 앞으로 해야할 일은 크롤링을 통해 논문을 자동으로 다운받는 것과 TF-IDF 이외의 자연어 처리 알고리즘을 조사하는 것이다. 또한 two combined word를 처리하기 위해 고민해야 한다.
      2. **백 프론트 개발 팀**  
         지난 해카톤에 집안일로 참여 못한 서영이와 함께 하였다. 따라서 서영이와 같이 구현한 웹 사이트에 대해서 어디까지 구현하였고, 어느 부분이 변경되었는지 공유하였다. 그리고 앞으로 하기로 한 작업 내용에 대해 공유하고, 역할을 분담하였다. 지난 해카톤에서는 nodejs와 mysql을 연동시켜 DB에 있던 데이터를 가져오는 코드를 만들었다. 그리고 알고리즘 팀이 python으로 코드를 만들고 있기 때문에 mysql과 python을 연동해 추출된 데이터를 DB에 저장하는 코드를 만들었다. 마지막으로 MySQL workbench를 두르는 법을 배우고 DB를 설계한 뒤 workbench로 만들었다. 이후의 작업에 대해서는 경민이와 서영이의 경우 최종으로 설계한 데이터베이스에 대해서 서버에 구현하고, 이에 데이터를 임시로 넣는 작업을 하기로 하였다. 또한 찬영의 경우 없애기로 한 로그인 모듈에 대해서 다시 넣기 위해서 작업을 하기로 하였다. 다시 넣는 이유는 오픈 소스가 상호 연관적인 기능이 많아서, 데이터베이스와 연결하여 지속적으로 데이터를 주고 받기 위해서 이를 먼저 구현할 필요가 있다고 판단되었다.
   2. **금요일 주간 개발**  
      금요일 점심 12시부터 4시까지 개발 회의를 진행하였다.
      1. **알고리즘 팀**  
         TF-IDF 이외의 알고리즘에 대해서 조사하기 위해 <https://monkeylearn.com/keyword-extraction/> 과 <https://medium.com/analytics-vidhya/automated-keyword-extraction-from-articles-using-nlp-bfd864f41b34> 를 참고하여, Text Mining에 있어서 다섯 가지 접근 방식(Simple statistics approaches, Linguistic approaches, Graph-based approaches, Machine learning approaches, Hybrid approaches)에 대해서 알아보았다. - 태규의 경우,   
         키워드 추출 알고리즘 종류가 무엇이 있는지 확인하고 그 중에서 머신 러닝의 한 알고리즘인 Conditional Random Field(CRF)를 직접 이용해보고자 하였고, 오픈 소스 코드를 해석해보았다.   
         - 진모의 경우,   
         1. TF-IDF에서 unigram, bigram, trigram에 대한 결과 확인 (ngram\_range parameter)  
         2. TF-IDF에서 cid가 계속 포함되는 문제를 stopword parameter를 통해 해결  
         3. RAKE를 구현함 - TF-IDF보다 성능이 좋지 않은 듯 함.  
         4. PDF를 한글-영어로 구분하는 기능을 pdf to text 기능의 코드에 추가  
         5. 크롤러 개발중 - 한글/영어에 따른 페이지까지는 접속 성공, pdf 다운로드에서 막힘  
         - 상윤의 경우,  
         RAKE 소스코드 한번 확인해보고 적용 가능성에 대해서 검토해보았다.
      2. **백 프론트 개발 팀**  
         static 데이터를 관리하는 데이터베이스 설계 및 데이터 추가하였다. id, title, author, school, Publisher, year, degree, department, keyword, text의 column 으로 구성하여 데이터를 추가하였고, 데이터베이스에 특수문자 및 한글은 추가하지 않기로 하였다.  
         로그인 모듈의 경우 기존의 env path 및 config를 수정하여 AWS의 mysql 서버와 연동하였다. 또한 knex config 수정을 통해 같은 DB로 웹 페이지의 쿠키 및 세션을 관리하도록 하였다. 또한 오픈 소스의 경우 NoSQL의 일종은 redis로 설계되어 있는 것인데, 우리는 mysql을 사용하기 때문에 이에 수반되는 각종 문제들을 검토해보았다. 그 결과, 특별한 문제는 없었다.