# 캡스톤 18 조 장우혁 멘토링 활용 보고서

| 프로젝트명 | O24Sec | 팀명 | 멜러리를 찾아서 |
|---|---|---|---|

## 팀 멘토링

**[05/10] 윤명근 교수님**

(Q) 현재 유사도 검증을 코사인 유사도로 진행하고 있고, 다른 거리 기반 알고리즘을 사용할 때는 정답률이 좋지 않아 코사인 유사도로 계속 가는 것이 좋을까요?
[알고리즘 선정 문제]

(A) 먼저 지금 하고 있는 방법에서 유사도를 볼 때 알고리즘기반으로 진행하고 있었는데 tf-idf 벡터를 사용하는 것이면 코사인 유사도로 진행을 하는 것이 좋을 것 같다. 거리 기반의 경우에는 각 컬럼에 대한 의미가 전달이 되지 않을 수 있으니 각 컬럼의 의미를 살릴 수 있는 코사인 유사도로 진행을 해보자.

(Q) 코사인 유사도를 통한 실험에서 벡터 사이즈가 현재 묶여진 그룹내의 전체 단어 벡터를 사용하고 있어 차원이 20000~50000 가량 됩니다. 때문에 한번 연산에 소요되는 시간으로 인해 전체 계산에 시간이 너무 많이 소요되는데요 이걸 줄일 만한 방법이 있을까요?
[데이터 연산 속도로 인해 결과를 얻는 시간이 너무 많이 소요됨]

(A) 일단 라이브러리 사용을 최소화하고 진행을 해보자 라이브러리 호출에 소요되는 시간을 줄이는 방법도 있겠지만 아니면 알고리즘을 다른걸 찾아보는 것도 좋을 것 같다.

(A) 코사인 유사도로 하나씩 살펴봐서 비교하는 것 보다 클러스터링 방법 중에 DBSCAN 을 사용해서 밀도 기반으로 모아 보는 것도 좋을 것 같다. 알고리즘을 지금처럼 코사인으로 유지하고 머신러닝 라이브러리를 통해 진행하면 시간을 절약할 수 있을 것 같다.

(맬러리를 찾아서) DBSCAN 으로 진행하면 다시 Eps 랑 MinNode 파라미터 설정에 문제가 있어서 이 부분에 대해서는 실험으로 결과를 찾아보겠습니다.

(Q) 해시 테이블을 써서 벡터 차원을 줄여서 적용해 보기도 하였는데 이 경우에는 해시 충돌이 빈번히 일어나 클러스터를 묶었을 때 유사도가 떨어지는 것을 확인했습니다. 해시테이블 말고 데이터 자체의 포함관계를 보는 자카드 유사도를 사용해 보는 것은 어떨까요?
[기존에 실험 진행했던 방법에서 한계가 있어서 조사해본 알고리즘을 제시하여 확인해봄]

(멘토) 해당 데이터가 지금 사람이 읽을 수 있도록 디코딩도 할 수 있기 때문에 충돌이 많이 나는 것은 어쩔 수 없는 것 같다. 확실히 단어 들을 자를 수 있는 기술이 있으니 그 단어들의 포함관계로 유사도를 보는 것도 좋을 것 같다. 그 방법으로도 실험을 진행해보자.

| | |
|---|---|
| | 느낀점 => 적용 가능성을 확인하기 위한 유사도 실험이 효율적이지 못하다면, 기존에 구현되어있는 클러스터링을 다시 적용해서 성능 비교를 해보거나 결과를 분석해봐야 할 것 같다.<br><br>도식화 관련 교수님 멘토링<br>(교수님) 현재 도식화 이미지와 간단한 서술 위주로 포스터가 구성되어있는데 실제로 데이터 분석, 혹은 기술을 적용한 결과창 등을 추가로 삽입한다면 더 좋을 것 같다.<br><br>느낀점 => 단순히 이해시키거나 직관적이게 도식화하는 것 보다, 실제 데이터 차트와 실제 실행 및 시연 화면이 주는 이점에 대해 다시 한 번 생각해볼 수 있게 되었다. |