캡스톤 18 조 김민송 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.

=> 유클리드 디스턴스도 이용해 진행해 보고 있었는데 tf-idf 벡터의 숫자에 대한 의미가 살지 않아 고민하고 있었는데 이 부분에 대해서 확실히 할 수 있었고 단순 거리를 이용한 방법은 앞에 되어 있는 전처리에 적용하기는 어려울 것 같다.

(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.

=> Eps 는 어차피 유사도 임계치를 고르는 비슷한 수준이기 때문에 이 부분에 대해서는 똑같이 진행하고 MinNode 같은 경우에는 클러스터 최소 노드 개수를 정해 줘야 하는데 이건 실험을 통해 결과를 찾아야 할 것 같다.

(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. => 현재 데이터들이 같은 클러스터로 뭉치는 경우 유사한 수준이 아니라 정말 한, 두 부분만 다르고 완전 똑 같은 것 같아서 이 경우에는 차라리 워드 벡터의 포함관계로 유사도를 확인하는 자카드 유사도를 사용해 보는 것이 연산 속도나 전처리 단계를 줄이는데 더욱 효율 적 일 것 같다.
개인 멘토링	[05/07] 윤명근 교수님 – 1 대 1 미팅 (Q) 지금 데이터들을 클러스터링 했을 때 현 라벨대로 잘 뭉치는지, 아니면 디텍트 네임 붙은 걸로 잘 뭉치는지 아니면 클러스터개수가 적어야하는 건지 애매합니다. (A) 지금의 경우에는 라벨을 의심하는 상황이기 때문에 먼저 디텍트 네임으로 잘 뭉치는지 보는 것이 좋을 것 같다. 하지만 이벤트 중에 IP 로만 확인하는 데이터가 있을 수 있기 때문에 이부분에 대해서는 예외로 봐줘야 할 것 같다. => 먼저 만들어야 하는 CSV 에서 어떤 것을 기준으로 잘 뭉쳐졌는지 보여줄 수 있게 기준을 만들 수 있었다 이부분을 통해 윈스에 데이터를 보낼 때 detectName 이 잘 뭉친 것 같으니 이 기준으로 다시 봐달라 해봐야 겠다. (Q) 각 클러스터에서 대표벡터를 지정해줘서 대표벡터만 분석하고 클러스터 내의 다른 벡터에서는 대표벡터와의 차이점만 분석하여 시간을 줄이는 방법으로 기술을 만들고 싶습니다. 이때 대표벡터를 지정해주는 과정에서 클러스터 내에서 모든 데이터들과 유사도를 재연산 해서 유사도가 많은 벡터들에게서 높은 걸로 정해주려 했었는데요 이 때 너무 시간이 오래 소요되어서 대표 벡터를 정해주는 알고리즘을 새로 찾아봐야 할 것 같은데 어떤 방법이 있을까요? (A) 음 클러스터 내에서 각 단어들을 이용해서 대표값을 찾아내가지고 대표값과 비슷한 벡터를 대표로 만들어 보는 것이 좋을 것 같다. 대표 벡터라 하면 그 클러스터를 대표해 줄 수 있어야 하기 때문에 전체를 평균을 내는 방법이나 가장 빈번히 나타나는 벡터 형태나 그런 걸로 한번 찾아보자 => 대표벡터를 선정할 때 현재 시간 복잡도가 $O(n^2)$ 이라서 실용 할 수 없는 기술이였다. 이 시간을 줄이는 방법이 필요했고 몇 가지 시도를 더 해봐야 겠다.

	[05/12] 윤명근 교수님 – 1 대 1 미팅 (Q) 교수님 결과를 CSV 로 다 출력해서 만들었는데 잘 뭉쳐져 있는거 같습니다. 근데 이게 라벨이 다른 부분들이 있어서 결과를 다시 봐야 할 것 같습니다. (A) 윈스 쪽에 연락을 해서 우리가 만든 결과 자료를 토대로 다시 라벨 분석해서 돌려달라고 해보자. => 지금까지 만든 데이터를 윈스를 통해서 검증 (Q) 최종 레퍼런스에 사용할 성과치를 만들어야 하는데 어떤 것들을 그래프로 만들어야 한눈에 들어오고 훨씬 나아진 것을 보여줄 수 있을지 고민입니다. (A) 일단 우리가 만든 기술은 데이터를 분석할 때 시간을 확실히 줄여 줄 수 있다. 비슷한 이벤트 N 개를 각각 분석하는데 걸리는 시간을 N*T 이라고 했을 때 클러스터로 뭉쳐서 같은 걸로 한 번에 볼 수 있는 경우 소요되는 시간이 T 가 될 수 있는 것이지. 이때 (N-1) 만큼의 시간을 절약했으니 이부분에 대해서 보여주면 좋을 것 같고, 우리가 보낸 문서에서 전문가가 확인해 라벨을 수정해서 다시 보내준다면 그 부분을 이용해도 좋을 것 같다. => 마지막 최종 보고서를 위한 자료를 만들 때 어떤 것으로 구성을 할지 고민하고 있었는데 많은 도움을 받을 수 있었다.