컴퓨터캡스톤디자인 중간보고서

과제명	빅데이터 활용을 위한 상호 연관성 분석 및 분석 최적화 기술 개발		
팀 명	똑똑한 땅콩	구성 인원	총 5명 (교수 1명, 대학생 3명, 기업체 1명)
과제유형	□ 작동 (H/W+S/W) □ S/W 및 문제해결 ■ 산학협력멘토 □ 학제간 융합		
지도교수	소속학과	소프트웨어학부	성 명	이동호
산학협력업체명	(주)와이즈넛			

구 분	성 명	성별	학 번
대표학생	강은지	여	2018044239
참여학생	박서연	여	2018044566
	석예림	여	2018044720

팀구성 학생수	3명	배정 지원금	0원

보고 내용

1. 과제 개요 및 목적

가. 과제1

1) 과제 개요

: 빅데이터의 손쉬운 활용을 위해 다양한 데이터의 상호 연관성 분석 기술 개발

2) 추진 배경

: 방대한 빅데이터 중에서 분석을 위해 필요한 데이터를 확보하기 어려움

3) 개발 목표

: 데이터에 대한 전처리 과정으로 Outliers, Duplicates 제거 및 데이터 간 상호 연관성 분석을 통해 데이터에 대한 쉬운 접근성 제공

나. 과제2

1) 과제 개요

: 전체 데이터 분석 단계를 최적화하기 위해 머신러닝을 활용한 각 분석 단계별 Cost Learning 기술 개발

2) 추진 배경

: 복잡한 데이터 분석 과정을 성공적으로 수행하기 위해서는 각 데이터 분석 단계별로 적합한 분석 플랫폼(Spark, R 등)의 선택이 필요하고, 이를 위해 개별 분석 플랫폼에 대한 지식 및 이해가 필요함

3) 개발 목표

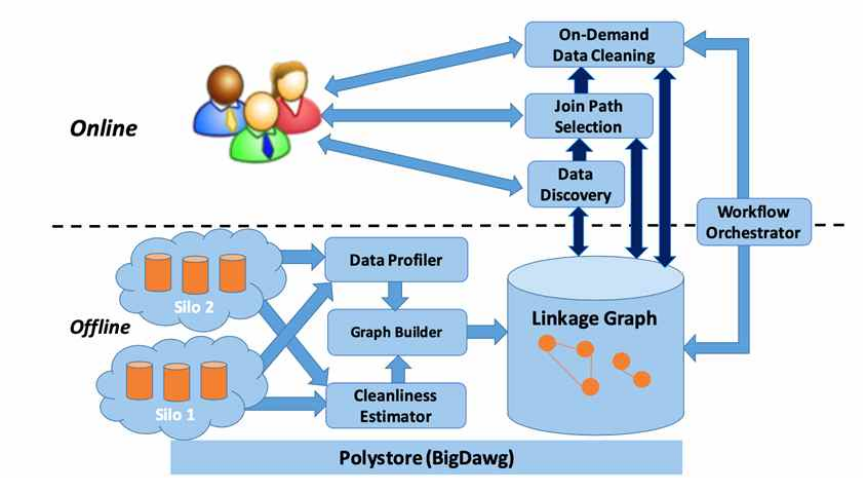
: 각 분석 단계별 학습된 Cost 기반으로 빅데이터 분석 플랫폼에서 자동으로 최적의 개별 분석 플랫폼을 선정 및 데이터 분석 진행.

2. 과제 설명 및 진행과정 보고

가. 과제설명

<과제 1> 데이터 상호 연관성 분석 기술 개발은 'The Data Civilizer System'이라는 논문을 바탕으로 진행되었다.

많은 기업에서 데이터가 기업 전반에 걸쳐 흩어져 있고, 일관성이 없는 경우가 많기 때문에 사용자가 원하는 데이터를 찾는데 많은 시간을 소비한다. 이를 해결하기 위해 제안된 방법이 Data Civilizer System이다.



Data Civilizer는 그림과 같이 크게 온라인, 오프라인으로 구분된다. 먼저, 오프라인 영역에서는 사용자가 원하는 데이터가 여러 데이터 저장소에 분산되어 있는 경우, 이들을 통합하는 작업을 먼저 진행해야한다. 여기서는 polystore를 사용한다. 통합된 데이터 저장소로부터 불러온 데이터 셋에 대해 프로파일링을 진행하고, 이 정보를 바탕으로 linkage graph를 만든다. 온라인 영역에서는 데이터 셋들의 검색, join path selection, 클리닝 작업이 혼합된 사용자 제공 워크플로우를 생성하여 오프라인 영역의 linkage graph와 상호작용한다.

이 시스템은 다음과 같은 방식으로 진행된다.

1. 수많은 테이블에서 사용자가 요청한 데이터 셋을 신속하게 검색
2. 관련 데이터끼리 연결(프로파일링 -> linkage graph 생성)
3. 각 데이터 셋들이 속해있는 데이터 저장소와의 응답시간을 계산
4. 원하는 데이터를 클리닝(cleanliness)
5. 1-4 반복

나. 구현상세

Polystore로 그래프 db인 ArangoDB를 사용하였다. 또한 그래프 디비로 보낸 데이터를 다시 가져와 클리니스 작업을 수행해야하는 번거로움을 줄이기 위해 데이터클리닝(클리니스)작업과 프로파일링 작업의 순서를 변경하였다. 따라서 구현된 시스템에서는 클리니스 작업을 거친 데이터를 가지고 프로파일링을 진행하게 된다. 각 데이터가 속해있는 데이터 저장소(DB)에서 데이터를 가져올 때 메모리 압박을 줄이기 위해 페이징 작업을 거쳐 데이터를 가져온다. 페이징 된 데이터가 아닌 전체 데이터 셋에 대한 메타 데이터가 필요하므로 증분 프로파일링을 수행한다.

시스템을 크게 프로파일링, 클리니스, 업데이트 세부분으로 나누어 구현하였다.

각 단계에 대한 세부 구현 사항은 다음과 같다.

1) 데이터클리닝(클리니스)

: 데이터 전처리 과정으로 데이터 클리닝 작업을 하기 위해 먼저, dataset의 비어있는 값(null, NaN)을 제거하였다. 날짜 데이터는 유효성 검사를 진행하였고, 수치형 데이터는 IQR을 사용하여 Outlier를 제거해주었다.

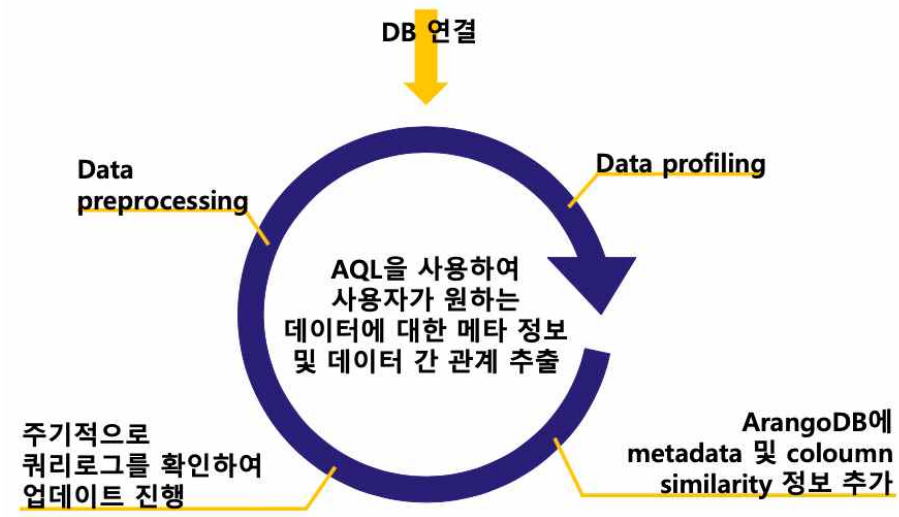
2) 프로파일링

: Linakge graph를 만들기 전, 데이터를 통계 및 요약하는 과정이다. 테이블의 메타데이터, 각 칼럼의 메타데이터, 칼럼 간 메타데이터를 계산하였다. 계산된 메타 데이터를 바탕으로 다음 단계로 전달할 정보를 결정하였다. 다음 단계에서는 전달된 정보를 바탕으로 원하는 정보(column similarity, inclusion dependency)를 계산하였다. pandas_pofiling 라이브러리를 사용하여 프로파일링을 진행하고, 메타 데이터를 추출한다. 이를 바탕으로 칼럼 간 유사성 및 inclusion dependency 계산을 진행하였다.

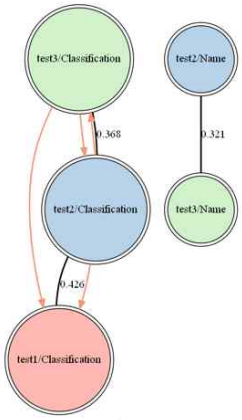
3) 업데이트(데이터(셋) 추가, 삭제 등)

: 업데이트가 발생한 경우, 어떤 데이터 셋이 업데이트 되었는지 파악하는 것이 업데이트 구현에서의 주된 논쟁이었다. 업데이트 전 메타데이터와 업데이트 후 메타데이터를 비교해 변화가 생겼을 경우만 그래프 디비에 변경된 메타데이터를 업데이트 하게 된다. 매번 모든 데이터 셋에 대해 프로파일링을 진행해 비교하는 것은 시간이 많이 소요되었다. 그래서 데이터 베이스의 쿼리로그를 읽어오는 방식을 사용해 업데이트 목록을 선제적으로 파악하기로 하였다. 시스템에 기록된 마지막으로 읽어온 쿼리로그의 시간 이후의 쿼리문만 모두 불러와서 변경사항을 업데이트한다.

4) 시스템의 전체적인 구조



주기적으로 업데이트를 진행하며 클리닝-프로파일링 작업을 반복하면서 사용자가 원하는 데이터가 있을 경우, AQL을 사용하여 사용자가 원하는 데이터에 대한 메타정보 및 데이터 간 관계를 추출할 수 있게 된다. 추출한 데이터를 사용자에게 보여주기 위해 ArangoDB로부터 이 정보를 json파일로 받아와 graphviz라이브러리를 사용하여 다음과 같은 그림으로 보여준다.



FOR x IN relation FILTER x.value > 0.3 RETURN x

칼럼 간 유사도는 수치와 함께 검은색 edge로, inclusion dependency는 분홍색 edge로 나타난다. 화살표 방향은 FK에서 PK 방향이다.

3. 향후 계획

가. 현재 우리 프로그램은 현재 수치 데이터를 중점적으로 개발되었다. 텍스트에 대한 부분을 더 강화해 나갈 계획이다. 텍스트 클리니스 및 유사도는 LSA, GloVe, fasttext 3가지 기법 중 프로그램에 적합한 방법을 채택하여 진행할 것이다.

나. 3가지 기법을 테스트 해 본 결과, 기법이 유사한 LSA와 GloVe 중 성능이 더 우수한 LSA와 fasttext를 함께 사용하기로 결정했다. 이 기법들을 중점으로 텍스트에 대한 부분을 개발해 나갈 것이다.

다. 텍스트 부분 보강 후 1월 안으로 <과제1>을 마무리하고, 동계 인턴십을 진행하며 <과제2>인 '전체 데이터 분석 단계를 최적화하기 위해 머신러닝을 활용한 각 분석 단계별 Cost Learning 기술 개발'을 진행할 것이다.