Education Today Report

Mengfan Zhu

1 Data Orepartion and Database Setup

Just keep the necessary columns which are related to the tasks from the original data files. The codes for this step is `/database/data_prepare.py`.

Tables	Attributes
Affiliations	AffiliationId DisplayName
Authors	AuthorId DisplayName LastKnownAffiliationId
Papers	PaperId PaperTitle
PaperReferences	PaperId PaperReferenceId
PaperAuthorAffiliations	PaperId AuthorId

Table 1: Necessary attributes for each table

Write sql code to create tables (`/database/create_tables.sql`) and load data into database (`/database/import_data.sql`) In this step, the processed txt files should be put in the folder `C:/ProgramData/MySQL/data/educationtoday/`

2 Basic frontend and backend Structure

Install neccssary package for react and flask.
For the frontend, auto-generate the basic react structure and delete unnecessary files. The folder `/app/` is for frontend. The common parts and routers are in `/app/App.js`. The pages for tasks will be put into `/app/pages/`.
For the backend, create new folder `/backend/`, setup the database connection and the routers in the `/backend/api.py`. The detailed implementation for tasks will be put into `/backend/controllers/`
Set up `proxy` in `package.json` to connect frontend and backend.

3 Frontend Pages

For each task, there is a page with a form to get the user input. (`/app/pages/src/task1.js` and `/app/pages/src/task2.js`) The form will let the user input the author/institution name to be search and choose how many results will be shown. Once the user submit the form, it will redirect to the result page(`/app/pages/src/result1.js` and `/app/pages/src/result2.js`) which will show the paper title/institution name and the frequency. For the implementaion,

4 Backend Implementation

The key point is to design the sql queries to get the result.

Task1

This task can be completed in following steps:

- Find the `AuthorId` based on the given name
- Get all papers wrote by this author based on `AuthorId`
- Find out the reference papers for each paper wrote by the author
- Count the frequency and get the required number of top papers

And then combined all of these operations into one sql query.
The detailed implementation is in `/backend/controllers/task_one_controller.py`

Task2

Use stored procesure for this part (`/database/Procedure_task2.sql`) That is because this task requires to find the co-authors for each author in an institution. Therefore, we need to use cursor to access each author in the institution. That is why I used stored procesure here.
The steps are:

- Find the `AffiliationId` based on the given institution name
- Set up cursor for all authors related to this affiliation
- Use the cursor to access each author in the affiliation, find out the set of co-authors for each author and stored them in a new table
- Get distinct `AuthorId` from the new table and count the frequency of the affiliations related to them
- Get the required number of top affiliations

5 Pass Parameters for Frontend and Backend

I pass the parameters using URL. I put the `?key1=value1&key2=vaule2` after the path to pass the necessary parameters when do redirection in the frontend and fetch the backend resource.