eduSearch

Aaron Wu, Aidan Gratton, Cindy Li, Simran Thind

Features

- Draws data from the GoogleCustomSearch and GoogleScholar APIs
- Result Filtering
  - Filters results using a domain whitelist, including many educational websites and all .edu domains
  - Combines data from multiple sources, returns only unique results
  - Scalable: interface supports addition of many different APIs
- User Site Specification
  - Specific domains to search can be specified
    - i.e. only results from khanacademy.org will appear
  - Specific domains can be excluded from results
    - i.e . no results from wikipedia.org will appear

How it Works

- Tech Stack
  - Frontend: React.js and Blueprint.js
  - Backend: Node.js and Express with Python
- Pipeline
  - User enters search query into web interface
  - Query handled by Express server and passed to Python script
  - Query sent to Google Custom Search and Google Scholar APIs
  - Script outputs filtered results
  - Results displayed on web interface
- Result filtering
  - Results from each API parsed into common format
  - Common results format passed to filtering interface, which returns a unique list of relevant results
- Frontend-Backend Interfacing
  - Node.js calls Python process with query string
  - Python process returns results JSON

How We Did It

- Project Planning
  - Overarching project infrastructure planned as a group before writing any code
  - Data formats agreed upon in advance
  - Implementation details left to each individual
- Division of Labor
  - Work split into Frontend (Simran, Aaron) and Backend (CIndy, Aidan) teams
  - Code structured as a series of modules to avoid merge conflicts
- Communication
  - Used discord for communication
  - Helped each other with debugging via screen sharing
- Prioritization
  - Sacrificed nice-to-have features to complete core functionality
    - Autocomplete implementation shelved due to time constraints

Potential Extensions

- Expand API support
- Search Autocomplete
  - Provides autocomplete search suggestions as the user types
  - Successfully implemented via the Bing Suggestions API
    - However we did not have enough time to integrate it with the frontend
- Public Hosting
  - Currently the webapp is hosted locally
  - Could be deployed for public access with minimal code changes