Link Analysis

Team S

Siyan Zuhayer, Erika Ingersoll, Asad Ul Haq, & Kellie Ng

What is Link Analysis?

Link Analysis

Process of analyzing hyperlinks between web pages to determine their importance, relevance, and relationships to other web pages through a directed graph

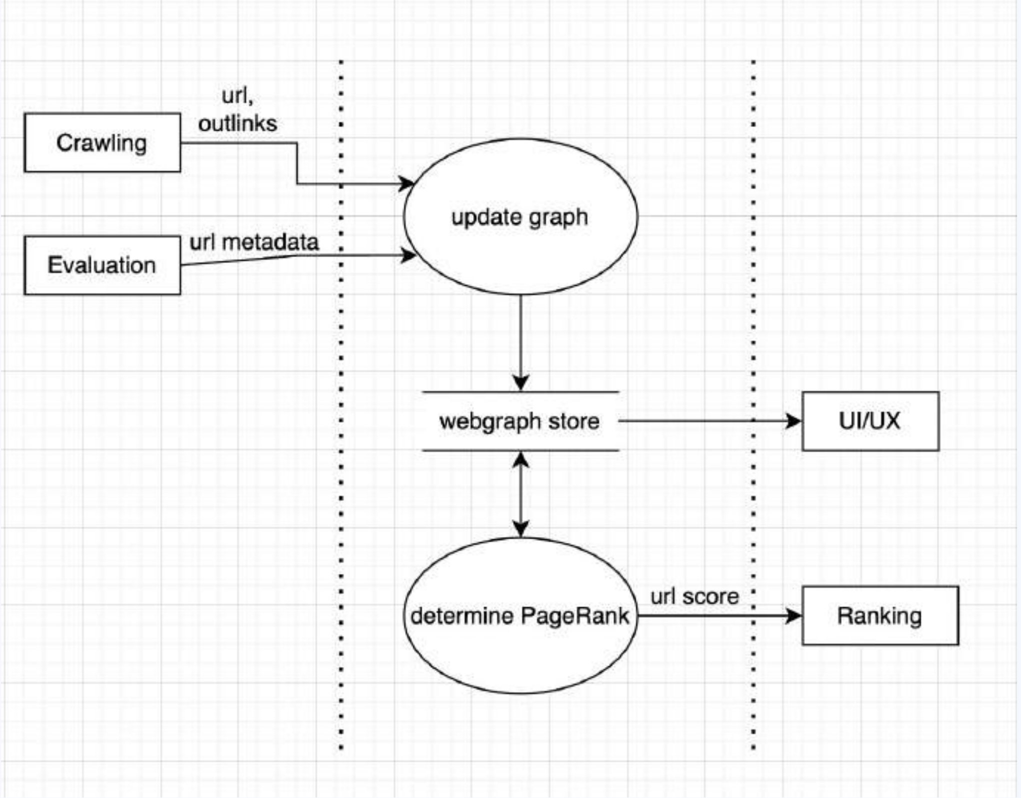
Relevance

Responsible for evaluating links from crawled documents and using link structure to support crawling, ranking, etc

PageRank

Treats the web as a directed graph where the nodes are the webpages and the edges are the hyperlinks from one page to another. Gives a score to pages based on the number of incoming links

DFDs

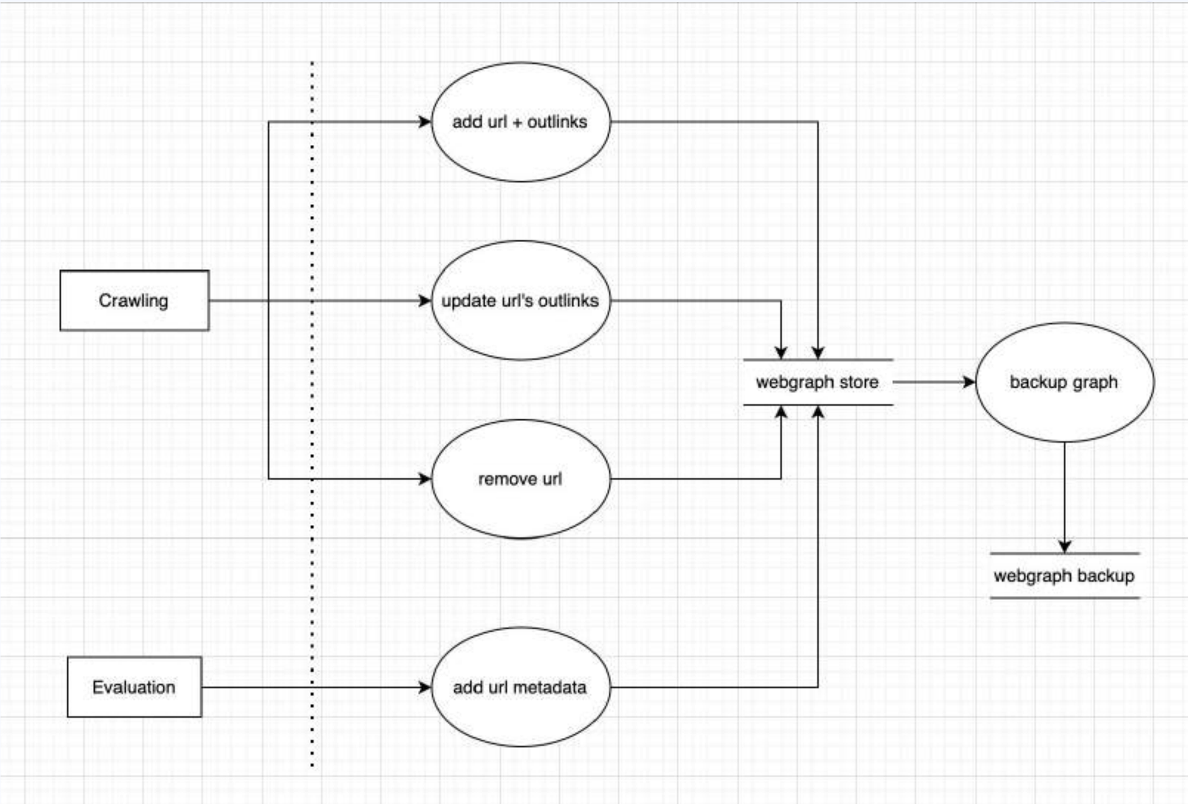


Level 0 DFD

DFDs

Update graph:

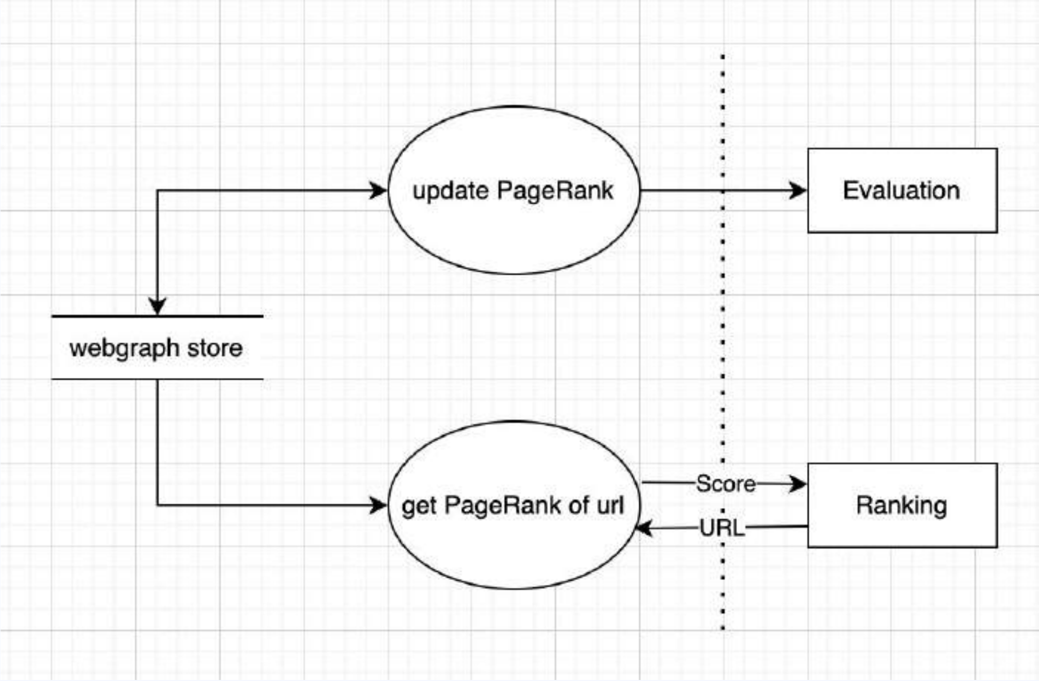
Level 1 DFD



DFDs

Determine PageRank:

Level 1 DFD



Important Design Decisions

01 PageRank scores need to be regularly updated

- ❖ We initially decided to update the scores every 20 minutes
- ❖ We will look into heuristics we can use to change the update frequency based on how often the link graph changes

Important Design Decisions

02 — **During PageRank score updates, new pages and updated pages will still be expected to be added to the link graph**

- ❖ We will have a mutex to prevent changes to the link graph from outside components during updates
- ❖ We will have a queue of updates to be performed on the link graph after the PageRank score updates finish

Important Design Decisions

03

Crawling tells us what nodes/edges to add to the graph

- We debated over who would tell the link analysis component to add new nodes and edges and when that would happen
    - Main issue: what links should be added? (crawled/uncrawled, links that violate politeness policies, etc.)
- We came to an agreement with Crawling where they would pass off URLs and outlinks to us, ensuring the outlinks are safe to add to the graph

Making Our Product Socially Aware

Block sites that use slurs, offensive terms, and hate speech

Block sites that may not be using offensive terms, but have negative connotations and a negative bias that may offend certain groups of users

Avoid keeping image links that may be offensive to specific groups of users

Making Our Component Socially Aware

When other components detect malicious links, we will remove those links from the webgraph

- We heavily depend upon other components detecting these issues since we can only look at links between pages

If a page was linked to or from a malicious link, we can notify other components of these pages to check if they're also malicious

Implementation Plans

Language: Python

Coding Standards/Style:

- Graph library: graph-tool
  - Performant since it's built in C++
  - Storage: built-in graph-tool method
- Use github and consistently push new code, using version control history, and write clear commit messages
- Conduct code reviews before merging code into main
- Write unit tests after each function is done to ensure smooth testing and developing

Implementation Plans

Implementing basic features

- Instantiating and storing the graph
- Ability to Add/remove nodes and edges
- Integration of PageRank Algorithm
- Interface to add metadata from evaluation team.
- Graph hand-off to UI/UX
- Regular PageRank updates

Implementing advanced features

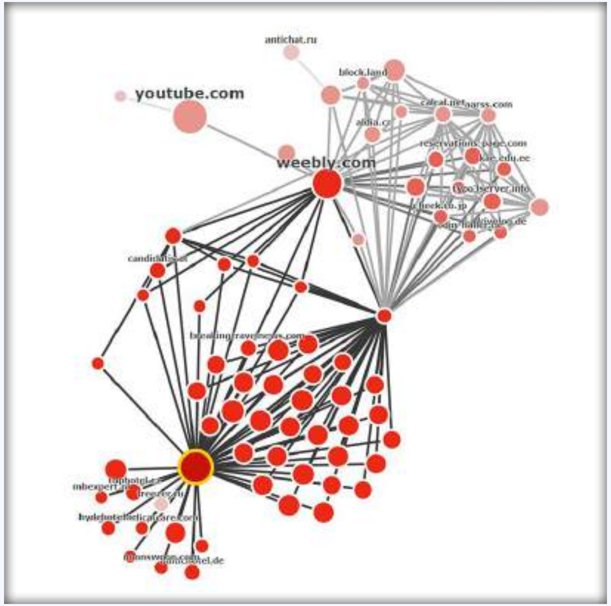
- PageRank update frequency heuristic
- Remove malicious links and notify other components of links to/from those links

Interesting Test Cases

Component Tests	System Tests	Integration Tests
❖ PageRank updates ➢ Don't allow nodes and edges to be added/removed during update ➢ Do all other updates after PageRank update finishes	❖ Multiple large scale PageRank updates ➢ Update needs to take a reasonable amount of time to quickly clear the update queue and allow for more updates to the graph before next update	❖ Adding/removing nodes/edges ➢ Simulates Input from Crawling ➢ Test different types of nodes/edges ➢ Attempts during PageRank updates ❖ Getting PageRank score given a URL ➢ Involves Ranking ➢ Valid/invalid URLs

What Beta Testing Will Include

- Basic functionality: adding nodes, removing nodes, link graph
- Regular updates to PageRank scores (constant intervals for now, will change in the future to improve on performance)



https://majestic.com/reports/site-explorer/link-graph

Thank you.

Questions?