Team S (Link Analysis)

Team Logistics

Team Members & Our Roles:

Brief summary of each team member's role.

Asad Ul Haq: I worked on making the DFD for our SE component, and liaised with other teams to agree on the interface.

Kellie Ng (ngk4): I worked on the Requirements section of this deliverable, as well as formatted it properly. I set up a team meeting outside of class time to ensure we finished the deliverable. Moving forward, I will send out a whenisgood to set up meeting times outside of class when required.

Siyan Zuhayer: I worked on the Architectural Divisions section and helped with making the DFD.

Communication Methods:

We have a group chat on both SMS and Discord. We will mainly be using Discord to communicate.

Where Our Project Work Will Be Stored:

We will be using GitHub to store our project work.
https://github.com/anotherAsad/LSPT_semester_project

Requirements Review (Kellie)

List of Requirements that tie directly to Link Analysis

- 5 – Show an interactive webgraph (pages and links)
- 6 – Show how old crawled pages are using color gradients on the interactive webgraph
- 12 – Provide a means to tune the ranking algorithm based on text statistics, link structure, query structure, etc.
- 17 – Do not allow sites to artificially raise their page/site rank in search results
- 39 – Ignore specific URLs via an admin-maintained list
- 42 – Provide a "safe" mode (default) to search in that omits any unsafe content from search results
- 44 – Support a "leave me alone" mode in which no tracking or history is recorded
- 52 – Record invalid URLs and broken links for later review by admins
- 63 – Search results must never be empty, i.e., not possible for "no results found"
- 68 – Search results should only show content related and relevant to RPI

Ambiguous or Unclear Requirements (at least 8)

State what is unclear and ask a clarifying question.

- 17 – Do not allow sites to artificially raise their page/site rank in search results
  - How do sites raise their rank?
- 22 – Support exact word/phrase matching using quoted queries, e.g., “CSCI” or “computer organization”
  - Does this apply more to text acquisition/crawling when the query is sent or does it apply more to LA?
- 23 – Support word exclusion using “-” (e.g., RPI -raspberry)
  - Does this apply more to text acquisition/crawling when the query is sent or does it apply more to LA?
- 38 – Automatically identify malicious/spam websites
  - Who handles sensitive data - text transformation, link analysis or crawling?
- 39 – Ignore specific URLs via an admin-maintained list
  - Who handles sensitive data - text transformation, link analysis or crawling?
- 42 – Provide a “safe” mode (default) to search in that omits any unsafe content from search results
  - Who handles sensitive data - text transformation, link analysis or crawling?
- 57 – Show “quick info” results in query results, e.g., prominent people, department, etc.
  - How / from where is the quick info found/identified?
- 61 – Optionally show key reason why each given search result was ranked as it was, e.g., document title, inlink count, etc.
  - How is the key reason identified? (probably implementation specific or something?)

Missing Requirements

Define at least eight missing requirements, at least half of which must be “implicit requirements” that relate to scalability, reliability, maintainability, etc.

- Limit the number of results shown and have multiple pages of results if needed
- Show related searches, e.g., “People also searched”
- Links that have been visited should be shown in a different color (blue if not before visited, purple if has been visited)
- Should be active 24/7 (implicit)
- Should have a backup engine in the case of failure or error (implicit)
- Maintenance in one area shouldn’t affect a different sector, e.g., LA maintenance shouldn’t affect the whole system (implicit)
- Ignore filler words such as “a”, “the”, etc.
- Make optimal use of CPU memory and storage (implicit)
- PageRank should be run at regular intervals and it should take into account the frequency of input form the crawler

New Requirements

Describe what our SE should or shouldn't do to avoid causing a negative impact on specific groups of users.

- ➢ Block sites that use slurs, offensive terms, and hate speech
- ➢ Block sites with negatively opinionated blogs/articles/pieces, e.g., blocking sites that may not be using offensive terms, but have negative connotations
- ➢ Bar the user from searching queries with slurs, offensive terms, and hate speech
- ➢ Avoid showing images that may be offensive to specific groups of users
- ➢ Bar the user from searching with images that include slurs, offensive terms, and hate speech

Architectural Divisions (Siyan Zuhayer)

Describe architectural divisions relevant to LA.

Ranking, UI/UX

getGraph():

- ➢ Input - none
- ➢ Output - webgraph (immutable)
- ➢ Side effects - none
- ➢ Called by Ranking, UI/UX

Crawling

addNode():

- ➢ Inputs - url, parent (where url was crawled from)
- ➢ Output - success/failure
- ➢ Side effects - add node to webgraph
  - ○ If the parent and the URL are already in the graph nothing should happen.
    - ■ If a link doesn't exist between them yet, add one.
  - ○ If the parent and/or URL isn't already in the graph, add them and a link between them.
    - ■ If anything changes, update page rank scores
    - ■ Additionally, it should have a timestamp showing when the last update occurred.
- ➢ Called by Crawler

Evaluation

reportMetric():

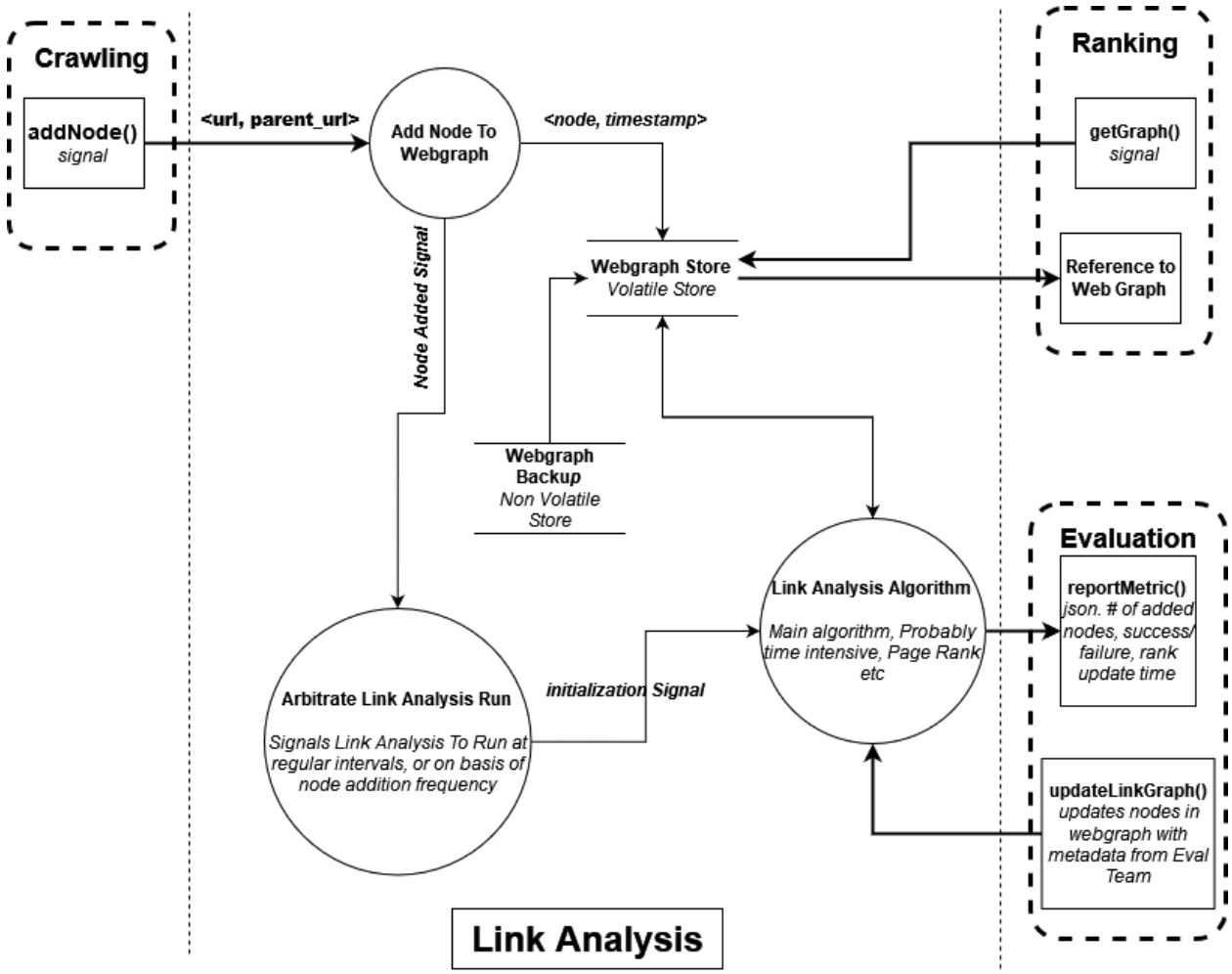
- ➢ Inputs - json of components of number of added nodes that succeeded or failed, and how long the page rank update took
- ➢ Output - none
- ➢ Side effects - updates evaluation data for link analysis

- ➢ Called by Link Analysis

updateLinkGraph():

- ➢ Inputs - list of nodes with data about them, e.g., was it clicked/ignored, timestamp at which query happened
- ➢ Output - none
- ➢ Side effects - updates nodes in webgraph with the sent metadata
- ➢ Called by Evaluation

Preliminary SE Design (Asad Ul Haq)



Focusing on your SE component, show a preliminary data-focused design for your component.
For your design, use one or more DFDs to show the flow of data through your component, as well as what data will need to be stored. The "edges" of your preliminary design are the interfaces to the other SE components. Note that your DFDs here will be more detailed than your top-level architectural DFD.
As you work on your design, new requirements might emerge; capture these as part of your requirements review.

Resources

https://submitty.cs.rpi.edu/courses/f24/csci4460/course_material/lectures/csci4460-f24-week-03-search.pdf
https://www.cs.cornell.edu/home/kleinber/networks-book/networks-book-ch14.pdf
https://en.wikipedia.org/wiki/PageRank