GOOGLE SEARCH

Overview

- What is Google Search?
- Life of a Search Query
- Indexing the Web
- Classic Google
  - PageRank
- Google Now
  - Improving Speed (MapReduce)
  - Improving Results
  - Reducing Costs
- Related Topics

What is Google Search?

IN

Keywords, Phrases

OUT

Relevant, Quality Results

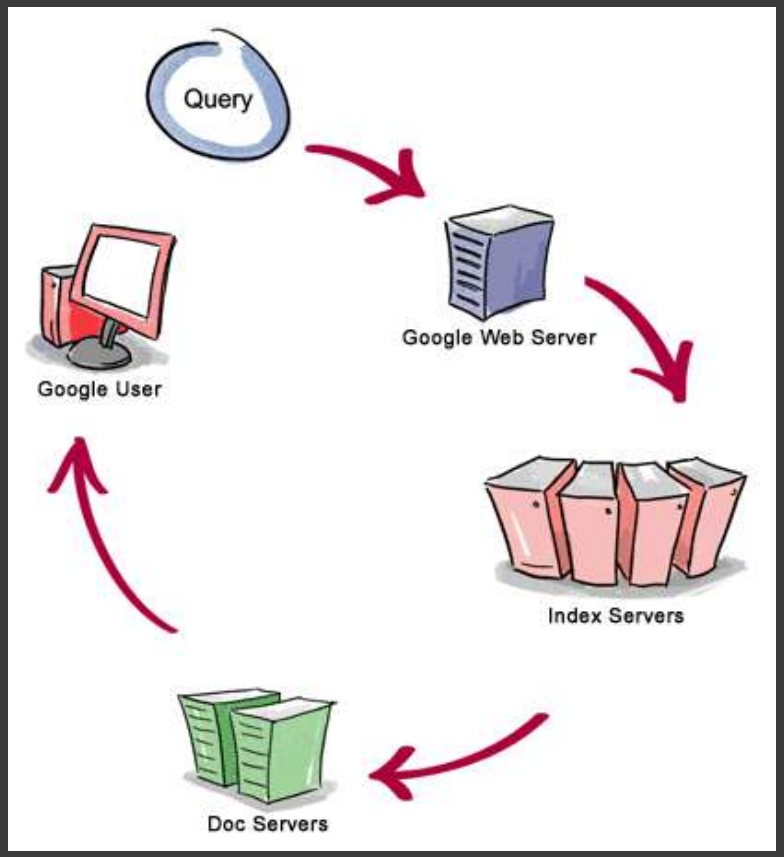
Google Strengths

- Extremely fast results
- Highly relevant results

- Expertise in storing and processing enormous data sets (Petabytes)

Life of a Search Query

Life of a Search Query



- Web Server
  - Communicates with user's browser
- Index Server
  - Translates keywords to relevant results
- Doc Server
  - Loads the page snippet shown to user

Decompose Query

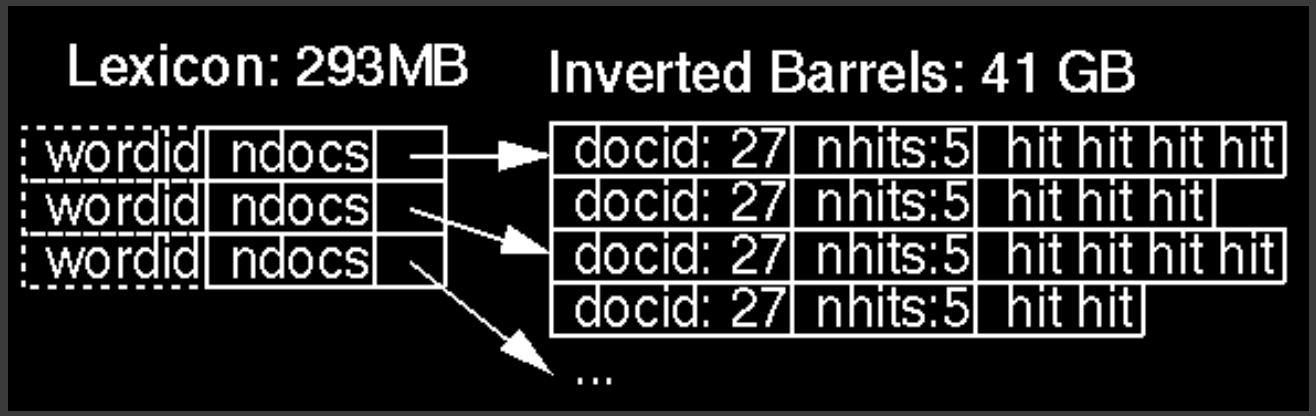
- Query broken into keywords
- Canonicalization
  - Minimal punctuation (keep hyphen)
  - Normalize casing
  - Spell checking
- Match keyword strings with Lexicon database to get *WordIDs*

```
This, is a query!  ->  this is a query  ->  #2 #54 #3 #9285
```

Introduction to Data Structures

Life of a Search Query

The Inverted Index



- Keyword *WordID* gives a list of documents and each occurrence of the keyword in document
  - These keywords in the documents are called ‘hits’
- Will discuss how this is generated later

A ‘hit’ in a document

- Describes a match of word in document
- Part of index database
- Annotated with:
  - Location in file
  - Relative font size (Headers are important)
  - IsCapitalized
- Hits for words in document as well as words in incoming link titles.

Ranking Documents

- Each document has intrinsic quality scores independent of the query
  - If two pages match a keyword, we want higher ranked one.
- PageRank(TM) is most well known
  - We will discuss later
- Google uses 200+ metrics for this
  - Most are trade secrets

Back to the Query!

Life of a Search Query

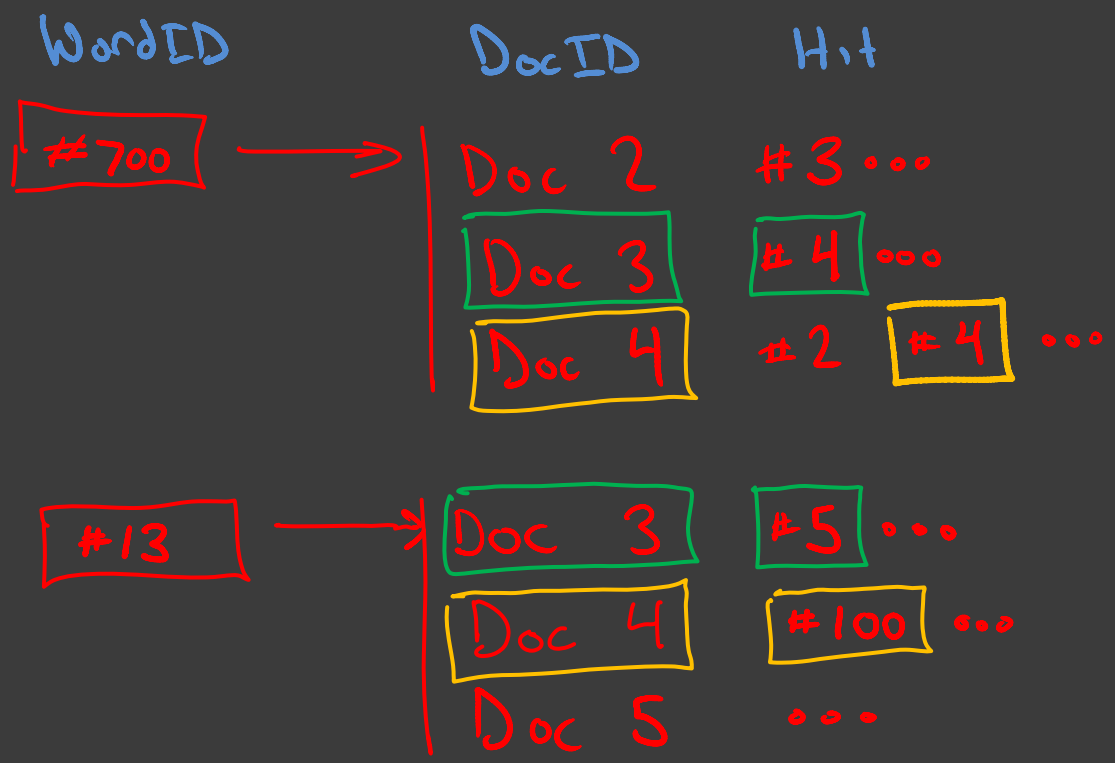
Simple Query

- Look up keyword in inverted index
- For each Document:
  - Classify Hits by font size class
  - Count number in each size class (clamped)
  - Each size class has a weight
  - Compute relevance score
  - Add documents intrinsic ranking score
- Pick top documents (Relevant + Quality)

Multiple Keywords

- Introduce concept of proximity classes
  - "*Phrase Match*" .. "*Not Even Close*"
- This allows strong matching of phrases and proper nouns (names)
- Classify Hits by {*Proximity*} x {*Font Size*}
- Walk Index for all keywords at once
- Classify proximity of keywords in each document
- Compute scores as before

Example

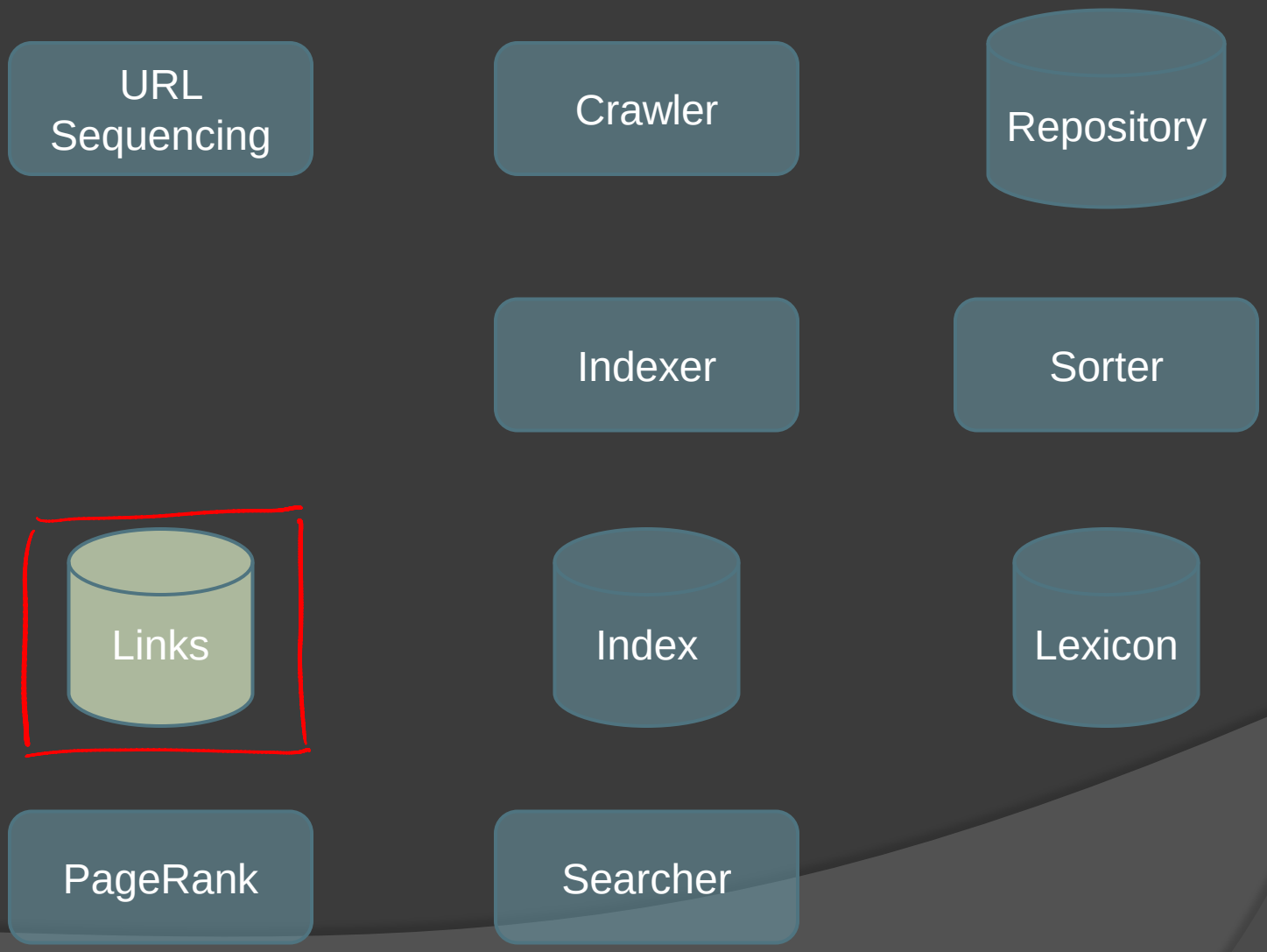


"Phrase Match"

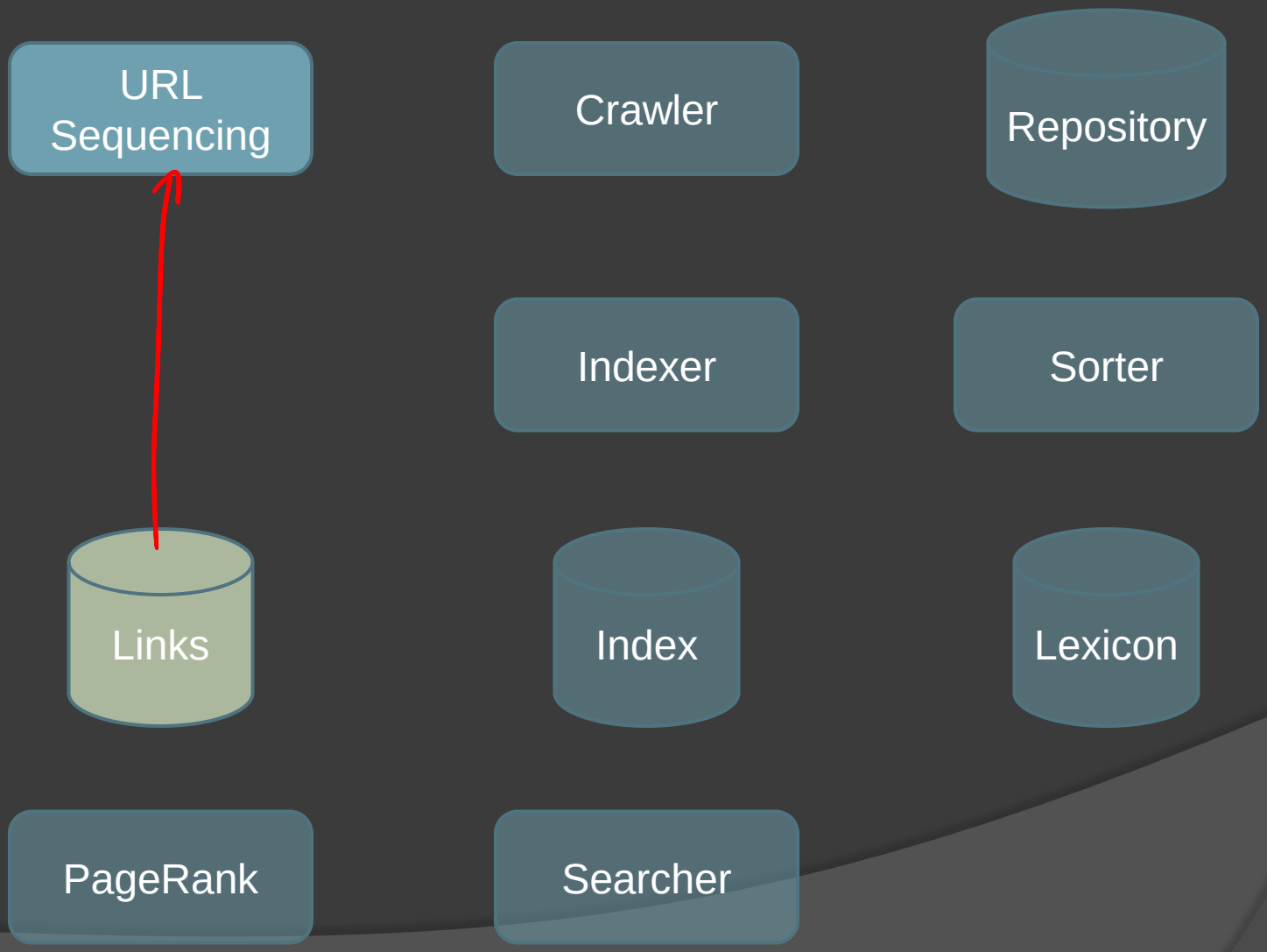
"Not Even Close"

Indexing the Web

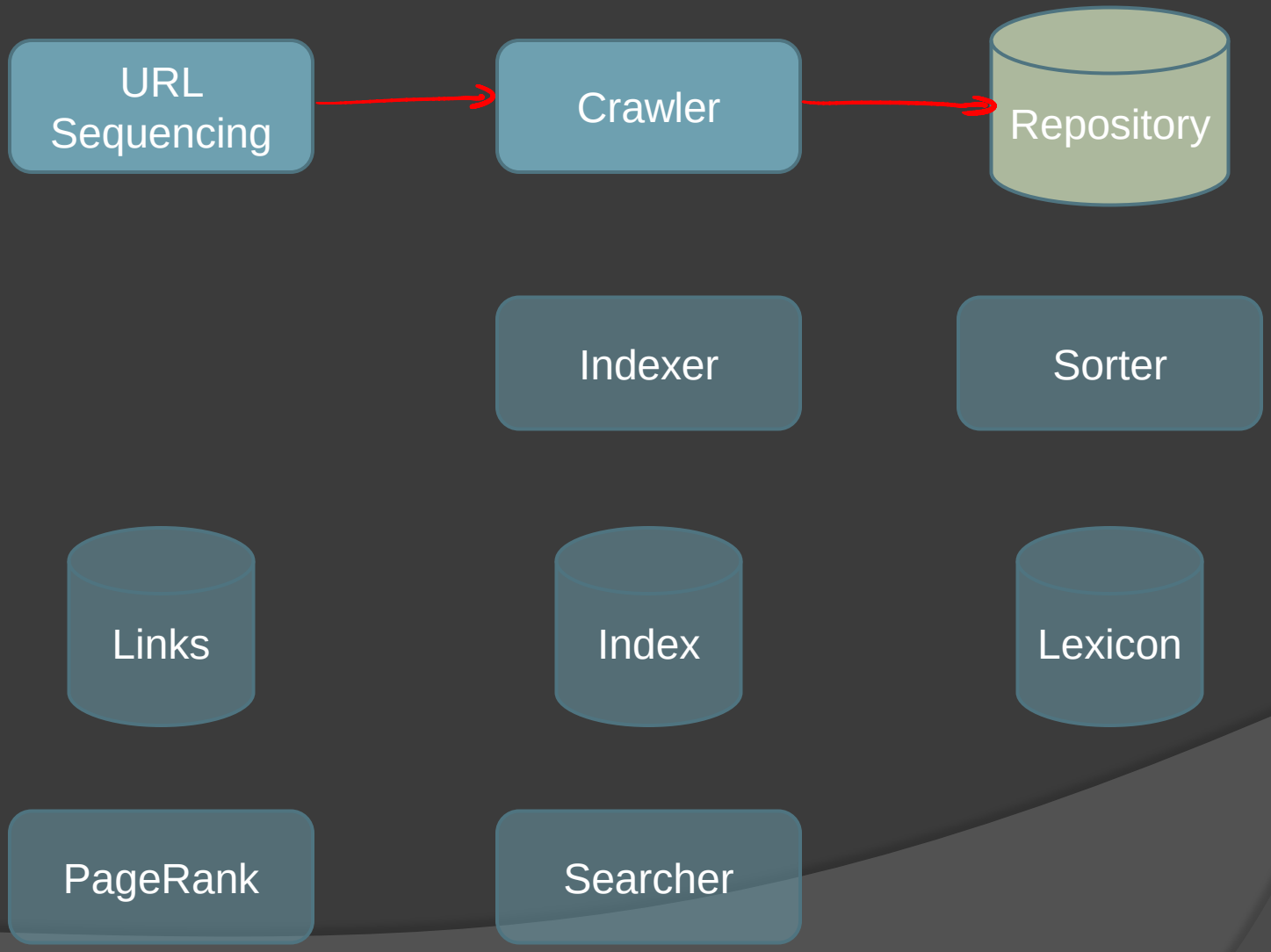
Start With Previous Results



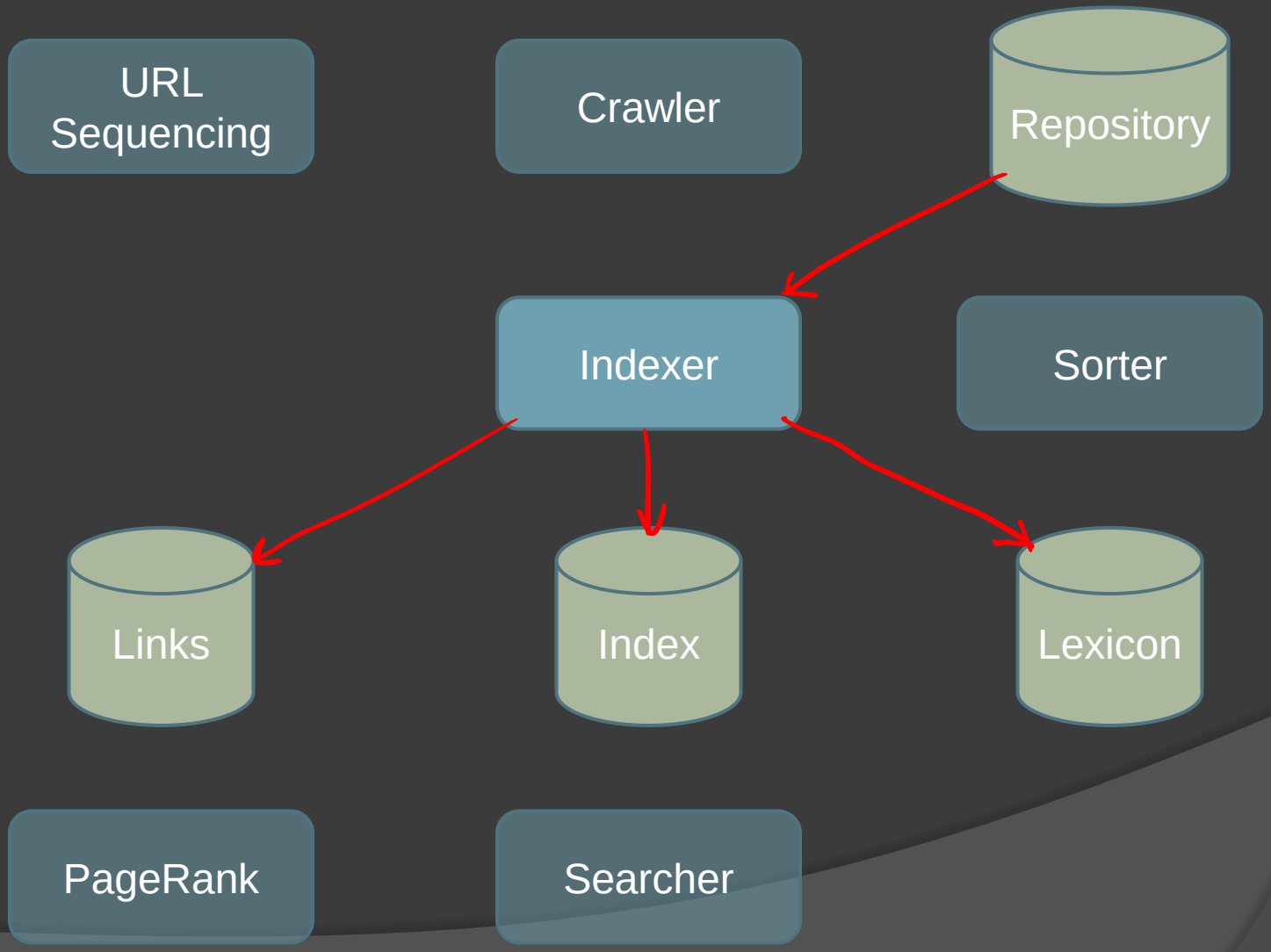
Generate URL Sequence



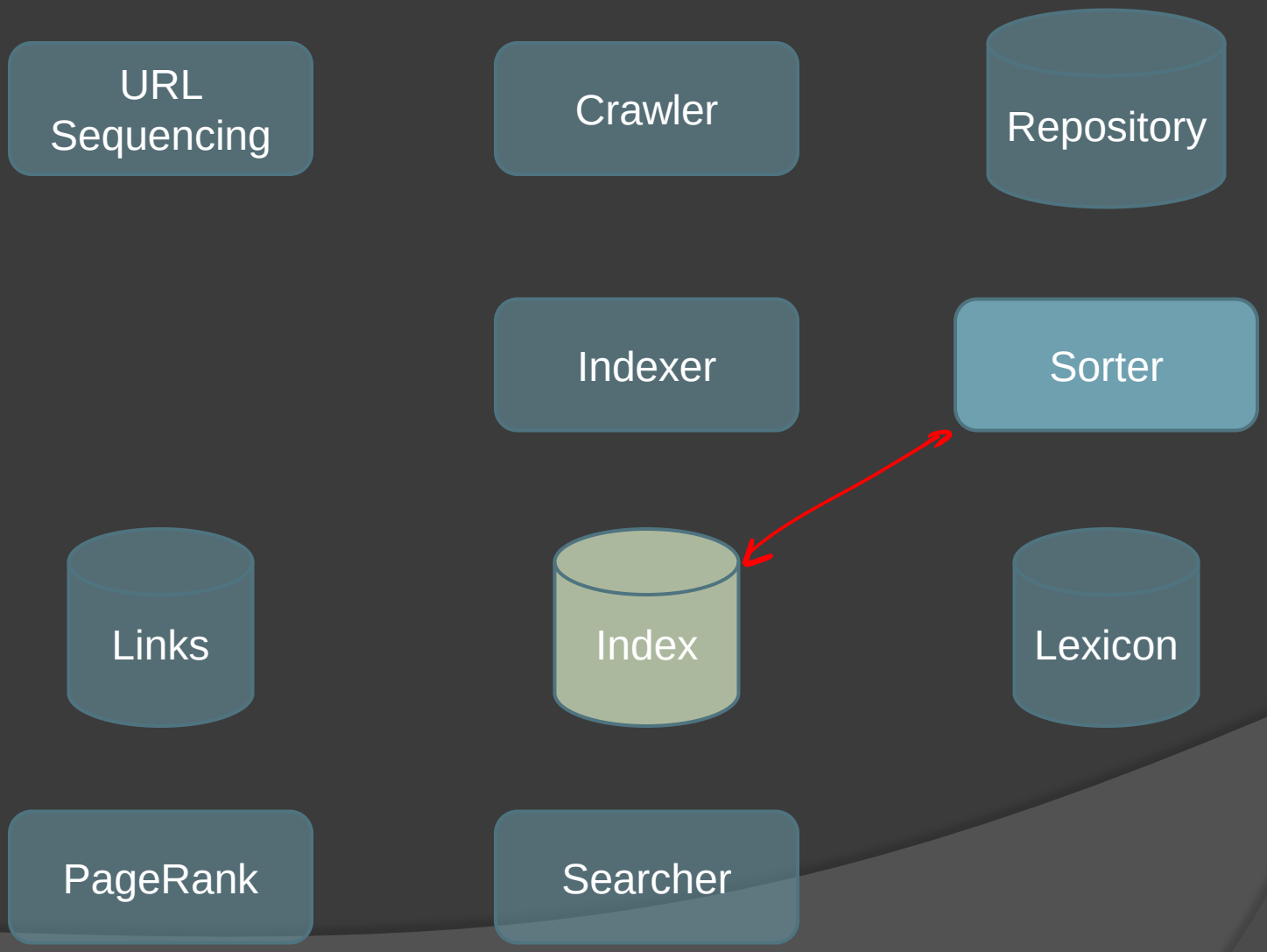
Download the Web



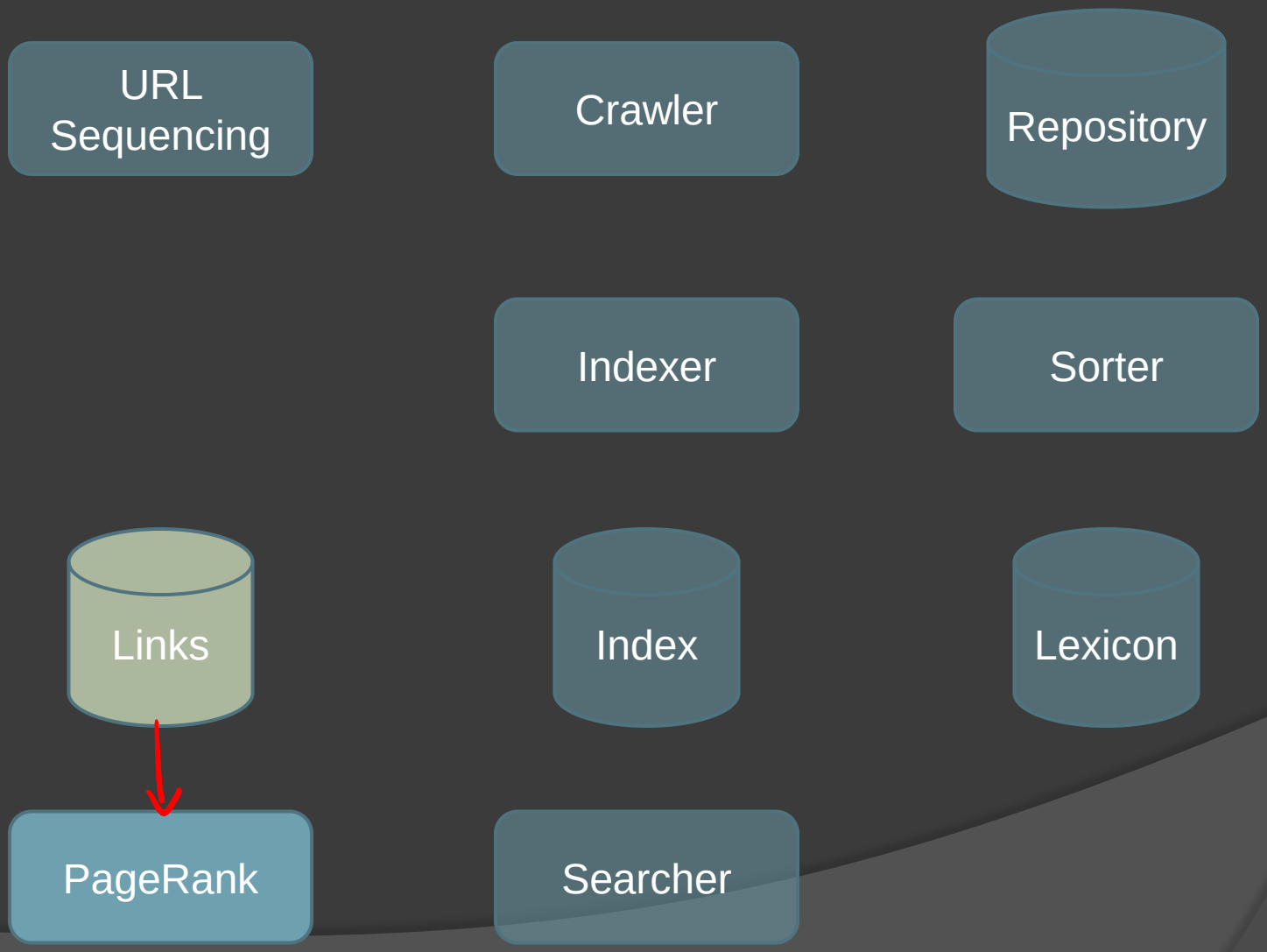
Decompose Documents



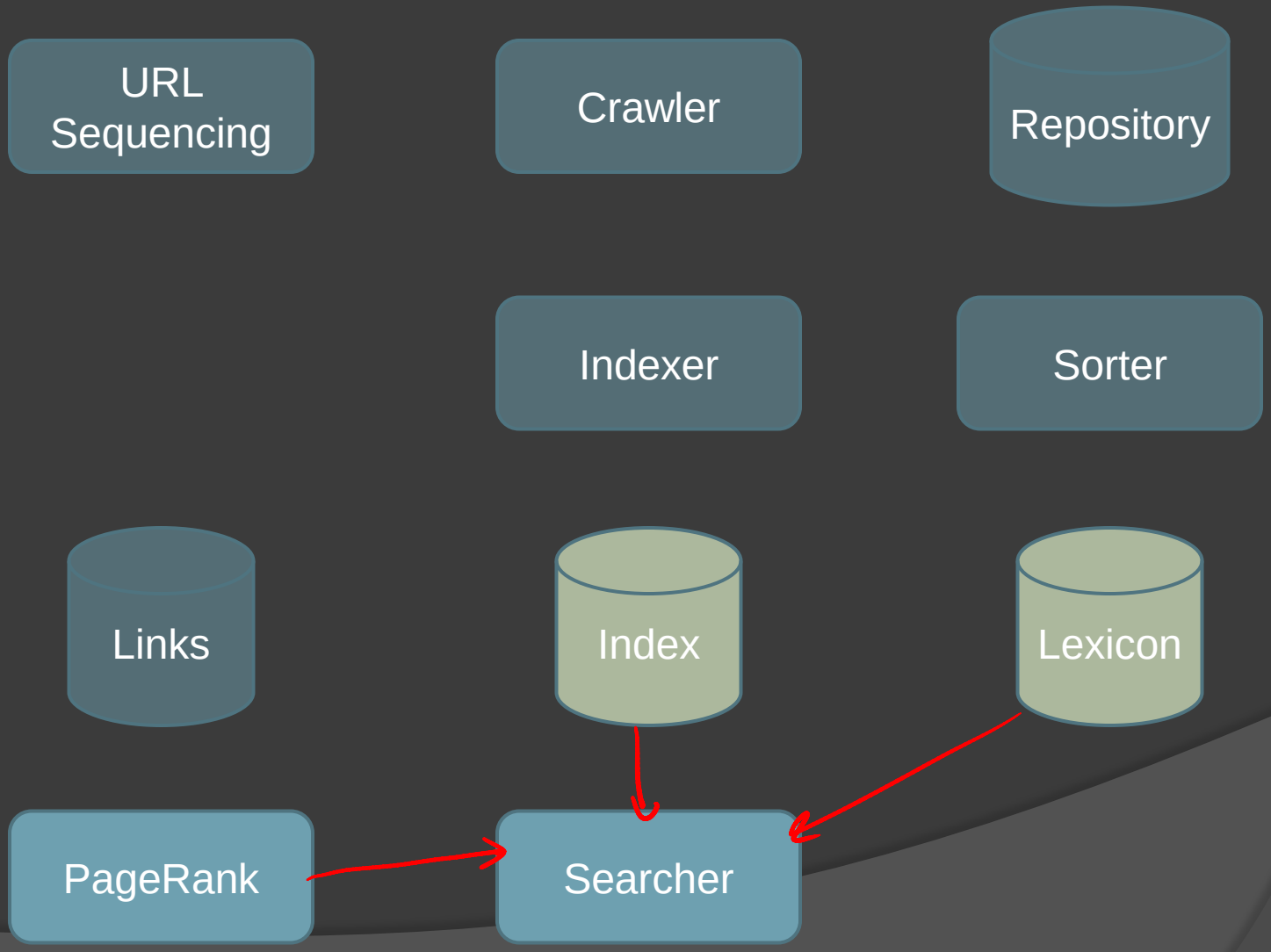
Invert Index



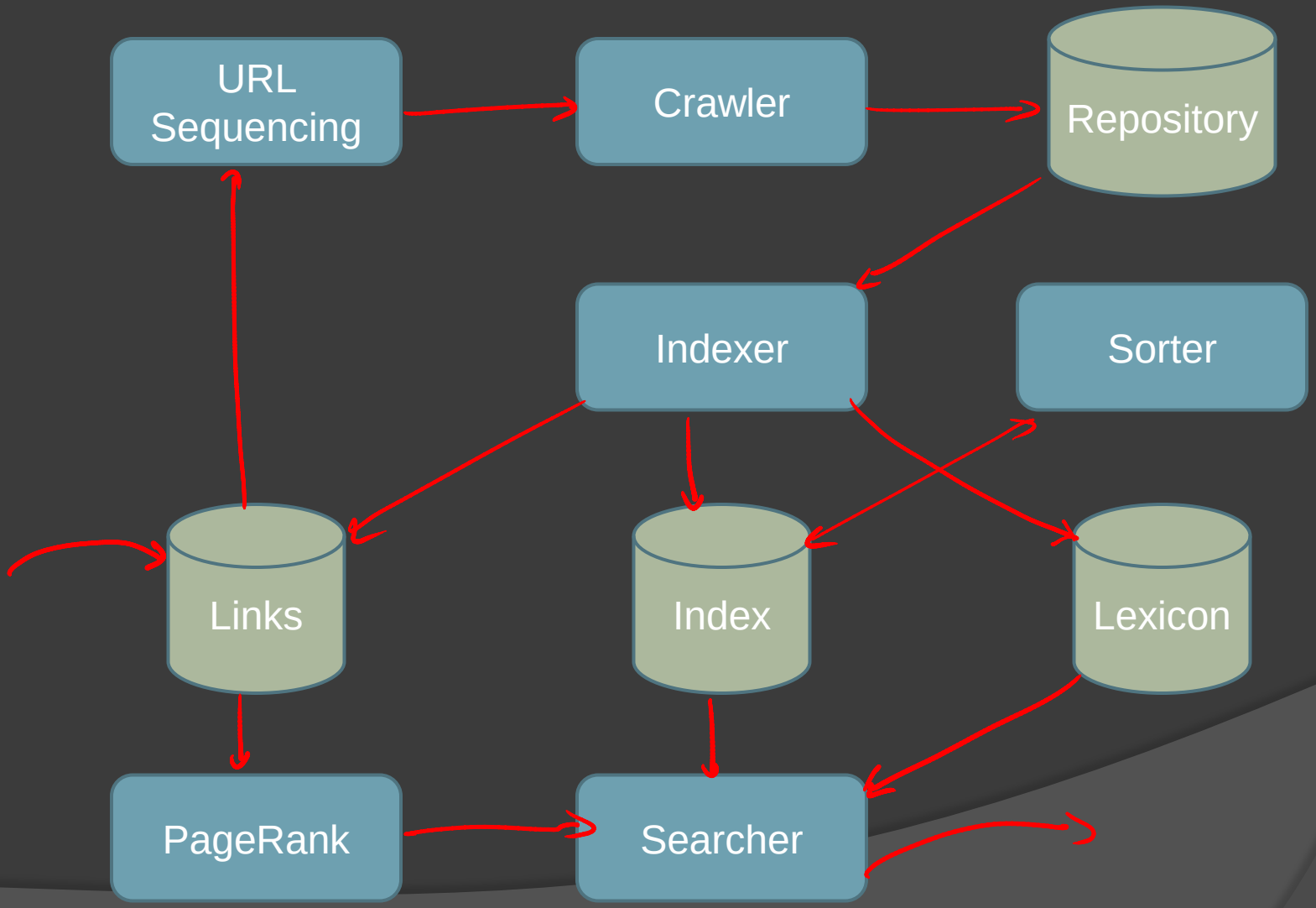
PageRank


URL Sequencing
Crawler
Repository
Indexer
Sorter
Links
Index
Lexicon
PageRank
Searcher

Ready to Search



Indexing Summary



Classic Google

Classic Google

- Academic version started at Stanford
- Documented in a few papers
- Included the PageRank algorithm
- Fun Fact!
  - Stanford University owns the PageRank patent and Google simply licenses it
  - Stanford was given shares of company in exchange for license, which were sold off for $330 Million

PageRank Algorithm

Classic Google

PageRank

- Basis of Google's original success
- Still used today

- Treat Internet as a graph
  - Page is node, Hyperlink is edge
- Use of back links not a new concept but specific scoring is

Key Ideas

- Internet has natural graph structure through hyperlinks
- Internet is full of garbage
  - Traditional keyword search performs poorly
- Link structure much harder to manipulate in large ways
- A ‘good’ page is one that is cited *often* by other ‘good’ pages.

Random Walk Model

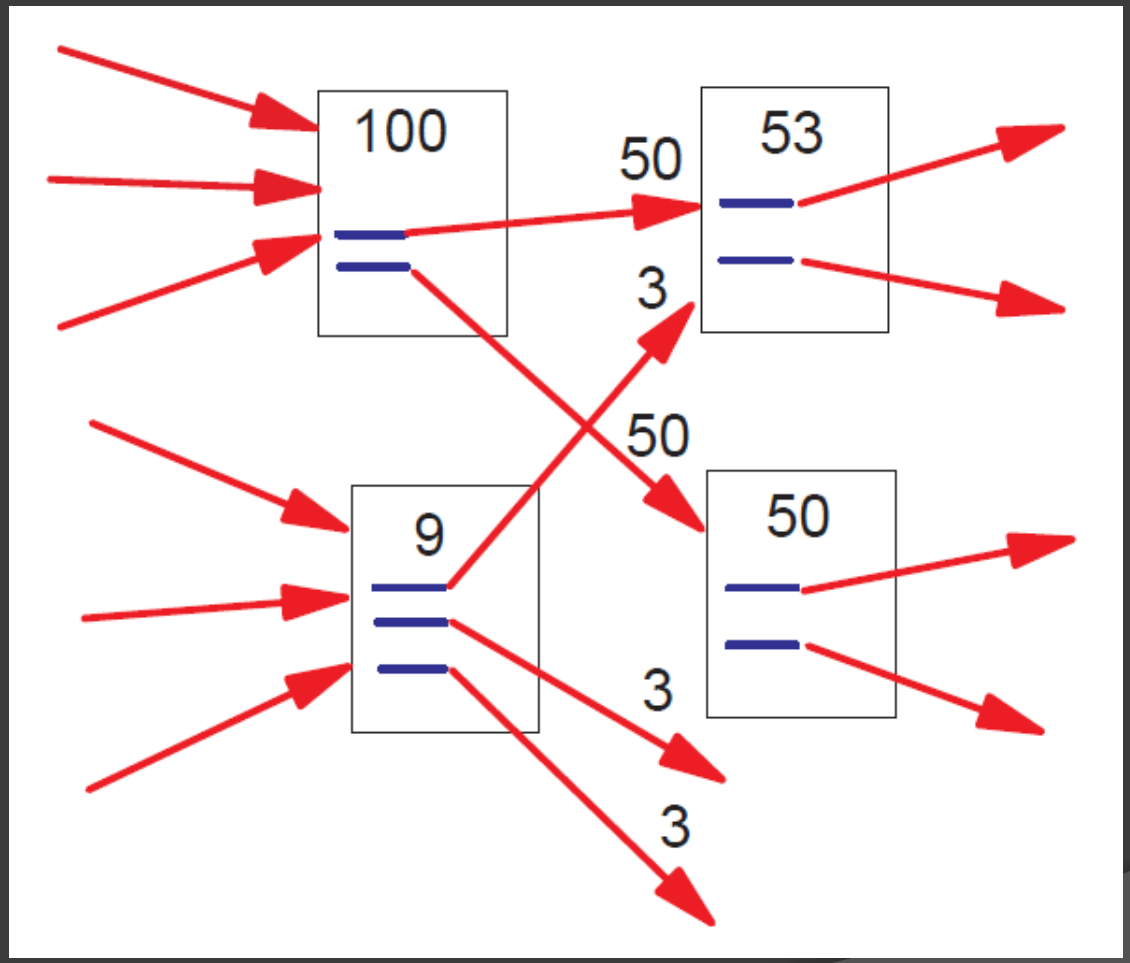
- PageRank can be understood as a random walk of the graph
  - Pages we end up at most often are the ones linked by other highly cited pages.
- The probability of ending up at a given page forms it's PageRank
- Ignore problems with cycles for now

PageRank Formula

$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v}$$

- *R(u)* is PageRank of page u
- B_u is set of incoming links
- N_v is the number of outgoing links
- *c* is a constant slightly smaller than one. Ignore for now.

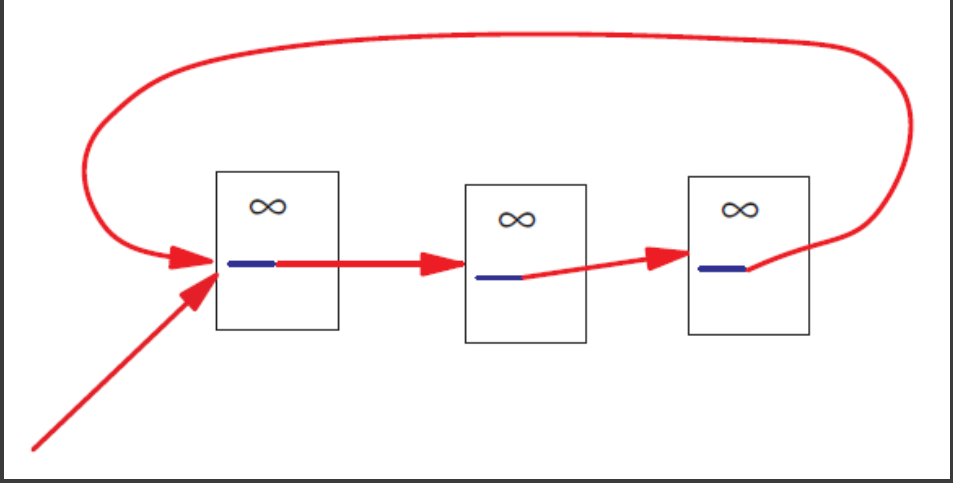
PageRank Example



PageRank as Eigenvector

- Consider the Adjacency Matrix
- Normalize values such that the sum of each column is 1
  - Thus each entry in column is 1/N
  - Some variants this is non-uniform!
- Compute Principal Eigenvector
- The resultant vector is PageRank values

No Outbound Links Problems



- If we have a cycle with no exits, PageRank cannot converge.
- Leaf nodes with no outgoing links can also be problematic.

The Solution

- Introduce a damping factor
- In random walk model, we say that N% of the time, a random page will be selected instead of a linked one

- Damping value E is uniform over pages

$$R'(u) = c \sum_{v \in B_u} \frac{R'(v)}{N_v} + cE(u)$$

PageRank (cont)

- Temporarily remove all leaf nodes until we let algorithm converge, and then reintroduce them and update their ranks.
- The value is computed iteratively until it converges within desired threshold

Personalized PageRank

- Proposed in PageRank paper, but not used by Google.
- If we set the damping function to be higher pages a user likes, the ranking will be biased to pages expanded from that source. This gives a personalized ranking system.

$$R'(u) = c \sum_{v \in B_u} \frac{R'(v)}{N_v} + cE(u)$$

Google Now

Google Now

- Google search has rapidly evolved, but a lot of the details developed since leaving Stanford remain secrets.
- Fundamentally the search is the same, but hundreds of new metrics for ranking, and heuristics for query processing have been developed.
- We can look at three big facets to see how things changed: Speed, Quality, Cost

Improving Speed

- As Google's dataset grows, they have evolved the way they manage certain big computation tasks.
- One of these tools is MapReduce, which the search indexer now runs on.

MapReduce

Google Now

What is MapReduce

- It is a software framework created by Google for doing data intensive computations (Petabytes) on huge clusters of machines (1000's).
- Inspired by the *map* and *reduce* functions common to functional programming.
- Operates on key-value pairs of strings.

Map

- Translates one list of key-value pairs into another list with appropriate keys.
- User provided function takes a single pair at a time and returns a list of zero or more key-value pairs.
- `list<k,v> user_map(k,v)`

Reduce

- Combines all values with a given key into a new list of values
- `list<v> user_reduce(k,list<v>)`

Example – Word Frequency

```
map(key, value):
    // key: document name
    // value: document contents
    for each word w in value:
        EmitIntermediate(w, 1);

reduce(key, list<values>):
    // key: a word
    // values: a list of counts
    int result = 0;
    for each v in values:
        result += v;
        Emit(result);
```

MapReduce Benefits

- If problem can be stated as map reduce, it is extremely parallel
- The framework provides mechanisms to distribute data as needed.
- The framework assigns jobs to computers automatically.
- The framework provides automatic fault detection and failover

Other Examples

- Distributed Grep
- Inverted Index
- Distributed Sort

Improving Quality

Google Now

Quality of Results

- There is constant work by Google to improve results with new metrics
- They have a wide variety of additional sources of data to utilized in creating results:
  - Links previous searches clicked
  - Adwords detects how often a page is viewed

Threats to Search Quality

- There is constant interest in manipulating Google to increase the presence of people's websites.
- Original PageRank was extremely effective against manipulation, but as things like botnets grow, manipulation attempts get bigger.
  - Ranking algorithms attempt to detect link farming automatically and penalize results

Reducing Cost

Google Now

Costs Reduction Measures

- Some estimate that Google has over 450 000 servers under their control
- Prefer multi-core over high clock
- Prefer cheap hard drives since they will fail anyways and their systems handle it
- Consumer-grade Intel processors
- Developing own power supplies with only 12 volt rails to increase efficiency

Related Topics

Related Topics

- Google File System
  - Distributed, redundant file system optimized for huge files
- BigTable
  - Google’s take on databases
  - Powers most MapReduce tasks

Advertising and Mixed Motives

“Currently, the predominant business model for commercial search engines is advertising. The goals of the advertising business model do not always correspond to providing quality search to users. ...

... In general, it could be argued from the consumer point of view that the better the search engine is, the fewer advertisements will be needed for the consumer to find what they want. This of course erodes the advertising supported business model of the existing search engines. However, there will always be money from advertisers who want a customer to switch products, or have something that is genuinely new. But we believe the issue of advertising causes enough mixed incentives that it is crucial to have a competitive search engine that is transparent and in the academic realm.”

The Anatomy of a Large-Scale Hypertextual Web Search Engine,

Sergey Brin and Lawrence Page

Questions?