“Catch me if you can”

Temporal Coherence of Web Archives

Marc Spaniol
Dimitar Denev, Arturas Mazeika, and Gerhard Weikum
Ârhus, September 18, 2008
Agenda

• Motivation

• Problem statement

• Measuring crawl coherence

• Discovering incoherence in web archives/ing
  - Online: At time of capture
  - Offline: From the archive (between captures)

• Experimental results

• Visualization by example

• Conclusions

• Outlook
Motivation

Databases and Information Systems
Prof. Dr. G. Weikum
MPII-Sp-0908-3/16
Problem Statement

• Crawler operations
  - Visit (pages)
    ▪ Extract (links from pages)
    ▪ Compare (versions of pages)
  - Follow (links)

• Website operations
  - Modifications “inside” pages
    ▪ Content (text)
    ▪ Structure (links)
  - Modifications “inside” site
    ▪ Page creation
    ▪ Page deletion

Taking place in parallel

Potentially incoherent
Potential Pitfalls in Web Archiving

- Crawling takes a long (!) time
  - Politeness
  - Multiple seeds per crawl
  - Spam farms

- Crawlers aren’t “really” smart
  - Highly volatile against dynamics in CMS
  - Easy to be trapped, if not exactly configured
  - Doesn’t recognize patterns of “identical” contents
    ⇒ Pre-analysis of site(s) needed

- Some examples of crawler behavior
  - Enjoy link generation from JavaScript, PHP, etc.
  - Tend to go for shopping
  - Like time travelling in calendars

⇒ Crawling is simply “unpredictable”
⇒ Crawlers need “constant” monitoring

Smart(er) Crawling Strategies

⇒ Archive Coherence in Danger!

Evaluation of Crawl Coherence
Measuring Crawl Coherence

A crawl graph is a set of \((p, [t_1, t_2], \mathcal{B}_C, \mathcal{B}_S)\) nodes \(n\), where:

- \(p\) is a web document
- \(t_1\) is the time of retrieving this document
- \(t_2\) is the time of invariance check for this document
- \(\mathcal{B}_C\) is defined as:
  - \(1\), if \(p.d[t_1] \sim p.d[t_2] \land p.d[t_i] \neq "404", i \in \{1,2\}\)
  - \(0\), otherwise
- \(\mathcal{B}_S\) is defined as:
  - \(1\), if sets \(p.H[t_1] \sim p.H[t_2] \land p.d[t_i] \neq "404", i \in \{1,2\}\)
  - \(0\), otherwise

We require that the crawl graph contains a non-empty time interval:

\[
\bigcap_{n \in N} [t_1, t_2] \neq \emptyset
\]
Measuring Crawl Coherence by Example

\[ t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8 \]

\[ n_A, n_B, n_C, n_D \]
Discovering Incoherence

• “Easy” for offline coherence analysis
  - Data is stored in the archive
  - Efficiency “unproblematic”

• “Difficult” for online coherence analysis
  - Data needs to be “tapped” from the crawler
  - Efficiency is a key issue

• Proper dating of page versions

• Multistage change measurement procedure
  1) Check HTTP timestamp
  2) Check content timestamp
  3) Compare a hash of the page with a stored hash
  4) Non-significant differences (ads, fortunes, request timestamp)
    - only hash text content, or “useful” text content
    - compare distribution of n-grams (shingling)
    - compute edit distance with previous version
Online Coherence Analysis

- **Do revisit** [yes]
- **revisit** [no]
- **revisit** [yes]
- **Store fetched as revisit** [no]
- **Store fetched as visit** [yes]
- **Fetch Content** [yes]
- **revisit** [no]
- **changed** [yes]
- **Fetch HTTP Headers** [no]
- **More URLs** [yes]
- **More URLs** [no]
- **Create Revisit** [yes]
- **Create Revisit** [no]
- **Load seeds from previous crawl** [yes]
- **Create Crawl** [no]
- **revisit** [no]
Experimental Results

- Crawling with Heritrix
  - (Meta-)Data extraction embedded
  - Revisiting strategy integrated

- Experimental crawls on the mpi-inf.mpg.de domain (so far)
  - ~ 65,000 contents
  - > 25 GB
  - ~ 4.5 hours
  - ~ 6 documents per second

- Online coherence analysis
  - Incoherence < 3‰
    - ~ 1‰ dynamics in dns look-ups
    - ~ 1‰ dynamically created wiki pages
    - ~ 1‰ dynamically created pages by bioinf.mpi-inf.mpg.de CMS
    - < 5 pages with relevant changes
  - Largest coherent sub graph: ~20,000 contents
Post Processing via GraphML

- File format for graphs
  - Core for the description of structural properties
  - Extension mechanisms for application-specific data

- Main features include
  - Directed, undirected, and mixed graphs
  - Hypergraphs
  - Hierarchical graphs
  - Graphical representations
  - References to external data
  - Application-specific attribute data
  - Light-weight parsers

- Based on XML

- Applied in many graph related software applications
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns/graphml"
          xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns/graphml
          http://www.yworks.com/xml/schema/graphml/1.0/ygraphml.xsd">
  ...
  <graph edgedefault="directed" id="G229">
    <node id="http://www.mpi-inf.mpg.de/index.html">
      <data key="d0">
        <y:ShapeNode>
          <y:Geometry width="10.003" height="10.003"/>
          <y:Fill color="#00FF00" transparent="false"/>
          <y:Shape type="ellipse"/>
        </y:ShapeNode>
      </data>
      <data key="d1">http://www.mpi-inf.mpg.de/index.html OK</data>
    </node>
    ...
    ...
  </graph>
</graphml>
Visualization of mpi-inf.mpg.de
Online Coherence
Conclusions

- Smart(er) revisiting strategy
  - Speeds up crawling
  - Reduces network and server load

- Incoherence of crawl coherence is
  - Quantifiable at time of capture
  - Visualizable in a subsequent post processing step

- Visualization helps crawl engineers to
  - Identify critical (sub-)graphs
  - Adjust future crawls
    - Strategy
    - Frequency
  - Understand the nature of incoherence
Outlook

- Development of an “incoherence cost model”
  - Content
  - Link structure
  - Status code
  - …

- Identification of CMS related changes
  - Application of shingling
  - Disregarding of non-relevant changes

- Analysis of different re-visiting strategies

- Crawling and comparison of additional domains
  - News
  - Sports
  - …

- Identification of change patterns
The End!