

# Intelligent and Adaptive Crawling of Web Applications for Web Archiving

{muhammad.faheem,pierre.senellart}@telecom-paristech.fr

Télécom ParisTech (<http://dbweb.enst.fr>)

BDA

23 October 2013

# Web Archiving



## Archiving the Social Web



## Archiving the Social Web

- ▶ Traditional crawling approach crawls the web sites independently of the nature of the sites and their content management system.
- ▶ **Goal:** Smart archiving of the Social Web;
  - Intelligent Crawling
  - Indexing Web objects

# Agenda

Traditional Crawling Approach

Application-Aware Helper

Methodology

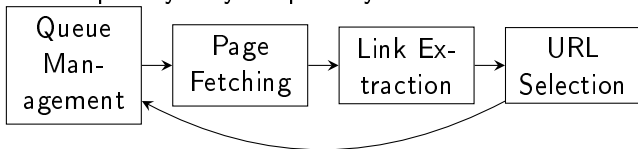
Web Application Adaptation

Experiments

Future Work

## Traditional Crawling Approach

- ▶ A traditional Web crawler (such as **Heritrix**) crawls the Web in a conceptually very simple way.



- ▶ This approach does not take into account the nature of the Web application.

## Introduction to Application-Aware Helper

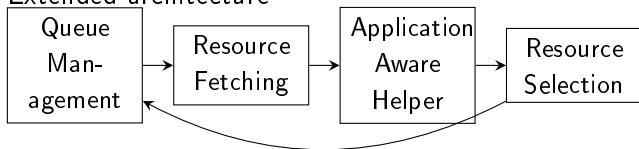
- ▶ Different crawling techniques for different social Web sites.
- ▶ Detect the type of Web application, kind of Web pages inside this Web application, and decide crawling actions accordingly.
- ▶ Our approach does not have the same purpose as *focused* crawling.

**Focused Crawling:** crawling based on a Topic.

**Application-Aware Helper:** crawling **optimized** for a particular Web Application.

## Introduction to Application-Aware Helper

- ▶ Extended architecture



- ▶ To be implemented in 2 Web crawlers: Internet Memory Foundation crawler, and into Heritrix.



## Knowledge base of Web applications

- ▶ Knowledge base of Web applications: describes how to crawl a Web site in an intelligent manner.
- ▶ Hierarchy: from general categorizations to specific instances (Web sites) of this Web application.
  - categories the web applications.
  - specifies the detection rules.
  - describes the specific crawling actions.

## Knowledge base of Web applications

- ▶ Different crawling actions for different kinds of Web pages under a specific Web application.
- ▶ Declarative, XML-based format.

## Web application detection Module

- ▶ One main challenge in intelligent crawling and content extraction is to identify the Web application and then perform the **best crawling strategy** accordingly.
- ▶ Detecting Web application using:
  - URL patterns,
  - HTTP metadata,
  - textual content,
  - XPath patterns, etc.
- ▶ For instance the **vBulletin** Web forum content management system, that can be identified by searching for a reference to a `vbulletin_global.js` JavaScript script by using a simple `//script/@src` XPath expression.

## Crawling and extraction

- ▶ **Next stage:** determining the corresponding crawling actions.
- ▶ **Crawling action:** not just a list of URLs; can be any action that uses **REST API**, complicated interaction with **AJAX-based** application, and extracts semantic **Web objects**.

## Crawling and extraction

- ▶ More specifically, crawling actions are of two kinds:
  - Navigation actions:** to **navigate** to another Web page or Web resource.
  - Extraction actions:** to **extract** individual semantic objects from Web pages (e.g., timestamp, the blog post, the comments).

## Adaptation to template change

- ▶ Two types of changes can occur in a Web page: **Web content** changes, and **Web structure** changes.
- ▶ It is complicated to adapt crawling action when a change occurs in a Web page structure.
- ▶ The AAH aims at determining when a change has occurred and adapting patterns and actions.
- ▶ The AAH deals with two different cases of adaptation: first, when a **recrawl** of Web application is carried out after template change; second, when a **new** Web application can be crawled with existing actions after slight adaptation.

## Recrawl of a Web Application

- ▶ The structural changes are detected by looking for the content in the archive.
- ▶ In the presence of structural changes, the system first marks the failed crawling actions and then align them according to structural changes.

## Crawl of new Web Application

- ▶ The WA type is detected but WA level or crawling actions do not work.
- ▶ For aligning WA level or crawling actions; the system collects all the candidate attributes, values, tag names from the knowledge base and then create all possible combinations of **relaxed expressions**.



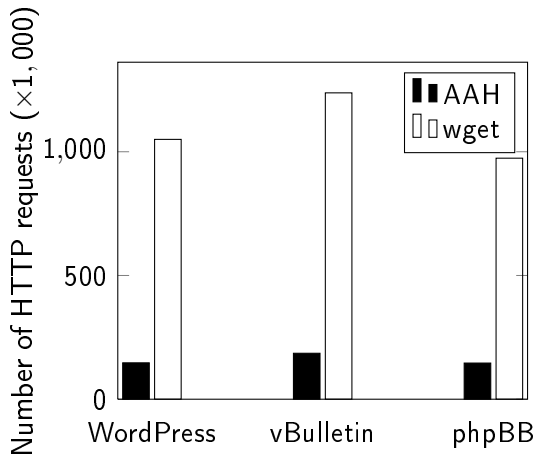
## Experiment setup

- ▶ The experiments are performed with both **AAH** and **GNU wget**.
- ▶ Crawled **100** WAs (totaling nearly **3.3** million Web pages) of two types of social Web sites (Web forum and blog), for **three** CMSs (vBulletin, phpBB, WordPress).
- ▶ The WA were randomly selected from three different sources:
  - <http://rankings.big-boards.com/>**, a database of popular Web forums.
  - A dataset related to the **European financial crisis**.
  - A dataset related to the **Rock am Ring music festival** in Germany.

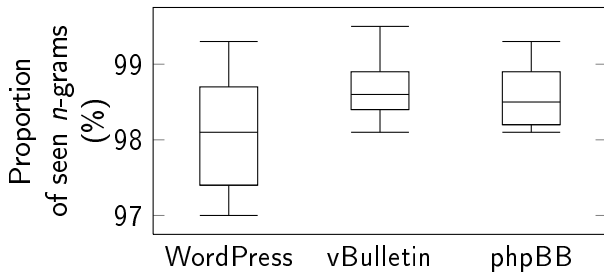
## Performance metrics

- ▶ The number of HTTP requests made by both systems vs the amount of *useful* content retrieved.
- ▶ Coverage of useful content is calculated by comparing the proportion of **2-grams** in the crawl result of both systems for every WA and by counting the number of **external links**.

## Crawl efficiency



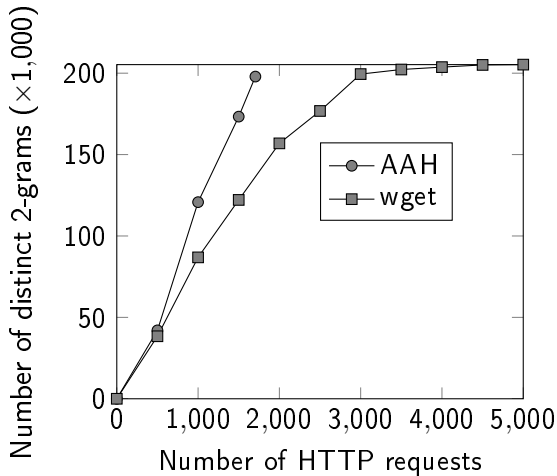
## Crawl effectiveness



## Crawl effectiveness

<b>CMS</b>	<b>External links</b>	<b>External links (w/o boilerplate)</b>
WordPress	92.7%	99.8%
vBulletin	90.5%	99.5%
phpBB	92.1%	99.6%

## Crawl effectiveness



## Adaptation performance

- ▶ Among the 100 WAs, the 77 did not require any adaptation.
- ▶ Remaining 23 had a structure that did not match the crawling actions in knowledge base.
- ▶ Most of the adaptation consisted in relaxing class or id attribute rather than replacing the tag name of an element.
- ▶ When there was tag name change then it was mostly div to span to article or vice versa.

## Future Work

- ▶ Automatic, possibly unsupervised, learning of new Web applications, either by involving human interactions, or using semi supervised machine learning techniques.
- ▶ Integrating the XPath to crawl complex Web applications by making use of AJAX or Web forms.



Merci

## Grammar for AAH

```
<expr> ::= <step> | <step> "/" <expr>
          <step> "//" <expr>
<step> ::= <nodetest> | <step> "[" <predicate> "]"
<nodetest> ::= tag | "@" tag | "*" | "@*" | "text()"
<predicate> ::= "contains(" <value> ", " string ")" |
              <value> "=" string | integer | "last()"
<value> ::= tag | "@" tag
```

## Example of the knowledge base

```
<knowledgebase>
  <cms name="vBulletin" type="webforum">
    <detection-rules>
      <xpath-expression>
        //script/@src[contains(.,'vbulletin_global.js')]
      </xpath-expression>
    </detection-rules>
    <page-level-cat>
      <list-of-forum>
        <detection-rules>
          <xpath-expression type="1">
            //a[@class="forum"]/@href
          </xpath-expression>
          <xpath-expression type="2">
            //h2[@class="forumtitle"]/a/@href
          </xpath-expression>
        </detection-rules>
      </list-of-forum>
    </page-level-cat>
  </cms>
</knowledgebase>
```

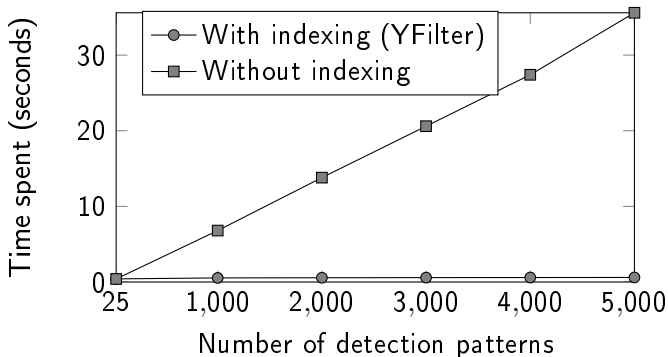
## Example of the knowledge base

```
< crawling-action >
  < action id="1" >
    //a.forum/@href
  < /action >
  < action id="2" >
    //td.forumtitle/div/a/@href
  < /action >
< /crawling-action >
< /list-of-forum >
< list-of-thread >
  .
< /list-of-thread >
< thread >
  .
< /thread >
< /knowledgebase >
```

## Indexing detection patterns

- ▶ The number of of detection patterns for detecting Web application type and level grows with the addition of knowledge about new Web Application.
- ▶ We integrated the AAH with the **YFilter** system (an NFA-based filtering system for XPath expressions) with some slight changes, for efficient indexing.
- ▶ In our integrated version of YFilter, the detection patterns will be submitted as queries. When a document satisfies a query, the system processing the document against all remaining queries (in contrast to standard behaviour of YFilter).

## Efficiency of detection patterns



## Comparison to iRobot

- ▶ The **iRobot** system assists the extraction process by providing the sitemap of the Web application being crawled.
- ▶ The **iRobot** system has considered **50,000** Web pages over **10** different Web forums.
- ▶ The completeness of content of the **AAH** is over **99 percent** as compared to **93 percent** of **iRobot**.
- ▶ The number of HTTP requests for **iRobot** is claimed to be **1.73** less than a regular crawler, whereas **AAH** makes **10 times** fewer requests than wget.