Learning to Discover Domain-Specific Web Content
Kien Pham, Aecio Santos and Juliana Freire

The **Goal** is to find all pages relevant to a given domain **continuously** and in a **timely** fashion.

- Tracking adult ads for human trafficking investigation.
  - 105,000 children in the U.S are being sexually exploited.
  - 70% of child sex trafficking survivors were at some point sold online.
  - 100,000 new escort ads are posted online everyday.

→ Information on the Web can be used to better understand the problem, identify victims and perpetrators.

- Monitoring secondary data to improve humanitarian crisis analysis
- Tracking online wildlife market places to identify illegal products and prevent wildlife trafficker.
- Surveilling online markets where weapons and drugs are sold to generate investigation leads.
- And more.

### Limitations of Search Engines

- Do not prioritize unpopular content
- Do not support analytics

**However** (not being search engines)

- Limited computing resource
- Restricted crawling rate

### Related Works

- Requires fully crawled snapshots
- Unable to incorporate new features

### Proposed Solution

We propose an iterative re-crawling framework that predicts the number of new relevant pages a seed page will yield using data from the previous crawls. Based on the prediction, it select the best k seed pages to be re-crawled.

#### Ranking

**Regression (REG):**

\[
\text{score}(s) = \log(O(s) + 1)
\]

**Learning to rank (LTR):**

The training set is denoted as \( \{(s, t), y_i\}_{i=1}^T \), where \( S_t \) list of seed pages selected to crawl at timestamp \( t \), \( y_i \) list of scores of seed pages in \( S_t \)

\[
\text{score}(s) = \log(O(s) + 1)
\]

**Featuarization**

A pair \((s, t)\) is featuarized by the following functions:

- **avg:** Average number of new pages discovered from s
- **std:** Standard deviation of the number of new pages discovered from s
- **age:** Time since s was last crawled.
- **tod:** Time of the day at timestamp \( t \)
- **dow:** Day of the week at timestamp \( t \)
- **aa:** Age of pages discovered by the best method

#### Overlap removal

If a set of seed pages has high overlap, the selector only picks one of them to crawl.Overlap is computed between pair of seed pages that are crawled at the same timestamp.

\[
\text{Overlap}(s_i, s_j) = \frac{O(s_i) \cap O(s_j)}{O(s_i) \cup O(s_j)}
\]

**Diversification**

We use UCB1 algorithm (BANDIT) to balance trade-off between exploitation and exploration. An arm in the algorithm is modeled as a ratio between a greedy method and a random based method.

\[
\text{UCB}_{i,t} = \mu_{i,t} + \frac{\ln(t)}{n_{i,t}}
\]

### Problem Definition

Given a set of seed pages \( S \) and we want to select the top-k pages \( S_k^t \) for every timestamp \( t \), where \( S_k^t \subseteq S \) and \( |S_k^t| \ll |S| \), such that re-crawling every \( S_k^t \) at a timestamp \( t \) maximizes the number of new relevant pages discovered.

**Relevant Pages:** pages that are relevant to the domain of interest

**Seed pages:** pages that contain links to relevant pages

**Coverage:**

\[
\text{Coverage} = \frac{\# \text{Discovered Pages}}{\# \text{Total Pages}}
\]

### Experimental Evaluation

We gather data from the 3 domains: Human Trafficking, Humanitarian Crisis and Politics. For each domain, we crawl all seed pages and out-links hourly during a 1-month period.

**Proposed methods:** REG (Linear Regression), REG-RR, LTR, BANDIT (UCB1)

**Baselines** [1]: OD-WIN, CLIQ-WIN*, COV*, RR, GREEDY

### References


**Acknowledgements:** This work was supported by the DARPA MEMEX and D3M programs.