First-order Focused Crawling

Qingyang Xu
College of Computer Science and Technology
Jilin University
Changchun 130012, P.R.China
xuqy@jlu.edu.cn

Wanli Zuo
College of Computer Science and Technology
Jilin University
Changchun 130012, P.R.China
wanli@jlu.edu.cn

ABSTRACT

This paper reports a new general framework of focused web crawling based on "relational subgroup discovery". Predicates are used explicitly to represent the relevance clues of those unvisited pages in the crawl frontier, and then first-order classification rules are induced using *subgroup discovery* technique. The learned relational rules with sufficient *support* and *confidence* will guide the crawling process afterwards. We present the many interesting features of our proposed first-order focused crawler, together with preliminary promising experimental results.

Categories and Subject Descriptors: H.5.4 [Information interfaces and presentation]: Hypertext/hypermedia; I.2.6 [Artificial intelligence]: Learning

General Terms: Algorithms, performance, measurements

Keywords: Focused crawling, Relational subgroup discovery

1. INTRODUCTION

While crawling the World Wide Web, a focused web crawler [2, 3, 1] aims to collect as many relevant web pages with respect to some predefined topic(s) and as few irrelevant ones as possible. To warrant the claimed focusing capability, a successful focused crawler has to predict precisely a web page's relevance *before* downloading it. However, the decision relies exclusively upon diverse indirect subtle relevance clues, which are ubiquitous but noisy, and extremely difficult to be exploited by traditional machine learning approaches. As a result, even the state of the art focused crawlers still waste significant amount of network and local resources downloading irrelevant pages, just to discard them afterwards.

In this paper, we approach the goal of focused crawling from a relational learning perspective. Our approach was motivated by the fact that all the relevance clues of an unvisited page are relational in nature: a hyperlink relates the page in question with an anchor occurring within a downloaded page, and the anchor itself has further structural relationships with other HTML elements in its link context, etc. We use straightforward predicates to model such relationships in an elegant and flexible way, and then feed the background knowledge to a relational learner to induce classification rules based on first-order logic, which lends itself to the focused crawling context for its great expressing power. For example, the following first-order rules succinctly represent the two underlying hypotheses in the originally proposed focused crawler [2]. Rather than ad-hoc heuristics, these rules can be discovered by our relational learner automatically.

- `relevant(X) :- links_to(Y, X), relevant(Y).`
- `relevant(X) :- links_to(Y, X), links_to(Y, Z), Z != X, relevant(Z).`

2. OUR APPROACH

Our proposed first-order focused crawler has the following two characteristics different from all the other approaches ever proposed:

Relational knowledge representation: We use predicates to represent the heterogeneous relevance clues within a consistent framework in an elegant way. The following predicates are utilized to represent an unvisited page X's background knowledge available to a focused crawler:

- **links_to**(Y, X, A): denotes that web page Y links to page X through an anchor element A
- **parent**(E_1, E_2): denotes that element E_1's parent is element E_2
- **tag**(E, t): denotes that element E's tag is t
- **text_has**(E, t): denotes that a text token t occurs in one of the child text elements of element E
- **url_has**(A, t): denotes that an anchor element A's *href* attribute contains text token t
- **target**(A): denotes that anchor element A's target page has been downloaded and classified as relevant by a web page classifier

For example, the relevance clues of page v in Figure 1 can be represented as following:

```
links_to(u, v, e8)   text_has(e8, hamlet)  parent(e8, e7)
tag(e7, li)          parent(e7, e3)        tag(e3, ul)
parent(e6, e3)       tag(e6, li)           text_has(e6, macbeth)
parent(e5, e3)       tag(e5, li)           text_has(e5, othello)
parent(e4, e3)       tag(e4, li)           text_has(e4, king)
text_has(e4, lear)   parent(e3, e1)        tag(e1, body)
parent(e2, e1)       text_has(e2, four)    text_has(e2, tragedies)
tag(e2, p)
```

Suppose v is downloaded and proved to be relevant by some preordained web page classifier, we denote such instance label information in another predicate: relevant(v). Note that our knowledge presentation scheme can accommodate new sources of background knowledge easily by inventing new predicates. For example, the information in HTTP response headers can be exploited with little effort.

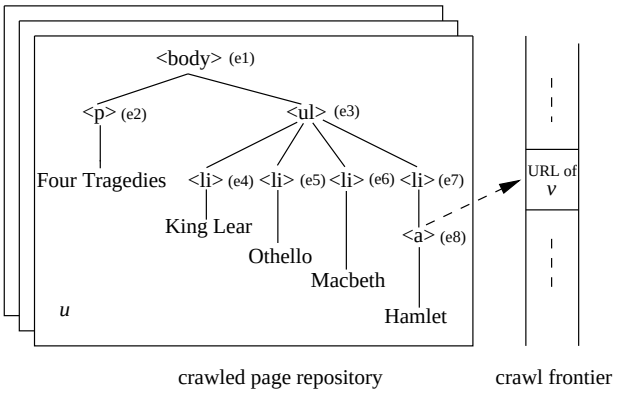

Figure 1: page v with its relevance clues from u

Subgroup discovery: After enough labeled instances with their background knowledge accumulate, we feed these facts to a relational learning algorithm to discover first-order rules to guide the crawling process afterwards. We first explored the traditional *sequential covering* algorithms (e.g. FOIL), but got disappointing results. Only the first few induced rules are statistically reliable and the resulting decision list falls prey to *overfitting*. In contrast, we met unexpected success in another relational learning approach based on *subgroup discovery* [4], which is a minor branch of data mining discipline that concerns itself with discovering instance subgroups which have unusual distribution in terms of interested property. In the focused crawling context, we use subgroup discovery technique to discover rules that cover subgroups containing many relevant pages and very few irrelevant ones, or using the terminology of *association rules*, with high *support* and *confidence*. Below are some rules induced from a focused crawling session with the topic of "Shakespeare" (lowercase symbols denote constants while capitalized symbols denote variables), together with their coverage of relevant and irrelevant instances during the learning phase and testing phase, respectively:

- `relevant(X) :- links_to(Y, X, A), url_has(A, shakespeare).`
  `learning phase: ⊕ 1694 ⊖ 47`
  `testing phase: ⊕ 5583 ⊖ 307`
- `relevant(X) :- links_to(Y, X, A), text_has(A, ii).`
  `learning phase: ⊕ 254 ⊖ 13`
  `testing phase: ⊕ 1408 ⊖ 34`
- `relevant(X) :- links_to(Y, X, A), parent(A, B), tag(B, p), text_has(B, scene).`
  `learning phase: ⊕ 215 ⊖ 16`
  `testing phase: ⊕ 230 ⊖ 7`
- `relevant(X) :- links_to(Y, X, A), parent(A, B), tag(B, li), parent(B, C), tag(C, ul), parent(D, C), tag(D, li), D != B, parent(E, D), tag(E, a), target(E).`
  `learning phase: ⊕ 230 ⊖ 4`
  `testing phase: ⊕ 1871 ⊖ 49`

Note that the above rules must be interpreted in the context of a specific focused crawling session. For example, the second rule above denotes that page X will be relevant if page Y has an anchor A pointing to it, and A's anchor text contains token "ii". At first sight, this rule seems arcane and brittle, but if you take the topic into account, some explanation will emerge. Usually, page Y is relevant, and the roman numeral "II" is a rather common token in the anchor text describing Shakespeare's drama, either in a play's name (e.g. Richard II) or, more commonly, as the serial number of some division of a play (e.g. Act II, Scene II).

Topics	best-first	accelerated	first-order
Shakespeare	70.24%	78.12%	91.36%
TeX	45.92%	59.48%	79.61%
Python	76.31%	84.76%	86.81%
Iraq War	70.68%	80.26%	91.25%

Table 1: Harvest ratios for some topics from DMOZ

In our current implementation, we use an A* search algorithm to seek those rules with sufficient support and confidence, and use the support threshold to prune the search space. After the learning phase, the learned rules will guide the crawling process afterwards. Using an embedded Prolog deduction engine, we pick out the URLs of those unvisited pages satisfying the rules and give them high downloading priorities.

3. EXPERIMENTAL RESULTS

In our comparative experiments, we choose the best-first algorithm and the accelerated focused crawler [1] as two other alternatives. Web page classifiers based on SVM algorithm are trained beforehand for a few topics of DMOZ (`http://dmoz.org`). For each topic, we download 10,000 pages using the best-first algorithm. After the first-order and accelerated focused crawlers learn from these training instances, the three focused crawlers resume their crawling process to download another 10,000 pages to test their performance. We adopt the "harvest ratio" as the performance metrics, which is the ratio of relevant pages among all the downloaded ones. The experimental results in Table 1 show that first-order focused crawler is promising, though a larger-scale evaluation is obviously needed.

4. CONCLUSIONS

In this paper, we present a novel focused crawling framework based on relational subgroup discovery. Preliminary experimental evaluation shows the potential of our proposed first-order focused crawler. While promising, our approach poses many technical challenges as well. Can we find more rules using other sources of relational background knowledge? Can we use relational learning approach to find typical "crawling paths" as in [3]? Can we induce relational rules more efficiently and overcome overfitting more effectively? We plan to explore these issues in our future work.

5. ACKNOWLEDGEMENTS

This work is sponsored by the Nature Science Foundation of China under grant number 60373099.

6. REFERENCES

[1] Soumen Chakrabarti, Kunal Punera, and Mallela Subramanyam. Accelerated focused crawling through online relevance feedback. In *WWW*, pages 148–159, 2002.
[2] Soumen Chakrabarti, Martin van den Berg, and Byron Dom. Focused crawling: A new approach to topic-specific web resource discovery. *Computer Networks*, 31(11-16):1623–1640, 1999.
[3] Michelangelo Diligenti, Frans Coetzee, Steve Lawrence, C. Lee Giles, and Marco Gori. Focused crawling using context graphs. In *VLDB*, pages 527–534, 2000.
[4] Nada Lavrac. Subgroup discovery techniques and applications. In *PAKDD*, pages 2–14, 2005.