Zero-cost Labelling with Web Feeds for Weblog Data Extraction

George Gkotsis*, Karen Stepanyan, Alexandra Cristea, and Mike Joy

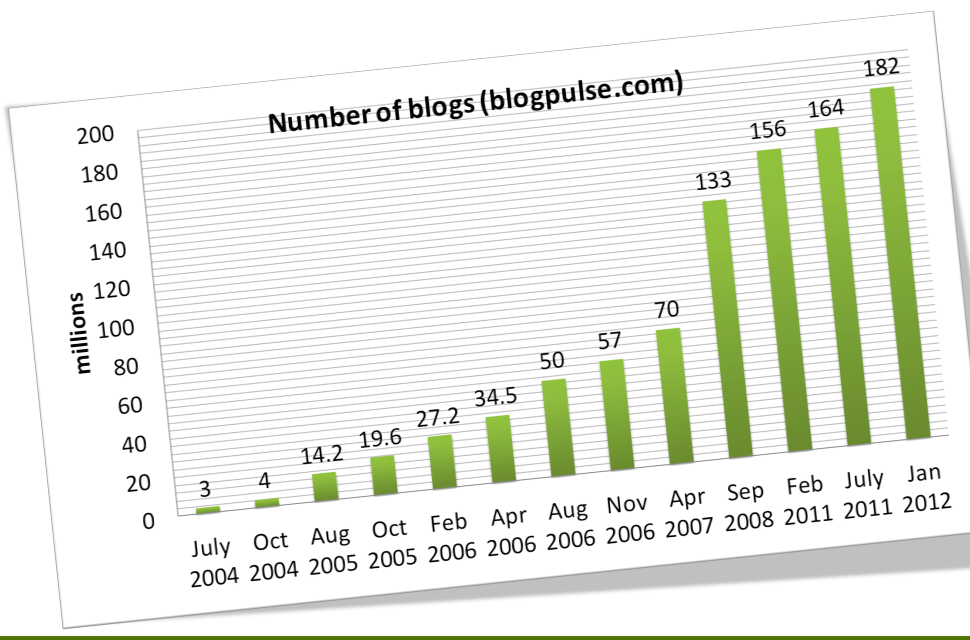
Dept. of Computer Science, University of Warwick, UK

Blogosphere

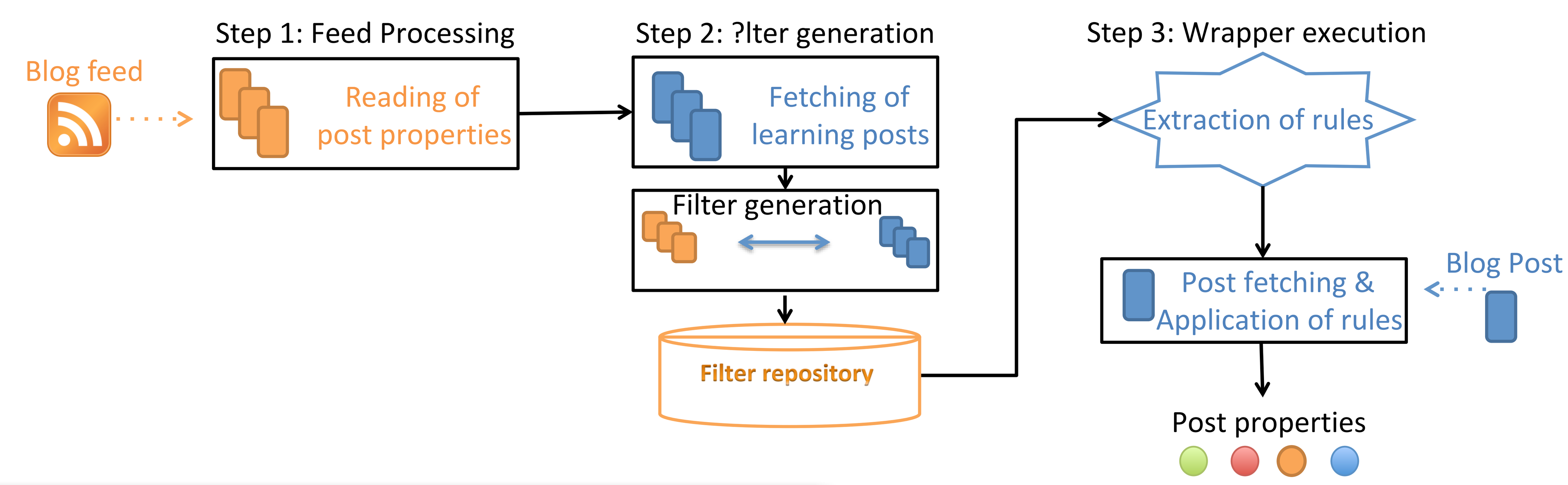
Volume
[2008] 900k posts/day
[2009] 133M English blogs in one month
[2011] 25% of Internet users in Britain have a blog

Diversity
- 1634 WordPress themes
- 469 different platforms

Web data extraction is a genuinely hard problem. The blogosphere, which constitutes a constituent part of the Web, remains bound to the limitations of modern data extraction. In general, data extraction is facing the trade-off between automation and accuracy/granularity.
The proposed model overcomes the above limitations by exploiting an inherent characteristic of weblogs: the Web Feed, commonly provided as RSS.



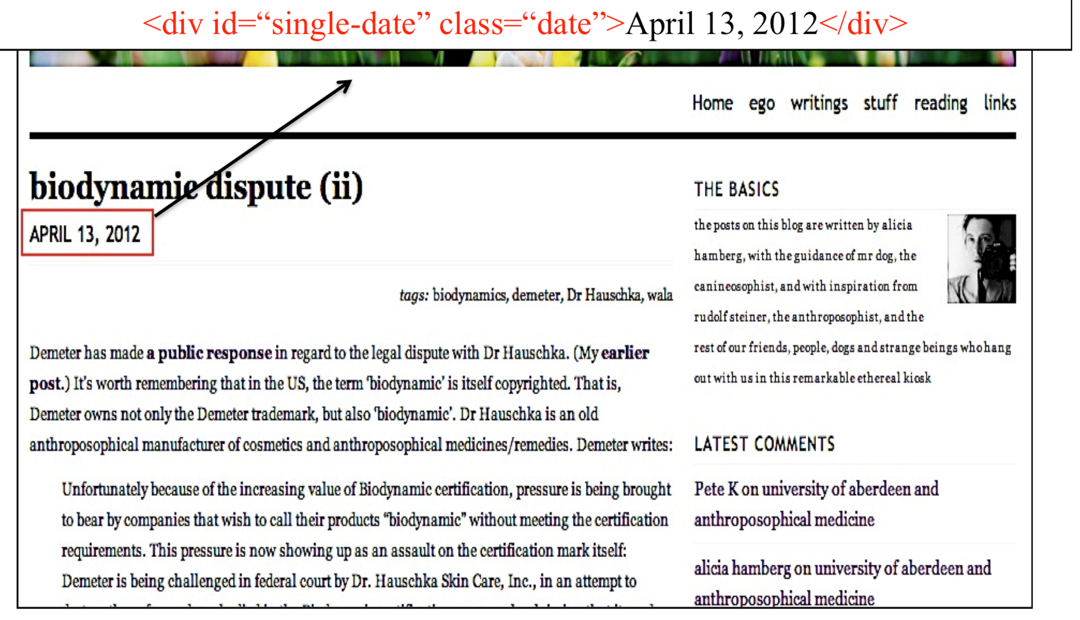
Proposed Methodology



Filter generation

- ✓ The filter is a structure that describes how to identify an HTML element.
- ✓ It is the result of the cross-matching of values between HTML documents and RSS entries

<div id="single-date" class="date">April 13, 2012</div>



Example of a post property

The **filter** is described using three basic attributes:
1. Absolute Path
2. CSS Classes
3. ID of the HTML element.

Once the HTML element is matched against its value, a filter is generated which describes it in these three attributes.

Example of a filter:

IDs	CSS Classes	Absolute Path
single-date	date	html[0]/body[1]/div[1]/div[1]/div[0]/div[0]/div[1]

Extraction of rules

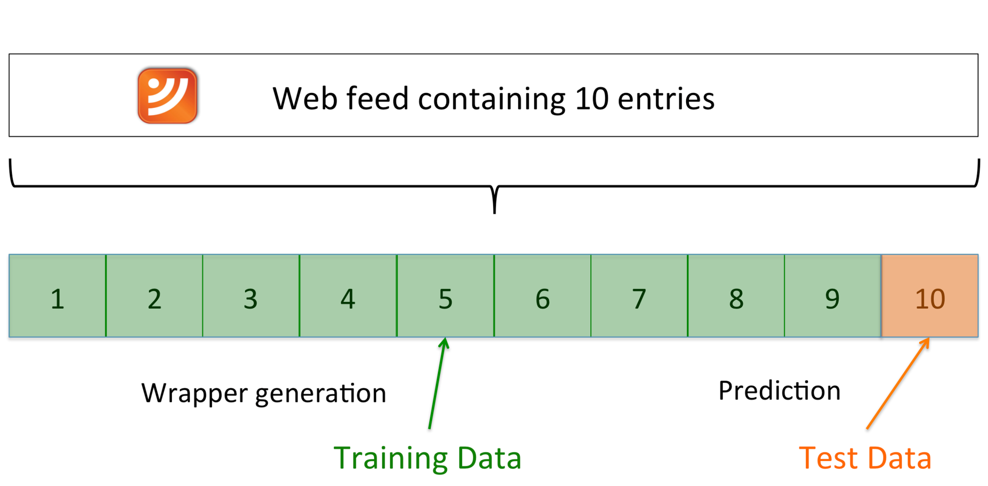
The last step transforms the filters into **rules**, in order to calculate the scores and select a rule for each of the desired properties. Essentially, a rule is the result of the transposition of a filter. This transposition can result in maximum three rules. Hence, a rule is described by its type (one of the three different attribute types of the filters), a value (the value of the corresponding filter's attribute) and a score, which is used to measure its expected accuracy. The need to calculate the score of each rule is justified by the inherent "noise" of the filters.

Rule Induction Algorithm

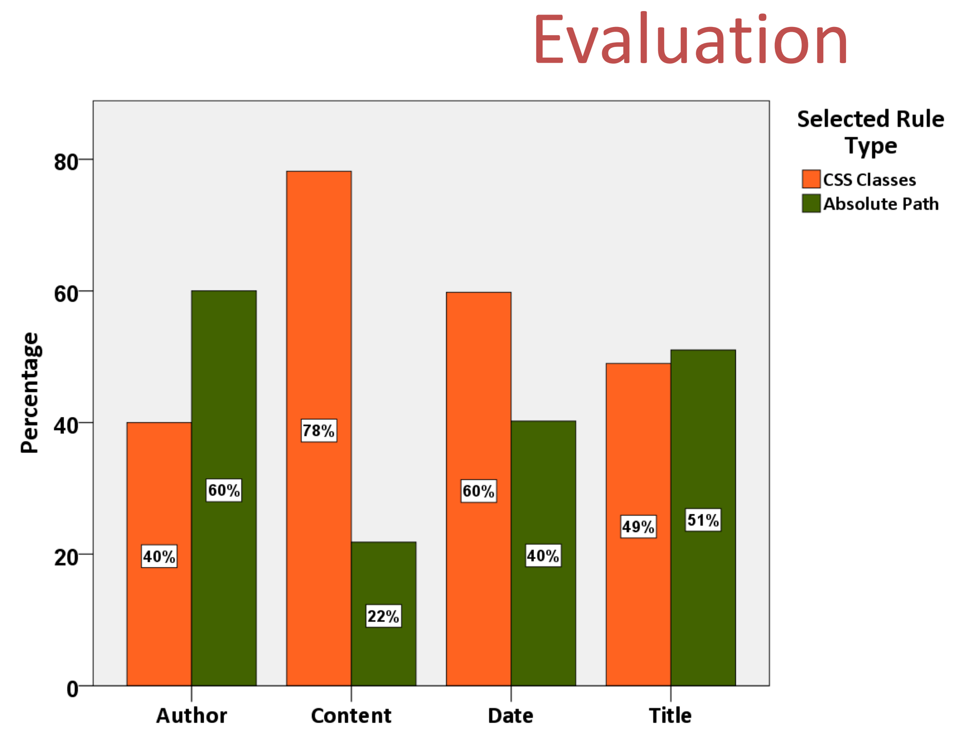
Data: Collection of training posts P, Collection of candidate rules R
Result: Rule with the highest score

```
/* Initialize all scores                                  */
forall the Rules r ∈ R do
    r.score = 0;
end
Rule rs = new Rule();
rs.score = 0;
forall the Rules r ∈ R do
    forall the Posts p ∈ P do
        /* Check if application r(p) of rule r, on
           post p succeeds                                */
        if r(p)==value-property of p then
            r.score++;
    end
    /* Normalize score values                             */
    r.score = r.score / |P|;
    /* Check if this is the best rule so far              */
    if r.score > rs.score then
        rs = r;
end
return rs;
```

Evaluation



We ran 10-fold validation across 240 feeds and 2.393 posts



The rules extracted show that the rule types vary for different weblog properties.

	Title	Content	Publication date	Author
Proposed Model	97,3% (65 misses)	95,9% (99 misses)	89,4% (253 misses)	85,4% (264 misses)
Boilerplate	0	77,4% (539 misses)	N/A	N/A

81,6% relative error reduction

A significant increase of prediction accuracy is found.

* gkotsis@gmail
gkotsis@twitter

This work was conducted as part of the BlogForever project funded by the European Commission Framework Programme 7 (FP7)

blog forever