Zero-cost Labelling with Web Feeds for Weblog Data Extraction

George Gkotsis, Karen Stepanyan, Alexandra I. Cristea, M. S. Joy
Department of Computer Science, University of Warwick, Coventry CV4 7AL, United Kingdom
{G.Gkotsis, K.Stepanyan, A.I.Cristea, M.S.Joy}@warwick.ac.uk

ABSTRACT

Data extraction from web pages often involves either human intervention for training a wrapper or a reduced level of granularity in the information acquired. Even though the study of social media has drawn the attention of researchers, weblogs remain a part of the web that cannot be harvested efficiently. In this paper, we propose a fully automated approach in generating a wrapper for weblogs, which exploits web feeds for cheap labelling of weblog properties. Instead of performing a pairwise comparison between posts, the model matches the values of the web feeds against their corresponding HTML elements retrieved from multiple weblog posts. It adopts a probabilistic approach for deriving a set of rules and automating the process of wrapper generation. Our evaluation shows that our approach is robust, accurate and efficient in handling different types of weblogs.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning – Induction

Keywords

data extraction, weblogs, wrapper induction

1. INTRODUCTION

The problem of web information extraction dates back to the early days of the web. Although exact numbers of weblogs are not known, it is evident that the size of the blogosphere is large. In 2008 alone Technorati reported to be tracking more than 112 million weblogs, with around 900 thousand blog posts added every 24 hours (`http://technorati.com/blogging/article/state-of-the-blogosphere-introduction/`). The volume of information published on weblogs justifies the attention of information retrieval, preservation and socio-historical research communities but is not the only challenge. Weblogs, due to their built-in personalisation options through plugins, themes and custom HTML code, exhibit large diversity and increase the complexity of generating a universal data extraction approach.

Our approach focuses on one of the most prominent characteristics of weblogs, the web feeds. Web feeds, commonly provided as RSS, are XML documents that allow access to the content of a website, such as a weblog, through a machine interpretable document. Until now, the feeds have been used as the *sole* sources of information and are therefore limited to a fixed number of entries, which is typically 10 [3]. Our approach is not to treat the web feeds as the only sources of information, but as a means that allows the *self-supervised* training and generation of a wrapper automatically.

Our research makes the following main contributions. 1)We use web feeds for training and generating a wrapper. The generated wrapper is described in simple rules that are induced by following a probabilistic approach. We provide a simple algorithm that is noise-tolerant and takes into account the information collected about the location of HTML elements found during training. 2)We make use of CSS Classes as an attribute that can supplement the more traditional XPath manipulation approach used to describe data extraction rules. 3)To the best of our knowledge, we are the first to propose a self-supervised methodology that can be applied on any weblog and features unique levels of granularity, automation and accuracy. We support all of the above through evaluation.

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

2. PROPOSED MODEL

Figure 1 presents an overview of the overall approach, which involves the execution of three steps.

Step 1: Feed Processing and Capturing of Post Properties

The first step includes the task of reading and storing the weblog properties found in the web feed. Similarly to standard RSS readers, we focus on the entries that contain the post title, author, main content and publication date.

Step 2: Generation of Filters

The second step includes training the wrapper through the cross matching of information found in the web feed and the corresponding HTML documents. This step leads to the generation of information, captured through the filters, which describes where the weblog data properties reside. The concept of a *filter* has already been used in research related to web information extraction. Baumgartner et al.[1] used the term filter as the building block of patterns, which in turn describe a generalised tree path in the HTML parse tree. In our approach, the filter is described using three basic attributes: the Absolute Path, the CSS Classes and the 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. The matching of the elements is treated differently for different properties: for

the title we look for absolute and complete matchings, for the content we use the Jaro-Winkler metric [4] which returns high similarity values when comparing the summary (feed) against the actual content (web page), for the date we use the Stanford NER suite for spotting and parsing the values (`http://nlp.stanford.edu/software/CRF-NER.shtml`), and for the author we use partial and absolute matching with some boilerplate text (i.e. "Written By" and "Posted By").

Step 3: Induction of Rules and Blog Data Extraction

The final 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. This noise is produced due to several reasons (e.g. the value of the Absolute Path may vary across the posts or more than one matching element may be found in a single post). As seen in Algorithm 1, an iteration takes place for each of the candidate rules which in turn is applied to each of the training posts. For each successful match, the score of the rule is increased by one. After all posts have been checked, the value is divided by the number of training posts which the rule was validated against, in order to represent a more normalised measurement. The rule having the highest score – if any – is returned.

Algorithm 1 Rule induction algorithm

Inputs:
Collection of training posts P, Collection of candidate rules R
Outputs:
Rule with the highest score
for all Rules $r \in R$ **do** ▷ Initialize all scores
$r.score \leftarrow 0$
end for
Rule $rs \leftarrow$ new Rule()
$rs.score \leftarrow 0$
for all Rules $r \in R$ **do**
▷ Check if application $r(p)$ of rule r, on post p succeeds
for all Posts $p \in P$ **do**
If $r(p)$ =value-property of p **then**
$r.score++$
end for
$r.score \leftarrow \frac{r.score}{|P|}$ ▷ Normalize score values
▷ Check if this is the best rule so far
If $r.score > rs.score$ **then**
$rs \leftarrow r$
end for
return rs

3. EVALUATION

We evaluated our model against a collection of 240 weblogs (2,393 posts) for the title, author, content and publication date. For the same collection, we used the Google Blogger and WordPress APIs (in the limits of free quota) in order to get valid and full data (i.e. full post content) and followed the 10-fold validation technique. As seen in Table 1, the prediction accuracy is high (mean value *92%*). Furthermore, we compare the accuracy of the post content extraction against Boilerpipe[2] and the results show that we achieve *81.6%* relative error reduction.

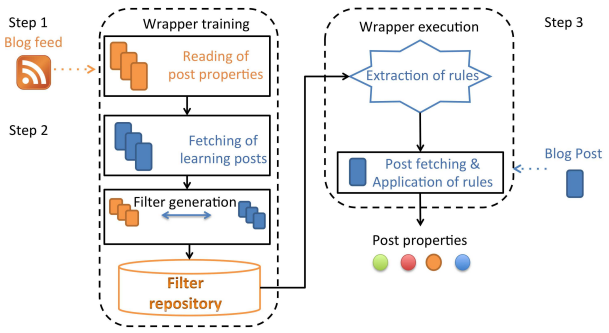

Figure 1: Overview of the weblog data extraction approach.

Table 1: Percentage of successfully extracted properties. Number of misses are in parenthesis.

	Title	Content	Publication Date	Author
Proposed Model	97.3%(65)	95.9% (99)	89.4% (253)	85.4% (264)
Boilerpipe	0	77.4% (539)	N/A	N/A

4. CONCLUSIONS

We have presented a method for fully automated weblog wrapper generation. Based on the weblogs' feeds, our model realises an effective and zero-cost labelling technique. The generated wrapper exhibits increased granularity, since it manages to identify and extract several weblog properties, such as the title, author, publication date and main content of the post.

5. ACKNOWLEDGMENTS

This work was conducted as part of the BlogForever project funded by the European Commission Framework Programme 7 (FP7), grant agreement No.269963.

6. REFERENCES

[1] R. Baumgartner, S. Flesca, and G. Gottlob. Visual Web Information Extraction with Lixto. In *Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, pages 119–128, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[2] C. Kohlschütter, P. Fankhauser, and W. Nejdl. Boilerplate detection using shallow text features. In *Proc. of the 3rd ACM international conference on Web search and data mining*, pages 441–450. ACM, 2010.

[3] M. Oita and P. Senellart. Archiving data objects using Web feeds. In *Proc. of International Web Archiving Workshop*, pages 31–41, Vienna, Austria, Sept. 2010.

[4] W. E. Winkler. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. *Proc. of the Section on Survey Research Methods American Statistical Association*, pages 354–359, 1990.