



A Framework for Authorship Identification in the Internet Environment

Authorship Recognition Tool

Aleš Horák, Jan Rygl



Structure

- Motivation: Why do we need ART?
- ART components
- The Internet component
- Technical issues

Motivation: The offline approach

- Problem A: An anonymous document of the known author
- We are given:
 - Collection of documents with known authorship
 - Anonymous document written by one author from the collection
- Goal:
 - Narrow set of potential authors
 - Assign the authorship to the document

Motivation: The offline approach

Examples:

- A known author published under a pseudonym
- A recidivist wrote another anonymous threat



Motivation: The online approach

- Problem B: Anonymous documents written by an author who is not in database
- We are given only anonymous texts
- Goal:
 - Create author's writeprint
 - Search Internet
 - Collect author's documents with identified authorship (advertisement, blogs, school works)
 - Determine author's identity

Motivation: The online approach

Examples:

- An anonymous discussion forum
- Anonymous blogs




ART components

- Data storage
- Internet component
- Language tools
- Authorship recognition component
- Web interface

Internet component 1

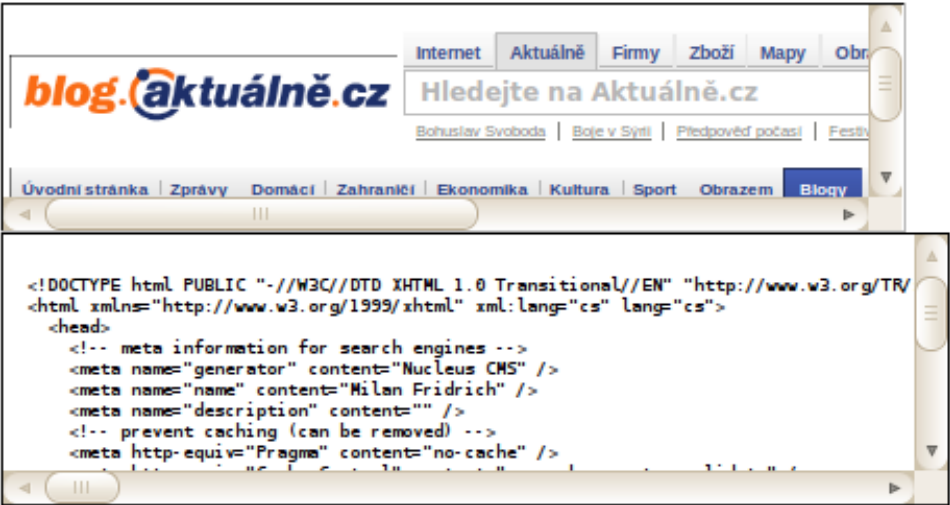
■ Expert selects domain

 **Authorship Recognition Tool**

[Home page](#) | [View projects](#) | [View data downloaders](#) | [Log out](#)

Configuration of new data source

Document url: [Detect structure](#)



The screenshot displays a web browser window showing the 'blog.aktuálně.cz' website. The page features a navigation menu with links like 'Internet', 'Aktuálně', 'Firmy', 'Zboží', 'Mapy', and 'Obr'. Below the menu, there's a search bar and a list of categories including 'Bohuslav Svoboda', 'Boje v Synti', 'Předpověď počasí', and 'Festi'. The browser's address bar shows the URL 'http://blog.aktualne.centrum.cz/blogy/milan-fridrich.php?itemid=14900'. Below the browser window, the HTML source code is visible, starting with the DOCTYPE declaration and including meta tags for generator, name, description, and caching.

Internet component 2

- ART tries to detect structure of domain automatically:
 - Every attribute (author, heading, text, ...) has predefined ordered list of tags and attribute keywords
 - Python XHTML selectors are used
 - Author:


```
<h7 [a-zA-Z]+='.*author.*'> ~ //h7[contains(@*, 'author')]  
<span class='postedby'> ~ //span[@class='postedby']
```

Internet component 3

author	value
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1
16	1
17	1
18	1
19	1
20	1
21	1
22	1
23	1
24	1
25	1
26	1
27	1
28	1
29	1
30	1
31	1
32	1
33	1
34	1
35	1
36	1
37	1
38	1
39	1
40	1
41	1
42	1
43	1
44	1
45	1
46	1
47	1
48	1
49	1
50	1
51	1
52	1
53	1
54	1
55	1
56	1
57	1
58	1
59	1
60	1
61	1
62	1
63	1
64	1
65	1
66	1
67	1
68	1
69	1
70	1
71	1
72	1
73	1
74	1
75	1
76	1
77	1
78	1
79	1
80	1
81	1
82	1
83	1
84	1
85	1
86	1
87	1
88	1
89	1
90	1
91	1
92	1
93	1
94	1
95	1
96	1
97	1
98	1
99	1
100	1

```
//div[contains(@*, 'autor')]/t
```

Milan Fridrich



Uvodní stránka | Zprávy | Domácí | Zahraničí | Ekonomika | Kultura | Sport | Obrazem | Blogy

III

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs" lang="cs">
<head>
  <!-- meta information for search engines -->
  <meta names="generator" content="Nucleus CMS" />
  <meta names="name" content="Milen Fridrich" />
  <meta names="description" content="" />
  <!-- prevent caching (can be removed) -->
  <meta http-equiv="Pragma" content="no-cache" />
```

1.

Attribute name	Guessed path	Guessed value	Manually filled path	Manually filled content	Preprocess function	Postprocess function	
raw_text	<code>//div[contains(@*, 'content')]</code>	tradiční České televize i celé české společnosti. Změna programu na kterém se			<code>lambda body:body</code>	<code>lambda body:body</code>	Update field
publication_time	<code>//small[contains(@*, 'date')]</code>	27. 11. 2011 10:00			<code>lambda body:body</code>	<code>lambda body:body</code>	Update field
author_value	<code>//div[contains(@*, 'autor')]/t</code>	Milan Fridrich			<code>lambda body:body</code>	<code>lambda body:body</code>	Update field
document_heading	<code>//h2[contains(@*, 'nadpis')]/l</code>	Proč se Večerníček přesouvá na			<code>lambda body:body</code>	<code>lambda body:body</code>	Update field

Internet component 4

- For unrecognized attributes expert manually selects examples of attributes

Attribute name	Guessed path	Guessed value	Manually filled path
raw_text	<code>//div[contains(@*, 'content')]</code>	<code><div class="contentbody-item"></code>	<input type="text"/>
publication_time	<code>//small[@class="contentitem"]</code>	27. 11. 2011 10:00	<input type="text"/>

Manually filled content	Preprocess function	Postprocess function	
<input type="text"/>	<code>lambda body:body</code>	<code>lambda body:body</code>	<input type="button" value="Update field"/>
27. 11. 2011 10:00	<code>lambda body:body</code>	<code>lambda body:body</code>	<input type="button" value="Update field"/>

Internet component 5

- Text preprocessing or postprocessing is set by the expert

Guessed value	Manually filled path	Manually filled content	Preprocess function	Postprocess function
<code><div class="contentbody- +am"></code>	<input type="text"/>	<input type="text"/>	<code>lambda body:body</code>	<code>lambda body:body</code>
<code>27. 11. 2011 10:00</code>	<input type="text"/>	<code>27. 11. 2011 10:00</code>	<code>lambda body:body</code>	<code>lambda body:body.replace(u' ,u' ').strip()</code>

Internet component 6

- ART creates document downloader (crawler) from information
- ART collects documents and stores into the database

Internet component 7

- ART regularly:
 - searches new documents
 - checks the structure of the domain using reference documents
 - If a redownloaded document differs from the reference document, all initialization steps are automatically repeated.

Technical issues 1

■ Non-standard HTML formatting

```
<tr><td>My first comment</td></tr>
```

```
<tr><td>Adam Novák</td></tr>
```

```
<tr><td>Hello.</td></tr>
```

```
<!-- next document -->
```

```
<tr><td>My second comment</td></tr>
```

```
<tr><td>Adam Novák</td></tr>
```

```
<tr><td>Hello again.</td></tr>
```

```
<!-- next document -->
```

Technical issues 2

- Non-HTML content
 - Javascript
 - Flash
 - ...

Technical issues 3

- Authorised access
 - Manual registration
 - Automatic authorization

Summary

- Web Domain Analysis still requires experts, but the process of the analysis is quicken
- Document downloading is automatic and responds to structure changes of the domain

Future work

- Remake prototypes to product versions
- Solve technical issues
- Reduce number of tasks which require experts