Crawling von Datenschutzhistorien

Alexander Prull, Jörn-Henning Daug, Simon Kaleschke

Februar 2017

1 Projektbeschreibung

2 Lösungsansatz

3 Softwarearchitektur

Die Software ist in drei Teile aufgeteilt: Frontend, Backend, und Extraktion.

3.1 Frontend

3.2 Backend

3.3 Extraktion

Die Ruby-Struktur zur Extraktion der Datenschutzhistorien funktioniert folgendermaßen:
Vom Backend aus wird *crawl.rb* aufgerufen. Dieses Skript beinhaltet alle Einstellungen zur Extraktion von jeder Webseite. Es sendet diese Einstellungen an *PrivacyExtractor.rb*, das die Extraktion dann vollzieht. Hierfür bezieht es Links zu älteren Versionen entweder vom Archiv der Webseite selbst, oder mit *ArchiveExtractor.rb* von dem Internetarchiv *archive.org*. Der Inhalt jeder Seite wird extrahiert, auf die notwenigen Elemente zugeschnitten und formatiert. Zuletzt werden mit *PrivacyDiffer.rb* die extrahierten Texte auf Unterschiede untersucht und gegebenenfalls in einer Datenbank abgespeichert. Die fertige Datenbank kann vom Frontend weiterverarbeitet werden.
Weitere Informationen sind auf der Abbildung 1 auf Seite 5 zu finden.

4 Bedienungsanleitung

4.1 Vorbereitung

Damit die Software reibungslos läuft, werden folgende vorinstallierte Pakete erwartet:
Für das Frontend:

- Bower.
- ...

Für das Backend:

- Java.
- SQLite3.
- Maven.
- ...

Für die Extraktion:

- Ruby 2.3.1 oder höher.
- Die Ruby-Gems *date, open-uri, open_uri_redirections, nokogiri, rubygems, sqlite3, json, differ.*

Tragen sie außerdem in der Datei *config.ppp* den von Ihnen gewünschten Pfad für die Ergebnisdatenbank ein. (Standard ist *∼/html/policies.db*)

4.2 Server einrichten

4.3 Serverstart und allgemeiner Programmablauf

4.4 Webseite einzeln crawlen

Normalweise wird der Aufruf zur Extraktion vom Backend aus gestartet. Wollen Sie jedoch eine einzelne Webseite crawlen, gehen sie folgendermaßen vor:

1. Sichern Sie, dass die Webseite und zugehörige Optionen im Quellcode vermerkt ist. (Siehe dazu Abschnitt 7.2)

2. Sind noch keine Informationen zu der Webseite vorhanden, oder wollen Sie alle Informationen von null an neu generieren lassen, wechseln sie in den Ordner *script* und starten sie den Aufruf `ruby crawl.rb webseite fetch`. Wollen sie bspw. alle Informationen für Google neu generieren lassen, ersetzen sie im Aufruf `webseite` durch `google`.

3. Wollen Sie nun überprüfen, ob eine neue Version verfügbar ist, starten Sie den Aufruf `ruby crawl.rb webseite update`.

4.5 Datenbankstruktur

Die SQLite-Datenbank hat folgende Spalten:

Spalte	Bedeutung
ID	Primärschlüssel.
SYSTEM_DATE	Datum, an dem die Datenschutzhistorie in dem aktuellen Zustand war.
DISPLAY_DATE	Gleiches Datum in benutzerfreundlicher Schreibweise.
LINK	Hyperlink, wo die extrahierte Historie zu finden ist.
CONTENT	Der extrahierte Plaintext.

5 Ergebnisse

5.1 Allgemeines

[...]

Eine Liste aller gecrawlten Webseiten ist auf Abbildung 2 auf Seite 6 zu finden.

5.2 Die Website

6 Einschränkungen

7 Erweiterungsmöglichkeiten

7.1 Daten weiterverarbeiten

Die gewonnenen Daten in der Datenbank können unabhängig von der Webseite vielseitig weiterverwendet werden. so wäre bspw. die Anbindung an weiter Anwendungsprogramme, zum Beispiel zur Bewertung der Datenqualität oder zur Suche nach Schlüsselwörtern, möglich.

7.2 Webseite hinzufügen

Wenn Sie eine Webseite zum Tool hinzufügen möchten, so brauchen Sie i.A. den Namen der Webseite und einen Hyperlink zur Webseite, auf der die Datenschutzerklärung zu finden ist. Wie weiter vorzugehen ist, wird im folgenden beschrieben:

7.2.1 Frontend

7.2.2 Backend

[...]

Fügen Sie außerdem in *crawl.rb* den Namen der Webseite, den Link, den Namen der Tabelle, in der die Informationen gespeichert werden sollen, wie die HTML-Struktur zu modifizieren ist, etc. ein. Näheres wird mit Beispielen im Skript selbst erläutert.

8 Anhang

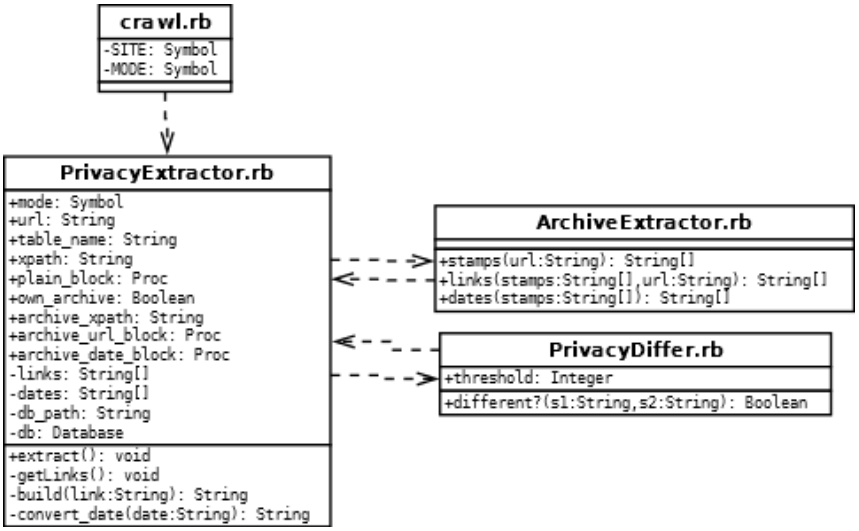

Abbildung 1: Architektur Extraktion

Firma	Eigenes Archiv	Versionen	Zeitspanne	Qualität
Alternate	✗	7	08/2014 - 07/2016	0.9
Amorelie	✗	11	01/2013 - 10/2016	0.7
Apple	✗	12	09/2014 - 09/2016	1.0
Burgerking	✗	2	02/2015 - 12/2016	0.7
Edeka	✗	4	11/2014 - 10/2016	0.7
Google	✓	22	06/1999 - 01/2017	0.8
Microsoft	✗	7	02/2016 - 01/2017	0.9
Payback	✗	10	09/2011 - 10/2016	0.9
Paypal	✗	2	04/2014 - 11/2016	0.9
RocketbeansTV	✗	4	10/2014 - 09/2016	1.0
Steam	✗	3	09/2012 - 11/2016	0.8
Subway	✗	3	05/2016 - 10/2016	0.9
Süddeutsche	✗	1	04/2015 - 04/2015	0.2
Trivago	✗	1	04/2016 - 04/2016	0.7
Twitter	✓	10	05/2007 - 01/2016	0.8
Uni Leipzig	✗	1	06/2013 - 06/2013	0.5
Vine	✗	5	03/2013 - 01/2017	1.0
WhatsApp	✓	2	07/2012 - 01/2017	0.9
Wikimedia	✓	4	06/2006 - 06/2014	1.0
Zalando	✗	3	09/2010 - 11/2012	0.9

Abbildung 2: Übersicht gecrawlte Webseiten