Некоторые проблемы автоматизированного извлечения данных из веб-страниц

Е.С. Чиркин

Тамбовский государственный университет имени Г.Р. Державина
chirkin@tsu.tmb.ru

Аннотация

Статья посвящена описанию частых проблем и некоторых их решений при автоматизированном извлечении данных (data extraction) для их последующего глубинного анализа (data mining) из сети Интернет.

Введение

В современном мире востребованы интеллектуальные системы, извлекающие (структурирующие) информацию из неструктурированных источников (в англоязычной литературе — data extraction как часть data mining (следует отметить, что последнее понятие не имеет устоявшегося перевода на русский язык)). Обычно в роли источников выступают простые тексты, однако возможны и иные случаи – веб-страницы, многоколоночный текст, таблицы, требующие предварительного анализа перед, собственно, этапом извлечения знаний. Типичный сложный случай представляет собой страница газеты или веб-сайта – новостного ресурса. Их страницы содержат многоколоночный текст, врезки, цитаты, таблицы, инфографику, комментарии читателей, различные типы рекламных блоков. Причём материал скомпонован характерным образом, формирующим у читателя необходимое мнение и обращающим его внимание на ключевые моменты публикации. Именно эта компоновка элементов превращает извлечение текста со страницы в проблему с нетривиальным решением.

Извлечение данных (извлечение информации) – это процесс получения структурированных данных из слабоструктурированных или неструктурированных источников. Классический пример – парсинг каталога товаров с сайта интернет-магазина, что чаще всего используется для мониторинга цен, товаров, услуг (это не всегда абсолютно законно согласно соглашению об использовании магазина, но весьма востребовано), либо для «грязных» и приближенных к ним SEO-технологий (создание сайтов-копий и др.), социологические исследования [2], создание агрегаторов, входной фильтр для поисковых систем.

Извлечение данных (data extraction) предваряет собой извлечение знаний (data mining) [6] и является необходимым этапом предварительной обработки, от которого, в значительной степени зависит качество извлечённых знаний. Даже такая простая задача, как извлечение текста из контента с веб-страницы сопряжена с определённой сложностью – требуется отсечь «шапку» сайта, верхнее и вертикальное меню, блоки навигации, рекламные блоки, нижнюю часть сайта с указанием правообладателя, студии-разработчика, контактов, нижнего меню и прочих элементов дизайна, не содержащих, собственно, полезной информации.

В данной статье под неструктурированными источниками понимаются веб-страницы, не имеющие значимой машиночитаемой семантической разметки, а под слабоструктурированными – логически взаимосвязанные фрагменты неструктурированного источника (например, таблица), размещённые в пределах одной или нескольких веб-страниц.

Проблема извлечения текста и слабоструктурированных данных из веб-страниц усугубляется повсеместным нарушением и смешением стандартов и рекомендаций по вёрстке веб-страниц. Вызвано это может быть как целями конкретного сайта (для газеты – формирование нужного мнения посредством специального оформления и компоновки материала), так и простой небрежностью и ошибками в коде страниц сайта (может быть связано с недостаточной квалификацией лиц, ответственных за создание и/или функционирование сайта). Ситуация ещё более усугубляется совершенством веб-браузеров, корректно отображающих данные страницы, однако специализированные парсеры обычно не способны справляются с данной проблемой.

Ряд примеров, иллюстрирующих последний абзац:

1) Фрагмент кода страницы с http://www.chaskor.ru/article/torrenty_tehnika_yuridicheskoj_bezopasnosti_:

```
…тренное
<http://www.zakonrf.info/koap/20.29/>
статьёй 20.КоАП</a>. В об…
```

По закрывающему тегу `</A>` можно предположить, что пропущен его фрагмент «`A HRef=`» после первой открывающей угловой скобки.

2) Ошибочный фрагмент кода:

```
…что эквивалентно z=x-l, z(-<span
style="face="Times New Roman">&#8734;
</span>,-1]. …
```

Видна определённая попытка смешения форматирования с помощью стилей, так и с помощью тега

Интернет и современное общество: сборник научных статей XVI Всероссийской объединенной конференции IMS-2013, Санкт-Петербург, 9 - 11 октября 2013 г.

<FONT>. Данный пример сложен тремя кавычками, которые должны употребляться строго попарно. Большинство парсеров, либо считает третью кавычку открытием и пропускает весь текст до конца страницы (пытаясь поместить его в значение атрибута тега), либо останавливаются на закрывающей угловой скобке.

3) Во многих текстах, замещающих рисунок, можно встретить подобные многострочные конструкции:

```
<img src="…" alt="первая строка<br />вторая строка<br¶/>другие¶ строки">
```

Проблемные моменты: а) использование разметки HTML внутри значения атрибута; б) несколько «жёстких» переводов строки (обозначены символом «¶»); в) смешение стандартов кода — HTML и xHTML. Первая проблема из них является принципиальной и поведение парсеров HTML при встрече с ней предсказать невозможно. Наиболее частое поведение – пропадает часть страницы между «открывающим» проблемным тегом и таким же проблемным тегом у следующей картинки, который рассматривается как «закрывающий». Подобное поведение характерно для парсеров HTML на основе конечных автоматов (т.е. практически всех) — встретив спорную конструкцию, он переходит в состояние ожидания «закрывающей» конструкции и когда находит — продолжает разбор, при этом из текста вычленяется неверный фрагмент.

Методы извлечения данных

Все методы извлечения данных с сайтов можно разбить на три основные группы:

Ручные методы — мониторинг интересующих страниц и перемещение необходимой информации в базу осуществляется специальным оператором (человеком). Достоинства подхода – высочайшее качество извлечения и низкие требования к квалификации оператора, недостатки — высокая трудоёмкость, общая незначительная скорость работы и человеческий фактор. На практике количество ошибок может составлять 2% и выше, что, часто бывает неприемлемо высокой величиной.

Полуавтоматические методы — к этой группе относятся решения, способные извлекать информацию после определённой настройки. Сюда можно отнести как предварительная разметка оператором элементов страницы в графическом пользовательском интерфейсе, так и более сложные случаи — например, составление регулярных выражений или запросов на языке XQuery (один из предков данного языка распространён под именем XPath). Наиболее распространенными универсальными коммерческими приложениями для полуавтоматического извлечения данных являются Website Data Extractor от Mozenda и Visual Web Ripper К достоинствам подобной группы методов следует отнести, в идеале, такое же высокое качество извлечения [7], как и при ручном способе, но намного более высокую скорость работы. Недостаток — созданная оператором разметка может быть ошибочной, требуется начальное и периодическое участие оператора, квалификация оператора должна отвечать определённым требованиями (например, ему следует уметь составлять сложные регулярные выражения).

Автоматические (интеллектуальные) методы. Принципиальным отличием является полное отстранение человека от процесса извлечения данных — страницы анализируются и информация извлекается автоматически без какого бы то ни было участия оператора [7].

Очевидно, в связи с нарастающим ростом интереса к технологиям Big Data и data mining, наиболее востребованным является именно последний подход — полностью автоматическое извлечение данных.

Этапы извлечения данных

Извлечение структурированных взаимосвязанных данных из веб-страниц можно свести к ряду последовательно решаемых задач.

Навигация

Под навигацией (navigation problem) подразумевается проблема поиска и анализа на веб-сайте страницы с извлекаемой информацией. При этом необходимо определить, вся ли интересующая информация находится на странице и выделить ссылки на другие её части (точнее, обычно делается наоборот – если нет ссылок на другие страницы – значит, других страниц нет). Основная проблема навигации заключается в использовании разработчиками кода веб-страниц современных или необычных технологий, а также просто нарушение следующих негласных правил проектирования сайтов.

Навигация посредством кода на JavaScript. Очевидно, что при этом для перехода на другую страницу сайта требуется исполнение кода, посредством которого осуществляется навигация, в то время как извлекающий данные парсер обычно намного проще браузера и не исполняет скриптов (как правило, к парсеру также предъявляются определённые требования по быстродействию, что автоматически исключает всякую необходимость исполнения каклибо скриптов парсером).

Загрузка контента посредством AJAX и «бесконечная» прокрутка страницы: проблема, дополняющая собой предыдущую — к невозможности извлечь ссылку на следующую страницу из самой страницы сочетается с загрузкой контента по запросу с самой страницы. Проблема является, в большинстве случаев, фундаментальной и не решается без исполнения кода, осуществляющего показ необходимого контента. Кстати, данный подход также, бывает, используется для защиты веб-сайтов от несанкционированного копирования. Основное предназначение — увеличение удобства использования сайта. Например, прокрутка списка сообщений посредством полосы прокрутки страницы браузером в социальной сети «ВКонтакте».

Использование во внутреннем механизме постраничного показа не номеров страниц. В настоящее время почти не встречается — обычно при навигации используется последовательная нумерация страниц (с шагом 1 или больше (например, на форумах по номерам первого выводимого сообщения на странице)). Однако встречается навигация по, например, неким id или датам.

Анализ будет происходить тем быстрее и эффективнее, чем меньше будет найдено дубликатов страниц с извлекаемой информацией. *Замечание:* это имеет значение, т.к. большинство алгоритмов выделения структуры в том или ином смысле используют статистику элементов. Дубликаты нарушают статистическое распределение элементов и могут нарушить распознавание. Проблема заключается в том, что многие CMS (системы управления содержимым сайта) предоставляют доступ к основным страницам по разным адресам.

Распознавание постраничных ссылок может некорректно работать с их незначительным количеством. Например, стандартная навигация для двух страниц может иметь шесть постраничных ссылок приблизительно следующего вида:

< << 1 2 >> >,

где:

«2» — текущая страница, а также гиперссылка на текущую страницу,

«1» — гиперссылка на первую страницу, причём первая подразумевается в смысле «предыдущая перед текущей»,

«<» — гиперссылка для перехода к первой странице,

«<<» — гиперссылка для перехода к предыдущей странице,

«>» — гиперссылка для перехода к последней странице,

«>>» — гиперссылка для перехода к следующей странице.

Таким образом, шесть ссылок на две страницы (1, 1, 1, 2, 2, 2) способны создать сложность для распознавания элементов постраничной навигации.

Преодолевается обычно массовым посещением с данной всех возможных страниц и последующий их анализ с постобработкой.

На практике возможно создание парсера на основе ActiveX-компонента Internet Explorer (или компонентов модуля QtWebKit), при этом парсер получает возможности соответствующего браузера, в том числе и полный доступ к динамически создаваемому и/или загружаемому контенту. Возможно также написание расширения к браузеру с аналогичной целью.

Некоторые современные CMS способны предоставлять доступ к интересующей информации по прямым ссылкам, через оглавление сайта и др., т.е. не только через страницу, использующую данный проблемный подход.

Помимо этого на практике встречаются и другие проблемы — сайт полностью или целиком может быть сделан с использованием технологии Flash, ограничение на количество запросов в интервал времени (в секунду, в минуту, в час), ограничение на количество запросов с одного IP-адреса, ограниченный доступ ко всем или отдельным страницам сайта. Общим, но действенным решением данной проблемы следует считать принудительное ограничение количества соединений с сайтом (не более 3-х в секунду).

Распознавание извлекаемых данных

Распознавание данных и распознавание структуры (data extraction problem) – задача заключается в необходимости определения участков на веб-странице, содержащих извлекаемую информацию, а также структуру извлекаемых данных.

Наиболее часто, но не всегда, данная задача заключается в поиске повторяющихся структур данных [4]. В простейшем же случае веб-страницу следует представить в виде дерева DOM-элементов и искать узлы, имеющие одинаковую структуру дочерних узлов. Иные способы:

1) На основе анализа графического представления веб-страницы. Чрезвычайно ресурсоёмкий, но эффективный подход. На практике: а) несложно найти сайты проблемной разметкой, которая некорректно отображается и, соответственно, плохо поддается анализу; б) на отдельных группах сайтов нормой стали всплывающие модальные окна (как с рекламой, так и с предложением использовать, например, приложения и социальные сервисы данного сайта), закрывающие собой контент.

2) Поиск на основе семантической разметки и микроформатов — отвечает первоначальным целям создания Всемирной паутины, но семантическая разметка в настоящее время массово используется практически с противоположными целями — для повышения рейтинга в поисковых системах, но не для смысловой разметки контента.

3) Текстовый анализ – выделение элементов по их содержимому с учётом обрамляющей HTML-разметки. Например, редкий текст обходится без имён собственных. Соответственно, база данных имён, фамилий, названий компаний, стран и населённых пунктов способна значительно повысить качество выделения информативных текстовых блоков. Другой вариант: известно, что в HTML есть потоковые элементы, а есть блочные (управляющие взаимным положением элементов при их отображении). Вся вёрстка формируется блоковыми элементами, всё форматирование — потоковыми. Редчайшие исключения являются специфичными и их допустимо игнорировать при извлечении данных.

Обеспечение единообразия

Обеспечение единообразия — следует обеспечить однородность извлекаемых данных, которые могут быть представлены на веб-странице с некоторой вариативностью атрибутов. Рассмотрим следующий сложный случай. Пусть существует каталог статей. Каждая статья представлена файлом с ней,

названием, автором, аннотацией, ключевыми словами. И значительное количество полей не будет заполнено, поэтому не будет выведено. Разумеется, при корректном извлечении данных следует недостающие поля оставить пустыми. Основная сложность здесь заключается в возможном сбое определения количества извлекаемых полей для каждой записи, поэтому, возможно, что при кластеризации данные будут сгруппированы неправильным образом (например, статьи с аннотациями образуют один набор данных, а статьи с аннотациями и авторами — другой, что, в общем случае, неверно). Помимо этого некоторые данные часто следует нормализовать отдельными алгоритмами: например, телефон может быть записан как «тел. 23-7-89», «8 (800) 555-5555», «88005555555» и другими способами. Аналогично — даты и время, населённые пункты (например, «г. Санкт-Петербург», «Санкт-Петербург», «Питер» и др.).

Объединение данных

Проблема объединения данных заключается в следующем: извлекаемая конкретная единица информации (запись) может быть представлена на сайте (или даже странице) неоднократно, поэтому по окончании выделения необходимо обеспечить удаление дубликатов. Это задачу легко можно решить по окончании извлечения данных, однако парсинг по сравнению с принятием решения по окончании более ресурсоёмкой операцией, поэтому оптимальнее дубликаты обнаруживать до него. В свою очередь, не всегда в принципе возможно определить дубликаты страниц. Например, в магазине радиодеталей «Чип и Дип» (`www.chipdip.ru`) существует множество полностью одинковых записей и только при открытии конкретного товара можно определить, что, например, речь идёт о разных партиях от разных поставщиков.

После осуществления данных этапов информация с веб-страницы приводится в систематизированную форму, пригодную для импорта в базу знаний.

Однако не всегда следует сосредотачиваться на полностью автоматическом извлечением информации из веб-страниц. Интересный факт, замеченный в нашей работе — в связи с повсеместным распространением одних и тех же CMS (например, Joomla, Wordpress, Drupal и др.) со стандартными модулями и расширениями всего 37 групп сложных правил способно корректно обработать более 98% русскоязычных сайтов в полуавтоматическом режиме.

Заключение

Извлечение структурированных данных из слабоструктурированных или неструктурированных источников с последующим извлечением знаний является базой для построения самообучающихся систем, лежащих в основе трёх самых востребованных направлений современности - экспертные системы (например, IBM Watson [3]), поисковые системы (например, Яндекс [1]) и, в идеале, предсказывающие экспертные системы (не существуют, вероятный кандидат — IBM Watson). В статье затронуты основные проблемы, которые встанут перед разработчиком системы извлечения информации из веб-страниц и упомянуты пути их возможного решения.

Литература

[1] Ершов А. «Мы фанаты машинного обучения»: главный специалист «Яндекса» по ранжированию рассказал о персонализации и счастье пользователей [Электронный ресурс]. URL: http://lenta.ru/articles/2013/06/17/yandexsearch/ (дата обращения: 13.08.2013).

[2] Chow T., Lin Y., Chan W. The Development of a Web-based Demographic Data Extraction Tool for Population Monitoring // Transactions in GIS, 2011. Vol. 15. P. 479—494.

[3] IBM Watson: a sophisticated data analytics & insight engine [Электронный ресурс]. URL: http://www-01.ibm.com/software/ebusiness/jstart/watson/ (дата обращения: 13.08.2013).

[4] Li Z., Ng W., Sun A. Web data extraction based on structural similarity // Knowledge and Information Systems, 2005. Vol. 8. P. 438—461.

[5] Liddle S., Yau S., Embley D. On the Automatic Extraction of Data from the Hidden Web // H. Arisawa, Y. Kambayashi (Eds.): ER 2001 Workshops, LNCS 2465, 2002. P. 212—226.

[6] Velasquez J., Palade V. A Knowledge Base for the maintenance of knowledge extracted from web data // Knowledge-Based Systems. 2007. Vol. 20. P. 238—248.

[7] Zhai Y., Liu B. Extracting Web Data Using Instance-Based Learning. // Proceedings of 6th International Conference on Web Information Systems Engineering (WISE-05), 2005.

The some problems of automated data extraction from web pages

E.S. Chirkin

The article describes the some frequent problems and their solutions in some automate the extraction of data (data ex-traction) for further in-depth analysis (data mining) from the Internet.