УДК 004.91

АВТОМАТИЧЕСКОЕ ВЫДЕЛЕНИЕ ИНФОРМАЦИИ ОБ АВТОРЕ И ИХ ТЕКСТАХ НА СТРАНИЦАХ ИНТЕРНЕТ-ФОРУМОВ

Пронин А. К., Копылов Н. Ю.

ООО «Аби ИнфоПоиск», Москва, Россия (127273, Москва, ул. Отрадная, 2Б.6), e-mail apronin@abbyy.com

В данной статье рассматривается метод автоматического выделения со страниц Интернет-форумов публично доступной информации об авторе сообщений (пола, возраста, местоположения) и принадлежащих ему текстах. Для построения алгоритма использовалась концепция деревьев стилей, представляющих собой подход по агрегации схожих вершин в древовидной структуре, представляющей объектную модель документа. Сходными считаются вершины, имеющие одинаковые имена соответствующих HTML-тэгов и имеющие одинаковый родительский узел. На конечных шагах алгоритма использованы простые эвристики, использующие наблюдения о характере текстов, содержащих псевдонимы пользователей и их тексты. При тестировании построенного алгоритма достигнута точность 80 %. Практическая ценность разработанного алгоритма заключается в расширении множества текстовых ресурсов, используемых в качестве источников естественных текстов в задаче построения очень больших корпусов.

Ключевые слова: атрибуция, извлечение информации, HTML разметка.

AN APPROACH OF AUTOMATIC EXTRACTION OF INFORMATION ABOUT THE AUTHORS AND THEIR TEXTS FROM WEB-FORUMS

Pronin A. K., Kopylov N. Y.

LLC «Abi InfoPoisk», Moscow, Russia (127273, 2B.6, Otradnaya, Moscow), e-mail apronin@abbyy.com

This article describes the approach of automatic extraction of information about the author and his/her texts from web forums. For building the algorithm the concept of style trees was used – approach of aggregating similar nodes in a tree representing Document Object Model. Nodes are similar if they all have the same name of the corresponding HTML-tags and have the same parent node. At final steps, simple heuristics were applied, employing observations about characteristics of texts containing users' pseudonyms and their messages. When testing the developed algorithm 80 % accuracy was reached. Practical value of the developed algorithm resides in expansion of text resources, used as sources for natural discourse, especially when faced with a problem of building very large text corpus.

Keywords: author attribution, text mining, HTML, markup.

Введение

Для наполнения корпуса терабайтного объема требуется значительное количество естественных текстов, размещенных на публично доступных интернет-ресурсах. Форумы содержат лексику, необходимую для создания корпуса, сбалансированного по коммуникативным целям.

При извлечении текста из форумов сложность заключается в том, что значительная часть текста естественного языка находится в коротких сообщениях – обычном обсуждении или же комментариях под статьей. Стандартные алгоритмы удаления обвязки могут выделить текст, но при этом точность отделения текстов пользователей от остального контента не гарантируется. Форумные тексты, как правило, могут быть соединены с сопутствующей информацией из профиля пользователя или сведений, сообщаемых им о себе и размещенных непосредственно в сообщении, что позволяет, при сохранении такой

метаразметки, группировать тексты корпуса по значению некоторого атрибута и использовать полученный корпус в задачах машинного обучения.

Для работы с веб-страницами, в том числе и парсинга, удобство и унифицируемость представляют следующие две технологии:

1) DOM (Document Object Model) – не зависящий от платформы и языка программный интерфейс, позволяющий программам и скриптам получить доступ к содержимому HTML, XHTML и XML-документов, а также изменять содержимое, структуру и оформление таких документов [3, 4].

2) XPath (XML Path Language) – язык запросов к элементам XML-документа, использующийся для указания точного пути к вершине в DOM-дереве [1].

Построение объектной модели документов производилось при помощи пакета org.w3c.dom [2] для языка программирования Java, что повлечет за собой использование терминологии и обозначений, присущих данным инструментам.

Как показывает детальное рассмотрение страниц большинства форумов, все множество сообщений, имен авторов и сопутствующей информации об авторах и сообщениях можно получить при помощи фильтрации с помощью довольно простых выражений XPath.

Для произвольного форума может использоваться неизвестное или нераспространенное программное обеспечение, при автоматизированном сборе информации в сети интернет пути для извлечения очищенных текстов и их атрибутов невозможно вручную подбирать выражение XPATH для каждого конкретного форума.

В данной работе предлагается алгоритм автоматического выделения группы однотипных вершин DOM-дерева, содержащих необходимую информацию, и построения для этой группы вершин выражения XPath, которым можно было бы воспользоваться для извлечения информации для подобных страниц этого же форума.

Обзор задачи удаления оформления

Схожей задачей является задача удаления обвязки (boilerpate) со страниц, с целью избавления от элементов веб-страниц, не несущих полезную информацию. Принципиально подходы для удаления обвязки можно разделить на два класса – методы страничного уровня и методы уровня сайта. Страничные методы обрабатывают каждую страницу по отдельности, не используя информацию об остальных страницах сайта. Методы уровня сайта для удаления обвязки собирают информацию со всех страниц сайта или с репрезентативного подмножества его страниц. Постраничный анализ применяется, когда нет возможности получить некоторое достаточное подмножество страниц сайта перед обработкой.

Интересный подход к решению проблемы удаления обвязки представляет алгоритм, описанный в работе Yi Lan и др. [5]. Алгоритм работает по принципу методов уровня сайта. В основе алгоритма лежит слияния множества DOM-деревьев страниц сайта в одно большое дерево стилей (style-tree). При этом одноименные и стоящие на одинаковых позициях в двух DOM-деревьях вершины и их последовательности сливаются в одну вершину стиля. Отбор шумовых элементов производится путем сравнения информационной энтропии, содержащейся в вершине стиля и порогового значения.

Алгоритм

Задача удаления элементов оформления со страниц является более общей по сравнению с выделением авторов и их сообщений на страницах форумов. Ключевым является факт того, что это форум и каждая страница содержит несколько (обычно не меньше 10) сообщений от различных пользователей. Чтобы воспользоваться данным преимуществом, было решено модифицировать алгоритм деревьев стилей следующим нижеописанным образом, адаптировав его для работы с отдельными страницами, а не с сайтом в целом. Дерево стилей, получаемое в результате объединения вершин, мы будем ниже называть редуцированным DOM-деревом. Акцентируем дальнейшие рассуждения на нахождении имени автора на странице, что является одной из главных целей.

Был совершен ручной анализ популярных русскоязычных форумов, таких как: rsdn.ru, 4pda.ru, gamedev.ru, sql.ru и прочие.

На основании полученной информации и известных фактах об устройстве веб-страниц и DOM-деревьев были получены следующие важные наблюдения, которые скорее всего будут применимы к большинству страниц форумов.

1) Вся информация о пользователе, включая его тексты, как правило, хранится в листовых вершинах DOM-дерева.
2) Имена пользователей отображаются на странице форума в виде гиперссылок на страницу профиля автора.
3) Пользователи либо самостоятельно предпочитают, либо вынуждены из-за ограничений форума использовать псевдонимы, содержащие только буквы английского алфавита и цифры.
4) Длина имен пользователей варьируется и довольно небольшая, со средним выборочным значением 8 и малой дисперсией.
5) За редким исключением, на одной странице содержится более 10 сообщений. Страницами, содержащими меньшее количество сообщений, можно пренебречь.

6) После некоторого уровня в DOM-дереве можно выделить вершину, наследниками которой будут являться идентичные по структуре поддеревья, содержащие сообщения авторов, информацию из профилей, онлайн статус, присвоенный «ранг» и т.п.
7) Число различных элементов, которые повторяются в различных сообщениях, должно быть значительное число, например, в каждом сообщении будет имя автора, ссылка на его профиль, его рейтинг, статус на форуме и прочие, и каждый такой элемент должен повторяться одинаковое количество раз, так как такие элементы входят в каждое сообщение.
8) Множество псевдонимов авторов не может содержать только один элемент и не может содержать все различные – должен соблюдаться баланс между ситуацией, когда человек беседует только сам с собой, и ситуацией, когда на странице отсутствует дискуссия, каждый отметился по разу.

Идея модификации алгоритма деревьев стилей заключается в следующем: сливать в одну вершину нелистовые вершины с одинаковым именем, а листовые вершины объединять в именные (листовые) группы. Однако стоит обратить внимание на то, что вершина может содержать несколько дочерних элементов с одинаковыми именами, которые могут являться различными элементами оформления и поэтому их не стоит объединять в одной вершине.

Для отсеивания последнего случая применяется следующая эвристика: алгоритм параметризуется значением нижнего порога количества сообщений на странице, далее происходит предобработка DOM дерева при помощи алгоритма обхода в ширину:

1) Для текущей обрабатываемой вершины производится подсчет дочерних вершин с одинаковыми именами.
2) Для каждой группы дочерних вершин с одинаковым именем производится оценка: если количество вершин ниже порогового (нижний порог количества сообщений на странице), то вершины переименуются в соответствии с их порядком следования.

Важно отметить, что данная методика будет работать только для случаев, когда поддеревья, содержащие непосредственно сообщения, имеют одинаковую линейную структуру для каждой вершины.

Пример предобработки изображен на рисунке 1. У вершины body только 2 дочерних элемента с именем table, и они будут переименованы, а дочерних элементов с именем div больше порогового, и они не будут переименованы.

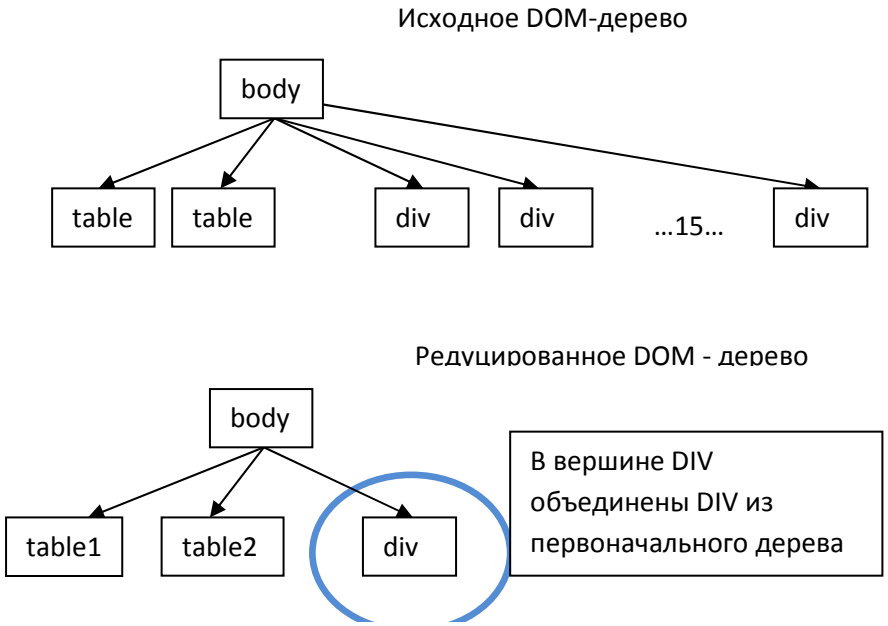

Рисунок 1. Предобработка DOM-дерева

После описанной выше предобработки следует этап слияния похожих верших. В отличие от оригинального алгоритма style-tree, будут сливаться не все вершины оригинального DOM-дерева, а только те, что содержатся в родительской вершине в количестве большем порогового.

Процесс слияния на произвольном шаге происходит следующим образом:

1) Если вершина с таким именем не встречалась среди уже рассмотренных подвершин рассматриваемой вершины, то мы выбираем ее в качестве базовой и создаем ее копию в редуцированном DOM-дереве.

2) Если вершина с таким именем уже встречалась – находится базовая вершина в редуцированном DOM-дереве, которой передаются все потомки текущей вершины.

3) Если вершина является листовой (то есть содержит текст), то листовая вершина прикрепляется к именованной группе вершин как дочерняя вершина. Именованная группа вершин выбирается таким образом, чтобы однотипные тексты, находящиеся в оригинальном DOM-дереве в разных вершинах, но на одних и тех же позициях, оказались объединены под одной вершиной.

После слияния получено редуцированное DOM-дерево, каждую вершину в котором необходимо оценить на предмет того, не содержат ли они все тексты, похожие на псевдонимы пользователей. Для этого необходимо проанализировать полученные листовые группы. Пользуясь предположением 7), интересующую нас информацию следует искать только в тех группах, которые встречаются ровно столько же раз, как и другие группы. Для

этого необходимо построить гистограмму, где каждый столбец отображает количество групп с одинаковым размером, нам необходимо выбрать самый высокий столбец.

После выбора набора групп с одинаковым размером следует отфильтровать все группы, кроме тех, что объединяют под собой текстовые вершины.

Дальнейший отбор стоит проводить с помощью простых правил и системы штрафов. К примеру: для отбора группы, содержащую имена авторов, стоит применить оценки, исходящие из того, что скорее всего родителем в DOM-дереве каждой вершины будет являться гиперссылка, будут содержаться преимущественно символы английского алфавита и конечно длина: порядка 5 – 10 символов.

Для каждого необходимого атрибута сообщения можно выделить свои наборы правил, выделяющие его главные характеристики. Для выделения текста автора можно сформулировать эвристики, сходные с теми, что применяются для автора: различие всех сообщений, средняя длина теста составляет 20–30 слов, и некоторые другие.

Тестирование

Для тестирования были выбраны следующие форумы: 4pda.ru, gamedev.ru, forums.goha.ru, sql.ru, forum.asus.ru.

В качестве объекта извлекаемой метаинформации было выбрано имя автора сообщения.

Для отбора лучшей группы были применены следующие правила:

1) группа набирает очко, если ее элемент длиннее 5 символов;
2) группа набирает очко, если ее элемент короче 10 символов.

Два первых правила рассчитываются для всех элементов группы, и полученные очки усредняются на длину группы;

3) чем больше различных элементов набирается в группе, тем больше очков она набирает.

Набор из трех простых правил дал значительный результат: на форумах gamedev.ru, forums.goha.ru, sql.ru, forum.asus.ru на страницах, где заведомо было больше 10 сообщений, алгоритм выбирал правильную группу вершин. На сайте 4pda.ru были выделены модели телефонов, указанные в профилях пользователей. Стоит учесть, что алгоритм отработает одинаково на всех страницах сайта: либо найдет всех авторов, либо найдет ни одного.

Заключение

Как показало тестирование – даже использование таких простых эвристик на этапе фильтрации вершин дает хороший результат. Для дальнейшего развития стоит выделить больше правил и возможно применить алгоритмы машинного обучения для подбора параметров правил.

Работы проводятся при финансовой поддержке Министерства образования и науки Российской Федерации в рамках выполнения ГК 07.514.11.4142 по теме «Разработка методов автоматического и полуавтоматического создания корпуса и подкорпусов современного русского языка на основе русскоязычного Интернета».

Список литературы

1. Document Object Model http://www.w3.org/TR/DOM-Level-2-Core/XPath http://www.w3.org/TR/xpath/

2. Org/w3c/dom http://docs.oracle.com/javase/7/docs/api/org/w3c/dom/package-summary.html

3. Pasternack J. and Roth D. Extracting article text from the web with maximum subsequence segmentation. In Proceedings of the 18th international conference on World wide web. – ACM, 2009. – P. 971-980.

4. Pomikalek Jan. Removing Boilerplate and Duplicate Content from Web Corpora. – Brno, 2011.

5. Yi L., Liu B., and Li X. Eliminating noisy information in web pages for data mining. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. – ACM, 2003. – P. 296-305.

Рецензенты:

Беликов В. И., д.ф.н., доцент, кафедра теоретической и прикладной лингвистики филологического факультета Московского государственного университета имени М. В. Ломоносова, г. Москва.

Гриненко М. М., д.ф.н., научный консультант, ООО «АбиИнфоПоиск», г. Москва.