ОБНАРУЖЕНИЕ СТРУКТУРНОГО ПОДОБИЯ HTML-ДОКУМЕНТОВ*

И. Некрестьянов, Е. Павлова
Санкт-Петербургский Государственный Университет, 198504, Россия,
Санкт-Петербург, Старый Петергоф, Университетский пр. 28
E-mails: nis@acm.org, katya@meta.math.spbu.ru
WWW: http://meta.math.spbu.ru

В работе исследуется возможность автоматического выявления HTML-документов подобной структуры. Эта информация, в частности, облегчает создание программ-посредников для извлечения слабоструктурированной информации.

Рассматриваемый подход основан на применении стандартных методов кластеризации. Основным предметом исследования является вид отображения информации о структуре документов в точки многомерного пространства, который позволяет получить наиболее качественный результат.

DISCOVERING STRUCTURAL SIMILARITY OF HTML-DOCUMENTS

I. Nekrestyanov, E. Pavlova
Saint-Petersburg State University, Universitetsky pr, 28, St.Petergoff,
St.Petersburg, 198504, Russia
E-mails: nis@acm.org, katya@meta.math.spbu.ru

This paper focus on the problem of identification of groups of HTML pages of similar structure. This information further can be used for automatic wrapper generation for semistructured data sources. Our approach is based on application of standard clustering techniques. Main research question is selection of best representation of structural properties of document in multidimensional space.

1. Введение

Веб – это, безусловно, одно из наиболее огромных и активно используемых хранилищ информации. Разнородность информационных потребностей пользователей Веб обуславливает одновременное существование различных подходов к организации доступа к информации - навигация по гиперссылкам, поисковые системы, тематические каталоги и т.д.

* Эта работа частично поддержана грантом РФФИ 01-01-00935.

Специфические особенности Веб обуславливают необходимость новых исследований в области организации доступа к информации в Веб. Например, за последние годы достигнут значительный прогресс в задаче создания текстовых поисковых систем для Веб [3].

Отметим, что многие новые методы поиска основаны на использовании не только информации о текстовом (тематическом) содержимом документов, но также пытаются использовать другую доступную информацию.

Одним из полезнейших источников такой информации является структура графа Веб, построенного на основе существующих гиперсвязей между страницами. Интуитивно ясно, что граф Веб не является случайным графом, так как огромное количество ссылок в этом графе создано вручную и отражает мнение создавшего их автора. Информация о структуре гиперссылок, например, является основой алгоритма ранжирования страниц *PageRank*, который успешно применяется в популярной поисковой системе Google.

Другим дополнительным источником информации является HTML-разметка документов. Например, эта информация может быть использована для вычисления значимости ключевых слов в зависимости от контекста их использования - так, слова, которые используются в заголовках, можно считать более важными [28]. Интересным использованием информации о разметки является автоматическое определение шаблона оформления страниц [9, 39].

Однако, методы поиска по текстовой информации далеко не всегда позволяют (даже потенциально) эффективно обнаружить искомую информацию. Например, текстовая поисковая система для Веб вряд ли поможет найти электронный книжный магазин, в котором можно купить нужную книгу по наиболее доступной цене.

Подобный запрос легко выражается средствами SQL в базах данных, однако традиционные базы данных работают с регулярной структурированной информацией и поэтому этот подход неприменим к Веб.

Тем не менее, в Веб доступно много информации, которая обладает некоторой определенной структурой (например, электронный каталог товаров) и возможность использования структурированных запросов для таких ресурсов могла бы быть очень полезной. Эти соображения стимулируют исследования в этой области организации доступа к такой информации, на которую принято ссылаться как на *слабоструктурированную* информацию [5, 1].

Традиционный подход состоит в извлечении структурированных данных из Веб и выполнении структурированных запросов уже над этими извлеченными данными. Этот подход реализуются на основе механизма *посредников* (wrappers) - программ, которые идентифицируют искомую

информацию в исходном документе и отображают ее в некоторый промежуточный формат [34, 31].

Разработка и поддержка посредников вручную очень трудоемкий процесс (в частности из-за разнородности и нерегулярности обрабатываемой информации) [30]. Поэтому уже несколько лет ведется работа над средствами автоматизации этой задачи.

Для этой цели предложен ряд языков описания программ-посредников, которые позволяют значительно снизить объем программного кода, который необходимо разрабатывать вручную [8,12].

Дальнейшая автоматизация ведется в направлении разработки алгоритмов, которые автоматически генерируют программы-посредники по результатам анализа набора документов, из которого предстоит извлекать информацию, и, возможно, множества примеров. Достижение полной автоматизации без использования обучающей информации в общем случае маловероятно, поскольку требуется извлечь не только вид схемы данных, но также и связанную с ним семантическую информацию (например, тип конкретного элемента данных) [19, 21, 35, 40, 27].

Одна из основных трудностей - это нерегулярность обрабатываемой информации. В частности, для автоматической генерации извлекающего информацию посредника необходимо подготовить набор документов, из которых посредник и будет извлекать информацию. Для успешного автоматического создания посредника необходимо, чтобы набор был относительно регулярным, т.е. страницы в этом наборе должны иметь схожую структуру в той части, которая содержит извлекаемую информацию. До сих пор существование этого набора предполагалось априори, и вопросы его создания не рассматривались.

Даже в случае создания посредников вручную информация о существующих группах страниц с похожей структурой может быть очень полезной для определения классов доступной информации и выбора множества посредников для разработки, которое позволит оптимизировать соотношение между уровнем покрытия информационного источника и затратами на его достижение.

В этой работе мы исследуем возможность автоматического выделения групп страниц с похожей структурой в приложении к страницам с HTML-разметкой.

В качестве основного механизма мы используем хорошо известные методы иерархической кластеризации. Методы кластеризации определяют группы схожих точек в многомерном пространстве, которые называются кластерами [23].

Применение этих статистических методов к прикладной задаче состоит в выборе способа отображения прикладных объектов (в нашем случае документов) в точки многомерного пространства. Вид этого отображения и определяет осмысленность получаемых результатов кластеризации.

Для оценки качества мы используем критерии, характеризующие однородность получаемых разбиений с точки зрения создания программ-посредников.

Статья организована следующим образом: в следующем разделе мы классифицируем предлагавшиеся подходы к созданию программ посредников, чтобы четче описать те из них, к которым применимы полученные нами результаты; в разделе 3 мы опишем рассматриваемый нами подход к выделению структурно схожих документов; далее, мы опишем методологию проведения экспериментов и полученные нами результаты.

2. Извлечение слабоструктурированной информации

Активные исследования в области работы со слабоструктурированной информацией привели к появлению большого количества альтернативных инструментов используемых для создания программ-посредников.

Для систематизации предлагавшихся подходов можно использовать следующую классификацию* [31]:

Специализированные языки. К этому классу относится большинство предлагавшихся инструментов «первой волны» - Minerva, TSIMMIS, Web-OQL, FLORID, Jedi [12, 8, 22, 25].

Вместо использования языков общего назначения, таких как Perl, C или Java, такие инструменты дают возможность использовать специализированный язык для описания посредника, что позволяет значительно снизить объем создаваемого вручную программного кода.

Использование HTML-разметки. Инструменты, относящиеся к этому классу, используют информацию о разметке обрабатываемых HTML документов. Например, анализируя дерево разбора, описывающее иерархию HTML-тэгов, можно пытаться автоматически (или полуавтоматически) генерировать правила для извлечения информации. К представителям этого класса относятся такие средства как W4F, XWRAP, RoadRuuner, Lixto [21, 38, 33, 13].

Работа с текстами на естественном языке. Инструменты, поддерживающие извлечение информации из текстов на естественном языке (т.е. без разметки или не используя ее), опираются на различные методы компьютерной лингвистики (например, определение частей речи и т.п.) для построения отношений между фразами и элементами предложений. Эти отношения потом используются для выведения правил для извлечения информации, которые идентифицируют искомую информацию внутри документа по синтаксическим и семантическим ограничениям.

---

* Отметим, что эта классификация не строгая и некоторые инструменты могут относиться сразу к нескольким группам.

Инструменты этого класса предназначены для работы с относительно грамматически связными текстами. Отметим, что регулярные в смысле структуры документы (например, таблица с прейскурантом цен) часто слабо связаны грамматически и в этом случае подобный подход демонстрирует низкую эффективность.

К представителям данной группы относятся, например, системы RAPIER, SRV и WHISK [20, 40].

Индуктивный подход. Основанные на индукции инструменты на основе анализа заданного множества тренировочных документов, для которых известно расположение искомой информации, выводят правила извлечения, которые основываются на разделителях (нетекстовых маркерах, таких как знаки препинания или тэги HTML).

При этом используется не информация о каких-либо лингвистических свойствах текста или его иерархической структуре разметки, а лишь его форматирующие свойства.

Подобные подходы используется, например, в системах WIEN, SoftMealy, STALKER [29, 24, 36].

Моделирование искомой информации. Эта категория включает инструменты, которые по заданной структуре искомых объектов информации пытаются обнаружить в документах кусочки информации, соответствующие этой структуре. Для описания структуры используется множество моделирующих примитивов (например, записи, списки и т.п.), которые соответствуют лежащей в основе модели данных.

Представителями этого класса являются системы NoDoSE и DEByE [6, 32].

Использование онтологии. Подходы этой группы опираются не столько на лингвистические или структурные особенности в окрестности, где расположена искомая информация, сколько на сами данные. Для этой цели можно использовать онтологию, описывающую конкретную предметную область.

Наиболее известным представителем такого подхода является система, разработанная в Brigham Young University [16].

В этом исследовании мы в первую очередь ориентируемся на подходы, использующие информацию об HTML-разметке и специализированных языках, хотя, вероятно, группировка по структурному подобию документов может оказаться полезной и для индуктивных подходов или подходов, моделирующих искомую информацию.

3. Обнаружение структурного подобия

Рассматриваемая нами задача состоит в поиске разложения множества HTML-документов $D = \{d_1,...,d_n\}$ на классы $C_1,...,C_k$, которые содержат документы со схожей структурой.

При этом нас интересует не только и не столько полное совпадение, а схожесть структуры с точки зрения возможности применения одной программы-посредника для извлечения информации. Так, документы, которые отличаются только числом строк в таблице, должны быть очень схожи.

С другой стороны, даже изменения вне той части документа, из которой будет извлекаться информация не должны оказывать существенного влияния на результат. Например, отличия типа «другой баннер» не существенны для посредников.

Отметим, что информация о том, какая именно часть документа содержит искомую информацию, нам заранее не известна.

Вообще говоря, схожесть иерархических структур изучается в разных контекстах, но почти во всех случаях существуют значительные особенности [10, 15, 37, 42].

3.1 Кластеризация

Задача кластеризации - это довольно хорошо известная статистическая задача, целью которой является выделение классов близко расположенных точек в многомерном пространстве. К настоящему моменту известно много различных методов кластеризации и исследования в этой области продолжаются [14, 17, 18].

Одной из серьезных проблем большинства известных методов кластеризации является необходимость указания количества ожидаемых кластеров или максимально допустимого размера кластера в качестве параметра. На практике, это не всегда осуществимо и на данный момент неизвестно как в общем случае эти параметры можно выбирать автоматически.

Тем не менее, кластеризация широко используется при решении различных прикладных задач, в которых рассматривается большой объем информации. В частности, она активно используется при решении задач по организации доступа к текстовой информации [11, 41].

При использовании методов кластеризации в решении прикладной задачи критическую роль играет способ отображения прикладных объектов (в нашем случае документов) в точки многомерного пространства.

Поэтому мы остановились на использовании метода классического агломеративно-иерархического метода кластеризации. Основная идея этого метода заключается в последовательном объединении группируемых объектов - сначала самых близких, затем все более удаленных друг от друга [23].

Такой подход позволяет нам сфокусироваться на исследовании методов представления документов в многомерном пространстве и в то же время избавиться от погрешности из-за неудачного выбора параметров для метода кластеризации (например, ожидаемого числа кластеров).

Использование иерархического подхода позволяет производить оценку качества разбиения на каждом из построенных уровней и произво-

дить сравнение разных методов отображения, рассматривая только результат на лучшем уровне.

3.2 Рассматриваемые отображения

Задание отображения прикладных объектов в точки многомерного пространства состоит в определении базиса признаков $\{e_i\}$, формирующих многомерное пространство, и метода разложения документа по этому базису (т.е. вычисления координат $\{\omega_i\}$).

Рассматривавшиеся методы задания этих двух составляющих мы и опишем в следующих двух подразделах.

3.2.1 Базисные свойства

Когда речь идет о кластеризации документов, наиболее часто используемыми признаками, безусловно, являются ключевые слова или фразы. Однако эти признаки характеризуют тематическое содержимое документа, а не его структуру, и поэтому не применимы в нашем случае.

Для того чтобы иметь возможность характеризовать структурные свойства документа, мы рассматриваем* его в виде дерева разбора согласно стандартной объектной модели представления документов DOM [4].

Корнем DOM-дерева является тэг `html`. Внутренние узлы дерева соответствуют другим используемым в документе тэгам, дуги между которыми характеризуют вложенность их использования.

От него выходят ветки, состоящие из его подэлементов, из них, в свою очередь выходят ветви, состоящие из их подэлементов, и так далее. Листья дерева могут быть не только тэгами, но также и представлять текстовые литералы. Пример фрагмента DOM-дерева изображен на рисунке 1.

Мы рассмотрели следующие альтернативные наборы базисных признаков $\{e_i\}$:

A. Тип тэга.

Каждый тип тэга соответствует одному признаку. Все текстовые литералы считаются относящимися к одному дополнительному искусственному типу.

B. Тип тэга и его атрибуты.

Признак определяется по имени тэга и набору использовавшихся в нем атрибутов. Т.е. `<BODY>` и `<BODY BGCOLOR=«...»>` будут соответствовать разным признакам.

C. Входящий путь длины *k*.

Для каждой вершины DOM-дерева рассматривается последовательность из не более чем *k* ее предков. Множество таких последовательностей и формирует множество признаков.

---

* Это, в частности, подразумевает использование синтаксически корректного HTML со вложенной структурой тэгов (т.е. по сути XHTML).

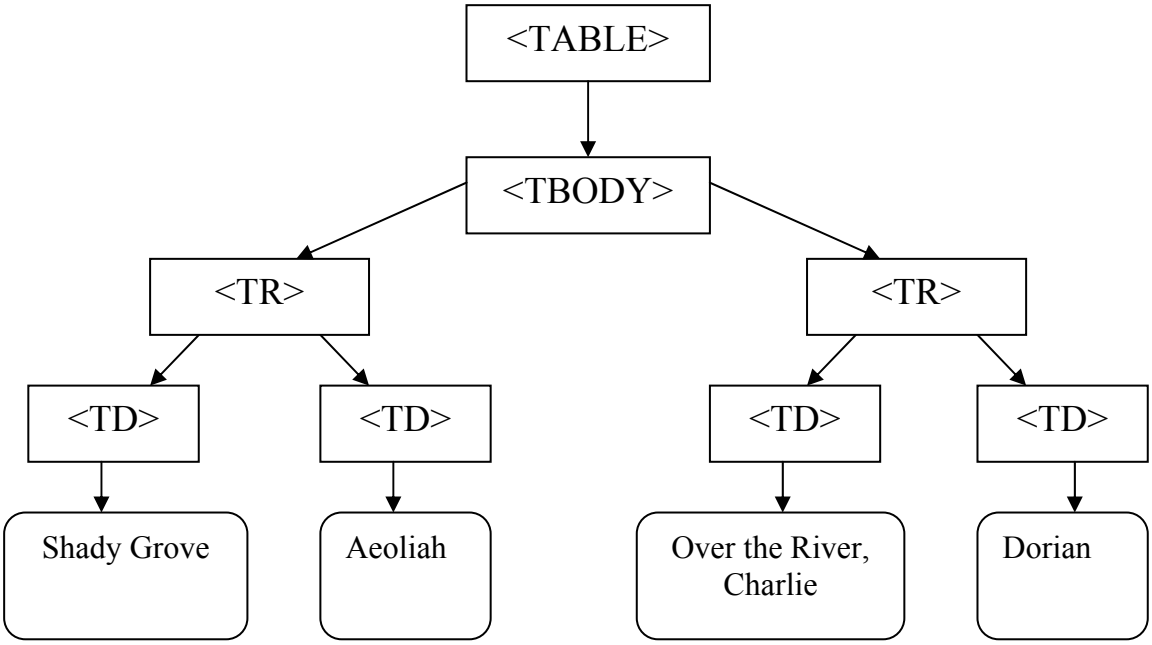

Рис. 1: Пример DOM-дерева для фрагмента HTML-документа

D. Тэг и его потомки.

Признак представляет собой поддерево исходного DOM-дерева некоторой фиксированной высоты l. При сравнении таких деревьев используется информация только об их структуре и о типах тэгов, соответствующих сравниваемым вершинам.

E. Тэг и его потомки с учетом информации об атрибутах.

Как и в предыдущем случае, но при сравнении вершин учитывается не только информация о типе соответствующих тэгов, но и наборах использованных атрибутов.

F. Комбинированный подход.

Признаком является объединение входящего пути длины k и дерева потомков некоторой заданной глубины l.

Основные структурные шаблоны соответствующие этим типам признакам проиллюстрированы на рисунке 2.

Многие из этих подходов являются обобщениями других и формально можно говорить, что мы рассматриваем параметризованное семейство базисных признаков $Basis(k,l,attr)$. Отметим, что более общие подходы означают расширение размерности пространства признаков и, как следствие, рост вычислительной трудоемкости задачи кластеризации.

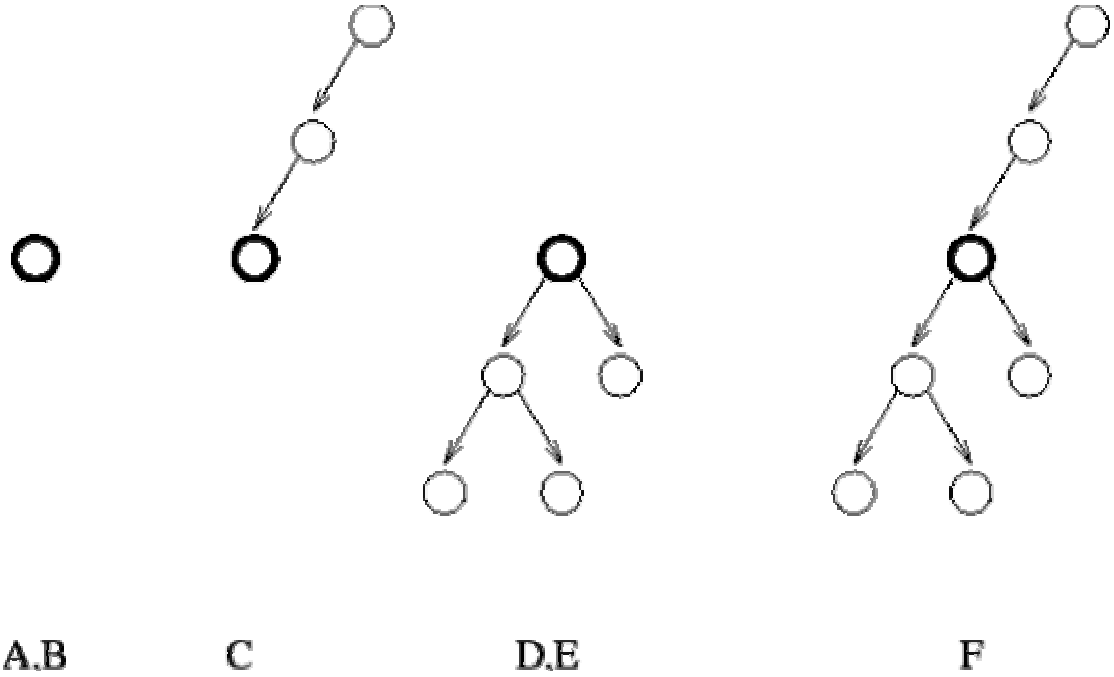

Рис. 2: Типы рассматриваемых признаков (информация об атрибутах не показана)

3.2.2 Вычисление координат

Для определения координат документа $\{w_i\}$ в пространстве базисных признаков $\{e_i\}$ использовались следующие подходы:

- **Наличие признака** *(bool)***.**

Если соответствующая *i*-му признаку структура встречается в DOM-дереве документа, то $w_i = 1$. Иначе $w_i = 0$.

- **Число использований признака** *(cnt)***.**

Координата w_i равна количеству вхождений соответствующей признаку структуры в документ N_i.

- **Частота встречаемости признака (***freq***).**

Это нормированный вариант предыдущего представления. В этом случае число использований признака нормируется так, чтобы сумма координат для одного документа равнялась 1.

- **TFIDF**

Координата признака определяется с учетом его частоты встречаемости в этом документе, но также и обратно пропорционально его распространенности в других документах. Подобный подход очень популярен при вычислении весов термов в поисковых системах, использующих векторную модель представления документов. Формально:

$$w_i = \frac{tf_i}{\log\left(\frac{N}{k_i}\right)}$$

где tf_i - это частота встречаемости i-го признака, k_i - количество документов, в которых он встречается, а N - общее количество рассматриваемых документов.

4. Методология оценки

Для проведения оценки необходимо определить не только предмет оценки (в нашем случае результат кластеризации), но также критерии, меры и инструменты измерения для вычисления количественных оценок [2].

4.1 Критерии качества

Для оценки качества кластеризации можно использовать разные подходы, но наиболее практически полезным является сравнение с идеальным разбиением [7,26].

Однако в случае последующего использования обнаруженных кластеров для создания посредников можно сформулировать более конкретные критерии. Действительно, целью оцениваемого процесса является создание посредников для некоторого набора страниц - чем больше страниц удается обработать автоматически, тем лучше.

Это приводит нас к понятию «хорошего» кластера, т.е. кластера по которому можно автоматически создать одного посредника, который бы мог извлекать информацию из всех документов, которые относятся к этому кластеру.

Более формально, кластер «хорош» если:

1. Все документы в этом кластере обрабатываются одним и тем же посредником (который можно автоматически сгенерировать).

2. Кластер достаточно велик для того, чтобы процедура генерации посредника могла с ним работать (т.е. размер кластера ≥ некоторого ε (обычно $\varepsilon \geq 2$)).

Собственно процедура обработки набора документов после выполнения кластеризации может выглядеть следующим образом: для каждого из еще не обработанных документов мы находим такого его предка в иерархии, для которого можно построить посредника автоматически и обрабатываем рассматриваемый документ вместе с другими документами, относящимися к этому кластеру.

4.2 Меры

Типичной мерой для оценки качества кластеризации является энтропия - чем меньше энтропия, тем лучше результат кластеризации.

Для кластера j значение энтропии вычисляется по следующей формуле:

$$E_j = -\sum_i p_{ij} \log\left(p_{ij}\right),$$

где p_{ij} - вероятность того, что элемент кластера *j* относится к классу *i*.

Значение же общей энтропии вычисляется по формуле:

$$E = \frac{1}{n}\sum_j n_j * E_j ,$$

где n_j - мощность кластера *j*, а *n* - общее число кластеризуемых элементов.

Для получения значений p_{ij} используется информацию о том, к какому классу действительно относится рассматриваемый документ.

Отметим, что вышеприведенные формулы рассчитаны на оценку качества иерархической кластеризации на отдельном уровне.

К сожалению, нулевая энтропия кластера не означает, что соответствующий кластер «хорош», а мера общей энтропии слабо связана с количеством страниц, которые удастся обработать автоматически. Поэтому мы воспользовались собственными мерами.

Формально, критерий «хорошести» кластера можно описать следующим образом:

$$Good_j = \begin{cases} 1, \textit{если } E_o = 0 \textit{ и } \left\| C_j \right\| \geq \varepsilon, \\ 0, \textit{иначе} \end{cases} \quad (1)$$

Тогда, аналогичный общей энтропии критерий, который определяет долю документов, которую удастся обработать автоматически при таком разбиении следующим образом:

$$W\,Score_j = \frac{1}{n}\sum_j n_j * Good_j$$

Однако, эта мера также не идеальна. В случае если представления двух документов, которые могут быть обработаны одним и тем же посредником, отличаются больше, чем два уже обнаруженных разных «хороших» кластера, то к тому уровню иерархии, на котором эти два документа сольются в общий кластер, будут слиты и ранее обнаруженные «хорошие» кластера.

Вообще говоря, описанная в предыдущем разделе процедура построения посредников по результатам кластеризации, не подразумевает того, что все кластера, которым сопоставляются посредники, должны располагаться на одном уровне иерархии. Это дает нам повод определить еще одну метрику:

$$W\ Score_j = \frac{1}{n}\sum_{i \in SelectedSet} n_j * Good_j$$

где множество выбранных кластеров определяется следующим образом

$$SelectedSet = \left\{ C_j : Good_j = 1 \,\&\, \left(\forall j' : j \subset j' \Rightarrow Good_{j'} = 0 \right) \right\}$$

4.2.1 Сбор информации для оценки

Для того чтобы применить меры из предыдущего раздела, необходимо иметь информацию о классах документов. При этом документы считаются относящимися к одному классу, если они могут быть успешно обработаны одним и тем же (автоматически созданным) посредником.

Прямолинейный подход к сбору этой информации состоит в применении процедур автоматического создания посредников к результатам кластеризации, как это и было описано в разделе 4.1. К сожалению, нам не удалось найти ни одной свободно доступной реализации предлагавшихся процедур построения посредников.

Поэтому мы попытались апроксимировать эти информацию о классах при помощи создания посредников вручную. Из-за большого объема данных мы, конечно же, не могли реализовать всех посредников вручную, но это и не было нашей целью. Реализовав несколько посредников, мы использовали информацию о симптомах их падения при попытке обработать несоответствующие им ресурсы для того, чтобы выделить классы среди ресурсов, для которых отсутствовал посредник.

Кроме очевидной возможной погрешности с определением таких классов (они могут требовать дальнейшего разложения на подклассы), создание посредников вручную могло повлечь использование сложных конструкций, которые не применяются при автоматическом построении посредников. Как следствие, даже те классы документов, которые удачно обрабатывались одним и тем же созданным вручную посредником, вполне возможно, требовали дальнейшего разложения для успешного применения автоматических генераторов посредников.

Однако, несмотря на эти погрешности, собранная информация, очевидно, коррелирует с информацией об искомых классах, и поэтому может быть использована для проведения оценки.

5. Экспериментальное сравнение

Целью проводимых экспериментов являлось определение представлений позволяющих получить наилучшую кластеризацию.

5.1 Наборы данных

Мы рассматриваем несколько разных, публично доступных наборов данных для того, чтобы проверить справедливость наблюдаемых закономерностей в разных окружениях.

5.1.1 List.Ru

Данные, содержащиеся в тематическом каталоге ресурсов *List.Ru* (http://www.list.ru), имеют довольно простую структуру - перечень ресур-

сов с комментариями, но сами страницы также содержат много прочей нерелевантной информации - реклама, прогноз погоды и т.п.

Мы использовали набор из 30000 документов, который содержал 10 классов ресурсов. Однако из-за большого количества базисных признаков и, как следствие, высокой трудоемкости метода кластеризации, при проведении экспериментов одновременно использовались наборы размером от 1000 до 3000 страниц, представляющие 5-11 классов.

5.1.2 IMDB

Сайт Internet Movie Database (http://imdb.org) - довольно популярный источник данных для тестирования различных подходов к извлечению слабоструктурированных данных.

Это сложный ресурс, который содержит много (34 класса) разнообразной структурированной информации - фильмографии актеров и режиссеров, информация о фильмах, и т.п. Более того, эта информация довольно часто комбинируется в рамках одной и той же HMTL-страницы, которая к тому же может содержать массу нерелевантной информации.

5.2 Результаты

Очевидно, что описанное в разделе 3.2.1 в параметризованное семейство множеств базисных свойств $Basis(k,l,attr)$ входит огромное количество элементов и перебор всех возможных значений не реалистичен. К тому же количество вариантов еще больше возрастает при варьировании метода вычисления весов.

Для проведения систематического сравнительного анализа было решено исследовать в первую очередь поведение качества кластеризации при изменении одного из параметров *k* или *l* при различных схемах вычисления весов как с использованием информации об атрибутах, так и без такового.

Информация о некоторых типичных случаях представлена в табл. 1:

Параметры эксперимента				*W Score*	Размерность Пространства
k	*l*	атрибуты	вес		
0	0	нет	*bool*	0	22
1	0	нет	*bool*	0,078	52
0	1	нет	*bool*	0,893	261
0	0	нет	*cnt*	0,933	22
0	0	нет	*tfidf*	0,945	22
2	0	нет	*tfidf*	0,959	66
2	0	да	*tfidf*	0,957	106
∞	0	да	*tfidf*	0,904	209

Таблица 1: Некоторые результаты эксперимента по кластеризации с набором List.Ru $(\varepsilon = 3)$.

По результатам наших экспериментов можно сформулировать следующие наблюдения:

- Расширение признаков вниз (рост *l*) в большинстве случаев ухудшает качество результатов.
- Использование входящих путей в качестве признаков дает положительный эффект. Наилучшее качество получается при длине пути *k=2*.
- Использование информации только о наличии признака стабильно проигрывает более сложным схемам взвешивания. Большинство наилучших результатов было достигнуто при использовании схемы взвешивания *tfidf*.
- Эффект от использования информации об атрибутах незначителен и зачастую вызывает снижение качества результатов. К тому же это часто вызывает значительный рост размерности и, как следствие, вычислительной трудоемкости.

6. Заключение

В работе рассматривается актуальная проблема автоматического выделения групп документов с похожей структурой HTML-разметки с целью облегчения автоматизации задачи создания программ-посредников для извлечения слабоструктурированной информации.

Исследовано несколько альтернативных способов отображения информации о разметке документов в точки многомерного пространства. Для оценки эффективности подходов используется характеристика однородности получаемых разбиений с точки зрения соответствия программам посредникам.

Проведение экспериментов на нескольких разных публично доступных тестовых наборах данных позволяет надеяться на обоснованность наблюдаемых результатов в приложении к другим наборам данных.

В дальнейшем мы планируем реализовать процедуру построения программ-посредников и изучить эффективность предлагаемого подхода в реальных условиях.

Благодарности

Мы хотели бы поблагодарить Никиту Зиновьева за помощь при проведении оценки результатов экспериментов.

Литература

[1] Д. Барашев, А. Высоцкий, С. Кукс, Е. Михайлова, И. Некрестьянов, Б. Новиков и Е. Павлова. Интеграция публично доступных архивов списков рассылки. В *«Трудах третьей всероссийской научной конференции Электронные библиотеки»*, Петразаводск, Россия, Октябрь 2001.

[2] И. Кураленок и И. Некрестьянов. Оценка систем текстового поиска. Программирование (в печати), 2002.

[3] И. Некрестьянов и Н. Пантелеева. Системы текстового поиска для Веб. Программирование (в печати), 2002.

[4] Document Object Model (DOM) Level 2 HTML Specification. W3C Working Draft, December 2001.

[5] Serge Abiteboul, Peter Buneman, and Dan Suciu. *Data on the Web*. Morgan Kaufmann Publishers, 1999.

[6] Brad Adelberg. NoDoSE-a tool for semi-automatically extracting structured and semistructured data from text documents. *SIGMOD Record*, 27(2):283-294, 1998.

[7] Seannie Alon, Noga nad Dar, Michal Parnas, and Dana Ron. Testing of clustering. In *Proc. of the Annual Symposium on Foundations of Computer Science (FOCS)*, 2000.

[8] G. O. Arocena and A. O. Mendelzon. WebOQL: Restructuring Documents, Databases, and Webs. In *Proc. of the 14th IEEE International Conference on Data Engineering*, pages 24-33, Orlando, Florida, 1998.

[9] Ziv Bar-Yossef and Sridhar Rajagopalan. Template detection via data-mining and its applications. In *Proc. of the WWW'2002*, May 2002.

[10] E. Bertino, G. Guerrini, M. Mesiti, I. Rivara, and C. Tavella. Measuring the structural similarity among XML documents and DTDs. Technical Report DISI-TR-02-02, Universita` di Genova, December 2001.

[11] M. Charikar, C. Chekuri, T. Feder, and R. Motwani. Incremental clustering and dynamic information retrieval. In *Proceedings of the 29th Symposium on Theory of Computing*, 1997.

[12] V. Crescenzi and G. Mecca. Grammars have exceptions. *Information Systems, Special Issue on Semistructured Data*, 1998.

[13] Valter Crescenzi, Giansalvatore Mecca, and Paolo Merialdo. Roadrunner: Towards automatic data extraction from large web sites. In *The VLDB Journal*, pages 109-118, Rome, Italy, 2001.

[14] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering in large graphs and matrices. In *Proceedings of the ACM-SIAM SODA'1999*, 1999.

[15] Ingvar Eidhammer, Inge Jonassen, and William R. Taylor. Structure comparison and structure patterns. Technical report, University of Bergen, July 1999.

[16] David W. Embley, Douglas M. Campbell, Y. S. Jiang, Stephen W. Liddle, Yiu-Kai Ng, Dallan Quass, and Randy D. Smith. Conceptual-model-based data extraction from multiple-record web pages. *Data Knowledge Engineering*, 31(3):227-251, 1999.

[17] Frederik Farnstrom, James Lewis, and Charles Elkan. Scalability for clustering algorithms revisited. *ACM SIGKDD Explorations*, 2(1):51-57, August 2000.

[18] Daniel Fasulo. An analysis of recent work on clustering algorithms.

[19] D. Freitag and N. Kushmerick. Boosted wrapper induction. In *Proc. of the AAAI-2000*, 2000.

[20] Dayne Freitag. Machine learning for information extraction in informal domains. *Machine Learning*, 39(2/3):169-202, 2000.

[21] S. Grumbach and G. Mecca. In search of the lost schema. In *Proc. of International Conference on Database Theory*, 1999.

[22] J. Hammer, J. McHugh, and H. Garcia-Molina. Semistructured data: The tsimmis experience. In *Proc. of the First East-European Symposium on Advances in Databases and Information Systems (ADBIS'97)*, pages 1-8, St. Petersburg, Russia, 1997.

[23] Jiawei Han and Micheline Kamber. *Data Mining: Concept and Techniques*. Morgan Kaufmann Publishers, 2001.

[24] Chun-Nan Hsu and Ming-Tzung Dung. Generating finite-state transducers for semi-structured data extraction from the web. *Information Systems*, 23(8):521-538, 1998.

[25] Gerald Huck, Peter Fankhauser, Karl Aberer, and Erich J. Neuhold. Jedi: Extracting and synthesizing information from the web. In *Proc. of the 3rd IFCIS International Conference on Cooperative Information Systems*, pages 32-43, New York City, New York, 1998.

[26] Ravi Kannan, Santosh Vempala, and Andrian Vetta. On clusterings - good, bad and spectral.

[27] Raymond Kosala, Ian Van den Bussche, Maurice Bruynooghe, and Hendrik Blockeel. Information extraction in structured documents using tree automata induction. In *Proc. of the PKDD'2002*, 2002.

[28] U. Kruschwitz. Exploiting Structure for Intelligent Web Search. In *Proceedings of the 34th Hawaii International Conference on System Sciences (HICSS)*, Maui, Hawaii, 2001. IEEE.

[29] Nicholas Kushmerick. Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence*, 118(1-2):15-68, 2000.

[30] Nicholas Kushmerick. Wrapper verification. *World Wide Web*, 3(2):79-94, 2000.

[31] Alberto H. F. Laender, Berthier A. Ribeiro-Neto, Altigran S.da Silva, and Juliana S. Teixeira. A brief survey of web data extraction tools. In *SIGMOD Record, To appear*, 2002.

[32] Alberto H. F. Laender, Berthier A. Ribeiro-Neto, Altigran S. da Silva, and Juliana S. Teixeira. DEByE - data extraction by example. *Data and Knowledge Engineering*, 20(2):121-154, 2002.

[33] Ling Liu, Calton Pu, and Wei Han. XWRAP: An XML-enabled wrapper construction system for web information sources. In *Proc. of the ICDE*, pages 611-621, San Diego, California, 2000.

[34] Ion Muslea. Extraction patterns for information extraction tasks: A survey. In *Proc. of the AAAI-99 Workshop on Machine Learning for Information Extraction*, 1999.

[35] Ion Muslea, Steve Minton, and Craig. A. Knoblock. A hierarchical approach to wrapper induction. In *Proc. of the 3rd International Conference on Autonomous Agents*, Seattle, WA, 1999.

[36] Ion Muslea, Steven Minton, and Craig A. Knoblock. Hierarchical wrapper induction for semistructured information sources. *Autonomous Agents and Multi-Agent Systems*, 4(1/2):93-114, 2001.

[37] Thorsten Richter. A new measure of the distance between ordered trees and its applications. Technical Report 85166-CS, University of Bonn, 1997.

[38] Arnaud Sahuguet and Fabien Azavant. Building intelligent web applications using lightweight wrappers. *Data Knowledge Engineering*, 36 (3): 283-316, 2001.

[39] H. Sakamoto, Y. Murakami, H. Arimura, and S. Arikawa. Extracting partial structures from html documents. In *Proc. the 14the International FLAIRS Conference*, pages 264-268, 2001.

[40] Stephen Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1-3):233-272, 1999.

[41] Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques. In *TextMining Workshop, KDD*, 2000.

[42] Jason Tsong-Li. Treediff: A system for document comparison by structure, 1997.