Использование интеллектуальных сетевых роботов для построения тематических коллекций*

Романова Е.В., Романов М.В., Некрестьянов И.С.
Санкт-Петербургский Государственный Университет, Санкт-Петербург.
emails: katya@tepkom.ru, rmv@sparc.spb.su, igor@meta.math.spbu.ru

Abstract

В работе рассматривается задача создания интеллектуального сетевого робота для сбора тематических коллекций. Для повышения производительности обнаружения тематических ресурсов используется специализированный алгоритм обхода сети, учитывающий информацию о тематической релевантности уже посещенных страниц. Робот также производит грубый отсев "мусора" среди посещенных документов, для того чтобы повысить качество рекомендаций.

1 Введение

В течение ряда лет вопросы создания и применения сетевых роботов привлекают все больше внимания [8, 10, 13, 11]. Сетевой робот или *Crawler* — это программа, которая, начиная с некоторой Интернет-страницы, рекурсивно обходит ресурсы Интернет, извлекая ссылки на новые ресурсы из получаемых документов.

Классической областью применения сетевых роботов является построение индексов Интернет-ресурсов для поисковых систем [14, 3, 5, 15]. Однако в последнее время сетевые роботы используются для выполнения множества других задач — сбора статистики, поиска определенных ресурсов сети (например, домашних страниц), проверки целостности существующих гипертекстовых ссылок, и т.п. Разработаны даже соответствующие правила "вежливого" поведения для сетевых роботов — Standard for Robot Exclusion и Rapid Fire Requests. Текущий вариант списка добровольно зарегистрированных роботов на странице `info.webcrawler.com` содержит более сотни позиций, а общее число существующих сетевых роботов по некоторым оценкам превышает десятки тысяч.

Большинство сетевых роботов посещают огромное количество Internet страниц, индексируя все полученные документы. Очевидно, что такой подход требует значительных сетевых и аппаратных ресурсов. Однако текущий объем доступной информации в Интернет оценивается в 6 терабайт и быстро растет, поэтому даже самый мощный сетевой робот не может посетить все Интернет-страницы.

Поскольку посещение всех Интернет-страниц не представляется возможным, то разумно посещать в первую очередь наиболее важные из них. Простейший критерий важности, используемый многими из современных сетевых роботов собирающими информацию для популярных поисковых систем, является глубина URL, т.е. количество промежуточных каталогов упоминающихся в URL между именем Интернет-узла и именем самого ресурса. Чем больше глубина, тем ниже важность соответствующего ресурса. Подобный подход позволяет быстро посетить стартовые и близкие к ним страницы на большом числе Интернет-узлов.

Гораздо более продвинутая стратегия используется в сетевом роботе поисковой системы Google [14], созданной в Стенфордском университете.

Интуитивно кажется очевидным, что страница, на которую ссылаются много различных страниц в Интернет, более важная, чем та, на которую мало ссылок. А также, что ссылку со страницы *Yahoo!* или *List.Ru* стоит оценивать выше, чем ссылку с чьей-то персональной страницы.

Эти соображения и используются в алгоритме [9] сетевого робота Google, согласно которому более важными считаются такие URL, на которые больше ссылок из других, наиболее часто цитируемых страниц в Интернет. Такой подход направлен на максимизацию количества обнаруженных наиболее часто используемых ресурсов.

В данной работе мы рассматриваем вопросы применения сетевых роботов при построения тематической коллекции. Мы столкнулись с этой задачей во время работы над проектом OASIS[1], посвященным разработке открытой распределенной архитектуры системы для поиска по множеству тематических коллекций [4].

Отметим, что проблема построения тематических коллекций не является специфичной для проекта OASIS и актуальна во многих других задачах информационного рынка, например, таких как построение тематических каталогов типа *Yahoo!* или *List.Ru*.

В рамках проекта OASIS был разработан сетевой робот (OASIS Crawler), главной целью которого является содействие при создании тематической коллекции. Мы полагаем, что наиболее важной задачей такого робота является обнаружение максимального количества темати-

*Эта работа была выполнена в рамках проекта Open Architecture Server for Information Search and Delivery (OASIS) и поддержана грантом Европейской комиссии (INCO Copernicus Programme Project PL 961116).

Первая Всероссийская научная конференция
ЭЛЕКТРОННЫЕ БИБЛИОТЕКИ:
ПЕРСПЕКТИВНЫЕ МЕТОДЫ И ТЕХНОЛОГИИ,
ЭЛЕКТРОННЫЕ КОЛЛЕКЦИИ
19 - 21 октября 1999 г., Санкт-Петербург

[1] Дополнительная информация по проекту OASIS доступна по адресу `www.oasis-europe.org`

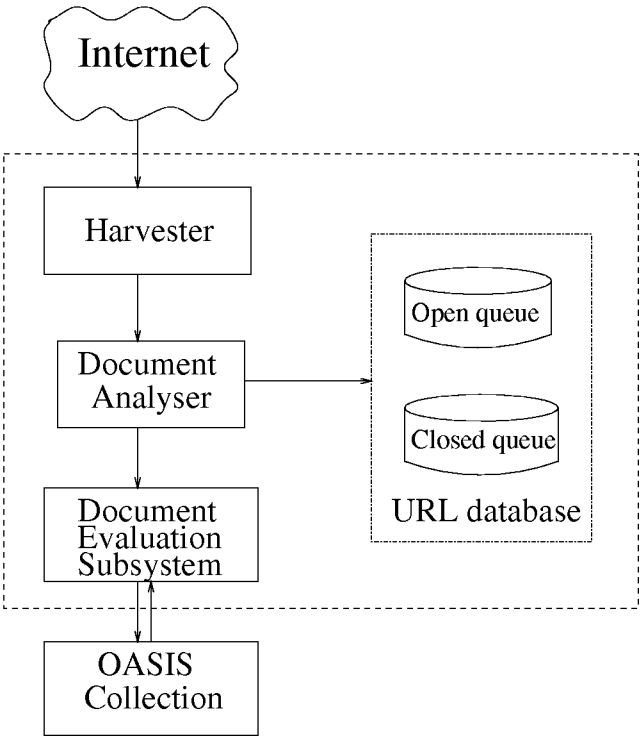

Рис. 1: Архитектура сетевого робота OASIS

чески релевантных ресурсов. Для выполнения этой задачи робот учитывает информацию о тематической релевантности уже обнаруженных страниц для определения дальнейшего порядка обхода.

Мы предполагали, что окончательное решение о тематической релевантности обнаруженного ресурса производится клиентом нашего робота (в дальнейшем мы будем называть этого клиента *Коллекцией*). Однако робот может рекомендовать Коллекции не все обнаруженные им ресурсы, а произвести грубый предварительный отсев для повышения качества его рекомендаций.

В дальнейшем статья организована следующим образом. В следующем разделе описывается базовая архитектура сетевого робота. В разделе 3 описывается структура используемого тематического фильтра и методы вычисления оценки тематической релевантности документа. Следующий раздел содержит описание алгоритма автоматического улучшения тематического фильтра. Использование тематической информации для выбора стратегии обхода обсуждается в разделе 5. Последний раздел посвящен результатам наших экспериментов с роботом.

2 Архитектура сетевого робота OASIS

Основными подсистемами сетевого робота OASIS, изображенными на рисунке 1, являются подсистемы сбора, анализа и оценки релевантности документов, а также база данных известных URL.

- Подсистема сбора документов, Harvester, отвечает за эффективное скачивание документов с HTTP или FTP серверов. При этом подсистема следует "этике сети", то есть избегает слишком частых обращений к серверу (Rapid Fire Request) и удовлетворяет протоколу для сетевых роботов (Standard for Robot Exclusion) [11].

  Цикл работы подсистемы сбора документов состоит в следующем: Crawler получает очередную ссылку из очереди непосещенных URL-ей, и скачивает соответствующую WWW-страницу, которая передается в подсистему анализа документов.

- Подсистема анализа документов, Document Analyser, ответственна за разбор HTML-документов.

  Из входящего документа удаляются все HTML-тэги и стандартные стоп-слова [2], а затем выполняется отбрасывание окончаний в остальных словах (выделяется основа слова). В результате создается профайл документа, представляющий из себя вектор из пар: терм, т.е. основа слова, и частота использования этого терма в данном документе. Этот профайл далее используется для вычисления "грубой" оценки релевантности рассматриваемого документа.

  Во время разбора подсистема анализа документов также добавляет все найденные на странице URL в очередь для последующего посещения.

  Чтобы избежать зацикливания, производится проверка на предмет того, что каждый из обнаруженных URL еще не посещен (т.е. нет ли его в закрытой очереди), а также не встречается ли он уже в очереди URL для последующего посещения (открытой очереди). Только прошедшие проверки URL попадают в открытую очередь. Информация о посещенном документе, в том числе его URL, добавляется в закрытую очередь URL.

  После анализа профайл документа передается подсистеме оценки релевантности.

- У подсистемы оценки релевантности документов две основных задачи:

  - Отфильтровывание мусора, то есть уменьшение количества нерелевантных документов, рекомендуемых Коллекции,
  - "грубое" вычисление оценок релевантности документов для последующего использования при выборе способа обхода известных Интернет страниц.

  Данная подсистема использует тематический фильтр для вычисления "грубой" оценки релевантности документа (см. раздел 3).

- База данных URL содержит информацию о всех посещенных (закрытая очередь) и об известных роботу не посещенных ссылках (открытая очередь).

  Элемент закрытой очереди хранит следующую информацию о посещенных документов: URL, дату последнего обновления, объем документа, а также "грубую" оценку релевантности документа, определенную самим роботом, и "точную" оценку релевантности документа, вычисленную Коллекцией.

[2] Стоп-слова — наиболее употребительные слова, не имеющие никакой тематической направленности, например, it the, and, or, и т.п.

Элемент открытой очереди содержит URL еще не посещавшегося документа, а также важность этого URL, определяемую согласно стратегии обхода (см. раздел 5).

3 Тематический фильтр

Основная задача сетевого робота OASIS — рекомендовать Коллекции как можно больше страниц, релевантных ее тематике, минимизируя при этом количество рекомендованного мусора. Принятие решения, стоит ли рекомендовать Коллекции найденный документ, основывается на тематическом фильтре.

Тематический фильтр сетевого робота OASIS описывает тематику ассоциированной с ним Коллекции и используется, чтобы грубо вычислять оценки релевантности для полученных документов. Все полученные документы в зависимости от значения оценки релевантности делятся на два класса: документы, которые могут быть рекомендованы Коллекции и "мусор". Точное вычисление релевантности документов выполняется Коллекцией, которая имеет доступ ко всей глобальной статистике системы.

Тематический фильтр состоит из вектора термов и порога рекомендации. Вектор термов содержит термы t и веса термов w^t_{filter}. Порог рекомендации T — это положительное число, используемое для определения того, стоит ли рекомендовать новый найденный документ или нет. "Грубая" оценка релевантности документа вычисляется, используя профайл частот термов для данного документа.

Пусть $f_{t,d}$ — частота использования терма в документе d.

Тогда "грубая" оценка релевантности вычисляется по формуле

$$r(d) = \sum_t f_{t,d} \cdot w^t_{filter} \ . \qquad (1)$$

Документ рекомендуется Коллекции только в том случае, если $r(d) > T$.

Начальный тематический фильтр поставляется самой Коллекцией. Чтобы более точно отвечать требованиям Коллекции, фильтр автоматически улучшается, используя при этом обратную связь (то есть вычисленные Коллекцией точные оценки релевантности для рекомендованных документов). Посредством механизма фильтрования Crawler может уменьшить число документов, заведомо нерелевантных тематике сервера, а следовательно, и улучшить производительность Коллекции.

"Грубые" оценки релевантности, вычисленные фильтром, используются не только для отфильтровывания тематически релевантных документов, но и для управления стратегией сбора Интернет страниц (см. раздел 5). Однако, если клиент робота, Коллекция, предоставляет точные оценки релевантности для рекомендованных документов, то именно они в дальнейшем используются для упорядочивания очереди еще не посещавшихся документов.

4 Автоматическое улучшение тематического фильтра

Используя точные оценки релевантности, которые Коллекция возвращает для рекомендованных документов, Crawler может существенно уточнить свой тематический фильтр, а значит, и улучшить качество рекомендаций.

Алгоритм улучшения тематического фильтра использует начальный фильтр, поставляемый Коллекцией, а также множество полученных от Коллекции точных оценок для рекомендованных документов, чтобы построить новый фильтр, добавляя до 100 новых термов к исходному фильтру.

Любой терм, который встречается в релевантном документе, является кандидатом для добавления в новый фильтр. Все термы из релевантных документов сначала упорядочиваются по частоте использования в множестве релевантных документов. 500 наиболее часто встречающихся термов переупорядочиваются, используя формулу Roccio [1, 2, 6]:

$$\mathrm{Rocchio_t} = \mathrm{w^t_{filter}} + 2\mathrm{w^t_{rel}} - \frac{1}{2}\mathrm{w^t_{non-rel}}$$

где w^t_{filter} — вес терма в начальном фильтре,
w^t_{rel} — вес терма t в множестве релевантных документов,
$w^t_{non-rel}$ — вес терма t в множестве нерелевантных документов (вычисляется аналогично w^t_{rel}).

$$w^t_{rel} = \frac{1}{\mid rel \mid} \sum_{d \in rel} bel_{t,d} \ ,$$

где

$$bel_{t,d} = 0.4 + 0.6 \cdot f_{bel_{t,d}} \cdot idf_t \ ,$$

$f_{bel_{t,d}}$ и idf_t вычисляются по формулам:

$$f_{bel_{t,d}} = \frac{f_{t,d}}{f_{t,d} + 0.5 + 1.5\frac{l_d}{AvgDocLen}} \ ,$$

$$idf_t = \frac{\log\left(\frac{N+0.5}{f_{d,t}}\right)}{\log(N+1)} \ .$$

где $f_{t,d}$ — количество употреблений терма t в документе d, l_d — длина документа d, $AvgDocLen$ — средняя длина документа, $f_{d,t}$ — число документов содержащих терм t.

Так как термы из начального фильтра, которые поставляются Коллекцией, более надежны, чем автоматически сгенерированные, то вес новых термов уменьшается при помощи умножения на некоторый коэффициент ($k < 1$).

Начальный фильтр используется в каждой итерации цикла улучшения тематического фильтра, для того, чтобы какая-нибудь необычная информация в анализируемых документах случайно не сдвинула фильтр в неправильном направлении.

5 Стратегия посещения Интернет страниц

Главная задача стратегии посещения документов — это эффективный выбор порядка обхода, в котором Crawler будет посещать известные ему URL, для того чтобы за минимальное время было обнаружено максимальное число документов релевантных тематике Коллекции.

Если Crawler намеревается посетить все Интернет страницы, а его аппаратные и сетевые ресурсы позволяют ему скачивать и индексировать такое огромное количество документов, то ему не надо заботиться о порядке обхода сети. Ведь, в конце концов, каждый URL будет посещен роботом.

Однако, большинство сетевых роботов не могут посещать все встретившиеся им страницы по нескольким причинам:

- Клиент сетевого робота имеет ограниченные аппаратные ресурсы, в том числе и память, а поскольку объем информации в Интернет огромен (в настоящее время общий размер всех страниц Интернет оценивается в 6 терабайт, и это величина постоянно возрастает), то клиент робота не в состоянии сохранять и индексировать все Интернет страницы.
- Crawler должен поддерживать базу данных уже скаченных документов, периодически посещая их заново, так как Интернет — очень динамичная система, в которой документы удаляются, создаются и модифицируются (по оценкам около 600 гигабайт в Интернет изменяется каждый месяц). Поэтому может оказаться так, что начиная с некоторого момента Crawler будет способен только отслеживать изменения в уже посещенных им документах, и ему не хватит ресурсов на посещение новых страниц.

Поэтому для сетевого робота очень важно посещать наиболее "полезные" документы в первую очередь, чтобы та часть WWW страниц, которую он сможет посетить и поддерживать после, отслеживая изменения, была наиболее важной с точки зрения клиента этого робота.

Качество стратегии посещения Интернет страниц особенно актуально для сетевых роботов, которые собирают определенный тип информации для своих пользователей. К такому типу роботов относится и OASIS Crawler, который обеспечивает своего клиента, тематическую Коллекцию, документами, релевантными ее тематике.

Перед сетевым роботом стоит достаточно сложная задача: Как определить, в каком порядке обходить не посещенные страницы? Ведь единственное, что известно роботу про эти страницы — это URL и содержимое их родительских страниц.

Стратегия обхода Интернет-страниц основывается на следующих предположениях [7, 12]:

- Содержимое страницы имеет тенденцию быть похожим на содержимое родительских страниц.
- Содержимое страницы имеет тенденцию быть похожим на содержимое "страниц-братьев", то есть страниц, имеющих общего с ней предка. Схожесть содержимого "страниц-братьев" обратно пропорциональна расстоянию между ссылками на эти страницы в родительском документе.

Crawler сохраняет все новые ссылки, которые он обнаружил при сканировании полученных документов, в очередь открытых URL, и выбирает из этой очереди следующую ссылку для посещения. Этой ссылкой является URL с самым высоким приоритетом. Соответственно, порядок ссылок в открытой очереди URL определяет порядок обхода Интернет-страниц.

Каждому URL в открытой очереди присваивается значение, основанное на оценке тематической релевантности родительских документов, и на оценке тематической релевантности тех "страниц-братьев", которые уже были скачены.

Важность полученного роботом документа, a, вычисляется по следующей формуле:

$$I(a) = r(a) \ , \tag{2}$$

Оценка тематической релевантности полученного роботом документа, $r(a)$, первоначально равна "грубой" оценке, вычисленной тематическим фильтром; затем, в случае получения уточнения от Коллекции, $r(a)$ полагается равной "точной" оценке релевантности (см. раздел 3).

Вероятная важность документа, который еще не был посещен роботом, вычисляется при помощи следующей формулы:

$$I'(a) = r'(a) + \gamma M(a) \ , \tag{3}$$

где $M(a)$ — это некая функция[3] зависящая от важности посещенных родительских страниц документа a. Коэффициент $0 \leq \gamma < 1$, отражает влияние содержимого родительских страниц на содержимое страниц-потомков. $r'(a)$ — вероятная релевантность тематической Коллекции.

Когда Crawler обнаруживает ссылку на новый документ a, он полагает, что

$$r'(a) = (1 - \gamma)\, M(a) \ , \tag{4}$$

Выбор такого начального значения мотивирован тем, что в этом случае вероятная важность не посещенного документа $I'(a)$ зависит только от важности его посещенных родительских страниц.

$I'(a)$ изменяется либо когда скачивается новая родительская страница (при этом модифицируется слагаемое $M(a)$), либо когда скачивается страница, имеющая общего предка со страницей a (при этом модифицируется слагаемое $r'(a)$).

$r'(a)$ зависит как от значения оценки релевантности посещенной "страницы-брата", так и от расстояния между ссылками на эти две страницы в общем родительском документе.

Каждый раз, после того, как была скачена новая страница a^0, и вычислена оценка ее тематической релевантности $r(a^0)$, значения $r'(a^k)$ перевычисляются для всех страниц a^k, где a^k — k-ый URL, расположенный сверху или снизу относительно ссылки на скаченную страницу a^0.

$$r'(a^k) = r'(a^k) + \beta \left(r'(a^{k-1}) - r'(a^k) \right) \ , \tag{5}$$

Коэффициент $0 \leq \beta \leq 1$ отражает влияние оценки тематической релевантности страницы, на вероятную релевантность ближайших "страниц-братьев".

На рисунке 2 алгоритм описан более детально. Очередь открытых URL (open) содержит все известные роботу не посещенные документы, упорядоченные по $I'(a)$. Очередь закрытых URL (closed) содержит все посещенные ссылки. В очереди модифицируемых URL (modified) содержатся такие ссылки из очереди открытых URL-ей, параметры $M(a)$ или $r'(a)$ которых изменяются на данной итерации алгоритма.

После того, как документ a скачен из WWW (строка 3), и вычислена его тематическая релевантность (строка 4), для каждой из не посещенных страниц-потомков изменяется значение функции важности родительских страниц M (строка 9).

Если при сканировании документа a обнаружились ссылки, которые встретились роботу впервые, то они

[3] В нашей реализации OASIS Crawler в качестве такой функции использовалось среднее арифметическое.

добавляются в очередь открытых URL (строка 13). C_a означает множество страниц-потомков, а P_a означает страницу-предка для a. P_a используется для перевычисления вероятной оценки релевантности "страниц-братьев" (строка 20). Для того чтобы сложность этой части алгоритма оставалась линейной при перевычислении используется только одна родительская страница (строки 12, 17).

При перевычислении вероятной оценки релевантности "страниц-братьев" мы изменяем значения $r'(a^k)$ для всех "страниц-братьев", в том числе и для посещенных роботом (строка 20). Но для посещенных страниц существует точное значение оценки релевантности $r(a^k)$, поэтому, уже на следующем шаге цикла, после перевычисления вероятной оценки релевантности для a^{k+1} страницы, $r'(a^k)$ вновь присваивается точному значению оценки релевантности $r(a^k)$ (строки 24–25).

В следующем цикле (строки 30–32) происходит изменение значений вероятной важности $I'(i)$ для всех элементов очереди модифицируемых URL (то есть, элементов очереди открытых URL, у которых на данной итерации алгоритма изменилось значение $r'(i)$ или $M(i)$).

Очередь открытых URL сортируется, используя новые значения вероятной важности (строка 33).

5.1 Улучшение алгоритма посещения Интернет страниц

В процессе работы OASIS Crawler сталкивается с некоторыми проблемами, которые могут существенно понизить его производительность, если им не уделять должного внимания.

5.1.1 Документы с кадрами

Зачастую WWW страницы практически не содержат текстовой информации, а только описывают расположение и URL содержимого множества кадров (frameset).

Из-за отсутствия содержательной текстовой информации, оценка тематической релевантности такого документа весьма вероятна окажется очень близкой к 0, и поэтому, ожидаемая полезность страниц-потомков (кадров) будет несправедливо занижена.

URL кадров окажутся в хвосте очереди, и скорее всего, никогда не будут скачены, хотя, вполне возможно, что они были тематически релевантными.

Для того чтобы избежать подобной проблемы OASIS Crawler обрабатывает страницы с кадрами специальным образом. Когда робот встречает документ a с кадрами, он откладывает вычисление оценки релевантности $r(a)$, до тех пор, пока не будет вычислены оценки релевантности $r(a^k)$ для всех его кадров a^k.

$$r(a) = \max_{k \in C_a} \{r(a^k)\} \tag{6}$$

Для кадров a^k вероятное значение тематической релевантности $r'(a^k)$ инициализируется, используя не значение важности документа с кадрами $r(a)$ (так как его вычисление отложено (см. формулу 6), а значение важности предка документа a — P_a.

$$r'(a^k) = (1-\gamma)\, M(P_a)\ . \tag{7}$$

```
1: REPEAT
2:   a ⇐ pop(open)
3:   Harvest a
4:   r'(a) ⇐ r(a) ⇐ topic filter score
5:   Add a to closed
6:   I(a) ⇐ r(a)
7:   FOR ALL i ∈ C_a
8:     IF i ∉ closed
9:       Update M(i)
10:      Add i to modified
11:      IF i ∉ open
12:        P_i ⇐ a , r'(i) ⇐ (1 − γ) M(i)
13:        Add i to open
14:      END IF
15:    END IF
16:  END FOR
17:  FOR ALL directions (i.e. up and down) from a in P_a
18:    k ⇐ 1
19:    WHILE a^k ≠ first or last URL on P_a AND a^k ≠ a^0
20:      r'(a^k) ⇐ r'(a^k) + β (r'(a^{k−1}) − r'(a^k))
21:      IF a^k ∈ open
22:        Add a^k to modified
23:      END IF
24:      IF a^{k−1} ∈ closed
25:        r'(a^{k−1}) ⇐ r(a^{k−1})
26:      END IF
27:      k ⇐ k + 1
28:    END WHILE
29:  END FOR
30:  FOR ALL i ∈ modified
31:    I'(i) ⇐ r'(i) + γM(i)
32:  END FOR
33:  Resort open
34:  Clear modified
35: UNTIL open empty
```

Рис. 2: Стратегия посещения Интернет страниц сетевого робота OASIS

5.1.2 Баннеры

В настоящее время бурно развивается использование рекламы в Интернет, и, в частности, применение рекламных баннеров. В большинстве случаев бывает так, что баннер содержит рекламу сайта, тематика которого не имеет ничего общего со страницей, на которой размещен баннер.

С точки зрения Crawler это означает наличие на страницы ссылок которые не соответствуют тематики страницы в целом. Получив новый URL, указывающий на страницу a, но не еще посетив ее, Crawler не может определить степень ее тематической релевантности априори. Робот просто помещает найденную ссылку в очередь открытых URL, причем начальное вероятное значение важности страницы $I'(a)$, а значит и позиция нового URL в очереди зависит только от значения релевантности страницы предка $I'(a) = M(a)$ (см. формулы 3, 4).

С точки зрения робота ссылка баннера ничем не отличается от других ссылок на родительской странице, поэтому робот разместит их в один отрезок очереди открытых URL с одинаковым приоритетом, причем возможно,

что ссылка баннера, которая встретилась на странице первой, будет располагаться в очереди URL раньше всех остальных "ссылок-братьев".

Чтобы решить проблему баннеров, OASIS Crawler использует наблюдение о том, что баннерная реклама обычно встречается в начале или в конце HTML-страницы.

Просканировав скаченную страницу a, робот устанавливает начальные значения $I'(a_k)$ для документов-потомков a_k так, чтобы для страниц, ссылки на которые находятся в начале и в конце документа, то есть возможных баннеров, значение $I'(a_k)$ было меньше.

$$I'(a_k) = M(a) + \epsilon e^{-\frac{([\frac{n}{2}]-k)^2}{n}} \quad (8)$$

где k — порядковый номер ссылки на странице a, n — общее количество ссылок. ϵ — коэффициент, отражающий вероятность того, что ссылки баннера находятся сверху и снизу страницы. Так как такая вероятность не очень велика, то имеет смысл выбирать значение $\epsilon \approx 10^{-6}$, чтобы вклад второго слагаемого в формуле 8 существенно не повлиял на порядок размещения ссылок в очереди, но ссылки из начала и конца страницы, то есть возможные баннеры, имели приоритет немного ниже всех остальных "ссылок-братьев".

6 Экспериментальные результаты

OASIS Crawler в настоящее время находится в стадии разработки, поэтому экспериментальные результаты для оценки производительности всей системы еще не получены. В данной статье представлены результаты экспериментов с тематическим фильтром и стратегией посещения Интернет страниц.

6.1 Тестирование тематического фильтра

Для экспериментов с тематическим фильтром с помощью робота мы построили англоязычные тестовые коллекции, состоящие из HTML страниц, причем каждая страница просматривалась вручную и определялась ее принадлежность той или иной тематике. Было создано десять тематических коллекций, приблизительно по 150 документов в каждой.

Были выбраны следующие тематики коллекций: Оценка производительности, Карточные игры, Автомобили, Мониторы, Музеи, Поиск Информации, Языки программирования, Исследовательские группы, Путешествия, Unix-Linux.

6.1.1 Автоматическое построение тематического фильтра для коллекции

Для того чтобы построить тематические фильтры, из каждой коллекции было случайным образом выбрано подмножество из 50 документов. Эти подмножества, C^i, использовались, чтобы сгенерировать тематические фильтры для каждой коллекции.

Для каждого документа d вычислялась частота использования терма t в данном документе $f_{t,d}$.

Для всех термов t из коллекции C^i, вычислялась средняя частота использования

$$f_{t,C^i} = \frac{\sum\limits_{d \in C^i} f_{t,d}}{N^i} \quad (9)$$

где N^i — количество документов в C^i. Все термы, средняя частота использования которых в C^i превышала среднюю частоту в C^0 (C^0 — объединение всех C^i), включались в тематический фильтр для i-ой коллекции с весом $w^{t,C^i} = f^{t,C^i} - f^{t,C^0}$

6.1.2 Измерение производительности тематического фильтра

Для оценки производительности поисковых систем обычно используется понятия точность и полнота. Можно использовать эти же самые критерии и для оценки производительности фильтра.

Точность — это параметр, показывающий, какова доля релевантных документов в общем числе рекомендованных (т.е. прошедших через фильтр). Например, если среди 100 документов, рекомендованных фильтром, 40 релевантных, то точность системы фильтрования 40%.

Полнота — это параметр, показывающий, какова доля рекомендованных (т.е. прошедших через фильтр) в общем количестве релевантных документов. Например, если среди 100 документов, относящихся к тематике (то есть релевантных), фильтр рекомендовал 20 документов, то полнота системы фильтрования 20%.

Очевидно, что хорошая система фильтрования должна иметь как можно большую полноту и точность (в идеале 100%). То есть рекомендовать все предоставленные ей релевантные документы и не рекомендовать ни одного документа, не относящегося к тематике.

Конечно, на практике стопроцентное качество работы фильтра невозможно. В частности, это можно объяснить тем, что понятия точности и полноты в некотором смысле противопоставляются друг другу. Соотношение "точность-полнота"аналогично соотношению "время-память"в программировании. То есть при попытке улучшить один из параметров системы с фиксированной производительностью автоматически происходит ухудшение второго параметра.

Мы ввели еще один параметр оценки производительности, процент мусора (ПМ), показывающий, какова доля нерелевантных документов в общем числе рекомендованных. В нашем случае более интересно вычислять производительность тематического фильтра с помощью параметров полнота и процент мусора, так как основная задача фильтра — максимизировать количество релевантных страниц, рекомендуемых Коллекции, минимизируя при этом процент рекомендованного мусора.

В таблице 6.1 представлены результаты экспериментов с тематическим фильтром. Порог рекомендации, T^i, был выбран таким образом, чтобы определенная часть документов, R^i, в коллекции C^i, рекомендовалась фильтром коллекции. Результаты, представленные в таблице, соответствуют порогам рекомендации $R^i = 0.90$ и $R^i = 0.98$.

6.2 Эксперименты со стратегией обхода

Основной целью этой группы экспериментов является практическая оценка полезности применения тематически-ориентированной стратегии обхода.

6.2.1 Измерение производительности стратегии посещения документов

Цель этого эксперимента — выяснить насколько хорошо предложенная стратегия справляется с предсказани-

Тематика фильтра	$R_i = 0.90$				$R_i = 0.98$			
	$T_i \cdot 10^6$	Точность	Полнота	ПМ	$T_i \cdot 10^6$	Точность	Полнота	ПМ
Оценка производительности	214	0.62	0.77	0.05	158	0.51	0.86	0.09
Карточные игры	564	0.33	0.92	0.18	388	0.31	0.97	0.22
Автомобили	522	0.78	0.87	0.03	423	0.72	0.93	0.04
Мониторы	872	0.89	0.79	0.01	473	0.80	0.94	0.03
Музеи	250	0.93	0.75	0.01	107	0.69	0.93	0.05
Поиск информации	178	0.62	0.78	0.05	99	0.32	0.96	0.21
Языки программирования	153	0.42	0.88	0.14	129	0.34	0.93	0.20
Исследовательские группы	446	0.79	0.78	0.03	359	0.71	0.89	0.04
Путешествия	199	0.57	0.87	0.10	42	0.17	0.98	0.70
Unix–Linux	359	0.78	0.79	0.03	229	0.44	0.94	0.14

Table 1: Оценка производительности тематического фильтра

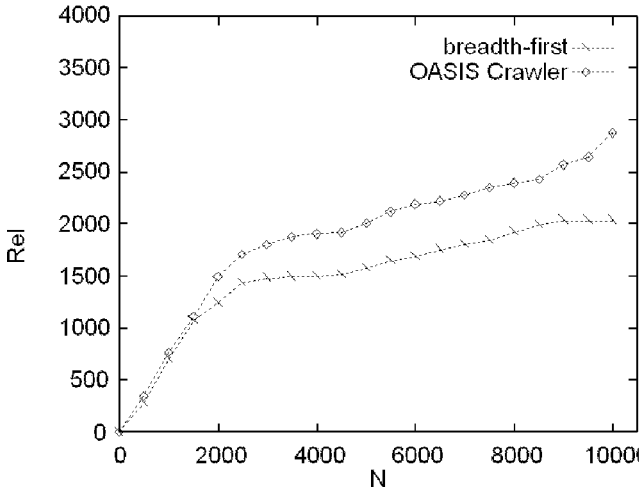

Рис. 3: Сравнение производительности сетевого робота OASIS и breadth-first робота (порог тематического фильтра T=0.00025)

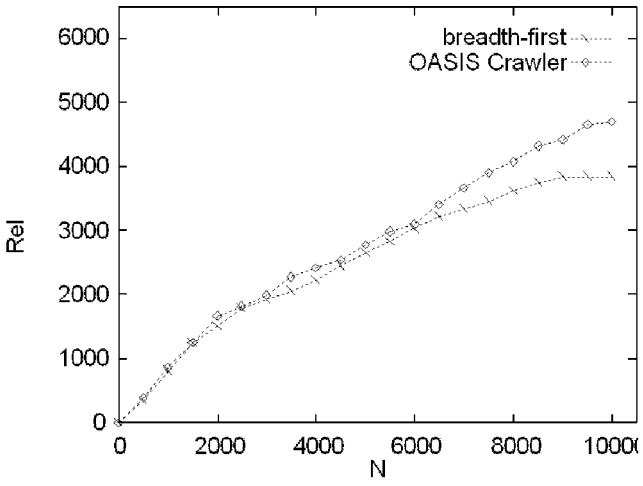

Рис. 4: Сравнение производительности сетевого робота OASIS и breadth-first робота (порог тематического фильтра T=0.000107)

ем тематической релевантности еще не посещенных страниц и, как следствие, насколько эффективен выбираемый способ обхода.

В качестве стартовых URL для проведения этого эксперимента были выбраны 15 ссылок из каталога "Наука" с *Yahoo!*.

Производительность стратегии посещения документов измерялась при помощи формулы:

$$P = \sum_{i=0}^{N} \frac{N-i}{N} r^i \ , \ N > 0 \ , \qquad (10)$$

где r^i — оценка релевантности тематическим фильтром, N — общее число страниц.

Основная задача стратегии посещения документов — отсортировать очередь URL таким образом, чтобы максимизировать P.

Производительность "жадного" алгоритма, который сортирует очередь, априори зная оценку релевантности страницы, на которую ссылается URL, была выбрана как верхняя граница производительности ($P = 15.20$). Производительность алгоритма, который случайным образом выбирает URL из очереди, была использована как нижняя граница ($P = 11.01$).

При наилучшем выборе коэффициентов производительность для предложенного нами алгоритма ($P = 13.98$). Если использовать "случайный" алгоритм в качестве нижней границы, производительность предложенного алгоритма составила 71 процент от производительности "жадного" алгоритма.

6.2.2 Сравнительный анализ

Целью этого эксперимента являлся сравнительный анализ производительности стратегии посещения документов OASIS робота и робота, использующего стратегию "обхода в ширину" (breadth-first).

Стратегии "обхода в ширину" (breadth-first) и "обхода в глубину" (depth-first) являются наиболее часто используемыми стратегиями среди традиционных сетевых

роботов.

Порядок, в котором робот добавляет в список URL, которые были обнаружены, но еще посещались роботом, определяет стратегию посещения документов. Если ссылки добавляются и извлекаются для скачивания из одного и того же конца списка, то робот реализует стратегию "обход в глубину" (depth-first). Если же ссылки добавляются и извлекаются из разных концов списка, то робот реализует стратегию "обхода в ширину" (breadth-first).

Целью этого эксперимента было сравнение производительности тематико-ориентированной стратегии обхода OASIS Crawler и стратегии обхода, не учитывающей тематику обнаруженных документов.

На базе популярной свободно распространяемой программы `wget` были разработаны прототипы простых роботов реализующих стратегии "обхода в ширину" и "обхода в глубину". Поскольку в ходе наших экспериментов стратегия "обход в ширину" всегда показывала лучшие результаты, то далее мы рассматриваем только ее.

OASIS Crawler с тематическим фильтром "Музеи" и робот, использующий стратегию "обхода в ширину", начав с одного и того же URL (релевантного тематике "Музеи"), посетили по 10000 HTML документов.

После этого мы выяснили какое число из посещенных документов удовлетворяет заданному тематическому фильтру "Музеи". Это число, конечно, не является точной оценкой количества обнаруженных релевантных документов, поскольку удовлетворяющие фильтру документы только потенциально релевантны соответствующей тематике. Однако подобная информация позволяет сравнить общее поведение рассматриваемых стратегий.

Этот эксперимент был проведен при двух значениях порога тематического фильтра — $T^1 = 0.00025$ и $T^2 = 0.000107$, соответствующих показателям полноты $R^1 = 0.90$ и $R^2 = 0.98$ при экспериментах с тематическим фильтром "Музеи" (см. раздел 6.1.2).

Полученные результаты проиллюстрированы на рисунках 3 и 4, соответственно. Преимущество стратегии OASIS Crawler довольно очевидно в обоих случаях.

7 Заключение

В работе рассматривается задача создания интеллектуального сетевого робота для сбора тематических коллекций.

Описана базовая архитектура системы, структура тематического фильтра и методы оценки тематической релевантности документа. Использование дополнительной информации от клиента робота во время работы для уточнения тематического фильтра позволяет улучшать качество оценок в процессе работы. Описываемая стратегия обхода сети учитывает тематические оценки уже посещенных документов, что позволяет посетить тематически релевантные документы в первую очередь.

Предварительные результаты экспериментов показывают преимущество тематически-ориентированной стратегии обхода над другими стратегиями для сбора тематических коллекций. Все это подтверждает перспективность предлагаемого подхода.

Библиография

[1] I.J. Aalbersberg. Incremental relevance feedback. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 11–22, 1992.

[2] J. Allan. Incremental relevance feedback. In *Proceedings of the 19th International Conference on Research and Development in Information Retrieval (SIGIR '96)*, pages 298–306, April 1996.

[3] A. Ardö and S. Lundberg. A regional distributed WWW search and indexing service - the DESIRE way. *Computer Networks and ISDN Systems*, 30(1-7):173–183, 1998.

[4] Mikhail Bessonov, Udo Heuser, Igor Nekrestyanov, and Ahmed Patel. Open architecture for distributed search systems. In *Proc. of the Sixth International Conference on Intelligence in Services and Networks*, April 1999.

[5] C. M. Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz. The harvest information discovery and access system. *Computer Networks and ISDN Systems*, 28(1-2):119–125, December 1995.

[6] J. Callan. Document filtering with inference networks. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 262–269, 1996.

[7] S. Chakarabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Ginson, and J. Klienberg. Automatic resource compilation by analyzing hyperlink structure and associated text. In *Proceedings of the Seventh International World Wide Web Conference*, April 1998.

[8] F. Cheong. *Internet Agents: Spiders, Wanders, Brokers, and Bots.* New Riders, Indianapolis, 1996.

[9] Junghoo Cho, Hector Garcia-Molina, and Lawrence Page. Efficient crawling through URL ordering. In *Proceedings of the Seventh International World Wide Web Conference*, April 1998.

[10] T. Koch, A. Ardö, A. Bremmer, and S. Lundberg. The building and maintenance of robot based internet search services: A review of current indexing and data collection methods. DESIRE State-of-the-Art Report D3.11, Lund University Library, Sweden, 1996.

[11] M. Koster. Robots in the Web: threat or treat? *ConneXions*, 9(4), 1995.

[12] R.R. Larson. Bibliometrics of the world wide web: An exploratory analysis of the intellectual structure of cyberspace. In *Proceedings of the 1996 ASIS Meeting*, April 1996.

[13] Steve Lawrence and C. Lee Giles. Searching the World Wide Web. *Science*, 280(5360):98–100, 1998.

[14] Lawrence Page, Sergey Brin, and Rajeev Motwani. The pagerank citation ranking: Bringing order to the web. In *Proc. of the Seventh International World Wide Web Conference*, February 1998.

[15] Brian Pincerton. Finding what people want: Expirences with the webcrawler. In *Proc. of the second International World-Wide Web Conference*, 1994.