ПОИСКОВЫЕ СИСТЕМЫ: КОМПОНЕНТЫ, ЛОГИКА И МЕТОДЫ РАНЖИРОВАНИЯ

А.В. Кириллов,
аспирант кафедры инноваций и бизнеса в сфере информационных технологий факультета бизнес-информатики Государственного университета-Высшей школы экономики, инженер-программист ЗАО Фирма «Клуб 400».
Адрес: г. Москва, Варшавское шоссе, д. 39,
e-mail: antonv.kirillov@gmail.com.

АннотацияВ статье рассматриваются принципы построения поисковой системы и ее компонентов, а также методы ранжирования результатов, в частности, основанный на рейтинге цитирования алгоритм PageRank. Детально анализируется проблема выборки документов, релевантных поисковому запросу, в гетерогенной среде, такой как World Wide Web. Продемонстрирована необходимость расширения классических методов поиска информации при помощи методов ранжирования, учитывающих рейтинг цитирования.

Ключевые слова: поисковые машины, рейтинг цитирования, гетерогенное окружение.

1. Введение

В настоящее время все большее количество знаний, накопленных человечеством, хранится в компьютеризированных репозиториях, таких как Всемирная Сеть (World Wide Web). Данная ситуация влечет за собой проблему поиска определенной информации в этих часто неструктурированных репозиториях, которая решается при помощи поисковых систем. Концепция поисковой системы достаточно проста: пользователь вводит поисковый запрос, состоящий из нескольких ключевых слов, относящихся к целевым документам, которые должны быть извлечены из репозитория. Результатом работы поисковой системы является упорядоченный набор документов, которые считаются релевантными данному запросу.

Фундаментальной проблемой при разработке поисковых систем является определение релевантности – ситуации, когда документ соответствует запросу. Архитектура поисковой системы основывается на идеях из области поиска информации, предоставляющей методы определения релевантности документа по его фактическому содержимому. Тем не менее, в гетерогенной среде, такой как WWW, эти методы сами по себе оказываются неэффективными. Кроме того, что гораздо хуже, эти методы не защищены от мошенничества со стороны людей, пытающихся извлечь из поисковых систем коммерческую выгоду.

Именно поэтому современные поисковые системы рассматривают весовые факторы, которые не имеют прямой зависимости от содержимого документа. Общий метод, пригодный в информаци-

онных репозиториях, использующих гиперссылки, таких как WWW, состоит в анализе структуры ссылок, образованной между документами. Наиболее успешным алгоритмом из этой категории является алгоритм PageRank, предложенный основателями поисковой системы Google.

2. Поисковые системы

В повседневной речи под термином «поисковая система» понимается программное обеспечение, состоящее из базы данных документов, снабженной пользовательским интерфейсом, позволяющим пользователю получить упорядоченное подмножество этих документов как ответ на его поисковый запрос. Основная задача поисковой системы заключается в выборе наилучшего возможного подмножества в ответ на конкретный запрос, т.е. множества документов, которые наиболее соответствуют тому, что ищет пользователь (обычно в порядке убывания релевантности).

Самыми распространенными примерами поисковых систем, используемых повсюду, являются поисковые системы для Веба (такие как Google и Yahoo, например), которые применяются для обнаружения текстовой информации (например, документы в формате HTML и PDF), хранящейся на веб-серверах, расположенных по всему миру. Схожие технологии используются и при поиске информации в корпоративных внутренних сетях.

3 Формальные компоненты поисковой системы

Большинство поисковых систем состоит из двух основных, независимых компонентов, которыми являются компонент индексирования и компонент поиска. Пользователю доступен только поисковый компонент. Компонент индексирования используется для создания внутреннего эффективного представления данных, в которых будет производиться поиск необходимой информации, а поисковый компонент отвечает за получение результатов из внутренней базы данных в ответ на поисковый запрос пользователя.

Формально компонент индексации может быть представлен функцией $I : U \to R$. Множество U называется универсумом и содержит данные, среди которых будет вестись поиск. Для поисковой системы Интернета – это страницы, которые мы загружаем из сети, для графической поисковой системы им будет являться набор изображений, а для академической поисковой системы универсум будет представлен, например, собранием работ, статей и книг. Множество R, являющееся внутренним представлением универсума U, называется репозиторием. Репозиторий имеет вид $R = \{\sigma_d | 1 \le d \le n\}$, где каждое σ_d является документом, а d — соответствующим уникальным идентификатором этого документа, называющимся DOCID. Когда речь идет о документе d, используется преобразование $d \mapsto \sigma_d$. Каждое представление σ_d зависит, в первую очередь, от поисковой системы.

Следует отметить, что функция I обычно применяется к подмножеству U' множества U, и поэтому поиск происходит только в части всего репозитория $R' = I(U')$. Объясняется это тем, что множество U слишком велико, чтобы быть проанализировано полностью (см. ниже).

Проиллюстрируем концепцию индексирования на примере поисковой системы для Веба. Местонахождение веб-страниц обычно определяется по Unified Resource Locator или URL (например, http://www.hse.ru). При индексировании сети система имеет дело с набором URL различных документов (которые в Вебе называются страницами) и последовательно присваивает им идентификаторы (DOCID). Затем данные страницы выгружаются из Веба и создается репозиторий, т.е. хранилище внутренних представлений каждой из страниц. Количество выгружаемых страниц обычно очень велико (в современных поисковых системах это порядка 1010 документов), но, тем не менее, оно значительно меньше реального числа страниц в U, т.е. количества страниц, находящихся в Интернете. Таким образом, основной задачей построения поисковых систем для Веба является определение адекватного подмножества U' множества U.

Рассмотрим компонент поиска, который обращается к документам, расположенным в репозитирии для того, чтобы осуществить выборку, соответствующую поисковому запросу. Формально, поисковый компонент может быть представлен как программа, реализующая преобразование $S : \omega \mapsto \tau$, где ω – поисковый запрос, т.е. конечная строка, введенная пользователем (принадлежащая используемому алфавиту). Поисковый запрос ω принято считать состоящим из термов, являющихся атомарными словами, поиск которых ведется, и операторов, которые описывают, как интерпретировать термы. Например, в поисковом запросе «цепи Маркова», запрос состоит из термов $\omega_1 = Маркова$ и $\omega_2 = цепи$. Оператором в данном случае будет являться «логическое И», что описывает ситуацию, когда нам необходимы документы, содержащие оба этих терма.

В преобразовании $S:\omega\mapsto\tau$ τ — это результат, являющийся упорядоченным набором (или же вектором) отдельных документов: $\tau=(\sigma_{\tau_1},\sigma_{\tau_2},...,\sigma_{\tau_r})\sim(\tau_1,\tau_2,...,\tau_r)$, где используется свойство изоморфности документов, такое, что $\tau_i:s$ фактически является идентификатором документа(DOCID). Количество возвращаемых документов, $r=|\tau|$, называется *эффективностью поиска* для данного поискового запроса. Очевидно, что $0\le r\le n$.

Результат τ – это информация, представляемая пользователю. Элементы τ – это все документы, которые поисковая система сочла достаточно подходящими для включения в результирующий набор. Более того, элементы в результирующем множестве расположены в таком порядке, что τ_i считается более значимым для пользователя, чем τ_{i+1}. При обычном веб-поиске 10 документов, представленных на первой странице результатов, будут соответствовать $\tau_1-\tau_0$. *Точность* определяется долей возвращенных документов, которые фактически релевантны, т.е.

$$Точность=\frac{|\{Релевантные\_документы\}\cap\tau|}{|\tau|}$$

Здесь понятие *релевантности* является абсолютно произвольным и полностью зависит от поисковой системы (или, возможно, от ее пользователей).

Рассмотрим проблему получения τ на основании поискового запроса ω и репозитория R. Поисковая система обычно осуществляет выборку τ в два этапа:

1. Выбор множества претендентов $\tilde{\tau}\subset R$, такого, что все элементы $\tilde{\tau}$ в той или иной степени релевантны поисковому запросу. Определение релевантности на данном этапе очень приближенное. Например, может быть использован *логический метод*, рассматриваемый далее.

2. Для каждого $\tau_i\subset\tilde{\tau}$ определяется его *релевантность* $\mathrm{Rel}(\tau_i)$, а затем $\tilde{\tau}$ сортируется в порядке уменьшения релевантности. В процессе сортировки некоторые элементы, имеющие релевантность ниже порогового значения, могут быть исключены из выборки. Результирующей выборкой будет являться τ.

4. Логический метод определения множества претендентов

Рассмотрим процесс определения множества претендентов $\tilde{\tau}$, который обычно происходит с использованием *логического метода*. Основная идея данного метода заключается в том, что результирующее множество поискового запроса (такого как, например, «цепи Маркова») должно содержать только страницы, относящиеся ко всем уникальным термам запроса (в данном случае ими будут являться «Маркова» и «цепи»). Затем ответ на поисковый запрос может быть дан после просмотра всех документов, содержащих термы «Маркова» и «Цепи», используя документы, содержащие пересечение данных термов как результирующее множество претендентов.

Так происходит по той причине, что основной задачей компонента индексации является построение *инвертированного индекса*, являющегося структурой данных, в которой термам в соответствие ставятся документы (или же DOCID), содержащие данные слова (как расширение в, например, поисковой системе изображений, терм «лицо» может быть привязан ко всем документам, которые классифицируются как содержащие лица).

Таблица 1.

Пример инвертированного индекса.

Терм	DOCID документов, содержащих данный терм
Маркова	35, 678, 432,1839, 6456, …
цепи	7834, 889, 8912, 325, 91, …
.	.
.	.
.	.

В *таблице 1* представлен пример инвертированного индекса. Инвертированный индекс является одной из важных частей вышеупомянутого внутреннего представления документов. Запрос, таким образом, подвергается декомпозиции в древовидную структуру с термами (т.е. атомарными словами или фразами) в качестве листьев и логическими операторами в качестве узлов.

Наиболее используемыми логическими операторами являются AND, OR и NOT, равнозначные операциям $\cap$ (пересечения), $\cup$ (объединения) и $\complement$ (дополнения) между множествами DOCID соответственно. В дальнейшем эти символы будут использованы для того, чтобы различать операции над множествами и логические операции. AND обычно подразумевает отсутствие оператора между двумя термами. Несколько примеров логических представлений поисковых запросов представлены в *таблице 2.*

Логическое сравнение является простым путем получения множества $\tilde{\tau}$ потенциально релевантных документов, но, конечно, не представляет их в порядке соответствия запросу. Поэтому необходимо использовать разные методы при сортировке $\tilde{\tau}$ и при получении τ.

Таблица 2.

Примеры логических интерпретаций поисковых запросов.

Запрос	Логическая интерпретация
Маркова Цепи	{Маркова} $\bigcap$ {Цепи}
Маркова –Цепи	{Маркова} $\bigcap \complement$ {Цепи}
Маркова (Цепь OR Процесс)	{Маркова} $\bigcap$ ({Цепь} $\bigcup$ {Процесс})

5. Проблема ранжирования: переход от $\tilde{\tau}$ к τ

В дальнейшем, после определения $\tilde{\tau}$, происходит поиск зависимой от поискового запроса функции *ранжирования* или *релевантности* $\mathrm{Re}l_{\omega}:\tilde{\tau}\to[0,\infty)$ такой, что $\mathrm{Re}l_{\omega}(\tau_i)>\mathrm{Re}l_{\omega}(\tau_j)$, если элемент τ_i считается более *релевантным* запросу, чем τ_j, и, таким образом, должен находиться в τ до него. Другими словами, результирующее множество τ должно быть отсортировано по убыванию значения $\mathrm{Re}l_{\omega}$. Функции класса $\mathrm{Re}l_{\omega}$ являются строго охраняемым секретным компонентом любой поисковой системы и определяют схему ранжирования, т.е. те характеристики документа, которые были определены как значимые при формировании результатов для определенного поискового запроса.

Логический метод в той или иной степени является применимым к любому набору данных, однако проблема ранжирования в высшей степени зависит от окружения U, из которого данные были извлечены. Например, поисковые системы для веба постоянно сталкиваются с проблемой спама: веб-страницы, которые пытаются «перехитрить» поисковые системы, предоставляя необычайно высокое значение $\mathrm{Re}l_{\omega}$ для конкурентоспособного $\omega:s$, тем самым, рассчитывая на увеличение количества появлений страницы в результатах поиска. Данная проблема приводит к тому, что функция $\mathrm{Re}l_{\omega}$ должна определяться как можно тщательнее и скептически. Тем временем также и не стоит отсеивать «честные» документы. Это приводит к тому, что решение проблемы ранжирования результатов в неконтролируемой среде становится очень востребованным и перспективным. Обычно спам не является проблемой в более контролируемых средах, таких как поисковые системы для академических работ или внутренних сетей.

Обратим внимание на особенности ранжирования в неконтролируемых средах, таких как Веб. Здесь функция ранжирования принимает в расчет как *внешние* факторы (*on-page factors)*: информационное содержимое и его размещение на странице, так и *внутренние* факторы (*inter-page factors*): обычно, информация о том, как страницы соотносятся с другими посредством гиперссылок и т.п. Основное внимание следует уделить внутреннему фактору гиперссылок между страницами: предварительно проведем небольшой обзор процесса ранжирования в целом. Мотивацией к изучению внутренних факторов является то, что все внешние факторы находятся под полным контролем автора страницы. Изучение различных отношений внутри документа с гораздо большим числом страниц позволяет более эластично оценить качество исследуемой страницы.

В общем случае, функция ранжирования поисковой системы для Веба выбирается следующим образом:

$$\mathrm{Re}l_{\omega}(\tau)=\mathrm{P}(\tau,\omega)q(\tau) \qquad (1)$$

где $\mathrm{P}(\tau,\omega)$ является показателем документа τ для запроса ω по внешним факторам, т.е. насколько релевантна информация, расположенная на странице τ, по отношению к запросу ω, а $q(\tau)$ является *качественной* функцией от τ, которая рассчитывается на основании факторов, не представленных непосредственно на самой странице. Качественная функция q может включать в себя такие вещи, как внутренние факторы страницы и ручное вмешательство (т.е. страница была специально изменена для поднятия рейтинга и позиции в результатах поиска). Следует отметить, что q не является функцией, зависящей от запроса, а скорее присваивает обобщенный весовой коэффициент каждой странице независимо от запроса. Функция q принимает значения в пределах [0;1], таким образом, умножение на q используется для дампинга рангов документов (т.е. набранных ими «очков» по внешним факторам).

Рассмотрим далее три возможных метода определения $\mathrm{P}(\tau,\omega)$.

5.1. Логический метод ранжирования

Представим простейшую поисковую систему, принимающую значение $\mathrm{P}(\tau,\omega)=1$, и в результате

имеющую $\mathrm{Re}\, l_{\omega}(\tau) = q(\tau)$. Результирующее множество τ будет состоять исключительно из множества претендентов $\tilde{\tau}$, отсортированного по убыванию значения q. Так функционирует чисто логическая поисковая система: все страницы, имеющие *любое* отношение к термам, которые ищет пользователь, одинаково релевантны поисковому запросу.

5.2. Ранжирование на основе вектора документа

Подход к ранжированию с использованием *вектора документа* является достаточно популярной технологией.

Первым предположением в данной модели является то, что документ τ должен иметь высокий рейтинг по терму ω_i, если данный терм часто встречается на этой странице. Предположим, что запрос ω состоит из L термов: $\omega_1,...,\omega_L$. Зададим частоту термов, $TF_{\omega_i}(\tau)$, как отношение количества появлений терма ω_i в документе к размеру (S_τ) документа в некоторых удобных единицах измерения (например, количество слов или байтов).

Далее, предположим, что некоторые термы более значимы при поиске, чем другие. Стандартный метод определения значимости термов заключается в нахождении инверсивной частоты документа. Предположим, что R_{ω_i} является подмножеством репозитория, состоящим из документов, содержащих терм ω_i. Вероятность p того, что документ, выбранный случайно, будет содержать терм ω_i, такова:

$$p = \frac{\left|R_{\omega_i}\right|}{\left|R\right|}.$$

В Теории информации Шэннона [1] это соответствует собственной информации (self-information) $\log_2(\frac{1}{p})$.

На основании этого определяется *инверсивная частота документа*

$$IDF(\omega_i) = \log(\frac{\left|R\right|}{\left|R_{\omega_i}\right|}),$$

т.е. логарифм отношения общего числа документов в репозитории к количеству документов, содержащих терм ω_i (обычно, принято использовать логарифм по основанию 10). Инверсивная частота документа представляет собой оценку количества информации, свойственной терму. Если терм часто встречается в документах, находящихся в репозитории, то вероятность того, что он весьма общий, высока, и поиск определенного ресурса при помощи поисковой системы не даст значительных результатов, поэтому ему присваивается низкое значение *IDF*. В *таблице 3* представлены примеры вычисленных значений *IDF* для некоторых термов (в примере используются словосочетания), относящихся к хорошо известной теории множеств, но с возрастающей степенью обобщения и, поэтому, с убывающим количеством содержащейся в документах полезной информации.

TF и *IDF* будут использоваться для определения оценки документа. Для каждого $\tau \in \tilde{\tau}$ определим *вектор документа* δ_τ, состоящий из L элементов (по одному для каждого терма), такой, что выполняется соотношение

$$\left[\delta_\tau\right]_i = IDF(\omega_i) \cdot TF_{\omega_i}(\tau).$$

Элементы вектора документа, таким образом, являются относительной единицей измерения отношения частоты вхождений терма в документ к частоте появления терма в репозитории в целом и, по существу, данные элементы принимают во внимание как значимость терма в документе, так и его предполагаемую информационную значимость.

Таблица 3.

***IDF*, вычисленные поисковой системой Yahoo, при $\left|R\right|$ приблизительно равном $2 \cdot 10^9$**

Терм	Количество вхождений	IDF
теорема Перрона - Фробениуса	8270	6.38
цепь Маркова	1050000	4.28
теория вероятностей	10900000	3.26
математика	92900000	2.33
наука	816000000	1.39

Можно рассматривать *поисковый запрос* как документ сортов, в котором каждый из термов запроса встречается только один раз. Пусть υ – это L – вектор для каждого $\upsilon_i = S_\upsilon^{-1} IDF(\omega_i)$ (где S_υ – это размер запроса, представленный в тех же единицах измерения, что и размер документа, о котором говорилось ранее). Помимо этого, можно рассматривать это как вектор документа для запроса. Так как неизвестно, каким образом пользователь задает приоритеты термам в его запросе, то весовые коэффициенты термам будут присвоены в соответствии с их *IDF*.

Определим зависящую от запроса часть функции отношения, т.е. $\mathrm{P}(\tau, \omega)$ в (1), чтобы установить соответствие между δ_τ и υ, а определением

соответствия, в данном случае, будет являться угол между векторами в L -пространстве:

$$\mathrm{P}(\tau,\omega)=\cos(\angle(\delta_\tau,\upsilon)\ =\frac{\delta_\tau\cdot\upsilon}{\|\delta_\tau\|^2\cdot\|\upsilon\|^2} \qquad (2)$$

Пример 1. Продемонстрируем векторную модель на практике, рассмотрев процесс поиска для запроса «связный граф».

В хорошо известной поисковой системе для Веба можно обнаружить примерно $20\cdot10^9$ документов, в которых терм «связный» встречается в $7\cdot10^9$ документах, а терм «граф» в $150\cdot10^6$ документах. Таким образом, значения *IDF* будут следующими: *IDF*(связный) = 0.46 и *IDF*(граф) = 2.1. Используя количество слов как единицу измерения, получаем размер запроса, равный 2, и вектор запроса

$$\upsilon=\frac{1}{2}(1\cdot IDF(связный),1\cdot IDF(граф))=(0.23,1.05)$$

Необходимо сравнить по релевантности два документа A и B, которые, предположим, одного размера. Следовательно, мы можем использовать количество вхождений терма в документ в качестве его *TF*-значения. В *таблице 4* приведены количество вхождений термов запроса в документы, а также соответствующие векторы документов.

Таблица 4.

Количество вхождений термов запроса в документы и соответствующие векторы документов.

	Документ A	**Документ B**
Количество вхождений терма «связный»	10	2
Количество вхождений терма «граф»	4	3
Вектор документа	$\delta_A=(4.6,8.4)$ $\|\delta_A\|=9.6$	$\delta_B=(0.92,6.3)$ $\|\delta_B\|=6.4$
Оценка (по (2))	0.96	0.9975

Простой подсчет количества вхождений термов дает преимущество документу A, однако видно, что B фактически лучше удовлетворяет запросу. Это интуитивно доказуемо. Например, большее число вхождений терма «связный» в документе A показывает, что этот документ посвящен связям в несколько другом контексте, чем «связные графы», и поэтому слабо удовлетворяет запросу.

5.3. Реалистичные модели ранжирования

Большинство поисковых систем в действительности используют улучшенную модель вектора документа. Тем не менее, существует множество противников данной модели, поскольку самый существенный ее недостаток в том, что подход, использующий подсчет частоты вхождений термов, может дать ошибочные результаты, т.к. количество слов на странице подсчитывается «вслепую». Поэтому вносятся корректирующие коэффициенты, основанные на таких факторах, как расположение термов относительно друг друга, статистические измерения корреляции между термами и аспектами форматирования страницы (такими как шрифт и размер шрифта, которым представлены термы).

Одним из популярных методов ранжирования является OKAPI BM25 [2], где рейтинг документа вычисляется на основе формулы:

$$P(\tau,\omega)=\sum_{\omega_i\in\omega}IDF(\omega_i)\frac{(k+1)TF_{\omega_i}(\tau)}{TF_{\omega_i}(\tau)+k(1-b+b\frac{d_\tau}{\overline{d}})},$$

где, обычно, k=1.2, b=0.75, d_τ – длина документа τ (т.е. количество слов в документе) и $\overline{d}$ – средняя длина всех документов. Данная функция пытается нормализовать рейтинги документов, исходя из их длины: большой документ может содержать гораздо больше повторений отдельных термов, чем маленький, и, тем не менее, быть менее релевантным запросу.

Следующим улучшенным вариантом является функция OKAPI BM25F, в которой ранжирующая функция разбивается на части относительно *полей* документа, таких как заголовки, ссылки, основной текст и т.д.

Однако у всех представленных методов существует еще одна проблема. Пользователь, совершающий поиск, ожидает найти авторитетную информацию в результатах поиска раньше, чем все остальное. Но в большинстве случаев стандартные слова в поисковом запросе не выделены особым образом на анализируемых при поиске страницах. Например, не все производители автомобилей используют слово «машина» на своих веб-сайтах! Более того, неавторитетные или даже вредоносные ресурсы могут легко обогнать авторитетные по рейтингу, просто используя термы по нескольку раз на своих страницах: к примеру, при поиске «Volvo», главная страница компании Volvo будет находится гораздо ниже в рейтинге, чем локальные дистрибьюторы автомобилей, использующие сло-

во «Volvo» десятки раз у себя на страницах. Это не тот эффект, который необходим. Необходимо решение, которое позволило бы оценивать *качество* страницы независимо от запроса.

Эта проблема, а также проблема спама, являются основными причинами введения в функцию ранжирования **(1)** коэффициента $q(\tau)$.

6. Оценка качества документа на основе цитирования: алгоритм PageRank

Рассмотрим один из наиболее популярных и широко используемых методов оценки качества документов, основанный на ссылках между документами, так называемый метод рейтинга цитируемости. Примерами цитат могут служить список цитированной литературы в научных работах или гиперссылки между веб-страницами. Идея рейтинга цитируемости заключается в определении качественной оценки документа на основании количества и качества ссылающихся на него документов.

Абстрагируясь от того, что цитата представляет собой только ссылку с одной страницы на другую без каких-либо специфических атрибутов (т.е. не учитывается размещение ссылки в документе, ее формат и т.д.), можно представить ссылочную структуру в виде графа. Предположим, репозиторий состоит из n документов, имеющих уникальные идентификаторы DOCID, последовательно присвоенные документам и находящиеся в интервале $V = [1, n]$.

Определение 1. *Цитатой*, или же другими словами, *ссылкой* называется упорядоченная пара документов $(i, j) \in V^2$. Ссылками называются *исходящая связь* документа i и *входящая связь* документа j.

Сформировав из всех ссылок между документами из V множество E, становится ясно, что $G = (V, E)$ является ориентированным графом с вершинами, являющимися ссылками. Назовем данный граф *графом ссылок*.

Определение 2. Пусть $G = (V, E)$ где V — конечное множество вершин графа, $E \subset V * V$, и $i \in V$. Тогда множество входящих связей будет обозначаться как $I(i)$, а множество исходящих связей как $O(i)$, т.е.

$$I(i) = \{e \in E | e = (j, i),\ j \in V\},$$

$$O(i) = \{e \in E | e = (i, j),\ j \in V\}.$$

Страница	$I(i)$	$\lvert I(i)\rvert$	Рейтинг
1	{(2,1)}	1	0.091
2	{(1,2),(3,2)}	2	0.18
3	{(1,3),(2,3)}	2	0.18
4	{(1,4),(2,4),(3,4),(5,4)}	4	0.36
5	{(1,5)}	1	0.091
6	{(4,6)}	1	0.091

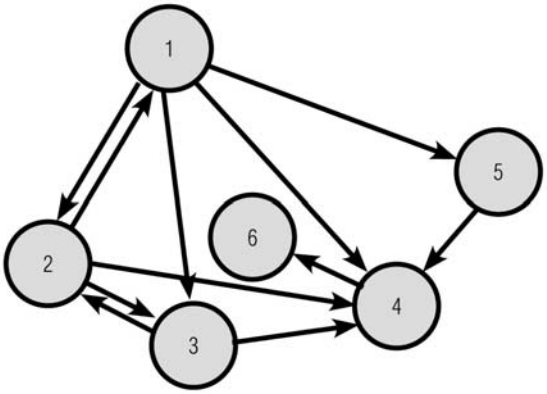

Рис. 1. Индекс цитирования с подсчетом входящих связей.

Определение 3. Документ $i \in V$ называется *висячим*, если $O(i) = \varnothing$.

Пример 2. Тривиальным примером для рейтинга цитируемости будет служить $q(i) = const \cdot |I(i)|$, т.е. документу i присваивается рейтинг прямо пропорциональный числу документов, ссылающихся на него. На *рисунке 1* представлен граф, в котором простой подсчет входящих связей для каждого узла и формирует представленные показатели рейтинга (после нормализации по общему числу связей в графе).

Данный метод ранжирования редко применяется при ранжировании печатных работ или авторов в академической сфере. Очевидный недостаток данного метода заключается в том, что всем цитатам присваиваются равные весовые коэффициенты. Другими словами, цитата автора, на которого имеется много ссылок из других ресурсов, приравнивается цитате автора, не имеющего ссылок с других ресурсов. Кроме того, в таких средах как Веб, данная оценка является абсолютно неадекватной, т.к. основной задачей данного метода является простой подсчет огромного количества входящих ссылок со страниц с низким качеством.

Проблема поиска метода оценки качества ссылок, который бы работал в такой разнородной среде как Веб, наиболее успешно решилась с изобретением алгоритма PageRank. Этот алгоритм был разработан двумя аспирантами Стэнфордского университета: Сергеем Брином и Лоренсом Пейджем, в

дальнейшем он послужил частью технологической базы поисковой системы Google (www.google.com). Впервые алгоритм был описан в [3] и [4], после этого была проведена многолетняя работа в связи с данными публикациями.

6.1. Вычисление рейтинга страницы по алгоритму PageRank

Можно провести аналогию между списками использованной литературы в академических работах и ссылками на определенные страницы в Вебе. Подсчет ссылок на страницу из разных источников дает приближенное значение важности или, другими словами, качества страницы. Алгоритм PageRank расширяет данный подход не только подсчетом количества ссылок (принимая значимость ссылок с каждой из страниц равной), но и упорядочивая страницы по количеству ссылок, содержащихся в них. Рейтинг страницы по PageRank определяется следующим образом [4]:

Предположим, что на документ A ссылаются страницы $T_1 ... T_n$. *А параметр* d *является коэффициентом затухания, находящимся в интервале* $(0;1)$. *Обычно* d *присваивается значение равное 0.85. Коэффициент* d *необходим для того, чтобы ограничить количество переходов по ссылкам в графе документов. Функция* $C(T)$ *определяет количество исходящих со страницы* T *ссылок. Тогда рейтинг страницы A по PageRank определяется как*

$$PR(A) = (1-d) + d(PR(T_1)/C(T_1) + ... + PR(T_n)/C(T_n))$$

Можно видеть, что при вычислении $PR(A)$ *(рейтинга страницы A по PageRank) также учитываются рейтинги страниц* $T_1 ... T_n$ *по PageRank (*$PR(T_k)$*). Таким образом, при определении рейтинга документа во внимание принимается рейтинг страниц, ссылающихся на него, т.е. рейтинг документа зависит от качества ссылающихся на него страниц*

Следует отметить, что PageRank определяет распределение вероятностей для каждой страницы таким образом, что сумма рейтингов PageRank всех страниц будет равна единице.

Рейтинг PageRank ($PR(A)$) может быть вычислен с использованием простого итеративного алгоритма и будет соответствовать главному собственному вектору нормализованной матрицы ссылок. Следует отметить, что рейтинг PageRank для 26 миллионов веб-страниц может быть вычислен за несколько часов на рабочей станции средней мощности [3].

6.2. Наглядное обоснование

Алгоритм PageRank можно рассматривать как модель поведения пользователя. Предполагается, что *веб-серфер* (пользователь, «путешествующий» по веб-страницам, т.е. переходящий по ссылкам с одной на другую) с заданной случайным образом стартовой страницы переходит по ссылкам (снова выбирая их случайным образом) на другие страницы и никогда не возвращается на предыдущую страницу, иногда прерывая переход по ссылкам и начиная снова с другой случайной страницы. Вероятность того, что веб-серфер посетит определенную страницу, и является ее рейтингом PageRank. А коэффициент затухания d определяет насколько скоро веб-серфер начнет процесс заново, перейдя на случайную страницу. Единственное важное различие в задании фактора d заключается в том, что он может быть присвоен как группе страниц, так и отдельным страницам. Данный подход позволяет персонализировать выборку и сводит к минимуму вероятность того, что система ошибется, присваивая странице рейтинг. Существует несколько расширений алгоритма Page Rank, описанных в [5].

Другое наглядное обоснование того, что страница может иметь высокий рейтинг PageRank, заключается в определении количества страниц, ссылающихся на нее и имеющих также высокий рейтинг PagRank. Таким образом, страницы, на которые ссылается множество документов в вебе, являются более предпочтительными. Кроме того, страницы, имеющие хотя бы одну ссылку, например, с домашней страницы Yahoo!, являются более предпочтительными. Если ссылка на страницу не работает, или страница низкого качества, то маловероятно, что домашняя страница Yahoo! будет ссылаться на нее. PageRank анализирует подобные ситуации, а также рекурсивные ссылки нескольких страниц, посредством которых их владельцы пытаются повысить их рейтинг.

6.3. Анкерный текст

Рассмотрим некоторые особенности поисковой системы Google, основой которой является алгоритм PageRank. В поисковой системе Google анкерный текст обрабатывается особым образом. *Анкером* называется слово или группа слов (фраза), к которым привязана гипертекстовая ссылка. Большинство поисковых систем связывают текст ссылки со страницей, на которой эта ссылка находится. В Google анкерный текст так же ассоциируется со страницей, на которую эта ссылка указывает. Дан-

ный подход имеет несколько преимуществ в силу того, что анкеры обрабатываются особым образом. Во-первых, анкеры содержат более точное описание страниц, чем сами страницы. Во-вторых, анкеры могут описывать документы, которые не могут быть проиндексированы системой без графического интерфейса, такие как изображения, приложения и базы данных. Таким образом, становится возможным отбирать веб-страницы, которые фактически не были проиндексированы.

Следует отметить, что неиндексированные страницы могут вызвать некоторые проблемы в силу того, что они не проверялись на точность до представления пользователю. В таких случаях поисковая система никогда не сможет вернуть страницу, которой фактически не существует, однако имеются гиперссылки, указывающие на нее. Тем не менее, сортировка результатов по-прежнему возможна, так как с данной проблемой сталкиваются крайне редко.

Идея привязки анкерного текста к странице, на которую он ссылается, впервые была реализована в World Wide Web Worm [6] именно из-за того, что данный подход позволяет находить информацию, представленную не в виде текста, а также расширяет возможности стандартной поисковой системы. В Google используют анкерную привязку в основном для того, чтобы получить наиболее качественную выборку. Эффективное использование анкерного текста очень проблематично с технической точки зрения: необходимо обрабатывать огромное количество информации. К примеру, для репозитория, содержащего 24 миллиона страниц, было проиндексировано более 259 миллионов анкеров [3].

7. Заключение

Поиск информации в гетерогенной среде, такой как World Wide Web, является актуальной задачей, для которой существует множество методов решения. При поиске данных в репозитории поисковой системы производится выборка проиндексированных документов и определяется их релевантность поисковому запросу, введенному пользователем. Если для определения множества претендентов для выборки подходит логический метод, то для определения релевантности документов используются гораздо более сложные методы, самым эффективным из которых является алгоритм PageRank. Этот алгоритм основан на анализе как ссылок, исходящих из документа, так и документов, ссылающихся на него. При этом также производится оценка качества ссылающихся документов.

Тем не менее, перед поисковыми системами стоит множество проблем, таких как проблема спама и ситуации, когда создатель страницы добавляет искусственную избыточную информацию с целью повысить рейтинг своей страницы. Кроме того, существуют проблемы анализа документов, не имеющих текстовой информации (таких, как изображения или медиа). Для их решения используются различные подходы, которые применяются в крупных поисковых системах и являются коммерческой тайной компаний, их разрабатывающих. В настоящее время ведется активная работа над существующими методами поиска, направленная на их оптимизацию. ■

Литература

1. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. 27 (1948)

2. Sparck-Jones, K., Walker, S., Robertson, S.E.: A probabilistic model of information retrieval: Development and comparative experiments. Inf. Process. Manag. 36(6), 779–808 (2000)

3. Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. In: Proceedings of the 7th International World Wide Web Conference, 1998

4. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. Stanford Digital Libraries Working Paper, Stanford University (1998)

5. Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Manuscript in progress. http://google.stanford.edu/~backrub/pageranksub.ps

6. Oliver A. McBryan. GENVL and WWWW: Tools for Taming the Web. First International Conference on the World Wide Web. CERN, Geneva (Switzerland), May 25-26-27 1994. http://www.cs.colorado.edu/home/mcbryan/mypapers/www94.ps