# МЕТОДЫ ПРЕДВАРИТЕЛЬНОЙ ОБРАБОТКИ ДАННЫХ ДЛЯ АЛГОРИТМА КЛЕЙНБЕРГА[1]

Александр Корявко, Игорь Некрестьянов
Санкт-Петербургский Государственный Университет, 198504, Россия,
Санкт-Петербург, Старый Петергоф, Университетский пр. 28
E-mails: kav@AK11115.spb.edu, nis@acm.org
http://ir.apmath.spbu.ru

Алгоритм ранжирования страниц Веб, предложенный Клейнбергом в 1998 году, является известным представителем группы методов, использующих информацию о связях между страницами Веб. Несмотря на ряд несомненных достоинств, многие исследователи отмечают нестабильность результатов, получаемых с помощью этого алгоритма.

Одним из наиболее популярных способов повышения качества результатов является предварительная обработка исходного набора данных. В этой работе исследуется влияние разных подходов к предварительной обработке данных для алгоритма Клейнберга на качество и устойчивость его работы.

# METHODS OF PRELIMINARY DATA PREPARATION FOR KLEINBERG ALGORITHM

Alexander Koryavko, Igor Nekrestyanov
Saint-Petersburg State University, Universitetsky pr, 28, St.Petergoff,
St.Petersburg, 198504, Russia
E-mails: kav@AK11115.spb.edu, nis@acm.org
http://ir.apmath.spbu.ru

Web-page ranking algorithm proposed by Jon Kleinberg in 1998 is well known algorithm that uses information about links between pages. Despite of some doubtless advantages, many researchers note unstability of results generated by this algorithm.

Preliminary preparation of source data is one of the most popular means to improve results quality. In this paper we study impact of different methods of preliminary data preparation for Kleinberg algorithm on quality and stability of algorithm's results. Also we propose iterative Kleinberg algorithm.

[1] Эта работа частично поддержана грантом РФФИ 01-01-00935.

## 1 Введение

Особенности задачи поиска в Веб – такие как огромный объем доступной информации, «расплывчатость» большинства запросов пользователей, которая влечет их низкую селективность, и неготовность пользователей долго анализировать результаты поиска – стимулируют проведение дополнительных исследований в области алгоритмов ранжирования документов по запросу [2, 17, 15, 18, 3, 14].

Классические подходы к ранжированию опираются на меру схожести текста запроса и текста документа, но бедность типичных запросов пользователей обуславливает невысокую эффективность таких подходов в контексте Веб.

Поэтому в течение ряда последних лет активно исследуются альтернативные подходы к оценке «полезности» страниц для данного пользователя, которые опираются не только на информацию о содержимом документа, но также и на метаинформацию как о документе, так и о пользователе.

Например, сервисы типа DirectHit сохраняют информацию о запросе пользователя и о том, какие документы и сколько времени пользователь просматривал после выполнения запроса. Эта информация учитывается при ранжировании документов так, чтобы первыми показывались те документы, которые просматривало большинство пользователей, задававших точно такой же запрос. Такой подход требует накопления большого количества информации про каждый запрос, и поэтому слабо масштабируем по числу запросов.

Перспективным выглядит применение подходов, основанных на анализе структуры графа Веб [2, 12].

Наиболее известным представителем таких подходов является алгоритм PageRank, предложенный в 1998 году и используемый системой Google [18]. Он основывается на идее использования вероятности попадания «блуждающего случайным образом» (random surfer model) на конкретную страницу Веб в качестве ее ранга. Первоначальная версия PageRank позволяла вычислить единственный ранг страницы вне зависимости от какого-либо контекста, но недавно предложенная модификация дает возможность обойти это ограничение и вычислять ранг в контексте некоторой тематики [13].

Почти одновременно с PageRank'ом Клейнбергом был разработан алгоритм HITS (Hyperlink Induced Topic Search) [15]. В отличие от PageRank в этом подходе выделяются две различные роли страниц – *первоисточника информации (authority)* и *посредника (hub)*, а также анализируется лишь структура относительно небольшого подграфа Веб, который строится по исходному запросу пользователя.

Однако «локальная» природа алгоритма обуславливает низкую предсказуемость качества результатов, которая сильно зависит от выбранного подграфа Веб [2].

В этой работе исследуется влияние разных подходов к предварительной обработке данных для алгоритма Клейнберга на качество и устойчивость его работы.

Статья организована следующим образом: в следующем разделе изложен алгоритм Клейнберга; в разделе 3 описаны рассматриваемые нами методы предварительной обработки данных, а в разделах 4 и 5 представлены экспериментальные результаты.

## 2 Алгоритм Клейнберга

Как уже было отмечено выше, Клейнберг предложил рассматривать две разные роли страниц Веб – роль *первоисточника*, характеризующую ценность информации на этой странице, и роль *посредника*, характеризующую ценность информации на страницах, доступных по ссылкам с этой страницы [15].

Такой подход мотивирован наличием в Веб большого числа тематических сообществ, т.е. наборов страниц близкой тематики, которые сильно связаны друг с другом ссылками. Типичный вид такого сообщества приведен на рисунке 1.

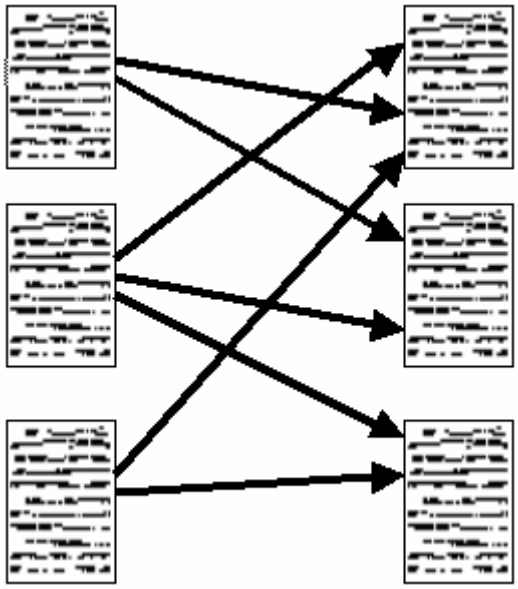

**Рисунок 1:** Структура тематического сообщества. Слева – ярко выраженные посредники, справа – хорошие первоисточники

Выделение ролей наиболее осмыслено в рамках некоторого локального тематического контекста (т.е. полезность страницы как первоисточ-

ника естественным образом зависит от темы запроса), что и обуславливает локальную природу алгоритма.

В работе алгоритма можно выделить две основные стадии – построение подграфа Веб и анализ этого подграфа для вычисления рангов конкретных страниц. Мы рассмотрим их по отдельности в последующих разделах.

## 2.1 Построение подграфа Веб

Процедура построения подграфа Веб опирается на использование какой-нибудь поисковой системы для Веб с относительно хорошим покрытием индекса. По исходному запросу подграф строится следующим образом:

1. **Построение *RootSet***

Это множество формируется из ***k*** (обычно порядка 200) первых результатов, возвращенных используемой поисковой системой для исходного запроса.

2. **Построение *BaseSet***

Это множество получается при помощи расширения ***RootSet*** за счет окрестностей страниц из ***RootSet***. Таким образом добавляются страницы, которые содержат ссылки на страницы из ***RootSet*** или, наоборот, ссылки на которые содержатся в каких-нибудь страницах из ***RootSet***. Для обнаружения страниц первого вида (т.е. страниц с *входящими* ссылками) также используются возможности поисковой системы общего назначения. При этом из вычислительных соображений обычно ограничивают максимальное число ***d*** учитываемых.

Искомый подграф Веб, который далее используется для вычисления рангов страниц, получается сужением полного графа Веб на ***BaseSet***, из которого удалены все внутридоменные ссылки. Удаление внутридоменных ссылок – это простейшая эвристика для подавления навигационных и протекционных ссылок, которые вызывают искажение результатов ранжирования.

Взаимосвязь между множествами ***BaseSet*** и ***RootSet*** проиллюстрирована на рисунке 2.

## 2.2 Вычисление рангов страниц Веб

Как уже было отмечено выше, каждой странице из построенного подмножества страниц Веб сопоставляется два ранга – ранг первоисточника и ранг посредника.

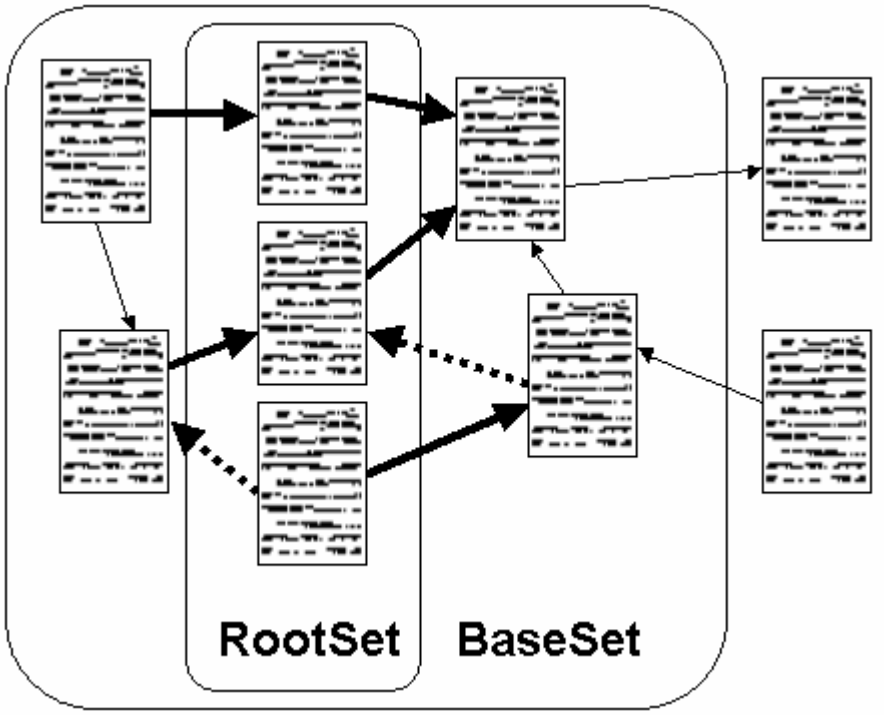


**Рисунок 2:** ***BaseSet***. Жирные стрелки – ссылки, которые участвовали в его образовании

Неформально понятно, что страница является хорошим посредником, если она содержит ссылки на ценные первоисточники, и наоборот, страница является хорошим первоисточником, если она упоминается хорошими посредниками.

Более формально это можно записать так:

$$A(u) = \sum_{v \to u} H(v) \qquad H(v) = \sum_{v \to u} A(u)$$

Здесь ***A(v)*** и ***H(v)*** – ранги страницы ***v*** как первоисточника и посредника соответственно. Обозначив за ***S*** матрицу смежности[2] подграфа, построенного на предыдущем этапе, формулы вычисления рангов можно переписать следующим образом:

$$A = S^T H \qquad H = SA$$

Подставив одно уравнение в другое, получаем, что ***A*** и ***H*** являются собственными векторами[3] соответственно матрицы социтирования $\boldsymbol{S^T S}$ и матрицы «библиографических сочетаний» (bibliographic coupling) $\boldsymbol{SS^T}$.

---

[2] Т.е. $\boldsymbol{S_{uv}=1}$, если существует дуга, ведущая из ***u*** в ***v***, и $\boldsymbol{S_{uv}=0}$ в противном случае.

[3] Таким образом, для эффективного вычисления рангов можно использовать стандартные численные методы [5,4].

## 2.3 Проблема смещения тематики

Алгоритм Клейнберга может повысить качество ранжирования не только за счет переупорядочивания страниц, которые обнаружила поисковая система. При построении ***BaseSet*** в него могут попасть (и затем быть высоко ранжированными) страницы, которые поисковая система вообще не считает релевантными из-за того, что они по тем или иным причинам не содержат слова из запроса.

Однако, именно эта особенность алгоритма обуславливает и отсутствие стабильности качества результатов HITS.

Метод вычисления рангов (см. раздел 2.2) влечет рост рангов страниц при увеличении количества и степени связанности страниц соответствующего сообщества.

В том случае, когда в ***BaseSet*** попадает много страниц на другие темы, и сообщество, соответствующее исходной теме, не является доминирующим, это свойство обуславливает присвоение наивысших рангов страницам на другую тему. Такой эффект получил название *смещения тематики* (topic drift).

Обычно такое смещение происходит в направлении более широкой предметной области (или лучше представленной в Веб). Например, запрос «WWW conferences» имеет тенденцию смещаться к теме «WWW».

## 2.4 Модификации HITS

На сегодняшний день известно довольно много модификаций HITS, которые условно можно разделить на три группы. К первой относятся надстройки над HITS, которые, например, предлагают изменить построение матрицы смежности, но не меняют принцип ранжирования [6,9].

Ко второй группе относятся попытки изменить сам алгоритм – например, используя вероятностный подход [11,16] или динамически изменяя гранулярность исходных документов [8,10].

Последнюю группу составляют методы, которые допускают, что самое крупное обнаруженное сообщество не является самым ценным, и пытаются автоматически определить наиболее полезное сообщество среди всех обнаруженных. Так, например, Клейнберг пробовал оценивать ценность сообщества как суммарную близость нескольких наиболее важных документов из сообщества к исходному запросу [15].

Обзор и сравнение эффективности некоторых модификаций HITS можно найти в [7].

# 3 Методы предварительной обработки

Целью этой работы является исследование методов предварительной обработки подграфа Веб, на основе которого происходит вычисление рангов (см. раздел 2.1).

Мы выделяем две группы подходов.

### 3.1 Однопроходные методы

К этой группе методов относятся все подходы, которые производят одноразовую модификацию подграфа перед вычислением рангов страниц.

В этом случае процедура обработки сводится к предварительному анализу и модификации матрицы смежности, построенной по ***BaseSet***.

#### 3.1.1 Постраничная обработка

К методам этого вида относятся подходы, которые по результатам оценки полезности отдельных страниц подграфа принимают решение об их возможном удалении. Например:

**M1.** *Близость к исходному запросу.*

Близость страницы к исходному запросу оценивается при помощи традиционной меры тематической схожести – скалярного произведения запроса и профиля документа с ***tfidf*** весами [19].

**M2.** *Тематическая близость к* ***RootSet****.*

Сравнение страниц с эталонным документом. Эталонный документ порождается как объединение документов из всего ***RootSet*** или его части. Сравнение осуществляется также при помощи ***tfidf*** и скалярного произведения.

#### 3.1.2 Обработка на уровне отдельных ссылок

В этой группе методов оценивается полезность индивидуальных ссылок, на основе анализа документов, которые они соединяют. По результатам этого анализа информация о малополезных ссылках удаляется из матрицы смежности.

К методам этой группы относятся:

**M3.** *Удаление внутридоменных ссылок*.

Информация о ссылках, соединяющих страницы из одного домена, удаляется. Это, вообще говоря, классическая эвристика, которая была предложена еще в работах Клейнберга.

**M4.** *Тематическая схожесть страниц*.

Полезность ссылки оценивается также при помощи скалярного произведения профилей документов, которые она соединяет, с ***tfidf*** весами [19].

### 3.2 Итеративные методы

К этой группе методов относятся методы, которые могут несколько раз производить обработку набора данных и вычисление рангов. При этом вычисленные ранги могут учитываться в процессе обработки подграфа на следующей итерации.

- *Итеративное сокращение*

В случае, если главный собственный вектор плохо соответствует тематике запроса, то это скорее всего означает, что используемый граф содержит слишком много шума. Для того, чтобы исправить эту ситуацию, можно удалить из матрицы смежности наиболее авторитетные страницы (или только ссылки) соответствующие главному собственному вектору.

После пересчета рангов на модифицированной матрице смежности главный собственный вектор станет представлять следующее по представительности сообщество в рассматриваемой окрестности. Идея этого подхода состоит в повторении процесса до тех пор, пока искомое сообщество не станет самым представительным.

- *Итеративное уточнение*

В отличие от предыдущего подхода в этом случае по результатам итерации строится новый ***RootSet***, и алгоритм Клейнберга применяется к нему. Выбор ***RootSet*** ограничен, с одной стороны, желанием не допустить в него много мусора, но с другой стороны, слишком маленький ***RootSet*** может быть сильно подвержен влиянию шумов.

Таким образом, матрица смежности на последующих итерациях может не только сокращаться, но и пополняться, хотя вряд ли станет больше первоначальной. Процесс останавливается, когда разница результатов между итерациями становится незначительной или по истечении заданного числа итераций.

Отметим, что в обоих случаях на каждой итерации возможно применение и однопроходных подходов для обработки текущего представления графа.

## 4 Постановка экспериментов

Для эмпирического изучения реальной эффективности различных методов предварительной обработки мы реализовали прототип системы, ранжирующей результаты обработки поискового запроса.

### 4.1 Набор данных

В качестве базовой ИПС мы использовали систему *Яндекс* (`www.yandex.ru`) и поэтому формально набором данных, по которому производился поиск, является множество страниц Веб, которые проиндексированы Яндекс[4].

Сводная статистика об использовавшихся в наших экспериментах запросах представлена в таблице 1. Отметим, что обработка одного запроса требовала до тысячи обращений к ИПС (для нахождения страниц-предков).

### 4.2 Критерии оценки качества

Качество ранжирования – это безусловно субъективный критерий, который невозможно формализовать, и, как следствие, невозможно оценить точно [1].

Поэтому для оценки качества мы используем некоторые количественные меры, предполагая, что они коррелируют с искомым критерием.

| Запрос | ***\|RootSet\|*** | ***\|BaseSet\|*** |
|---|---|---|
| атеизм | 200 | 2272 |
| электронные библиотеки | 172 | 5698 |
| русский tex | 200 | 2242 |
| программирование | 200 | 6847 |
| crack | 178 | 3748 |
| ленинград | 200 | 2540 |

Таблица 1: **Характеристика тестовых запросов**

#### 4.2.1 Релевантность страниц

Мы рассматривали два способа оценки релевантности ***g(d)***отдельных страниц ***d*** – косвенная автоматическая оценка и экспертные оценки.

В первом случае релевантность страницы оценивалась как скалярное произведение запроса и профиля страницы с ***tfidf*** весами [19]. Конечно, такой подход не позволяет получить точную оценку, так как тестирует лишь на объективную встречаемость слов из запроса, а не на полезность документа как ответа.

[4] На 20 мая 2002 года объем проиндексированной Яндекс информации составил более 63 миллионов документов общим объемом в терабайт.

Экспертные оценки, напротив, отражают субъективное мнение пользователей. Вообще говоря, известно, что мнение эксперта меняется во времени, и поэтому даже при использовании одного эксперта собранные оценки могут плохо согласовываться друг с другом [1]. Однако, поскольку нас интересует лишь относительное сравнение качества при использовании разных методов, то погрешность собираемых оценок для нас не очень принципиальна.

Для сбора экспертных оценок использовалась шкала от 0 до 1 с шагом 0.1. Большой объем данных делает невозможным сбор оценок для всех документов, поэтому мы использовали разновидность «техники общего котла» [1], предъявляя экспертам несколько высоко ранжированных документов из наиболее крупных сообществ[5]. Релевантность документов, для которых экспертные оценки отсутствовали, полагалась равной 0.

### 4.2.2 Меры для оценки качества ранжирования

Для оценки качества ранжирования $r=(d_1,\ldots,d_n)$ на уровне $k$ ($k<n$) использовалось несколько известных мер [1]:

1. *Совокупная выгода (cumulated gain):*

$$CG_k(r) = \sum_{i=1}^{k} g(d_i)$$

2. *Обесцениваемая совокупная выгода (discounted cumulated gain):*

$$DCG_k(r) = g(d_1) + \sum_{i=2}^{k} \frac{g(d_i)}{\log(i)}$$

Отметим, что в качестве $g(d)$ здесь может быть использована как автоматическая оценка, так и экспертная. Несмотря на свою неточность, автоматическая оценка позволяет составить предварительное мнение о сравнительной эффективности того или иного подхода, которое впоследствии можно уточнить, собрав оценки экспертов.

Такие меры применяются по отдельности к набору первоисточников и к набору посредников. Чтобы целиком оценить сообщество, состоящее из вектора первоисточников $r^a$ и вектора посредников $r^h$, мы использовали следующую формулу:

[5] На самом деле для запроса *атеизм* было собрано 1585 экспертных оценок, т.е. было оценено две трети всех страниц.

$$CW_k(r^a, r^h) = \frac{CG_k(r^a)}{\sum_i CG_k(r_i^a)} + \frac{CG_k(r^h)}{\sum_i CG_k(r_i^h)}$$

Здесь суммирование осуществляется по всем найденным векторам, вместо $\boldsymbol{CG_k}$ можно использовать $\boldsymbol{DCG_k}$.

### 4.2.3 Характеристики метода обработки

Поскольку в этой статье нас в первую очередь интересует влияние методов предварительной обработки данных, то мы заинтересованы в сравнении их объективных характеристик.

Например, такими параметрами являются евклидова норма матрицы смежности $\|\boldsymbol{S}\|$ и степень ее разреженности, которую удобно оценивать с помощью доли удаленных из исходной матрицы ссылок $\boldsymbol{RL(S)}$.

## 5 Результаты экспериментов

Для проведения экспериментов мы использовали ряд типичных запросов пользователей поисковых систем для Веб, некоторые из которых перечислены в таблице 1. Разный размер ***RootSet*** объясняется тем, что не всегда поисковой системе удалось найти 200 и более документов, которые предполагалось использовать в алгоритме, поэтому пришлось довольствоваться тем, что есть.

К сожалению, большой объем работы по проведению экспериментов и сбору экспертных оценок не позволил нам детально проанализировать и обобщить большое количество запросов. Поэтому в рамках этой статьи[6] мы ограничимся лишь некоторыми результатами, полученными при анализе наиболее интересного из исследованных запросов – запроса *атеизм*.

Для более ясного описания результатов мы будем использовать понятие *нерелевантных* документов для описания не имеющих отношения к теме запроса страниц, и понятие *слаборелевантных* страниц для страниц, которые имеют некоторое отношение к теме запроса, но имеют низкую ценность в силу других причин (например, неглавные страницы сайтов, специально посвященных теме запроса).

### 5.1 Однопроходные эвристики

В ходе проведения экспериментов мы применяли как отдельные, так и их комбинации. Основные результаты этих экспериментов собраны в таблице 2.

---

[6] Более подробную информацию о полученных результатах можно найти по адресу `http://ir.apmath.spbu.ru/~kav/HITS/`.

| Метод | Автоматический выбор | | Экспертная оценка | | Главные вектора | | $\|\|S\|\|$ | $RL(S)$ |
|---|---|---|---|---|---|---|---|---|
| | $CG_{25}$ | $DCG_{25}$ | $CG_{25}$ | $DCG_{25}$ | $CG_{25}$ | $DCG_{25}$ | | |
| Яндекс | — | — | — | — | 12+12 | 7.06+7.06 | — | — |
| **M0** | 34(6.1+4.5) | 34(3.21+3.67) | 35(11.9+11.3) | 35(6.15+5.39) | 1.4+0.5 | 1.06+0.17 | 200.197 | 0% |
| ***M1*** | 2(8.3+11.6) | 2(3.67+5.51) | 1(18.1+19) | 10(9.44+8.16) | 0.5+0.5 | 0.17+0.17 | 94.435 | 78% |
| ***M2*** | 0(1.5+4.6) | 15(0.29+0.89) | 8(20.3+20.3) | 8(10.36+10.10) | 0+0 | 0+0 | 21.105 | 70% |
| ***M3*** | 49(8.7+5.5) | 17(9.13+4.64) | 17(17.5+6.6) | 17(9.13+4.64) | 0+0.9 | 0+0.9 | 115.212 | 67% |
| ***M4*** | 17(3.5+3.5) | 17(2.77+2.77) | 19(14.5+19.1) | 19(8.15+9.89) | 0+0 | 0+0 | 118.585 | ~0% |
| ***M3+M4*** | 12(16.9+15.0) | 12(9.14+8.29) | 12(16.9+15.0) | 12(9.14+8.29) | 0+3.5 | 0+2.46 | 16.770 | 67% |
| ***M1+M3*** | 15(14.2+11.7) | 20(6.45+5.10) | 24(16.5+20.6) | 24(8.56+10.33) | 13.3+18.4 | 7.80+9.53 | 14.764 | 88% |

Таблица 2: **Результаты для запроса атеизм. Представлены лучшие вектора, выбранные по мере $CW_{25}$. В скобках указано качество сообщества как сумма оценок набора первоисточников и набора посредников соответственно**

Как и следовало ожидать, сообщества, полученные при применении HITS к полной матрице (***M0***), состоят в основном из страниц с одного сайта или маленькой группы сайтов. Лучшее сообщество в этом случае содержит много нерелевантных страниц, а также некоторое количество малорелевантных.

Взвешивание ссылок в полной матрице (***M4***) приводит к тому, что лучшие сообщества представляют собой объединение небольшого количества клик, где клика – набор страниц, которые сильно связаны друг с другом ссылками (например, образуют полный граф) и потому высоко ранжируются HITS.

Похожий результат дает и удаление нерелевантных страниц на основе сравнения с запросом (***M1***), но в этом случае преобладают клики, образованные слаборелевантными страницами.

Удаление внутридоменных ссылок (***M3***) привело к обнаружению совершено нерелевантного сообщества, соответствующего главному собственному вектору, но в то же время одно из неглавных сообществ первоисточников получилось очень хорошим, практически не содержащее нерелевантных или слаборелевантных документов. К сожалению, качество соответствующего вектора посредников оставляет желать лучшего.

Более того, в этом случае корреляция между качеством векторов первоисточников и посредников практически отстуствует. Это, в частности, сказывается на эффективности автоматического выбора векторов, поскольку оценка векторов для посредников работает лучше, чем для первоисточников.

Частично ситуация улучшилась при сравнении страниц с ***RootSet*** (***M2***), где также обнаружилось весьма хорошее неглавное сообщество первоисточников, которому соответствовал очень неплохой набор посредни-

ков. При исключении внутридоменных ссылок с последующим взвешиванием (***M3+M4***), результаты становятся заметно лучше, и наблюдается корреляция между качеством векторов.

Тем не менее, главные собственные вектора имеют низкое качество. Такое поведение объясняется тем, что в ***BaseSet*** присутствуют распределенные сайты (использующие разные Веб сервера для представления разных разделов сайта, но сильно связанные ссылками), и удаление внутридоменных ссылок тут бессильно (это ограничение можно обойти, используя модификацию HITS [8]).

Совместное использование метода исключения внутридоменных ссылок, сравнения страниц с запросом и взвешивание ссылок с различными параметрами (***M1+M3***) резко повышает степень разреженности матрицы. Однако главный собственный вектор порождает сообщество, в целом хорошо удовлетворяющее запросу. Отметим, что порядок страниц в этом сообществе сильно похож на порядок, порождаемый простым счетчиком ссылок на страницы в построенной матрице.

### 5.2 Итеративный подход

Эксперименты с *итеративным уточнением* проводились без скачивания новых страниц из Веб (но в матрице могли появляться новые элементы за счет ранее скачанных страниц). Результаты, полученные при использовании экспертных оценок для определения качества векторов, представлены в таблице 3. Интересно, что первые несколько шагов приводят к повышению качества результата – как качества объективно лучшего вектора, так и качества главного вектора, но потом наблюдается регрессия. Это поведение требует дополнительного изучения.

При проведении того же эксперимента, но с автоматическим выбором векторов и взвешиванием ссылок по схожести страниц, также

наблюдается улучшение качества результата, хоть и менее значительное, но зато монотонное. Тем не менее, лучший результат уступает лучшему результату из таблицы 3.

### 5.3 Типы сообществ

Большинство примеров тематических сообществ, упоминаемых в литературе, состоят (как правило) из больших сайтов, которые играют роль первоисточников и внешних подборок ссылок, которые являются посредниками. Такое сообщество можно назвать *гетерогенным*.

С другой ситуацией мы столкнулись при анализе запроса *атеизм*. Здесь лучшие страницы ссылок как раз принадлежат самым авторитетным первоисточникам. Такое сообщество можно назвать *гомогенным*. Отметим, что в отличие от протекционных сетей (типа LinkExchange), в этом случае ссылки имеют не протекционный характер.

| Итерация | Лучшие первоисточники | | Лучшие посредники | | Лучшие сообщества | |
|---|---|---|---|---|---|---|
| | Вектор | $CG_{25}$ | Вектор | $CG_{25}$ | Вектор | $CW_{25}$ |
| 0 | 17 | (**17.5**+6.6) | 16 | (2.7+**19**) | 17 | (0.109) |
| | 27 | (**12**+4.7) | 34 | (1.1+**16.1**) | 27 | (0.075) |
| | 49 | (**9.8**+0) | 44 | (5.1+**12.7**) | 49 | (0.059) |
| 1 | 4 | (**17.1**+11) | 18 | (3.2+**18.4**) | 3 | (0.046) |
| | 3 | (**16.9**+17.2) | 28 | (5.6+**17.9**) | 4 | (0.041) |
| | 7 | (**15.4**+12.4) | 4 | (8.6+**17.4**) | 7 | (0.039) |
| **2** | **1** | (**18.9**+16.3) | **16** | (11.3+**20.6**) | **1** | **(0.041)** |
| | **3** | (**15.4**+15.6) | **43** | (7.9+**18.9**) | **0** | **(0.037)** |
| | **0** | (**15.3**+18.3) | **14** | (5.4+**18.9**) | **8** | **(0.036)** |
| 3 | 9 | (**16.7**+14.5) | 9 | (16.7+**14.5**) | 9 | (0.056) |
| | 7 | (**15.7**+11.3) | 7 | (9.8+**13.8**) | 7 | (0.050) |
| | 11 | (**14**+6.3) | 14 | (0.9+**13.7**) | 7 | (0.040) |
| 4 | 7 | (**16.9**+3.9) | 3 | (8.7+**13.3**) | 7 | (0.047) |
| | 15 | (**14.9**+5.1) | 17 | (3.5+**13.2**) | 6 | (0.047) |
| | 22 | (**13.6**+8) | 7 | (0+**13.1**) | 22 | (0.046) |
| 5 | 12 | (**16.2**+8.7) | 5 | (11.2+**16.1**) | 12 | (0.056) |
| | 17 | (**15.1**+6) | 16 | (3.6+**14.5**) | 5 | (0.055) |
| | 10 | (**14.5**+7.1) | 17 | (4.5+**13.6**) | 10 | (0.049) |
| 6 | 13 | (**17.2**+7.7) | 10 | (0+**16.8**) | 13 | (0.053) |
| | 15 | (**15.3**+3.1) | 7 | (0+**15.4**) | 15 | (0.042) |
| | 19 | (**10**+4.8) | 30 | (4.4+**14.8**) | 6 | (0.037) |

Таблица 3: **Лучшие собственные вектора и их оценки, полученные итеративным вариантом алгоритма**

В гетерогенном случае подборки ссылок содержат слова из запроса пользователя, именно они попадают в ***RootSet***. Являющиеся хорошими первоисточниками сайты, в свою очередь, признаются не наиболее релевантными запросу традиционными ИПС и попадают в построенный ***BaseSet*** при расширении ***RootSet***. Более того, такие сайты обычно содержат мало ссылок друг на друга (поскольку они зачастую представляют собой конкурирующие источники информации – например, сайт Internet Explorer не содержит ссылок на сайты других Веб-броузеров; или просто

физически не могут отслеживать положение дел в Интернете, чтобы поддерживать качественные наборы ссылок).

Для таких сообществ построенный ***RootSet*** содержит много хороших посредников. В гомогенном случае первоисточники высоко ранжируются ИПС и попадают в ***RootSet***, но тем самым «экранируют» свои страницы-посредники. В ***RootSet*** оказываются только смешанные посредники, содержащие, кроме ссылок на авторитетные сайты, еще и массу посторонних. К таковым, в частности, относятся тематические каталоги, которые вызывают появление большей части нерелевантных документов. Таким образом, ***RootSet*** оказывается состоящим из хороших первоисточников и плохих посредников, и это, видимо, отрицательно сказывается на качестве ранжирования.

Гомогенные сообщества, по-видимому, также отличаются существенно меньшим размером, хотя у нас безусловно недостаточно статистической информации, чтобы подтвердить или опровергнуть эту гипотезу. Однако, снижение размера сообщества значительно повышает важность шага построения ***RootSet*** и его предварительной обработки, как способа повышения устойчивости качества работы HITS.

## 6 Заключение

Огромный объем и особенности пользователей Веб обуславливают интерес к исследованиям в области ранжирования результатов поиска.

В этой работе рассматривается одним из наиболее перспективных алгоритмов ранжирования учитывающих структуру графа Веб – алгоритм Клейнберга.

В работе исследуется влияние методов предварительной обработки данных на качество и устойчивость получаемых результатов.

Хотя на данный момент наши эксперименты не позволяют делать статистически значимых выводов, но они уже позволяют сформулировать новые интересные предположения – например, гипотезу о существовании разных видов тематических сообществ и итеративный вариант алгоритма Клейнберга.

В дальнейших исследованиях мы предполагаем не только закончить исследование стабильности HITS, но также надеемся получить модификацию HITS, качественно работающую в случае небольших гомогенных сообществ. Перспективной также выглядит попытка использования формального аппарата теории возмущений для анализа стабильности алгоритма.

**Благодарности**

Мы хотели бы выразить свою благодарность компании Яндекс, которая любезно разрешила нам делать автоматические запросы к своей поисковой системе *www.yandex.ru* при проведении этого исследования.

Также нам хотелось бы выразить признательность ЗАО «Ланит-Терком» (*http://www.tercom.ru/*) за материально-техническое содействие в проведении данного исследования.

**Литература**

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

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

[3] Dimitris Achlioptas, Amos Fiat, Anna R. Karlin, and Frank McSherry. Web search via hub synthesis. In *IEEE Symposium on Foundations of Computer Science*, pages 500-509, 2001.

[4] Dimitris Achlioptas and Frank McSherry. Fast computation of low rank matrix approximations. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 2001.

[5] Michael Berry, Zlatko Drmac, and Elizabeth Jessup. Matrices, vector spaces and information retrieval. *SIAM Review*, 41(2):335-362, 1999.

[6] Krishna Bharat and Monika Rauch Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Proc. Research and Development in Information Retrieval*, pages 104-111, 1998.

[7] Alan Borodin, Gareth O. Roberts, Jeffrey S. Rosenthal, and Panayiotis Tsaparas. Finding authorities and hubs from link structures on the world wide web. In *Proc. WWW10*, pages 415-429, 2001.

[8] Soumen Chakrabarti. Integrating the document object model with hyperlinks for enhanced topic distillation and information extraction. In *Proc. WWW10*, 2001.

[9] Soumen Chakrabarti, Byron E. Dom, S. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins, David Gibson, and Jon Kleinberg. Mining the Web's link structure. *Computer*, 32(8):60-67, 1999.

[10] Soumen Chakrabarti, Mukul Joshi, and Vivek Tawde. Enhanced topic distillation using text, markup tags, and hyperlinks. In *Proc. SIGIR*, 2001.

[11] David Cohn and Huan Chang. Learning to probabilistically identify authoritative documents. In *Proc. 17th International Conf. on Machine Learning*, pages 167-174. Morgan Kaufmann, San Francisco, CA, 2000.

[12] Cathal Gurrin and Alan Smeaton. A connectivity analysis approach to increasing precision in retrieval from hyperlinked documents. In *Proc. of teh TREC'8*, November 1999.
[13] Tahec Haveliwala. Topic-sensitive pageRank. In *Proc. of the WWW'2002*, May 2002.
[14] Tahec Haveliwala, Aristidis Gionis, dan Klein, and Piotr Indyk. Evaluating strategies for similiarity search on the web. In *Proc. of the WWW'2002*, May 2002.
[15] Jon M. Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S. Tomkins. The Web as a graph: Measurements, models, and methods. In *Proc. 5th Annual Int. Conf. Computing and Combinatorics, COCOON*, number 1627. Springer-Verlag, 1999.
[16] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (salsa) and the tkc effect. In *Proc. WWW9*, 2000.
[17] Maxim Lifantsev. Voting model for ranking web pages. In *Proc. International Conference on Internet Computing*, pages 143-148, 2000.
[18] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, 1998.
[19] G. Salton. *Automatic Text Processing*. Addison-Wesley Longman Publ. Co., 1988.