# Анализ проблем навигации в мобильном представлении научной информации

**А.М.Бородин, С.В. Поршнев**
**Уральский Федеральный Университет, г.Екатеринбург**

## Введение

КонецXX − началоXXI вв.стали в истории человечества точкой отсчета экспоненциально быстрого роста объемов доступной информации. Сегодня данный факт является общепризнанным и осознается, в том числе и неспециалистами в области информационных технологий. Увеличение объёмов доступной информации сопровождается одновременным увеличением «дисперсии» информации − необходимые пользователю данные размываются в океане сопутствующих, связанных с требуемой информацией сведений. При этом скорость роста «дисперсии» равняется скорости роста объемов информации.

В этой связи даже научная информация, сохраняемая и систематизируемая сотрудниками той или иной научной школы, с течением времени обнаруживает существенное увеличение длительности выполнения поисковых запросов − «диффузионная девальвация». Необходимо отметить, что истоки проблемы экспоненциально быстрого роста информации имманентно присущи современным технологиям работы с информацией. Действительно, в модель основной информационной магистрали (Интернет) изначально заложена идея информационного роста; протокол HTTP – основной транспортный протокол в своём названии содержал идею связности, гипертекста; основной способ адресации данных–URI, содержит идею глобальности данных, а в своей структуре упоминание связности и глобальности (префикс «www.»,напоминающий об этом, де-факто является стандартом).

Проблемы оценки качества способов ранжирования и классификации научной информации разрабатываются наукометрией–дисциплиной, изучающей эволюцию науки через измерения научной информации. При этом необходимо отметить, что областью исследований наукометрии является наука, рассматриваемая в качестве некоторой централизованной научной среды. Данный подход, определявший, в свою очередь, выбор методов исследования, используемых в наукометрии, сегодня, в условиях распределённости научных исследований, в известной мере, становится сдерживающим фактором. Здесь, под термином «распределённость» мы понимаем не только и не столько распределение информации по географическому принципу, сколько методологическим, мотивационным, структурным принципам. В качестве примеров подобной распределённости можно привести информационные системыGoogleXLabs[1], KonturLabs[2].

В этой ситуации, с нашей точки зрения, требуется не всеобщий метод ранжирования и классификациинаучной информации, позволяющего провести глобальное сравнение важности каких-либо публикаций, но методика *персональной навигации* в ней − способ автоматизации выбора направления в графе научных документов, представляющего интерес для конкретного потребителя научной информации.

## Постановка задачи

Предположим, что имеется некоторая сеть *N*, состоящая из адресов документов $d_1, d_2, \dots, d_n \subset N$ . Пусть для каждого адреса документа определено некоторое представление данных документа $V(d_x)$ такое, что оно включает некоторое множество *A* адресов других документов $A = \{d_{dx1}, d_{dx2}, \dots, d_{dxn}\} \subset V(dx)$. Тогда для множества обучающих адресов документов *F* можно определить функционал навигации *n*:

$$(d, F, V) \xrightarrow{n} R^1,$$

либо в менее точном виде

$$(d, F, V, C) \xrightarrow{n} R^1,$$

Где *C* – некоторый контекст навигации,$R^1$–линейно упорядоченное множество.

Задачей исследования является разработка алгоритма расчёта *n*из возможных составляющих контекста *C* алгоритма ранжирования ссылок между документами и выделение ключевых параметров данного алгоритма.

**Структура алгоритма**

Предваряя обсуждение структуры разрабатываемого алгоритма, отметим, что в условиях широкого распространения и повсеместного использования портативных мобильных устройств, имеющих ограниченные вычислительные ресурсы, очевидно, эффективные реализации алгоритма могут быть выполнены только на платформе облачных вычислений [3]. Выбранный подход накладывает ограничения по прозрачности и безопасности сетевого API алгоритма навигации, позволяя реализовать централизованное кэширование результатов и обеспечить обработку достаточно больших объёмов данных.

Работа алгоритма с точки зрения организации сетевого взаимодействия выглядит следующим образом:

1. Перед началом работы браузер клиента устанавливает https[4] соединение с навигационным сервером и аутентифицирует клиента. В дальнейшем взаимодействие браузера с навигационным сервером проходит средствами httpтранспорта.
2. Браузер по команде пользователя отправляет http [5] запрос на контент-сервер за требуемым документом.
3. Браузер информирует навигационный сервер о намерении просмотреть документ по определённому адресу.
4. Браузер по завершении загрузки ответа от контент-сервера генерирует хэш-коды параграфов документа и отсылает их на навигационный сервер, асинхронно ожидая ответа.
5. Начиная с шага 3 навигационный сервер выполняет httpзапрос на контент-сервер по полученному адресу. Для параграфов, у которых хэш-код на клиенте и сервере совпадает, навигационный сервер генерирует рекомендации по подсветке ссылок в соответствии с уровнем их потенциальной важности для пользователя. Сгенерированная информация отправляется ответом браузеру клиента. Кэширование ответов осуществляется согласно комбинацииETag и Expired заголовка http ответа контент-сервера и идентификатора пользователя.
6. Браузер клиента, получив ответ, осуществляет предписанную навигационным сервером визуализацию.

**Функционирование навигационного сервера**

Использование программной реализация описанного выше алгоритма на навигационном сервере, предполагает применение алгоритмов машинного обучения [6] − тренировку классификатора по обучающей выборке. Здесь основным параметром алгоритма является *F*–набор обучающих документов – статей, заданных пользователем алгоритма, поэтому именно они рассматриваются далее в контексте задачи персональной навигации.

Представление данных документа *V* технически состоит из двух основных компонентов: выделения текста и выделения ссылок. Выделение текста, по сути, представляет собой очистку электронного документа от html-разметки, что является тривиальной задачей, за исключением случая очистки от html-разметки статей об html-

разметке. Выделение ссылок из научной литературы – более сложная техническая задача, так как ссылку, фактически, нужно преобразовать к адресу документа, что не всегда возможно, поэтому данную задачу приходится решать отдельно для каждого крупного online-публикатора (ACM,IEEE). Здесь наиболее тривиальным случаем является http-ссылка.

Для обработки каждой статьи используется выделение терминов и терминоподобных конструкций. Обзор технологий выделения терминов можно найти в [7]. Рассмотрим кратко метод *C-value*[8], являющийся одним из наиболее простых в реализации. Здесь значение терминологичности для словосочетания рассчитывается по формуле

$$C\_value(a) = \begin{cases} log_2|a| \cdot freq(a), \text{если } a \text{ не вложен в другие словосочетания,} \\ log_2|a| \cdot freq(a) - \frac{\sum_{b \in T_a} freq(b)}{P(T_a)}, \end{cases} \quad (1)$$

где

$a$ – кандидат в термины;

$|a|$ – длина словосочетания, измеряемая в количестве слов;

$freq(a)$ – частотность $a$;

$T_a$ – множество словосочетаний, которые содержат $a$;

$P(T_a)$ – количество словосочетаний, содержащих $a$.

Из (1) видно, что метод поощряет словосочетания, не входящие в состав других, более длинных словосочетаний.

Перед расчётом терминологичности документ разбивается на подстроки по признаку знаков препинания, деепричастий, глаголов. Строки между этими разделителями рассматриваются как кандидаты в термины для расчёта *C-value*. Кандидаты в термины выбираются по пороговому значению. В [7] указано, что данный метод не даёт высоких результатов чистоты терминов по экспертной оценке, но для нашей задачи получаемые результаты вполне подходят, так как мы не собираемся составлять словарь на основе выделенных терминоподобных конструкций, но использовать их исключительно для машинной обработки. Отметим, что данный метод использует относительно небольшое количество вычислительных ресурсов. Это важно, учитывая, что компьютеру приходится обрабатывать большое количество документов, во время ожидания пользователем ответа от навигационного сервера.

Для каждого документа рассчитывается оценочное множество *M*,состоящее из пар ($t$,$tfidf_t$), где $t$–термин, $tfidf_t$–TF-IDF[9]мера оценки важности терминоподобной конструкции в документе:

$$tfidf = \frac{|(d \ni a)|}{|d|} \cdot log \frac{|N|}{|(N \ni a)|},$$

где

$|d \ni a|$–количество включений термина в документ;

$|d|$– количество слов в документе (без слов-разделителей);

$|N|$ – количество документов в корпусе;

$|N \ni a|$ –количество документов, включающих $a$.

Релевантность документа $R$ рассчитывается по формуле

$$R(d) = \sum_{t \in M(d) \cap M(F)} tfidf_{t|F} \cdot tfidf_{t|d}.$$

Отметим, что данный подход к вычислению релевантности не позволяет найти документы с терминами, которые являющиеся синонимами, которые, однако, не эквивалентны использованным в обучающей выборке $F$.

Для решения данной проблемы следует оценить навигационный потенциал документа $n$:

$$n = R(d) \cdot sign(d) + Ref(d, 0),$$

$$Ref(d,l) = \begin{cases} R(d) \cdot sign(d_i), \text{если } l > 2, \\ \dfrac{\sum_{d_i \in V(d)} (R(d_i) \cdot sign(d_i) + Ref(d_i, l+1))}{|V(d)|} \end{cases},$$

где *sign(d)* –признак того что документ прочитан (0,5 –для прочитанного документа, 1 – для непрочитанного документа).

Таким образом, навигационный потенциал представляет собой сумму релевантности самого документа и релевантности дерева документов, на которые ссылается рассматриваемый документ, следовательно, навигационный алгоритм может привести пользователя к определённому важному документу. Ограничение на два уровня в глубину связано с сильной ветвистостью графа ссылок. Однако жёстко ограничивать глубину оценки навигационного потенциала в два перехода не имеет смысла, так как рассчитанное значение *Ref*(*d*,*l*) может существовать в кэше для меньшего значения *l*, т.е. поиск будет произведен с большей точностью.

Таким образом, если пользователь когда-то исследовал документ вблизи документа с высокой релевантностью, алгоритм и в дальнейшем будет пытаться вывести пользователя на важный документ, в т.ч. и теми путями в графе, длина которых превышает 2 перехода.

**Контекст навигации**

Контекст навигации содержит термины$F'$, расширяющие обучающее множество *F*. При этом для каждого состояния расширяющего множества $F'$ создаётся отдельный кэш. В отличие от основного кэша для $F' = \emptyset$ расширенные кэши необходимо проверять по прошествии определённого интервала времени на наличие возможных конфликтов.

Отметим, что расширяющий набор также может содержать стоп-слова, которые снижают релевантность документов. В более общем виде, расширяющий набор может содержать переопределение *tfidf* термина, в том числе на отрицательные и экстремально большие значения. Также в контекст может быть включена глубина оценки навигационного потенциала. Однако следует учитывать, что глубина менее 2 устраняет возможность навигации к важным документам по длинным путям.

Контекст может содержать предикативные ограничения на документы *N*такие, что алгоритм навигации будет работать только с подграфом сети. Так, например, можно ограничить документы по дате публикации, точнее изучать документы, дата создания которых не позднее определённой, или самые новые документы. Задержка в ответе контент-сервера также может оцениваться параметром контекста навигации, поскольку навигационный сервер должен направлять пользователя на наиболее быстро доступные ресурсы.

**Существующие проблемы и направления дальнейших исследований**

Опыт практической реализация описанного выше алгоритма навигации, однако, позволяет выявить ряд проблем и недостатков описанного выше алгоритма.

1) Алгоритм, как минимум, удваивает нагрузку на контент-сервер, но, фактически, фактор повышения нагрузки составляет несколько порядков. (Это является следствием основной идеи навигации – возложение на компьютер обязанностей анализа содержания и отклонения документов, не представляющих для пользователя. Отметим, что в этой ситуации важным становится использование разделяемых межпользовательских кэшей.)

2) Навигация становится серьёзной проблемой при генерации документов контент-сервером по сложному алгоритму, так как в случае работы одного навигационного сервера в интересах многих пользователей крупные контент-серверы

могут по ошибке или намерено внести его в чёрный список, прекратив тем самым возможность навигации.

3) На практике далеко не все контент-сервера следуют рекомендациям по организации URI документов, поэтому достаточно часто в адресе могут оказаться идентификаторы сессии работы пользователя. Как следствие, либо контент-сервер не сможет получить доступ к документам, либо работа с сервером будет потенциально небезопасной с точки зрения перехвата сессии. Таким образом, в любом случае будет нарушена работа межпользовательского fetch-кэша загруженных документов. Кроме того контент-сервер может генерировать сессионно-зависимые фрагменты документа, тогда, навигация по отдельным параграфам не будет осуществляется. Наконец, в самом наихудшем случае, GET-методы могут оказаться не нулипотентными, т.е. будут изменять состояние сервера. (В комбинации с непреднамеренным перехватом сессии, это может привести к тому, что пользователь случайно выберет ссылку «удалить файл», а навигационный сервер захочет узнать интересный ли документ будет в том случае, если ответить «да, я уверен, что хочу удалить файл».)

4) Наличие эффекта, известного в информационной навигации как «пузырь фильтров»[10], проявляющийся в том, что фокусировка получаемой информации на определённом круге вопросов создаёт исчерпывающее глубинное понимание узкой области, закрывая от исследователя новые идеи, а также «впечатление того, что наши узкие собственные интересы и есть всё, что существует и окружает нас» [11]. (Здесь, с одной стороны, можно возразить, что навигация, в отличие от ранжирования поисковой машины, не скрывает ни одной ссылки на документ. С другой стороны, при достаточно большой обучающей выборке $F$ и высокой кардинальности графа $N$, навигационный алгоритм проведёт нас тем путём, на котором окажутся только авторы документов, мыслящие сходно с пользователем.)

5) На данный момент нет объективного способа оценки качества работы алгоритма, поскольку не существует объективного критерия оптимизации. (Теоретически можно рассчитать корреляцию между количеством кликов и рейтингом ссылки, присвоенным алгоритмом навигации. Однако этот показатель отражает только доверие пользователя к используемому алгоритму, так как на момент клика мнение пользователя может основываться, исключительно, на названии и реквизитах цитируемого документа.)

Приведённый выше перечень проблем персональной навигации может быть условно разделен на технические, которые представляются вполне разрешимыми, и философские (список проблем, которых, по-видимому, не является окончательным).

Исследование методов оценки качества навигации является предметом дальнейших исследований.

**Выводы**

Проведенный анализ проблем навигации в мобильном представлении научной информации на основе статистической классификации позволил получить следующие результаты.

1. Обоснована целесообразность применения клиент-серверных облачных технологий при использовании мобильных устройств для навигации и представления научной информации.
2. Предложена структурная модель, описывающая взаимодействие браузера пользователя, контент-сервера и навигационного сервера.
3. Разработан алгоритм классификации документов по обучающей выборке с наложением контекста навигации и описан возможный состав контекста навигации.
5. Выявлены проблемы алгоритма навигации, основной из которых является проблема отсутствия объективных метрик качества навигации.

Работа выполнена при финансовой поддержке Министерства образования и науки РФ (ГК № 14.514.11.4037, шифр № 2012-1.4-07-514-0060-048).

**Литература**

1. Google X Labs: With Steve Jobs Gone, Could Google Take the Torch in Inventing the Future?[Электронныйресурс]//Режим доступа: http://www.deathandtaxesmag.com/160133/google-x-labs-with-steve-jobs-gone-could-google-take-the-torch-in-inventing-the-future/(дата доступа 2 октября 2012)
2.KonturLabs[Электронный ресурс] // Режим доступа: http://www.skbkontur.ru/press/news/company/2012/2/1412 //(дата доступа 2 октября 2012
3. Mell P. The NIST Definition of Cloud Computing / Mell, Peter and Grance, Timothy //Рекомендациинациональногоинститутастандартизациитехнологий, NIST Special Publication 800-145, Gaithersburg, октябрь 2011, –3 с.
4.RFCHTTPOverTLS[Электронный ресурс] // Режим доступа: http://tools.ietf.org/html/rfc2818 //(дата доступа 2 октября 2012)
5. RFCHypertextTransferProtocol–HTTP/1.1 [Электронный ресурс] // Режим доступа: http://tools.ietf.org/html/rfc2616 //(дата доступа 2 октября 2012)
6. Anderson J.R. Machine Learning: An Artificial Intelligence Approach / John Robert Anderson // Tioga Publishing Company, 1994. –572 с.
7. Браславский П., Соколов Е. Сравнение пяти методов извлечения терминов произвольной длины // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог» (Бекасово, 4–8 июня 2008 г.). Вып. 7 (14).– М.: РГГУ, 2008. – С. 67–74.
8. Frantzi K., Ananiadou S., Mima H. Automatic recognition of multi-word terms: the C-value/NC-value method // Int. J. Digit Libr.(2000) 3. –С. 115–130.
9.СолтонДж. Динамические библиотечно-поисковыесистемы / Солтон, Джон // М.: – Мир, 1979. –557с.
10.Pariser E. The Filter Bubble: What the Internet Is Hiding from You / Pariser, Eli // Penguin Press, New York, 2011, –С. 304
11. First Monday: What's on tap this month on TV and in movies and books: The Filter Bubble by Eli Paraiser / Газетная публикация // USA Today, 29 апреля 2011.