УДК 019.963.2:004.62

А. А. Князева [1], **О. С. Колобов** [2], **И. Ю. Турчановский** [1], **А. М. Федотов** [3]

[1] Отдел проблем информатизации ТНЦ СО РАН
пр. Академический, 10/4, Томск, 634021, Россия
E-mail: amili@lib.tpu.ru; tur@hcei.tsc.ru

[2] Институт сильноточной электроники СО РАН
пр. Академический, 2/3, Томск, 634055, Россия
E-mail: okolobov@hcei.tsc.ru

[3] Новосибирский государственный университет
ул. Пирогова, 2, Новосибирск, 630090, Россия
E-mail: fedotov@nsc.ru

РАНЖИРОВАННЫЙ ПОИСК В БИБЛИОГРАФИЧЕСКИХ БАЗАХ ДАННЫХ

В работе дается описание подхода к ранжированному поиску в библиографических базах данных, с помощью которого решается проблема эффективного тематического поиска в автоматизированных библиотечных каталогах. Библиографические записи рассматриваются в работе как структурированные короткие документы, которые могут состоять из нескольких зон. Это позволяет нам вычислять оценку документа по отношению к поисковому запросу с учетом различных зон документа. Полученные оценки документов применяются для ранжирования результатов поиска. Рассмотрены различные модели ранжированного поиска, а также подход к мета-поиску, для множества библиографических баз данных.

Ключевые слова: информационный поиск, базы данных, ранжирование, метаданные, мета-поиск.

Введение

Проблема поиска является вечной проблемой для построения любых информационных систем [2]. В настоящее время в автоматизированных библиотечных каталогах наиболее широко распространен классический подход к осуществлению поиска, основанный на булевой модели. Эффективный поиск на основе этой модели подразумевает, что пользователь, при необходимых знаниях и должном опыте, может сформулировать такой поисковый запрос, который точно определяет предмет поиска (например, поиск известного пользователю издания с известным заглавием или именем автора). В случае тематического поиска, если перед пользователем стоит задача найти неизвестное ему издание, применение булевой модели поиска не является эффективным. Это приводит к тому, что примерно половина запросов в случае тематического поиска в библиотечном каталоге дает «нулевой результат» [10]. Получается, что для тематического поиска требуется иной подход, который выходит за рамки булевой модели.

Библиотечные специалисты рассматривают тематический поиск как важный и полезный поиск для библиотечных каталогов и электронных библиотек. Для этого в библиотеках и сопутствующих им организациях выполняется работа по созданию лингвистического обеспечения: списков предметных рубрик, классификаторов, авторитетных файлов и тезаурусов. Эффективное использование результатов этой работы возможно при условии, что одно и то же лингвистическое обеспечение применяется как для создания библиографических записей, так и для их поиска. К сожалению, созданные в России автоматизированные библиотечные каталоги не имеют *единого* лингвистического обеспечения. В работе [1] выполнен анализ библиотечных каталогов и электронных библиотек, который показал, что имеются существенные недостатки отечественных систем, часть из которых мы приводим ниже.

1. Системы предметизации библиотечных каталогов основаны на собственных списках предметных рубрик, о которых пользователь часто не имеет представления, и более того, таких систем множество и они отличаются друг от друга. Самая мощная отечественная система предметизации действует в Российской национальной библиотеке, но она не является широко распространенной.

2. Поиск в библиотечном каталоге по кодам рубрикаторов и индексам (ББК, УДК и ДКД) не является популярным среди пользователей, так как их использование требует дополнительных знаний о той или иной системе рубрик или системе индексов. Кроме того, во многих каталогах поля с этими кодами не являются поисковыми.

3. Часто встречаются библиотечные каталоги, в которых не учитываются очевидные связи между записями, и нет возможности сортировать результат поиска по степени релевантности поисковому запросу.

Перечисленные недостатки показывают, что проблема не только в лингвистическом обеспечении, которое есть, пусть даже не единое, а в том, что пользователь выполняет поиск иначе, не так, как предполагают создатели библиотечных каталогов. Другими словами, пользователь предпочитает задавать поисковый запрос на естественном для него языке, в виде *свободного текста* и не стремится делать сложные логические запросы. Этот факт был отмечен при анализе поисковых запросов к автоматизированным библиотечными каталогам Томского научного центра СО РАН и Сибирского государственного медицинского университета.

В данной работе рассматриваются библиографические записи в качестве документов, и ставится задача расширения поисковых возможностей для библиотечных каталогов. Для этого подробно рассматривается ранжированный поиск и показывается, как можно получить новые, ранее не реализованные возможности для пользователя. Ключевыми моментами данной работы являются учет структуры библиографической записи, работа с запросами в свободной форме и предоставление результатов поиска в виде списка, сортированного в соответствии со степенью соответствия запросу.

В работе также показано, что применение ранжированного поиска для библиотечных каталогов открывает новые возможности для мета-поиска. Из-за ограниченного объема работы за пределами рассмотрения остается ряд вопросов, связанных со стеммированием, построением индекса документов и т. д.

Обзор

Булева модель информационного поиска [16] является первой и все еще самой распространенной моделью. Она основана на булевой логике и теории множеств: документы и запросы представляются в виде множеств терминов, документ выдается системой в том случае, когда в нем содержатся термины запроса. Основным ограничением классической булевой модели является то, что она не позволяет сортировать документы по степени их соответствия запросу, т. е. ранжировать результат поиска. В работе [3] была предложена модифицированная модель булевого поиска, получившая название расширенной. Она позволяет присваивать веса терминам (например, в соответствии со схемой $tf - idf$), что помогает при ранжировании набора документов в выдаче. Однако расширенная булева модель, как и классическая, основана на требовании формулировки запроса с применением логических операторов, а не на естественном языке, в виде свободного текста.

Векторная модель, в отличие от булевых, сразу создавалась для ранжированного поиска в условиях свободно сформулированных запросов [20]. Кроме того, она позволяет использовать единый подход к вопросам поиска, классификации и кластеризации документов. Документы и запросы в рамках этой модели представляются в виде векторов. Степень соответствия документа запросу определяется корреляцией между векторами, которую можно оценить, например, с помощью их скалярного произведения. Впервые данная модель была реализована Г. Селтоном в системе SMART (Salton‘s Magical Automatic Retriever of Text), разработанной в Корнелльском университете в 1960-е гг. [17].

В основе вероятностных моделей информационного поиска лежит вероятностный принцип ранжирования документов, впервые предложенный в работе [4], и бинарная модель независимости [5].

Одной из практических реализаций вероятностных методов является модель Okapi [6], получившая свое название по имени системы, в которой она была впервые реализована. Актуальная модификация модели носит название Okapi BM25 [9]. В работах [7; 8] приводится расширение этой модели на случай документов, разбитых на поля и зоны.

Применение ранжированного поиска для автоматизированных библиотечных каталогов дано в работе [10]. В этой работе дается описание проекта Cheshire II, который интересен новым подходом как к тематическому поиску в коллекции библиографических записей, так и к созданию клиентских интерфейсов. Проект Cheshire II был создан для тестирования вероятностных методов поиска для библиографических записей в формате MARC [1]. Для оценки вероятностей релевантности применялась логистическая регрессия, где в качестве независимых переменных выступали такие статистики коллекции, как средняя длина документов, частота появления термина в документе, инвертированная частота документа и т. п. В работах [11; 12] утверждается, что вероятностная модель поиска в комбинации с классификацией кластеризации работает лучше для тестовых записей MARC, чем булева модель или модель векторного пространства.

Слияние ранжированных результатов поиска на основе различных моделей мета-поиска исследуется в работе [13]. Интересны две модели borda-fuse и bayes-fuse. Модель borda-fuse основана на демократической стратегии выбора, bayes-fuse – на bayesian inferenece [13–15]. Обе модели отличаются от других существующих моделей мета-поиска тем, что при выполнении слияния применяется ранг документа, оценка релевантности которого не требуются. Описанные модели мета-поиска могут быть применимы для слияния результатов поиска из разных коллекций документов в системах, где оценка релевантности документа не предоставляются.

Модели поиска

Классическая булева модель. Булева модель поиска основана на точном совпадении терминов запроса и документа. Запрос может содержать логическое выражение, включающее булевы операторы И (AND), ИЛИ (OR) и И НЕ (NOT), что позволяет пользователю управлять полнотой и точностью поиска. При этом, как известно, применение оператора AND дает высокую точность и низкую полноту поисков, а применение оператора OR дает низкую точность и высокую полноту. И порой, применяя булеву модель поиска, трудно или даже невозможно найти необходимую «золотую середину». Например, рассмотрим автоматизированный библиотечный каталог, который применяет булеву модель поиска и содержит библиографическую запись на издание *«История Государства Российского / И. М. Карамзин. М.: Наука, 1989»*. Для поиска этого издания можно применить поисковый запрос вида *«T1TbE=“История Государства Российского” AND AUTH0R=Kapa.M3UH AND DATE=1989»*. Здесь кавычки указывают на то, что в качестве терма запроса используется фраза, а знак ’=’ указывает на точное совпадение. В этом запросе применяются поисковые атрибуты заглавие (TITLE), автор (AUTHOR) и дата публикации (DATE). Эти атрибуты соответствуют полям записи, причем с одним атрибутом могут быть связанны одно или несколько полей записи [2]. С таким запросом проблем не возникает. Соответствующая запись либо есть, либо ее нет в библиотечном каталоге, третьего варианта нет. Именно поэтому многие пользователи, в основном профессионалы, предпочитают булеву модель. В случае, когда запрос задан в форме свободного текста, например, *«История России Карамзин»* или кратко *«Карамзин История»*, вероятнее всего, пользователь получит «нулевой результат» или результат, в котором искомая запись есть, но располагается в результатах поиска случайным образом. Разумеется, результат работы булевой модели можно сортировать по опре-

[1] См.: MARC Standards – http://www.loc.gov/marc/

[2] Применение поисковых атрибутов упрощает работу пользователя, так как ему нет необходимости знать структуру записи и для поиска можно использовать распространенные категории, например: автор, название, дата публикации и др.

деленному критерию (заглавию, автору, году издания и др.), но для тематического поиска необходима, в первую очередь, возможность сортировки документов по степени соответствия запросу.

Ранжированный поиск. Первый шаг к организации ранжированного поиска заключается в том, чтобы учитывать, где именно встретился термин из запроса: в заголовке, описании, основном тексте документа и т. п. В общем случае поиска по полнотекстовым документам, наряду с самим текстом, выделяют и разбивают на части дополнительную структуру – метаданные. Под метаданными мы подразумеваем специальную форму данных о документе, в которой представляется, например, автор, заглавие и дата публикации. При библиографическом поиске запись библиотечного каталога представляет собой метаданные для издания, и ее можно рассматривать как совокупность раздельно индексированных составных частей. Можно рассмотреть два варианта таких составных частей: поля и зоны. Под полями будем понимать часть записи, которая может принимать лишь ограниченное количество значений (и для этих значений у нас есть специальный словарь); примером поля может служить год издания документа. Зона в нашем понимании представляет собой то же поле, только с неограниченным количеством значений (зонами, например, будут заглавие, имя автора, ключевые слова, предметные рубрики, аннотация и др.). При составлении запроса будем учитывать, где именно хотим видеть искомые термины. Обработка запроса состоит в пересечении нескольких списков отобранных документов, которые берутся из обычного инвертированного индекса, а также из параметризированного индекса. Имеется только один параметризированный индекс для каждого поля (например, дата публикации), и он позволяет нам выбрать только те документы, которые совпадают по указанным в запросе данным. В общем случае некоторые поля могут содержать упорядоченные данные, которые позволяют применять отношения вида – больше, меньше, а также задавать интересующий нас интервал.

В приведенном примере (см. «Классическая булева модель») словарь для параметрического индекса DATE создается из фиксированного списка значений (год публикации), а для каждой из зон документа TITLE, AUTHOR можно создать соответствующий инвертированный индекс. Словарь для индекса зоны должен иметь структуру, которая позволит нам хранить список терминов из текста зоны. Такой подход увеличивает размер словаря, поскольку необходимо хранить данные о том, в какой зоне встречается тот или иной термин документа. Следующим логическим шагом будет вычисление оценки документа, которая называется в литературе *взвешенным рейтингом зоны* [16].

Рабочий пример. Мы будем рассматривать записи библиотечного каталога. Отдельная такая запись представляется нам как короткий структурированный документ, который содержит поля данных со свободным текстом (например, поля: заглавие, аннотация, ключевые слова, предметные рубрики и др.). В качестве рабочего примера будем рассматривать коллекцию, состоящую из 100 документов по медицине. Приведем фрагмент таблицы, содержащей некоторые характеристики, которые можно получить до начала обработки запроса (табл. 1).

Наряду с количеством документов, в которых встречается тот или иной термин, можно рассчитать инверсную частоту *idf*, позволяющую приписать более редкому в коллекции термину больший вес. В табл. 1 приведены два варианта расчета инверсной частоты документов, содержащих термин. Также допустим, что запрос состоит из двух слов: «средства профилактики». После предварительного стеммирования получим следующий вид запроса: «профилакт* средств*». Как видно из табл. 1, термин «профилакт*» встречается в 9 документах, а термин «средств*» – в 6 документах.

Взвешенный рейтинг зон. Пусть дан булев запрос q и документ d, тогда взвешенный рейтинг зоны назначает паре (q, d) оценку в интервале [0, 1], который вычисляется как линейная комбинация оценок зон, где каждая зона документа добавляется исходя из булевого запроса. Рассмотрим множество документов, каждый из которых имеет l зон. Пусть $g_1, ..., g_l \in [0, 1]$, так что

$$\sum_{i=1}^{l} g_i = 1.$$

Для $1 < i < l$, пусть s_i есть булева оценка, которая обозначает совпадение (или несовпадение) между q и г-той зоной. Например, булева оценка для зоны должна быть равна 1, если все термины запроса встречаются в зоне, и 0 в противном случае. Тогда взвешенный рейтинг зоны определяется так

$$\sum_{i=1}^{l} g_i s_i .$$

Взвешенный рейтинг зоны иногда называют ранжированным булевым поиском. Это простой метод оценки документов для приблизительного ранжирования результатов булевого поиска. Он учитывает вес зон документа, но не учитывает *важность* термина из запроса в документе и *важность* термина во всей коллекции документов.

Таблица 1

Характеристики распространенности терминов
(N – количество документов в коллекции)

Термин	Число документов df_t	Варианты инверсной частоты idf	
		$\log\frac{N}{df_t}$	$\log\frac{N-df_t+0{,}5}{df_t+0{,}5}$
антиаритмическ*	8	1,1	1,04
вторичн*	15	0,82	0,74
инсульт*	10	1	0,94
применен*	24	0,62	0,49
профилакт*	9	1,05	0,98
свойств*	13	0,89	0,81
средств*	6	1,22	1,16
терапевтическ*	14	0,85	0,78
фармакологическ*	22	0,66	0,54

Вернемся к рабочему примеру. Предположим, что все документы коллекции можно разбить на 3 зоны: заглавие (TITLE), ключевые слова (KEYWORDS) и основной текст (BODY). Для каждой из этих зон можно построить свой инвертированный файл и подсчитать частоту документов, в которых искомый термин встречается в определенной зоне. Прежде всего, необходимо задать веса для каждой из трех зон, как это показано в табл. 2. Причем, согласно требованиям ранжированной булевой модели, сумма весов зон должна равняться единице.

Таблица 2

Веса для зон документа

Зона документа	Соответствующий вес
TITLE	0,5
KEYWORDS	0,3
BODY	0,2

Заметим, что пользователь, формулируя запрос на естественном языке, не указал булев оператор (AND или OR). Однако для работы в рамках булевой модели необходимо задать такой оператор. Рассмотрим ранжированный поиск для оператора AND, это означает, что в выдачу попадут только те документы, которые содержат оба термина запроса (всего 4 доку-

мента). В табл. 3 представлены индикаторные величины, указывающие на присутствие обоих терминов запроса в зонах документа и результирующая оценка документа.

Таблица 3

Индикаторы появления обоих терминов запроса по зонам

DocID	TITLE	KEY	BODY	оценка
2	0	0	1	0,2
3	0	1	1	0,5
5	0	0	1	0,2
15	0	0	1	0,2

Таким образом, на первое место в выдаче попадет документ с идентификатором 3, а документы 2, 5 и 15 последуют за ним в произвольном порядке.

Модель векторного пространства. Поскольку любой документ можно рассматривать как множество терминов, коллекцию документов можно представить как множество векторов в векторном пространстве, где каждая ось соответствует одному уникальному термину. При таком подходе теряется относительный порядок терминов в документе, так что документы *«TeX лучше, чем Word»* и *«Word лучше, чем TeX»* являются идентичными, и об этом необходимо помнить, когда утверждается, что документы подобны.

Пусть $\vec{V}(d)$ обозначают вектор документа d, с компонентами ($v_1(d)$, $v_2(d)$, ..., $v_M(d)$). Для оценки подобия между двумя документами d_1 и d_2 вычисляется косинус подобия соответствующих векторов $\vec{V}(d_1)$ и $\vec{V}(d_2)$

$$sim(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{\left|\vec{V}(d_1)\right| \left|\vec{V}(d_2)\right|}, \qquad (1)$$

где в числителе представлено скалярное произведение двух векторов $\vec{V}(d_1)$ и $\vec{V}(d_2)$, а в знаменателе – произведение их евклидовых длин. Выполнив нормализацию векторов $\vec{V}(d_2)$ и $\vec{V}(d_2)$ к единичным векторам $\vec{v}(d_1)$ и $\vec{v}(d_2)$, выражение (1) можно записать так:

$$sim(d_1, d_2) = \vec{v}(d_1) \cdot \vec{v}(d_2).$$

Скалярное произведение двух нормализованных по длине векторов представляет собой косинус угла между этими векторами, т. е. оценка подобия двух документов сводится к вычислению косинуса угла между соответствующими векторами. Для любого документа d из коллекции документов d_1, d_2, …, d_N можно получить числовую оценку подобия с другими документами, и на основе этого выбрать наиболее близкие к документу d документы в коллекции. Результатом такой операции может быть, например, предоставление сервиса с известным названием *похожие документы*.

Ключевым моментом этой модели поиска является идея представления запроса в виде вектора в том же векторном пространстве, которое используется для документов.

Пусть запрос q представляется в виде единичного вектора $\vec{v}(q)$, тогда каждому документу d можно назначить скалярное произведение $\vec{v}(q) \cdot \vec{v}(d)$, которое является оценкой документа для запроса. Итак, имеем

$$sim(q, d) = \vec{v}(q) \cdot \vec{v}(d).$$

Документ может иметь наивысшую оценку, даже если не все термины запроса содержатся в документе. При этом документ имеет нулевую оценку, в случае если вектор запроса ортогонален вектору документа. Такая ситуация возникает, если в документе нет терминов из запроса, т. е. такой документ не должен включаться в выдачу. Таким образом, полученные оценки для каждого из документов в коллекции могут быть использованы для их сортировки по степени соответствия запросу, т. е. для ранжирования.

Для взвешивания документа могут применяться различные схемы, но в данной статье ограничимся только схемами взвешивания, которые основаны на статистических методах расчета.

Взвешивание запроса и документа. Ранее мы определили, что оценка документа *d* для запроса *q* есть скалярное произведение векторов, которое можно представить в виде суммы произведений соответствующих компонентов векторов $\vec{v}(d)$ и $\vec{v}(q)$

$$\sum_{i=1}^{M} \vec{v}_i(q) \cdot \vec{v}_i(d),$$

где *M* – это размерность векторного пространства. Численное значение компонент того или иного вектора определяется взвешиванием.

Для взвешивания вектора будем применять классическую схему *tf – idf*. Эта схема определяется как произведение частоты встречаемости термина в документе *tfu* и инвертированной частоты документа, в котором встречается термин запроса

$$idf_t = \log \frac{N}{df_t},$$

где *N* – общее количество документов в коллекции

$$tf - idf_{td} = tf_{tq} idf_t.$$

Итак, $tf - idf_{td}$ назначает вес для термина *t* в документе *d*, так что:

1) наибольший вес имеет термин, который встречается много раз в небольшом количестве документов;

2) меньший вес имеет термин, который не часто встречается в документе или встречается во многих документах;

3) наименьший вес имеет термин, который встречается во всех документах.

Для получения оценки документа нам необходимо взвешивать компоненты векторов запроса и каждого из документов коллекции. Видно, что прямолинейное использование этого подхода является затратным по количеству необходимых операций на запрос. Поэтому на практике применяют различные варианты для взвешивания документа и запроса. Это позволяет избежать большого количества вычислений на запрос, путем подготовки необходимых оценок на этапе индексирования документов. В работе [16] дан обзор различных вариантов взвешивания векторов запроса и документа.

Вернемся к примеру. Взвешивание документов примера будем проводить по схеме nnc.ntn [16], где первый триплет отвечает за схему взвешивания вектора документа, а второй – вектора запроса.

Начнем со взвешивания запроса, поскольку этот сомножитель будет общим для всех документов. Частота обоих терминов в запросе равна единице, частота документа для терминов и соответствующая инверсная частота вычислены в табл. 4, в последней колонке записаны нормализованные веса терминов запроса.

Таблица 4

Взвешивание запроса

Термин	tf	df	idf	wq
профилакт*	1	9	1,05	1,05
средств*	1	6	1,22	1,22

Заметим, что в качестве частоты термина в документе была использована скорректированная с учетом весов зон частота

$$tf_{td}^{'} \sum_{f=1}^{F} W_f tf_{tdf},$$

где веса зон W_f взяты из предыдущего примера (см. табл. 2). Заметим, что в отличие от ранжированной булевой модели модель векторного пространства не требует задания весов, со-

ставляющих в сумме единицу. Легко убедиться, что применение евклидовой нормализации к весам снимает такое требование. В последней колонке указана общая оценка вклада каждого из терминов для документа 3, полученная произведением оценки запроса и оценки документа для соответствующего термина. Общая оценка документа 3 в соответствии с векторной моделью составляет 0,18 + 1,20 = 1,38 (табл. 5). Аналогично производится расчет и для остальных документов, содержащих хотя бы один из терминов запроса. Результат ранжирования представлен в табл. 6.

Таблица 5

Взвешивание документа 3

Термин	tf	wf	wd	Оценка
профилакт*	5,3	5,3	0,17	0,18
средств*	31,1	31,1	0,99	1,20

Таблица 6

Результат работы векторной модели

Документ	Оценка	Место в выдаче
1	1,05	9
2	1,60	2
3	1,38	3
5	1,61	1
15	1,26	4
17	1,22	6
18	1,05	10
45	1,05	11
50	1,05	7
56	1,22	5
98	1,05	8

Заметим, что документы 17 и 56 имеют одинаковую оценку 1,22 балла, а оценку в 1,05 балла получили сразу 5 документов. При распределении мест такие документы были перечислены в произвольном порядке.

Вероятностные модели поиска. Все вероятностные модели основаны на оценке вероятности того, что документ релевантен по отношению к запросу пользователя. При этом можно рассматривать индикаторную величину $R_{d,q}$, равную 1, если данный документ d является релевантным по отношению к данному запросу q, и 0 в противном случае. Вероятность релевантности документа d относительно запроса q можно записать как $R_{d,q} = 1$.

В основе рассматриваемых моделей поиска лежит бинарная модель независимости, которая базируется на следующих предположениях:

1) термины встречаются в документе независимо друг от друга;

2) достаточно отметить факт наличия термина в документе, не учитывая количества его повторений.

Естественно, эти предположения могут быть далеки от реальности, однако, как правило, они дают приемлемые результаты при практическом применении.

Итак, документы можно представить в виде векторов инцидентности $x = (x_1, x_2, \ldots, x_M)$, где $x_t = 1$, если термин t представлен в документе d, и 0 в противном случае. При этом одно и то же векторное представление может соответствовать сразу нескольким документам. Аналогично запрос пользователя представляется в виде вектора q. Задача сводится к оценке того, как термины, содержащиеся в документе, влияют на его релевантность, в особенности интересно влияние таких характеристик, как частота появления термина в документе, частота до-

кумента (количество документов в коллекции, содержащих данный термин), длина документа и другие показатели, которые можно рассчитать.

Решение о релевантности документа принимается в случае, если вероятность того, что он релевантен, больше вероятности того, что он нерелевантен относительно данного запроса: $P(R = 1|d, q) > P(R = 0|d, q)$.

Используя байесовское правило для условных вероятностей, можно записать

$$P(R=1 \mid \bar{x}, \bar{q}) = \frac{P(\bar{x} \mid R=1, \bar{q}) P(R=1 \mid \bar{q})}{P(\bar{x} \mid \bar{q})},$$

и соответственно

$$P(R=0 \mid \bar{x}, \bar{q}) = \frac{P(\bar{x} \mid R=0, \bar{q}) P(R=0 \mid \bar{q})}{P(\bar{x} \mid \bar{q})}.$$

Для того чтобы уменьшить количество неизвестных величин, подлежащих оцениванию, рассмотрим следующее отношение шансов, отражающее релевантность документа:

$$O(R \mid \bar{x}, \bar{q}) = \frac{P(R=1 \mid \bar{x}, \bar{q})}{P(R=0 \mid \bar{x}, \bar{q})} = O(R, \bar{q}) \prod_{t=1}^{M} \frac{P(x_t \mid R=1, \bar{q})}{P(x_t \mid R=0, \bar{q})}.$$

Заметим, что $O(R, q)$ – величина, постоянная для конкретного запроса, и, следовательно, при решении задачи ранжирования документов ее можно опускать. Кроме того, можно упростить задачу, рассматривая не саму величину $O(R, q)$, а ее логарифм. Тогда, учитывая, что величины x_t могут принимать значение 0 либо 1, и обозначив вероятность присутствия термина t в релевантном документе $P(xt = 1|R = 1, q)$ через p_t, а вероятность присутствия термина в нерелевантном документе $P(xt = 1| R = 0, q)$ через u_t, сможем записать выражение для логарифма вклада отдельного термина в оценку значимости документа:

$$c_t = \log \frac{p_t}{1-p_t} + \log \frac{1-u_t}{u_t}. \qquad (2)$$

На практике, в случае достаточно большой коллекции документов и небольшого процента среди них релевантных данному запросу, для определения второго слагаемого в выражении (2) можно использовать соотношение

$$\log \frac{1-u_t}{u_t} \approx \log \frac{N}{df_t}.$$

Что касается первого слагаемого, отвечающего за релевантные документы коллекции, то для его оценки существуют различные способы.

1. Можно оценить частоту появления термина в документах, релевантность которых нам известна (релевантная обратная связь).

2. Можно использовать константу, предполагая, например, что вероятность встретиться в релевантных документах для всех терминов запроса одинакова и равна 0,5. Однако такой подход может быть слишком далек от реальности.

Кроме того, возможно итеративное построение оценок вероятности p_t на основе обратной связи.

Okapi BM25. Одной из распространенных реализаций вероятностного подхода является схема взвешивания BM25, часто называемая также схемой *Okapi* – по названию поисковой системы, в которой она впервые была реализована. Можно сказать, что эта модель находится на стыке вероятностных идей и принципов векторной модели: она сочетает в себе оценку вероятности релевантности документа и внимание к таким характеристикам коллекции, как частота появления терминов и длина документа.

Основное уравнение для ранжирования документов в этой модели имеет вид

$$RSV_d = \log\left[\frac{N}{df_t}\right] \cdot \frac{(k_1+1)tf_{td}}{k_1\left((1-b)+b\left(dl/avdl\right)\right)+tf_{td}} \cdot \frac{(k_3+1)tf_{tq}}{k_3+tf_{tq}},$$

где первый сомножитель отвечает за ранжирование по частоте документа в коллекции, второй – по частоте термина в документе, а третий – по частоте термина в запросе. В качестве первого сомножителя наряду с *idf* могут быть использованы и другие схемы взвешивания

документа. Например, модифицированная схема *idf*, исключающая возможность обращения знаменателя в ноль:

$$\log\frac{1-u_t}{u_t} \approx \log\frac{N-df_t+0{,}5}{df_t+0{,}5}.$$

Принятые обозначения:

N – количество документов в коллекции;

df_t – количество документов, содержащих термин t в коллекции;

tf_{td} – частота термина t в документе d;

tf_{tq} – частота термина t в запросе q;

dl – длина документа d;

$avdl$ – средняя длина документов коллекции;

k_1, k_3, b – параметры модели.

Коэффициенты k_1, k_3, b могут быть выбраны с помощью тестовой выборки. Как показано в [6], на основе экспериментов с коллекциями TREC рекомендуемые величины коэффициентов: $1, 2 < k_1, k_3 < 2$ и $b = 0{,}75$. Параметр k_3, характеризующий зависимость от частоты термина в запросе, актуален только в случае длинных поисковых запросов. Для коротких запросов можно принять $k_3 = 0$.

Все величины, входящие в уравнение модели, включая коэффициенты, могут быть найдены до обработки запроса и сохранены в системе, что позволит экономить время выполнения поиска.

Okapi BM25 для структурированных документов. Как правило, текстовые документы можно разбить на несколько зон, например, выделить заглавие, аннотацию, ключевые слова, собственно текст документа и т. п. В этом случае можно воспользоваться моделью Okapi BM25F, описанной в работах [7; 8].

В предположении, что система, в основном, предназначена для обработки не слишком длинных запросов примем $k_3 = 0$. Основное уравнение для ранжирования документов в этом случае принимает следующий вид

$$RSV_d = \sum_{t=1}^{M} \log\left[\frac{N}{df_t}\right] \cdot \frac{\left(k_1^{'}+1\right)tf_{td}^{'}}{k_1^{'}\left(\left(1-b\right)+b\left(dl^{'}/avdl^{'}\right)\right)+tf_{td}^{'}}.$$

Главное отличие от простой схемы Okapi заключается в подсчете частоты термина в документе. Допустим, каждый документ коллекции состоит из F зон, тогда скорректированная частота термина примет вид

$$tf_{td}^{'} = \sum_{f=1}^{F} W_f tf_{tdf},$$

где tf_{tdf} – количество появлений термина t в зоне f документа d. Такой взвешенный учет появлений термина позволяет придать большее значение его присутствию, например, в заголовке документа. Аналогично взвешенный подход приведет к новым значениям длины документа и средней длины по всей коллекции

$$dl^{'} = \sum_{j=1}^{M} tf_{td}^{'},$$

$$avdl^{'} = \frac{1}{N}\sum_{i=1}^{N} dl_i^{'}.$$

Для применения взвешенной схемы необходимо скорректировать коэффициент k_1, новый коэффициент рассчитывается на основе старого:

$$k_1^{'} = k_1 \frac{avdl^{'}}{avdl}.$$

Таким образом можно избежать повторного его подбора на основе тестовой коллекции.

Возвращаемся к примеру. Согласно схеме взвешивания Okapi BM25F, применяемой для структурированных документов, для каждого термина запроса необходимо рассчитать его вклад в оценку документа, а затем просто суммировать эти вклады для получения общей

оценки документа относительно запроса. Как и в предыдущих примерах, необходимо задать веса для зон документа. Возьмем те же веса, что и ранее (см. табл. 2).

Рассчитаем обычную и взвешенную частоты появления терминов «профилакт*» и «средств*» в документах коллекции (табл. 7–10).

Вообще говоря, в отличие от булевого ранжированного поиска не требуется, чтобы веса в сумме давали единицу, но, как и в векторной модели, необходимо провести корректировку. В данной модели она касается длины документа, средней длины документов коллекции и коэффициента k_1.

Средняя длина документов в коллекции 4 943, средняя взвешенная длина – 990 слов.

Схема взвешивания Okapi BM25F предполагает задание двух параметров, в данном примере использовались следующие значения: b = 0,75, k_1 = 1,2. Скорректированный параметр $k_1^{'}$ = 1,2 · (990/4 943) = 0,24.

Таблица 7

Расчет частоты появления для термина «профилакт*»

Номер документа	Количество появлений термина по зонам			Частота термина tf_t	Взвешенная частота tfl
	TITLE tf_{t1}	KEYWORDS tf_{t2}	BODY tf_{t3}		
1	1	1	50	52	10,8
2	0	0	30	30	6
3	0	1	25	26	5,3
5	1	1	150	152	30,8
15	0	0	10	10	2
18	1	1	50	52	10,8
45	2	2	200	204	41,6
50	0	2	19	21	4,4
98	1	1	43	45	9,4

Таблица 8

Расчет частоты появления для термина «средств*»

Номер документа	Количество появлений термина по зонам			Частота термина tf_t	Взвешенная частота tfl
	TITLE tf_{t1}	KEYWORDS tf_{t2}	BODY tf_{t3}		
2	0	0	30	30	6
3	1	2	150	153	31,1
5	0	0	170	170	34
15	2	1	300	303	61,3
17	0	1	70	71	14,3
56	0	0	80	80	16

Таблица 9

Расчет взвешенной длины документов

Номер документа	Количество слов в документе по зонам			Длина документа *dl*	Взвеш. длина *dl'*	Отношение к средней длине *dl' /avdl'*
	TITLE dl_1	KEY-WORDS dl_2	BODY dl_3			
1	3	3	239	245	50,2	0,05
2	2	7	2857	2866	574,5	0,58
3	5	7	4108	4120	826,2	0,83
5	1	9	8882	8892	1779,6	1,80
15	5	5	3630	3640	730	0,74
17	5	9	3021	3035	609,4	0,62
18	3	7	4281	4291	859,8	0,87
45	2	6	6739	6747	1350,6	1,36
50	3	11	1073	1087	219,4	0,22
56	2	7	282	291	59,5	0,06
98	3	3	7191	7197	1440,6	1,45

Таблица 10

Результаты работы модели Okapi BM25F

Документ	Оценка документов			Место в выдаче
	профилакт*	средств*	общая	
1	1,21	0	2,151	5
2	1,19	1,19	4,215	3
3	1,17	1,21	4,232	2
5	1,21	1,21	4,279	1
15	1,11	1,22	4,133	4
17	0	1,2	2,133	7
18	1,2	0	2,122	9
45	1,21	0	2,149	6
50	1,19	0	2,116	10
56	0	1,2	2,124	8
98	1,18	0	2,091	11

Результаты на основе трех моделей поиска. В табл. 11 приведены результаты работы трех рассмотренных моделей информационного поиска на основе рабочего примера. Поскольку пример не включает сведений о релевантности документов по отношению к запросу «средства профилактики», не представляется возможным полноценное сравнение рассматриваемых методов поиска.

Мета-поиск. На практике часто встречается задача одновременного поиска в двух системах с последующим объединением результатов поиска, или иначе – задача мета-поиска.

В связи с этим возникает вопрос о том, как рассмотренные модели поиска работают для мета-поиска?

Таблица 11

Результаты поиска различными методами

Документ	Место в выдаче		
	Взвешенный рейтинг зон	Векторная модель	Вероятностная модель
1	–	9	5
2	2	2	3
3	1	3	2
5	3	1	1
15	4	4	4
17	–	5	7
18	–	10	9
45	–	11	6
50	–	7	10
56	–	6	8
98	–	8	11

К потенциальным преимуществам мета-поиска, как это отмечено в работе [13], можно отнести следующее.

1. Улучшение коэффициента полноты поиска. Коэффициент полноты поиска есть отношение извлеченных релевантных документов к общему количеству релевантных документов. Применение мета-поиска может улучшать полноту поиска до тех пор, пока извлекаются различные, по релевантности, документы [19].

2. Улучшение точности поиска. Точность поиска есть отношение извлеченных релевантных документов к извлеченным документам [19].

3. Устойчивость результатов поиска. Устойчивость результатов поиска при мета-поиске может быть выше, чем при поиске в отдельно взятой поисковой машине [18].

Результат булевого поиска в двух системах может быть объединен на основе простого чередования документов, который известен в литературе как round-robin метод. Полученный объединенный результат поиска может быть сортирован по полям или зонам документа, при условии, что предварительно результаты поиска в каждой из систем уже сортированы по заранее выбранному критерию. В случае если результат поиска в каждой из систем уже сортирован по рангу (ранжирован), то для получения объединенного ранжированного результата поиска нет необходимости предварительно извлекать документы, как в случае булевого поиска. В литературе процесс объединения результатов поиска в один результат поиска для выдачи пользователю называется слиянием (fuse). Для слияния могут применяться различные алгоритмы, которые называют алгоритмами мета-поиска.

Наибольший интерес представляют алгоритмы, не требующие оценок документов и ориентированные на то, что каждая из систем может предоставить документы в порядке убывания их оценок. В данной работе мы будем рассматривать только один алгоритм мета-поиска borda-fuse, поскольку для него не требуется обучение на основе тестовой коллекции документов с известной релевантностью.

Borda-fuse – это простой, быстрый и эффективный алгоритм слияния сортированных по рангу списков документов. Этот алгоритм основан на известном методе подсчета для голосования «подсчет Борда», отличающимся от большинства аналогичных процедур применимостью для случая небольшого количества избирателей и большого количества кандидатов [13]. Эта особенность делает его наиболее подходящим к задаче мета-поиска. В качестве кандидатов выступают искомые документы, а в качестве избирателей – поисковые машины.

Подсчет Борда работает следующим образом: каждый избиратель ранжирует фиксированный набор из *c* кандидатов в порядке предпочтения. Для каждого избирателя наиболее предпочтительный кандидат получает *c* очков, следующий кандидат – *c* – 1 очков и т. д. Если некоторые кандидаты остаются не ранжированными избирателем, то остаток очков распределяется поровну между ними. Далее кандидаты ранжируются в порядке полученных в сумме очков от всех избирателей, и кандидат с наибольшим количеством баллов выигрывает выборы.

Рассмотрим пример подсчета Борда для выборки из 7 документов и трех поисковых машин (табл. 12). Первая машина ранжировала все семь документов, вторая не оценивала документы *d*, *f* и *g*, поэтому они разделили между собой оставшиеся 6 очков. Аналогично в колонке третьей поисковой машины документы *c* и *d* разделили поровну 3 очка. Расположив документы в порядке убывания суммы очков, присвоенных им поисковыми машинами, получим ранжированный список этих документов.

Таблица 12

Слияние по методу borda-fuse

Документ	Поисковые машины			Сумма очков	Результат слияния
	ПМ1	ПМ2	ПМ3		
a	7	7	4	18	I
b	6	5	3	14	III
c	5	4	1,5	10,5	IV
d	4	2	1,5	7,5	VI
e	3	6	7	16	II
f	2	2	5	9	V
g	1	2	6	9	V

Таким образом, для применения подсчета Борда к слиянию ранжированных списков не требуется знать оценки релевантности, вычисленные отдельными системами для конкретного документа. Кроме того, в отличие от алгоритма Байесовского слияния (bayes-fuse), не требуется обучения на основе тестовой выборки.

В случае, если в отличие от демократических выборов необходимо неодинаково взвешивать «голос» различных поисковых машин, можно применять взвешенный алгоритм подсчета Борда. Однако для определения оптимальных весов, соответствующих каждой из поисковых машин, потребуется обучение на основе тестовой коллекции документов.

Заключение

В работе показана принципиальная возможность применения ранжированного поиска для библиографических баз данных на основе векторной и вероятностной моделей поиска. Это открывает возможность для реализации тематического поиска в библиографических базах данных на основе поисковых запросов, заданных на естественном языке. Кроме того, применение ранжированного поиска дает возможность выполнять мета-поиск для двух и более различных библиографических баз данных. Каждая из рассмотренных моделей поиска может применяться наряду с классической булевой моделью.

Для демонстрации работы этих моделей в работе использована тестовая коллекция документов, для которой получены оценки документов по отношению к конкретному поисковому запросу. Для векторной модели применялось взвешивание по схеме *tf – idf*, а для вероятностной модели – схема OkapiBM25 для структурированных документов. Полученный результат показывает только, как выполнять ранжирование документов на основе получаемых оценок

документов. Для полноценного сравнения этих моделей требуется вычислительный эксперимент на основе тестовых коллекций документов с экспертной оценкой их релевантности.

Список литературы

1. *Лавренова О. А.* Тематический поиск в электронных каталогах и электронных библиотеках // Библиотеки и ассоциации в меняющемся мире: новые технологии и новые формы сотрудничества: Тр. конф. М., 2004.

2. *Федотов А. М., Барахнин В. Б.* Проблемы поиска информации: история и технологии // Вестн. Новосиб. гос. ун-та. Серия: Информационные технологии. 2009. Т. 7, вып. 2. С. 3–17.

3. *Salton G., Fox E. A., Wu H.* Extended Boolean Information Retrieval // Commun. ACM. November 1983. Vol. 26 (11). P. 1022–1036.

4. *Robertson S. E.* The Probability Ranking Principle // Journal of Documentation. 1977. Vol. 33. P. 294–304.

5. *Rijsbergen C. J. van.* Information Retrieval. 2nd ed. Dept. of Computer Science, University of Glasgow, 1979.

6. *Sparck J. K., Walker S., Robertson S. E.* A Probabilistic Model of Information Retrieval: Development and Comparative Experiments // Information Processing and Management. 2000. Vol. 36. Part 1. P. 779–808; Part 2. P. 809–840.

7. *Robertson S., Zaragoza H., Taylor M.* Simple BM25 Extension to Multiple Weighted Fields // CIKM'04. 2004. P. 42–49.

8. *Lu W., Robertson S., MacFarlane A.* Field-Weighted XML Retrieval Based on BM25 // INEX. 2005. P. 161–171.

9. *Robertson S. E., Walker S., Hancock-Beaulieu M., Gatford M., Payne A.* Okapi at TREC-4 // Proceedings of the 4th Text REtrieval Conference (TREC-4). 1995.

10. *Larson R. R., McDonough J., O'Leary P., Kuntz L., Moon R.* Cheshire II: Designing a Next-Generation Online Catalog // J. Am. Soc. Inf. Sci. Jul. 1996. Vol. 47 (7). P. 555–567. URL: http://dx.doi.org/10.1002/(SICI)1097-4571(199607)47:7<555::AID-ASI7>3.0.CO;2-T

11. *Larson R. R.* Classification Clustering, Probabilistic Information Retrieval, and Online Catalog // Library Quarterly. Vol. 61. P. 133–173.

12. *Larson R. R.* Evaluation of Advanced Retrieval Techniques in an Experimental Online Catalog // Journal of the American Society for Information science. Vol. 43. P. 34–53.

13. *Aslam J. A., Montague M.* Models for Metasearch // Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in information Retrieval (New Orleans, Louisiana, United States). SIGIR ´01. ACM. N. Y., NY, 2001. P. 276–284. URL: http://doi.acm.org/10.1145/383952.384007

14. *Thompson P.* A Combination of Expert Opinion Approach to Probabilistic Information Retrieval, Part 1: The Conceptual Model // Information Processing and Management. 1990. Vol. 26 (3). P. 371–382.

15. *Thompson P.* A Combination of Expert Opinion Approach to Probabilistic Information Retrieval. Part 2: Mathematical Treatment of CEO Model 3 // Information Processing and Management. 1990. Vol. 26 (3). P. 383–394.

16. *Christopher D.* Manning, Prabhakar Raghavan, and Hinrich Schutze. Introduction to Information Retrieval. Cambridge University Press, 2008.

17. *Сэлтон Г.* Автоматическая обработка, хранение и поиск информации. М.: Сов. радио, 1973. 560 с.: ил., табл.

18. *Selberg E., Etzioni O.* On the Instability of Web Search Engines // Proceeding of RIAO. 2000.

19. *Ng K. B., Kantor P. B.* An Investigation of the Preconditions for Effective Data Fussion in Ir: A Pilot Study // Proceedings of the 61th Annual Meeting of the American Society for Information Science, 1998.

20. *Salton G., Wong A., Yang C.S.* A Vactor Space Model for Automatic Indexing // Communications of the ACM. 1975. Vol. 18. No. 11. P. 613–620.

Материал поступил в редколлегию 14.09.2009

A. A. Knyazeva, O. S. Kolobov, I. Yu. Turchanovskij, A. M. Fedotov

RANKED SEARCH IN BIBLIOGRAPHIC DATABASES

The paper describes the approach to the ranked search in bibliographic databases. The problem of effective subject search in automated library catalogs is solved with this approach. Bibliographic record is considered as structured document, which may consist of a several zones. This allows us to calculate the assessment in relation to the search query according to different zones of the document. The obtained estimates documents are used for ranking search results. There are considered various models of ranked search, as well as the approach to meta-search, for a variety of bibliographic databases.

Keywords: information retrieval, databases, ranking, metadata, meta-search.