Методы информационного поиска в компьютерных сетях ссверхнасыщенными информационными ресурсами

Book · January 2004

CITATIONS
0

READS
141

1 author:

Vagif Gasimov
Azerbaijan Technical University
42 PUBLICATIONS **9** CITATIONS

SEE PROFILE

В.А.КАСУМОВ

МЕТОДЫ ИНФОРМАЦИОННОГО ПОИСКА В КОМПЬЮТЕРНЫХ СЕТЯХ С СВЕРХНАСЫШЕННЫМИ ИНФОРМАЦИОННЫМИ РЕСУРСАМИ

В.А.Касумов

МЕТОДЫ ИНФОРМАЦИОННОГО ПОИСКА В КОМПЬЮТЕРНЫХ СЕТЯХ С СВЕРХНАСЫШЕННЫМИ ИНФОРМАЦИОННЫМИ РЕСУРСАМИ

(Монография рекомендована к печати Ученым Советом Академии Министерства Национальной Безопасности Азербайджанской Республики)

БАКУ-2004

Научный редактор: **А.М.Аббасов**, академик НАНА, д.т.н., проф.

Научные рецензенты: **Т.А.Алиев**, академик НАНА, д.т.н.

С.Г.Багиров, член кор. НАНА, д.т.н., проф.

Ф.А.Алиев, академик НАНА, д.ф.м.н.

В.А.Касумов.

Методы информационного поиска в компьютерных сетях ссверхнасыщенными информационными ресурсами. Монография Баку. Елм. 2004. 230 стр.

В монографию были включены основные идеи и результаты исследований автора, выполненные им в последние годы по тематике докторской диссертации. В данной работе рассматриваются методы и принципы создания интеллектуальных информационно-поисковых систем, исследуются классификационные характеристики ресурсов Web-пространства, анализируются существующие модели, методы и средства индексирования и поиска информации в сети Интернет, приводится модель информационного поиска на базе нечетких знаний, излагаются методы автоматического определения тематики документов, улучшения качества тематического каталога, поиска информации по нечеткому запросу пользователя, организации распределенного поиска, предложенные автором.

Кроме этого, в работе предлагается иерархическая модель Web-пространства, исследуются интерфейсы пользователей в распределенных информационно-поисковых системах, а также описываются информационно-поисковая система и структура хранения данных в базе индексов этой системы, которые были разработаны автором на базе академической сети Интернет в конце 90-х годов. При изложении материалов в монографии использованы наглядные примеры и графические иллюстрации.

Елм - 2004

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Информационно-поисковая система является интеллектуальным инструментом, позволяющим ориентироваться в огромном информационном пространстве Интернет [10, 52, 58, 89, 94, 98-101, 120].

Как правило, все поисковые службы в ответ на запросы пользователей выдают список ссылок на источники информации, которые по мнению системы наилучшим образом отвечают потребностям пользователя. Обычно в начале списка указывается количество найденных документов, релевантных каждому слову запроса. Кроме этого, для каждого найденного документа вычисляются весовые коэффициенты, которые приписываются к этим документам в списке [48, 59, 77, 89, 112, 144].

Любой пользователь, работающий в Интернете впервые увидев в списке сотни тысячи (возможно миллионы) ссылок на документы задает себе следующие вопросы: как поисковые системы ищут документы в информационном пространстве, каким же образом она определяет степень релевантности документов, как обрабатывает этот список.

Перечисленные и множество не перечисленных вопросов сегодня все больше и больше волнуют разработчиков специализированных информационно-поисковых систем (ИПС) на базе WWW. Здесь большое значение имеет структура навигационных графов (паутина гиперссылок) Web-сайтов, корректировка и жизненный цикл Web-страниц.

Обычно создатели Web-страниц мало думают о "завтрашнем" дне Web-серверов. Планирование, масштабирование или оптимизация Web-серверов не всегда основывается на всесторонние анализы, оценки и проектирования перспективного развития. Необходимо иметь ввиду, что рано или поздно все эти вопросы возникнут и готовить основу для их решения имеет смысл уже сейчас, используя накопленный опыт эксплуатации, моделирования и анализа Web-серверов и установленных на них ИПС.

Следует отметить, что основное развитие в области ИПС с ограниченным контролем словаря индексирования относятся к 70-ым годам двадцатого века. Созданные в те времена системы такие как INIS, INSPEC, STN, NTIS, MEDLAR хорошо известны [89, 115, 143, 154-156, 165, 166]. Не менее популярны российские реферативные базы данных ВИНИТИ [87, 105].

Опыт эксплуатации именно этого класса систем и лег в основу современных информационно-поисковых систем и служб Internet. История их развития началась с первой распределенной информационно-поисковой системы Интернет, т.е. системы WAIS [98-100].

Отметим, что системы поиска информации гораздо старше систем управления базами данных. ИПС продолжают успешно развиваться оказывая влияние и на информационные ресурсы глобальных компьютерных сетей. Эти системы имеют строго определенную структуру хранения документов, которая наиболее полно описана в стандарте для разработчиков распределенных ИПС - Z.3950. Стандарт Z.3950 по своим потенциальным возможностям столь обширен, что ни одна из существующих систем не реализует его в полной мере. Реализация такой задачи в рамках реляционной модели

данных не эффективна как с точки зрения реализации системы, так и ее администрирования.

Кроме этого, в ИПС поиск информации строится на основе преобразования предложений некоторого информационно-поискового языка (ИПЯ) в запросы информационной системы [17, 59, 89, 95, 100, 162]. Язык может основываться на терминах, словоформах или устойчивых словосочетаниях, всю совокупность которых обычно называют словарем системы. Как показала практика, наилучшим решением здесь являются инвертированные списки. При этом можно над одним уровнем списков строить другие списки и т.д.

При создании словаря информационных ресурсов Интернет последний тезис мог бы не оправдывать себя, из-за быстрого роста публикаций в сети и постоянного опроса сети "пауками" - программами сканирования. Но все не так просто – здесь на помощь приходят языки запросов, модели информационных массивов и потоков, которые используются в теории информационного поиска на протяжении уже почти 20 лет и, надо сказать, хорошо себя зарекомендовали [21].

Как отметили выше, одним из способов обнаружения релевантных документов на Web пространстве является запуск программы робота, т.е. средств поиска, которых часто называют "исследователем", "червем", "пауком" и т.п. Программа робота после получения запроса пользователя систематически исследует Web-пространства, находит документы, оценивает их релевантность и возвращает пользователю ранжированный список найденных документов.

Однако, из-за больших непроизводительных потерь и экспоненциального роста Web-пространства применение подобного способа для каждого запроса пользователя является малоэффективным.

Другим подходом в поиске информации является заранее скомпилированный индекс, который периодически формируется и обновляется программами робота. Созданный таким путем индекс представляет собой архив, в котором можно искать ссылки на Web-документы. Такой подход является более практичным и поэтому многие современные инструменты поиска основаны на таком подходе [33, 50, 78, 121, 136, 138].

В настоящее время очень многие пользователи находят необходимые им материалы, обращаясь к услугам поисковых служб, таких как Yahoo (http://www.yahoo.com), Lycos (http://www.lycos.com), AltaVista (http://www.altavista.digital.com), OpenText (http://www.opentext.com), HotBot (http://www.hotbot.com) и т.д. [10, 70, 86, 123,124].

Отметим, что за последние годы развивались также русские поисковые службы и продолжают развиваться. Наиболее известные из них являются Яндекс (http://www.yandex.ru) - один из самих мощных поисковых служб в России, индексная поисковая служба «Русский Интернет» (http://www.rosit.ru/au/default.asp), охватывающая более 50 тысяч серверов Всемирной Паутины, поисковая служба «Все звезды» (http://www.stars.ru), сервер «апорт» (http://www.aport.ru), где можно найти не только официальные или частные страницы, но и тексты официальных документов различного кругозора [10, 86].

Необходимо отметить, что создание полного индекса требует систематического обхода Web-узлов Интернет, определения местонахождения каждого документа и перекачки их на сервер. Так как структура Web-пространства аналогична ориентированному графу, то здесь для охвата всего Web-пространства можно применять алгоритмы обхода графа.

Известно, что индексирование документов сети может выполняться вручную или автоматически [33, 37, 89]. Традиционно индексирование документов осуществлялось в ручную [21, 64, 100, 111]. Однако значительный объем информации Web-пространства и разнообразие ее тематики делают ручное индексирование практически неприемлемым.

Автоматическое индексирование является относительно новым направлением, чем ручное индексирование. Оно не требует наличия строго контролируемых словарей и потенциально способно отразить больше различных аспектов документа. Необходимо подчеркнуть, что несмотря на многолетние исследования, автоматическое индексирование находится пока на пути развития [36, 60, 97].

Данная монография посвящена вопросам исследования и разработки методов, средств и моделей индексирования документов информационного пространства и поиска нужной информации среди них. С этой целью исследуются классификационные характеристики ресурсов Web-пространства, анализируются существующие средства индексирования и поиска информации.

Далее в работе описываются методы и модели индексирования и поиска, а также информационно-поисковая система, разработанные автором в конце 90-х годов прошлого века впервые в республике на

базе академической сети, подробно рассматриваются архитектура системы, принцип работы поисковой системы и робота, алгоритм индексирования, структура хранения данных в базе индексов, приводятся методы декомпозиции Web-пространства, определения направления поиска и моделирования интерфейсов поисковой системы.

Монография состоит из введения, 7 глав и заключения, а также списка литературы.

I ГЛАВА.

ИНФОРМАЦИОННЫЙ ПОИСК В ДОКУМЕНТАЛЬНЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ

1.1. Основные понятия информационных систем

Сегодня, развитие общества непосредственно связана с информацией. В зависимости от характера информация разделяется на научно-техническую, экономическую, политико-управленческую, коммерческую и т.д., которые накапливаются, обрабатываются и используются в разных сферах человеческой деятельности. Начиная с 60-х годов прошлого века в тех областях, где были накоплены значительные объемы данных стали применять средства компьютерной обработки хранимой информации. Для ввода, накопления, обработки и модификации информации, на компьютере разрабатывали специальные программные средства, так называемые автоматизированные информационные системы (АИС) [5,61, 162].

Информация, используемая в традиционных АИС имеет характер фактических сведений. Такую информацию называют данными. Под ***данными*** обычно понимают информацию, которая создает фактографическое представление (перечисление характеристик, определение показателей, фактическое описание состояния объекта, процесса и т.д.) объектов в конкретной области. Другими словами, данными является множество фактических сведений об объектах, поступающих в систему обработки. Данные представляются в конкретных формах, которые адекватны возможным процессам ее обработки.

Следует отметить, что первое время АИС, в основном, работали с фактической информацией, которые характеризовали объекты и связи между ними. А интеллектуальная обработка информации, т.е. обработка текстовых документов на естественном языке, графических, звуковых и других видов данных в таких АИС появились позже. В зависимости от характера обрабатываемых информационных ресурсов АИС делят на два класса: ***фактографические АИС*** и ***документальные АИС*** [58, 59].

Как видно от названия, фактографические системы оперируют фактическими сведениями, представленными в виде совокупности записей данных, которые организованы и формализованы специальным образом. Центральным функциональным звеном таких систем являются ***системы управления базами данных***. Фактографические информационные системы используются как для реализации справочных функций, так и для решения задач обработки данных.

Под справочными функциями понимается детализация или конкретизация запрашиваемой информации, а также получение больше сведений, уточнение реквизитов, характеристик, показателей и т.д., связанных с объектом. А задачи обработки данных включают в себя класс подзадач, решаемых на компьютере и связанных с вводом, изменением, хранением, актуализацией, отбором и извлечением данных.

Данные, накопленные в фактографических АИС, как обычно, являются детально структурированными, т.е. они специальным образом организуются и систематизируются для хранения, обработки и извлечения на компьютере. Эти данные, в основном, представляют собой сведения о параметрах, показателях и других характеристиках

информационных объектов в виде числовых значений или словесных описаний. В качестве примера таких систем можно привести следующие: учет кадров, начисление зарплаты, учет материальных ценностей и товаров, банковские расчеты и т.д. [3, 6, 71].

Отметим, что фактографические системы традиционно называли ***автоматизированными информационными системами*** или ***системами управления базами данных***, в которых данные записываются в специально структурированные файлы – ***базы данных*** в виде отдельных записей.

Каждая запись представляет описание одного объекта и состоит из одного и более полей, которые содержат сведения о конкретных характеристиках или параметрах данного объекта. В таких системах реализуются все необходимые функции для работы и обработки данных, к которым относятся ввод, изменение, удаление, поиск, извлечение, сортировка, составление выводных форм (отчетов) [23, 24, 57, 61, 63].

Как видно из выше отмеченных, в традиционных фактографических системах данные представляются в четко структурированном и заранее определенном формате, а запросы для поиска формируются точно. Данные для просмотра и выходные отчеты выдаются в фиксированных формах, структура которых ограничена системой.

Однако практика показывает, что чаще всего информационные массивы содержат документы, включающие в себя тексты, таблицы, графические и звуковые данные и т.д., а не структурированные и формализованные данные. Последнее время объемы баз данных, содержащих документы, астрономически увеличиваются. Поэтому

помимо традиционных фактографических систем, в настоящее время разрабатываются документальные информационные системы.

Документальные информационные системы составляют второй основной класс АИС и их часто называют полнотекстовыми системами. Они выполняют работу не с данными, сведениями и т.д., а с документами на естественном языке, такими как книги, учебники, монографии, диссертации, авторефераты, научные статьи, законы, коммерческие предложения, описания, инструкции и т.п. Другими словами, документальные системы оперируют полнотекстовыми документами, созданные для обработки и понимания со стороны специалистов и заинтересованных людей [3, 59, 72, 89, 135].

Современные документальные системы часто охватывают огромный объем информационных ресурс, которые могут создаваться разными авторами в разных форматах и кодировках, для разных целей. В качестве ресурсов документальных информационных систем могут быть тексты в кодировках ASCII, Win, КОИ, гипертексты, рисунки в формате GIF, JPG, Photoshop, Corel, Paint, базы данных – Access, Oracle, Foxbase, Paradox, таблицы – Excel, сложные документы Word и Acrobat, содержащие тексты, графики, рисунки, таблицы и т.д., а также аудио-видео информацию.

Отсюда следует, что в документальных системах данные имеют более сложную смысловую структуру, поэтому принципы и функции представления, хранения, обработки, извлечения данных, а также составления запросов потребителя и поиска нужной ему информации коренным образом отличается от традиционных фактографических систем.

По структуре АИС делятся на ***локальные*** и ***распределенные***. Исторически сначала появились локальные АИС, включающие в себя

ресурсы одного предприятия или ведомства [5, 19, 59, 89, 107]. Локальные фактографические АИС в основном включали в себя базы данных о сотрудниках, производстве, показателях, статистике, перспективных планах, т.е. в целом информацию о деятельности предприятия. К локальным документальным АИС можно отнести системы документооборота внутри предприятия, отчеты проводимых работ, труды сотрудников, результаты научных исследований и т.д.

На этапе развития большинство этих систем постепенно расширялись и выходили из рамки одного предприятия, владельца данной системы. Это было связано, с тем, что они уже охватывали так много информации, что она с одной стороны была связана с данным предприятием, с другой стороны также входила в интересы других лиц и предприятий.

Таким образом, появились корпоративные АИС, которые создавались, заполнялись и обновлялись не одним предприятием, а множеством организаций, имеющих общий интерес. Такие системы обычно содержали научно-техническую, экономическую, правовую и др. информации. Необходимо отметить, что на развитие корпоративных информационных систем в значительной мере содействовали компьютерные сети [5, 9, 20, 31, 35, 53].

Необходимо отметить, что фактографические АИС в основном используются теми людьми, для которых они предназначены. В этом случае структура данных, формат запроса, форма диалога и т.д. известны пользователям системы, т.к. они проходят специальное обучение. А документальные АИС могут быть использованы разными потребителями для разных целей, категория и контингент которых заранее неизвестны. Кроме этого, потребители могут не иметь

представление о самой системе, а также имеющихся в ней данных, их структуре, методах поиска, принципах формулировки запросов и т.д.

Ввиду того, что традиционные документальные АИС, предназначенные для накопления большого объема документов, включали средства поиска, основанные на смысловом анализе текстовой информации по различным критериям, заданным запросами потребителей, то исторически их называли ***информационно-поисковыми системами***.

К таким системам относятся практически все документальные АИС, созданные в рамках одного предприятия и ведомства. В настоящее время такие системы также создаются и успешно функционируют на базе Интернет. В качестве примера можно привести такие ИПС, как Gopher, Wais, Archie,Whois, FTP, Listserv и т.д. [10, 30, 61].

Однако с появлением компьютерных сетей, в том числе Интернет, а также гипертекстовой технологии (языка разметки Web-страниц - HTML) и протокола HTTP, стали появляться документальные АИС без поисковых сервисов. Эти системы в основном содержат большой объем информационных ресурсов, позволяют просмотреть все ресурсы, переходить из одного ресурса на другой, извлекать и распечатать ресурсы, но функции смыслового интеллектуального поиска информации в таких системах не реализуются [11, 12, 59, 60].

Подобные системы называют просто информационными системами или информационными сайтами. Из-за того, что уже все информационные ресурсы практически создают, накапливают, хранят и обрабатывают с помощью информационных систем на базе компьютерной техники, то эти системы являются

автоматизированными. Поэтому их называют просто информационными системами и слово "автоматизированные" практически не употребляется.

Кроме этого, в настоящее время практически все фактографические и документальные АИС постепенно подключаются к Интернет и становятся всеобщими. Здесь исключениями являются личные, секретные, коммерческие информационные базы. Если даже такие АИС подключаются к Интернет, то необходимо предусматривать меры для их защиты, а вход к ним обеспечивается строгим парольным или другим безопасным методом доступа.

Поэтому в дальнейшем под ***информационными системами*** (ИС) будем понимать любую компьютеризованную систему, которая содержит как фактические и документальные, так и другие виды информационных ресурсов, а также имеет возможность подключиться к Интернет и предоставляет доступ из вне через Интернет. Информационные системы, снабженные мощными поисковыми механизмами и предназначенные для организации поиска не только в рамках данной системы, а также по массиву данных других информационных систем, будем называть ***информационно-поисковыми системами*** [22, 25, 72, 85].

Как отметили выше, ИС может содержать разнородные данные: тексты на естественном языке, простые или сложные таблицы, графические рисунки, графики, диаграммы, аудио-видео информацию – звуки, музыка, клипы, видеофильмы, анимации и т.д. Данные, входящие в ИС называются ***документами*** или ***информационными ресурсами*** (по контексту), потребители информационных ресурсов – ***пользователями*** системы.

1.2. Информационный поиск в документальных системах

Как отметили выше, с увеличением объема и разнообразия информационных ресурсов в ИС, все более остро стоит проблема поиска нужной информации в огромном потоке информации. Практически каждый человек в своей повседневной деятельности нуждается в определенной информации. Для удовлетворения своих потребностей он должен сформулировать свой поисковый запрос, обращаться к источникам или владельцам и таким образом запрашивать нужную ему информацию.

ИС как источник информации также предназначены для удовлетворения определенных информационных потребностей определенного круга пользователей. Здесь возможны два подхода:

- ***прямой доступ.*** Пользователю точно известно местонахождение и форма хранения интересующей его информации, куда нужно обращаться для получения санкции доступа (если это требуется или он еще не зарегистрирован в данной системе), принцип работы системы, методы доступа и получения информации. Таким образом он может связаться непосредственно с ИС (работая за терминалом или подключаясь через сеть) и просматривать информационные ресурсы.

- ***поиск информации.*** Пользователь знает что (какая информация) ему нужно, но не знает где и как искать ее. В таких случаях используются ИПС (или поисковые механизмы конкретных ИС). Для этого он обращается к ИПС, формулирует свой поисковый запрос и передает его в ИПС. В ответ на свой запрос пользователь получает информационные ресурсы, найденные ИПС согласно данному запросу.

Как видно, первый вариант доступа к ИС возможен тогда, когда пользователь является непосредственно владельцем информационных ресурсов или зарегистрированным пользователем, т.е. абонентом ИС. Второй вариант сегодня является наиболее распространенным и актуальным, но до конца не решенной проблемой: - поиск информации в распределенных информационных системах на базе компьютерных сетей, в частности Интернет.

Информационный поиск (ИП) является процедурой нахождения документов (информационных ресурсов), соответствующих его запросу и содержащих ответ на заданные пользователем вопросы. Осуществление информационного поиска и удовлетворение информационной потребности пользователя по его запросу является главной функцией ИПС [1, 29, 52, 89].

По объективным и субъективным причинам информационная потребность пользователей постоянно меняется, поэтому невозможно однозначно их выразить и описать. Несмотря на это, в конкретный момент времени информационную потребность можно представить в виде информационного запроса. ***Информационный запрос*** является частной формой информационной потребности, который выражается на естественном языке и передается системе пользователем.

Несмотря на то, что каждый пользователь знает свои информационные потребности, однако он может их неправильно отразить их в своем поисковом запросе, и таким образом неадекватно сформулировать его. В таком случае, ИПС может выдать документы, релевантные конкретному запросу пользователя, которые в действительности не будут соответствовать его потребностям. Поэтому для определения результативности поиска, необходимо

выяснить соответствие выданных документов не к запросу пользователя, а к его потребностям [2, 28, 73, 81, 95, 100, 101].

В теории информационного поиска для анализа результата поиска введены несколько понятий. Два основные из них являются релевантность и пертинентность. Под ***релевантностью*** понимается соответствие смыслового содержания выданных документов информационному запросу. Документы, содержания которых отвечает запросу пользователя называются ***релевантными***, а не отвечающие – ***нерелевантными***. Соответствие содержания выданных документов к информационной потребности пользователя называется ***пертинентностью***, а документы, содержания которых отвечают информационным потребностям пользователя называются ***пертинентными***.

Различают понятия фактической и формальной релевантности. ***Фактическая релевантность*** понимается как релевантность, определяемая пользователем путем осмысления содержания документов, полученных от ИПС в результате поиска, и установления степени близости их к данному запросу. ***Формальная релевантность*** - это релевантность, которая устанавливается со стороны ИПС, в результате сопоставления поискового запроса пользователя и документов информационного массива с помощью поисковых механизмов.

Понятно, что степень релевантности зависит от качества поиска, т.е. "интеллектуальных способностей" ИПС, а степень пертинентности – от пользователя, от его знания предметной области, умения выражать и описать свои потребности [86, 90-94]. Здесь, естественно, немаловажную роль играет поисковый аппарат ИПС, а

также информационно-поисковый язык (ИПЯ), которые будут рассмотрены позже.

Отметим, что степени релевантности и пертинентности документов, выданных ИПС являются неодинаковыми, т.е. нельзя сказать, что все найденные релевантные документы будут одинаково удовлетворять потребность пользователя. Документы, очень близкие запросу считаются наиболее релевантными, а другие менее релевантными. Релевантные документы, найденные ИПС выдаются пользователю в виде отсортированного списка по степени релевантности. В начале списка идут наиболее релевантные документы, после чего следует менее релевантные.

Эффективность работы ИПС характеризуется двумя основными параметрами – полнотой и точностью. ***Полнота поиска*** характеризует долю выданных ИПС релевантных документов по сравнению с количеством всех имеющихся релевантных документов в информационном (поисковом) массиве. ***Точность поиска*** характеризует долю релевантных документов, действительно соответствующих запросу пользователя среди всех выданных ИПС документов. Другими словами можно сказать, что полнота – это способность ИПС найти и выдавать все релевантные документы, а точность – это способность ИПС задерживать, т.е. не выдавать нерелевантные документы [89, 92, 100, 118, 149].

Пусть D^v – общее количество выданных ИПС документов, D^r – общее количество релевантных документов в информационном массиве, D^{vr} – количество выданных ИПС документов, являющихся релевантными. Тогда полноту и точность поиска можно выразить следующими формулами:

$$R = \frac{D^{vr}}{D^{r}}, \qquad (1.1)$$

$$P = \frac{D^{vr}}{D^{v}}, \qquad (1.2)$$

где R – коэффициент полноты (от английского слова "recall") и P – коэффициент точности (от английского слова "precision") поиска.

Однако, в таком определении не учитываются влияние некоторых немаловажных факторов, таких как общее количество документов в информационном массиве, количество выданных и нерелевантных документов и т.д. С этой целью введем следующие обозначения: D^{nr} - количество выданных, но нерелевантных запросу документов, D^{nv} - количество релевантных запросу, но не выданных документов. Тогда для вычисления значений коэффициентов полноты и точности можно использовать следующие формулы:

$$R = \frac{D^{vr}}{D^{vr} + D^{nv}}, \qquad (1.3)$$

$$P = \frac{D^{vr}}{D^{vr} + D^{nr}}, \qquad (1.4)$$

Эти параметры получают значения от 0 (наихудший случай – 0%) до 1 (наилучший случай – 100%) и они сильно взаимосвязаны. Как правило все меры, принимаемые для повышения полноты, приводят к снижению точности и наоборот, повышение точности приводит к снижению полноты. На рисунке 1.1 показана усредненная кривая зависимости между полнотой и точностью.

На практике, требования к этим параметрам, т.е. характеристикам ИПС у разных пользователей разные. Одни пользователи желают высокую полноту, т.е. получить все, что может представлять для них интерес в какой то мере, другие требуют высокой точности, т.е.

отбрасывать (не выдавать) все, что не представляет для них интереса. Эффективные ИПС находят оптимальное равновесие между

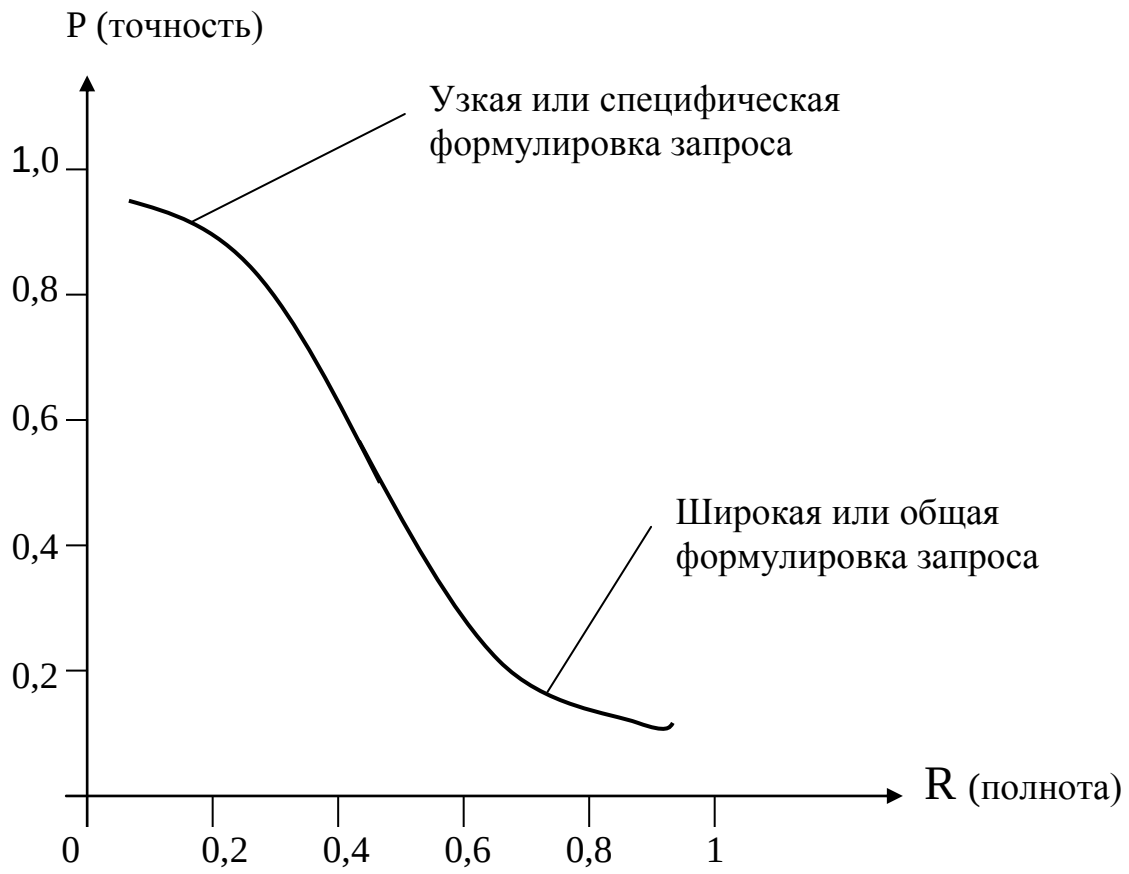

Рис.1.1. Типичная усредненная кривая зависимости "полнота-точность"

коэффициентами полноты и точности, т.е. обеспечивают как высокую полноту, так и высокую точность.

Для обеспечения оптимального уровня полноты и точности в ИПС должны быть реализованы интеллектуальные методы поиска информации. Основные принципы информационного поиска впервые были сформулированы в 30-40 гг. двадцатого века У.Е.Баттеном для системы поиска патентов, которые по настоящее время не

изменились. Хорошо известные в настоящее время операции информационного поиска можно разбить на четыре основных класса: информационный анализ, хранение данных, поисковые механизмы, выдача результата, которые будут рассмотрены в следующих параграфах.

Процедура информационного поиска в общем виде схематично показана на рисунке 1.2. Как видно из рисунка процесс информационного поиска является итеративной процедурой, т.е. в случае не удовлетворения потребностей пользователя, им ведется коррекция запроса, а потом выполняется повторный поиск по новому

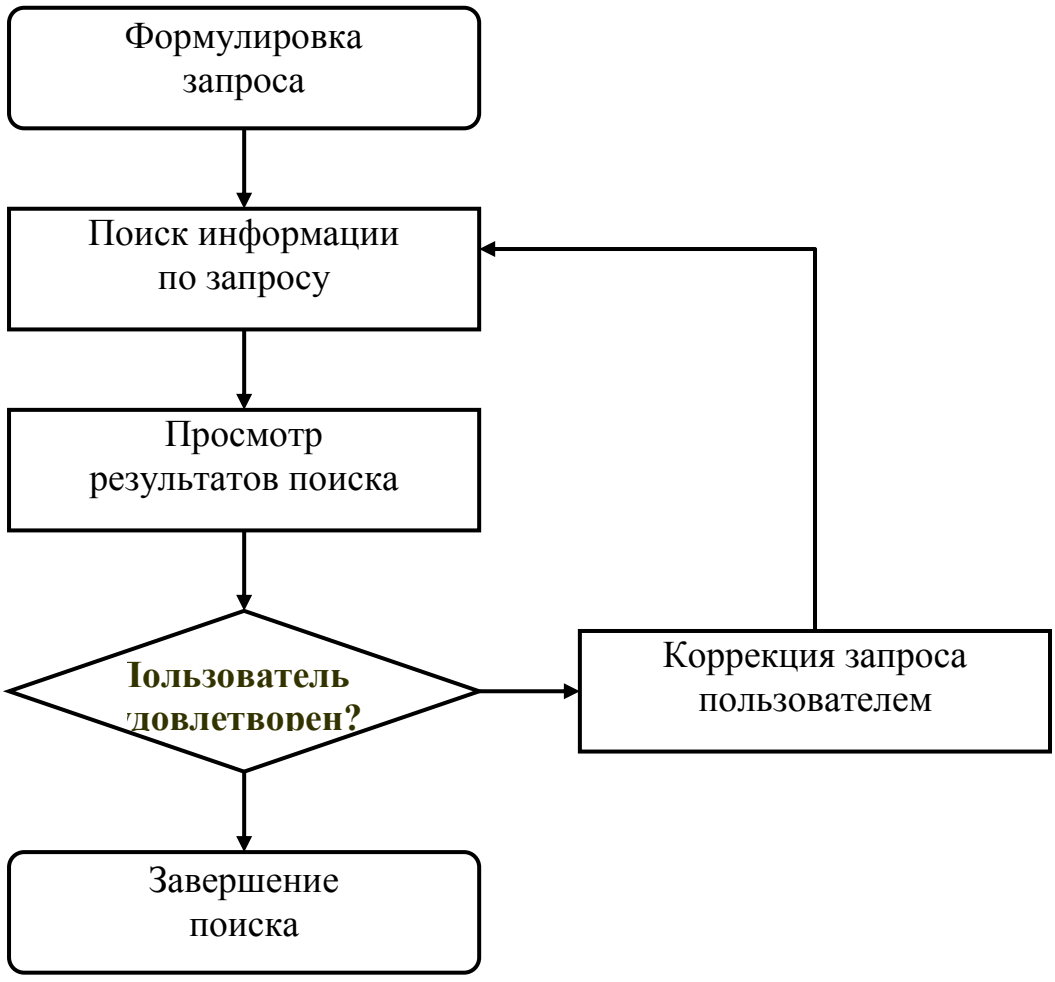

Рис.1.2. Общий вид процедуры информационного поиска

запросу. Этот процесс продолжается до полного удовлетворения потребностей пользователя.

Для организации поиска на информационном массиве должно быть создано множество указателей документов массива, которое называют поисковым образом. В ИПС содержания документов, а также запросы пользователя представляются наборами индексов - указателей, т.е. терминов (ключевых слов), состоящих из отдельных слов или словосочетаний. Иногда к индексам приписываются весовые коэффициенты, отражающих предполагаемую важность каждого из них для данного документа.

Указатель является комбинацией ключевых слов (словосочетанием) и ссылками. Слова (словосочетания) обладают неким свойством документа, описывают его характеристики, а ссылки указывают на документы, которые обладают этим свойством. В традиционных документальных ИС используются различные виды указателей. Наиболее распространенными указателями являются ***авторский, предметный*** и ***системный*** указатели.

В обоих случаях процесс создания указателей на документы называется ***индексированием***, а термины или ключевые слова, использующиеся для индексирования, называются ***терминами индексирования***. Совокупность используемых терминов индексирования называется ***словарем***. Массив указателей, полученный в результате индексирования информационных ресурсов, называется ***базой данных индексов*** или чаше всего ***индексом***, а индексы отдельных информационных ресурсов называются их ***поисковыми образами***.

После создания индекса пользователям представляется возможность обращаться к нему с помощью специальных программ –

поисковых систем посредством запросов. Так как процесс поиска заключается в сопоставлении запроса пользователя с имеющимися

Рис.1.3. Упрошенный процесс информационного поиска

поисковыми образами документов, то полученный запрос также должен быть индексирован подобно документам массива (рис.1.3).

Решение о выдаче релевантных документов принимается в результате сравнения наборов терминов, относящихся соответственно к документам и запросу. Пользователям выдаются те документы,

поисковые образы которых совпадают с поисковыми образами запроса. Однако следует отметить, что совпадение поисковых образов запроса и документов может быть частичным, т.е. возможно, что они полностью не совпадают, а являются очень близкими. При этом критерий выдачи может быть основан на вычислении коэффициента подобия для каждой пары документ-запрос, а в качестве результата следует выдать пользователю все документы, для которых значение коэффициента подобия превышает заданное пороговое значение. В таком случае возможно строго ранжировать найденные документы и выдавать их пользователю в порядке уменьшения величины подобия документа запросу.

1.3. Информационные ресурсы Интернет

На современном этапе развития, информация является стратегическим ресурсом и главной ценностью современного общества. Успех деятельности человека, результативность того или иного мероприятия определяется наличием необходимой информации. В своей повседневной деятельности человеку необходимо иметь под рукой некую информацию, для получения которой значительную роль играет новые информационные технологии, в частности Интернет.

Повсеместное применение новых информационных технологий привело к широкомасштабному переводу накопленных годами человечеством традиционных информационных ресурсов в электронную форму и созданию принципиально новых видов информационных ресурсов, которые в электронной форме приобретают новое качество, обеспечивая возможности более

широкого распространения, удобного и эффективного использования. В настоящее время использование Интернет технологий при построении информационных систем как специализированных, так и общего назначения становится доминирующим в мировом информационном пространстве.

По своей структуре Интернет является гигантской, но достаточно гибкой информационной сетью, предоставляющей широкие возможности для размещения разнотипных информационных ресурсов и весьма простой интерфейс доступа к ним. Сегодня на базе Интернет построены и успешно функционируют многочисленные серверы информационных систем, которые оказывают различные информационные услуги пользователям этой сети [8, 18, 62].

В работе приведен список основных информационных и телекоммуникационных услуг Интернет, которые являются источниками и поставщиками информации.

Электронная почта. Обычно отдельные лица и организации в своих Web-страницах указывают гиперссылку на адрес их электронной почты с помощью команды *mailto*, которая автоматически открывает режим отправки клиентской програмы e-mail [62, 69, 84]. Кроме этого, в интерактивных (on-line) средствах коммуникации пользователей, а также в системе USENET адрес электронной почты пользователя является необходимым реквизитом.

Для идентификации отдельных лиц и организаций традиционно используются их адреса электронной почты, которые свободно могут быть индексированы поисковыми системами, а также использованы через поисковые машины для поиска. В отчетах системы Alta Vista показывается, что в каждых двух документах из трех, индексированных со стороны поисковой системы встречаются адреса

электронной почты. Однако, отметим, что при получении адреса электронной почты (особенно в почтовых серверах, оказывающие бесплатные услуги) возможно регистрация пользователя под псевдонимом.

В системах, предназначенных специально для поиска людей и организаций накапливаются адреса электронной почты, которые доступны по WWW.

Для получения данных, не доступных иным способом, используются специальные программы, так называемые ***почтовые роботы***, которые работают в off-line режиме. Запрос пользователя поступает на поисковый робот по электронной почте. Почтовый робот имеет электронный адрес, подобный обычному пользовательскому адресу. Обычно запросы посылаются пользователем на поисковый робот как обыкновенное электронное сообщение. Если сообщение содержит единственное слово *help* и поле subject является пустым, тогда в качестве ответа почтовым роботом на адрес пользователя высылается перечень допустимых команд, используемых для работы с почтовым роботом. Получая от пользователя запрос, составленные согласно форматам команд, почтовый робот в ответ посылает ему соответствующую информацию.

Usenet – система телеконференции организована по принципу электронных досок объявлений [59, 62, 84]. Доски объявлений создаются по различным тематикам, на которые подписываются пользователи и составляют группу Usenet. Каждая группа имеет уникальное название, являющееся ключевым словом для глобальной системы Usenet. К настоящему времени число всех групп новостей Usenet превышает 20 тысяч. Естественно, все группы телеконференции не могут быть поддержаны одним сервером,

поэтому они разбросаны по разным серверам. Здесь трудным является нахождение самого сервера, который поддерживает нужную группу или тематику.

Зарегистрированный в системе Usenet любой пользователь может разместить свою информацию в одной из тематических групп новостей, которая будет передана всем пользователям данной группы. Пользователи, входящие в одну группу обычно имеют общие интересы и стараются по возможности помочь друг другу. Любой из пользователей, получивший это сообщение и владеющий данной информацией может передать ее запрашивающему пользователю. Данная система удобна в основном для получения сведения по какому-то узкому вопросу, частной или неофициальной информации.

Интерактивная беседа или интерактивный обмен информацией. Этот сервис предназначен для обмена информацией в режиме реального времени между двумя или несколькими пользователями Интернет с помощью специальных серверов типа ICQ, NetMeeting, Chat и др. Для организации обмена информацией может быть использован экран текстового диалога, передачи графического изображения, мультимедийный голосовой или видео режим. Пользователь желающий пользоваться услугами этой системы, должен установить у себя на компьютере клиентскую программу, после чего подключиться к подходящему серверу и пройти процедуру регистрации. Каждому зарегистрированному пользователю выдается уникальное имя или номер-идентификатор.

Система поиска людей и организаций позволяет вести поиск информации о людях и организациях, как подключенных так и неподключенных к сети Интернет. Последнее время в Интернете появляются бизнес справочники, телефонные, адресные и другие базы

данных, включающие в себя информацию об отдельных лицах или организациях [62].

Списки рассылки являются систематической рассылкой информации определенной группе пользователей в виде сообщений по электронной почте. В списки рассылки включаются адреса электронной почты пользователей по узко специальным направлениям или для рекламных целей. В настоящее время в Интернет имеется большое количество списков рассылок по различным специальным направлениям. Благодаря этой системе на адреса пользователей поступают последние новости, информация от специальных систем и серверов, информация в области профессиональных интересов.

Базы данных доступные через telnet. Благодаря более усовершенствованным сервисным услугам (WWW, FTP, E-mail и т.д.) последнее время протокол ***telnet*** используется очень мало. Однако до сих пор имеются уникальные системы, содержащие ценную и полезную информацию, которые доступны только через протокол ***telnet.*** К таким системам в основном относятся электронные каталоги библиотек, автоматизированные информационные системы государственных организаций (учет кадров, управление деятельностью и производством учреждений) и т.д. [63, 84]. Они обычно работают в командном режиме подобные операционным системам MS DOS, Unix и т.д.

FTP – протокол перемещения файлов. FTP является огромным хранилищем и архивом файлов [62, 69, 84]. С появлением Web-технологий система FTP не потеряла свое значение и до сих пор стабильно используется пользователями, так как в FTP накоплено

огромное количество ценной информации, программных продуктов, специальных баз данных и т.д.

С развитием Web-технологии созданы гипертекстовые варианты FTP-сервиса, однако из-за простоты получения и передачи файлов, удобства работы и навигации в файловой системе, организации поиска в полном объеме, переход на Web-пространство не осуществляется.

Создана глобальная поисковая система поиска по ресурсам FTP архива, которая называется Archie и доступна через WWW. Адрес одного из Web-серверов - http://ftpsearch.ntnu.no, через которые можно получить доступ к Archie. Существуют также региональные поисковые системы по FTP ресурсам. Российская система имеет адрес hhttp://ftpsearch.city.ru. Поиск информации в этих системах ведется по названию файлов и каталогов.

Система Gopher и поиск по ее ресурсам. Gopher является информационной системой в виде иерархического меню, содержащие в основном текстовую информацию. Можно сказать, что практически все ресурсы системы Gopher перенесены на WWW. Сервер gopher://gopher2.tc.umn.edu является главным сервером, который включает в себя большинство gopher-ресурсов сети Интернет [62].

WWW - всемирная паутина основана на гипертекстовой технологии и позволяет размещать в сети, просмотреть, передать и получать обычные тексты, гипертексты, графическую и мультимедийную информацию, коды программ и т.п. [11, 12, 69, 84]. Она является очень удобной для использования. Ресурсы в WWW представляются в виде Web-страниц, которые могут содержать ссылки на другие страницы, информационные ресурсы, графику,

рисунки, анимацию и т.д. ***Ссылки***, обеспечивают переход из одной страницы на другую, т.е. навигацию по WWW пространству.

В настоящее время среди других услуг Интернет WWW занимает главное место. Мощность и объем информационных ресурсов, а также возможности этой системы растет с астрономической скоростью. Объем информационных ресурсов WWW перерос настолько, что найти нужную информацию в этом грандиозном информационном массиве стало очень трудно решаемой задачей. Здесь на помощь пришли поисковые средства, которые содержат полную информацию об информационных ресурсах Интернет и позволяют вести поиск информации по запросу пользователя.

На базе современных информационно-поисковых систем в среде WWW стоят два основных подхода:

- ***тематические каталоги;***

- ***автоматические индексы.***

Тематические каталоги являются иерархической системой тематически систематизированных информационных ресурсов. База данных тематических каталогов содержит информацию об информационных ресурсах, которые тематически разбиваются на подкаталоги, а последние также разбиваются на подкаталоги по подтемам и т.д.

Таким образом, информационные ресурсы классифицируются по тематике, организуется конкретизация тематики и локализация нужных ресурсов, а также доступ к ним путем последовательного перехода по иерархической структуре тематического каталога начиная от верхнего уровня до нижних. Отметим, что последнее время появилась тенденция разработки методов автоматического создания

тематических каталогов, однако в настоящее время эта работа в основном выполняется вручную человеком - специалистом.

Автоматические индексы создаются с помощью специальных программ – ***роботов***. Программа робота сканирует информационные ресурсы Web-пространства, извлекает из них термины и ключевые слова, включает их в свою базу. К терминам и ключевым словам приписываются коэффициенты важности и реквизиты документов. Поиск документов ведется по ключевым словам, которые пользователи включают в свои запросы.

Баннерные системы используются для рекламной цели и являются графическими изображениями. Баннеры появляются на открываемых пользователем Web-страницах и могут не иметь никакого отношения к основному содержимому данной страницы, гиперссылки которых ведут пользователей к владельцам, т.е. на сервер рекламодателя.

Активные информационные ресурсы. На некоторых специализированных Web-серверах организуются подписки на информационные каналы. Пользователи, интересующиеся тематикой сервера, проблемами, рассматриваемыми сервером или представляемой в нем информацией, могут регистрироваться в них, после чего появляющие в системе новости и новые данные регулярно отправляются на адрес пользователя. Системы такого назначения называются push-технологией.

Отметим, что последнее время в Интернет появились новые услуги, такие как Интернет-телефония, Интернет-телевидения, Интернет-радио и т.п. Все выше рассмотренные информационные системы и ресурсы специфичны и обладают своими особенностями. Однако учитывая тот факт, что практически все эти ресурсы уже

перенесены на WWW или доступны через нее, то достаточно будет рассматривать особенности ресурсов Web-пространства.

1.4. Особенности поисковых средств Интернет

Технология создания информационных служб сети Интернет очень привлекательна и удобна для размещения, получения и передачи любой информации, но, с другой стороны, с увеличением объема информационного массива, размера архивов данных, количества информационных ресурсов и т.д. поиск нужной информации становится все более сложным и проблематичным.

Для приобретения нужной информации каждый раз пользователи вынуждены связаться с различными информационными серверами, переходить из одного источника информации к другому и таким образом пронизывать всю сеть, при этом затрачивая на поиск массу времени и определенные информационные, сетевые и материальные ресурсы. Используемый таким образом инструментарий может существенно помочь поиску и подбору нужной информации или чаше всего, наоборот, способствовать напрасной трате времени и привести в заблуждение в лабиринтах бесчисленных Web-узлов.

Необходимо отметить, что информационное пространство Интернет по своей природе является распределенной информационной системой, однако в отличии от традиционных информационных систем, где все ресурсы расположены в базах данных локальной или корпоративной сети (возможно даже база данных является автономной) и под контролем администратора сети, оно имеет ряд существенных особенностей, связанных с недостатками и преимуществами сети и ее услуг:

Размер информационного пространства. Объем информации, расположенной по многотысячным узлам Интернет, очень большой. Поэтому поисковые сервера не могут полностью охватить все ресурсы информационного массива. Часто информационные ресурсы являются распределенными, часть которых размещается на одном сервере, другая часть на другом.

Хаотичность. Информационные ресурсы Интернет по размещению являются очень хаотичными. Отсутствует закономерность их создания, сбора и хранения, следовательно, информация бывает в значительной степени в раздробленной форме и разбрасывается по разным узлам сети (сетевой хаос) по всему миру.

Отсутствие систематизации. Так как создание новой и изменение старой информации очень легко, любой желающий пользователь как имеющий отношение к Интернет, так и не имеющий, может создать собственную страницу, включить в нее любую информацию и "повесить" в Интернет. Таким образом информация в сети появляется случайным образом. В целом т.е. организация информационного обеспечения серверов осуществляется по желанию владельцов информационных ресурсов, страниц или сайтов, при котором невозможно обеспечить их систематичность.

Неполнота, избыточность, противоречивость. По этой же причине возникает другая проблема. Так как информационные ресурсы в Интернет появляются автономно, независимо друг от друга, в разное время, в разных местах, то естественно, они друг с другом не согласуются, и в этом случае имеет место неполнота, избыточность и взаимная противоречивость информации.

Различные языки и кодировки. Так как информационные ресурсы создаются разными людьми, в разных местах (городах и странах), на разных компьютерах и системах для разных целей, то в результате возникают дополнительные проблемы, связанные с различием языков и кодировок ресурсов (особенно национальных ресурсов).

Разнообразие терминологии. Используемая авторами терминология меняется в зависимости от даты подготовки, природы появления и назначения, научно-методического подхода разработки, индивидуальных способностей и умения авторов и т.д. А это создает проблемы поиска и автоматической классификации по тематическим профилям, требует применения дополнительного аппарата (справочников словарей, тезаурусов и ассоциирующих слов).

Ценность и жизненный цикл информации. Часто в сети появляются информационные ресурсы, не представляющий никакой ценности для других пользователей, кроме ее автора. А информация, появляющаяся в Интернет и представляющая ценность, через некоторое время может потерять свое значение и ценность ее может падать до нуля, т.е. со временем информация может стареть. Возможен еще такой случай, что информация уже при появлении в Интернет оказывается неинтересной и устаревшей. В некоторых случаях Web-страницы и сервера создаются отдельными людьми и организациями, а потом не обслуживаются, т.е. информация на сайте или на странице не обновляется, не дополняется, не модифицируются и т.д. Другими словами, они являются заброшенными "информационным мусором".

Преимущества и недостатки использования гиперссылок. Применение гипертекстовой технологии с одной стороны облегчает

работу, обеспечивая переход из одного сервера на другой, близкий по тематике, с другой стороны она может привести в тупик, указывая на неправильное направление или совсем не нужное место, так как гиперссылки создаются по субъективному мнению авторов. Они могут ссылать на заброшенные, никому не нужные, не представляющие ни какой ценности, давно устаревшие, удаленные, модифицированные или перенесенные страницы или сайты.

Астрономический рост мощности потоков документов в Интернет требовал разработку и применение интеллектуальных поисковых средств, которые могли обеспечить "человеческие" масштабы результатов поиска. При этом не следует забывать, что естественный язык (по сути, много разных естественных языков) играет важную роль в сфере электронного документооборота и информационного поиска, что требует значительного развития средств автоматизированной обработки, а также поиска электронной естественно-языковой информации.

Важность поисковых средств по-настоящему была осознана лишь с развитием Web-технологии, т.е. всемирной паутины (World Wide Web -WWW), однако первые поисковые инструменты Интернет были созданы намного раньше. Еще в 80-х годах, когда на базе услуг Интернет, таких как FTP, Gopher, E-mail, Telnet, создавались, обрабатывались, накапливались и передавались достаточно большие потоки информационных ресурсов, возникла потребность на разработку и применения поисковых средств.

Однако с развитием WWW увеличивалась возможность легко передавать, получать, размещать на сервере и извлекать из сервера различные виды информационных ресурсов: тексты всех форматов (ASCII, Win, KOI и т.д.), высококачественные цветные изображения,

графику, аудио и видео информацию, а также смешанные данные – тексты, содержащие графику, рисунки и т. д.

Благодаря WWW и другим технологиям объем данных стал увеличиваться столь стремительно, что в скором времени Интернет превратился в огромное хранилище информации, напоминающее непроходимые киберджунгли, ориентироваться в которых стало крайне трудно. Гипертекстовая технология, стоящая на базе WWW по гиперссылкам всегда указывает куда идти дальше, но как отметили выше эти гиперссылки не всегда приводят в нужное место. Таким образом пользователь может попасть на страницу, которая совсем его не интересует или через некоторое время он может вернуться на тоже самое место, что было несколько шагов назад, т. е. прийти в тупик и т.д. Однако в отличии от настоящих джунглей, в киберджунглях запоминаются дороги назад, но вернуться назад на первоначальное место в киберджунглях также требует время, материальные и сетевые ресурсы, аналогично тому как идти вперед.

Для демонстрации различия между ИС Интернет достаточно перечислить некоторые разновидности информационных ресурсов:

- ***электронные издания*** - периодические электронные журналы, газеты, диссертации, авторефераты, монографии, обзоры, бюллетени, материалы конференций, симпозиумов и т.д.

- ***электронные библиотеки*** – электронные варианты традиционных библиотек, электронные каталоги и т.д.

- ***информационные массивы Интернет*** – информационные системы WWW, FTP-архивы, ресурсы Gopher, Usenet и др.

- ***побочные продукты издательской деятельности*** – электронные варианты издаваемых книг, учебников, монографий, журналов, газет т т.п.

- ***информационные ресурсы специального назначения*** – данные, вводимые в компьютеры для служебных целей, тексты отчетов, результаты научных исследований, статьи для публикаций, разные документы и т.д.

К настоящему времени для организации поиска информации в Интернет, разрабатывались многочисленные специализированные ИПС, которые позволяют вести поиск по всему пространству Интернет. Описание функциональной структуры, назначение и возможности ИПС будут рассмотрены в следующих главах.

1.5. Основные задачи и критерии информационного поиска в Интернет

В общем виде предполагается, что традиционно все ИПС распределенных информационных систем таких как Интернет должны решать пять основные задачи.

Первой задачей, поставленной перед ИПС для Интернет является определение местонахождения информации и загрузка ее из места хранения на поисковый сервер. Под местонахождением понимается Web-сайт или Web-страница, FTP или Gopher сервер, хранилище статей Usenet и т.д. ***Загрузка информации*** – это перекачка файлов данных, содержащих информационные ресурсы, т.е. тексты, рисунки, аудио-видео информацию и т.д.

Вторая задача – подбор и извлечение терминов и ключевых слов (индексирование) для наиболее точного отражения содержания источников, на основе которых создается поисковый образ документов. Так как сбор и хранение всех ресурсов, расположенных в

Интернет, на поисковый сервер не возможно и не имеет смысла, поэтому в базе ИПС вместо самих документов хранятся их образы, для создания которых используются два механизма:

- из названия, заголовка и текста информации, а также из других служебных данных этих ресурсов извлекаются значимые ключевые слова и включаются в базы индексов;

- на основе смыслового анализа документов из специального словаря выбираются термины, которые наиболее точно отражают содержания документов, и приписываются к ним. В дальнейшем эти термины включаются в базу индексов в качестве ключевых слов документов.

Третьей задачей является организация пользовательского интерфейса, т. е. диалога с пользователем. Интерфейс пользователя должен полностью отражать возможности ИПС, позволять пользователю удобным образом сформулировать свои потребности в виде простых и сложных запросов, модифицировать, расширять или сузить область поиска, обеспечить обратную связь с пользователем для улучшения формулировки запроса и поисковых образов документов путем уточнения степени близости выданных документов к запросу.

Осуществление непосредственно в базе индексов поиска документов, релевантных запросу пользователя является ***четвертой задачей***.

Наконец, ***пятая задача*** – эта систематизация и ранжирование найденных документов специальным образом и выдача их пользователю в виде отсортированного списка, в начале которого идут наиболее важные документы.

Учитывая особенности информационных ресурсов Интернет для решения вышеперечисленных задач к разрабатываемым поисковым средствам (к ИПС) предъявляются дополнительные требования, основные из которых являются.

- обеспечение максимальной полноты охвата информационных ресурсов Интернет;

- обеспечение достоверности и непротиворечивости информации, полученной из Интернет;

- реализация интеллектуальных методов автоматического индексирования информационных ресурсов и создания тематического каталога;

- обеспечение высокой скорости реакции системы на запрос пользователя и проведения поиска по нему;

- обеспечение точности выданных результатов, т.е. полученные результаты должны наиболее точно соответствовать потребностям пользователя;

- реализация пользовательского интерфейса на естественном языке.

Не возможно найти такую идеальную ИПС, которая удовлетворяла бы все выше отмеченные требования на достаточном уровне. В общем виде существующие ИПС можно классифицировать по следующим критериям.

1. ***По виду базы метаданных***, состоящих из поисковых образов документов, выделяются три типа ИПС:

- *Тематические каталоги* аналогичны каталогам в традиционных библиотечных системах, которые создаются вручную операторами или администратором системы. Тематические каталоги требуют от

составителя хорошее знание тематики источников и ИПЯ, используемого в ИПС.

- *Автоматические индексы*, создаваемые специальными программами, называемыми роботом, пауком, спайдером и т.д., которые сами автоматически собирают данные об информационных системах.

- *Гибридные базы*, включающие в себя как тематические каталоги, так и автоматические индексы.

2. ***По методам выбора терминов***, описывающих источники информации, ИПС можно разбить на две группы:

- Первая группа использует *приписной метод* выбора терминов. Согласно данному методу термины, приписываемые к документам могут отсутствовать в их содержании, но они наиболее точно должны отражать содержание этого документа по смыслу, для чего используются различные словари, тезаурусы и справочники.

- Ко второй группе относятся ИПС, реализующие *выборочный метод*. В этом случае термины выбираются и извлекаются из тела документа с помощью специальных методов индексирования.

3. ***По способам реализации процедур индексирования*** ИПС также разделяют на две группы:

- *ИПС с ручным индексированием*, где процесс индексирования выполняется вручную человеком, который хорошо знает данную область.

- *ИПС с автоматическим индексированием*, где индексирование информационных ресурсов Интернет осуществляется программными средствами - роботами.

4. ***По используемому ИПЯ*** можно выделить следующие категории:

- *ИПС с* простым *диалогом*, позволяющие вести простой поиск по ключевым словам или тематическому каталогу.

- *ИПС с* развитым *диалогом*, позволяющие составить сложные поисковые запросы с помощью логических операций.

5. ***По методам выдачи результатов*** пользователю в ИПС реализуются два подхода:

- *простой список* без анализа степени релевантности;

- *ранжированный (систематизированный) список* по степени релевантности документов к запросу.

6. ***По методу организации связи с пользователями*** выделяются следующие ИПС:

- *без обратной связи* с пользователями;

- имеющие *обратную связь с пользователями*.

Для решения вышеуказанных задач и достижения хорошего поискового эффекта требуется создания интеллектуальных систем поиска. Однако следует отметить, что эти задачи являются укрупненными и их можно разбить на подзадачи. Перечислим наиболее важнейшие из них:

- представление документов (источников информационных ресурсов) в Интернет, а также в базе индексов ИПС;

- поиск сайтов, Web-страниц и других ресурсов в Интернет;

- индексирование документов, создание их поисковых образов;

- создание рубрикаторов, справочников терминов, синонимов, тезаурусов и ассоциирующих слов;

- создание тематических каталогов, определение тематики документов и распределение их по каталогам согласно определенному рубрикатору;

- создание структурированной базы индексов, т.е. базы метаданных;

- построение интеллектуальных интерфейсов между пользователем и системой;

- обработка (сканирование и индексирование) запросов пользователей;

- поиск документов из базы индексов, релевантных запросу пользователя;

- навигация по тематическому каталогу системы;

- определение степени релевантности выданных документов и организация выдачи результатов в ранжированном виде;

- реализация механизмов обратной связи с пользователем;

- корректировка весов терминов документов и улучшение базы индексов.

Последнее время с развитием информационных технологий становятся актуальными и следующие задачи:

- распределение поисковых средств;

- распределение баз индексов;

- создание поисковых агентов;

- автоматическое определение тематики документов и автоматическое создание тематических каталогов;

- повышение эффективности методов индексирования и поиска информации;

- создание интеллектуальных интерфейсов;

- применение методов нечетких множеств, искусственного интеллекта и нейронных сетей для реализации средств информационного поиска.

В следующих главах данной монографии рассматриваются и исследуются наиболее важные и актуальные задачи информационного поиска, однако автор не ставит перед собой цель дать исчерпывающий ответ на все вопросы в области информационного поиска.

II ГЛАВА.

МОДЕЛИ И МЕТОДЫ ИНФОРМАЦИОННОГО ПОИСКА В ИНТЕРНЕТ

2.1. Архитектура информационно-поисковых систем Интернет

В предыдущей главе были рассмотрены основные задачи ИПС для распределенных информационных сетей, таких как Интернет, а также и требования к ним. Исходя из этого можно сказать, что какие функции должны выполнять ИПС. Прежде всего ИПС должны иметь возможность доступа к информационным ресурсам сети, быть способным обрабатывать различные виды информации, знать и соблюдать требования владельцев информационных ресурсов, удовлетворять информационные потребности пользователей.

По статистическим данным информационные ресурсы Интернет удваиваются каждый полгода. Увеличение объемов данных и отсутствие механизмов их систематизации приводят к проблемам поиска и нахождения нужных данных, источников информации, документов, файлов и т.д. Как было сказано выше, для облегчения работы пользователей при поиске нужной им информации используются ИПС.

В настоящее время существуют достаточно много поисковых систем, цель создания которых заключается в накоплении данных в таком виде, чтобы пользователи получали богатый сервис и быстрый доступ к данным, а также с их помощью смогли определять где можно

найти нужную им информацию, хранимую на информационных серверах Интернет не обращаясь к этим серверам.

В целом, ИПС можно рассматривать как путеводитель по информационному пространству для пользователей информационных услуг. Другими словами, как обычно, пользователь знает какая информация ему нужна, но не знает где и как ее найти, хотя он знает и уверен, что такая информация в сети существует. Именно для решения данной проблемы, т.е. для удовлетворения информационных потребностей пользователей используются программы посредники, путеводители – ИПС, которые разными путями (создавая тематические каталоги, автоматически индексируя источников информации, запрашивая информацию у владельцев ресурсов и т.д.) накапливают "знания" о содержании - контенте информационных ресурсов сети.

Общий вид функциональной структуры ИПС для Интернет приведена на рисунке 2.1, основными компонентами которой являются [47, 49, 100]:

- ***программа робота*** - индексирует информационные ресурсы Интернет и создает их образы;
- ***БД индексов*** - содержит образы (индексы) информационных ресурсов Интернет;
- ***поисковая машина*** - программа поиска информации в БД индексов поисковой системы по запросам пользователей;
- ***интерфейсная программа*** - позволяет пользователям сформулировать свои запросы для поиска необходимой информации с помощью CGI-скриптов;
- ***клиентская программа*** – это программное обеспечение, позволяющее подключиться к ИПС и выдать на экран пользователя

окно интерфейса Роль клиентской программы выполняют браузеры Web-страниц, такие как Internet Explorer, Netscape Communicator и т.д.

Рис.2.1. Общая архитектура информационно-поисковой системы Интернет

Как видно, информационно-поисковые системы Интернет состоят из комплекса программных систем и баз данных, каждая подсистема которого является самостоятельным блоком и работает в тесной связи с другими.

2.2. Информационно-поисковые языки

При обращении к информационно-поисковым системам Интернет пользователю предоставляется интерфейсная служба,

которая использует язык запросов системы. Как обычно, ИПЯ являются до некоторой степени искусственными. Так, невозможность использования естественного языка в качестве основного средства общения с системой и представления информации в ИПС приводит к необходимости применения искусственных языковых средств. Создание ИПЯ как естественного языка сегодня не является реальным. Эпоха такого подхода давно осталась позади.

Информационно-поисковым языком называется специализированный искусственный язык, предназначенный для представления основного смыслового содержания документов в базе индексов системы, а также, запросов, с целью обеспечения возможности последующего поиска [59, 67, 89, 98, 99].

После сравнения запроса с поисковыми образами документов поисковая система с помощью ИПЯ выдает пользователю список ссылок на конкретные документы. По мнению ИПС выданные пользователю документы являются наиболее релевантными его запросу.

В существующих поисковых системах Интернет, таких как Alta Vista, Excite, Open Text, Yahoo, Google, Yandex, Rambler и т.д. реализованы ИПЯ с различными возможностями, позволяющие формировать запросы от самых простых до самых сложных.

Естественно, ИПЯ должен быть понятным и удобным как для поисковой системы, так и для ее пользователей. Это означает, при разработке ИПЯ должны быть хорошо и всесторонне продуманы форма, словарный состав и синтаксис языка.

Словарный состав - это совокупность терминов, дескрипторов, ключевых слов, а также их синонимов и ассоциирующих слов, используемых в языке, для описания документов и запросов.

Синтаксис языка - это совокупность правил построения из элементов словарного состава лексических единиц, значения (смысл) которых нельзя выразить с помощью отдельных слов основного (базового) словаря. ***Лексическими единицами*** называются элементы ключевых слов, терминов, словосочетаний и т.д., которые позволяют отличать одни документы (или группы документов) от других.

В общем различают два основных типа ИПЯ, которые реализуются в ИПС: классификационные и дескрипторные.

Классификационные ИПЯ. В словарный состав таких языков заранее включаются как простые термины, так и сложные (словосочетания, фразы, переводы терминов, синонимы, родственные слова и т.д.). Для представления смыслового содержания документов и запросов используются отдельные элементы этого набора, а также готовые сложные выражения. В таких языках составление сложных запросов заменяется выбором подходящих сложных элементов из словарного состава. Таким образом, с помощью уже имеющихся в составе языка простых и сложных лексических элементов документы привязываются к классификационным классам (например, рубрикам тематического рубрикатора).

Дескрипторные ИПЯ позволяют описать смысловое содержание документов в запросах во время их составления путем включения и объединения лексических единиц. В таких языках не имеются заранее подготовленные предложения и фразы, отсутствуют какие-либо ограничения на формы и структуры при составлении сложных запросов.

По используемым правилам формирования синтаксических конструкций, дескрипторные ИПЯ делят на языки ***с грамматикой*** и ***без грамматики***. В языках с грамматикой используются жесткие

правила для описания документов и формулировки запросов. В отличии от естественного языка в таких языках могут быть использованы строгие формы представления, например, предложение на естественном языке:

"поисковая система имеет язык запросов",

синтаксис: {субъект, действие, объект}

может быть представлена в виде:

"иметь, поисковая система, язык запросов",

синтаксис: {действие, субъект, объект}.

А языки без грамматики не имеют никаких ограничений и синтаксических правил.

По лексическому составу дескрипторных ИПЯ различают языки ***с контролируемой лексикой*** и ***со свободной лексикой***. Лексический состав языков с контролируемой лексикой строго ограничен и зафиксирован, а для языков со свободной лексикой нет ограничений и они могут постоянно пополняться путем включения новых лексических единиц.

В отношении к ИПЯ поисковых систем Интернет выделяют некоторые наиболее важные понятия.

1. *Словарь терминов индексирования.* Множество поисковых терминов, такие как дескрипторы (имена понятий) и ключевые слова, которые могут объединяться в словосочетания типа фразы, например, "информационный поиск".

2. *Логические операторы.* Для формирования запросов в ИПЯ используются логические операторы, т.е. булевские операции такие, как "AND" ("И"), "OR" ("ИЛИ"), "NOT" ("НЕ"). Синтаксис составления логических выражений с помощью этих операторов в различных системах отличаются.

3. *Нормализация запросов.* Чтобы одни и те же термины и ключевые слова использовались в единой форме, в информационно-поисковых системах реализуются механизмы нормализации (преобразования в единую форму) лексических единиц. Принцип нормализации лексических единиц документов и запросов может ***задаваться пользователем во время формирования запроса*** с помощью возможностей поискового языка или ***реализоваться в поисковых алгоритмах системы,*** о существовании которых пользователи представление не имеют и не замечают их действий во время процесса поиска. Следует отметить, что существуют и другие механизмы нормализации запросов.

4. *Грамматика языка.* В состав языка входят средства, используемые совместно с индексируемыми терминами для расширения или сужения определенных понятий. В качестве таких средств можно отнести следующие: использование словосочетаний; применение логического оператора "AND", означающего условие совместного вхождения терминов в документы; реализация логического оператора "NEAR", указывающего расстояние между терминами в документе и т.д.

5. *Критерий релевантности (нерелевантности).* В составе языка определяются правила признания документов релевантными (нерелевантными) запросу пользователя, а также выдачи (невыдачи) этих документов в качестве результата поиска. Результат выводится на экран пользователя как ранжированный список, для чего используются весовые коэффициенты, присваиваемые терминами, позволяющие определить величину близости документа запросу пользователя.

6. *Дополнительные условия поиска.* Для повышения точности и полноты поиска используются различные дополнительные условия поиска, к которым относятся поиск в определенных частях документа, сужения области поиска, использование временного интервала даты создания документов, тезаурусов, словарей ассоциативных слов и т.д.

7. *Форма выдачи результатов поиска.* Для удобности использования, список найденных документов выдается пользователю в определенной форме, которая характеризуется следующими параметрами:

- ранжирование списка выдаваемых документов;
- вид списка;
- включение в список дополнительных данных (например, фрагмент текста) о документах;
- объем порции списка документов, т.е. количества документов на каждой выходной странице;
- ограничение общего количества выдаваемых документов.

8. *Обратная связь с пользователем.* В процессе поиска пользователю представляется возможность уточнить свой запрос, улучшить представление документов, а также вести поиск документов, "похожих" конкретному указанному документу из числа найденных.

В завершении отметим, что создание хорошего диалога в информационно-поисковых системах между пользователем и системой позволяет ему более точно и удобно формировать свои запросы, которые в конечном итоге влияют на процесс и результаты поиска. ИПЯ, используемые для организации интеллектуального пользовательского интерфейса и службы формирования запросов напрямую зависят от моделей представления документов и хранения

данных, а также методов индексирования и поиска информации, которые рассматриваются в следующем параграфе.

2.3. Индексирование и представление информационных ресурсов Интернет

Как отметили выше, для того, чтобы ИПС удовлетворяла информационные потребности пользователя, т.е. нашла и выдала ему все документы, соответствующие его запросу, система заранее должна обладать некоторой информацией о всех ресурсах информационного пространства. Для сбора необходимой информации о документах в свою базу используются следующие механизмы, каждый из которых имеют свои преимущества и недостатки:

- автоматические индексы, создаваемые программами робота;
- тематические каталоги, создаваемые ручным способом и с помощью специальных программ (автоматические тематические каталоги);
- рейтинговые поисковые индексы;
- комбинированные индексные базы, включающие как автоматические индексы, так и тематические каталоги.

В поисковой системе каждый документ представляется поисковым образом - ПОД (поисковый образ документов). ПОД – это информация о документе информационного массива, которая полностью отражает суть содержания документа. ПОД является результатом применения некоторой процедуры индексирования и отображения ресурсов информационного массива в некоторую форму в рамках используемой модели.

Существуют ряд методов описания документов, моделей

индексирования и поиска информации, а также связанные с ними информационно-поисковые языки.

Для создания ПОД система должна решать проблему приписывания множества терминов документу. Процедура определения списка соответствующих терминов и приписывания их документу называется индексированием, а ПОД – индексом документа.

Одной из проблем, связанных с индексированием, заключается в том, что для приписывания терминов документу необходимо создание словаря - фиксированной совокупности терминов, из которого эти термины выбираются.

Подобно ИПЯ, традиционно в поисковых системах использовались два вида словарей:

- контролируемые словари;
- свободные словари.

Контролируемый словарь является лексической базой данных, добавление терминов в которую производится администратором системы. Все новые документы могут быть заиндексированы только существующими в этой базе данных терминами.

Свободный словарь пополняется автоматически по мере появления новых терминов в документах. На момент актуализации свободный словарь также фиксируется, который предполагает полную перезагрузку базы данных. В момент обновления производится переиндексация документов. При этом процедура актуализации занимает достаточно много времени и доступ к системе в момент ее актуализации закрывается.

Из-за того, что информационные ресурсы в Интернете появляются и исчезают ежедневно, то должен быть реализован

механизм периодического обновления индексов, т.е. переиндексации документов, в качестве которых, как отметили выше, используются поисковые роботы.

Отметим, что при индексировании, в основном, используются следующие механизмы:

- ***приписка терминов*** из тематического рубрикатора, соответствующих контексту, т.е. смысловому содержанию документа;

- ***извлечение*** терминов из тела информационных ресурсов путем смыслового анализа текста документа согласно тематическому рубрикатору;

- комбинированный ***метод***, включающий в себя как первый, так и второй механизм. Например, из тела документа извлекается какой-то термин, а далее с помощью словарей тезаурусов, ассоциирующих слов ПОД дополняется однозначными терминами, синонимами и т.д.

Следует отметить, что первый подход требует применения мощного семантического аппарата или ручного индексирования со стороны специалистов в области тематики индексируемого ресурса. Методы, основанные на семантическом анализе, трудно реализовать и, вообще, мало развиты.

Ручное индексирование является более точным, однако, создание хорошего тематического каталога зависит от профессионализма персонала, для чего требуется огромный интеллектуальный труд, знание и понимание предметной области, т.е. помощь специалистов и экспертов по предметным областям.

Традиционно, работу по созданию и адаптации тематических каталогов выполняет администратор или оператор поисковой системы ручным способом. Именно поэтому процесс создания тематического каталога называют ручным индексированием.

ИПС, в которых реализованы тематические каталоги называются ***тематическими каталогами***. Они подобно библиотечной системе имеют в своем распоряжении ПОД в виде тематических каталогов, составленные по направлениям и содержащие подкаталоги по более узким направлениям. В подкаталоги тоже могут входит свои подкаталоги и т.д. Таким образом, тематические каталоги имеют иерархическую древовидную структуру.

В таких системах поиск ведется или по тематике, которая определяется названием этих каталогов, или по ключевым словам, т.е. по терминам, входящих в тематические каталоги.

Методы, позволяющие автоматически извлечь из содержания информационного ресурса наиболее важные термины и вычислить коэффициенты их важности, называются ***автоматическими индексами*** [33, 69].

Такого типа поисковые системы работают без вмешательства специалистов и экспертов. Как отметили выше, ***поисковые роботы*** «гуляют» по сайтам сети, собирают информацию и заносят их в свои базы данных. В этом случае отсутствует тематическое разделение документов и поиск ведется исключительно по ключевым словам [50, 78, 114, 122].

При индексировании поисковые роботы определяют какие страницы можно индексировать, а какие нельзя. Для это используется ***стандарт исключения для роботов***, суть которого заключается в следующем: администратор Web-сайтов на своем сервере создает файл "robot.txt", в котором он указывает индексирование каких файлов разрешается, а каких не разрешается. Другими словами, в этом файле точно описывается политика индексирования данного сайта для "посторонних" роботов [136, 138].

Слабым местом методов автоматического индексирования является следующее:

- использование разных терминов разными авторами, смыслы которых очень близки (возможно идентичны), например, “information retrieval” и “information search”;

- в индексируемом источнике выделяется важное ключевое слово, которое является не распространенным термином в данной области или не входит в тематический рубрикатор, а синонимы данного термина могут оказаться значимыми терминами;

- одни термины могут ассоциировать другие термины, достаточно близкие по тематике;

- в информационном ресурсе вместо термина могут быть использованы их английские, латинские или другие эквиваленты;

- языки источников по одной и той же тематике могут быть разными.

Без учета вышеперечисленных обстоятельств нельзя достичь достаточного уровня индексирования, от которого непосредственно зависит результат поиска. Выходом из положения является создание и использование словарей синонимов и ассоциирующих слов, а также их переводов на другие языки. Это приводит к организации мета-поисковых и мульти-языковых поисковых систем.

А также разработчиками поисковых систем используются словари запрещенных ("stop-words"), общих и служебных слов (союзы, предлоги, местоимения, глаголы и т.д.), которые не могут быть использованы в качестве индексируемых терминов. Кроме этого, при индексировании производится нормализация лексики.

Методы извлечения терминов из текста документов называются методами ***полнотекстового индексирования*** [51, 60, 102,-104, 133,

149]. При полнотекстовом индексировании выбранные из содержания документа слова попадают в ПОД только после сравнения с множеством различных словарей, а потом они включаются в базу индексов системы. Для того, чтобы искусственно не раздувать словари и индексы, к терминам приписываются весовые коэффициенты, которые показывают важность данного термина для документа. Весовой коэффициент получает значение в интервале [0,1], где значение 0 означает, что термин не встречается в документе, т.е. термин и документ не имеют никакого родственного отношения, а значение 1 – термин стопроцентно соответствует содержанию документа.

Таким образом, наиболее важные термины, например, термины имеющие "вес" больше, чем 0,5 (т.е. в интервале [0,5-1]) включаются в базу индексов.

Следует отметить, что исследование ***методов создания автоматических тематических каталогов*** является многообещающим. Для решения данной проблемы требуется автоматизация работы распределения документов по тематическим каталогам путем определения тематики документа и нахождения тематических каталогов, имеющих наиболее релевантный профиль [29, 31, 36, 44, 103, 113, 114].

Поисковые системы, ориентированные на тематические каталоги имеют более высокие показатели точности, чем автоматические индексы. Однако полнота автоматических индексов обычно намного превышает полноту тематических каталогов. Автоматические индексы более гибкие и легко поддаются адаптации, т.е. программы-индексаторы (robots, spider и др.) периодически без особого труда могут обновлять базу индексов.

К настоящему времени созданы множества известных информационно-поисковых систем для Интернет, такие как Yahoo, Excite, AltaVista, Lycos, Stars, Infoseek, Rambler, Aport и т.д. Подключившись к ним пользователь имеет в своем распоряжении некоторый сайт, с помощью которого он может сформулировать свои запросы, вести поиск и получить URL-адреса (ссылки) Web-страниц, содержащие искомую информацию.

Некоторые из этих поисковых систем представляют тематические каталоги, содержащие разнообразные документы, Web-страницы, файлы и т.д., начиная от спорта, кино, музыки, игры до самых серьезных научных статей, книг, проектов. Другие системы являются автоматическими индексами и позволяют вести поиск по ключевым словам (терминам) документов разного содержания из разных сайтов всего мира. Эти системы различаются по видам представляемых услуг, охватываемых тематик и сайтов, а также по алгоритмам индексирования и каталогизации.

2.4. Модели информационного поиска

В зависимости от выбранных методов и средств представления и индексирования документов, а также поиска информации по этим документам различают несколько моделей информационного поиска, эффективность которых в целом определяется качеством разработанных методов и средств индексирования, поиска и т.д.

Прежде, чем перейти к описанию моделей информационного поиска отметим, что их в целом можно разбить на две части:

- модели поиска информации
- модели индексирования источников информации.

Существуют следующие модели и методы информационного поиска и индексирования:

- теоретико-информационные модели;
- вероятностные модели;
- векторная и линейная модель;
- статистические методы, модели индексирования по частоте вхождений;
- модель информационного поиска с лингвистическим обеспечением;
- теоретико-множественные модели;
- нечеткие модели.

Последняя модель рассматривается в следующей главе, а другие в последующих разделах данной главы.

Помимо вышеотмеченных методов и моделей существуют еще методы повышения эффективности поиска и индексирования:

- использование синонимов и тезаурусов;
- методы ассоциативного поиска;
- модели с обратной связью.

2.5. Теоретико-информационные модели

На базе данной модели лежит определение полезности, т.е. ценности терминов, входящих в документы информационного пространства [59, 128]. Здесь предполагается, что наибольшую информационную ценность представляет наименее предсказуемый термин. Наименее предсказуемыми терминами называются термины, вероятность вхождения в документы которых минимальна.

При индексировании для оценки полезности термина, используемого в качестве индекса документов, вычисляются специальная характеристика в виде отношения "сигнал – шум", для чего применяются методы и концепции теории информации.

Согласно данной модели предпочтение отдается таким терминам, которые сконцентрированы в отдельных документах. Таким образом, по основным свойствам этот подход напоминает статистические методы, использующие частотные характеристики документов для каждого термина.

Последнее время в работах западных ученых упоминаются разновидности теоретико-информационных моделей:

- классификационные информационные модели поиска;
- информационные модели для поддержки Web-основанного поиска.

Информационная модель классификаций включает в себя способы идентификации классификационных характеристик, например, названия объекта, тип и диапазон значений классификаций. При поиске, пользователи указывают типы классификации, значение соответствия искомых объектов к классификациям и т.д.

Использование методов информационного моделирования Web-пространства основывается на определении информационных потребностей доменов интересов, которое может использоваться для двух целей:

- для создания Web-основанных хранилищ комплексных информационных объектов;
- для конфигурирования Web-основанной поисковой системы для интеллектуального поиска в таком хранилище данных.

Чтобы можно было вести интеллектуальный поиск по хранилищу данных, необходимо, чтобы хранимые данные были характеризованы таким же образом, каким они будут использованы для поиска. По этой причине, система обеспечивает инструментальные средства, которые исследуют информационные объекты и профилируют их согласно критериям, определенным данной моделью.

2.6. Вероятностные модели поиска и индексирования

Вероятностные модели основаны на принципе ранжирования выдаваемых в результате поиска документов [75, 150]. Так, найденные документы можно упорядочить в соответствии с вероятностью их положительной оценки пользователем или их полезностью для него. Здесь допускаются следующие упрощения:

- релевантность каждого документа не зависит от других документов информационного массива;
- полезность каждого документа может зависеть от количества релевантных документов, уже просмотренных пользователем.

По мере увеличения количества просмотренных документов полезность остальных релевантных документов уменьшается.

В вероятностной модели учитываются все взаимозависимости и связи между терминами, определяются основные параметры поиска, такие как веса терминов запросов и форма соответствия "запрос - документ".

В данной модели каждому документу сопоставляется бинарный вектор, указывающий используемые и неиспользуемые для индексирования данного документа термины, которым может быть

приписано состояние релевантности или нерелевантности. Это определяется двумя главными параметрами:

- Prel - вероятность релевантности документа запросу пользователя;

- Pnrel - вероятность нерелевантности документа запросу пользователя.

Эти параметры вычисляются на основе вероятностных весовых коэффициентов терминов и фактического присутствия терминов в документе. Подразумевается, что релевантность является бинарным свойством, и поэтому $P^{rel}=1-P^{nrel}$.

Исходя из этого, в данной модели состояние документов описывается бинарным вектором d, определяющим какие термины были использованы и какие не использованы для индексирования данного документа, которому приписывается соответственно состояние релевантности s_1 или нерелевантности s_2.

Вероятность того, что документ d является релевантным или нерелевантным можно определять по формуле Бейеса следующим образом:

$$P(s_i / d) = \frac{P(d / s_i) \cdot P(s_i)}{P(d)}, \qquad (2.1)$$

где $P(s_i)$ - априорная вероятность релевантности (при i=1) или нерелевантности (при i=2) документа;

$P(d / s_i)$ - вероятность того, что определенный документ из числа выданных окажется релевантным или нерелевантным. По формуле полной вероятности:

$$P(d) = P(d / s_1) \cdot P(s_1) + P(d / s_2) \cdot P(s_2). \qquad (2.2)$$

Для определения величин $P(d/s_i)$ и введения поисковой функции используется взвешивание терминов запроса на основе следующих предположений:

- термины для индексирования документов используются независимо;

- вероятность релевантности документа запроса оценивается исходя из наличия или отсутствия терминов в поисковых образах.

На основе этих предположений вводится весовая функция:

$$w_i = log\frac{r_i \cdot (N - n_i - R - r_i)}{(R - r_i)\cdot(n_i - r_i)}, \qquad (2.3)$$

где N – количество документов в массиве;

R – количество документов, релевантных запросу;

n_i – количество документов, заиндексированных термином t_i;

r_i – количество релевантных документов, заиндексированных термином t_i. Для очень малых или нулевых значений r_i, R и n_i предлагается заменять r_i на $(r_i + 0.5)$.

Для определения величин R и r используются предварительные оценки, уточняемые по результатам обратной связи с пользователями. Кроме того, могут использоваться результаты, полученные на части массива. В случае отсутствия обратной связи предлагается проводить оценку, считая релевантными документы, которым автоматически системой приписываются наибольшие значение вероятности релевантности.

2.7. Векторная и линейная модель

В векторной модели (ее иногда называют алгебраической

моделью) документы информационного пространства представляются набором векторов в этом пространстве, которое определяется базисом из *n* нормализованных векторов терминов. В этом пространстве каждый документ будет представлен *n*-мерным вектором. Значение первого элемента данного вектора отражает вес термина в документе, соответствующем первому измерению векторного пространства, и т.д.

В векторной модели подразумевается, что векторы терминов, на которые натянуто пространство, ортогональны и существующие взаимосвязи между терминами не должны приниматься во внимание [26, 79, 100, 128, 164]. Обычно в векторной модели информационного поиска выделяют следующие основные понятия:

- **Словарь**. Под словарем понимают множество терминов, мощность которого обозначают как *T*.

- **Документ** – это двоичный вектор d_i размерности *L*. Если термин входит в документ, то в соответствующем разряде этого двоичного вектора представляется 1, в противном же случае – 0. Обычно в линейной модели индексирования все операции поиска документов выполняются над поисковыми образами документов, но при этом их, как правило, называют просто документами.

- **Информационный поток или массив** *D*, представляет собой матрицу размерности *N*L*, где в качестве строк выступают поисковые образы *N*-документов, т.е.

$$D = \{d_1, d_2, ..., d_N\}.$$

- **Запрос пользователя** также представляется вектором размерности *N*. Другими словами,

$$Q = \{q_1, q_2, ..., q_N\}$$

- **Процедура поиска**. Показатель RSV, определяющий соответствие документа запросу, задается скалярным произведением

векторов запроса и документа. Чем выше RSV, тем больше документ релевантен запросу. При таком рассмотрении процедуру поиска можно сформулировать следующим образом:

$$R = Q \cdot D, \qquad (2.4)$$

где Q - вектор запроса, R - отклик (ответ) системы на запрос.

- **Коррекция запроса по релевантности.** Многие системы применяют механизм коррекции запроса по релевантности. Это означает, что процедура поиска носит интерактивный и итеративный характер. После проведения первичного поиска пользователь отмечает из всего списка найденных документов релевантные. На следующей итерации система расширяет (или уточняет) запрос пользователя терминами из этих документов и снова выполняет поиск. Так продолжается до тех пор, пока пользователь не сочтет, что лучшего результата, чем он уже имеет, добиться не удастся. Коррекция запроса по релевантности - это достаточно широко внедренный способ уточнения запросов. В некоторых системах пользователь может и не знать, о том, что эта процедура применяется, например, OpenText. В этом случае несколько итераций выполняется без его вмешательства.

Преимущество векторной модели в основном заключается в простоте ее реализации. Она позволяет также легко реализовать обратную связь с пользователем для оценки релевантности выданных документов. Однако, в отличии от булевой модели при оформлении запросов невозможно достичь той выразительности и спецификации запроса. Кроме того, в такой модели не специфицируется степень соответствия "запрос - документ", и она оценивается достаточно произвольно.

2.8. Модель индексирования по частоте вхождений терминов

Пусть $D = \{d_1, d_2, ..., d_n\}$ – множество документов в информационном массиве, n – количество документов в информационном массиве, $T = \{t_1, t_2, ..., t_m\}$ - множество терминов, наиболее полно описывающих содержания документов или определяющих их тематическую принадлежность, $W = \{ w_{ij} \}_{n \times m}$ - матрица отношений между терминами и документами. Значение элементов w_{ij} определяет вес термина t_j в документе d_i с учетом всех документов информационного массива. Здесь w_{ij} получает значение в интервале $[0,1]$, где $w_{ij} = 0$ - означает, что термин t_j не встречается в документе d_i, а $w_{ij} = 1$ - термин t_j стопроцентно соответствует содержанию документа d_i.

Весовые коэффициенты терминов w_{ij} используются для определения степени релевантности документов запросу пользователя, который задается пользователем в виде набора терминов. Как уже было сказано, документы также представляются набором терминов из множества *T*. Поэтому согласно данной модели ведется поиск документов, в которые с наибольшими весовыми коэффициентами входят термины запроса пользователя [37, 77, 100].

Теперь рассмотрим статистический метод определения значений весовых коэффициентов w_{ij}, который состоит из двух этапов. На первом этапе вычисляется частота вхождения терминов в отдельные документы, а на втором этапе анализируются все документы информационного массива и определяется вес данного термина с

учетом характеристик встречаемости его в других документах массива.

Пусть f_{ij} - частота вхождения термина t_j в документ d_i, тогда:

$$f_{ij} = \frac{m_{ij}^t}{m_i^w}. \qquad (2.5)$$

Здесь m_{ij}^t - количество вхождения термина t_j в документ d_i, m_i^w - общее количество слов в документе d_i. f_{ij} является весовым коэффициентом термина t_j в документе d_i без учета характеристик остальных документов. Отсюда видно, что любое наиболее часто встречаемое слово в документе можно принять как важный термин, наиболее точно отражающий его содержание.

Однако, практика показывает, что слова, с наибольшим коэффициентом встречаемости в документе не всегда являются важными терминами для данного документа. В качестве примера можно указать служебные слова, связки, предлоги, местоимения и т.д.

Согласно результатам исследований контекстного анализа содержания документов, в т.ч. закону Ципфа можно сказать, что если какое-то слово имеет большую частоту в документе и не является термином, то оно должно встречаться и во многих других документах, возможно даже с большой частотой. Анализируя соотношение n - общего количества документов в информационном пространстве и n_j^d - количества документов, в которых встречается термин t_j, можно сделать вывод о том, что является ли данное слово значимым термином для данного документа или нет. n_j^d называют документной частотой.

Отметим, что чем меньше значение n_j^d, тем больше вес можно приписать к термину в документе. Величина $log\frac{n}{n_j^d}$ может служить хорошим индексатором того, что является ли термин t_j дискриминатором документов или нет, т.е. позволяет отличить документы с значимыми терминами от тех, где эти термины не встречаются.

Данная величина называется обратной документной частотой, которая обозначается через

$$f_j^d = log\frac{n}{n_j^d}. \qquad (2.6)$$

Величина частоты термина в документе и обратной документной частоты можно объединить в рамках единой модели индексирования по частоте:

$$w_{ij} = f_{ij} \cdot f_j^d. \qquad (2.7)$$

Учитывая (2.5) и (2.6) в (2.7), получим выражение весового коэффициента w_{ij} термина t_j в документе d_i с учетом отношений всех других документов информационного пространства:

$$w_{ij} = \frac{m_{ij}^t}{m_i^w} \cdot log\frac{n}{n_j^d}. \qquad (2.8)$$

Из (2.8) видно, что чем больше частота термина t_j в документе d_i и меньше количество документов, содержащих термин t_j, тем больше значение получит вес термина t_j в документе d_i. Другими словами, если d_i является документом, в котором сосредотачивается

термин t_j, то скорее всего термин t_j окажется значимым термином для данного документа.

Отметим, что индексирование на основе частоты вхождения терминов в документы позволяет в основном улучшить одну из характеристик, т. е. ***полноту поиска***. Однако фактор концентрации терминов в отдельных документах массива может быть использовано также для достижения высокой точности поиска

2.9. Модель информационного поиска с лингвистическим обеспечением

В лингвистической модели лексическое индексирование предназначено для оптимизации булевых запросов [31, 66]. Так присваивание индексаторов синтаксических классов, таких как прилагательное, существительное или глагол, может повысить качество используемого в системе статистического метода. В этом случае формирование фраз ограничивается предложениями с заданными синтаксическими индикаторами (например, существительное-существительное или прилагательное-существительное).

Для идентификации синтаксических единиц можно использовать простой процесс синтаксического анализа. Обычно элементы фраз выбираются в рамках этих же синтаксических единиц.

Лексическое индексирование является наиболее старым и традиционным методом. За время своего существования оно почти достигло совершенства в области поиска по ключевым словам и целым фразам. Развитые поисковые лексические системы используют стоп-листы, грамматические словоформы и расширенный язык

запросов. Используются возможности задавать близость данных слов в тексте (проксимити) и близость слова к началу текста. Во многих системах достигается интеллектуальность за счет вручную построенных тезаурусов. Хотя к настоящему времени разработаны различные методы автоматического построения тезауруса, их эффективность вне пределов той специальной среды, в которой они сгенерированы, все еще остается под вопросом. Определенно можно сказать, что возможности современных систем лексического поиска вполне удовлетворяют потребности экспертов, интересующихся информацией в какой-нибудь узкой области.

Однако, для большинства пользователей поисковых систем Web эффективность лексического поиска выглядит крайне неудовлетворительной. Одной из основных причин заключается в следующем: обычному пользователю, в отличие от “эксперта”, трудно сформулировать свои запросы на языке ключевых слов предметной области, особенно если в этой предметной области нет устоявшейся или регламентированной терминологии. Кроме того, при попытке выйти за рамки узкой предметной области сразу остро встают проблемы синонимии и полисемии (одинаковые слова с разным смыслом).

2.10. Теоретико-множественные модели

В теоретико-множественных моделях для описания документов и запросов пользователей используются булевы переменные. По этой причине эти модели называют ***булевыми моделями*** [59, 79, 128]. Так в булевой модели документы представляются с помощью набора терминов, присутствующих в индексе, каждый из которых

рассматривается как булева переменная. Если какой-то термин встречается в документе, то соответствующая логическая переменная принимает значение **True**, в обратном случае **False**. Согласно данной модели присваивание терминам весовых коэффициентов не допускается.

Пользователи формулируют свои запросы в виде произвольных булевых выражений. Термины, входящие в запросы связываются с помощью стандартных логических операций, таких как AND, OR или NOT. После получения запроса поисковая система для каждого документа вычисляет ***значения статуса выборки*** - retrial status value (переменная RSV), которые определяют меру соответствия данного запроса документам пространства.

Оператор AND может очень сильно сократить число документов, которые выдаются в ответ на запрос. При этом все будет очень сильно зависеть от того, что насколько типичными для базы данных являются поисковые термины. Оператор OR напротив может привести к неоправданно широкому запросу, в котором полезная информация затеряется за информационным шумом.

Для успешного применения такого ИПЯ следует хорошо знать лексику системы и ее тематическую направленность. Как правило, для системы с таким ИПЯ создаются специальные документально лексические базы данных со сложными словарями, которые называются тезаурусами и содержат информацию о связи терминов словаря друг с другом.

В данной модели переменная RSV получает значение либо 1, либо 0. Если результат вычисления выражения для данного запроса дает True, то переменная RSV получает значение 1. Документы, для которых RSV=1 считаются релевантными данному запросу. Если

результат вычисления выражения является FALSE, т.е. RSV=0, то документ принимается как нерелевантным.

Отметим, что булева модель проста в реализации, в настоящий момент ее применяют в поисковых средствах многих коммерческих сетей. Благодаря логическим операциям, данная модель позволяет пользователям формулировать свои запросы в виде произвольных сложных выражений.

Также следует отметить, что эффективность поиска в данной модели обычно бывает невысокой. Так как терминам не присваиваются весовые коэффициенты и переменная RSV для всех найденных документов получает одинаковые значения, то результаты поиска невозможно ранжировать.

Для повышения эффективности поиска часто применяется метод обратной связи с пользователем. Когда ИПС выдает пользователю список найденных документов, то система просит его указать релевантность или нерелевантность нескольких первых документов в начале списка.

Модификацией булевого поиска является *взвешенный булевый поиск*. Идея такого поиска достаточно проста. Считается, что термин описывает содержание документа с какой-то точностью, и эту точность выражают в виде веса термина. При этом взвешивать можно как термины документа, так и термины запроса. Запрос может формулироваться на ИПЯ, описанном выше, но выдача документов при этом будет ранжироваться в зависимости от степени близости запроса и документа. При этом измерение близости строится таким образом, чтобы обычный булевый поиск был бы частным случаем взвешенного булевого поиска.

2.11. Методы улучшения эффективности поиска

Эффективность поиска можно оценить опираясь на качественные и количественные характеристики найденных документов. ***Качественные характеристики*** измеряются уровнем соответствия выданных релевантных документов запросу, а ***количественные*** - соотношением выданных и не выданных релевантных документов.

Для определения качественных характеристик необходимо проводить анализ результатов поиска, степени удовлетворения пользователя - "заказчика". Количественные характеристики вычисляются с помощью статистических данных. Однако, в обоих случаях встречаются определенные трудности. Так в первом случае требуется обратная связь с пользователем, которую не всегда удается осуществлять на желаемом уровне, а во втором случае - определение количества не выданных релевантных документов информационного массива сервера.

Для оценки эффективности ИПС необходимо качественное измерение основных параметров поиска - точности и полноты. Это можно осуществлять с помощью такой системы, в которой количество документов фиксировано, а также имеются стандартный набор запросов и множества документов, релевантных и нерелевантных каждому обрабатываемому запросу.

На практике создавать подобные условия для информационного пространства Интернет, в том числе Web среды очень трудно. Реально существующие поисковые серверы работают не только со своими документами, а также с индексами информационных ресурсов различных сайтов.

Несмотря на то, что при оценки эффективности встречаются

определенные трудности, существуют различные методы улучшения эффективности поиска в целом, которые основываются на различные подходы. Выбор и применение метода зависит от требований пользователя в конкретных случаях [27, 76, 100].

В целом, методы повышения эффективности поиска можно разделить на методы улучшения показателей полноты и точности.

2.11.1. Методы улучшения полноты поиска

Методы, повышающие полноту поиска следует применять тогда, когда пользователь хочет получить всевозможное множество документов, относящиеся в какой-то мере к его запросу. Например, примером желания получить такой исчерпывающий ответ является поиск экспертом всех патентов, имеющихся в интересующей его области.

Методы, применяемые для улучшения показателя полноты поиска позволяют добиться дополнительных совпадений терминов запроса пользователя и документа информационного массива. С этой целью существующие термины запросов и документов заменяются другими терминами или к ним добавляются новые, другими словами множество терминов запроса расширяется.

Существуют несколько методов расширения множества терминов:

- использование словаря синонимов или тезауруса;
- ассоциативный поиск;
- вероятностное индексирование;
- использование библиографических данных.

Наиболее известным из вышеперечисленных является метод

использования словаря синонимов или тезауруса. Эти словари состоят из множества классов синонимии или эквивалентности, а в каждом таком классе группируются термины - синонимы, тезаурусы, а также слова, описывающие одни и те же объекты или действия, являющиеся очень близкими по смыслу, относящие к одной тематике и т.д.

Использование таких словарей во время инициализации процесса поиска позволяет заменять термин запроса на идентификатор класса синонимов, что фактически заменяет данный термин на множества терминов, входящих в этот класс.

Следующим методом получения дополнительных совпадений терминов является ***использование ассоциативных слов***. Для каждого термина запросов и документов определяются набор дополнительных терминов, которые ассоциируются с исходным термином.

Для определения показателя ассоциируемости терминов можно использовать методы индексирования, например, статистический метод. Согласно таким методам для множества терминов составляется некоторая матрица ассоциируемости: $A=\{a_{ij}\}_{m\times n}$, элементы которой определяют значения показателя ассоциируемости для каждой пары терминов соответствующие строке и столбцу.

Коэффициент ассоциируемости терминов i и j вычисляется как сумма произведения частот этих терминов по всем документам массива:

$$a_{ij} = \sum_{k=1}^{n} f_{ik} \cdot f_{jk}\,, \qquad (2.9)$$

где a_{ij} - коэффициент ассоциируемости терминов i и j, f_{ik} - частота встречаемости термина i в документе k, n - количество документов в массиве.

Как было отмечено выше, частоты терминов f_{ik} и f_{jk} получают значения в интервале [0,1]. Однако согласно формуле (2.9) значение показателя ассоциируемости a_{ij} может оказаться намного больше, поэтому требуется их нормализация, для чего можно применять следующую формулу:

$$f_{ij} = \frac{\sum_{k=1}^{n} f_{ik} \cdot f_{jk}}{\sum_{k=1}^{n} (f_{ik})^2 + \sum_{k=1}^{n} (f_{jk})^2 - \sum_{k=1}^{n} f_{ik} \cdot f_{jk}} \qquad (2.10)$$

Для значений показателя ассоциируемости терминов устанавливают пороговое значение δ. Если $f_{ij} \geq \delta$, то предполагают, что термин *i* ассоциируется с термином *j*.

При ***вероятностном индексировании*** определяется наличие определенных терминов в документах, после чего на основе этих отношений документам приписываются идентификаторы тематических классов, т.е. документы включаются в тематические классы, содержащие данные термины.

И наконец, **методы использования библиографических данных** позволяют увеличить количество ключевых слов и, следовательно, расширять круга охватываемых документов в результате поиска. В качестве библиографических данных используются такие реквизиты документов, как фамилии авторов, названия изданий, ссылки, цитаты и.т.д., которые приписываются к данным документам в качестве ключевых слов.

2.11.2. Методы улучшения точности поиска

Методы улучшения показателя точности служат для сужения

множества выданных релевантных документов пользователю, путем отсечения менее подходящих, случайных и ненужных, таким образом, оставив в списке наиболее релевантные из них.

Как отметили выше, улучшение полноты поиска достигается путем расширения множества терминов, используемых при описании как запроса, так и документа, а также замены этих терминов на другие -родственные. В отличии от полноты, другой не менее важный параметр поиска - точность можно улучшить путем применения очень узких терминов или нескольких терминов в комбинации, т.е. словосочетаний.

Для выделения узких терминов и словосочетаний используются специальные методы. Однако эти методы имеют свои достоинства и недостатки.

Суть ***статистического метода*** образования словосочетаний или комбинаций терминов заключается в следующем. Как известно, словосочетания являются множествами терминов. Предполагается, что частота появления словосочетаний, т.е. совместного появления терминов в массиве намного превышает частоту этих терминов по отдельности. Тогда степень связанности двух терминов можно определять как

$$C^{ij} = \frac{F^{ij}}{F^{i} \cdot F^{j}} \cdot N, \qquad (2.11)$$

где F^{ij} - частота совместного появления пары терминов i и j, а F^{i} и F^{j} - соответственно частоты терминов i и j в массиве, N - общее количество терминов или слов в этом массиве.

Для повышения эффективности образования словосочетания из текстов документов массива сначала следует исключить все служебные слова, потом из оставшихся слов выбирать комбинации

терминов с высокой частотой совместного появления и коэффициентом связанности. При этом вводят некоторые пороговые значения, где предполагают, что все комбинации терминов, частота совместного появления F^{ij} и степень связанности C^{ij} которых выше, чем заданный порог являются словосочетаниями.

Статистические методы построения словосочетаний не всегда являются пригодными, так как они приводят к чрезмерному сужению содержания документа и потере полноты.

Синтаксические и семантические методы, основаны на исследовании грамматики составляющих словосочетаний и анализа лингвистических структур текста. Эти методы привязаны к языкам, так как синтаксические и семантические свойства текста непосредственно зависят от особенностей языка. Несмотря на то, что синтаксические и семантические методы мало развиты, в настоящее время существуют такие полнотекстовые поисковые системы, в которых эти методы реализованы.

Эти методы образования словосочетаний оказываются малоэффективными или очень сложными для реализации и практического использования.

2.11.3. Использование обратной связи с пользователем для повышения релевантности

Обычно в поисковых системах документы представляются нечетко, поэтому пользователи начинают вести поиск с неточного и неполного запроса, который, естественно, приводит к нежелаемым результатам. Но, далее в последующих этапах работы пользователи постепенно могут итеративно повысить эффективность и таким

образом улучшить результат поиска путем уточнения своего запроса.

Для этого в поисковых системах реализуют обратную связь с пользователем, что позволяет ему оценивать степень релевантности найденных системой документов по первоначальному запросу и передать свое мнение системе. Это осуществляется на основе двух- или многоуровневых отношений.

В двухуровневых отношениях пользователь может указывать один из двух вариантов: документы, выданные системой, являются релевантными или нерелевантными запросу. А при использовании многоуровневых отношений, кроме этих двух отношений, еще может быть указано отношение частичной релевантности документов. Для упрощения реализации методов обратной связи обычно используют двухуровневую релевантность.

Существуют два метода обратной связи:

- изменение представления запроса;

- изменение представления документов.

Методы изменения представления запроса предназначены для повышения эффективности поиска при текущем сеансе и не влияют на другие запросы. А ***методы изменения представления документов*** модифицируют пространства представления документов и поэтому улучшают результат поиска не только на данном сеансе, а также на последующих сеансах поиска.

Существуют три способа повышения эффективности поиска с помощью методов изменения представления запросов:

- метод модификации весов терминов;

- метод расширения запроса;

- метод расщепления запроса.

Метод модификации весов терминов осуществляют улучшения

весов терминов в запросе. Для того, чтобы корректировать веса терминов, необходимо сложить вектор запроса и векторы документов с положительной обратной связью. Также можно корректировать весовые коэффициенты терминов путем вычитания векторов документов с отрицательной обратной связью из вектора запроса.

Отметим, что ***документами с положительной обратной связью*** называются документы, отмеченные пользователем как релевантные своему заданному запросу, а ***документы с отрицательной обратной связью*** - отмеченные как нерелевантные.

После корректировки запроса количество релевантных документов, выданных поисковой системой, должно быть больше, чем количество до корректировки. Кроме предыдущих релевантных документов в это множество включаются ещё дополнительные документы, являющиеся подобными выданным ранее релевантным документам, которые с помощью обратной связи пользователем отмечаются положительно.

Пользователь может продолжить данный процесс итеративно до тех пор, пока качество и число выданных релевантных документов не достигнут желаемого уровня.

Метод расширения запроса расширяет исходный запрос путем добавления к нему дополнительных терминов. Сначала из документов с положительной обратной связью выбирается множество терминов. Затем после сортировки из начала списка данного множества выделяется заранее заданное число терминов и добавляется к запросу.

Метод расщепления запроса применяется в тех случаях, когда результат поиска содержит неоднородные документы с положительной обратной связью. В таких случаях выданное множество документов разбивают на несколько подмножества

(кластера) однородных документов. Если это возможно, то запрос пользователя разбивается на подзапросы таким образом, чтобы каждый подзапрос представлял одно подмножество. Для каждого подзапроса по отдельности улучшаются весовые коэффициенты соответствующих терминов или этот подзапрос расширяется с помощью предыдущих методов.

Модификация представления документов на основе обратной связи позволяет улучшить векторы представления документов. При этом весовые коэффициенты терминов корректируются таким образом, чтобы векторы найденных релевантных документов приближались к вектору запроса, а векторы нерелевантных документов отдалялись от этого запроса.

После такой корректировки векторов представления документов, в последующих аналогичных запросах система выдает более релевантные документы, чем в предыдущих.

III ГЛАВА.

МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННОГО ПОИСКА В ИНТЕРНЕТ НА БАЗЕ НЕЧЕТКИХ ЗНАНИЙ

3.1. Нечеткая модель информационного поиска

Для построения эффективных ИПС необходимо изучить информационную среду поиска в целом, разработать методы тематического разбиения, индексирования, поиска и представления документов и запросов. Модель поисковой системы должна включить в себя как необходимые множества документов, так и отношения между ними.

Здесь рассматривается модель информационного поиска, основанная на теории нечетких множеств и отношений [4, 68, 74, 106, 109, 168]. Такая модель позволяет автоматически создавать тематические каталоги без вмешательства человека и улучшить качества существующих тематических каталогов с применением синонимов и ассоциирующих терминов, а также найти документы и тематические каталоги, наиболее релевантные запросам пользователей. С этой целью в данной модели используется аппарат нечетких множеств и подход Беллмана-Заде [34, 39, 40, 41, 43].

Пусть $D = \{d_i\}_I$ – множество документов информационного пространства Интернет, которое требуется разбить на подмножества (тематические каталоги) по тематике, аналогично библиотечно-информационной системе. Множество классификационных направлений, которые назовем тематическими каталогами, определяемыми своими тематическими профилями, обозначим через

$K = \{k_l\}_L$. Каждый такой каталог в соответствии с его тематическим профилем характеризуется собственными своими дескрипторами, ключевыми словами или другими лексическими единицами, которые называются терминами. Множество терминов, характеризующее тематическое направление профиля K_l, обозначим через T_l^K. Необходимо отметить, что эти множества могут частично пересекаться, т.е.:

$$T_i^K \cap T_j^K \neq \varnothing, \text{ для } \forall\ i \neq j. \qquad (3.1)$$

Совокупность множеств терминов всех тематических профилей поисковой системы составляет множество терминов этой системы. Обозначив множество терминов поисковой системы через $T = \{t_j\}_J$, получаем

$$T = \bigcup_{l=1}^{L} T_l^K. \qquad (3.2)$$

В качестве множества терминов можно использовать любой универсальный библиографический классификатор, такой как УДК, ББК и т.д. Возможен другой подход, лежащий на основе многих современных ИПС Интернет, суть которого заключается в создании и дополнении множества терминов самой информационно-поисковой системой или ее администратором, как база индексов или метаданных, в процессе функционирования системы.

Введем еще одно множество - множество синонимов и ассоциирующих слов терминов $t_j \in T$, $j = \overline{1,J}$, которое представляется в виде нечеткого отношения. Для простоты будем предполагать, что синонимы и ассоциирующие слова терминов составляют одно множество, которое обозначим через $S = \{s_v\}_V$.

Учитывая вышесказанное, информационный поиск можно представить в виде

$$I_R = \{ K, D, T, Q, R \}, \qquad (3.3)$$

где I_R является результатом информационного поиска, выдаваемым пользователю в виде вектора (списка) названий, адресов и других реквизитов источников информации, Q - запрос пользователей, который рассматривается ниже. $R = \{R^D, R^K, R^S, R^Q\}$ - множество отношений, которое определяет отношения типа “документ – термин” (R^D), “тематический каталог – термин” (R^K), “термин - синоним” (R^S) и “запрос – термин” (R^Q). Необходимо отметить, что все эти отношения, определяемые здесь и рассматриваемые далее являются нечеткие.

На рисунке 3.1 информационный поиск представлен в виде

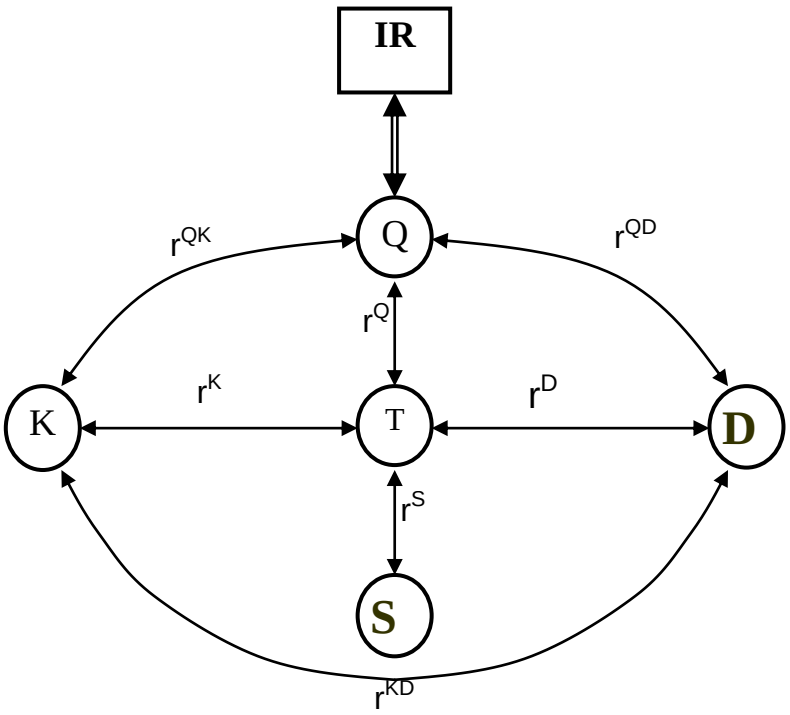

Рис.3.1. Представление информационного поиска в виде ориентированного графа

направленного графа, вершинами которого являются выше описанные множества, а дугами представляются отношения между этими множествами. В данный граф включены все множества и отношения информационного поиска.

Отношение между множествами тематических профилей и терминов называется ***тематическим рубрикатором***. Как было сказано выше, каждый тематический профиль также определяется множеством терминов, где одни термины могут быть релевантными одному тематическому профилю, другие термины - нескольким тематическим профилям. Тематический рубрикатор представляется в виде нечеткой реляционной матрицы размерности *LxJ*, строки которой соответствуют тематическим профилям, а столбцы - терминам.

Таким образом, тематические рубрикаторы можно представить в виде:

$$K_l = \{t_j / \varphi_{t_j}(K_l)\},\ l = \overline{1,L},\ j = \overline{1,J},\ , \qquad (3.4)$$

где $\varphi_{t_j}(K_l)$ - функция принадлежности термина к тематическому профилю, т.е. степени релевантности (принадлежности), термина t_j тематическому каталогу K_l. Здесь R^K - отношение между множеством тематических профилей и множеством терминов определяется как:

$$r_l^K = \left\{\varphi_{t_j}(K_l) : T \times K \to [0,1]\right\},\ l = \overline{1,L},\ j = \overline{1,J},$$

$$R^K = \left\{r_l^K\right\}_L. \qquad (3.5)$$

Значения элементов матриц отношений R^S (рассматривается далее) и R^K определяются на начальном этапе, при создании ИПС, для чего можно применять так называемый метод экспертных оценок.

В процессе функционирования системы, значения отношений $\varphi_{t_j}(K_l)$ подвергаются адаптации, т.е. обучению.

Теперь рассмотрим, как проиндексированные документы можно представлять в базе поисковой системы. Как отметили выше, в результате индексирования каждому документу информационного пространства сопоставляются некоторые термины из множества *T*. Терминам приписываются весовые коэффициенты, определяющие степень важности этих терминов для каждого документа.

Множество терминов (ключевых слов) документа d_i обозначим через T_i^D, которое является нечетким множеством. Весовые коэффициенты терминов относительно каждого документа определяются функцией принадлежности термина в множество T_i^D, которая получает значения в пределе [0,1]. Все множества T_i^D нормируются, т.е. дополняются терминами множества *T*, отсутствующими в множестве T_i^D, функция принадлежности которых в T_i^D получает нулевые значения.

Таким образом, множество документов *D* можно представить в виде нечеткой реляционной матрицы размерности *IxJ*. Элемент, находящийся на пересечении *i*-ой строки и *j*-го столбца определяет вес термина t_j для документа d_i, т.е.:

$$d_i = \left\{ t_j / \omega_{t_j}(d_i) \right\},\ i = \overline{1,I}\ ,\ j = \overline{1,J}\ , \qquad (3.6)$$

а отношение между элементами d_i и t_j определяется следующим образом:

$$R_i^D = \left\{ \omega_{t_j}(d_i) : D \times T \rightarrow [0,1] \right\},\ i = \overline{1,I},\ j = \overline{1,J}\ ,$$

$$R^D = \left\{ R_i^D \right\}_I , \qquad (3.7)$$

где $\omega_{t_j}(d_i)$ - функция принадлежности термина t_j в множество терминов T_i^D, значения которой определяются в результате индексирования документов. Множество отношений R^D часто обозначают через *W*.

3.2. Разбиение информационного пространства на тематические каталоги по профилям документов

Теперь рассмотрим задачу автоматического распределения документов Интернет по тематическим направлениям, т.е. автоматического разбиения их на тематические каталоги [36, 41, 44]. Другими словами данная задача является задачей автоматического создания тематических каталогов.

Если T_i^D - множество терминов документа d_i и T тематический рубрикатор (3.2), тогда релевантность определяется в виде пересечения этих множеств:

$$T \cap T_i^D = \left(\bigcup_{l=1}^{L} T_l^K \right) \cap T_i^D = \bigcup_{l=1}^{L} \left(T_l^K \cap T_i^D \right). \qquad (3.8)$$

Для выявления наиболее релевантного тематического каталога для ресурса d_i можно использовать подход Беллмана-Заде.

На первом этапе определяются степени релевантности документа d_i каждому тематическому каталогу множества *K* по отношению всех терминов данного документа. Как отметили выше, документ d_i в

системе представляется отношением R_i^D данного документа к терминам множества T:

$$R_i^D = \{ \omega_{i1}, \omega_{i2}, ..., \omega_{iJ} \}, i = \overline{1,I} . \quad (3.9)$$

где $\omega_{ij} = \omega_{t_j}(d_i)$.

Аналогично представляются тематические каталоги:

$$R_l^K = \{ \varphi_{l1}, \varphi_{l2}, ..., \varphi_{lJ} \}, \; l = \overline{1,L} , \quad (3.10)$$

где $\varphi_{ij} = \varphi_{t_j}(K_l)$.

Тогда релевантность документа d_i тематическому каталогу K_l можно рассмотреть как пересечение множеств отношений R_i^D и R_l^K, т.е.:

$$R_{il}^{DK} = R_i^D \cap R_l^K . \quad (3.11)$$

Как известно, пересечение нечетких множеств определяется как алгебраическое произведение соответствующих элементов этих множеств, т.е.:

$$\eta_{ij}^l = \omega_{ij} \cdot \varphi_{lj} , \quad (3.12)$$

где $\eta_{ij}^l \in R_{il}^{DK}$, $l = \overline{1,L}$, $j = \overline{1,J}$.

Здесь η_{ij}^l - степень релевантности тематики документа d_i к профилю тематического каталога K_l по отношению термина t_j.

Далее определим наиболее предпочтительный (недоминируемый) тематический каталог. Пусть K^{ab} - абстрактный тематический каталог, объединяющий в себе все наилучшие отношения релевантности документа d_i ко всем профилям тематического каталога

по всем его терминам. K^{ab} можно определять путем объединения всех нечетких множеств R_{il}^{DK} для всех $l=\overline{1,L}$, т.е. нахождения максимумов среди η_{ij}^{l} по всем терминам для всех каталогов:

$$\eta_{ij}^{ab} = \max_{l=1,L}\left\{\eta_{ij}^{l}\right\},\ j=\overline{1,J}\ , \qquad (3.13)$$

где η_{ij}^{ab} - степень релевантности d_i к K^{ab} по отношению термина t_j. Исходя из этого, можно сказать, что K^{ab} представляет собой абстрактный каталог, являющийся наиболее релевантным документу d_i.

Теперь найдем каталог K_l^* из множества $\{K_l\}_L$, тематический профиль которого является наиболее близким профилю абстрактного каталога K^{ab} относительно всех терминов t_j документа d_i. С этой целью вычислим суммарное среднеквадратическое отклонение коэффициентов релевантности всех тематических каталогов K_l от коэффициентов релевантности K^{ab} по следующей формуле:

$$\lambda_{il} = \frac{\overline{J}}{J}\sum_{j=1}^{J}\left(\eta_{ij}^{ab}-\eta_{ij}^{l}\right)^2, l=\overline{1,L}, \qquad (3.14)$$

где $\overline{J}$ - количество терминов документа d_i, весовые коэффициенты которых получают не нулевые значения для тематического каталога K_l.

Отсюда видно, что тематический каталог, имеющий минимальное среднеквадратическое отклонение является наиболее предпочтительным тематическим каталогом для данного документа:

$$\lambda_i^* = \min_{l=\overline{1,L}} \{\lambda_{il}\}. \qquad (3.15)$$

Таким образом, тематический каталог K_l^*, имеющий минимальное отклонение λ_i^* является наилучшим из множества $\{K_l\}_L$, т.е. профиль K_l^* наиболее релевантный профилю K^{ab} и, соответственно, наиболее подходящий тематике документа. Отсюда следует, что данный ресурс необходимо включить в тематический каталог K_l^*.

Исходя из (3.1), можно предположить, что каждый документ может оказаться релевантным не только к одному тематическому каталогу, а к нескольким. Для определения всех наиболее подходящих каталогов введем пороговые значения $\delta_l, l = \overline{1,L}$ для степени релевантности тематических каталогов. Таким образом, вытекает следующая задача: определять все наиболее предпочтительные тематические каталоги, степень релевантности которых не ниже пороговых значений.

Для определения наиболее предпочтительных каталогов необходимо из (3.13) найти все тематические каталоги, для которых отклонение тематики λ_{il} не превышает пороговое значение, т.е. удовлетворяют условию:

$$\lambda_{il} \le \delta_l, l = \overline{1,L}. \qquad (3.16)$$

Для наглядной иллюстрации предложенного метода и доказательства адекватности рассмотрим пример.

Пример 1. Пусть множество терминов состоит из десяти ($J=10$) терминов, а тематический рубрикатор включает в себя четыре каталога ($L=4$), т.е.:

$$T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}\}$$

и

$$K = \{K_1, K_2, K_3, K_4\}\ .$$

Отношение каталогов K_l к терминам t_j задано нечеткой реляционной таблицей R_l^K:

$$K_1(t_j / \varphi_{1j}) = \{t_1 / 0.75; t_2 / 0; t_3 / 0.63; t_4 / 0;$$
$$t_5 / 0.95; t_6 / 0.82; t_7 / 0; t_8 / 0; t_9 / 0.78; t_{10} / 0\},$$
$$K_2(t_j / \varphi_{2j}) = \{t_1 / 0; t_2 / 0.8; t_3 / 0.79; t_4 / 0.65;$$
$$t_5 / 0.6; t_6 / 0; t_7 / 0.9; t_8 / 0; t_9 / 0.69; t_{10} / 0.78\},$$
$$K_3(t_j / \varphi_{3j}) = \{t_1 / 0.56; t_2 / 0.83; t_3 / 0; t_4 / 0;$$
$$t_5 / 0.64; t_6 / 0; t_7 / 0.76; t_8 / 0.95; t_9 / 0; t_{10} / 0.69\},$$
$$K_4(t_j / \varphi_{4j}) = \{t_1 / 0; t_2 / 0.78; t_3 / 0.68; t_4 / 0.54;$$
$$t_5 / 0.8; t_6 / 0.2; t_7 / 0.86; t_8 / 0; t_9 / 0.71; t_{10} / 0.9\}.$$

Допустим, поисковая система проиндексировала документ d_1 и выявила следующие коэффициенты важности терминов для данного документа:

$$d_1(t_j / \omega_{1j}) = \{t_1 / 0; t_2 / 0.9; t_3 / 0.71; t_4 / 0.87;$$
$$t_5 / 0.54; t_6 / 0; t_7 / 0.6; t_8 / 0; t_9 / 0.95; t_{10} / 0.58\}.$$

Тематические каталоги и документ можно представить в табличном виде:

$$R_l^K = \begin{pmatrix} 0.75 & 0 & 0.63 & 0 & 0.95 & 0.82 & 0 & 0 & 0.78 & 0 \\ 0 & 0.8 & 0.79 & 0.65 & 0.6 & 0 & 0.9 & 0 & 0.69 & 0.78 \\ 0.56 & 0.83 & 0 & 0 & 0.64 & 0 & 0.76 & 0.95 & 0 & 0.69 \\ 0 & 0.78 & 0.68 & 0.54 & 0.8 & 0.2 & 0.86 & 0 & 0.71 & 0.9 \end{pmatrix}$$

и

$$R_1^D = (0 \quad 0.9 \quad 0.71 \quad 0.87 \quad 0.54 \quad 0 \quad 0.6 \quad 0 \quad 0.95 \quad 0.58).$$

Вычислим значения η_{1j}^l по формуле (3.12), т.е.:

$$\eta_{ij}^l = \omega_{ij} \cdot \varphi_{lj},\ l = \overline{1,4},\ j = \overline{1,10}.$$

Получим:

$$R_{jl}^{DK} = \begin{pmatrix} 0 & 0 & 0.45 & 0 & 0.51 & 0 & 0 & 0 & 0.74 & 0 \\ 0 & 0.81 & 0.56 & 0.57 & 0.32 & 0 & 0.54 & 0 & 0.66 & 0.45 \\ 0 & 0.75 & 0 & 0 & 0.35 & 0 & 0.46 & 0 & 0 & 0.4 \\ 0 & 0.68 & 0.48 & 0.47 & 0.43 & 0 & 0.52 & 0 & 0.67 & 0.52 \end{pmatrix}$$

По формуле (3.13) вычислим значения весовых коэффициентов абстрактного каталога K^{ab}, т.е.:

$$\eta_{1j}^{ab} = \max_{l=\overline{1,4}} \{\eta_{1j}^l\},\ j = \overline{1,10}.$$

$$\{\eta_{1j}^{ab}\} = \{0 \quad 0.81 \quad 0.56 \quad 0.57 \quad 0.51 \quad 0 \quad 0.54 \quad 0 \quad 0.74 \quad 0.52\}$$

Для всех тематических каталогов вычислим отклонения по формуле (3.14):

$$\lambda_{11} = \frac{5}{10} \cdot \sum_{j=1}^{10} \left(\eta_{1j}^{ab} - \eta_{1j}^1\right)^2 = 0.78,$$

$$\lambda_{12} = \frac{6}{10} \cdot \sum_{j=1}^{10} \left(\eta_{1j}^{ab} - \eta_{1j}^2\right)^2 = 0.03,$$

$$\lambda_{13} = \frac{6}{10} \cdot \sum_{j=1}^{10} \left(\eta_{1j}^{ab} - \eta_{1j}^3\right)^2 = 0.74,$$

$$\lambda_{14} = \frac{7}{10} \cdot \sum_{j=1}^{10} \left(\eta_{1j}^{ab} - \eta_{1j}^{4} \right)^2 = 0.04 .$$

Наименьшее отклонение от K^{ab} имеет тематический каталог K_2, т.е.

$$\lambda_1^* = \min_{l=\overline{1,4}} \{\lambda_{1l}\} = \min\{\lambda_{11}, \lambda_{12}, \lambda_{13}, \lambda_{14}\} = \lambda_{12} = 0.03 .$$

Это означает, что тематический каталог K_2 является наиболее подходящим тематическим каталогом и документ d необходимо включить в него.

Если пороговое значение для отклонений тематик каталогов от абстрактного тематического каталога взять

$$\delta_1 = \delta_2 = \delta_3 = \delta_4 = 0.1,$$

тогда тематические каталоги K_2 и K_4 являются наиболее релевантными, так как:

$$\lambda_{12} < \lambda_{14} \le 0.1.$$

Соответственно, документ d нужно включить как в тематический каталог K_2, так и в K_4.

3.3. Методы повышения точности выбора профиля и улучшения качества тематического каталога

Как отметили выше, для улучшения полноты и точности поиска можно использовать множество синонимов, ассоциирующих слов и словарей. Также было отмечено, что синонимы и ассоциирующие слова терминов объединяются в составе одного множества

$$S=\{s_v\}_V=\{S_j\}_J, \qquad (3.17)$$

где S_j - множество синонимов и ассоциирующих терминов t_j. Для удобства в дальнейшем множество S назовем множеством синонимов. Необходимо отметить, что синонимы также являются терминами. Естественно, любое ключевое слово, которое является синонимом термина одного документа, может оказаться важным (часто встречающимся) термином для другого. Это означает, что для любого термина $t_j \in T$ множество его синонимов S_j является подмножеством множества терминов T. Отсюда вытекает, что множество S является подмножеством T. Поэтому для удобства в дальнейшем вместо множества S будем использовать множество T, таким образом, вместо отношения между t_j и s_v, рассмотрим отношение между t_j и t_v, где $j,v=\overline{1,J}$. Это отношение обозначим через r_{jv}^S, которое показывает степень близости t_v к t_j и представляется в виде нечеткой реляционной матрицы размерности JxJ:

$$r_{jv}^S=\left\{\nu_{jv} : T\times T \rightarrow [0,1]\right\}, j,v=\overline{1,J}, \ , \ R^S=\left\{r_{jv}^T\right\}_{J\times J} \qquad (3.18)$$

где ν_{jv} - функция принадлежности термина t_v в множество синонимов S_j, иначе говоря степень близости терминов t_j и t_v. Если для синонимов $t_j \in T$ и $t_v \in T$ не выполняется условие $t_v \in S_j$, тогда $\nu_{jv}=0$.

Знание о синонимах и степени смысловой близости их к терминам дает возможность расширять знание о тематике документа, что позволяет лучше определять тематический профиль и выбрать наиболее релевантный тематический каталог. Для улучшения (дополнения) знаний о документе нужно объединить знания о

терминах и синонимах терминов. Если T_i^D - множество терминов информационного ресурса d_i и S_j - множество синонимов термина $t_j \in T_i^d,\ j = \overline{1,J}$, тогда

$$\tilde{T}_i^D = \bigcup_{t_j \in T_i^D} \left(S_j \cap T_i^D \right), \tag{3.19}$$

является улучшенным множеством терминов данного документа.

Значения новых отношений $\tilde{\omega}_{ij}$, соответствующие новому множеству $\tilde{T}_i^D$, можно найти следующим образом:

$$\tilde{\omega}_{ij} = \max_{v=\overline{1,J}} \left\{ \nu_{jv} \cdot \omega_{iv} \right\},\ j = \overline{1,J} \tag{3.20}$$

Формула (3.20) дает нам новые (улучшенные) значения весовых коэффициентов всех терминов. Решение задачи (3.12)-(3.15) используя вместо ω_{ij} новые весовые коэффициенты $\tilde{\omega}_{ij}$ позволяет достичь более эффективных результатов.

Пример 2. Возьмем те же множества *T* и *K* (также отношения R_l^K и R_1^D), которые определены в предыдущем примере. Пусть еще задана матрица отношений $R^S = \left\{ r_{jv}^T \right\}_{J \times J}$:

$$R^S = \begin{pmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \\ S_9 \\ S_{10} \end{pmatrix} = \begin{pmatrix} 1 & 0.9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.9 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.9 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0.9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.9 & 0 & 0 & 1 & 0.9 \\ 0 & 00 & 00 & 0 & 0 & 0 & 0 & 0 & 0.9 & 1 \end{pmatrix}$$

Тогда по формуле (3.20) получим значения:

$$R_1^D = (0.81 \quad 0.9 \quad 0.78 \quad 0.87 \quad 0.54 \quad 0.86 \quad 0.6 \quad 0 \quad 0.95 \quad 0.86)$$

Вычислим отклонения тематики каталогов по формуле (3.12)-(3.15) используя новые значения $\tilde{\omega}_{ij}$, тогда получим:

$$\lambda_{11} = 0.94,$$

$$\lambda_{12} = 0.64,$$

$$\lambda_{13} = 1.1,$$

$$\lambda_{14} = 0.56.$$

Отсюда видно, что в отличии от предыдущего примера тематический каталог K_4 является наиболее релевантным данному документу, т.е.

$$\lambda_{14} = \min_{l=1,4} \lambda_{1l} = 0.56.$$

Следует отметить, что K_2 также является релевантным каталогом, коэффициент отклонения которого немного превышает λ_{14}.

3.4. Алгоритм поиска наиболее релевантных документов по нечеткому запросу пользователя

Как отметили выше, запросы пользователя составляются из поисковых признаков, т.е. ключевых слов или терминов $Q = \{t_l\}_L$. В начале поиска в ИПС запросы пользователя также преобразуются в формат, соответствующий формату представления документов. Для каждого признака определяется отношение его к данному запросу. Другими словами, указывается степень важности признаков для

данного запроса, который представляется в виде нечеткого отношения [34, 43]:

$$Q = \left\{ t_l / \alpha_{t_l}(Q) \right\},\ l = \overline{1,L}\ , \tag{3.21}$$

а отношение "запрос-термин" определяется следующим образом:

$$R^Q = \left\{ r_l^Q \right\}_L = \left\{ \alpha_{t_l}(Q) : D \times Q \to [0,1] \right\},\ t_l \in Q, l = \overline{1,L}\,, \tag{3.22}$$

где $\alpha_l = \alpha_{t_l}(q)$ - важность признака t_l для запроса Q, r_l^Q - нечеткое отношение термина t_l запросу Q.

Сначала необходимо нормировать множество Q, т.е. множество Q дополняется элементами множества T, которые отсутствуют в запросе, с нулевыми коэффициентами важности. Тогда отношение (3.22) получит следующий вид:

$$R^{QT} = \left\{ \alpha_l^T : T \times Q \to [0,1] \right\},\ l = \overline{1,J},\quad R^{QT} = \left\{ r_l^{QT} \right\}_L, \tag{3.23}$$

здесь если $t_l \in Q$, то $\alpha_l^T = \alpha_l$, иначе $\alpha_l^T = 0$.

Для улучшения полноты и точности поиска будем использовать множество синонимов, т.е.:

$$\tilde{Q} = \bigcup_{t_j \in Q} \left(S_j \cap Q^T \right),\ j = \overline{1,J} \tag{3.24}$$

и, соответственно,

$$\tilde{\alpha}_l = \max_{j=\overline{1,J}} \left\{ v_{jl} \cdot \alpha_l^{QT} \right\},\ l = \overline{1,J}\,. \tag{3.25}$$

В дальнейшем когда будем говорить о запросе Q будем предполагать, что это нормализованный и дополненный вектор $\tilde{Q}$ нечетких отношений.

Теперь рассмотрим задачу поиска информации на основе такого нечеткого запроса Q. Результатом такого поиска является список документов, релевантных к запросу пользователя, т.е.

$$I_R = \{d_g\}_G . \tag{3.26}$$

Тогда

$$I_R = Q \cap D = Q \cap \left(\bigcup_{i=\overline{1,I}} d_i \right) = \bigcup_{i=\overline{1,I}} (Q \cap d_i). \tag{3.27}$$

Задача (3.25) определения наиболее релевантных документов к запросу Q аналогична задаче (3.8) и для ее решения можно также использовать метод Беллмана-Заде.

Пересечение множеств отношений R_i^D и R^Q дает нам степени релевантности d_i к запросу Q, т.е.

$$R_i^{DQ} = R_i^D \cap R^Q . \tag{3.28}$$

Если

$$R_i^{DQ} = \{\gamma_{ij}\},\ i = \overline{1,I}\ \ j = \overline{1,J}\ , \tag{3.29}$$

тогда

$$\gamma_{ij} = \mu_{ij} \cdot \alpha_j ,\ i = \overline{1,I},\ j = \overline{1,J}\ . \tag{3.30}$$

где γ_{ij} - степень релевантности документа d_i к запросу Q по отношению термина t_j.

Пусть d_i^{ab} - абстрактный документ, который является идеально релевантным к запросу Q, тогда для степени релевантности d_i^{ab} к запросу Q по отношению термина t_j напишем:

$$\gamma_j^{ab} = \max_{i=\overline{1,I}}\{\gamma_{ij}\},\ j=\overline{1,J}, \qquad (3.31)$$

Теперь определим суммарное среднеквадратическое отклонение $\gamma_{ij}, i=\overline{1,I}, j=\overline{1,J}$ от коэффициентов $\gamma_j^{ab}, j=\overline{1,J}$:

$$\lambda_i = \frac{\overline{J}}{J}\sum_{j=1}^{J}\left(\gamma_j^{ab}-\gamma_{ij}\right)^2, i=\overline{1,I}. \qquad (3.32)$$

Документ с минимальным отклонением от "идеально подходящего документа" является наиболее релевантным:

$$\lambda^* = \min_{i=\overline{1,I}}\{\lambda_i\}. \qquad (3.33)$$

Решение (3.30)-(3.33) позволяет определять один наиболее релевантный информационный ресурс d^*. Однако на практике требуется найти не один источник, а все наиболее релевантные источники информации.

Для решения данной проблемы введем параметр ε_i - пороговое значение релевантности. Тогда все документы, удовлетворяющие условия

$$\lambda_i \le \varepsilon_i,\ i=\overline{1,I} \qquad (3.34)$$

будем считать релевантными к запросу.

Пример 3. Пусть $Q=\{t_3/0.95, t_5/0.8, t_9/0.8\}$ - пользовательский запрос, а $D=\{d_1,d_2,d_3,d_4,d_5\}$ - документы информационного пространства, которые индексировались поисковой системой. В результате индексирования выявлены отношения терминов множества $T=\{t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8,t_9,t_{10}\}$ к документам d_j.

$$d_1(t_j/\mu_{1j})=\{t_1/0.7;t_2/0;t_3/0.9;t_4/0.8;t_5/0;$$
$$t_6/0.7;t_7/0;t_8/0.8;t_9/0;t_{10}/0.9\},$$

$$d_2(t_j/\mu_{2j})=\{t_1/0.6;t_2/0;t_3/0.8;t_4/0;t_5/0.9;$$
$$t_6/0.9;t_7/0;t_8/0;t_9/0.8;t_{10}/0\},$$

$$d_3(t_j/\mu_{3j})=\{t_1/0;t_2/0.6;t_3/0.9;t_4/0;t_5/0.9;$$
$$t_6/0;t_7/0;t_8/0;t_9/0.7;t_{10}/0\},$$

$$d_4(t_j/\mu_{4j})=\{t_1/0;t_2/0;t_3/0;t_4/0;t_5/0.9;$$
$$t_6/0.7;t_7/0.8;t_8/0;t_9/0.6;t_{10}/0\},$$

$$d_5(t_j/\mu_{5j})=\{t_1/0.6;t_2/0.7;t_3/0.9;t_4/0;t_5/0;$$
$$t_6/0;t_7/0.6;t_8/0;t_9/0;t_{10}/0.8\}.$$

Множество Q дополним элементами T:

$$Q=\{t_1/0,t_2/0,t_3/0.95,t_4/0,t_5/0.8,t_6/0,t_7/0,t_8/0,t_9/0.8,t_{10}/0\}.$$

По формуле (3.30) найдем γ_{ij}:

$$\{\gamma_{ij}\}=\begin{pmatrix}0&0&0.86&0&0&0&0&0&0&0\\0&0&0.76&0&0.72&0&0&0&0.64&0\\0&0&0.86&0&0.72&0&0&0&0.56&0\\0&0&0&0&0.72&0&0&0&0.48&0\\0&0&0.86&0&0&0&0&0&0&0\end{pmatrix}$$

Тогда из (3.31) вычислим γ_j^{ab}, $j=\overline{1,J}$:

$$\{\gamma_j^{ab}\}=(0\quad 0\quad 0.86\quad 0\quad 0.72\quad 0\quad 0\quad 0\quad 0.64\quad 0).$$

Из (3.32) получим:

$$\{\lambda_i\} = \begin{bmatrix} 0.82 \\ 0.05 \\ 0.06 \\ 0.41 \\ 0.68 \end{bmatrix},$$

а из (3.33) имеем

$$\lambda^* = 0.05,$$

т.е. документ d_2 является наиболее релевантным к данному запросу. Если пороговые значения $\varepsilon_i = 0.1$ для всех $i = \overline{1,5}$, тогда d_2 и d_3 являются релевантными.

3.5. Поиск релевантной информации на основе нечетких отношений предпочтительности

На практике обычно запросы, сформулированные пользователем для поиска информации состоят из нескольких ключевых слов (терминов). В таком случае поисковые системы выдают пользователю множество документов с определенным уровнем релевантности к заданному запросу в целом. Здесь релевантность документов запросу по отдельным ключевым словам отличаются.

Кроме этого, важности ключевых слов в запросе разные, пользователю известно какое ключевое слово является более важным, а какое менее важным. Поэтому коэффициенты их важности определяются пользователем во время подготовки запроса. Исходя из вышесказанного, приходится делать средневзвешенный выбор среди выданных информационных ресурсов, отличающиеся наиболее высоким уровнем релевантности запросу в целом.

Далее в этом параграфе рассматривается метод поиска документов, наиболее релевантных запросу пользователя, на основе нечетких отношений предпочтительности [38].

Задача поиска релевантных документов из информационного пространства рассматривается как задача принятия решений по выбору наиболее подходящих (релевантных) источников информации в рамках модели (3.3). Как сказано выше, здесь информационное пространство представляется множествами документов и поисковых признаков (терминов), а также множеством нечетких отношений между ними.

Здесь уточним некоторые детали этой модели. Согласно определениям в параграфе 3.1, ω_{ij} является функцией принадлежности термина t_j документу d_i, она описывает нечеткие отношения терминов к документам, которые определяются с помощью методов индексирования (в случае автоматического индексирования) или знаний экспертов в данной области (в случае тематических каталогов). Определенные таким образом отношения между терминами и документами обычно называются весовыми коэффициентами терминов. Далее в работе предполагается, что весовые коэффициенты заранее известны, т.е. определены в процессе индексирования и занесены в базу индексов поисковой системы.

Аналогичным образом представляются и запросы пользователей $Q = \{t_l\}_L$, где t_l - поисковые признаки, т.е. термины, характеризующие искомые документы. Запрос Q является вектором размерности L, который в дальнейшем нормализуется согласно представлению (размеру) множества $D = \{d_i\}_I$, т.е. он преобразуется системой в такой же формат, в каком представлены документы D.

Как отметили выше, включенные в запросы термины имеют различные степени важности для искомых документов, поэтому во время оформления запроса он помимо списка ключевых слов определяет степень важности каждого ключевого слова для данного запроса, т.е. вводит коэффициенты важности терминов данного запроса (3.21) и (3.22).

Теперь рассмотрим задачу поиска релевантных документов на основе нечетких отношений предпочтительности. Пусть пользовательский запрос состоит из подмножества терминов T^Q и для всех терминов $t_l \in T^Q$ определены $\alpha_{t_l}(Q) \to [0,1]$. Степень важности терминов, входящих в T ($t_l \in T$), но не входящих в T^Q ($t_l \notin T^Q$) являются нулевыми. Другими словами, множество T^Q дополняется терминами множества T с нулевыми коэффициентами важности $\alpha_{t_l}(Q) = 0, \forall t_l \notin T^Q$.

С целью учета степеней важности терминов запроса в процессе выбора релевантных документов для всех пар элементов множества T^Q (т.е. ключевых слов запроса) определяются нечеткие отношения предпочтения:

$$\gamma : T^Q \times T^Q \to [0,1] \qquad (3.35)$$

с функцией принадлежности $\gamma(t_u, t_v)$, которая показывает насколько термин t_u важнее термина t_v. Значения $\gamma(t_u, t_v)$ вычисляется на основе коэффициентов важности $\alpha_{t_u}(Q)$ и $\alpha_{t_v}(Q)$, соответственно, терминам t_u и t_v для запроса:

$$\gamma(t_u, t_v) = \begin{cases} 1 - \left| \alpha_{t_v}(Q) - \alpha_{t_u}(Q) \right|, & \text{если } \alpha_{t_u}(Q) < \alpha_{t_v}(Q) \\ 1, & \text{в обратном случае} \end{cases} \qquad (3.36)$$

Поиск документов, релевантных запросу представляет из себя процедуру выбора документов, наиболее близких запросу по всем поисковым признакам. Для этого сначала определяются степени предпочтительности (т.е. релевантности) документов множества *D* по отдельным терминам t_v. Степень предпочтительности документов описывается с помощью нечетких отношений предпочтения документов, которое вычисляется для всех пар документов на множестве документов D относительно каждого термина на основе их весовых коэффициентов. Иначе говоря определяется функция принадлежности $\mu_{t_v}(d_i, d_j)$ следующего вида:

$$\mu_T : D \times D \to [0,1], \qquad (3.37)$$

где $\mu_{t_v}(d_i, d_j)$ - функция принадлежности, описывающая нечеткое отношение предпочтения документа d_i относительно документа d_j по термину t_v, а ее значение понимается как степень предпочтительности (релевантности) документа d_i относительно документа d_j по термину t_v, т.е. насколько больше весовой коэффициент термина t_v для документа d_i, чем для документа d_j. Если для документа d_i термин t_v "не менее важен", чем для документа d_j, то предполагается, что документ d_i не менее предпочтительный (не менее релевантный), чем документ d_j.

Значения функции принадлежности $\mu_{t_v}(d_i, d_j)$ определяются на основе весовых коэффициентов $\omega_{t_v}(d_i)$ и $\omega_{t_v}(d_j)$ термина t_v для документов d_i и d_j следующим образом:

$$\mu_{t_v}(d_i, d_j) = \begin{cases} 1 - [\omega_{t_v}(d_j) - \omega_{t_v}(d_i)], & \text{если } \omega_{t_v}(d_i) < \omega_{t_v}(d_j) \\ 1, & \text{в обратном случае} \end{cases} \qquad (3.38)$$

С учетом этих нечетких отношений предпочтения между парами документов на множестве документов D определяется нечеткое подмножество предпочтительных документов при фиксированных терминах $t_v \in T$, функция принадлежности которого задается в виде:

$$\mu_{t_v}^{pr\cdot}(d_i) = 1 - \max_{d_j \in D} \left\{ \mu_{t_v}^{s} \left(\mathrm{d_j}, \mathrm{d_i} \right) \right\}, \qquad (3.39)$$

где значения функции принадлежности $\mu_{t_v}^{pr}(d_i)$ показывают, что насколько документ d_i предпочтительнее (более релевантный), чем другие документы множества D по термину t_v, а $\mu_{t_v}^{s}\left(d_j, d_i\right)$ - нечеткое отношение строгого предпочтения документа $\mathrm{d_i}$ относительно документа d_j по термину t_v, которое определяется по формуле:

$$\mu_{t_v}^{s}\left(\mathrm{d}_j, \mathrm{d_i}\right) = \begin{cases} \mu_{t_v}\left(\mathrm{d_i}, \mathrm{d_j}\right) - \mu_{t_v}\left(\mathrm{d_j}, \mathrm{d_i}\right), & \textit{если } \mu_{t_v}\left(\mathrm{d_i}, \mathrm{d_j}\right) > \mu_{t_v}\left(\mathrm{d_j}, \mathrm{d_i}\right) \cdot \\ 0, & \textit{в обратном случае} \end{cases} \qquad (3.40)$$

После того, как известны степени предпочтительности всех документов множества D (т.е. нечеткое подмножество предпочтительных документов) по отдельным терминам запроса можно определять нечеткие отношения предпочтения между парами документов множества D с учетом всех терминов множества T, используя при этом нечеткие отношения предпочтения между парами терминов $\gamma(t_u, t_v)$:

$$\mu_T(d_i, d_j) = \max_{t_u \in T} \left\{ \min_{t_v \in T} \left\{ \mu_{t_u}^{pr\cdot}(d_i), 1 - \mu_{t_v}^{pr\cdot}(d_j), \gamma(t_u, t_v) \right\} \right\}, \qquad (3.41)$$

где $\mu_T(d_i, d_j)$ - функция принадлежности множества нечеткого предпочтения документов множества D, значение которой определяет

степень предпочтительности документа d_i документу d_j по всем терминам запроса, т.е. по запросу в целом.

Аналогично выше приведенному подходу с помощью полученных нечетких отношений предпочтительности для всех пар документов множества D можно определить нечеткое подмножество нечетких отношений предпочтительности документов множества D, т.е. степени предпочтительности документов множества D для запроса в целом:

$$\mu_T^{pr}(d_i) = 1 - \max_{d_j \in D} \left\{ \mu_T^{s}(d_j, d_i) \right\}, \qquad (3.42)$$

где

$$\mu_T^{s}(d_j, d_i) = \begin{cases} 0, & \text{если } \mu_T(d_j, d_i) \le \mu_T(d_j, d_i) \\ \mu_T(d_i, d_j) - \mu(d_j, d_i), & \text{в обратном случае} \end{cases}. \qquad (3.43)$$

Здесь $\mu_T^{pr}(d_i)$ - функция принадлежности, характеризующая подмножество нечетких отношений предпочтительности. Эта функция описывает степень предпочтительности документа d_i для запроса, что равносильно степени релевантности документа d_i к запросу пользователя.

Естественно, $\mu_T^{pr}(d_i)$ может принимать любое значение в интервале [0,1], однако пользователя интересует документы с высокими степенями релевантности. Чтобы учесть этот фактор, можно ввести пороговое значение $\lambda \in [0,1]$ (обычно пороговое значение принимается $\lambda > 0.5$) для степени предпочтительности, релевантности. Тогда все документы, для которых выполняется условие

$$\mu_T^{pr}(d_i) \ge \lambda, \qquad (3.44)$$

считаются наиболее релевантными.

Таким образом, подмножество документов

$$D^Q = \left\{ d_i ; d_i \in D;\ \mu_T^{pr}(d_i) \geq \lambda \right\} \qquad (3.45)$$

является множеством релевантных запросу документов и выдается в качестве результата поиска. А документ d^* с функцией принадлежности

$$\mu(d^*) = \max_{d_i \in D^Q} \left\{ \mu_T^{pr}(d_i) \right\} \qquad (3.46)$$

является самым релевантным.

Предложенный метод поиска информации позволяет найти наиболее релевантные документы, т.е. наиболее полно удовлетворяющие желания пользователя. Другими словами, данный метод сравнивает все документы информационного пространства Интернет и находит те документы, которые наиболее предпочтительны, чем другие документы. При этом учитываются как отношения между терминами и документами, так и отдельными документами.

3.6. Организация распределенного поиска на основе отношений предпочтительности

Как сказано выше, в среде Интернет существуют множество поисковых систем, каждый из которых имеет свои преимущества и недостатки. Разные поисковые машины индексируют источники информации (Web-сайтов, архивов файлов, баз данных и др.) по разному. Т.е. поиски, проводимые разными ИПС по одинаково

оформленным запросам, дают разные результаты. В таких результатах коэффициенты релевантности одних и тех же документов отличаются. Кроме этого, поисковые системы могут иметь тематическую направленность, которая может отразиться на результатах поиска.

Поэтому, представляет актуальность задача сравнительного анализа результатов разных поисковых машин и выбора наиболее подходящих документов из числа выданных на один и тот же запрос этими поисковыми машинами. Следует отметить, что множество документов, выданных поисковыми машинами, намного меньше, чем множество всех документов информационного пространства, которое охватывается поисковыми системами. Можно предполагать, что обработка множества документов, выданных поисковыми машинами может намного улучшить результаты поиска.

Организация распределенного поиска позволяет улучшить такие основные характеристики информационно-поисковых систем, как [46]:

- быстродействие, путем параллельного выполнения подсистем поиска;

- возможность принятия самостоятельных локальных решений без привлечения удаленных компонентов системы;

- модульность - независимость разработки подсистем, расширение системы и т.д.;

- отказоустойчивость - выход из строя одного узла или канала не остановит в целом работу системы;

- сокращение сетевых и временных ресурсов и т.д.

Организация поиска еще более существенна при создании распределенной поисковой системы, состоящей из подсистем поисковых роботов и машин, т.е. мульти-поисковой системы.

Выдаваемые подсистемами локальные решения необходимо проанализировать, обобщить и после этого среди них сделать выбор.

Далее рассматривается метод распределенного поиска путем сравнительного анализа локальных решений поисковых подсистем и принятия общего решения по нечетким отношениям предпочтения между ними.

Другими словами рассмотрим задачу создания мульти-поисковой системы, состоящей из множества поисковых машин, и метод принятия общих решений по результатам этих поисковых машин.

Пусть $S = \{s_k, k = \overline{1,m}\}$ - множество поисковых машин мульти-поисковой системы, $D = \{d_i, i = \overline{1,n}\}$ - множества документов, среди которых необходимо выбрать наиболее релевантные. Отметим, что $D = D_1 \bigcap D_2 \bigcap ... \bigcap D_m$, где $D_k, k = \overline{1,m}$ - документы, выданные поисковой системой S_k и $D_i \bigcup D_j \neq \varnothing$ для всех $i \neq j$. Наконец, Q - запрос пользователя.

Предположим, что каждый документ d_i, выданный поисковой машиной S_k, удовлетворяет запрос пользователя с некоторой степенью, которая определяется функцией принадлежности $\eta_{S_k}(d_i)$ в интервале $[0,1]$. Исходя из значений этой функции, можно определять нечеткое отношение предпочтения на множестве документов. Это отношение описывается функцией принадлежности $\psi_{S_k}(d_i, d_j)$, значение которой определяется в интервале $[0,1]$ и выражает степень предпочтения документа d_i документу d_j по результатам поисковой машины S_k. Она определяется по следующей формуле:

$$\psi_{s_k}(d_i,d_j)=\begin{cases}1-\left[\mu_{s_k}(d_j)-\mu_{s_k}(d_i)\right], & \text{если } \mu_{s_k}(d_j)\geq\mu_{s_k}(d_i)\\ 1, & \text{если } \mu_{s_k}(d_j)<\mu_{s_k}(d_i)\end{cases} \qquad (3.47)$$

где $k=\overline{1,m},\ i,j=\overline{1,n}$.

С помощью (3.47) определяются матрицы нечетких отношений предпочтения документов для всех поисковых машин. Однако следует учесть тот факт, что поисковые машины, входящие в мульти-поисковую систему из-за объективных причин не имеют одинаковые степени результативности по тематике поиска, а также характеристики, связанные географическим расположением, пропускной способностью каналов, объемом охватываемой зоны информационного пространства Интернет и т.д. Пусть степень результативности поисковых машин мульти-поисковой системы выражается коэффициентом $\gamma(s_k)$, определяемым в $[0,1]$.

На основе коэффициентов $\gamma(s_k)$ определяется нечеткое отношение предпочтения поисковых машин

$$\gamma(s_i,s_j)=\begin{cases}1-\left[\gamma(s_j)-\gamma(s_i)\right], & \text{если } \gamma(s_j)\geq\gamma(s_i)\\ 1, & \text{если } \gamma(s_j)<\gamma(s_i)\end{cases}, \qquad (3.48)$$

где $\gamma(s_i,s_j)$ - показывает степень, с которой поисковая машина s_i результативнее s_j.

Задача поиска документов, из информационного пространства Интернет релевантных запросу пользователя, сводится к задаче выбора наиболее релевантных документов из множества документов, выданных поисковыми машинами, на основе вышеописанной информации. Сначала определим нечеткое отношение предпочтений $\psi_{s_k}(d_i,d_j)$ для всех фиксированных поисковых машин $s_k\in S$.

Подмножество недоминируемых документов определяется функцией принадлежности

$$\psi^{n}_{s_k}(d_i) = 1 - \sup_{d_j \in D} \left[\psi_{s_k}(d_i, d_j) - \psi_{s_k}(d_j, d_i) \right]. \qquad (3.49)$$

Документы, функция принадлежности $\psi^{n}_{s_k}(d_i)$ которых получает наибольшие значения, являются с индивидуальным решением поисковой машины s_k. Для того, чтобы учесть решения других поисковых машин далее определяется индуцированное нечеткое отношение предпочтения документов с учетом коэффициентов $\gamma(s_i, s_j)$:

$$\eta(d_i, d_j) = \sup_{s_k, s_l \in S} \min \left\{ \psi^{n}_{s_k}(d_i), \psi^{n}_{s_l}(d_j), \gamma(s_k, s_l) \right\} . \qquad (3.50)$$

Это нечеткое отношение предпочтения является результатом "свертки" семейства нечетких отношений $\psi_{s_k}(d_i, d_j)$ в единое результирующее нечеткое отношение предпочтений с учетом информации о результативности поисковых машин в данной предметной области. Задача выбора документов с несколькими отношениями предпочтения сводится к задаче выбора документов с единственным отношением предпочтения. Для ее решения на основе индуцированных отношений предпочтения на множестве документов $\eta(d_i, d_j)$ определяется соответствующее множество недоминируемых документов

$$\tilde{\eta}^{n}(d_i) = 1 - \sup_{d_j \in D} \left[\eta(d_i, d_j) - \eta(d_j, d_i) \right] . \qquad (3.51)$$

И, наконец, определяется нечеткое множество недоминируемых документов:

$$\eta^n(d_i) = min\left[\tilde{\eta}^n(d_i), \eta(d_i, d_j)\right]. \qquad (3.52)$$

Документы, входящие в это подмножество являются результирующим множеством наиболее релевантных документов. А по формуле

$$\eta(d^*) = \min_{d_i \in D}\left[\eta^n(d_i)\right] \qquad (3.53)$$

определяется самый релевантный документ среди документов, выданных поисковыми системами.

Рассматриваемый метод распределенного поиска информации сравнивает все документы информационного пространства и находит наиболее предпочтительные, наиболее релевантные из множества документов, выданных поисковыми системами.

IV ГЛАВА.

МЕТОДЫ ПОСТРОЕНИЯ ПОИСКОВЫХ СИСТЕМ НА БАЗЕ ИЕРАРХИЧЕСКОЙ МОДЕЛИ ИНФОРМАЦИОННОГО ПРОСТРАНСТВА ИНТЕРНЕТ

4.1. Иерархическая модель информационного пространства Интернет

Известно, что количество Web-страниц стремительно растет и объем информации удваивается почти в течении полгода. Однако исследования информационно-поисковых систем Интернет показывают, что несмотря на это, эффективность поиска информации явно отстает от требуемого уровня. Очевидно мы не ошибемся, если главную вину в этом будем искать в беспорядочности (несистематизированности) Web-документов и других информационных ресурсов Интернет, которая выражается в следующих обстоятельствах:

- информационные ресурсы Интернет создаются независимо друг от друга в произвольной форме;
- каждый информационный ресурс характеризуется в лучшем случае только набором признаков и их частотными характеристиками;
- ссылки между информационными ресурсами ставятся интуитивно, логическая связь информационных ресурсов существует только между соседними ссылками, по крайней мере через две-три ссылки;
- существующие браузеры дают возможность квази-случайного поиска по паутине Web-документов,

эффективность которого зависит от специальной и библиотечной подготовки пользователей.

Вышесказанные показывают, что для создания эффективных поисковых систем необходимо систематизировать (индексировать, классифицировать и т.п.) информационные ресурсы Интернет. Это в какой-то мере осуществляется в наиболее известных поисковых системах Интернет, таких как Yahoo, Excite, Alta Vista и др. Основным недостатком такого подхода является то, что каждый документ характеризуется набором ключевых слов и его местонахождением, т.е. URL - адресом, либо каждому ключевому слову или словосочетанию соответствует набор адресов URL, в которых расположены документы с данными информационными признаками.

Семантическая связанность информационных ресурсов и их близость по тематике практически не рассматривается. Следовательно, при таком подходе поиск информации не может быть направленным, возможно даже зацикливание. И неслучайно, что эффективность информационно-поисковых систем Интернет составляет от 16% до 30%.

Несмотря на большую запутанность информационного пространства больших сетей, в том числе Интернет, его можно систематизировать с целью рационального покрытия поисковыми системами. Особенно хорошие результаты можно получить используя корпоративные сети, где информационные ресурсы являются более управляемыми.

Суть предлагаемого подхода заключается в том, что информационное пространство можно разбить на семантически мало связанные поисковые зоны, которые являются зонами охвата отдельных поисковых систем. Информационные ресурсы, входящие в

одну зону поиска должны иметь большую содержательную связанность. При таком подходе благодаря зональному распределению поиск информации получает частично направленный характер и идет по принципу прогулки по содержательным связям, а не по "случайным" гипертекстовым связям или ключевым словам.

Ниже рассматривается задача разработки модели информационного пространства Интернет, которая описывает содержательную связь между имеющимися документами и географическую прозрачность распределенных информационных ресурсов [41].

Информационное пространство Интернет формально можно представить пятеркой.

$$I = \{ S, T, W, D, R \}, \tag{4.1}$$

где

$$S = \{ s_1, s_2, ..., s_m \}, \tag{4.1.1}$$

$$T = \{ t_1, t_2, ..., t_n \}, \tag{4.1.2}$$

$$W = \left\| w_{ij} \right\|_{n \times m},\ w_{ij} \in \{ 0,1,2,... \},\ i = \overline{1,n},\ j = \overline{1,m}, \tag{4.1.3}$$

$$D = \left\| d_{jj'} \right\|_{m \times m},\ d_{jj'} \in \{ 0,1,2,... \},\ j, j' = \overline{1,m}, \tag{4.1.4}$$

$$R = \left\| r_{jj'} \right\|_{m \times m},\ r_{jj'} \in \{ 0,1,2,... \},\ j, j' = \overline{1,m}, \tag{4.1.5}$$

Здесь:

S - множество Web–сайтов;

T - множество информационных признаков (ключевых слов, терминов);

W - матрица отношений информационных признаков к Web-сайтам, т.е. весов признаков;

D - матрица расстояний между Web-сайтами (или ресурсами, либо подпространствами);

R - матрица отношений между Web-сайтами по ссылкам;

$r_{jj'}$ - количество ссылок между Web-сайтами;

$d_{jj'}$ - расстояния, определяемые количеством промежуточных маршрутизаторов между хостами, в которых размещены соответствующие Web-сайты. Эти расстояния можно определить более реально, например, через пропускные способности каналов и узлов и т.д.

Если учесть весовые коэффициенты информационных признаков w_{ij}, то с учетом введенных параметров содержательная связанность Web–сайтов определяется выражением:

$$\alpha_{jj'} = \sum_{i=1}^{n} w_{ij} \cdot w_{ij'} , \quad j, j' = \overline{1,m}. \qquad (4.2)$$

Далее рассматривается задача распределения информационного пространства Интернет по трехуровневой иерархизации. На первом уровне иерархии определяется множество локальных Web-областей, которым могут соответствовать отдельные поисковые серверы или их группы, объединенные по тематическому или географическому признакам.

На втором уровне иерархии определяются региональные информационные ресурсы, которые могут состоять из множества групп или объединенных Web-сайтов.

Наконец, на третьем уровне определяются глобальные информационные среды, объединяющие в своем составе как

тематически, так и географически распределенные информационные ресурсы.

Ниже рассматривается задача, позволяющая формально получить отображение информационного пространства больших сетей, подобных Интернет на предложенную трехуровневую структуру в зависимости от информационно-топологической структуры информационного пространства. Такая структуризация соответствует иерархизации информационного пространства и создает предпосылки организации многоуровневой мета-поисковой системы.

4.2. Разбиение информационного пространства на поисковые зоны

Задача, поставленная в предыдущем разделе распадается на несколько подзадач, первой из которых является разбиение информационного пространства на максимально связанные зоны (Web-области) и с ограничением их числа сверху. Под связанностью Web-областей $v_l \in V,\ l = \overline{1,L}$ подразумевается степень связанности Web-сайтов $s_j \in S, j = \overline{1,m}$, входящих в эти области.

Введем переменные

$$x_{j\ell},\ j = \overline{1,m},\ l = \overline{1,L}, \qquad (4.3)$$

при этом $x_{j\ell} = 1$, если Web-сайт s_j входит в Web-область v_l; $x_{j\ell} = 0$, в обратном случае.

Тогда оптимальное разбиение информационного пространства на максимально связанные Web-области аналитически выражается через суммарную связанность областей $v_l \in V,\ l = \overline{1,L}$:

$$\sum_{l=1}^{L}\sum_{j'=1}^{m}\sum_{j=1}^{m} x_{jl}\, x_{j'l}\, \alpha_{jj'} \to \max_{x_{jl}}, \qquad (4.4)$$

при этом должны быть выполнены следующие условия:

1) каждый Web-сайт должен входить хотя бы в одну область из множества $v_l \in V,\ l = \overline{1,L}$:

$$1 < \sum_{l=1}^{L} x_{jl} < k,\ j = \overline{1,m}, \qquad (4.5)$$

где k - максимально допустимое количество копий Web-сайтов.

2) количество Web-сайтов в каждой области должно быть ограниченно как сверху, так и снизу:

$$Q_{min} \le \sum_{j=1}^{m} x_{jl} \le Q_{max}, l = \overline{1,L} \qquad (4.6)$$

Таким образом, решением задачи (4.4)-(4.6) является предполагаемое разбиение информационного пространства.

4.3. Виртуальное объединение Web-областей

Вторая подзадача поставленной задачи в разделе 4.1 заключается в объединении Web-областей $v_l \in V,\ l = \overline{1,L}$ в некоторые подмножества $\omega_\upsilon \in \Omega,\ \upsilon = \overline{1,\upsilon^m}$ по мере их близости. Такая задача возникает либо при необходимости построения поисковых машин метаданных, либо при построении тематических поисковых машин. Близость между Web-областями определяется выражением:

$$\alpha_{ll'} = \sum_{j'=1}^{m} \sum_{j=1}^{m} x_{j\ell} x_{jl'} \alpha_{jj'} . \qquad (4.7)$$

Для определения, входят ли области $v_l \in V$, $l = \overline{1,L}$ в объединяемые подмножества, введем переменные $y_{l\upsilon}, l = \overline{1,L}, \upsilon = \overline{1,\upsilon^m}$. Тогда $y_{l\upsilon} = 1$, если область v_l входит в подмножество ω_υ, $y_{l\upsilon} = 0$ - в противном случае. Здесь υ^m - заранее задаваемое максимальное количество подмножеств. Такое лимитирование ограничивает эффективность решения поставленной задачи на практике, однако для ее упрощения это необходимо.

Таким образом, суммарная близость подмножеств ω_υ при оптимальном объединении Web-областей $v_l, l = \overline{1,L}$ будет:

$$\sum_{\upsilon=1}^{\upsilon^m} \sum_{l=1}^{L} \sum_{l'=1}^{L} \alpha_{ll'} y_{l\upsilon} y_{l'\upsilon} \to max \qquad (4.8)$$

Далее определим условия, вытекающие из практических требований:

1) каждая Web-область $v_l, l = \overline{1,L}$ должна входить хотя бы в одно подмножество ϖ_υ:

$$\sum_{\upsilon=1}^{\upsilon^m} y_{l\upsilon} \geq 1, l = \overline{1,L}, \qquad (4.9)$$

2) каждое подмножество ω_υ может включать в себя только ограниченное количество областей $v_l, l = \overline{1,L}$

$$\sum_{l=1}^{L} y_{l\upsilon} \leq B, \upsilon = \overline{1,\upsilon^m} \qquad (4.10)$$

Таким образом, задача объединения Web-областей по степени их близости сведена к решению задачи билинейного программирования, т.е. нахождению таких $y_{l\upsilon}, l=\overline{1,L}, \upsilon=\overline{1,\upsilon^m}$, которые бы максимизировали функционал (4.8) и удовлетворяли ограничения (4.9)-(4.10).

На основе решения вышеуказанных двух задач можно предварительно сделать следующие выводы:

- попавшие в одно подмножество области должны быть покрыты одной поисковой машиной (search engine);
- области, попавшие одновременно в несколько подмножеств $\left(\sum_{\upsilon=1}^{\upsilon^m} y_{l'\upsilon} > 1, l'=\overline{1,L}\right)$ должны быть покрыты несколькими поисковыми машинами, которые являются глобальными или мета-поисковыми машинами.

Нередко существующее множество ссылок $\|r_{ii'}\|_{m\times m}$ оказывается далеким от реальности, т.е. не отражает реальную связанность Web-страниц. Поэтому важно знать, какой должна быть паутина, чтобы при ограниченном числе ссылок содержательная связка была максимальной. Такая задача становится весьма актуальной, когда разрабатывается корпоративная Web-среда, либо создается мета-поисковая машина, или вообще необходимо знать идеальную схему паутины по той или иной причине.

Формально эту задачу можно представить следующим образом: найти такие $r_{ii'}, i,i'=\overline{1,m}$, которые позволили бы синтезировать паутину с наилучшими ссылками. Под наилучшими подразумеваются такие ссылки, которые связывают Web-документы с максимально содержательной близостью $l_{ii'}$ по наименьшему расстоянию:

$$l_{ii'} = \frac{1}{d_{ii'}} \sum_{j=1}^{n} w_{ij} w_{i'j},\ i,i' = \overline{1,m}, \qquad (4.11)$$

где $l_{ii'}$ - коэффициент содержательной близости Web-сайтов s_i и $s_{i'}$, $d_{ii'} = \{1,2,...,j,...,n\}$ – расстояние между Web-сайтами s_i и $s_{i'}$.

Множество наилучших ссылок можно определить с помощью следующего функционала:

$$\sum_{i=1}^{m} \sum_{i'=1}^{m} l_{ii'}\, r_{ii'} \to max. \qquad (4.12)$$

Число ссылок ограничивается наличием путаницы в паутине:

$$r_{ii'} \le r^m, i,i' = \overline{1,m}. \qquad (4.13)$$

Таким образом, для построения наилучшей паутины необходимо определить ссылки $r_{ii'}$, $i,i' = \overline{1,m}$, которые приводят к максимуму функционала (4.12) и удовлетворяют ограничениям (4.13). Однако в этом случае эффективность решения (4.12) сильно зависит от выбора максимального значения числа ссылок r^m, которое невозможно однозначно определить.

Для избежания такой неопределенности (нечеткости) непрерывную задачу поставим в следующем виде. Пусть Web-сайты составляют множество $S = \{s_1, s_2, ..., s_m\}$ материальных точек, на которые действуют силы отталкивания и притяжения. Известно, что сила отталкивания и притяжения выражается формулой:

$$F_{ii'} = \frac{\gamma_{m_i m_{i'}}}{d_{ii'}^2}. \qquad (4.14)$$

В нашем случае силы, действующие между материальными точками (т.е. Web-сайтами) определяются как:

$$F_i = \sum_{i'=1}^{m} \frac{(l_{ii'} - l^{cp})\, k(v_i, v_{i'})}{d_{ii'}^2}, \qquad (4.15)$$

где средняя величина коэффициентов содержательной близости Web-сайтов l^{cp} будет:

$$l^{cp} = \frac{1}{m^2} \sum_{i=1}^{m} \sum_{i'=1}^{m} l_{ii'}, \qquad (4.15.1)$$

$k(v_i, v_{i'})$ - согласующий коэффициент, $l_{ii'}$ - коэффициент содержательной близости Web-сайтов s_i и $s_{i'}$.

Если $l_{ii'} - l^{cp} < 0$, то между v_i и $v_{i'}$ действует сила отталкивания, в противном случае - сила притяжения.

Если примем, что под действием сил отталкивания и притяжения материальные точки начинают движения с некоторого начального положения и через некоторое время система приходит в положение равновесия, то имеем уравнения

$$\begin{aligned} m_i \ddot{x}_i &= \sum_{i'=1}^{m} \frac{l_{ii'} - l^{cp}}{(x_i - x_{i'})^2} k(v_i, v_{i'}),\ i = \overline{1,m} \\ m_i \ddot{y}_i &= \sum_{i'=1}^{m} \frac{l_{ii'} - l^{cp}}{(y_i - y_{i'})^2} k(v_i, v_{i'}),\ i = \overline{1,m} \end{aligned} \qquad (4.16)$$

Система (4.16) описывает движение по проекциям на осях X и Y. Начальные условия могут быть следующие:

$$\begin{aligned} X_i(0) &= X_i^0; \quad \dot{X}_i(0) = 0 \\ Y_i(0) &= Y_i^0; \quad \dot{Y}(0) = 0 \end{aligned}, \qquad (4.17)$$

т.к. предполагается, что в момент начала движения материальные точки находились в состоянии покоя. В качестве X_i^0 и Y_i^0, $i = \overline{1,m}$

могут быть взяты произвольные значения. Решение задачи (4.16) весьма трудоемко. Однако нас интересует случай, когда $t \to \infty$. Если предположить, что

$$\begin{aligned} & m_i = 1,\ i = \overline{1,m} \\ & k(\nu_i, \nu_{i'}) = 1,\ \ i,i' = \overline{1,m} \end{aligned}, \tag{4.18}$$

и что система материальных точек обязательно придет в состояние равновесия, т.е.

$$\lim_{t\to\infty} \ddot{x}_i = 0; \quad \lim_{t\to\infty} \ddot{y}_i = 0,\ \ i = \overline{1,m}, \tag{4.19}$$

тогда решение (4.16) при $t \to \infty$ сводится к решению системы нелинейных алгебраических уравнений

$$\begin{aligned} & \sum_{i'=1}^{m} \frac{l_{ii'} - l^{cp}}{(x_i - x_{i'})^2} = 0,\ i = \overline{1,m}, \\ & \sum_{i'=1}^{m} \frac{l_{ii'} - l^{cp}}{(y_i - y_{i'})^2} = 0,\ i = \overline{1,m}, \end{aligned} \tag{4.20}$$

Пусть решением (4.20) являются $\{x_i, y_i\},\ \ i = \overline{1,m}$ - координаты точек на плоскости. Эти точки определяют такое расположение Web-сайтов на плоскости, что декартовое расстояние между ними характеризует их содержательную близость. Таким образом, решение задачи (4.16)-(4.20) определения наилучших ссылок $r_{i,i'},\ \ i,i' = \overline{1,m}$ сводится к определению $\{x_i, y_i; x_{i'}, y_{i'}\}$ пар точек с определенной физической близостью Δx и Δy, т.е. $r_{ii'} = 1$, если выполняются условия

$$\begin{aligned} & |x_i - x_{i'}| \le \Delta x \\ & |y_i - y_{i'}| \le \Delta y \end{aligned}, \tag{4.21}$$

$r_{ii'} = 0$, если не выполняются, при этом можно принять, что

$$\Delta x = \frac{1}{m}\sum_{i=1}^{m} x_i,$$
$$\Delta y = \frac{1}{m}\sum_{i=1}^{m} y_i. \qquad (4.22)$$

4.4. Оценка эффективности структур поисковых систем

В зависимости от распределенности поисковой системы ее эффективность можно оценивать отношением “производительность-стоимость”. За оценку производительности возьмем реакцию системы, или обратную величину от времени реакции T, т.е. $1/T$. Стоимость структуры поисковой системы с точки зрения ее распределенности можно оценить следующим образом:

$$C = C^{\nu} + C^{g}, \qquad (4.23)$$

где

C^{ν} - оценка стоимости локальных поисковых машин,

C^{g} - стоимость глобальных поисковых машин, т.е. мета-поисковых машин.

Параметр C^{ν} пропорционален числу локальных сред ν^m и географической распределенности Web-серверов:

$$C^{\nu} = k^{\nu}\nu^m \sum_{l=1}^{L}(1-\delta_l), \qquad (4.24)$$

где

$$\delta_l = 1, \text{ при } \sum_{\nu=1}^{m} y_{l\nu} > 1,$$

$$\delta_l = 0, \text{ при } \sum_{\nu=1}^{m} y_{l\nu} \le 1. \qquad (4.25)$$

Параметр C^g пропорционален числу глобальных участков G^Σ:

$$C^g = k^g G^\Sigma, \qquad (4.26)$$

где параметр числа глобальных участков G^Σ вычисляется по формуле:

$$G^\Sigma = \sum_{l=1}^{L} \delta_l . \qquad (4.27)$$

Для определения оценки эффективности E следует оценить время реакции мета-поисковой системы в целом, которое определяется взвешенной суммой средних времен реакции локальных поисковых систем:

$$T = \sum_{l=1}^{L} t_l P_l . \qquad (4.28)$$

Здесь P_l - есть вероятность (или относительная частота) использования l-ой поисковой машины. Величины P_l , $l = \overline{1,L}$ должны быть нормированы, т.е. должно выполняться следующее условие:

$$\sum_{l=1}^{L} P_l = 1. \qquad (4.29)$$

Наконец, имея аналитические выражения для определения производительности (обратная величина от времени реакции T) и стоимости структуры поисковой системы, по модели

“производительность-стоимость” можно аналитически оценить ее эффективность E в зависимости от распределенности структуры с помощью следующего выражения:

$$E = \frac{1}{CT}. \qquad (4.30)$$

4.5. Определение степени сложности поисковых зон

Пусть α - степень информационной связанности и β - степень географической рассредоточенности информационного пространства. Тогда сложности поисковых зон Z, характеризующие степень сложности информационной связанности пространства, можно определить по формуле:

$$s = 1 - l^{-\alpha\beta}, \qquad (4.31)$$

где

$$\alpha = \frac{1}{m^2} \sum_{i=1}^{m} \sum_{i'=1}^{m} \alpha_{ii'},$$

$$\beta = \frac{1}{m^2} \sum_{i=1}^{m} \sum_{i'=1}^{m} d_{ii'}. \qquad (4.32)$$

4.6. Определение степени распределенности структуры поисковой системы

Распределенность структуры поисковой системы можно определить для всех видов структур, начиная от отдельных локальных поисковых серверов до широкомасштабной распределенной среды

поиска. Степень распределенности структуры информационно-поисковой системы можно выразить через число локальных сред ν^m и количество глобальных участков G^Σ. Из постановки задачи следует, что эти параметры получают значения в интервалах

$$0 \le \nu^m \le L \text{ и } 0 \le G^\Sigma \le L. \qquad (4.33)$$

Ясно, что максимальная распределенность структуры поисковой системы достигается при

$$\nu^m = L \text{ и } G^\Sigma = L. \qquad (4.34)$$

Учитывая (4.33) и (4.34), степень распределенности структуры поисковой системы можно определить по формуле:

$$R = \frac{G^\Sigma + \nu^m}{2L}, \qquad (4.35)$$

где R получает значения в интервале [0,1].

Таким образом, мы получили аналитические выражения, определяющие эффективность поисковой системы E, степень сложности поисковых зон S и степень распределенности структуры поисковой системы R через информационно-топологические и технические параметры поисковой системы.

Установление явного функционального отношения $E = F(R,S)$ путем аналитических преобразований представляется довольно трудным. Однако нахождение оптимальной точки для максимальной эффективности $E^{max} = F^*(R,S)$ может быть осуществлено с помощью полученных расчетных формул, что иллюстрируется в следующем параграфе.

4.7. Определение степени распределенности структуры поисковой системы для корпоративной сети Академии Наук Азербайджана

Для проведения конкретных расчетов возьмем корпоративную сеть Академии Наук Азербайджана, в которой количество информационных элементов т.е. Web-серверов равно *m=50*; а количество ключевых слов - *n=30*. В состав Web-страниц входят сгруппированные информационные массивы по таким показателям, как анкетные данные сотрудников, их публикации, финансовые показатели учреждений, научно-технические показатели исследований и их результаты и т.п. Отметим, что данные по структуре академической сети соответствуют ее положению к концу 2000 года.

Количество максимально несвязанных областей, в которые группируются Web–серверы, взято как?*L=10*. Для упрощения решения задачи примем, что Web-серверы не имеют копий и их количество в каждой области снизу не ограничено. Тогда, используя специальную программу решения задачи (4.4)-(4.6) по методу ветвей и границ, приведенных в таблице 4.1, получим $x_{jl}, j = 1 \div 50,\ l = 1 \div 10$.

Пусть определены степени близости между попарно взятыми Web-областями v_l и $v_{l'}$, $l,l' = \overline{1,10}$, которые приведены в таблице 4.2.

Решая задачу (4.8)-(4.10) для $v^m = 3$, получаем объединение областей v_l, $l = \overline{1,10}$ в следующие три подмножества (таблица 4.3).

Из таблицы 4.3 видно, области *l=1,4,9* попали в общую область и, следовательно, они должны быть объединены в глобальную среду.

Таблица 4.1.

Распределение Web-серверов по информационным областям

Номера областей	Нормера Web-серверов									
	1-5	5-10	10-15	15-20	20-25	25-30	30-35	35-40	40-45	45-50
1.	3	6	12	19	21	26	31	36	41	49
2.	1	8	14	20	25	27	35	40	42	46
3.	2	10	13	18	22	30	31	37	45	47
4.	3	7	15	19	23	28	32	39	42	48
5.	4	6	11	19	24	26	34	37	42	50
6.	3	9	14	17	23	30	33	38	44	46
7.	5	9	11	18	22	29	32	38	42	49
8.	2	10	12	16	24	30	34	40	44	48
9.	3	8	15	19	21	27	33	36	43	50
10.	1	7	13	19	25	29	35	39	42	47

Теперь рассчитаем оценку качества и сравнительных характеристик структуры поисковой системы, синтезированной выше. На рисунке 4.1. приведена зависимость времени реакции T (измеряется в секундах) от степени распределенности структуры R системы для пяти значений степени сложности информационного пространства. Как видно из рисунка, кривая $T=F(R)$ имеет точку минимума, которая растет и перемещается вправо вдоль оси R с увеличением S.

Таблица 4.2.

Степеней близости между информационными областями

Номера областей	1	2	3	4	5	6	7	8	9	10
1.	0	3	4	5	2	1	1	2	5	3
2.	3	0	4	2	1	2	3	5	3	4
3.	4	4	0	1	3	5	4	1	2	1
4.	5	2	1	0	4	2	1	1	4	3
5.	2	1	3	4	0	3	2	5	1	4
6.	1	2	5	2	3	0	1	3	4	1
7.	1	3	4	1	2	1	0	2	5	1
8.	2	5	1	1	5	3	2	0	1	4
9.	5	3	2	4	1	4	5	1	0	5
10.	3	4	1	3	4	1	4	3	5	0

На рисунке 4.2 показана зависимость времени реакции *T* от степени сложности информационного пространства *R*. Здесь внимание привлекает тот факт, что наклон кривых *T=F(S)* убывает с увеличением степени распределенности *R*.

На рис.4.3 приведена зависимость эффективности поисковой системы от степени распределенности структуры. Здесь *T* - время реакции системы (в секундах) и *C* - стоимость структуры (в условных единицах). Заштрихованная область соответствует рабочему диапазону *R* и включает точку $T_{min}=F(R)$.

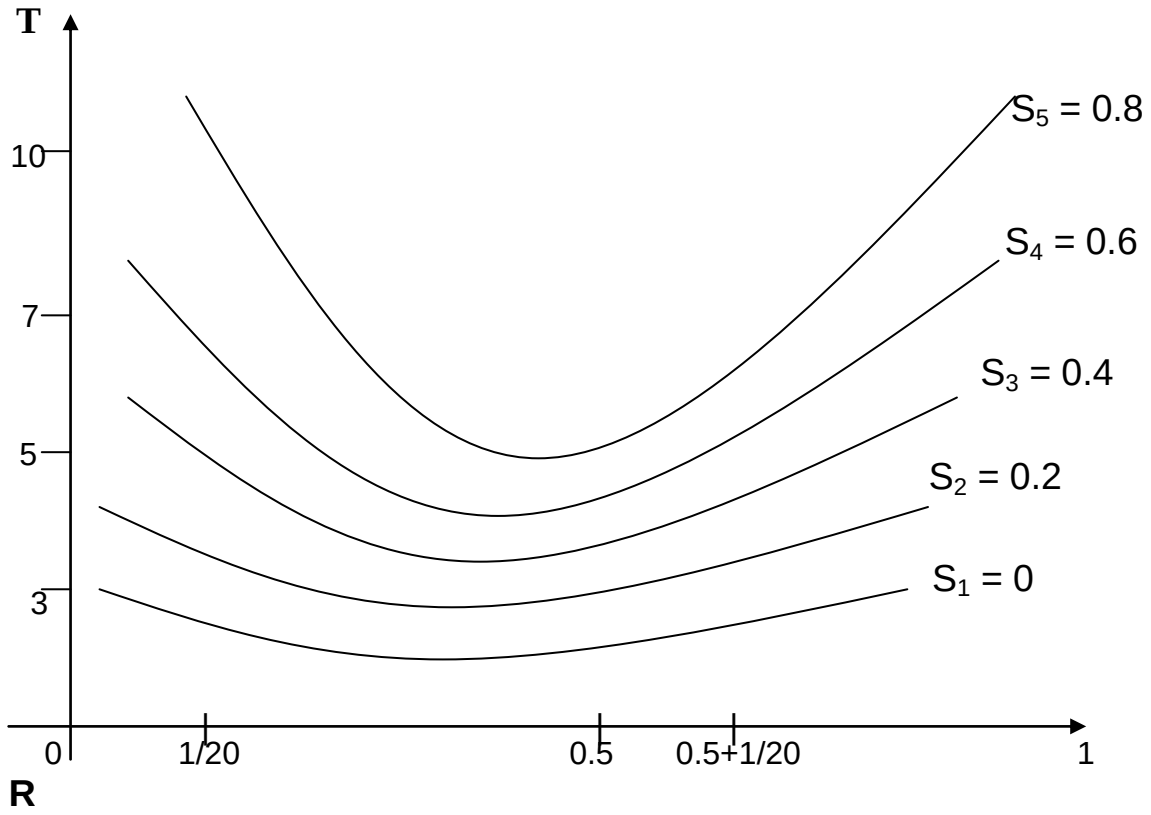

Рис. 4.1. Зависимость времени реакции от распределенности структуры системы

Таблица 4.3.

Распределение информационных областей по уровням

Уровни	Web-области
1.	1, 4, 9
2.	2, 8, 10, 4, 9
3.	3, 5, 6, 7, 1

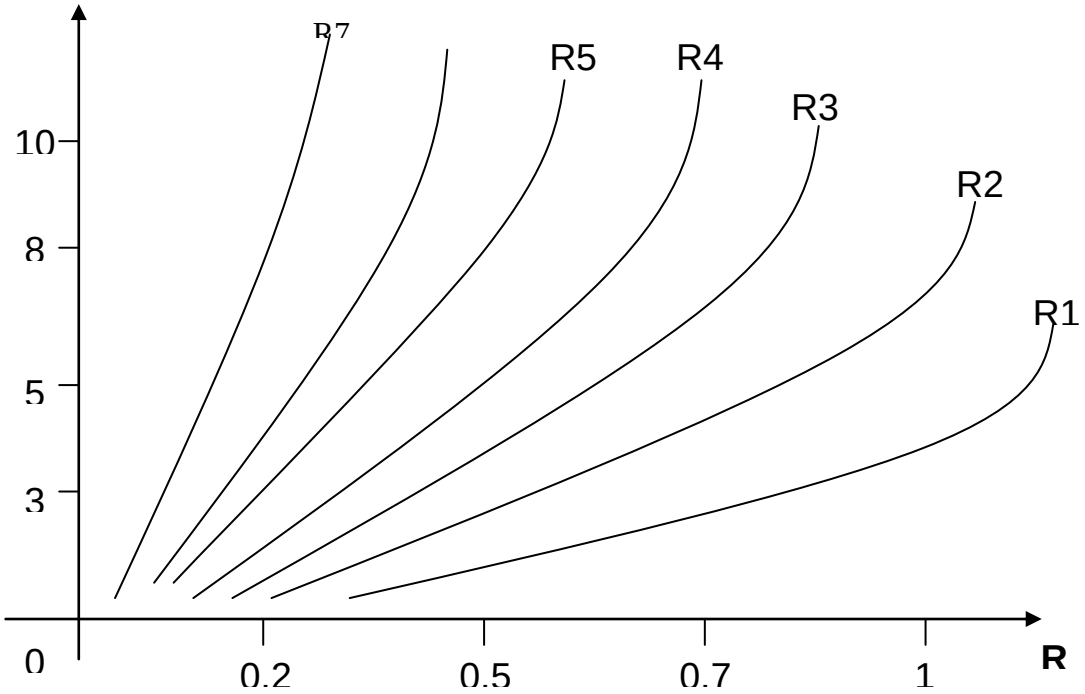

Рис.4.2. Зависимость времени реакции от сложности предметной области

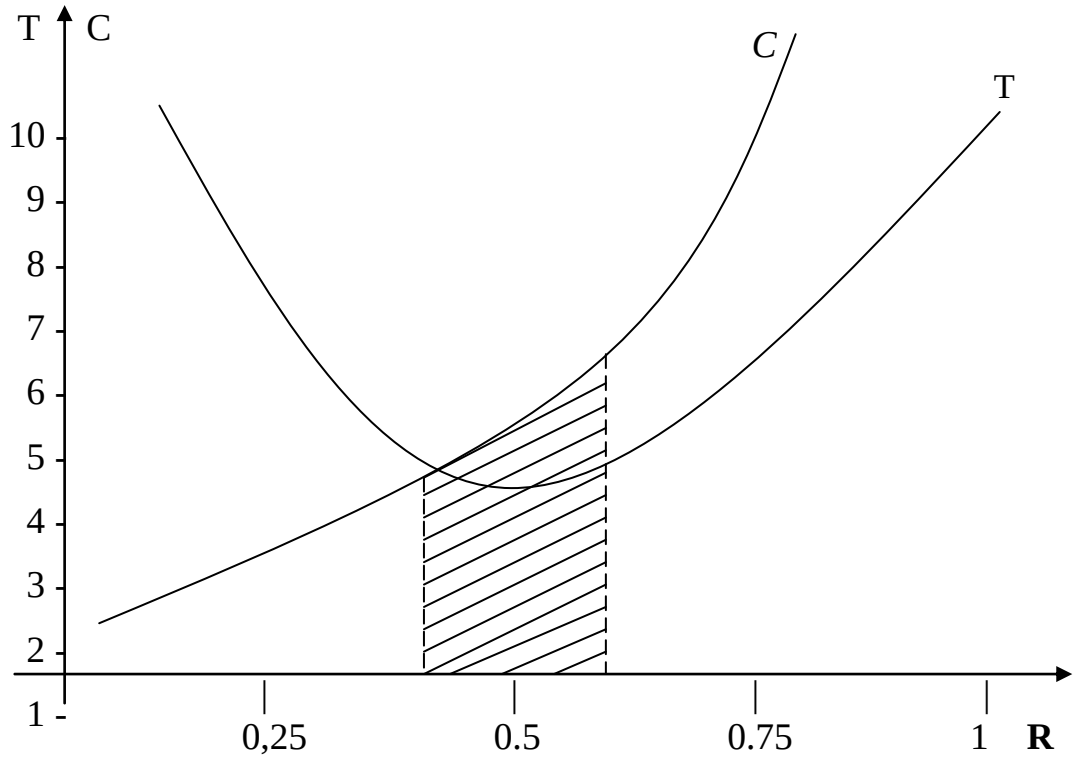

Рис.4.3. Зависимость эффективности поисковой системы от распределенности структуры системы

Поскольку с ростом степени распределенности растет стоимость поисковой системы, а также учитывая предыдущие факторы можно сделать вывод, что для достижения эффективного значения показателя “производительность-стоимость” поисковой системы с увеличением степени сложности предметной области степень распределенности структуры поисковой системы должна расти, и наоборот. Предложенные методы позволяют оптимально разбить информационное пространство Интернет на поисковые зоны, при составлении которых помимо тематической близости ресурсов учитываются такие факторы, как их связанность, расстояние между субъектами сети, содержащие эти ресурсы, а также стоимость и производительность сетевых ресурсов при поиске.

Необходимо отметить, что решением предложенных задач можно достичь требуемый уровень эффективности информационного поиска без изменения параметров поисковых систем, т.е. методов индексирования и алгоритмов поиска. Также с их помощью можно аналитически выразить зависимость эффективности поисковой системы от ее структуры и сложности поисковых зон информационного пространства.

V ГЛАВА.

МОДЕЛИРОВАНИЕ ИНТЕРФЕЙСОВ ПОЛЬЗОВАТЕЛЯ В РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННО-ПОИСКОВЫХ СИСТЕМАХ

5.1. Интерфейсы в распределенных информационных системах

В связи с непрерывным развитием функциональных возможностей информационных и компьютерных сетей, а также со значительным ростом информационных ресурсов, сервисы, предоставляемые информационными системами, становятся все чаще труднодоступными для широкого круга пользователей. При этом трудность связана как с реализованными в этих системах технологиями доступа к информационным системам и ресурсам, так и программно-техническими и другими характеристиками сети и системы.

Это можно объяснить тем, что доступ к большинству информационных систем осуществляется через интерфейс с меню, в котором пользователь ограничен определенными системными рамками, что противодействует повышению производительности. С другой стороны, меню успешно применяется в тех случаях, когда пользователь непосредственно работает с одним или с небольшим числом информационных ресурсов.

Как отметили выше, одной из информационных услуг сети Интернет является поиск информации, эффективность которого определенно зависит от интерфейсов между пользователем, системными и сетевыми ресурсами. Разновидность методов и средств

представления и хранения информационных ресурсов (текстовые, реляционные, гипертекстовые, аудио, видео и т.д.) сети Интернет требует разработки универсального пользовательского интерфейса в ИПС для доступа к различным ресурсам. Это особенно заметно для ИПС с географически и функциональной распределенной структурой [25, 17, 95].

Предполагается, что использование оптимального интерфейса пользователя в распределенных ИПС может привести к значительному улучшению эффективности поиска.

Известно, что результат поиска зависит от методов поиска и обнаружения источников, их индексирования, применяемых в ИПС. Однако, путем реализации оптимального алгоритма поиска информации по заданным признакам в базах данных, содержащих знания о ресурсах, а также методах определения наиболее подходящего направления поиска среди множества направлений, можно значительно повысить производительность ИПС.

Исходя из этого предположения, далее в этой главе рассматриваются принципы организации интерфейсов различного рода и анализируется эффективность использования этих интерфейсов в зависимости от пропускной способности канала связи. Кроме этого здесь предлагаются поисковый алгоритм и метод определения наилучшего направления поиска, основанные на теории нечетких множеств и отношений.

5.2. Интерфейсы поисковых систем и требования к ним

В распределенных системах, в частности в распределенных информационно-поисковых системах, могут быть реализованы следующие интерфейсные средства [2, 42, 45]:

- интерфейс пользователя с поисковой машиной;

- интерфейс между поисковыми машинами и системными ресурсами;

- интерфейс пользователя с сетевыми ресурсами (например, браузеры и т.п.).

Интерфейс пользователя - это граница, на которой пользователь и система осуществляют взаимодействие между собой. Особенности интерфейса пользователя вытекают из "интеллекта" пользовательских рабочих мест. Для обычных терминалов все функции взаимодействия пользователя с системой возлагаются на узел, к которому подключен терминал, или на другие узлы сети. Для интеллектуальных рабочих мест пользовательский интерфейс реализуется непосредственно на рабочей станции пользователя (Netscape, Explorer и т.п.).

Если синтаксический аспект разработки интерфейса пользователя не находит детального анализа (т.к. считается, что синтаксические вопросы в распределенной обработке не порождают принципиально новых задач, связанных с технологией распределенной обработки), то вопрос о семантическом аспекте реализации интерфейса пользователя в ИПС выглядит гораздо шире и сложнее.

Во-первых, в распределенных информационных системах объем информационно-программных ресурсов слишком большой, что исключает возможность знакомства или запоминания пользователем каких-либо признаков организации информационных ресурсов, программных средств или баз данных.

Во-вторых, распределенность и динамичность ресурсов системы отделяет пользователей от этих сетевых ресурсов и форматов обращения к данным. По этой причине семантический аспект

реализации интерфейса пользователя в ИПС считается определяющим фактором для разработчиков сетевых систем и интерфейсов пользователя для распределенной обработки. С другой стороны, интерфейс пользователя не является самостоятельным продуктом, а образует часть клиентских и сетевых программных систем, следовательно, рекомендации и правила, выработанные для разработки интерфейса пользователя одновременно являются и требованиями (или ограничениями) для прикладных программистов при реализации сетевых программных систем.

На современном этапе развития информационных систем требуется, чтобы в ИПС реализовался интеллектуальный пользовательский интерфейс, который может иметь ***двухуровневую структуру***.

На первом уровне для общения с поисковой машиной пользователю предоставляется ИПЯ со специальным словарем, с помощью которого определяются имена ресурсов для формулировки и реализации пользовательского запроса. *На втором уровне* пользователю предоставляются средства общения с сетевыми ресурсами для полного удовлетворения его запроса.

Общая структура интерфейсов в распределенных поисковых системах приведена на рисунке 5.1, где использованы следующие обозначения:

- П – пользователь;
- ПМ – поисковая машина;
- СР - сетевой ресурс;
- $И_1$ - интерфейс пользователя с ПМ;
- $И_2$ - интерфейс ПМ с СР;
- $И_3$ - интерфейс пользователя с СР.

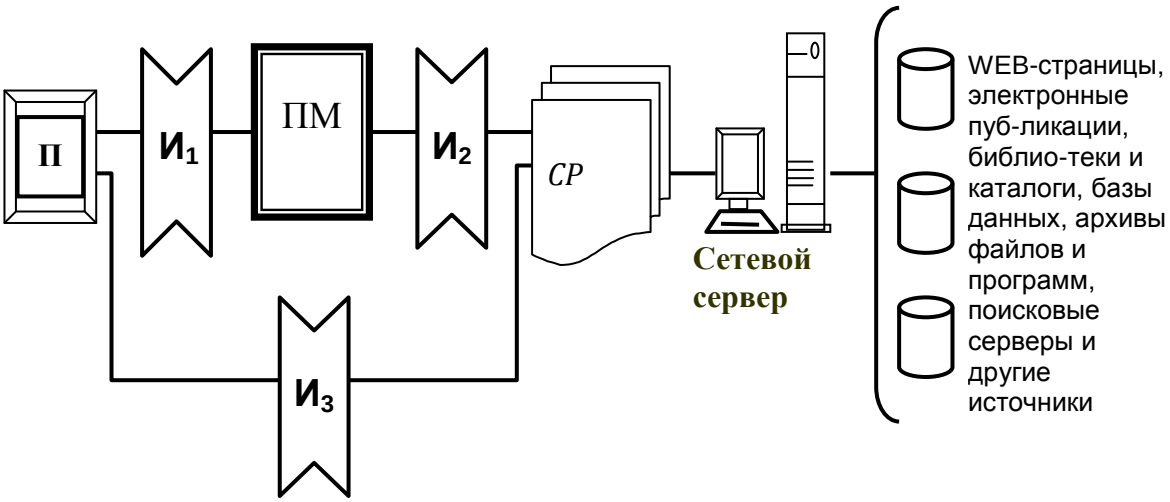

Рис.5.1. Структура системы общения в ПС

Сетевыми ресурсами могут быть приложения широкого спектра, включая Web-серверы, электронные почты, поисковые машины и т.п. К приведенным интерфейсам предъявляются специфические требования.

Как обычно, поисковые машины (по крайней мере ее интерфейсные программы) располагаются на сетевых серверах, обслуживающих пользователей, следовательно, может отсутствовать географическая разобщенность между пользователем и поисковой машиной.

Однако, как правило, сетевые ресурсы разбрасываются по узлам сети и часто пользователь и сетевые ресурсы оказываются географически рассредоточенными.

Таким образом, интерфейсы между поисковой машиной и сетевыми ресурсами ($И_2$), а также пользователем и сетевыми ресурсами ($И_3$) должны обладать значительным быстродействием ведения диалога. Поисковая машина через интерфейс $И_2$ устанавливает связь между пользователем и интересующим его сетевым ресурсом, соответствующим заранее подготовленному

запросу пользователя (через интерфейс $И_1$), который реализуется один раз перед установкой сеанса между пользователем и сетевым ресурсом. А после установления сеанса начинается их прямое взаимодействие с помощью интерфейса $И_3$. Однако $И_3$, как обычно, является диалоговым, и чем больше расстояния между пользователем и сетевым ресурсом, тем менее эффективным будет их взаимодействие из-за низкого коэффициента использования канала связи.

Одним из способов устранения этого недостатка является пересылка программы диалога на узел пользователя для оформления запроса. Однако это дает положительный эффект лишь в том случае, когда пользователь имеет множество запросов. Для одноразовых запросов такая пересылка программ диалога из узла в узел является нерациональным и может привести к усложнению управления системой.

Здесь узлом пользователя называется узел распределенной компьютерной сети, к которому непосредственно подключен пользователь, а программой диалога - программный компонент поисковой системы, позволяющий вести диалог или взаимодействие между пользователем и системой.

Исходя из задач, поставленных перед современными распределенными информационно-поисковыми системами, к предложенным интерфейсным средствам систем предъявляются в основном следующие требования:

- максимальная приближенность к естественному языку;

- максимальная полнота и гибкость;

- независимость от архитектуры системы и организации сетевых ресурсов;

- высокая реакция системы и надежность;
- синтаксическая и семантическая устойчивость.

5.3. Зависимость эффективности канала связи от структуры интерфейсов

Вопрос структурной организации интерфейса пользователя и технология его взаимодействия с сетевыми ресурсами в распределенной среде имеет определенное значение. Разнообразность и постоянная наращиваемость сетевых ресурсов предопределяет целесообразность применения в ИПС два вида интерфейсов пользователя с сетевыми ресурсами:

- с установлением постоянного соединения во время сеанса работы, называется интерактивный, т.е. on-line интерфейс;
- без установления такого соединения, т.е. работа в автономном (off-line) режиме

Эти интерфейсы придерживаются двух противоборствующих подходов. Во-первых, если общение пользователя с сетевым ресурсом будет проводиться в интерактивном режиме (on-line), то канал связи между пользователем и сетевым ресурсом будет излишне загружен, и, следовательно, стоимость общения будет высока. Кроме того, даже в распределенных системах с малым коэффициентом загрузки, но с высоким коэффициентом распределенности структуры (ресурсов и функциональных компонентов), это обстоятельство может привести к резкому снижению эффективности системы в целом. С другой стороны, в автономном (off-line) режиме для формирования запроса необходимо переслать программу диалога на узел пользователя, что

может привести к излишней загрузке канала из-за пересылки части сетевых ресурсов (программы диалога).

Здесь возникает вопрос, какой из этих двух вариантов диалога с точки зрения эффективности использования средств (в т.ч. канала) связи рационален. Для исследования данного вопроса рассмотрим процедурное описание интерфейсов пользователя (рис.5.2) для обоих вариантов. Из рисунка видно, что суммарное время, затрачиваемое на процесс взаимодействия пользователя с сетевыми ресурсами в обоих случаях, определяется суммой времени, затрачиваемых на выполнение отдельных процедур, т.е.:

$$T^i = t^c + t^f + t^p + t^r, \qquad (5.1)$$

$$T^a = t^c + t^d + t^s + t^p + t^r, \qquad (5.2)$$

где

T^i – время, затрачиваемое пользователем на взаимодействие с сетевым ресурсом (продолжительность использования канала связи) в интерактивном режиме, т.е. без загрузки программы диалога;

T^a – время, затрачиваемое пользователем на взаимодействие с сетевым ресурсом (продолжительность использования канала связи) в автономном режиме, т.е. с предварительной загрузкой программы диалога для формулировки запросов;

t^c – время, затрачиваемое на установление связи с сетевым ресурсом;

t^d – время загрузки программы диалога из сетевого сервера пользователем;

t^f – время, затрачиваемое на формирование запроса;

t^s – время доставки запроса;

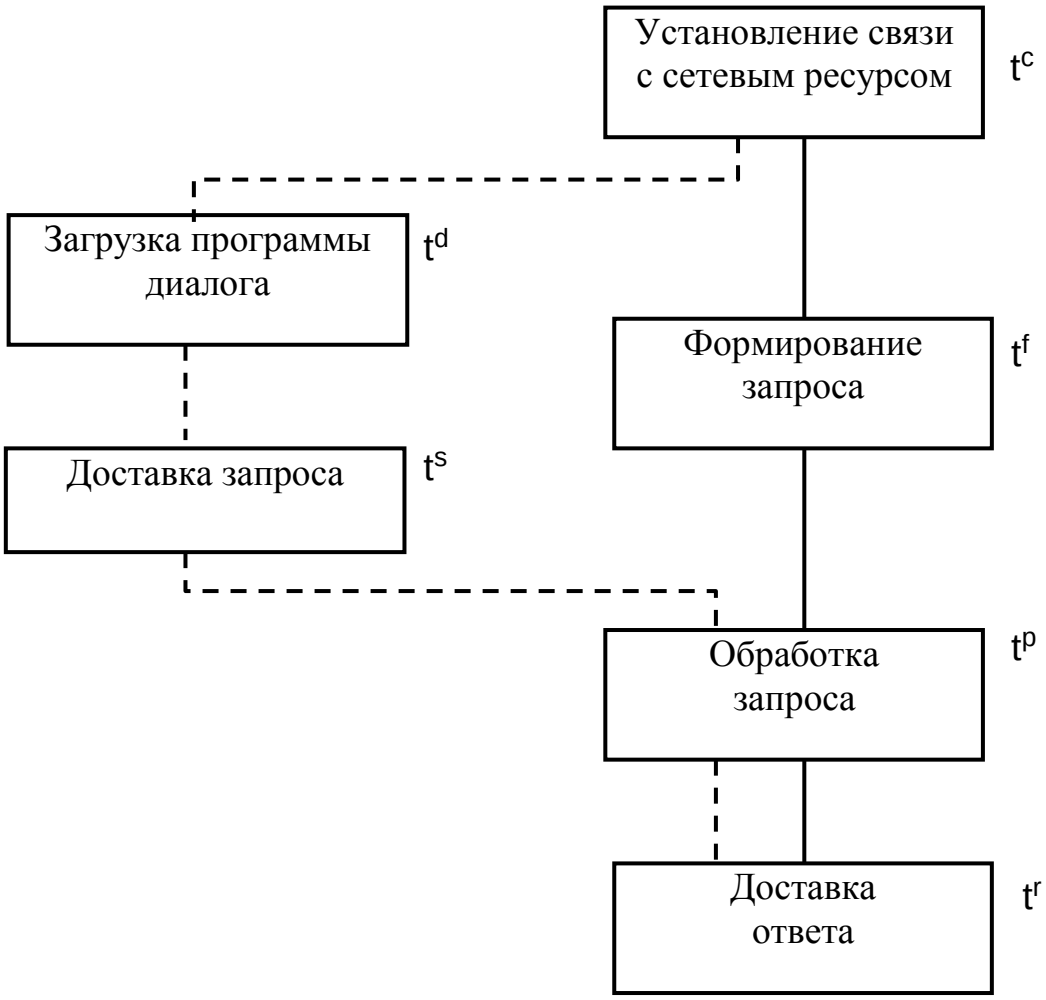

Рис.5.2. Описание интерфейса пользователя с сетевым ресурсом

t^p – время, требуемое для обработки запроса;

t^r – время доставки ответа.

Учитывая специфические особенности интерфейсов пользователя по отношению к каналу связи, выражения (5.1) и (5.2) напишем относительно времен использования канала связи:

$$T^i = t^c + t^f + t^r, \qquad (5.3)$$

$$T^a = t^c + t^d + t^s + t^r, \qquad (5.4)$$

Для сравнения T^i и T^a их можно представить без общих членов. Удаляя t^c и t^r в (5.3) и (5.4) получим сравнительные формулы для T_c^i и T_c^a:

$$T_c^i = t^f, \tag{5.5}$$

$$T_c^a = t^d + t^s, \tag{5.6}$$

Понятно, что при загрузке программы диалога пользователь может формировать не один, а несколько запросов, поэтому продолжительность использования канала связи в интерактивном режиме будет расти пропорционально интенсивности запросов λ, т.е.:

$$T_c^i = \lambda t^f. \tag{5.7}$$

Отметим, что при формировании запроса пользователь может использовать несколько поисковых признаков. Пусть N^q - количество поисковых признаков в запросе, τ- время, требуемое на подготовку части запроса, определяемой одним поисковым признаком, тогда время, затрачиваемое на формирование запроса в целом пропорционально N^q:

$$t^f = \tau N^q. \tag{5.8}$$

Из (5.7) и (5.8) получим:

$$T_c^i = \lambda \tau N^q. \tag{5.9}$$

С другой стороны также можно вычислить времена загрузки программы диалога t^d и доставки запроса t^s, которые имеют прямую зависимость, соответственно, с размерами программы и запроса, но обратную зависимость с пропускной способностью канала. Пусть L^p

- средний размер вызываемой программы диалога, L^q - средний размер сформулированного запроса и C - пропускная способность канала связи. Тогда для t^d и t^s можем писать следующие выражения:

$$t^d = \frac{L^p}{C}, \tag{5.10}$$

$$t^s = \frac{L^q}{C}. \tag{5.11}$$

Если вставить выражения (5.10) и (5.11) в формулу (5.6) получим:

$$T_c^a = \frac{L^p + L^q}{C}. \tag{5.12}$$

С учетом выше сказанных, можно сделать вывод, что временная разница $\Delta T = T_c^i - T_c^a$ будет расти пропорционально интенсивности запросов, так как в автономном режиме время, затрачиваемое на процесс взаимодействия не зависит от интенсивности запросов, а в интерактивном режиме пропорционально интенсивности запросов λ. Другими словами, эффективность использования каналов связи при интерфейсе пользователя с распределенной структурой пропорционально растет с ростом интенсивности пользовательских запросов. Таким образом, с уверенностью можно сказать, что при слишком большой интенсивности пользовательских запросов ΔT стремится к бесконечности, т.е.:

$$\lim_{\lambda \to \infty} \Delta T \to \infty. \tag{5.13}$$

Рассмотрим зависимость временной разности ΔT от нагрузки системы на примере академической сети Азербайджана и приведем сравнительный расчет вышесказанных параметров. Пусть среднее

число поисковых признаков в запросе $N^q=5$, доля времени на формирование части запроса, определяемый одним поисковым признаком $\tau=5$*сек.*, пропускная способность канала $C=33.6$*Кбит/сек.*$=3360$*Байт/сек.*, средний размер программы диалога $L^p=25.6$ *Кбайт* и средний размер запроса $L^q=64$ *байт*. Тогда

$$T_c^i = \tau N^q = 5 \cdot 5 = 25 \text{сек.}$$

$$T_c^a = \frac{25600+64}{3360} = 7{,}5 \text{сек.}$$

На рисунке 5.3 приведены результаты сравнительного расчета времени реакции системы, для различных вариантов реализации интерфейса пользователя в сети, проведенного на имитационной модели вышеупомянутой сети. Здесь ρ - географическая распределенность нагрузки, которая определена как

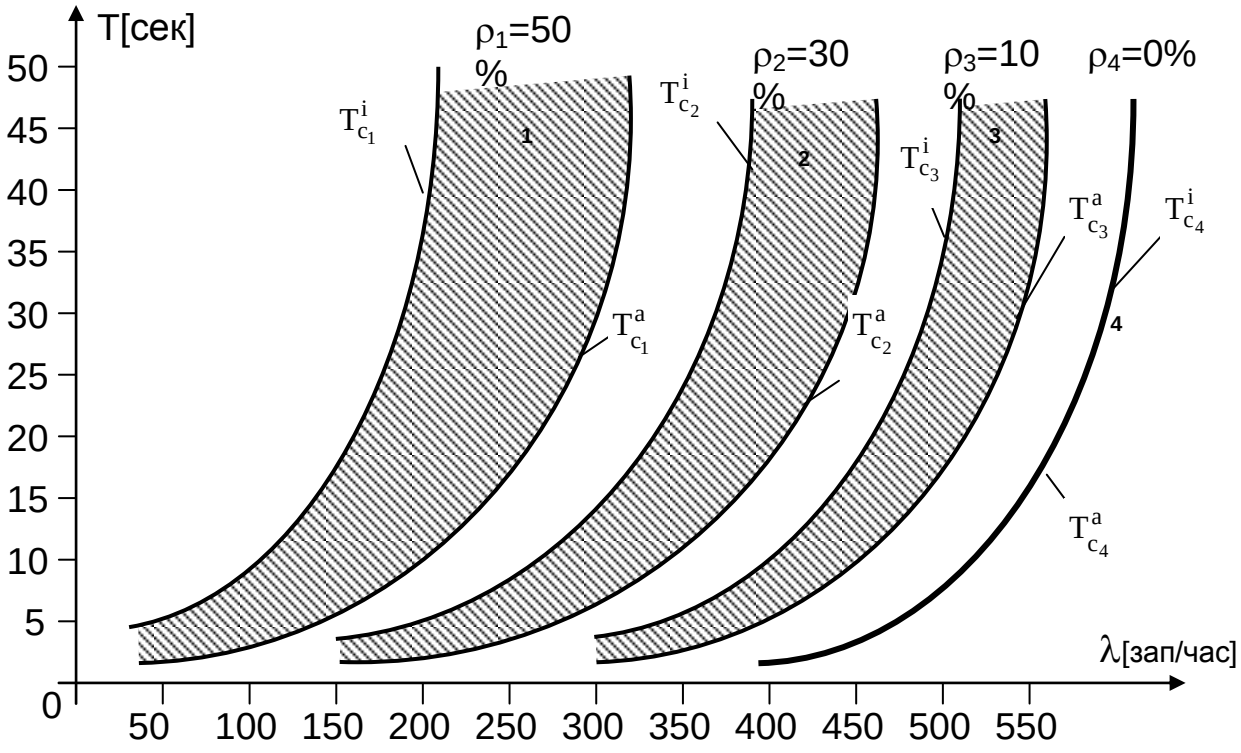

Рис.5.3. Зависимость времени реакции от нагрузки

$$\rho = \frac{\lambda_p}{\lambda} \cdot 100\%, \qquad (5.14)$$

где λ_p - интенсивность запросов к системе, реализуемых географически удаленными сетевыми ресурсами.

Как видно из рисунка, эксперименты проведены на четырех этапах с разными значениями параметра распределенности нагрузки:

I. $\rho_1 = 50\%$ - означает, что половина пользовательских запросов реализуется удаленными сетевыми ресурсами;

II. $\rho_2 = 30\%$

III. $\rho_3 = 10\%$

IV. $\rho_4 = 0\%$ - означает, что все запросы являются локальными, т.е. реализуются локальными сетевыми ресурсами.

Кривые, расположенные слева от заштрихованных областей соответствуют случаю, когда интерфейс пользователя является локальным, т.е. он обращается к локальным сетевым ресурсам, а расположенные справа - распределенным, т.е. пользователь обращается к удаленным сетевым ресурсам. В четвертом случае кривая одна (оба кривые совпали друг на друга), что показывает отсутствие потока распределенных запросов.

Несмотря на то, что пример академической сети подтверждает теоретический вывод о прямой зависимости временной разности ΔT от нагрузки системы, но эта зависимость неявная, поэтому следует дать анализ графической иллюстрации (рис.5.3):

- линейность зависимости временной разности ΔT от интенсивности запросов λ не видна, т.к. кривые описывают время реакции системы, которая включает в себя также времена задержек в каналах;

- главным воздействующим параметром на временную разность ΔT является не сама нагрузка, а ее часть, определяемая долей распределенных пользовательских запросов (в случае ρ_1, ρ_2, ρ_3) и, следовательно, заштрихованная область, ширина которой соответствует указанной разности, сужается слева направо с уменьшением распределенности, и в случае $\rho_4 = 0$ полностью исчезает;
- изменение ширины заштрихованной области в пределах каждого случая соответствует изменению временной разности в зависимости от нагрузки.

5.4. Поисковый алгоритм

Пользователь для удовлетворения своих запросов использует имеющийся в его распоряжении интерфейс между ним и сетевым ресурсом, предназначенный для формирования запроса и получения желаемого сетевого ресурса. Важным здесь считается тот факт, что пользователь не имеет подробного представления об информационном обеспечении системы - он только знает что ему нужно.

Так как стоимость сетевых и системных ресурсов, а также каналов связи очень высока, одновременное обслуживание многочисленного потока пользовательских запросов приводит к перегрузке информационных сетей, сетевых ресурсов и каналов связи и, в конечном счете, производительность информационных, в.т. поисковых систем сильно снижается.

Поэтому ИПС помимо удовлетворения запросов пользователей, должна решать проблему оптимальной адресации запросов

пользователей к источникам и другим информационным системам. Интерфейс между пользователем и сетевым ресурсом организуется с применением браузеров и поисковых машин, которые по своему принципу построения являются системами управления базами данных, содержащих информацию о сетевых ресурсах и об их адресах расположения.

В системах, где используется база знаний об информационных ресурсах, пользовательский запрос формально можно представлять четверкой

$$Q=\{D,S,T,P\}, \qquad (5.15)$$

где D – множество информационных ресурсов, т.е. документов, S – множества свойств, т.е. поисковых признаков документов (ключевых слов, терминов, синонимов, тезаурусов и т.д.), T – временные характеристики документа, P – пространственные характеристики документа.

Так, например, если требуется литература по информационно-поисковым системам за последние три года на русском языке, то по данной модели запрос представляется в следующем виде:

q_i - запрос,

d_i - информационные ресурсы, т.е. документы,

s_i - ключевые слова, типа "поисковая система", "информационный поиск" и т.д.,

t_i - временной период 2000-2003 гг.,

p_i - URL-адрес или адрес географического расположения сайтов, содержащих русскоязычные документы.

Однако полное перечисление предлагаемой модели для Интернет, оказывается практически невозможным.

Положение облегчается, если вместо полного описания используется база знаний о ресурсах. Для представления такого типа знаний используем реляционную модель с нечеткими элементами, т.е. нечеткими отношениями предпочтения сетевых ресурсов (NR) по документам (D), свойствам (S), временным (T) и пространственным (P) характеристикам.

$$NR \times D,S,T,P:\ \varphi(NR,\ D,S,T,P) \to [0,1]. \qquad (5.16)$$

С другой стороны, соответствие сетевых ресурсов к объектам, свойствам, временным и пространственным характеристикам можно представить нечеткими отношениями следующего типа:

$$G_q(X): \mu(x) \to [1,0], X = \{NR_1, NR_2, ..., NR_n\}, \quad q \in \{D,S,T,t\} \qquad (5.17)$$

Пользуясь методом Беллмана-Заде для решения классической задачи нечеткого математического программирования, получим:

$$\mu^{opt.}(x) = max\{min\{\mu_D(x), \mu_S(x), \mu_T(x), \mu_P(x),\}\} \qquad (5.18)$$

Для определения значений φ и $\mu_z(x)$ применяется метод экспертных оценок, однако во время эксплуатации их значения подвергаются адаптации, т.е. обучению.

Таким образом, выбор требуемого сетевого ресурса сводится к выбору сетевого ресурса, наиболее подходящего заданному списку элементов из $\{D,S,T,P\}$. На языке нечетких множеств данная задача сводится либо к выбору эффективных альтернатив, либо к задаче многокритериальной оптимизации с нечеткими критериями, которая решается минимаксным методом.

Адаптация значений осуществляется методом поощрения и штрафов. Суть применения данного метода заключается в

следующем. Если при выборе пользователь не находит искомого ответа, то ему предоставляется возможность возврата и формирования нового списка из элементов $\{D,S,T,P\}$. Если при повторном входе в систему с новым списком элементов, запрос пользователя будет удовлетворен, то осуществляется сравнение нового списка со старым и определяются исключенные из старого списка элементы. Затем отношения между сетевым ресурсом и исключенными элементами подвергаются штрафованию.

5.5. Определение наилучшего направления поиска

Поисковые машины всегда предоставляют пользователям не одно, а множество направлений поиска по введенному пользователем набору признаков поиска, в частности ключевых слов. Несмотря на непрерывную адаптацию баз знаний поисковых машин, из-за огромности объема и динамичности информационных ресурсов Интернет, не обеспечивается аппарат выбора единственного направления поиска в Web пространстве.

Для осуществления выбора наилучшего направления поиска определим оценку релевантности запроса пользователя к направлениям поиска следующим образом. Пусть ПМ содержит таблицу отношений поисковых признаков к направлениям, в частности к Web-сайтам, в виде двухмерной матрицы $\|k_{ij}\|_{n\times m}$, где k_{ij} - неотрицательные числа (рис.5.4), характеризующие вес i-го признака к j-му направлению, которые вычисляются на основе соответствующих наборов степеней соответствия поисковых признаков к Web-сайтам:

$$k_{ij} = \frac{z_{ij}}{\sum_{j}^{m} z_{ij}}, i = \overline{1,n},\ \ j = \overline{1,m}, \qquad (5.19)$$

где z_{ij} - коэффициенты важности поисковых признаков для Web-сайтов.

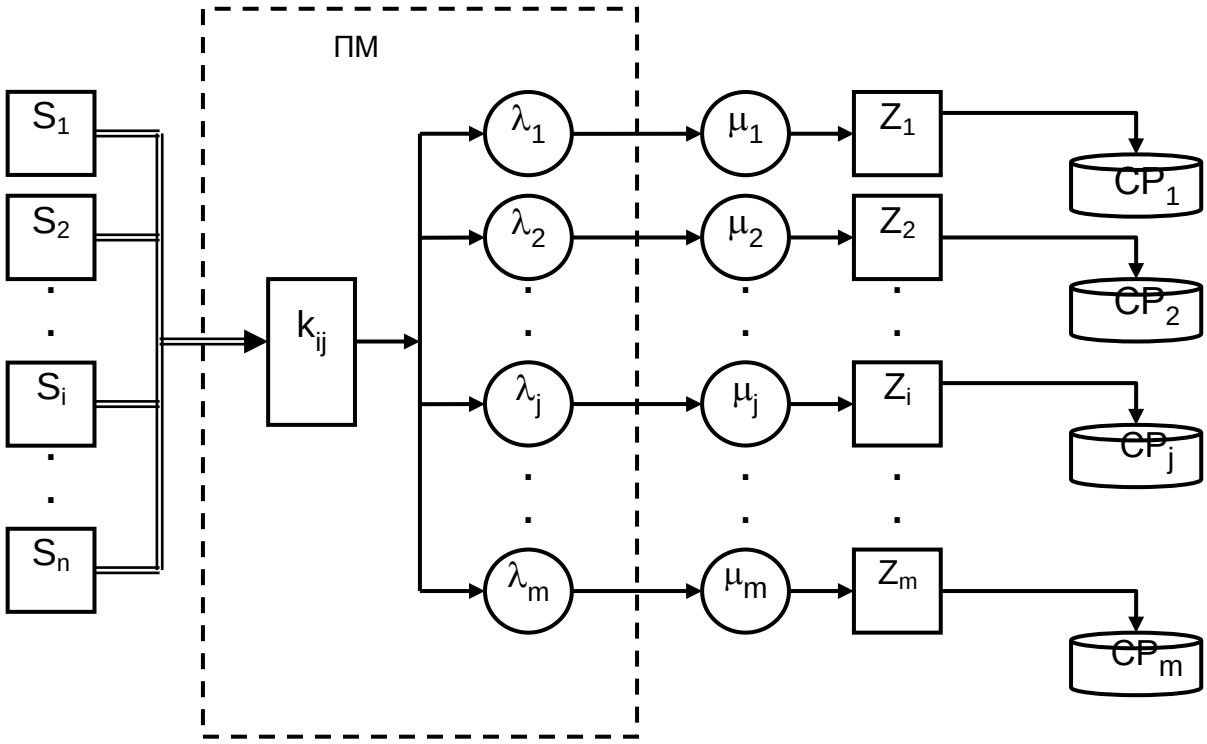

Рис.5.4. Схематическое описание процедуры поиска

Выбор направления поиска сводится к определению степеней значимости направлений μ_j, $j = \overline{1,m}$, характеризующие способности Web-сайтов удовлетворить запросы по совокупности поисковых признаков, которые определяются следующим образом:

$$\mu_j = \sum_{i}^{n} k_{ij},\ j = \overline{1,m}, \qquad (5.20)$$

С другой стороны определим также запрос пользователя как набор чисел $S = \{s_i, i = \overline{1,n}\}$, соответствующих важности ключевых признаков в запросе пользователя. Тогда значимости направлений поиска для запроса пользователя можно определить как:

$$\lambda_j = \sum_i^n \lambda_{ij},\ j = \overline{1,m}, \qquad (5.21)$$

здесь

$$\lambda_{ij} = \min_j \{(s_i - k_{ij})^2, i = \overline{1,n},\ j = \overline{1,m}\},$$

где λ_{ij} - среднеквадратичное отклонение запроса пользователя от направлений по отдельным признакам.

Релевантность между запросом пользователя и направлением поиска можно оценить по следующей формуле:

$$R = \frac{1}{\lambda} \sum_j^m \lambda_j r_j\ , \qquad (5.22)$$

здесь

$$r_j = \frac{\lambda_j}{\mu_j - \lambda_j},\ j = \overline{1,m},$$

где r_j - отклонение направлений поиска от запроса пользователя. Чем меньше r_j, тем больше степень релевантности направления к запросу. В рамках приведенных определений обеспечение максимальной релевантности поиска в зависимости от структуры построения

поисковой машины сводится к минимизации (5.22) по $\lambda_j, j=\overline{1,m}$, удовлетворяющих условию $\lambda=\sum_{j}^{m}\lambda_j$.

Для демонстрации адекватности предложенного алгоритма выбора наилучшего направления приведем пример. Пусть $W=\{w_1,w_2,w_3,w_4,w_5\}$ - Web-сайты в системе (m=5), которые определяются поисковыми признаками $T=\{t_1,t_2,t_3,t_4,t_5\}$ (n=5). Коэффициенты важности признаков для Web-сайтов задается следующей матрицей:

$$\{z_{ij}\}_{n\times m}=\begin{pmatrix} 0 & 0.5 & 0 & 0.9 & 0.55 \\ 0.8 & 0.9 & 0 & 0 & 1 \\ 0 & 0 & 0.6 & 1 & 0.8 \\ 0.65 & 0 & 0.5 & 0.95 & 0.9 \\ 0.6 & 0.6 & 0.9 & 0.8 & 0 \end{pmatrix}.$$

Важность этих признаков в пользовательском запросе задается вектором: $S=\{0.9,\ 0,\ 0.9,\ 0.95,\ 0.6\}$.

По (5.19) получим следующую матрицу весов поисковых признаков для направлений:

$$\{k_{ij}\}_{n\times m}=\begin{pmatrix} 0 & 0.26 & 0 & 0.46 & 0.28 \\ 0.3 & 0.33 & 0 & 0 & 0.37 \\ 0 & 0 & 0.25 & 0.42 & 0.33 \\ 0.22 & 0 & 0.17 & 0.32 & 0.3 \\ 0.21 & 0.21 & 0.31 & 0.28 & 0 \end{pmatrix}.$$

Вычислим среднеквадратичное отклонение запроса от направлений по отдельным признакам согласно (5.21):

$$\{\lambda_j\}_m=\{0.48,\ 0.48,\ 0.39,\ 0.19,\ 0.32\}.$$

Значимости направлений по совокупности поисковых признаков μ_j, $j = \overline{1,m}$ по формуле (5.20) получают следующие значения:

$$\{\mu_j\}_m = \{0.72, 0.80, 0.73, 1.47, 1.29\}.$$

Таким образом, согласно (5.22) отклонение направлений от запроса в целом будет:

$$\{r_j\}_m = \{2, 1.5, 1.13, 0.15, 0.34\}.$$

Видно, что четвертое направление имеет минимальное отклонение ($r_4 = 0,15$), т.е. четвертое направление является наиболее подходящим направлением запросу пользователя.

Таким образом, рассмотренные в этой главе принципы организации интерфейсов пользователя с системными ресурсами в зависимости от пропускной способности канала связи и методы оценки эффективности взаимодействия пользователя с поисковой системой, а также результаты исследования интерфейсов пользователя с ИПС и сетевыми ресурсами, зависимости эффективности использования канала связи от параметров интенсивности и распределенности пользовательских запросов, дает основание полагать, что оптимизируя структуру интерфейсов можно повысить эффективность ИПС и использования канала связи путем сокращения времени реакции системы, снижения нагрузки системы и сетевых ресурсов, так же загрузки канала связи.

Предлагаемый поисковый алгоритм позволяет из множества альтернативных вариантов информационных ресурсов выбрать наиболее предпочтительный ресурс, т.е. наиболее эффективную альтернативу на основе их свойств, временных и пространственных характеристик с помощью методов многокритериальной оптимизации

с нечеткими критериями. А метод определения наилучшего направления поиска обеспечивает выбор единственного наиболее подходящего направления поиска информации в Web-пространстве среди множества направлений, выданных пользователю поисковой машиной. Данный метод может быть использован как для повышения эффективности работы существующих ИПС, так и при разработке новых алгоритмов поиска информации в распределенных информационных системах.

VI ГЛАВА.
ИНФОРМАЦИОННО-ПОИСКОВЫЕ СИСТЕМЫ ИНТЕРНЕТ

6.1. Первая информационно-поисковая система на базе академической сети Азербайджанской части Интернет

При использовании услуг многочисленных поисковых систем из географически удаленного расстояния, пользователи встречаются с некоторыми неудобствами. С одной стороны, подключение к ним требует большие и временные сетевые ресурсы, такие как прием и передача мультимедийной информации, интерактивный сеанс с сервером поисковой системы и.т.д., с другой стороны, к известным поисковым серверам постоянно обращаются многочисленные пользователи Интернет, что существенно увеличивает трафик, забивает каналы связи.

Нельзя сказать, что существующие поисковые сервера содержат исчерпывающую информацию и полноценны в смысле удовлетворения всех потребностей пользователей. Так поисковые роботы ИПС не охватывают всех информационных ресурсов Интернет не всегда индексируют их на желаемом уровне. А зарубежные поисковые роботы из-за объективных и субъективных причин не всегда доходят или вообще не доходят к информационным ресурсам периферийных сайтов Интернет, в том числе и Азербайджанских [47].

Исходя из этого в конце 90-х годов XX века под руководством автора, являющегося руководителем и главным сетевым администратором академической сети, был разработан и осуществлен проект для решения задачи создания поискового сервера в

Азербайджанской части Интернет, включающего образы как республиканских, так и зарубежных информационных ресурсов [52]. Следует отметить, что академическая сеть объединяла все научно-исследовательские институты и другие организации Национальной Академии Наук Азербайджана [54, 55].

Разработанная ИПС включает в себя необходимые механизмы индексирования, поиска и интеллектуального интерфейса, т.е. диалога с пользователями на естественном языке. Программа робота собирает информацию об информационных ресурсах по всей сети Интернет и создает специальную структурированную базу данных, которая содержит образ информационных ресурсов всемирной паутины.

Согласно описанной в предыдущих главах структуре поисковых систем, ИПС, разработанная в Азербайджанской части Интернет также является комплексом программно-технических средств и баз данных (рис.6.1). Из рисунка видно, что кроме подсистем стандартных ИПС, указанных в предыдущих главах (программа робота, БД индексов, поисковая машина, интерфейсная программа, клиентская программа), в состав данной ИПС входят следующие подсистемы [36, 49]:

- ***Исходная база индексов информационных ресурсов*** - эта временная база, которая создается поисковым роботом в процессе индексирования.

- ***Программа-преобразователь базы данных*** оптимизирует структуру исходной базы и преобразует ее в конечную индексную базу данных, которая в дальнейшем используется поисковой машиной в процессе поиска.

- ***Каталогизатор*** - определяет тематику информационных ресурсов Интернет, создает их тематический каталог, подобной

библиотечному тематическому каталогу. Обычно данный процесс в основном выполняется человеком вручную, однако в последнее время появилась тенденция разработки интеллектуальных методов автоматического создания тематических каталогов.

Рис.6.1. Структура первой ИПС академической сети Интернет

- ***Тематический каталог*** - база данных, являющая результатом процесса каталогизации информационных ресурсов, которая доступна как напрямую через интерфейсные программы, так и с помощью поисковой машины в результате поиска по ключевым словам.

6.2. Принцип работы информационно-поисковой системы академической сети

Программа ***робота***, которая является одной из основных компонентов поисковой системы, независимо от других подсистем регулярно, один раз в неделю (этот период определяется администратором поисковой системы и может быть изменен) "прогуливает" по информационному пространству Интернет собирая информацию об информационных ресурсах, составляет индексы и создает исходную базу данных, т.е. образы информационных ресурсов по индексированным информационным ресурсам (Web-серверам, FTP-архивам, хранилищам программ и т.д.). Образы информационных ресурсов как обычно включают в себя термины и ключевые слова, встречаемые в документах, файлах и т.д., а также их URL-адреса.

После завершения работы поисковый робот передает управление ***программе, которая в свою очередь преобразует структуры*** полученной исходной базы в специальную структуру, позволяющую уменьшить время доступа к данным, а также сократить размер базы. Такая структура эффективна именно для баз данных, содержащих индексы ИПС (см. следующий раздел). Составленная таким образом база индексов в дальнейшем используется всеми подсистемами ИПС.

Интерфейсная программа является сервисной CGI-программой, позволяющей пользователю подготовить свои запросы и передать поисковой машине. Пользователь может обращаться к этой программе с помощью ***клиентских программ***, в качестве которых используется обычные браузеры для Интернет: Netscape, Internet Explore и т.д. Интерфейсная программа выводит на экран окно (рис.6.2), содержащее поле для ключевых слов, ссылки на часто используемые

сайты, кнопки переключения на специальные режимы, тематические указатели и т.д. На экране есть также кнопка ***«Find it now»***, нажатие которой инициирует процесс поиска, который также запускается нажатием клавиши Enter.

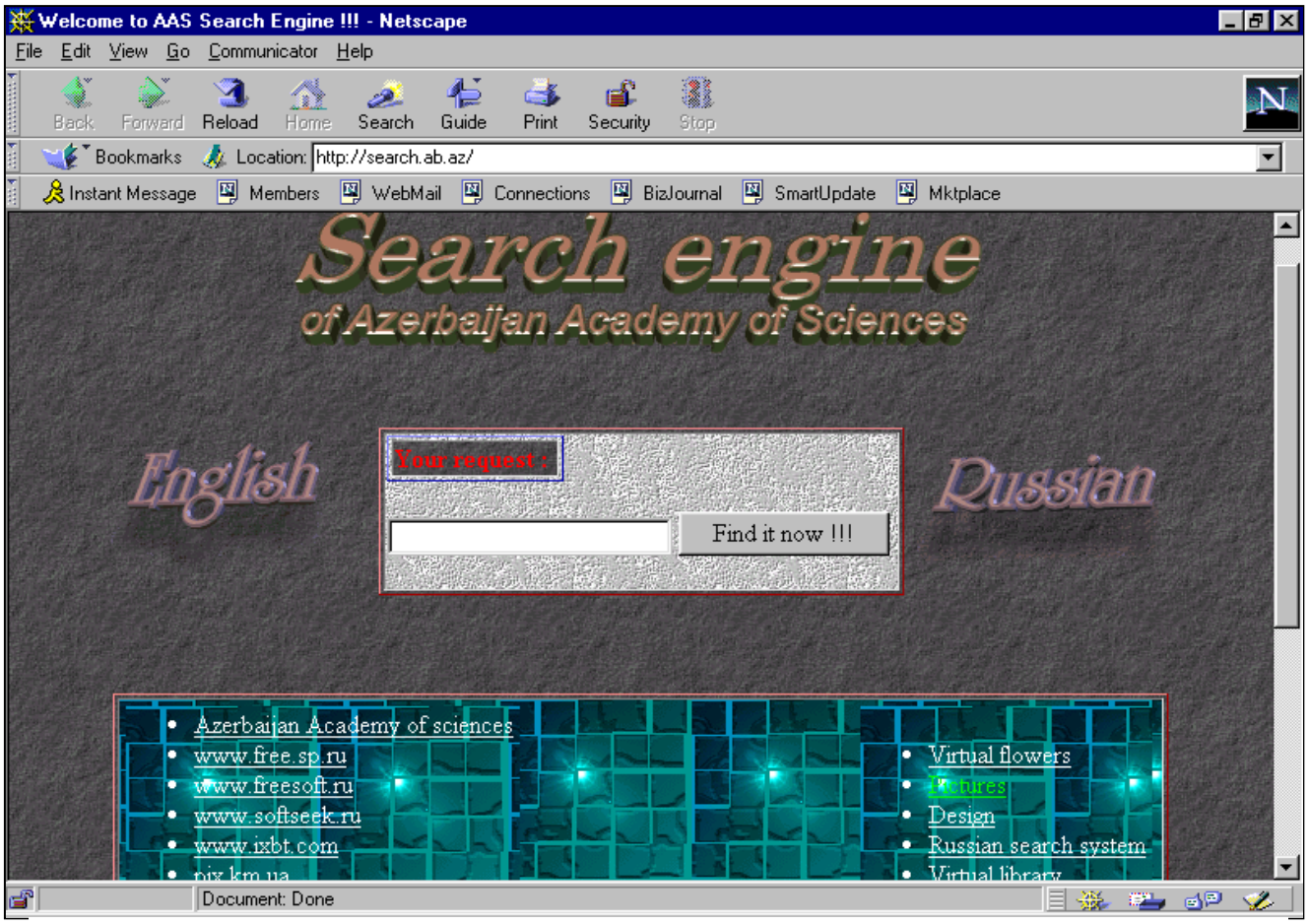

Рис.6.2. Интерфейсное окно поисковой системы

После инициализации процесса поиска ***поисковая машина*** получает запрос пользователя от интерфейсной программы. Запрос индексируется подобно индексированию документов поисковыми роботами. После чего поисковая машина начинает вести поиск ключевых слов запроса в базе индексов. Результат поиска возвращается CGI-программе, входящая в состав интерфейсной программы пользователя, которая в свою очередь передает его пользователю в специальной форме (рис.6.3). Информация о

найденных ресурсах выводится на экран в виде списка порциями по 25 документов. Для перехода к следующей порции документов можно использовать кнопку «Next Page», а для перехода на предыдущие - кнопку «Last Page».

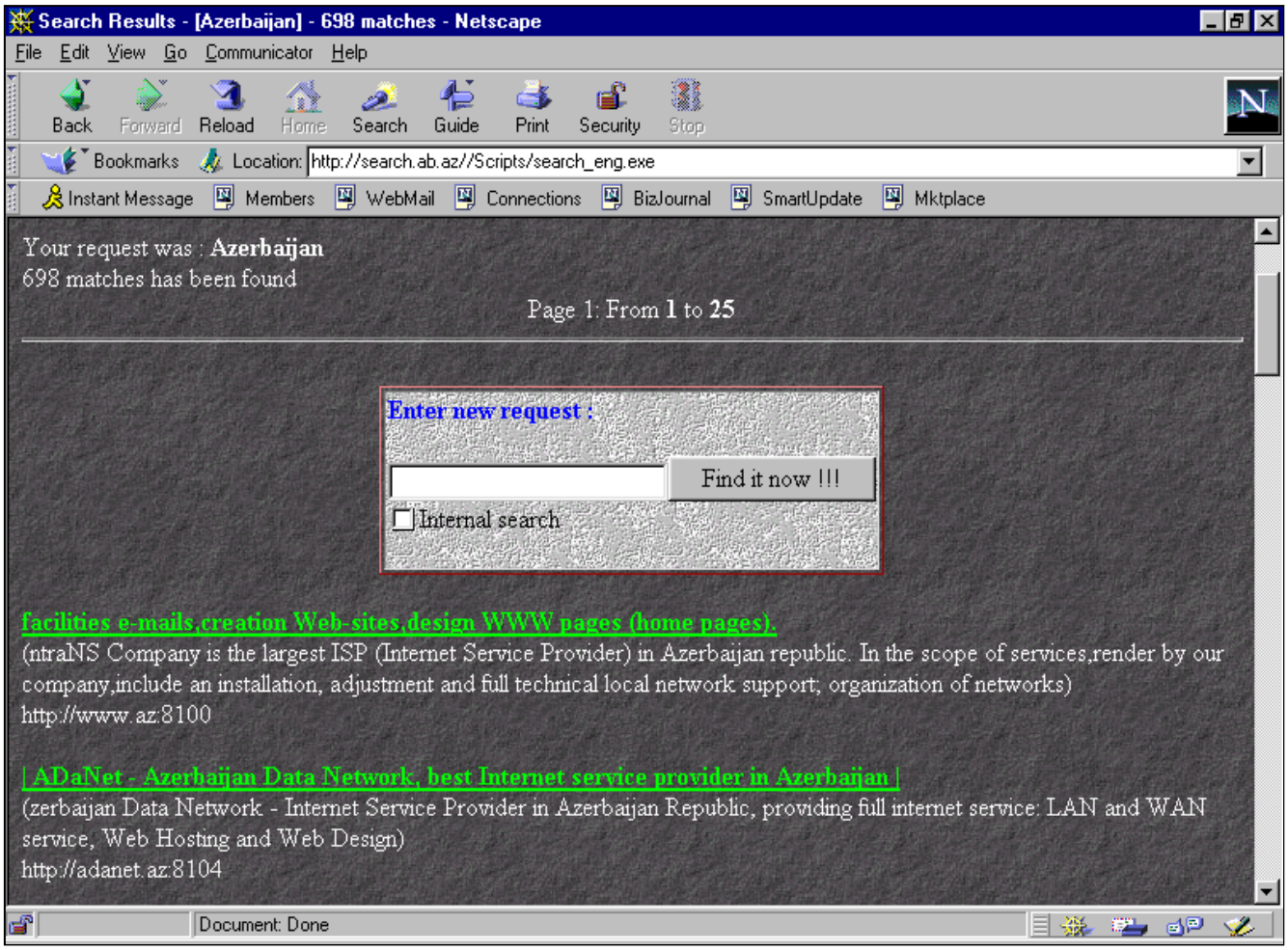

Рис.6.3. Окно результатов поиска (Запрос - ключевое слово «Azerbaijan»)

Одновременно, выведенные на экран результаты поиска также записываются во временную базу данных - в кеш-память. В дальнейшем, при вторичном обращении к ИПС с таким же запросом, CGI-программа использует эти записи и выдает ответ пользователю не обращаясь к поисковой машине.

6.3. Поисковый робот информационно-поисковой системы Интернет

Из-за астрономического увеличения объема данных Интернет, а также отсутствия систематизации информационных ресурсов возникают определенные трудности во время поиска источников информации, документов, файлов и т.д., содержащих нужные данные [49, 50, 52].

Как было сказано выше, для облегчения работы пользователей при поиске нужной им информации применяются ИПС, при проектировании, создании и эксплуатации которых особое значительное место занимают поисковые роботы, осуществляющие процедуры индексирования, т.е. создания «образов» информационных ресурсов. Поисковые роботы выполняют огромную работу, «прогуливая» по всем сайтам Интернет и собирая информацию о них. Они должны уметь определять какие страницы можно индексировать, а какие нет. Кроме этого, от них зависит степень важности ключевых слов, выбираемых из документов, к самим документам, которая определяет качество результатов поиска.

В целом поисковые роботы предназначены для загрузки информационных ресурсов из удаленных узлов, индексирования и записи их в базу согласно внутреннему формату, который в дальнейшем применяется в поисковых системах при создании и эксплуатации их баз данных. Рассмотрим алгоритм работы поискового робота, разработанного для Азербайджанской части Интернет и его поведение в типичных ситуациях, встречаемых во время индексирования информационных ресурсов Интернет.

При обращении к Web-сайтам для загрузки документов поисковый робот использует HTTP протокол. В начале на входе процедуры требуется файл со списком URL-адресов серверов (Servers.lst), который составляется администратором системы и используется поисковым роботам при индексировании, а на выходе он выдает файл (Source.txt) следующей структуры:

#<1-ый URL>

<Description>

<1-ое слово> <позиция 1>, … , <позиция n>

<2-ое слово> <позиция 1>, … , <позиция n>

…

<i-ое слово> <позиция 1>, … , <позиция n>

…

<последнее слово> <позиция 1>, … , <позиция n>

#<2-ой URL>

<Description>

<1-ое слово> <позиция 1>, … , <позиция n>

<2-ое слово> <позиция 1>, … , <позиция n>

…

<i-ое слово> <позиция 1>, … , <позиция n>

…

<последнее слово> <позиция 1>, … , <позиция n>

…

Здесь *<1-ый URL>, <2-ой URL>* - URL-адреса Web-страниц типа *http://www.ab.az* или *http://www.ab.az/Institutes/Mathematics.html*, *<1-ое слово>*, *<2-ое слово>* и т.д. - ключевые слова из вышеуказанных страниц, *<Description>* - краткое описание страницы, *<позиция>* -

позиция ключевого слова на Web-странице (используется для поиска словосочетаний). Символы «<» и «>» используются в качестве разделителя между элементами при описании и, естественно, отсутствуют в выходном файле.

Никто не может гарантировать, что в БД иерархия URL-адресов сверху вниз будет сохраняться. Иначе можно сказать, что в БД строка *«#http://www.ab.az/Institutes/Mathematics.html»* вполне возможно встретится раньше, чем *«#http://www.ab.az/Institutes/»*, хотя по иерархии первая страница должна появиться после второй. Это означает, что логически ссылку на страницу *«#http://www.ab.az/Institutes/Mathematics.html»* можно встретить как на URL *«#http://www.ab.az/Institutes/»*, так и на других страницах более верхнего уровня. Естественно, если ссылка на одну страницу встречается раньше, чем другая, то, соответственно первая страница индексируется раньше, чем вторая.

Для создания окончательной индексированной базы индексов, сначала требуется сортировка исходной базы данных по иерархии URL-адресов, а потом необходимо применение логического метода сжатия при хранении БД. Структура баз данных, методы структуризации, сжатия, хранения данных и доступа к ним рассматривается в следующей главе.

6.4. Индексирование документов поисковыми роботами

Прежде чем приступить к проектированию и разработке поискового робота для индексирования Web-страниц, следует тщательно изучить все международные стандарты и соглашения относительно WWW, Web- серверов, протокола HTTP, языка HTML,

CGI-скриптов и структур Web - документов. На сегодняшний день существуют множество стандартов, соглашений и протоколов в данной области, которые влияют на процесс разработки приложений.

Одним из существующих стандартов является так называемый «стандарт исключения для роботов», который предназначен для запрета тем или иным поисковым роботам индексировать ту или иную страницу или весь сервер целиком. Этот стандарт все еще не является общепринятым и в данный момент находится в состоянии разработки. Несмотря на это, многие известные ИПС уже используют этот стандарт. Можно предположить, что данный стандарт получит широкое распространение в ближайшем будущем.

Согласно этому стандарту информация о запрете действий поисковых роботов хранится в файле /robots.txt из корневого каталога Web-сервера. Если данный файл отсутствует на сервере или он существует, но в нем не имеются записи, тогда это понимается как разрешение на индексирование всей информации на сервере. В этом случае роботы самостоятельно принимают решение, каким образом индексировать сайт, т.е. индексировать все страницы, начиная от первой (index.html и т.п.) или не индексировать вообще.

Следует отметить, если на сервере имеется файл /robots.txt и прописаны в нем запреты на индексацию каких-либо страниц, то примет ли робот эти запреты целиком, зависит от робота. Суть данного стандарта заключается в том, что роботы индексировали все, что им не запрещается, а запреты ставятся самим администратором Web-сервера. “Нормальные” поисковые роботы должны соблюдать эти правила.

При разработке поискового робота нами использован стандарт исключения для роботов и выполнения всех требований серверов.

Робот открывает файл /Robots.txt, ищет строки *“User-Agent:”*, включающие имя настоящего робота или “*”. Последний является маской имени и указывает, что все условия и запреты относятся ко всем роботам, в том числе данному роботу или группе роботов, куда может входить данный робот тоже. Если имеется хотя бы одна такая строка, то ищется соответствующая строка *“Disallow:”*, входящая в блок соответствующих *“User-Agent:”*. Название всех файлов и каталогов (возможно сервер полностью), встречаемых в строке *“Disallow:”* исключаются из списка URL-адресов при индексировании (или не индексируется сервер).

Если на индексируемом сервере отсутствует файл /Robots.txt или не найдена запись, которая относится к настоящему роботу, то робот индексирует все страницы сервера подряд.

В поисковых системах Интернет для индексирования используются два класса методов обхода Web-серверов. В первом варианте названия серверов задаются администратором поисковой системы, т.е. в этом случае какие сервера будут индексированы заранее известны и зафиксированы. Администратор вводит имена (URL-адреса) всех Web-серверов в определенный файл - Servers.lst, который является файлом текстового типа. Каждая строка (запись) представляет собой адрес отдельного сервера. Робот извлекает адрес первого сервера из этого файла и индексирует его, а потом извлекает адрес второго сервера и индексирует и т.д. В этом случае все внешние ссылки, ведущие на другие сервера, исключаются.

Второй способ отличается от первого тем, что здесь в файл Servers.lst вводится одна запись - URL-адрес стартового сервера из числа наиболее популярных Web-серверов, так как их популярные страницы, как обычно, содержат ссылки на URL-адреса,

соответствующие наиболее часто запрашиваемой информации на данном или других серверах. Далее, в процессе индексирования обрабатываются как внутренние, так и внешние ссылки. Таким путем обеспечивается рекурсивный переход на другие серверы и здесь заранее неизвестно, какие серверы будут индексироваться. В отличие от предыдущего варианта в данном случае направление «движения» робота не регулируется. В данном случае для ограничения области индексирования роботами, Web-пространства делится на поисковые зоны на основе доменной системы имен Интернет или кодов стран. Для индексирования этих зон можно использовать одну или несколько программ робота.

Следует также отметить, что в первом варианте работа робота завершается логически как только будут обработаны все страницы на указанных серверах. В случае, если робот обнаруживает внешнюю ссылку, он заносит ее в дополнительный файл адресов серверов - в файл NewServers.lst для последующего использования администратором поисковой системы, в частности как прототип следующего файла-источника на Web-сервере для индексирования роботом.

А во втором способе процесс индексирования может заканчиваться в бесконечности (могут существовать бесконечно много ссылок, которые выводят на новую страницу или на новый сервер) или преждевременно (на последующих страницах не окажется ни одной ссылки на другие страницы или серверы). Поэтому, на каком то этапе необходимо прервать процесс искусственно (возможно вручную).

Для поискового робота ИПС академической сети реализованы оба метода обхода. Выбор методов осуществляется администратором

Web-сервера или сети. Здесь далее рассматривается первый способ индексирования.

Так процесс индексирования начинается с первой страницы (это обычно файл index.html) указанного администратором Web-сервера. Дальнейшее расширение области индексирования, то есть переход на другие страницы происходит путем добавления всех обнаруживаемых гиперссылок на Web-страницах в конец временного файла URL-адресов - Url.lst, который находится в начале процесса индексирования и куда записывается URL-адрес первого индексируемого Web-сервера, типа http://www.ab.az.

Далее программа-робот создает файл БД (Sourse.txt), перемещает туда первую строку из файла URL-адресов и начинает индексировать соответствующую страницу. После URL-адресов в файл добавляется краткое описание страницы, в качестве которого используется или содержимое Meta-taga NAME – Description или первые предложения из тела страницы. Все ключевые слова добавляются в файл БД сразу пос-ле строки URL, а обнаруживаемые гиперссылки в конец файла URL-адресов. После завершения обработки первой страницы, открывается страница, соответствующая следующей строке (уже первой!) в файле URL-адресов.

Во время индексирования для управления страницами робот использует META-таги HTML-языка, с помощью которых владельцы страниц задают определенные правила для роботов. Когда робот открывает страницу, прежде всего он ищет строки, содержащие META-таги, которые находятся в заголовке страниц. Для этой цели применяются атрибуты NAME и HTTP-EQUIV, каждый из которых может получить несколько значений.

Если на странице обнаруживается параметр “Robot” атрибута NAME, содержимое которого может быть ALL, NONE, INDEX, NOINDEX, FOLLOW, NOFOLLOW, то робот определяет, разрешается ли ему индексировать данную страницу и ссылки из этой страницы.

Параметр “Description” атрибута NAME содержит краткое описание страницы. Если “Description” существует, то во время поиска робот выводит на экран пользователя его содержимое с целью краткого описания страницы. Если робот обнаруживает параметр “Keywords” атрибута NAME, то он в качестве ключевых слов использует также содержимое этого параметра. Параметр “URL” атрибута NAME позволяет роботу узнавать является ли страница обычной страницей или страницей, порождаемой CGI-скриптом. Во втором случае, робот пропускает данную страницу и переходит на следующую страницу.

С помощью значения параметра “Expires” атрибута HTTP-EQUIV робот определяет, изменилась ли данная страница после того, как робот последний раз индексировал ее, путем сравнения даты последнего индексирования и содержимого данного параметра. Если изменение было, то страница заново индексируется, иначе она пропускается. Содержимое параметров “Content-Type” и “Content-language” атрибута HTTP-EQUIV позволяют роботу определить кодировку и язык данной Web-страницы. Определение кодировки необходимо как при выдаче сообщения на экран, так и при поиске. Определение языка документа дает возможность роботу из документов разного языка делать выбор по желанию пользователя.

Необходимо также отметить, что информация на разных серверах может находиться в разных кодировках. В этом случае, ключевое слово, введенное пользователем поисковой системы, может не

совпадать со словом, находящимся в базе, если у них разные кодировки, несмотря на то, что по смыслу они являются идентичными.

Поэтому осуществляется преобразование кодировок содержимого всех страниц, загружаемых роботом в некую единую кодировку (скорее всего в WIN), прежде, чем записать их в базу данных. В программе пользовательского интерфейса во время поиска запрос пользователя также преобразуется в эту же кодировку и этим обеспечивается правильное сравнение ключевых слов из запросов пользователя и в базе данных.

Существует также проблема, связанная с IP-адресами сайтов, которая не решена в данной версии реализации робота, но продолжается исследование этой проблемы и результаты его будут отражаться в следующих версиях программы-робота. Суть этой проблемы заключается в том, что http адреса Web-серверов могут быть заданы как в числовом виде, т.е. в IP-адресах, так и в символическом виде, т.е. именами доменов. Так, визуально разные адреса могут указать на один и тот же сервер. Например, адреса

http://www.ab.az/index.html

и

http://195.239.219.30/index.html

роботом понимаются как разные адреса, хотя они являются адресами одного и того же Web-сервера. Эта проблема может привести к логической ошибке, так как робот может индексировать одну и ту же страницу два раза и есть вероятность дублирования информации в базе данных. С целью решения этой проблемы робот может использовать DNS услугу Интернет, которая входит в рамку дальнейших исследований.

6.5. Алгоритм загрузки Web-страниц

Теперь рассмотрим шаговый алгоритм работы поискового робота, который имеет следующий вид:

1. Обнулить значение *ServerPosition*.
2. Если значение *ServerPosition* меньше, чем количество записей в файле *Servers.lst*, то необходимо увеличить значение *ServerPosition* на 1, иначе перейти к **шагу 17**.
3. Запись по номеру *ServerPosition* из списка *Servers.lst* присвоить переменной *CurrentServer***.**
4. Содержимое переменной *CurrentServer* занести в конец файлов *Source.txt* и *Url.lst*.
5. Если отсутствует файл «*/Robots.txt*» на сервере, адрес которого определяется из содержимого *CurrentServer*, то перейти к **шагу 7.**
6. Если обнаружены записи, которые запрещают настоящему роботу индексировать сервера, то перейти к **шагу 2.**
7. Если файл (список) *Url.lst* является пустым, то перейти к **шагу 2**, иначе первую запись из файла *Url.lst* присвоить переменной *CurrentUrl* и удалитьее из файла.
8. Считывать файл Web-страницы по адресу, хранимому в *CurrentUrl* по HTTP в память.
9. Если в теле документа не найдены META-таги, которые запрещают индексировать или задают правила ограничения обработки страницы, то перейти к **шагу 12**.
10. Если существует запрет среди META-тагов на индексирование данной страницы, то перейти к **шагу 7.**

11. Если существует META-таг *Description*, то его содержимое записать в конец файла *Source.txt* и перейти к **шагу 14.**

12. Если содержимое заголовка *TITLE* не пусто, то его записать в конец файла *Source.txt* и перейти к **шагу 14.**

13. Первый абзац тела страницы записать в конец файла *Source.txt*.

14. Извлечь все ссылки из документа. Внешние ссылки записать в конец файла *NewServers.lst*, а внутренние в конец файла *Url.lst*.

15. Все слова из тела документа, включая мета-таг "*Keywords*", заголовок (*TITLE*) и описание (META-таг "*Description*") занести в конец файла *Source.txt*.

16. Перейти к **шагу 7**.

17. Выход

При описании алгоритма использованы следующие переменные:

– ***ServerPosition*** – целочисленная переменная, определяющая номер текущей записи в файле *Servers.lst*, которая определяет название сервера.

– ***CurrentServer*** – переменная строкового типа, в которой хранится URL-адрес индексируемого сервера в данный момент времени.

– ***CurrentUrl*** – переменная строкового типа. Она содержит URL-адрес Web-страницы, которая индексируется в данный момент времени.

Кроме этого, здесь используются файлы, назначение которых описано выше.

– ***Servers.lst*** - файл, содержащий список URL-адресов Web-серверов, которые будут индексированы на данном сеансе работы программы-робота. Он создается администратором ИПС.

– ***NewServers.lst*** - файл, содержащий список URL-адресов новых серверов, которые отсутствуют в файле *Servers.lst*, но встречаются на индексируемых страницах в качестве ссылок. Создается роботом во время работы и может быть использован администратором ИПС в дальнейшем.

– ***Url.lst*** - временной файл, содержащий список URL-адресов страниц, индексируемых во время сеанса работы. Создается в начале сеанса работы роботом, а в конце сеанса работы он должен быть пустым и удаляется с диска.

– ***Source.txt*** - файл БД. Создается роботом и постоянно обновляется при каждом сеансе работы. Его структура описана выше.

6.6. Анализ экспериментального исследования информационно-поисковой системы

Программы компонентов ИПС разработаны на языке Delphi 5.0 в среде Windows NT 4.0 на базе ПК-сервера. В состав системы включены «интеллектуальные» алгоритмы для индексирования и поиска. В последней версии, разработанный в 2000 году, для определения степени релевантности документов к пользовательским запросам применялись методы искусственного интеллекта и теории нечетких множеств [47, 49].

Для выбора терминов и ключевых слов из тела документа используется гибридный метод, включающий в себя статистический, множественный и лингвистический методы. В результате индексирования терминам привязываются коэффициенты важности, определяющие степени соответствия терминов содержанию документов.

Поисковый робот запускается один раз в неделю в ночное время, когда загрузка каналов связи является наименьшей, который индексирует практически все азербайджанские и множество зарубежных сайтов. В середине 2000 года объем базы данных индексов составлял 400-500 Мбайт, и увеличивался при каждом индексировании. Количество документов - Web-страниц, индексированных роботом превышает два миллиона.

Статистика роста обращений к системе за первый полугодовой период работы с начала ее функционирования (с начала 1999 года) приведена на рисунке 6.4.

Кроме этого были исследованы характеристики ИПС, такие как:

- время реакции;

- полнота поиска, т.е. охват информационных ресурсов массива;

- точность поиска, т.е. релевантность найденных документов к запросу пользователя.

Все эти характеристики определены путем усреднения по результатам выполнения 20 разных тестовых запросов без учета задержки в каналах связи. Во время проведения (в начале 2000 года) теста база индексов ИПС содержала свыше одного миллиона Web-страниц. В результате проведенных опытов была построена зависимость времени обработки запросов пользователей от количества документов в базе ИПС (рис.6.5).

Здесь T - время реакции системы на запросы пользователей, которое измеряется в минутах, k - количество индексов в базе. Из рисунка видно, что для обработки запроса с одним ключевым словом требуется меньше времени, чем с двумя, причем это особенно заметно при больших объемах базы.

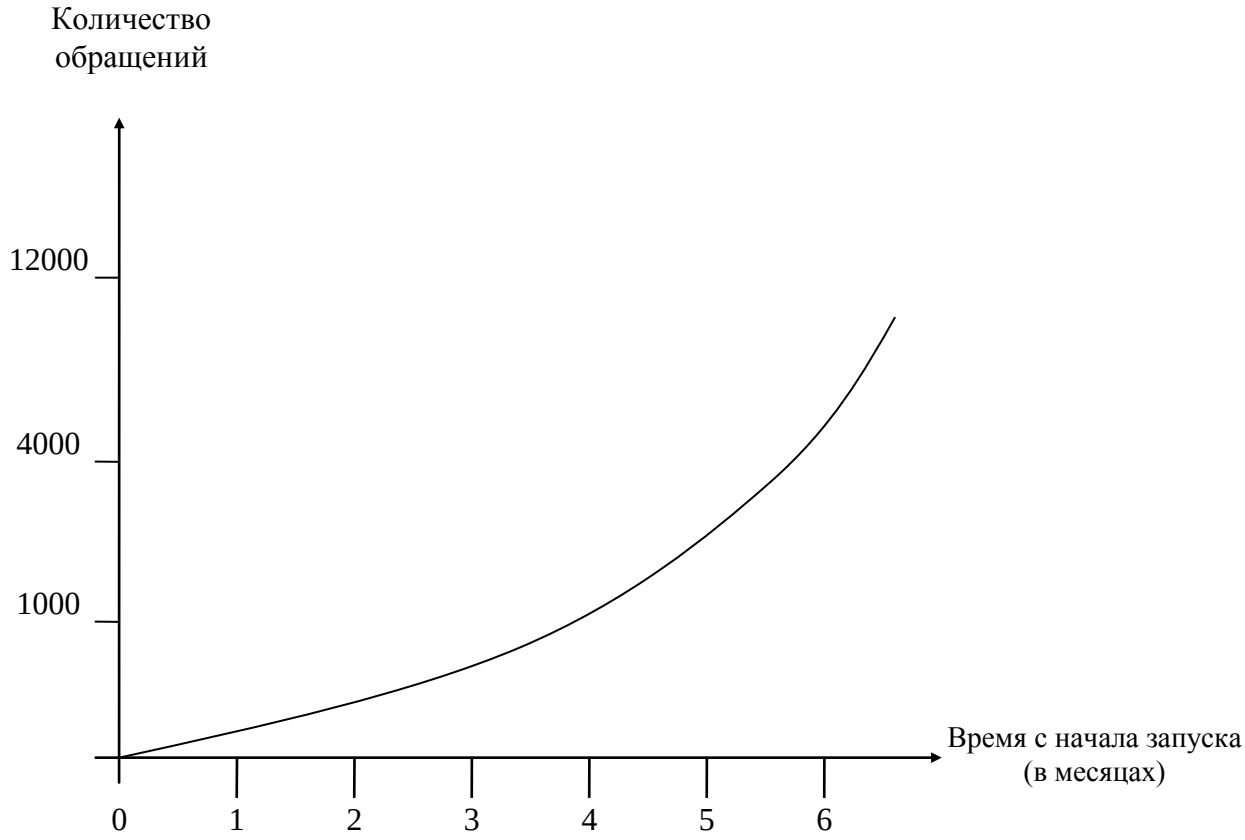

Рис.6.4. Рост обращений к поисковой системе

Кроме этого следует отметить, что время реакции системы также зависит от технического обеспечения сервера и интенсивность обращений. Данные измерения проведены на двухпроцессорном сервере NetFRAME Pentium II/400, при интенсивности обращений 100 запрос/мин без пиков.

Усреднение по результатам проведенных опытов показывают, что полнота выборки документов является 30%, а точность выборки - 60%. Значения характеристик полноты и точности выборки вызваны тем, что индексируется не только содержание документов, а также их названия и описания. Кроме этого, здесь используются дополнительные ключевые слова, приписанные к документам.

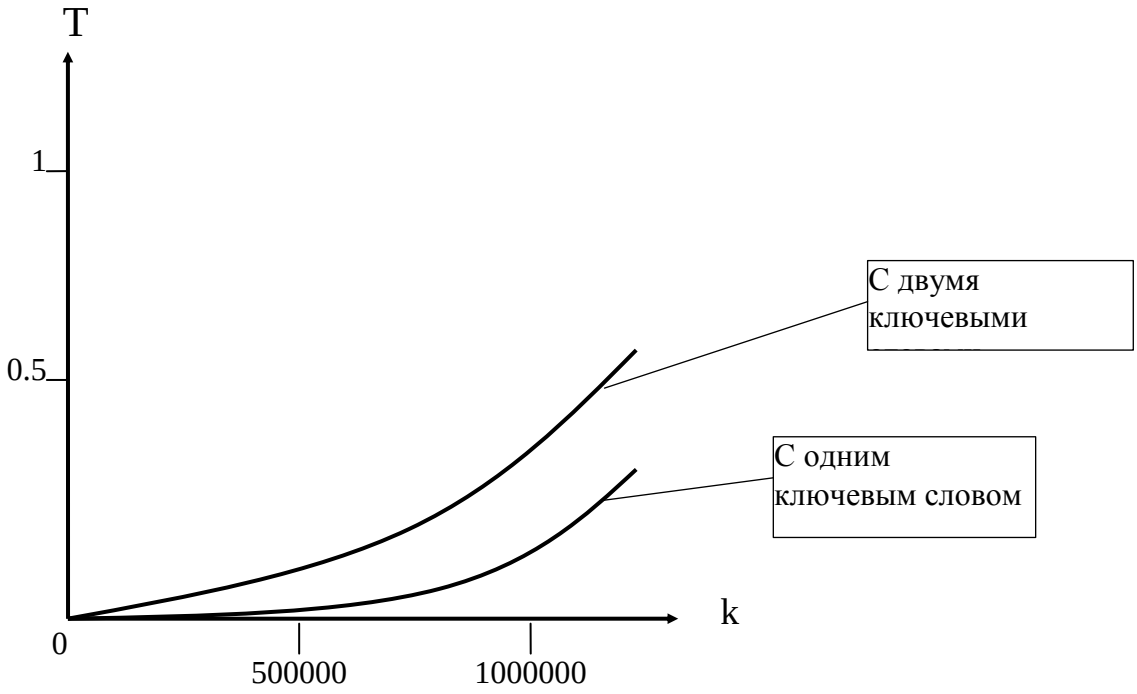

Рис.6.5. Зависимость времени реакции системы от количества индексов

Автоматический индекс включает в базу все вхождения терминов, за исключением служебных слов (глаголов, местоимений, часто используемых слов и т.д.), для чего используется файл стоп-слов.

VII ГЛАВА.

СТРУКТУРА ХРАНЕНИЯ ДАННЫХ В БАЗЕ ИНДЕКСОВ ИНФОРМАЦИОННО-ПОИСКОВЫХ СИСТЕМ ИНТЕРНЕТ

7.1. База индексов информационно-поисковых систем Интернет

Как было сказано выше, ИПС имеют свою базу данных, которая содержит образы документов, размещенных на Web-сайтах и серверах Интернет. Образы документов называются индексами, а база данных ИПС - базой индексов. Известно, что ИПС специальным образом (с помощью поисковых роботов или каталогизаторов) собирают информацию о миллионах информационных ресурсов, Web-страниц, документов, файлов, программ и т.д. Поэтому объемы баз индексов ИПС являются огромными.

Для своевременной реакции ИПС должны очень быстро обрабатывать запросы пользователей, вести поиск нужной информации и выдать ему определенный ответ. Огромный размер баз индексов одна из главных причин задержки обработки запросов и своевременной выдачи результатов поисковыми системами.

Несмотря на то, что объем информационных ресурсов Интернет растет астрономически, проблемы с их сбором и хранением практически отсутствует. Так современные технологии дисковых устройств - накопителей информации (DAS, NAS, SAN и т.д.), позволяют использовать достаточно огромную дисковую память для хранения информации и коллективного использования. Однако, с большим объемом баз индексов возникают трудности с ее обработкой во время поиска информации.

Следует отметить, что к поисковым серверам по сети могут обращаться одновременно сотни или тысячи пользователей. Поэтому, поисковый сервер загружается с различными запросами пользователей и процесс поиска в целом замедляется. Из-за такой задержки растет время занятости канала связи, в течении которого удаленный пользователь работает с ИПС.

Так как стоимость канала связи является дорогим и основным определяющим фактором для глобальных и распределенных сетей, то для достижения требуемой эффективности работы поисковой системы применяют специальные методы структуризации баз индексов и работы с данными в этих базах.

В данной главе рассматриваются вопросы структуризации баз индексов ИПС Интернет. Отметим, что информация, полученная в результате индексирования информационных ресурсов и хранимая в базе индексов, имеет специфический формат, с учетом которого можно оптимизировать базу. Исходя из этого предлагается специальная структура хранения данных в базе индексов ИПС Интернет, отличающая от традиционных структур.

Чтобы определить преимущества и недостатки предложенной структуры базы индексов необходимо показать ее отличительные особенности, для чего следует рассматривать наиболее подходящие методы и способы организации данных.

7.2.Традиционные структуры хранения данных

К настоящему времени разработаны различные структуры хранения данных на диске, эффективность которых в основном зависит от характера поставленной задачи. Эти структуры

оцениваются возможностями и производительностью используемых в них методов и способов организации доступа к данным [6, 24, 25, 68, 96]. Исследования показывают, что универсальные структуры хранения данных, которые одинаково удовлетворяли бы все требования решаемых задач и идеально подходили бы для различных целей практически отсутствуют. Другими словами, выбор или разработка оптимальной структуры базы сильно зависит от предметной области.

Традиционные БД, в основном, создаются на основе реляционных моделей. По следующим соображениям реляционная структура является не совсем подходящей для баз данных ИПС Интернет:

- реляционные БД неэкономично используют дисковое пространство, т.е. исходя из природы таких баз можно сказать, что содержимое полей могут иметь пустые или очень длинные строки;

- методы доступа к данным не совсем эффективны, так как поиск отдельных записей нужно вести в огромных файлах, поэтому приходится обрабатывать большие блоки данных.

Для повышения эффективности реляционных БД применяются дополнительные методы, однако не достигается желаемый эффект. Одним из таких методов является индексирование БД по ключевому полю. Отметим, что понятие “индексирование БД” отличается от понятия “индексирование информационных ресурсов поисковым роботом”. В первом случае, индексирование является сортировкой записей БД по ключевому полю в определенном порядке (обычно по алфавиту). При этом создается индексный файл, включающий в себя ключевое поле (или поля) и поле идентификационного номера записи

(RID). Во втором случае, индексирование означает сбор информации о документах, файлах и т.д., а также создание их образа.

Иногда индексные файлы оказываются настолько огромными, что при обработке этих файлов возникают большие трудности. Для решения этой проблемы применяется модифицированный вариант метода индексирования, который называется методом неплотного индексирования. При этом в индексном файле группируют все записи, содержания ключевых полей которых являются одинаковыми, а в поле указателей заносятся адреса начала или конца блока с одинаковыми индексами. Неплотный индекс намного улучшает индексный файл, но несмотря на это желаемый эффект также не достигается.

Следует также отметить, что если объем индексируемого файла БД является очень большим, то размер его индексного файла также будет очень великим. Учитывая тот факт, что в вышеуказанных методах индексирования (как плотных, так и неплотных индексах) используется последовательный метод просмотра индексов, то очевидно, что для последовательного просмотра даже одного индексного файла требуется значительное время.

В этом случае вместо метода последовательного просмотра используются другие методы выборки записей. Например, можно рассмотреть индексный файл как обычный файл данных и создать для него еще один индексный файл, т.е. создать индекс индексов. При этом второй индекс должен быть неплотным относительно первого индекса. Эту иерархию можно продолжить несколько раз, в зависимости от размера файла данных.

Вторым альтернативным методом доступа, который не использует последовательный способ просмотра, является

древовидный индекс. Частным случаем древовидного метода является структура типа B-дерева, которая состоит из двух частей: набора индексов и набора последовательностей (самих данных). Используются несколько уровней наборов индексов, с помощью которых исходный файл данных логически делится на адресуемые блоки и, таким образом, сужается область, содержащая искомую информацию.

Кроме этого, существуют методы хеширования и цепочек указателей данных. Основой метода хеширования (иногда его называют хеш-индексированием) является технология прямого доступа к хранимой записи. Прямой доступ осуществляется по хеш-адресам, которые вычисляются с помощью хеш-функций и привязываются к хранимым записям. Для вычисления адресов записей используются значения специального поля, которое называется "хеш-полем".

Метод цепочки указателей для адресации записей использует указатели на записи. Ключевое поле отделяется от других и записывается в отдельный файл, который называется родительским файлом, а все остальные поля записываются в другой, так называемый "дочерний файл". В родительском файле к ключевым словам привязывают указатели, которые указывают на остальную часть записей. В конце каждой записи имеется указатель на следующую запись, относящуюся к данному ключевому слову, а в конце последней записи указатель содержит ссылку на ключевое поле, что означает: больше нет записей с данным ключевым полем.

Применение вышеописанных методов хранения данных для организации структуры баз ИПС в Интернет является неэффективным. Так как хранимая информация в базах индексов ИПС

имеет специфическую структуру, то для оптимизации этих баз требуется разработка комплекса методов, которые должны выполняться поэтапно, позволять осуществлять удобный доступ к данным в базе и обеспечить эффективность этих структур.

7.3. Символьно-указательная структура файлов данных

Известно, что предметной областью задачи, рассматриваемой в данной работе является база индексов ИПС Интернет и его информационные ресурсы. Для повышения эффективности таких баз необходимо разработка соответствующей структуры хранения данных.

После подробного изучения предметной области, т.е. формата и содержания данных в таких базах, была предложена специальная структура хранения данных, которая по мнению автора является наиболее оптимальной для представления и хранения описаний информационных ресурсов Интернет [52, 56].

Предлагаемая структура с помощью специальных методов и алгоритмов поиска, извлечения, дополнения, удаления и изменения данных в базе позволяет сэкономить дисковое пространство и значительно сократить время доступа к данным, т.е. выборки и извлечения записей.

Так как объектами ИПС являются Web-страницы, документы и другие источники информации, то в общем виде каждая запись в ее базе индексов, в основном, должна включать в себя следующие поля:

- ***поле KeyWords*** - поле ключевых слов, выделяемых из тела документов или приписанных к ним для определения их смыслового содержания;

- ***поле Address*** - поле адресов, которые содержат URL-адреса, определяющие местонахождение в Интернет;

- ***поле Position*** - поле позиции ключевых слов, определяющее позиции ключевых слов в источниках;

- ***поле Description*** - поле краткого описания содержания источников, часто это поле содержит название документа или первый абзац.

Для повышения эффективности работы ИПС, т.е. для уменьшения времени на выборку и извлечение данных, а также для сокращения объема базы индексов предлагается применяеть комплекс методов индексирования, сортировки, логического и иерархического сжатия, а также декомпозиции структуры БД по символьно-указательному принципу. Методы индексирования, сортировки и сжатия используются последовательно, а к полученному результату применяется символьно-указательный метод структуризации.

Суть символьно-указательного метода структуризации заключается в следующем: исходный файл базы индексов делится на несколько частей, т.е. на несколько отдельных файлов, чтобы при этом можно было бы значительно ускорить процесс доступа к данным в базе, путем повышения скорости обработки и сокращения общего объема данных.

Структуризация БД осуществляется поэтапно с помощью специально разработанной программы, которая также входит в состав ИПС и называется оптимизатором БД.

На первом этапе на основе исходной БД данной программой создается индексный файл. ***Индексный файл*** - это файл ключевых слов, который организуется по индексному полю, т.е. полю ключевых слов. В результате исходная БД индексов делится на три части:

- ***файл ключевых слов***, содержащий поле ключевых слов и указателей на остальную часть записей БД;

- ***файл URL-адресов***, который содержит поля URL-адреса и указатели на запись описания информационных ресурсов;

- ***файл Description*** содержит описания информационных ресурсов, расположенных по URL-адресу.

На втором этапе к файлу ключевых слов применяется ***иерархический метод сжатия данных***. Отметим, что при индексировании документов всякий раз, когда встречаются новые слова они последовательно добавляются в файл ключевых слов. Поэтому в исходном файле ключевые слова могут повторятся, здесь только меняются указатели на URL-адреса документов. Данный файл является не отсортированным последовательным списком.

Согласно данному методу, сначала файл ключевых слов сортируется по полю ключевых слов, потом из отсортированного файла ключевых слов удаляются все записи, содержания полей ключевых слов которых встречаются в предыдущих записях. А содержания полей указателей удаляемых записей последовательно дописываются в поля указателей оставшейся записи.

Таким образом, все записи с одинаковыми ключевыми словами кроме первой удаляются из файла, а к первой записи добавляются содержания указателей всех удаленных записей.

В файл URL-адресов записываются URL-адреса всех проиндексированных документов, т.е. файлов на серверах. Так как индексируются все страницы Web-сервера, то естественно URL-адреса этих страниц могут частично совпадать. Если индексированы несколько файлов на одном и том же сервере, то в их URL-адресе повторяются адрес сервера, а если файлы находятся в определенном

каталоге, то в URL-адресе кроме адреса сервера будут повторяться и названия данного каталога и т.д.

Поэтому на третьем этапе применяется специальный ***метод сжатия по повторению в URL-адресах***, в результате которого в файле URL-адресов начиная со второго URL-адреса удаляется фрагмент строки адреса, который встречается в строке предыдущего адреса, а вместо него заносится числовое смещение, указывающее на адрес, где можно найти удаляемый фрагмент.

На четвертом этапе с помощью ***символьно-указательного метода структуризации*** файла ключевых слов преобразуется и разрабатывается новая структура, согласно которой файл ключевых слов делится на *файлы указателей* и *файл остатка ключевых слов*, количество которых зависит от уровня применения указателей. Отметим, что в файле остатка ключевых слов отсутствуют начальные буквы ключевых слов, а количество отсутствующих букв равно количеству файлов указателей.

Так как поиск информационных ресурсов Интернет производится в базе индексов по ключевым словам, описывающих эти ресурсы и входящих в них, то естественно путем ускорения процесса доступа к конкретным ключевым словам в базе (т.е. локализация записи с заданным ключевым словом) можно повысить эффективность работы ИПС в целом. Благодаря такому делению, путем сужения области данных, содержащая искомую информацию достигается высокоскоростной доступ к данным в базе индексов.

Таким образом, в результате файл базы индексов ИПС делится на следующие файлы:

- *файл URL-адресов*;
- *файлы указателей*;

- *файл остатка ключевых слов;*

- *файл описаний*.

Теперь на конкретном примере более подробно рассмотрим вышеописанный метод. Пусть после индексирования исходная база поисковой системы имеет следующий вид:

...

#http://www.ab.az

Azerbaijan Academy of Sciences...

academy 1

president 2

institutes 5

engine 11

...

#http://www.ab.az/Institutes.html

Scientific-researcher institutes...

institutes 1

physics 2

...

#http://www.ab.az/Departments.html

There are 6 departments in Academy of Sciences...

...

nuclear 3

academy 4

zoology 5

...

После первого этапа исходная база делится на три отдельных файла (рис.7.1, рис.7.2 и рис.7.3).

Ключевое слово	*URL адрес*	*Позиция слова в документе*
Academy	*1*	*1*
president	*1*	*2*
institutes	*1*	*5*
engine	*1*	*11*
institutes	*2*	*1*
physics	*2*	*2*
nuclear	*3*	*3*
academy	*3*	*4*
zoology	*3*	*5*

Рис.7.1. Файл ключевых слов

URL-адреса

#http://www.ab.az
#http://www.ab.az/Institutes.html
#http://www.ab.az/Departments.html

Рис.7.2. Файл URL-адресов

На втором этапе к файлу ключевых слов применяется иерархический метод сжатия данных. Согласно этому методу удаляются все поля, содержание которых встречалось в предыдущих записях (например, слова *academy* и *institutes*). Таким образом, файл ключевых слов получает вид, указанный на рисунке 7.4.

Краткое описание
Azerbaijan Academy of Sciences... *Scientific-researcher institutes...* *There are 6 departments in Academy of Sciences...*

Рис.7.3. Файл описания источников

Ключевое слово	*URL адрес*	*Позиция в документе*
academy	*1, 3*	*1, 4*
president	*1*	*2*
institutes	*1, 2*	*5, 1*
engine	*1*	*11*
physics	*2*	*2*
nuclear	*3*	*3*
zoology	*3*	*5*

Рис.7.4. Файл ключевых слов после применения метода сжатия

Преобразованный вид файла URL-адресов на третьем этапе после применения метода сжатия по повторению в URL-адресах приведена на рисунке 7.5.

URL-адреса	*Указатели на descriptions*
http://www.ab.az	1
смешение 1, Institutes.html	2
смешение 2, Departments.html	3

Рис.7.5. Файл URL-адресов после применения метода сжатия

На следующем этапе структура файла ключевых слов с помощью символьно-указательного метода преобразуется в новую структуру, согласно которой файл ключевых слов делится на файлы указателей и ключевых слов.

7.4. Файл URL-адресов

Как отметили выше, записи исходного файла URL-адресов содержат полные Интернет адреса источников информации, которые записываются в файле последовательно друг за другом. Сначала в файл записывается запись с полным адресом самого сервера, (т.е. корневого каталога), потом идут записи с адресами индексированных страниц, которые находятся в этом корневом каталоге сервера. Далее следует полные названия каталогов первого уровня и файлы этого каталога и т.д. [52, 56].

Из-за того, что названия серверов и каталогов, содержащих информационные ресурсы, произвольны (это зависит от желания владельцев) и имеют различные длины, то записи этого файла также могут иметь произвольную длину.

Так как в файл записываются полные URL-адреса, то в записях с адресами двух страниц, размещенных на одном и том же каталоге некоторого сервера, будет повторяться фрагмент записей, состоящий из названия сервера и каталога. Например: пусть индексированы файлы *index.htm* и *universities.htm* в каталоге *education* сервера *www.medic.com*, тогда полные адресы этих страниц соответственно будут:

http://www.medic.com/education/index.htm

и

http://www.medic.com/education/universities.htm.

Понятно, что строка адреса может быть намного длиннее. Поэтому специально для таких файлов разработан ***метод сжатия по повторению в URL-адресах***, суть которого заключается в следующем: если начиная со второй записи начальный фрагмент (длина может быть любой) строки URL-адреса (URL-адрес серверов, название каталогов и т.д.) является повторением начала одной из предыдущих записей, то повторяемый фрагмент удаляется из строки полного адреса. Для того, чтобы потом можно было восстановить удаленный фрагмент, на начало данной записи вместо удаленного фрагмента добавляется число - указатель на запись, где можно найти удаляемый фрагмент.

Общий вид структуры записей файла URL-адресов приведена на рисунке 7.6. Здесь поле ***Смещения*** - ссылка с фиксированной длиной - 4 байта, показывающая расстояние (число в диапазоне от 0 до 4294967296) от начала предыдущей записи в байтах, содержащей удаляемый фрагмент, до начала данной позиции. Если URL-адрес является (полным) адресом самого Web-сервера, т.е. первым адресом из серии данного сервера, тогда значение поля *Смещения* принимается равным 4294967296. Это значить, что данное поле содержит полный адрес и здесь завершается процесс восстановления полного адреса.

4 байта	1 байт	Длина указывается в предыдущем поле	4 байта
Смещение	*Длина URL*	*URL-адрес или часть URL-адреса*	*Указатель*

Рис.7.6. Формат записи файла URL-адресов

Поле ***Длина URL*** имеет фиксированную длину - 1 байт. Оно содержит целое число, определяющее длину строки URL-адреса, т.е. содержания поля *URL-адрес* данной записи, который расположен непосредственно после этого байта.

Поле ***URL-адрес*** содержит полный URL-адрес или продолжение (остатка) URL-адреса. Длина поля заранее неизвестна и определяется значением содержания предыдущего поля. Начало (удаленный фрагмент) URL-адреса восстанавливается из предыдущих записей, на которую указывает *Смещение*. Следует отметить, что запись, на которую указывает *Смещение* данной записи, тоже может оказаться неполной, тогда читается очередная запись, которая определяется из поля *Смещения* этой записи. Процедура продолжается до полного восстановления URL-адреса, пока содержание поля *Смещения* не совпадает со значением *4294967296*. Таким образом, восстанавливается полный URL-адрес

Поле ***Указатель*** - это адрес записи в файле Description, которая является описанием данного источника. Длина этого поля равна 4 байтам. Для наглядности продемонстрируем структуру файла на примере (Рис.7.7).

Из рисунка видно, что записи содержат полные URL-адреса или его фрагменты. Если значение поля *Смещения* равно 4294967296, то значение *URL-адрес* является полным адресом (например, www.ab.az или www.azeri.com). Если *Смешение* получает другое значение, то соответствующий *URL-адрес* неполный (например, Institutes.html). В таком случае по значению поля *Смешение* определяется сколько байтов впереди (например, для Institutes.html - 18 байт) находится нужный фрагмент для дополнения данного адреса.

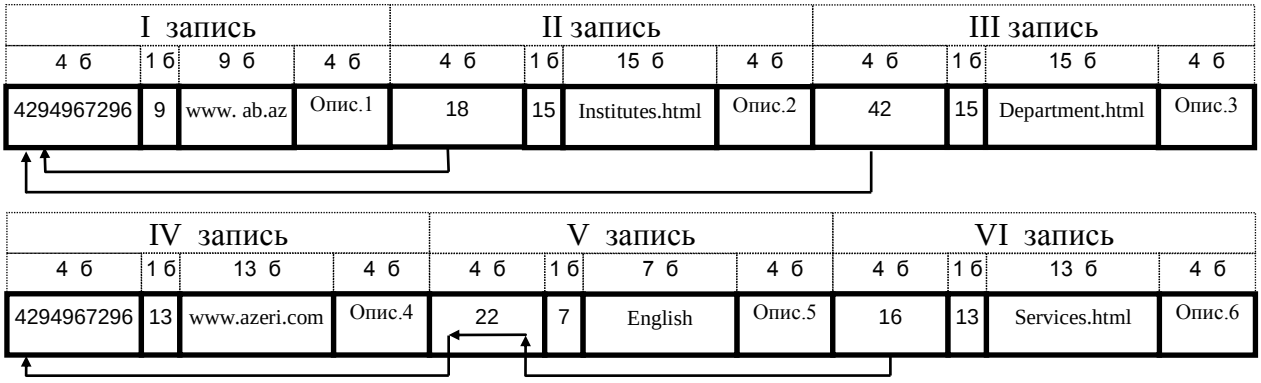

Рис.7.7. Структура файла URL-адресов

В найденной записи читается значение пятого байта, т.е. поля *Длина URL* и определяется длина строки поля *URL-адрес* - в данном случае 9 байтов. Далее считывается следующие 9 байтов и их содержание принимается как нужный фрагмент, который добавляется на начало фрагмента адреса предыдущей записи, а между ними ставится символ “слеш” - “/” (www.ab.az/Institutes.html). Так как значение поля *Смешения* равно 4294967296, то процесс восстановления URL-адреса прекращается. Иначе процесс продолжался бы до тех пор, пока не встречалась бы запись со *Смешением*, равным 4294967296.

Следует отметить, что для правильной записи форматов адресов, служебная информация типа “http://”и“/” добавляется системой автоматически, там где это необходимо.

7.5. Алгоритм определения полного URL-адреса

Пусть из файла ключевых слов указатель указывает на запись в файле URL-адресов, адрес которой является *POSITION***.** В качестве

начального значения переменной *POSITION* присваивается содержание поля *Указатель* файла ключевых слов [52, 56].

Для описания алгоритма используются следующие переменные :

- *FULL_URL* - строчная переменная, содержащая полный URL-адрес источников;

- *POSITION* - целое число, указывающее позицию головки от начала файла;

- *NEXT* - целое число - смешение, указывающее на запись с фрагментом адреса;

- *LENGTH* - целое число - длина фрагмента в текущей записи;

- *FRAGMENT* - строчная переменная, которая содержит фрагмент адреса в текущей записи.

Теперь рассмотрим алгоритм определения полного URL-адреса:

1. Обнулить значение строки полного адреса *FULL_URL=""*;
2. Открыть файл URL-адресов и головку диска устанавливать на позицию *POSITION*;
3. Считывать содержание *Смешения*, т.е. следующие 4 байта начиная с *POSITION* и значение присвоить переменной *NEXT*;
4. Считывать содержание *Длина URL*, т.е. следующий 1 байт и значение присвоить переменной *LENGTH*;
5. Считывать содержание следующих *LENGTH* байтов и значение присвоить переменной *FRAGMENT*;
6. На начало строки *FULL_URL* добавить значение переменной *FRAGMENT: FULL_URL=FRAGMENT+FULL_URL*;
7. Если *NEXT=4294967296*, тогда перейти на шаг 10;
8. На начало строки *FULL_URL* добавить символ *"/"*: *FULL_URL="/"+FULL_URL*;
9. Головку диска сместить назад на позицию:

POSITION= POSITION-NEXT и перейти к шагу 3;

10. На начало строки FULL_URL добавить фрагмент “http:// ”: FULL_URL=“http:// ”: + FULL_URL;

11. Конец

7.6. Принцип деления файла ключевых слов на логические блоки

Для описания структуры файлов указателей и алгоритма их построения, а также для поиска ключевого слова в базе предположим, что все символы, встречаемые в БД индексов ИПС, составляют один алфавит, который обозначим через □, а символы данного алфавита назовем буквами [52, 56]. Пусть *L* - количество букв алфавита □. Файл ключевых слов, как было сказано выше, должен быть отсортированным по алфавиту □.

Файл ключевых слов поэтапно делится на логические блоки, записи которого до деления имеют формат, указанный на рис.7.8. Верхняя строка на рисунке показывает длины полей, а нижняя строка назначение полей. Рассмотрим поля записей данного файла.

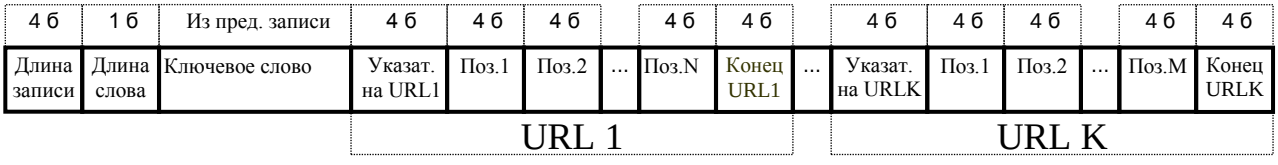

Рис.7.8. Общий формат записи файла ключевых слов

Поле ***Длина слов,*** имеет длину 1 байт и определяет длину ключевого слова. Как отметили выше, каждое ключевое слово может встречаться в нескольких документах, поэтому в записи каждого слова для всех документов, где встречается данное слово, добавляются несколько полей. Из рисунка видно, что в записях для каждого URL-адреса выделяется группа полей,:

- поле ***Указатель на URL K*** - поле указателя на URL-адрес документа в файле URL-адресов;

- поле ***Позиция 1, Позиция 2, ... , Позиция M*** - поля позиций ключевого слова в документе;

- поле ***URL K*** показывает конец группы для *к*-го URL-адреса.

На первом этапе файл данных разбивается на логические блоки, в которых группируются все слова, начинающиеся с одинаковых букв. Эти блоки являются блоками I уровня. Блок I уровня условно обозначим через первые буквы слов, входящих в эти блоки. Таким образом, все слова, начинающиеся буквой “A” входят в “блок A”, буквой “B” в “блок B” и т.д. Логические блоки в файле ключевых слов располагаются в упорядоченном виде согласно алфавиту .

Далее создается I файл указателей, куда заносятся названия блоков I уровня (буквы “A”, “B” и т.д.). В файле ключевых слов удаляются первые буквы из содержания всех полей ключевых слов. После этого в I файле указателей ко всем названиям блоков записываются адрес, определяющий начало соответствующего логического блока в файле ключевых слов, т.е. указатель на начало соответствующего логического блока.

Ясно, что конец каждого блока определяется началом следующего блока. Таким образом, начало, конец и длина каждого логического блока становятся известными.

На втором этапе каждый логический блок I уровня делится на подблоки, аналогично вышеописанному принципу: так как после удаления первых символов ключевых слов, каждый логический блок I уровня, являющийся частью общей БД, в отдельности можно рассмотреть как самостоятельную БД, записи которой также отсортированы по алфавиту , и их аналогичным образом можно разбить на логические подблоки - блоки II уровня.

Таким образом, в результате второго логического деления получаем блоки II уровня, т.е. подблоки блоков I уровня. Все слова, начинающиеся двумя одинаковыми буквами (первые из которых уже удалены), входят в один логический блок II уровня, которые обозначим этими первыми двумя буквами слов. Например, “блок AA”, “блок AB”, “блок AC” и т.д.

После этого создается II файл указателей и в него заносятся вторые буквы названий всех логических подблоков II уровня (“A”, “B” и т.д.). В файле ключевых слов удаляются первые буквы из содержания всех полей ключевых слов, которые фактически являются вторыми, так как на I этапе первые буквы этих слов уже удалены.

Далее во II файле указателей после названий блоков добавляются указатели на начало логических подблоков II уровня в файле ключевых слов. При этом необходимо отметить, что названия всех блоков II уровня, входящие в один и тот же блок I уровня, составляют блок указателей в II файле указателей. В I файле указателей содержание поля указателей заменяется на адрес начала соответствующего блока указателей во II файле.

Аналогичным образом блоки II уровня делятся на подблоки, т.е. на логические блоки III уровня. Они обозначаются через “блок AAA”, “блок AAB”, ... , “блок TEC” и т.д. Далее создается III файл указателей

и третьи буквы названий всех подблоков III уровня заносятся в него, после которых добавляются указатели на начало соответствующих логических подблоков в файле ключевых слов. А в II файле указателей значение поля указателей заменяется на адрес начала соответствующих блоков в III файле указателей.

Принципиальная схема деления файла ключевых слов на логические блоки приведена на рис.7.9, а процесс логического деления файла ключевых слов - на рис.7.10.

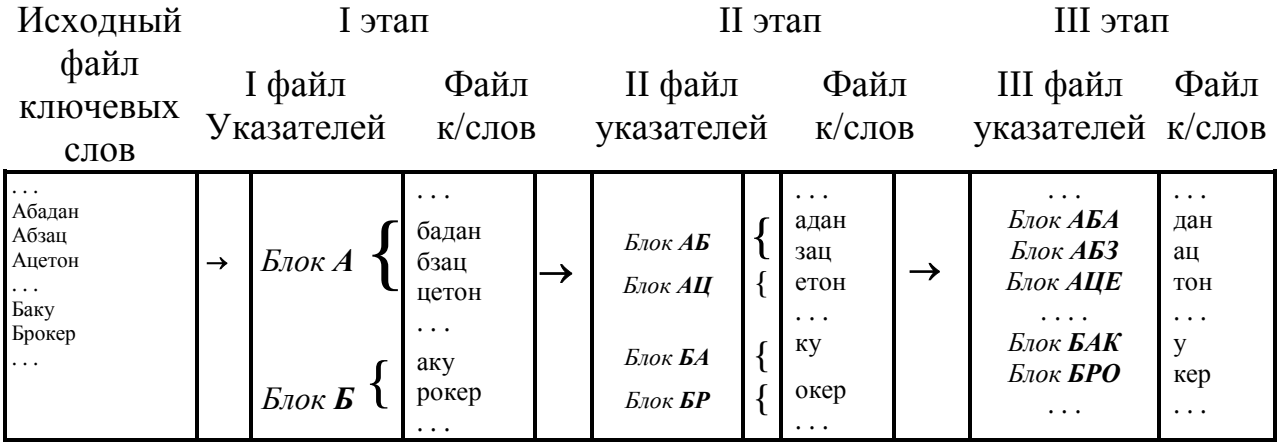

Рис.7.9. Структурное преобразование и деление на логические блоки

Следует отметить, что процедуру деления на логические блоки можно продолжать до тех пор пока не достигается конец какого-либо (самого короткого) ключевого слова в базе. Исследования показывают, что для БД ИПС оптимальным является трехуровневая структура.

Файл ключевых слов, полученный в результате трехэтапного логического деления, назовем остаточным файлом ключевых слов.

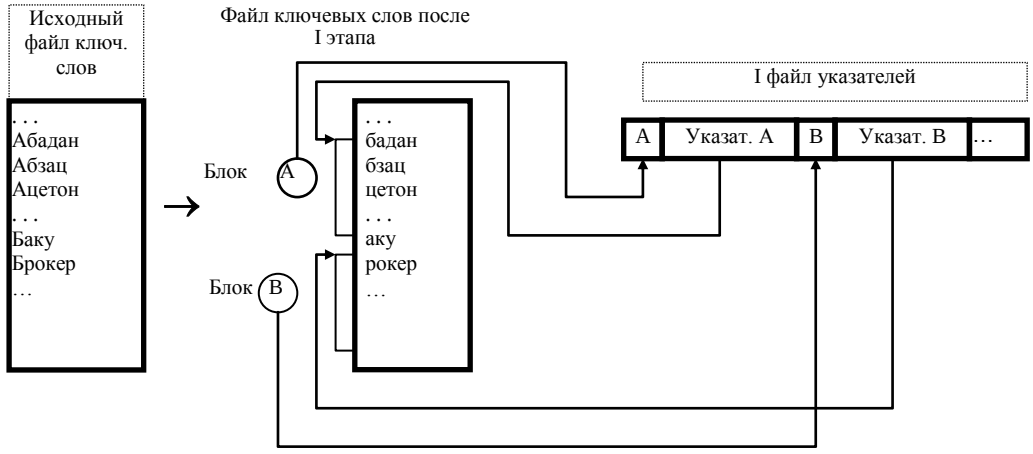

Рис.7.10. Процесс логического деления файла ключевых слов

7.7. Остаточный файл ключевых слов

Записи остаточного файла ключевых слов содержат ключевые слова, указатели на URL-адреса и позиции слова в теле каждого источника информации [52, 56]. Как отметили выше, к этому файлу применяется иерархический метод сжатия, после чего все ключевые слова в файле встречаются только один раз, а указатели на разные URL-адреса и позиции слов в этих источниках записываются подряд после слова. Записи остаточного файла ключевых слов для БД индексов ИПС Интернет имеют формат, указанный на рис.7.11.

Формат данного файла аналогичен формату файла ключевых слов, указанного на рисунке 7.8. Однако, в отличии от общего формата записей в данном формате в поле ключевых слов отсутствуют первые три буквы, поэтому мы его подробно описывать не будем.

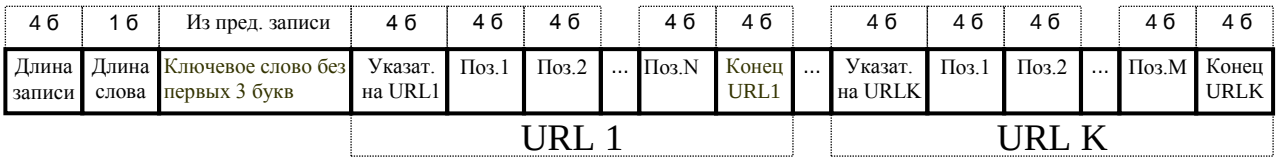

Рис.7.11. Формат записи файла ключевого слова

7.8. Файлы указателей

7.8.1. I файл указателей

Пусть L^I - количество записей в I файле указателей [52, 56]. Ясно, что L^I - число логических блоков первого уровня в файле ключевых слов, которое означает, что количество разных первых символов в ключевых словах является L^I. Понятно, что число L^I получает значение от нуля (когда база пуста) до максимального размера алфавита L (когда все буквы алфавита уже встречались в качестве первого символа), т.е.

$$0<L^I\leq L. \qquad (7.1)$$

Следует отметить, что записи в I файле указателей добавляются только тогда, когда встречается слово с первой буквой, отличающаяся от первых букв предыдущих слов.

Каждая запись в этом файле имеет длину 3 байта и состоит из двух полей (рис.7.12):

- Поле ***букв*** или поле ***названия блоков***, содержащее названия блоков, типа “А”, “Б”, “В” и т.д. Длина данного поля является 1 байт.

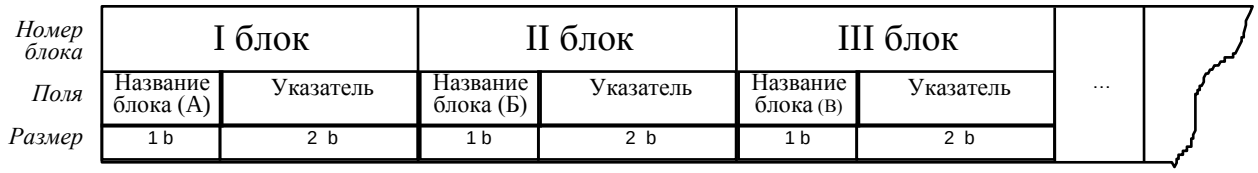

Рис. 7.12. Формат записи I файла указателей

- Поле ***указателей*** содержит адреса начала соответствующих блоков записей во II файле указателей, определяющее подблоки ключевых слов II уровня. Длина поля указателей - 2 байта.

7.8.2. II файл указателей

Записи этого файла содержат названия блоков II уровня (подблоков блоков I уровня) и указателей на начало блоков III уровня [52, 56].

Структура записей II файла указателей аналогична структуре записей I файла указателей, но длина его записей - 4 байта. Однако, здесь *поле букв* (длина 1 байт) - это буквы, которые являются вторыми буквами слов соответствующего блока I уровня, т.е. вторые буквы названий блоков типа “АА”, “АБ”, “АВ” и т.д. *Поле указателей* является указателем на начало подблоков блоков II уровня в III файле указателей, т.е. на блоки III уровня. Для указателей III уровня выделяется 3 байта.

Если количество записей файла обозначим через L^{II}, тогда очевидно:

$$L^{II} = \sum_{i=1}^{L^I} L_i^{II} , \qquad (7.2)$$

где, L_i^{II} - число логических подблоков во II файле i-го блока I уровня, т.е. количество слов с разными вторыми буквами, первые буквы которых являются i-ой буквой алфавита. Ясно, что L^{II} теоретически может получать значения в пределах $L^I \le L^{II} \le L^I \cdot L$.

7.8.3. III файл указателей

III файл указателей включает в себя блоки III уровня, т.е. подблоки блоков II уровня, и указатели на блоки ключевых слов, которые имеют одинаковые первые три буквы и составляют один логический блок [52, 56]. Структура записей аналогична структуре записей предыдущих файлов указателей, однако записи имеют длину 5 байтов: *поле букв* - 1 байт и *поле указателей* - 4 байта.

Если через L^{III} обозначим количество записей в III файле указателей. Тогда

$$L^{III} = \sum_{i=1}^{L^I} \sum_{j=1}^{L_i^{II}} L_{i,j}^{III} = \sum_{i=1}^{L^I} L_{II,i}^{III}, \qquad (7.3)$$

где

L^I и L^{II} - общее количество блоков соответственно I и II уровней, т.е. число записей в I и II файлах указателей;

L_i^{II} - число подблоков II уровня *i*-го блока I уровня во II файле указателей;

$L_{i,j}^{III}$ - число подблоков I уровня, входящих в *j*-й блок II уровня, который в свою очередь также является подблоком *i*-го блока I уровня;

$L_{II,i}^{III}$ - число блоков III уровня, входящих в L_i^{II}-й блок II уровня, который также в свою очередь входит в *i*-й блок I уровня.

$$L_{II,i}^{III} = \sum_{j=1}^{L_i^{II}} L_{i,j}^{III} . \qquad (7.4)$$

Здесь $L^{II} \le L^{III} \le L^{II} \cdot L$.

Общая структура файлов и связей между ними в общем виде представлена на рис.7.13.

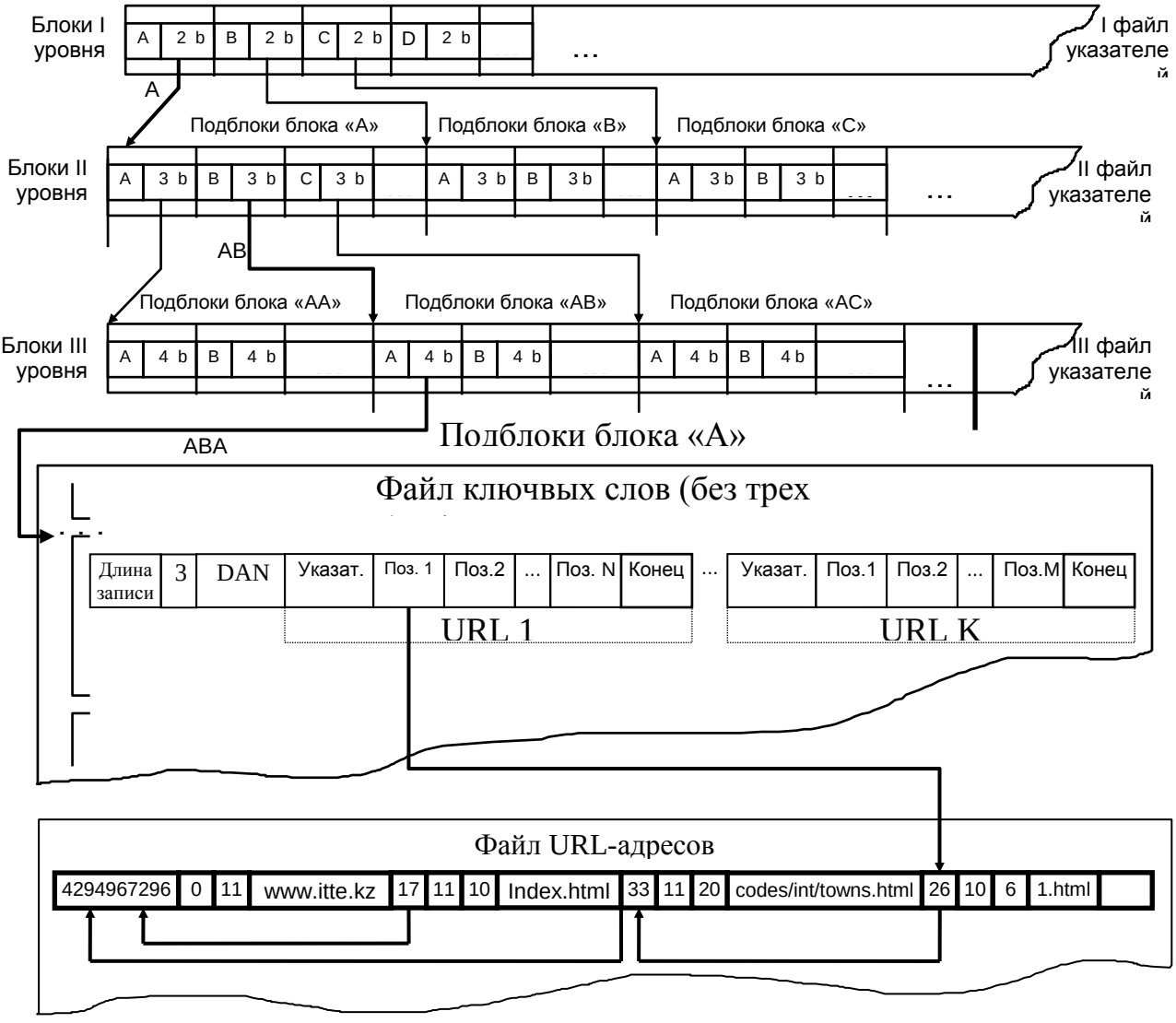

Рис.7.13. Файлы базы поисковой системы и связи между файлами

Из рисунка видно, как осуществляются переходы от одного указательного файла к другому и каким образом связываются блоки со своими подблоками. После трех этапов выделяется очень узкая область данных, где поиск ведется не по неполному содержанию ключевого слова, а по оставшейся части, т.е. без трех букв. Из найденной записи по содержанию поля указателя на URL определяется адрес источника, который выдается на экран пользователя.

ЗАКЛЮЧЕНИЕ

На основе проведенных исследований построены основные принципы разработки интеллектуальных информационно-поисковых систем для компьютерных сетей с большими информационными ресурсами таких, как Интернет. В процессе научных исследований были получены следующие основные результаты:

1. Дана классификация информационных ресурсов Интернет.
2. Анализированы особенности существующих поисковых средств, сформулированы основные задачи и критерии информационного поиска в Интернет.
3. Предложена модель информационного поиска на базе нечетких знаний (нечетная модель информационного поиска).
4. Разработан метод автоматического определения тематики документов и разбиения информационного пространства на тематические каталоги по профилям документов.
5. Разработаны методы повышения точности выбора профиля и улучшения качества тематического каталога.
6. Разработан метод поиска наиболее релевантных документов по нечеткому запросу пользователя.
7. Разработан метод поиска релевантной информации на основе нечетких отношений предпочтительности.
8. Разработан метод распределенного поиска на основе нечетких отношений предпочтительности.
9. Разработана иерархическая модель информационного пространства Интернет.

10. Разработаны методы разбиения информационного пространства на поисковые зоны, виртуального объединения Web-областей, оценки эффективности и определения степени распределенности структуры поисковой системы, а также определения степени сложности поисковых зон.
11. Построена модель интерфейсов пользователя в распределенных информационно-поисковых системах, описаны возможные интерфейсы, сформулированы требования к ним, исследована зависимость эффективности канала связи от структуры интерфейсов.
12. Разработан метод определения наилучшего направления поиска.
13. Разработана концептуальная модель информационно-поисковых систем для Интернет.
14. Разработана первая в республике информационно-поисковая система на базе академической сети Интернет.
15. Разработаны поисковый робот для информационно-поисковой системы и алгоритмы индексирования, а также загрузки Web-страниц в индексную базу поисковой системы.
16. Разработана структура хранения данных в базе индексов информационно-поисковой системы.

Разработанные методы индексирования, определения тематики документов, поиска информации по ключевым словам, основанные на нечеткие знания об информационных ресурсах позволяют повысить эффективность поиска информации в целом. Наилучший результат достигается особенно тогда, когда ресурсы информационного пространства представлены нечетко, пользователь точно не знает где и как искать нужную ему информацию, и поэтому сформулирует

нечеткий запрос. Также для улучшения качества результатов поиска с помощью предложенных методов используются нечеткие отношения документами, терминами и т. д.

Предложенная концептуальная модель информационно-поисковой системы включает в себя все подсистемы, реализующие вышеописанные методы и алгоритмы, что позволяет обеспечить требуемый уровень интеллектуальности поисковой системы.

ЛИТЕРАТУРА

1. Аббасов А.М., Касумов В.А., Гулиева А.Ч. Организация поиска информации в библиотечно-информационных системах. // Известия НАНА. Серия физико-технических и математических наук. Том XXI. Проблемы информатики и управления. №3. 2003. стр.3-10.
2. Аббасов А.М., Касумов В.А. Интерфейсы пользователя в распределенных информационно-поисковых системах. Устройства систем и машин. 2003. №5. стр.67-74.
3. Аббасов А.М., Касумов В.А., Гулиев Р.А. Методы принятия решений в интеллектуальных информационных системах. Учебник. Баку.2003.256 с.
4. Аббасов А.М., Мамедова М.Г. Методы организации баз знаний с нечеткой реляционной структурой. Элм. Баку. 1997. 256 с.
5. Аббасов А.М., Махмудов Ю.А. Распределенные системы обработки данных. Баку. Элм.1990. 185 с.
6. Ако А.В, Хонкрофт Д.Э., Ульман Д.Д. Структуры данных и алгоритмы. Перевод с англ.: Уч. пособ. -М.: Издательский дом «Вильямс», 2000. 384 с.
7. Аликберов А. Использование метаданных (HHTP-EQUIV, NAME, REL, REV, BASE) при создании HTML документов. // Internet: http://www.citforum.ru/win/internet/search /metatags.shtml.
8. Аничкин С.А., Белов С.А., Бернштейн А.В. и др. Протоколы информационно-вычислительных сетей: Справочник. Под ред. И.А.Мизина, А.П.Кулешов. -М.: Радио и связь, 1990. 504 с.
9. Антопюк Б.В., Кочетков Г.Б. Автоматизированные системы обеспечения принятия решений в Американском управлении. Для служ. исп. -М.: инст. США и Канады, 1985. 90 с.

10. Байков В. Интернет. Поиск информации. Продвижение сайтов. Санкт-Петербург: БХВ, 2000. 288 с.
11. Барахнин В.Б., Федотов А.М. Разработка базы данных «Web-ресурсы математического содержания». // VII Международная конференция по электронным публикациям «EL-Pub2002». 23-27 сентября 2002. Новосибирск. http//www/ict.nsc.ru/ws/elpub2002/3963/.
12. Беляков В.А. Динамический информационный Web-сервер многоцелевого назначения. // VII Международная конференция по электронным публикациям «EL-Pub2002». 23-27 сентября 2002. Новосибирск. http//www/ict.nsc.ru/ws/elpub2002/3116/.
13. Борисов А.Н., Левчинко А.С. Методы интерактивной оценки решений. -Рига: Зинатне, 1982. 139 с.
14. Булгаков М.В., Внотченко С.С., Гридина Е.Г. Принципы формирования каталога Интернет-ресурсов федерального портала «Российское образование». // Всероссийская научно-техническая конференция. http://tm.ifmo.ru /tm2003/db/doc/get_thes.php?id=305.
15. Васкевич Д. Стратегии клиент/сервер. Руководство по выживанию для специалистов по реорганизации бизнеса. -К.: «Диалектика», 1996. 384 с.
16. Венда В.Ф. Системы гибридного интеллекта: Эволюция, психология, информатика. -М.: Машиностроение, 1990. 448 с.
17. Востриков А.Н. Пользовательский интерфейс в информационной системе библиотеки. // Информационные ресурсы России. №1. Москва. 1996. стр. 10-19.
18. Гринберг И., Гарбер Л., Разработка новых технологий информационного поиска. // Открытие Системы. №9-10. 1999.

19. Гриценко В.И., Котиков Е.А., Урсатьев А.А., Никулин В.Н. Модель распределенной информационной системы широкого применения // Управляющие систем и машины. 1999. №5. стр.32-42.
20. Гусев. В.Ф., Залялов Р.Г., Дьячков В.В. Система концепций информационного пространства. // Труды Международной Академии Связи. (Приложение к журналу «Электросвязь»). №1(25). Москва. 2003. стр.2-7.
21. Гэффин А. Пауки воюют с хаосом и беспорядком. // Сети. №8. 1996.
22. Девони К. Механизмы поиска в intranet набирают обороты. // Сети. №7. Москва 1996.
23. Дейт К. Введение в системы баз данных.: Пер. с англ. 6-е изд. -К: Диалектика. 1998. 782 с.
24. Диго С.М. Проектирование баз данных. – М.: Финансы и статистика, 1988. 216 с.
25. Дубинский А.Г. Модель мультиагентной системы информационного поиска в глобальной сети // Искусств. интеллект. 1999. №2. стр. 271-279.
26. Дубинский А.Г. Некоторые вопросы применения векторной модели представления документов в информационном поиске. // Управляющие системы и машины. 2001. №4. стр.77-83.
27. Дубинский А.Г. Пути улучшения качества функционирования информационно-поисковых систем глобальной сети // Актуальні проблеми автоматизації та інформ. технологій: Зб. наук. пр. - Д.: Навч. кн., 2000. Т. 3. стр.55-61.
28. Дубинский А. Г. Факторы, влияющие на качество информационного поиска // Системный аналіз та інформаційні

технології: Зб. тез доп. Міжн. наук.-практ. конф. студ., аспірантів та молод. вчених. - К.: НТУУ "КПІ", 2001. Ч.2. стр.43-48.

29. Ермаков А.Е. Тематический анализ текста с выявлением сверхфразовой структуры. // Информационные технологии. №11. Новосибирск. 2000. стр.37-40.
30. Жигалов В., Как нам обустроить поиск в Сети? // Открытые системы. №12. Москва. 2000.
31. Зайцева Е.М. Лингвистическое обеспечение системы корпоративной каталогизации: предложения на этапе проектирования. ГПНТБ. Москва.
32. Захаров В.П. Особенности поисковых средств в информационных сетях с архитектурой «клиент-сервер». // Науч. и техн. б-ки. 1998. №2. стр. 105-111
33. Касумов В.А. Анализ методов полнотекстового индексирования документов. // VI Республиканская научная конференция аспирантов и молодых ученых . Баку. 23 февраля 2000 г. стр.40-42.
34. Касумов В.А. Информационный поиск в Интернет на базе нечетких знаний. // VI рабочее совещание по электронным публикациям El-Pub2001. Новосибирск. 25-27 апреля 2001 г. Эл.публикация.
    http://www.ict.nsc.ru/ws/elpub2001/1450/rep1450.pdf .
35. Касумов В.А., Зайцева Т.Н. Разработка полнотекстовой поисковой системы по информационным ресурсам Азербайджана. // VI Международная научно-практическая конференция Проблемы создания, интеграции и использования научно-технической информации на современном этапе. Киев. Украина. 17-19 декабря 1999 г.

36. Касумов В.А. Методы автоматического создания тематических каталогов информационных ресурсов Internet для информационно-библиотечных систем. // VIII Международная конференция Крым-2001. Библиотеки и ассоциации в меняющем мире, новые технологии и новые формы сотрудничества. Судак. Автономная Республика Крым. Украина. 9-17 июня 2001. стр.254-258.
37. Касумов В.А. Методы индексирования информационных ресурсов Internet. // Свободная молодежь Азербайджана и новые информационные технологии. I Республиканская научно-практическая конференция. Баку. 14 декабря 1999. стр.91-96.
38. Касумов В.А. Методы информационного поиска в Internet на основе нечетких отношений предпочтения. // Автоматика и Вычислительная Техника. Рига. Выпуск №4. 2003. стр.71-78.
39. Касумов В.А. Моделирование информационного пространства Internet. // VIII Международная научно-практическая конференция Система научно-технической информации: проблемы развития и функционирования. Киев. Украина. 30-31 мая 2001 г.
40. Касумов В.А. Методы нечеткого представления информационных ресурсов Интернет и поиска информации по этим ресурсам. // Интеллект. №:1(9). Тбилиси. 2001. Стр.43-45. http://www.geocities.com/intelectige/ magazine/ rus /1_9/05_tech.htm#05_05.
41. Касумов В.А. Методы построения информационно-поисковых систем на базе иерархической модели информационного пространства Интернет. // Автоматика и Вычислительная Техника. Рига. Выпуск № 1. 2002. стр.40-51.

42. Касумов В.А Моделирование интерфейсов поисковых систем для Internet. // Известия Академии Наук Азербайджана. Серия Физико-Технических и Математических Наук. XX том. Информатика и Проблемы управления. №2-3. Баку. 2000. стр.13-18.
43. Касумов В.А. Моделирование информационного поиска в Интернет на базе нечетких знаний. // Автоматика и Вычислительная Техника. Рига. Выпуск: №2. 2002. стр.49-61.
44. Касумов В.А. Нечеткое представление ресурсов Интернет и их классификация по тематике. // Ученые записки Азербайджанского Технического Университета. X том. №3. Баку 2001. стр.62-66.
45. Касумов В.А. Организация интерфейсов в поисковых системах. // Открытые системы. №9. Москва. 2001. стр.37. http://www.osp.ru/os/2001/09/037.htm
46. Касумов В.А Организация распределенного поиска на базе нечеткой модели. // Известия Национальной Академии Наук Азербайджана. Серия Физико-Технических и Математических Наук. XXI том. Информатика и Проблемы управления. №2. Баку. 2001.
47. Касумов В.А. Организация системы поиска в Азербайджанской части Internet. // Открытые системы. №3. Москва. 2000. стр.59-62.
48. Касумов В.А. Поисковые механизмы библиотечно-информационных систем Internet. // VI Международная конференция Крым-2000. Библиотеки и ассоциации в меняющем мире, новые технологии и новые формы сотрудничества. Судак. Автономная Республика Крым. Украина. 3-11 июня 2000. стр.240-244.

49. Касумов В.А Поисковые механизмы информационно-поисковых систем Азербайджанских узлов Internet. // Известия Академии Наук Азербайджана. Серия Физико-Технических и Математических Наук. XVIII том. Информатика и Проблемы управления. №6. Баку. 1998. стр.32-40.
50. Касумов В.А. Поисковый робот информационно-поисковой системы узлов Internet академической сети. // Устройства систем и машин. №5. Киев. 2001. стр.80-86.
51. Касумов В.А. Создание поисковых систем для компьютерных сетей со сверхнасыщенными информационными ресурсами. // Отчет о НИР, ИТНЦ АНА. № гос.регистрации 0100Az00037, инв. №0200Az00184. АзНИИНТИ. Баку. 1999. 41 с.
52. Касумов В.А. Создание цифровой библиотеки для университетов на базе Internet-технологий. // Республиканская научно-практическая конференция по проблемам заочного и дистанционного образования. Баку.17 мая 2001 г. стр.29-34.
53. Касумов В.А. Создание эффективных структур распределенных компьютерных сетей. // Отчет о НИР ИТНЦ АНА. № гос.регистрации 0100Az00036, инв. №0200Az00183. АзНИИНТИ. Баку. 1999. 22 с.
54. Касумов В.А Состояние и перспективы академических узлов Internet. // Известия Академии Наук Азербайджана. Серия Физико-Технических и Математических Наук. XX том. Информатика и Проблемы управления. №2-3. Баку. 2000. стр.96-101.
55. Касумов В.А. Структура хранения данных в базе индексов информационно-поисковых систем для Internet. // Известия Академии Наук Азербайджана. Серия Физико-Технических и

Математических Наук. XVIII том. Информатика и Проблемы управления. №3-4. Баку. 1999. стр.141-149ю

56. Конолли Т., Бегг К., Страпан А. База данных: проектирование, реализация и сопровождение. Теория и практика, 2-е изд.: Пер. с. англ: Учеб. пос. -М.: Издательский дом «Вильямс», 2000. 1120 с.
57. Конюховский П.В., Колесов Д.Н. Экономическая информатика. - Санкт-Петербург: Питер, 2000. 560 с.
58. Корнеев В.В., Гарев А.Ф., Васютин С.В., Райх В.В. Базы данных: интеллектуальная обработка информации. - М.: Нолидж, 2000. 352 с.
59. Крейнес М.Г., Афонин А.А., Ключи от текста. Смысловой поиск и индексирование текстовой информации в электронных библиотеках. // Научно-практическая конференция «Проблемы обработки больших массивов неструктурированных текстовых документов». 2002.
60. Криницкий Н.А., Миронов Г.А., Фролов Г.Д. Автоматизированные информационные системы. Под ред. А.А.Дородницына. -М.: «Наука», 1982. 384 с.
61. Крол Э. Всё об İnternet. Пер. с англ. -К: Торгово-издательское бюро BHV, 1995. 592 с.
62. Ладыженский Г.М. Базы данных: коротко о главном. // НИИСИ РАН. 2000. 114 с. http://www.bolero.ru/catalog /book/pages/pages-1442641.html.
63. Липинский Ю.В. Средства информационного поиска и навигации в больших массивах неструктурированной информации. // Научно-практическая конференция. «Проблемы обработки больших массивов неструктурированных текстовых документов». http//www.tep.ru/text /dataarrays04/html.

64. Лоуренс С., Контекст при поиске в Web. // Открытые системы. № 12. 2000.
65. Львович Я.Е., Леденева Т.М., Винокурова Т.Н. Лингвистическая модель информационно-поисковой системы. // Информационные технологии. №11. 2000. стр.22-31.
66. Мальковский М.Г., Шикин И.Ю. Нечеткий лингвистический интерфейс. // Программирование. 1998. №4. стр.50-61.
67. Мамедова М.Г. Принятие решений на основе баз знаний с нечеткой реляционной структурой. Элм. Баку. 1997. 296 с.
68. Мельников Д.А. Информационные процессы в компьютерных сетях. Протоколы, стандарты, интерфейсы, модели. -М.: КУДИЦ-ОБРАЗ, 1999. 256 с.
69. Меткалф Боб. AltaVista и ее продукты для Internet. // Сети. №1. Москва. 1997. http://www.osp.ru/nets/1997/01 /32.htm.
70. Михайлов Е.Г. Структура хранения для временных баз. // Программирование. №6. 1997. с.73-80.
71. Молородов Ю.И., Федотов А.М. Разработка Internet/Intranet технологий при построении информационных систем. // VII Международная конференция по электронным публикациям «EL-Pub2002». 23-27 сентября 2002. Институт вычислительных технологий СО РАН. Новосибирск. http//www/ict.nsc.ru/ws/elpub2002 /4523/.
72. Некрестьянов И. С. Маршрутизация запросов в системах распределенного поиска. // Труды второй всероссийской научной конференции "Электронные библиотеки". Протвино. Россия. Сент. 2000. с. 280-287.
73. Орловский С.А. Проблемы принятия решений при нечеткой исходной информации. - М.: Наука. 1981. 208 с.

74. Панйр Дж. Вероятностные модели в информационно-поисковых системах. // "Nachr. Dok.". 1986. 37. №2. с.60-66.
75. Попов А. Эффективная методика поиска информации в сети Интернет. // Citforum. http://www.citforum.ru/.
76. Попов И.И., Храмцов П.Б.. Распределение частоты встречаемости терминов для линейной модели информационного потока. // НТИ. Сер.2. №2. 1991. стр.23-26.
77. Просис Дж. Ползающие по Web. // PC Magazine. July. 1996. p.277.
78. Решетников В.Н. Алгебраическая теория информационного поиска. // Программирование. №3. 1979. стр.68-73.
79. Ричардсон Р., Найди то, не знаю что. // LAN/Журнал Сетевых решений. № 3. Москва. 1998.
80. Романенко А.Г. Адресация запросов в распределенных информационных системах и сетях. // Информационный поиск. НТИ. Москва. Сер.2 №5. 1988. стр.7-14.
81. Романова Е. В., Романов М.В., Некрестьянов И.С. Использование интеллектуальных сетевых роботов для построения тематических коллекций. // Программирование. 2000. №3. стр.63-71.
82. Рыжов В.С. Деревовидные структуры для хранения и представления данных. // VII Международная конференция по электронным публикациям «EL-Pub2002». 23-27 сентября 2002. Новосибирск. http//www/ict.nsc.ru /ws/show_abstract.dhtml?ru+45+4312.
83. Семенов Ю.А. Протоколы и ресурсы İnternet-М. Радио и связь, 1996. 320 с.
84. Сенна Д. Fulcrum предлагает единую систему поиска. // Computerword. Москва. №22. 1997.

85. Сергеева Н. Информационные службы Internet: возможности и услуги. // Открытые системы. №1. Москва. 1996. http://www.osp.ru/os/1996/01/22.htm.
86. Система баз и банков данных. Нормативные и методические материалы. -М.: ВИНИТИ. 1992.
87. Смирнов А.В., Шереметев Л.Б., Многоагентная технология проектирования сложных систем. // Автоматизация проектирования. № 3. 1998.
88. Солтон Дж. Динамические библиотечно-информационные системы. - М.: Мир. 1979. 558 с.
89. Талантов М. Поиск в Интернете: использование имен. // Компьютер Пресс. №2. 2000. http://www/citform/ru/internet /namesearch02/.shtml.
90. Талантов М. Профессиональной поиск в Интернете: планирование поисковой процедуры, // Компьютер Пресс. №7. 1999. http://www/citform/ru/internet/namesearch02/.shtml.
91. Талантов М. Профессиональной поиск в Интернете: полнота, достоверность, скорость. // Компьютер Пресс. № 7. 1999. http://www.cpress.ru.
92. Тихонов В. Архитектура метапоисковых систем. «Поисковые системы в сети Интернет». http://www/citform/ru/internet/search/starchsystems.shtml.
93. Тихонов В. Поисковые системы в сети Интернет. «Поисковые системы в сети Интернет».http://www/citform/ru/internet/search/starchsystems.shtml .
94. Узилевский Г.Я. Пользовательский интерфейс: его функции и требования к нему в контексте эргосемиотика. // Научно-

техническая информация: Инф. процессы и системы, 1999. №3. стр. 7-14.

95. Урсатьев А.А., Гриценко Д.В., Кривенко С.Н. Технология доступа к информационным ресурсам корпоративной сети // Устройства систем и машин. 2000. №5/6. стр.36-42.
96. Федотов В.Б. Технология многоагентных систем и доступ к распределённым информационным ресурсам. // VII Международная конференция по электронным публикациям «EL-Pub2002».
http//www/ict.nsc.ru/ws/show_abstract.dhtml?ru+45+4372.
97. Храмцов П., Информационная система WAIS. // Открытые системы. №6. Москва. 1998.
98. Храмцов П., Информационно-поисковые системы Internet. // Открытые Системы. №3. Москва. 1996.
99. Храмцов П. Моделирование и анализ работы информационно-поисковых систем Internet. // Открытые Системы. Москва. №6. 1996. http://www.osp.ru/os/1996/06 /46.htm.
100. Храмцов П. Поиск и навигация в Internet. // Computerword. №19, 20, 22. Москва. 1996.
101. Царева П.В. Алгоритмы для распознавания позитивных и негативных вхождений дескрипторов в текст и процедура автоматической классификации. // Информационные процессы и системы. №12. 1999. стр.15-27.
102. Чанышев О.Г. Автоматическая классификация текстов по доминантным лексемам. // VII Международная конференция по электронным публикациям «EL-Pub2002». 23-27 сентября 2002. Новосибирск. http//www/ict.nsc.ru /ws/elpub2002/4388/.

103. Шилейко А.В., Ляпунцова Е.В. Информация или контент? Труды // Международной Академии Связи. (Приложение к журналу «Электросвязь»). №1(25). Москва. 2003. стр.16-18.
104. Шрайберг Я.Л. Основные положения и принципы разработки автоматизированных библиотечно-информационных систем и сетей-М.: ГПНТБ России, 2000. 130 с.
105. Abbasov A.M. Adaptation of knowledge bases with fuzzy relation structures // Proc. of the 8th IFAC/IFORS/IMACS/IFIP Symposium on Large Scale Systems: Theory and Applications, Rio Patras, Greece. 1998, pp.1156-1161.
106. Abbasov A., Alguliev R., Gasumov V., Aliev E. System management for large computer network: experience on design and creation of the Azerbaijan Republic information computer network. // INET-93. San-Francisco. August 17-20. 1993. A27-A32
107. Abbasov A., Gasumov V. Internet and perspectives Shamakhy Astrophysical Oservatory. // Circular of the Shemakhy Astrophysical Observatory named after N.Tousi of Azerbaijan Academy of Sciences. №94. 1998.
108. Abbasov A., Mamedova M., Gasimov V. Fuzzy relational model for knowledge processing and decision making. // International Journal of Mathematics, Game Theory and Algebra. Nova Science Publishers Inc. Volume 1. 2002. pp.191-223.
109. Agichtein E., Lawrence S., Gravano L. Learning search engine specific query transformations for question answering. // In Proc. of the WWW10. pp.169-178. 2001.
110. Akaho E., Bandai A., Fujii M., Comperision of manual and online searches of Chemical Abstracts.”J. Chem.Inf.and Comput.Sci.”. 1986. 26. №2. p.59-63.

111. Arasu A., Cho J., Garcia-Molina H., Paepcke A., Raghavan S. Searching the web. // ACM Transactions on Internet Technology. №1(1). pp.2-43/ Aug. 2001.
112. Ardo A., Koch T. Automatic classification demonstration page (desire II). 2000. URL: http://www.lub.lu.se/desire/ demonstration.html.
113. Ardo A., Koch T. Creation and automatic classification of a robot-generated subject index. June 7, 1999. http://www.lub.lu.se/desire/demonstration.html.
114. Atkinson S. Zero result searches. How to minimize them. // “Online”. 1986. 10. №3. p.59-66.
115. Bart S. The new British Library: First anniversary // New Libr. World. 1998. 99. №1145. p.276-386.
116. Bernard P.Z. Object-Oriented simulation with hierarchical, modular models. Academic press. Inc. San Diego. C.A.1990. 395p.
117. Blaning R.W. An entry-relationship framework for information resource management. // Inf. and Manag. 1988. 15. №2. c.113-119.
118. Carter J., Bitting E., Ghorbani A. Reputation Formalization for an Information-Sharing Multi-Agent System. Computational Intelligence. Volume 18. Issue 4. November. 2002. p.515 http/www.Blackwell-synergy.com/links/ doi/10.1111/ 1467-8640.t01-1-00201/abs/.
119. Cooper M. Desing considerations in instrumenting and vjnitaring Web-based information retrieval systems. // J.Amer.Soc.Inf.Sci. 1998. 49. №10. p.903-919.
120. Ellis D., Forol N., Furner J. In search of the unknown user: indexing, hypertext and the world wide web. // Journal of Documentation. Vol.54. №1. January 1998. pp.28-47.

121. Fiedler J., Hammer J. Using mobile crawlers to search the web efficiently. // International Journal of Computer and Information Science. №1(1). 2000. pp.36-58.
122. Filman R., Pena F. Compare of search systems of Internet. // IEEE Internet Computing. IC Online. 1998.
123. Filman R., Pena-Mora F., The compare of search systems of Internet, // ComputerWeekly. №1. 1998. стр.15.
124. Gandhi S. Proliferation and categories of Internet directories. A database Internet subject directories. // Ref. and User Serv. Guart. 1998. 37, №4. p.319- 331.
125. Gasimov V.A. The informatization of educational process by means of information technology. // Proceeding of WISTCIS Workshop "E-working, distant training, environmental monitoring: new opportunities". Baku. 13-14 Dec. 2001. Pp. 64-66.
126. Gillispie J. Assuring user success in a networked environment for government information. // Collect. Manag. 1998. 23. №3. p.1-8.
127. Gudivada V.N. Information search on World Wide Web. // Compouter Weekly. Moscow. №35. 1997. pp.19-21, 26,27.
128. Haslam W. A browser driven by classification information model. // STEED/T5/01/1. University of Manchester. 1997. pp.1-6.
129. Hawkins D.T. Hunting, grazing, browsing; A model for online information retrieval. // Online. 1996. 20. №1. p.71-73.
130. Ingrid. H / The retrieval power of selected search Engines: How well do they address general reference questions and subject questions? // Ref. Libr. 1998. №60. p.27-47.
131. Kahle, B., and Medlar, A., An Information System for Corporate Users: Wide Area Information Servers. // Technical Report TMC-199. Thinking Machines, Inc. April 1991.

132. Kai O., Kenneth S., James W. Full text searching and information overload // Int. Inf. and Libr. Rev. 1998. 30. №2.
133. King J.L., Kraemer K.L. Information resource management: is it sensible and it work? // Inf. and Manag. 1988. 15. №2. c.7-14.
134. Koffley J.J. A proposal for a space technology R&D information repository. // AIAA Pap. 1995. №3802. p.1-6.
135. Kollar C., Leavitt J., Mauldin M., Robot Exclusion Standard Revisited, http://www.kollar.com/robots.html, June 2, 1996.
136. Koster, M., ALIWEB: Archie-like Indexing in the Web. // Computer Networks and ISDN Systems. №27(2). 1994. pp. 175-182.
137. Koster M., Standard for robot exclusion. http://info.webcrawler.com/mak/projects/robots/robots.html.
138. Lassalle E. Text retrieval: from a monolingual system to a multilingual system. // Documentation and Text Managing. 1993. 1. №1. p.65-74.
139. Lassia O. Web metadata: a matter of semantics. // IEEE Internet Computing. Jule-August 1998. pp.30-37.
140. Leigh W., Evans J., Interpretation of natural language database queries using optimization methods. // IEEE Trans. Syst., Man and Cybern. 1986. 16. №1. pp.40-52.
141. Lewis F.L. Applied optimal and estimation. Digital design and implementation. Texas instruments. New Jersey. 1992. 624p.
142. Martin B. An Overview of Information Retrieval Subjects. // IEEE Computer. №5. 1985. p.67-84.
143. Norault T., McGill M., and Koll M.B. A performance Evaluation of Similarity Measures, Document Term Weighing Schemes and Representations in Boolean Environment, Information Retrieval Search. // R.N. Oddy et al., eds. Butterworth. London. 1981. p.57-76.

144. Notess Greg R. On the Net: more Internet search strategies. // On-line. 1998. 22. №5. p.71-72, 74.
145. Okada R., Lee E. A method for personalized Web searching with hierarchical document clustering. // Transaction of Information Processing Society of Japan. Vol.39. №4. 1998. p.868-877.
146. Okada R., Lee E., Kinoshita T., Shiratori N. A method for personalized web searching with hierarchical document clustering. // Transaction of Information Processing Society of Japan. Vol.39. №4. Apr. 1998. pp.868-877.
147. Paice C. Expert systems for information retrieval? //Aslib Proc. 1986. 38. №10. p.343-353.
148. Paijmans H. Indexing texts with smart. // Linux Gazette. № 13.
149. Panyr J. Probabilistische Modelle in Information-Retrieval-Systemen. // “Nachr. Dok.”. 1986. 37. №2. p.60-62.
150. Peng C.S., Chen S.K., Chung J-Y., Roy-Chowdhury A., Srinivasan V. Accessing existing Business data from the World Wide Web. // IBM Syst. J. 1998. 37. №1. p.115-132.
151. Rabenseifer A. Sucasne tendencie v analyze a projektovani informaenych systemov. // “Inf. Syst.”. 1986. 15. №3. p.245-256
152. Robertson A., Willett P. Application of N-qrams in textual information systems // Journal of Documentation. 1998. 54. №1. p.48-69.
153. Salton D., Lesk M.E. Computer Evaluation of Indexing and Text Processing. Journal of the ASM. 15. № 1 (January 1968). pp.8-36.
154. Salton G. Another look at automatic text-retrieval systems. // Commun.ACM. №7. 1986. 26. p.648-656.
155. Salton G. Search and Retrieval Experiments in Real-Time Information Retrieval, // Information Processing. 1968. pp.1082-1093.

156. Sato O. Horiuchi M. Information resource management as a coordinating mechanism: a study in large Japanese firms. // Inf. and Manag. 1988. 15. №2. c.93-103.
157. Schatz B.R. Information retrieval in digital libraries: Bringing search to the net. // Science. 1997. 275. №5298. p.327-334.
158. Shih-Fu C., John S., Mandis B., Ana B. Visual information retrieval from large distributed online repositories. // Commun. ACM. 1997. 40. №12. p.63-71.
159. Simon B. End-user searching at Cranfield University. // New Libr. World. 1998. 99. №1139. p.31-40.
160. Spiegler I., Elata S. A priori analysis of natural language queries. // Information Process and Managing. 1988. 24. №6. p.619-631.
161. Tricker R.I. Information resource management - a cross-cultural perspective. // Inf. and Manag. 1988. 15. №2. c.37-46.
162. Voorhees E., Harman D. Overview of the ninth text retrieval conference. // In Proc. of the TREC9. 2000. pp. 1-15.
163. Yanhong Li. Toward a qualitative search engine. // IEEE Internet Computing. July-August. 1998. pp.24-29. Internet: http://coputer.org/internet/.
164. Yu C.T., Lam K., Salton G. Term Weighting in Information Retrieval Using the Term Precession Model. // Communication ACM. V.29. 1982. p.152-170.
165. Yu C.T., Salton G. Effective Information Retrieval Using Term Accuracy. // Communication ACM. V.20. №3. p.135-142.
166. Yuwono Budi, Dik L.Lee. Search and Ranking Algorithms for Locating Resources on the World Wide Web. // In Proceedings of the Forth International Conference on the World Wide Web. New York. November, 1995.

167. Zadeh L.A. Fuzzy Sets. // Information and control. 1965. 8. №3. pp.338-353.
168. Zijlker A.W. Setting the science for information resource management. // Inf. and Manag. 1988. 15. №2. c.79-84.