СПИСОК ЛИТЕРАТУРЫ

1. Theory and Practice of Cryptography Solutions for Secure Information Systems / А. Ю. Атисков, В. И. Воробьев, Л. Н. Федорченко и др. // IGI Global, 701 E. Chocolate Ave. Hershey, PA 17033, USA. 2012. Dec. P. 101–130.
2. Воробьев В. И., Фаткиева Р. Р. Природа уязвимостей программного кода // Программируемые инфокоммуникационные технологии: сб. статей / под ред. В. В. Александрова, В. А. Сарычева. М.: Радиотехника, 2009. С. 53–55.
3. ISO/IEC Standing document 11. URL: http://www.din.de/blob/78920/e0cb93d9370a69c2e6b7b0f46571854b/sc27-sd11-overview-of-work-of-sc27-data.pdf.
4. Сайт компании Аудит информационной безопасности. URL: http://www.audit-ib.ru/audit/security-audit/information-flows/program-risk-analysis/ ©2011–2015 audit-ib.ru.
5. IT Governance Green Paper INFORMATION SECURITY& ISO 27001. URL: https://www.itgovernance.co.uk/files/RiskAssessmentSoftware (3).pdf.
6. Перспективные направления развития науки в Петербурге / отв. ред. Ж. И. Алферов, О. В. Белый, Г. В. Двас, Е. А. Иванова. СПб.: Изд-во ИП Пермяков С. А., 2015.

V. I. Vorobiev, R. R. Fatkieva
Saint Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS)

Yu. A. Shichkina
Saint Petersburg Electrotechnical University «LETI»

AUTOMATION OF THE CORRECTION PROCESS REGULATORY DOCUMENTS IN ACCORDANCE WITH INFORMATION SECURITY STANDARDS

Deals with an approach to standards and algorithms analysis on the basis of ontological modeling. Ontology apparatus was also applied to the development of Security Policy of an enterprise and of User's Profile as well as to User's Preferences studies.

Standards harmonization, regulatory acts, ontologies, protection model

УДК 51-37/.004

М. С. Попова
Международный государственный экологический институт им. А. Д. Сахарова БГУ (г. Минск, Беларусь)

Сравнительный анализ алгоритмов поиска информации в различных средах

Выполнен анализ существующих методов и алгоритмов поиска информации в различных видах информационно-поисковых систем. Рассмотрены основные типы информационно-поисковых систем. Кратко рассмотрена история развития технологий информационного поиска. Рассмотрены различные классификации алгоритмов поиска информации. Рассмотрены некоторые методы построения онтологии для задач библиографического поиска документов, их классификации и аннотирования.

Информационно-поисковая система, алгоритм информационного поиска, индексирование, ранжирование, поисковый запрос, библиографический поиск, онтология

Одним из основных информационных процессов является поиск информации. Вместе с ростом количества информации в доступных документах для решения этой задачи разрабатываются новые, все более совершенные, методы поиска необходимых документов. Существующие методы поиска информации разнообразны, что определяется условиями конкретных задач поиска, например

поиск фрагмента текста в данном документе, поиск документа в локальных и глобальных сетях, поиск записей в базах данных и знаний и т. д. [1].

Разнообразие источников информации, инструментов и методов поиска определяет существование множества видов информационно-поисковых систем (ИПС). Наиболее распространенными являются:

– классификационные ИПС;

– предметные ИПС;

– словарные ИПС.

Информационно-поисковая система – это сложный комплекс программ, локальных и сетевых, базирующихся на мощной специализированной аппаратной части. ИПС имеет веб-интерфейс, дающий возможность формирования запросов, поиска нужной информации и фильтрации полученных данных на основе информационно-поискового языка и соответствующих правил поиска.

В 1990 г. А. Эмтэдж, Б. Хилан и Дж. П. Дойч разработали первую компьютерную программу для поиска в Интернете, названную Арчи. Программа Арчи не имела функции индексации содержания документов, так как объем обрабатываемых данных позволял легко найти всю необходимую информацию вручную. Первая поисковая система в вебе W3Catalog была создана в 1993 г.

JumpStation [2], созданный также в 1993 г. Дж. Флетчером, мог искать веб-страницы, индексировать найденные документы с помощью поискового робота и использовать веб-интерфейс для задания поисковых запросов. JumpStation был первым поисковиком в Интернете, имеющим все 3 главные составляющие поисковой системы: поисковый робот, индексатор и поисковую машину.

Первая полнотекстовая ИПС WebCrawler, индексирующая веб-страницы при помощи робота, была запущена в 1994 г.

Наиболее популярной является ИПС Google, созданная С. Брином и Л. Пейджем. Компания Google занимает первую позицию по популярности благодаря лучшим результатам поиска с помощью итеративного алгоритма PageRank, ранжирующего веб-страницы на основе оценки количества гиперссылок на них [3].

Все ИПС работают по одному обобщенному алгоритму – формулируется задача поиска, выбирается совокупность информационных ресурсов, строится запрос, из найденных массивов документов извлекается запрошенная информация и оцениваются результаты поиска.

По используемым поисковым технологиям ИПС делятся:

– на тематические каталоги;

– специализированные каталоги (онлайновые справочники);

– поисковые машины (полнотекстовый поиск);

– средства метапоиска.

Целью настоящей статьи является анализ распространенных алгоритмов поиска информации. Последовательно по тексту основной части статьи анализируются алгоритмы поиска информации в порядке возрастания их сложности и, соответственно, качества результатов поиска. Создателям поисковых систем этот анализ может помочь определиться с выбором подходящего алгоритма поиска информации с точки зрения соотношений стоимость/качество или время разработки/качество.

Поисковые системы характеризуются, в основном, следующими параметрами:

– полнотой;

– точностью;

– актуальностью;

– скоростью поиска;

– наглядностью.

Эти параметры определяются технологией поиска. Результаты запроса выдаются на основе формулы ранжирования, определяющей релевантность каждого сайта и построенной обычно на нескольких сотнях факторов. Среди основных факторов можно указать доменные имена; названия сайтов; описания к сайтам; наличие заголовков и подзаголовков страниц; удобное меню и панель навигации; количество и качество контента; популярность сайта.

Этапы работы алгоритмов всех ИПС имеют много общего. Первый этап – поисковый робот собирает информационные ресурсы, второй этап – индексатором создается поисковый индекс, и наконец, на третьем этапе поисковик обрабатывает индексируемые данные [2].

Поисковый робот просматривает все ссылки, найденные на странице, и выделяет их. Затем по этим ссылкам или по заданному заранее списку адресов ищет новые документы, еще не проиндексированные поисковой системой.

ИПС анализирует содержание каждой веб-страницы для дальнейшего индексирования. Термины могут содержаться в заголовках, самом тексте страницы или в метатегах – специальных полях, в которые авторы веб-страниц помещают ключевые тематические слова для облегчения поиска их ресурса. Индексация заключается в раз-

бивке страницы на части на основе алгоритмов лексического и морфологического анализа. Данные об информационных ресурсах хранятся в базе данных индексов для использования в будущих запросах. Индексация значительно ускоряет процесс поиска информации по запросу пользователя [4]. Ряд поисковых систем, подобных Google, хранят кеш – исходную страницу целиком или ее часть, а также различную информацию о веб-странице. Другие системы, подобные системе AltaVista, хранят все термины найденных страниц.

Поисковик работает с индексными файлами, переданными индексатором. Кроме того, он обрабатывает пользовательские запросы при помощи индекса и возвращает результаты поиска [2]. По запросу пользователя ИПС на основе ключевых слов проверяет свой индекс и выдает список наиболее подходящих веб-страниц, обычно отсортированный по заданному критерию. Вся информация, которую поисковая система загружает и анализирует, содержится в хранилище данных (базе данных). Многие ИПС сопровождают список ссылок краткими аннотациями, содержащими заголовки документов.

Выдаваемый пользователю список ссылок ранжируется. Каждая ИПС имеет свои, определяемые многими, в том числе политическими, экономическими и социальными факторами критерии ранжирования.

Алгоритм ранжирования полученных ссылок по релевантности – один из важнейших в современных ИПС.

Главные критерии, используемые при ранжировании в ИПС:

– наличие терминов из строки запроса, их количество, близость к началу документа и друг к другу в тексте;

– наличие терминов из строки запроса в заголовках и подзаголовках документов;

– количество ссылок на данный документ из других документов;

– «респектабельность» ссылающихся документов.

Методы поиска определяют 2 основных типа поисковых систем:

– использующие предопределенные и иерархически упорядоченные ключевые слова;

– использующие инвертированный индекс, полученный на основе анализа документа (инвертированным индексом называется структура данных, в которой для каждого слова из найденных документов в списке индексов перечислены все документы, в которых оно встретилось).

Простейшим алгоритмом информационного поиска является *булев поиск*. В нем запросы строятся на основе элементарных терминов документов (слов или словоформ), связанных между собой логическими операциями, такими, как дизъюнкция ($\vee$), конъюнкция ($\wedge$) и отрицание ($\neg$). Типичную форму булева запроса можно представить в виде

$$P = (p_1 \wedge \overline{p}_2 \wedge \ldots \wedge p_n) \vee \ldots \vee (\overline{p}_1 \wedge p_2 \wedge \ldots \wedge p_n),$$

где p_i – i-й термин поискового запроса.

Булевы запросы обычно используются в поиске на точное соответствие, при котором в документах проверяется наличие заданного термина.

Алгоритм поиска по релевантности. Задачей релевантного поиска является как можно более полная и точная выборка подмножества документов, наиболее близких по смыслу (релевантных) поисковому запросу. В большинстве ИПС функция релевантности документа запросу является вероятностной (т. е. о соответствии документа поисковому запросу можно судить только с какой-то степенью вероятности) и базируется на некоторой информации о документах, автоматически получаемой системой в ходе синтаксического и семантического анализа информационного контекста документа.

Эффективность алгоритма поиска по релевантности характеризуется точностью поиска, определяемой как отношение полученных в результате поиска релевантных запросу документов ко всем выбранным в ходе поиска документам, и объемом выборки – отношением выбранных в результате запроса релевантных документов ко всем релевантным документам, содержащимся в ИПС.

Алгоритм поиска по сходству является модификацией булева поиска, в которой учитываются возможные неточности в задании поисковых терминов или в электронном представлении документов. В качестве меры сходства обычно используется расстояние редактирования, задаваемое функцией Левенштайна [5], [6].

Алгоритм последовательного поиска по ключевым терминам документов. Для ИПС, построенных на этом алгоритме, поисковым индексом является сам набор исходных документов. Такой алгоритм достаточно трудоемок, но прост, а потому при небольших объемах данных является наиболее быстрым. Это позволяет применять его в небольших поисковых системах для решения задач нечеткого поиска.

Существует 2 усовершенствованных алгоритма последовательного поиска – алгоритмы Бойера–Мура [7] и Кнута–Морриса–Пратта [8]. Методом последовательного сканирования при решении задач точного поиска в большинстве случаев быстрее работает алгоритм Бойера–Мура, поэтому он часто используется для редактирования текста. Термины сравниваются справа налево, от последнего символа шаблона P и на m символов текста T, где $|P| = m$ – длина шаблона. При обнаружении несовпадения на паре терминов p_i (шаблон) и t_j (фрагмент текста) находим, в каких позициях шаблона содержится несовпадающий символ t_j, и на основе этого определяем сдвиг шаблона по тексту [7].

Рассмотрим алгоритмы, основанные на построении поискового индекса. Все они имеют общие этапы работы:

– все найденные тексты анализируются, в каждом выделяется основная информация;

– эта информация используется для построения поискового индекса;

– поисковые запросы преобразуются в формат, позволяющий использовать поисковый индекс для определения релевантности запросов и документов и выборки релевантных запросу документов.

На первом и втором этапах используются методы статистического, семантического, синтаксического и лингвистического анализа текста. Наиболее распространен статистический анализ.

Для выделения основной информации исходные тексты нормализуются, т. е. составляются списки ключевых словоформ, необходимые для классификации текстов и построения текстового индекса.

Хеширующие алгоритмы характеризуются простотой реализации и высокой скоростью поиска на сравнительно небольших объемах данных. Они основаны на построении массива ячеек и заполнении их исходными терминами документов в соответствии с заданной хеш-функцией $H(w_i)$. Если в одну ячейку попадает 2 различных термина (т. е. происходит коллизия алгоритма), то один из них переносится в другую ячейку по заданному правилу (обычно соседнюю ячейку) или в этой ячейке сохраняются ссылки на список терминов с одинаковыми значениями хеш-функции.

Алгоритмы построения инвертированных файлов. Каждому термину сопоставляется список его вхождений в документы. При этом обычно точное местоположение термина в документе не указывается. Поиск заключается в сканировании инвертированного списка ключевых слов для нахождения соответствующих терминов запроса и документов. Затем следует обработка результатов поискового запроса, полученных для каждого термина.

Алгоритмы построения сигнатурных файлов обеспечивают бóльшую компактность инвертированного списка по сравнению с алгоритмами построения инвертированных файлов.

Алгоритм заключается в построении списка сигнатур в виде битовых векторов, таких, что i-я компонента вектора равна единице тогда, когда в документе существует термин w_k, такой, что хеш-функция $f(w_i) = i$. Затем поиск терминов сужается до множества тех документов, сигнатуры которых удовлетворяют этому требованию. В настоящее время алгоритм, в отличие от инвертированных файлов, не имеет широкого распространения.

Все алгоритмы, использующие списки при построении полнотекстовых индексов, имеют существенные недостатки:

– время поиска (при условии, что списки ключевых терминов отсортированы) не лучше, чем $O(\log n)$.

При оценке за функцию берется количество операций, возрастающее быстрее всего. Другими словами, если в программе одна функция, например умножение, выполняется $O(n)$ раз, а сложение – $O(n^2)$ раз, то общая сложность программы – $O(n^2)$, так как в конце концов при увеличении n более быстрые (в определенное, константное число раз) сложения станут выполняться настолько часто, что будут влиять на быстродействие куда больше, нежели медленные, но редкие умножения. При оценке $O()$ константы не учитываются. Пусть один алгоритм делает $2500n + 1000$ операций, а другой – $2n + 1$. Оба они имеют оценку $O(n)$, так как их время выполнения растет линейно. В частности, если оба алгоритма, например, $O(n*\log n)$, то это отнюдь не значит, что они одинаково эффективны. Первый может быть, скажем, в 1000 раз эффективнее. $O()$ значит лишь то, что их время возрастает приблизительно как функция $n*\log n$. Другое следствие опускания константы – алгоритм со временем $O(n^2)$ может работать значительно быстрее алгоритма $O(n)$ при малых n... за счет того, что реальное количество операций первого алгоритма может быть $n^2 + 10n + 6$, а второго – $1\,000\,000n + 5$. Впрочем, второй алгоритм рано

или поздно обгонит первый, так как n^2 растет куда быстрее 1 000 000n. Основание логарифма внутри символа $O()$ не пишется. Причина этого весьма проста. Пусть имеется $O(\log_2 n)$. Но $\log_2 n = \log_3 n/\log_3 2$, а $\log_3 2$, как и любую константу, асимптотика – символ $O()$ – не учитывает. Таким образом, $O(\log_2 n) = O(\log_3 n)$. К любому основанию можно перейти аналогично, а значит, и писать его не имеет смысла. Недостатком является то, что время выполнения алгоритма мало зависит от упорядоченности массива. На обработку почти отсортированного файла уходит столько же времени, что и на обработку файла, упорядоченного случайным образом; отсортированные списки трудно модифицировать. Количество операций сортировки не меньше чем $O(n)$.

Полнотекстовые индексы в MySQL обозначаются типом «FULLTEXT», который может применяться для столбцов типов «VARCHAR» и «TEXT». При массовом добавлении данных в таблицу с полями «FULLTEXT» индекс будет создаваться сразу, что замедлит работу. Для избежания эффекта рекомендуется модернизировать поля уже после добавления.

Указанные недостатки отсутствуют у структур данных, основанных на деревьях. Например, бинарные деревья широко используются в алгоритмах построения инвертированных файлов и других алгоритмах полнотекстового поиска. Существуют также алгоритмы, использующие деревья со степенью ветвления больше двух. Практический интерес представляют тернарные деревья. Поиск на их основе выполняется за $O(m+\log n)$ операций сравнения.

Алгоритмы, основанные на классификации документов по кластерам (*фактор-группам*). Кратко рассмотрим их на примере векторной модели.

Исходные документы разбиваются на элементарные термины t_i, которые затем сохраняются в словаре. Если в словаре n разных терминов, то каждый документ D_i можно представить в виде вектора $\mathbf{S}_i = w_1^i w_2^i ... w_k^i ... w_n^i$ размерности n, в котором только компоненты с номерами $t_1^i ... t_{k(i)}^i$ отличны от нуля и равны весам терминов, а $k(i)$ – число уникальных терминов, содержащихся в документе D_i. Веса терминов можно определить следующим образом: пусть f_1^i – частота вхождения термина под номером l в документ D_i, а $F_l = \sum f_1^i$ – суммарная частота вхождения данного термина по всем документам. Тогда вес термина l в документе D_i определим как

$$w_l^i = \sum f_l^i / F_l .$$

Затем полученный вектор весов используется для поискового запроса.

Релевантность документов запросу может определяться в соответствии с различными метриками в векторном пространстве размерности n [9]:

$$C(D_i, D_j) = 2\frac{\sum_{k=1}^{n} w_k^i w_k^j}{\sum_{k=1}^{n} w_k^i + \sum_{k=1}^{n} w_k^j}$$

– коэффициент Дайса;

$$C(D_i, D_j) = \frac{\sum_{k=1}^{n} w_k^i w_k^j}{\sum_{k=1}^{n} w_r^i + \sum_{k=1}^{n} w_r^j - \sum_{k=1}^{n} w_r^i w_r^j}$$

– коэффициент Жаккара;

$$C(D_i, D_j) = \frac{\sum_{k=1}^{n} w_k^i w_k^j}{\sqrt{\left(\sum_{k=1}^{n} \left(w_k^i\right)^2\right)\left(\sum_{k=1}^{n} \left(w_k^j\right)^2\right)}}$$

– коэффициент косинуса.

Обычно релевантность документов запросу определяется в 2 этапа – кластеризация документов и поиск в каждом кластере.

Рассмотрим методы построения онтологии для задач библиографического поиска документов, их классификации и аннотирования. Использование онтологий – это эффективный подход к выявлению и обработке смысла текстовых документов. Один из подходов к созданию экспертной базы для библиографического поиска – использование универсальной десятичной классификации (УДК) [10]. Благодаря обязательному индексированию всех публикаций по классификационному коду можно определить публикации, содержащие информацию по данной теме. Применение онтологии для этой задачи схематично показано на рис. 1.

Подход имеет некоторые недостатки – современные разработки могут не учитываться структурой УДК, которая довольно редко дополняется новыми разделами; многие библиографические базы данных не используют коды УДК, поэтому при формировании онтологии они не учитывают-

ся. Тот факт, что библиограф, осуществляющий библиографические записи, не является экспертом в предметной области, порождает проблемы с выделением ключевых слов.

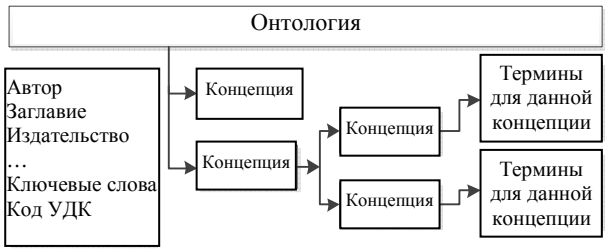

Рис. 1

Избежать указанных недостатков рассмотренного подхода можно, построив онтологию на основе кластеризации полнотекстовых документов [11]. Для этого каждый документ преобразуется в набор терминов. Вся коллекция документов разделяется на кластеры по близости тематики, в результате получаем группы терминов одной тематики. Такой подход устанавливает связи между терминами и концепциями. Каждому термину присваивается вес, характеризуемый частотой встречаемости. Термины с максимальным весом выбираются в качестве концепции. Термины онтологии задаются терминами с весом больше среднего.

Коллекция документов разделяется на кластеры в месте наибольшего расстояния внутри каждого кластера между ближайшими документами (рис. 2).

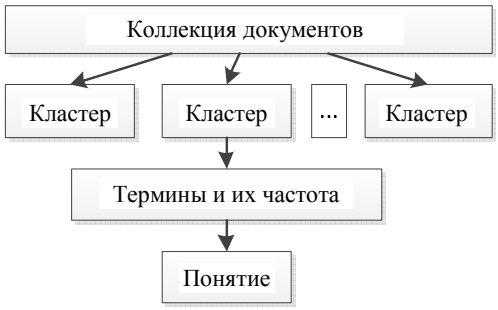

Рис. 2

В качестве метрики для определения расстояния между документами используется частота встречаемости терминов в документах коллекции.

При разработке онтологии необходимо учитывать государственные и международные стандарты, что позволит включить в онтологию международные номера ISBN и ISSN, а также определения и понятия, относящиеся к видам изданий: непериодическое, серийное, периодическое, продолжающееся, серия [12].

В самом начале создания онтологии необходимо определить ее масштаб. Это можно сделать сформировав вопросы компетенции. Онтология будет эффективной, если она на все вопросы в пределах своей структуры находит ответы. Важнейшим этапом разработки онтологии, а также ее модификации является формирование вопросов компетенции.

Существенными характеристиками поисковых алгоритмов являются: число факторов, влияющих на ранжирование сайта; классификаторы для содержимого сайта и ссылок; наличие геоклассификатора; наличие расстояния между словами поискового запроса, «понимаемое» системой; возможность распознавания аббревиатур и обработки транслитерации (в том числе и в URL документа); присвоение документам региональной принадлежности; обработка многослойных запросов; определение автора контента; присвоение запросам категорий; наличие языковой персонализации поиска; перевод простых популярных слов (например, поисковик может понимать, что computer = компьютер и т. п.); постоянное совершенствование поиска по большим многословным запросам.

Все эти и многие другие характеристики невозможно оценить численно, так как реализация алгоритмов несравнима для каждой поисковой системы, их сочетание с другими элементами алгоритмов приводит к уникальности качественных и количественных характеристик поисковых систем в целом.

Исходя из изложенного, анализ, выполненный в настоящей статье, предназначен для общей ориентации разработчиков в динамике и приоритетах развития алгоритмов поисковых систем. При разработке стратегии создания поисковой системы нет смысла говорить о предпочтительности или эффективности алгоритмов. Эфективность системы определяется используемыми элементами алгоритмов, их сочетанием и особенностями конкретной реализации.

Оценка эффективности модели. В модели для тестов используются наборы данных WePS2. Для оценки эффективности модели используются B-cubed значения и четкостные значения (purity scores). Официальной основой оценки модели в WePS2 является F-измерение (среднее гармоническое) B-cubed точности и полноты.

В локальной совокупности данных WePS2 минимальное число кластеров 1, а максимальное – 56.

Весовые коэффициенты для разных слов настраиваются на основании важности для кластеризации, их значения представлены в табл. 1. Все

Таблица 1

Весовые коэффициенты	Признаки					
	Заголовок	URL-адрес	Метаданные	Фрагменты	Контекстные предложения	Совокупность слов
	1	1	2	0.8	2	1

Таблица 2

Системы	F-измерение		B-cubed	
	0.5	0.2	Pre	Rec
PolyUHK	0.82	0.80	0.87	0.79
UVA_1	0.81	0.80	0.85	0.80
ITC-UT_1	0.81	0.76	0.93	0.73
Предложенная модель	0.85	0.83	0.92	0.82

Таблица 3

Системы	F-измерение	
	0.5	0.2
BEST-HAC-TOKENS	0.85	0.84
BEST-HAC-BIGRAMS	0.85	0.83
PolyUHK	0.82	0.79
UVA_1	0.81	0.80
ITC-UT_1	0.81	0.76
Предложенная модель	0.85	0.83

эти параметры устанавливаются согласно результатам экспериментов на тестовых данных WePS2.

В табл. 2 сравнивается эффективность предложенной модели трех самых эффективных на данный момент алгоритмов согласно оценкам WePS2. Табл. 3 иллюстрирует оценку эффективности предложенной модели по сравнению с двумя другими моделями с известным верхним порогом и тремя самыми эффективными на данный момент моделями согласно оценкам WePS2. Из таблицы видно, что предложенная модель превосходит все топ-системы по F = 0.5 и F = 0.2 измерениям. По сравнению с другими системами предложенная модель дает улучшение на 5.5 %.

Высокая эффективность в обеих схемах измерения доказывает, что предложенную модель можно применить в реальных приложениях.

Поскольку ИПС являются автоматизированными, а не автоматическими средствами поиска, эффективность их использования зависит не только от качества и сложности применяемых в ИПС алгоритмов поиска, но и от того, насколько хорошо пользователь знает свойства этих средств и природу операционных объектов. Процесс поиска информации обычно носит эмпирический характер. Степень полноты и точности ответов, получаемых пользователем от ИПС, в равной мере зависит от точности сформулированных им запросов и от возможностей поисковых систем. Можно надеяться, что в недалекой перспективе появятся ИПС с гораздо более развитым искусственным интеллектом, позволяющим автоматически адаптироваться к уровню знаний и запросов конкретных пользователей, воспринимать запросы на естественном языке, хранить больше информации о предыстории запросов каждого пользователя, об области его интересов и особенностях формулирования запросов.

СПИСОК ЛИТЕРАТУРЫ

1. Chu H., Rosenthal M. Search engines for the World Wide Web: A comparative study and evaluation methodology // Proc. of the annual meeting-american society for information science. 1996. Vol. 33. P. 127–135.

2. Risvik K. M., Michelsen R. Search engines and web dynamics // Computer Networks. 2002. Vol. 39, № 3. P. 289–302.

3. The Anatomy of a Large-Scale Hypertextual Web Search Engine Sergey Brin and Lawrence Page. URL: http://ilpubs.stanford.edu:8090/361/1/1998-8.pdf.

4. Jawadekar, Waman S. Knowledge Management: Tools and Technology // Knowledge management: Text & Cases. New Delhi: Tata McGraw-Hill Education Private Ltd, 2011. P. 278–319.

5. Graham A. Stephen. String Search / School of Electronic Engineering Science University College of North Wales, 1992. 76 p.

6. Navarro G. Approximate Text Searching: Technical Report TR/DCC-98-14. 1998. URL: http://www.mathcs.emory.edu/~cheung/papers/Matching/Navarro-Survey-Approx-Matching.pdf.

7. Boyer R. S., Moore I. S. A fast string searching algorithm. Communications of the ACM. 20:762-772, 1977. URL: http://www.cs.utexas.edu/~moore/publications/fstrpos.pdf.

8. Knuth D., Morris J., Pratts V. Fast pattern matching in strings. 6:322–350, 1977. URL: http://delab.csd.auth.gr/~dimitris/courses/cpp_fall05/books/SIAM_JNL_Comp_77_KMP_string_matching.pdf.

9. Van Rijsbergen C. J. Information Retrieval / Dept. of Computer Science. University of Glasgow, 1979.

10. Melnikov A. V., Zakharova I. V. Method of automatic ontology creation based on bibliographic databases // Workshop on computer Science and Information Technologies CSIT. Ufa, 2005. P. 270–272.

11. Онтологии и тезаурусы / В. Д. Соловьев, Б. В. Добров, В. В. Иванов, Н. В. Лукашевич. Казань: Изд-во Казан. гос. ун-та; М.: Изд-во Моск. ун-та, 2006.

12. Стандарты по издательскому делу: сб. док. / сост. А. А. Джиго, С. Ю. Калинин. 3-е изд. М.: Экономистъ, 2004.

M. S. Popova
International State Ecological Institute of A. D. Sakharov – BGU (Minsk, Belarus)

COMPARATIVE ANALYSIS OF INFORMATION RETRIEVAL ALGORITHMS IN VARIOUS ENVIRONMENTS

Analysis of existing methods for information retrieval in different types of information retrieval systems has been performed. Various types of information retrieval systems have been considered. The history of the development for information retrieval technologies has been briefly considered. Different classifications of the information retrieval algorithms have been considered. Some methods of building ontologies for the tasks of bibliographic document searching, document classification and annotation have been considered.

Information retrieval system, information retrieval algorithm, indexing, ranging, search query, bibliographic search, ontology

УДК 539.3: 621.382

С. В. Воробьев, О. П. Кормилицын, Е. А. Лебедева
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Современные информационные технологии и текущий контроль знаний студентов по дисциплине «Прикладная механика»

Представлена методика контроля знаний студентов в течение всего периода изучения дисциплины, индивидуальные задания для каждого этапа проверки знаний, контрольные вопросы для подготовки студентов.

Деформация, внутренние усилия, составляющие напряжений, главные напряжения, напряженное состояние

Цель изучения дисциплины «Прикладная механика» – дать студентам знания по основным понятиям механики, физико-механическим явлениям, которые происходят в твердом теле, и умение использовать основные теоретические и практические методы расчета прочности и жесткости конструкций приборостроения при статическом, динамическом и температурном внешнем воздействии.