УДК: 004.89, 004.85, 004.421, 519.688, 004.58

Модели и алгоритмы для адаптивного поиска в информационно-поисковых системах

В статье представлен анализ моделей и алгоритмов поиска информации в информационно-поисковых системах, анализ существующих проблем и подходов к поиску информации, предложены авторские модели и алгоритмы организации данных и поиска по логическим выражениям в информационно-поисковых системах, произведен анализ эффективности предложенных моделей и алгоритмов.

А. Г. САВЕНКО,
магистр техн. наук, ст. преподаватель, ученый секретарь

Кафедра информационных систем и технологий
Института информационных технологий БГУИР

А. С. ШЕРСТНЕВ,
инженер-программист ООО «АйтиРексГрупп Бел»

Ключевые слова:
адаптивный поиск, поисковые алгоритмы, модель организации данных, графовая база данных, поисковый запрос, логические выражения, релевантность, машинное обучение, информационные системы.

Анализ предметной области. Поиск информации является важной задачей практически любой автоматизированной системы, связанной с обработкой текстовой информации. В настоящее время уже существует множество методов поиска, систематизированных следующим образом: методы на основе ключевых слов; методы на основе метрик (Дамерау, Левенштейна, Жаккара и др.); ассоциативные методы (фильтр Блюма и методы нечеткого хеширования Корнблюма и Чарикара); методы последовательного поиска (Бойера-Мура); поисковые деревья и др.

Критериями качества поиска в информационно-поисковых системах (далее – ИПС) могут являться точность, полнота поиска, выпадение и F-мера, а также быстродействие поисковых алгоритмов. К сожалению, понятие степени соответствия результатов поиска и их полнота, т. е. релевантность, являются субъективными и зависят от конкретного человека, оценивающего полученные результаты.

Также следует отметить, что дополнительными критериями, влияющими на качество и эффективность работы поисковых алгоритмов, являются гибкость и удобство формулирования поискового запроса. Если для оптимизации существующих алгоритмов поиска в основном используется различного рода упорядочивание данных (от сортировки данных до сложных индексирующих систем), то для задания критериев самого поиска практически нет никаких рекомендаций или оптимизаций. В основном гибкость формирования поисковых запросов определяется неточным совпадением символов или последовательностей символов при поиске подстроки в строке [1, 2]. Кроме того, следует отметить, что различные поисковые алгоритмы по-разному воспринимают пробел (как разделитель между ключевыми словами): некоторые заменяют его логическим оператором «И», некоторые – логическим оператором «ИЛИ», другие же вовсе не используют логические операторы [3].

Поисковые алгоритмы использующие логические операторы (AND – логическое умножение, OR – логическое сложение, NOT – логическое отрицание) и их комбинации, относятся к булевой модели поиска. Такая модель формирования поискового запроса позволяет получать более точные и, соответственно, релевантные результаты. Однако булева модель также имеет и ряд недостатков [6], таких как:

– поисковые запросы, сформированные в виде логических выражений, затрудняют варьирование глубины поиска с целью получения большего или меньшего количества результатов поиска в зависимости от требований пользователя;

– при использовании булевой модели отсутствует эффект от функций совпадения векторов, которые дают непрерывный спектр совпадений (полных, частичных или нулевых) между запросами и поисковыми образами документов. Это обстоятельство приводит к жесткому требованию «все или ничего» на выходе;

– в результате выполнения поиска множество результатов не может быть представлено пользователю в ранжированном виде в порядке уменьшения сходства между полученным результатом и запросом. Результат либо соответствует запросу полностью, либо в остальных случаях вообще не соответствует.

Под адаптивным поиском понимается подстройка модели поиска для более точного определения подмножества данных из всего множества информационной системы. Часто адаптивный поиск отождествляют с персонифицированным, однако в общем случае это разные понятия, т. к. второй является разновидностью первого. При персонифицированном поиске происходит отслеживание предыдущих запросов и их учет при определении результатов нового запроса. Таким образом, поисковые алгоритмы подстраиваются под интересы конкретного пользователя, собирая и анализируя информацию о его действиях в поисковых системах, используя его геолокацию, демографические характеристики. Однако такая персонификация имеет существенный недостаток, называемый «эффектом пузыря», или «пузырем фильтров». В результате такой адаптации пользователи получают намного меньше противоречащей своей точке зрения информации и становятся интеллектуально изолированными в своем собственном «информационном пузыре» [4].

Основными недостатками существующих ИПС является их проприетарность и невозможность изменения поведения поиска под конкретную предметную область, а также ограниченность по возможностям использования логических операторов и описанные выше недостатки булевой модели поиска.

Постановка задачи. В рамках исследования необходимо разработать модели и алгоритмы адаптивного поиска по логическим выражениям в ИПС, которые бы предоставляли достаточно гибкий инструментарий для формирования запросов, обеспечивали высокую точность (релевантность) результатов поиска, обладали возможностью адаптации к любой предметной области, имели невысокие требования к вычислительной мощности и начальному объему данных системы, высокое быстродействие. Разрабатываемые модели и алгоритмы поиска по логическим выражениям должны частично или полностью решать проблемы варьирования глубины поиска, эффекта от функции совпадения векторов поиска (при сохранении релевантности результатов) и ранжирования результатов. Поисковые алгоритмы должны быть не персонифицированными. Адаптивность поиска должна быть реализована исходя их гипотезы, что пользователь может не обладать необходимыми компетенциями для формирования точного поискового запроса и понимания предметной области запроса.

Модель организации данных информационно-поисковой системы. В соответствии с поставленными задачами и с точки зрения информационной модели ИПС необходимо реализовать сложный поиск на основе логических выражений. В данном случае целесообразно хранить данные информационной системы в виде связей между сущностями, при комбинировании которых можно добиться нужного критерия поиска. Сами сущности должны описывать как можно меньший объем информации, чтобы обеспечивать меньшую гранулярность и, следовательно, предоставлять более точные результаты. Связи должны быть односторонними и направленными, а их количество – минимальным.

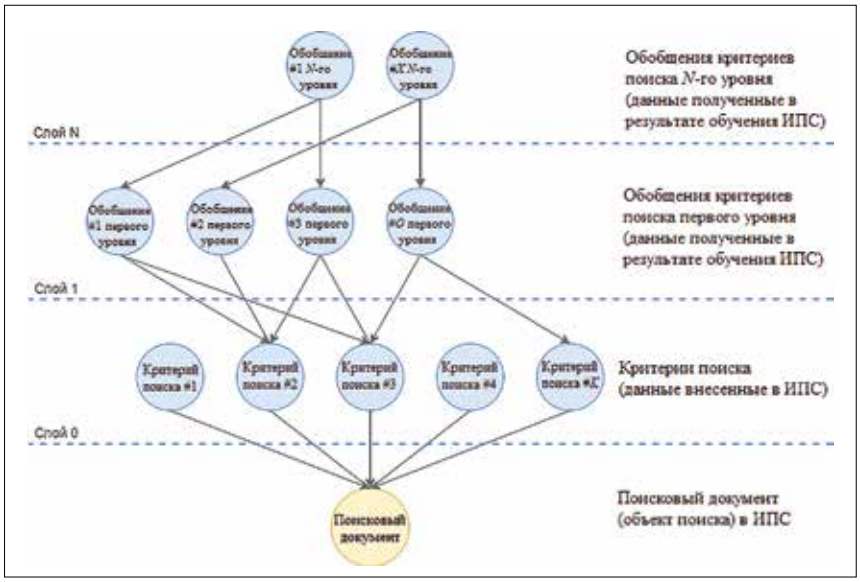

Рисунок 1 – Пример графовой модели организации данных в разрезе одного поискового документа

В соответствии с гипотезой, сформулированной при постановке задачи, пользователь ИПС может не обладать необходимыми компетенциями для формирования точного поискового запроса и понимания предметной области запроса. Отсюда возникает необходимость реализации адаптивного поиска, с той точки зрения, что модель организации данных должна «обучаться» и накапливать

также данные, отсутствующие в ИПС, но необходимые для получения релевантного результата.

Исходя из вышеизложенного, модель организации данных ИПС целесообразно реализовать в виде графа с послойной организацией данных. Пример такой организации данных в разрезе одного поискового документа представлен на рис. 1.

Как видно из рис. 1, на самом нижем уровне находится узел с информацией о поисковом документе. Данный узел связан с узлами критериев поиска односторонней связью. В свою очередь узлы критериев связанны со следующим слоем обобщений данных в более широкие группы понятий. Данное обобщение позволит алгоритму поиска находить документы по более широким понятиям их критериев. На одном условном слое связи между узлами запрещены, чтобы предотвратить цикличность при поиске.

Рассмотрим пример использования предложенной модели в информационно-поисковой системе библиотеки. Поисковым документом может являться книга с определенным названием. Тогда критериями поиска могут быть фамилия автора, год издания, издательство, количество страниц и т. д. Эти данные изначально заносятся и хранятся в базе данных. Обобщениями, полученными при обучении ИПС, например, для такого критерия как год издания могут быть следующие: обобщение первого уровня – «период Второй мировой войны», обобщением второго уровня – «советский период» и т.д. Таким образом, не зная определенного года издания, но зная, что книга написана автором в Советском Союзе или во время Второй мировой войны, в результате запроса «Купала AND советский период» мы получим релевантные результаты поиска.

Таким образом, послойность организации данных и направление связей между слоями, отсутствие связей между узлами одного слоя предотвращают пересечение лишних поисковых документов и при использовании слоев обобщений позволяют сделать поиск адаптивным, а результаты релевантными. Поиск осуществляется в направлении от верхнего слоя (N) к нижнему, а результаты поиска также будут содержать информацию одного верхнего слоя (нулевого).

За счет использования предложенной модели также решаются проблемы варьирования глубины поиска и эффекта от функции совпадения векторов поиска. Эти проблемы решаются за счет использования слоев обобщений критериев поиска. Используя обобщения более высоко уровня, можно увеличить глубину поиска, а при использовании обобщений более низкого уровня – уменьшить. Аналогично за счет обобщений решается проблема жесткости релевантных результатов поиска при использовании логических выражений.

В целом использование графа в качестве структуры данных в большинстве случаев будет являться оптимальным, т. к. позволяет сохранить все связи между сущностями, а также открывает широкий набор инструментов и подходов работы с графом.

Анализ быстродействия предложенной модели. Важной задачей при разработке модели адаптивного поиска является ее быстродействие, т. к. необходимо обрабатывать большой объем данных за приемлемое время ожидания пользователя. Исходя из этого, необходимо рассмотреть пограничные случаи работы модели, при которых поиск будет максимально неэффективным.

Количество связей между узлами двух слоев на заданном уровне определяется функцией $L\ (k, m, n)$, где k – количество узлов первого слоя, m – количество узлов второго слоя, n – номер уровня и при этом $k > 0; m > 0; n \geq 0; k, m, n \in N$. Количество узлов следующего слоя задается функцией $G\ (t, n)$, где t – количество узлов предыдущего слоя, n – номер уровня и при этом $t > 0; n \geq 0; t, n \in N$. Общее количество связей в многослойном графе R, с ограничениями на связи описывается формулой 1.

$$R(t,n,L,G) = \begin{cases} R(G(t,n), n-1, L, G) + L(t, G(t,n), n), & n > 0 \\ 0, & n = 0 \end{cases}, \quad (1)$$

где t – количество узлов на начальном слое; $t > 0$; $t \in N$;

n – общее количество слоев; $n \geq 0$; $n \in N$;

L – функция, определяющая количество связей между узлами двух слоев на заданном уровне;

G – функция, определяющая количество узлов следующего слоя.

Общее количество узлов E определяется по формуле 2.

$$E(t,n,G) = \begin{cases} E(G(t,n), n-1, G) + G(t,n), & n > 0 \\ 0, & n = 0 \end{cases}. \quad (2)$$

Крайним худшим случаем для поиска будет являться количество связей, которое необходимо проверить с самого верхнего слоя до самого нижнего, то есть обойти весь граф. Так как на самом верхнем слое находятся максимальные обобщения, а самый нижний слой описывает поисковый документ, то, в случае если пользователь введет такой запрос, что будет логическая операция конъюнкции и все критерии поиска будут попадать на верхний слой (одни только обобщения), необходимо будет проверить весь граф для нахождения нужного поискового документа. Функция, описывающая данный крайний

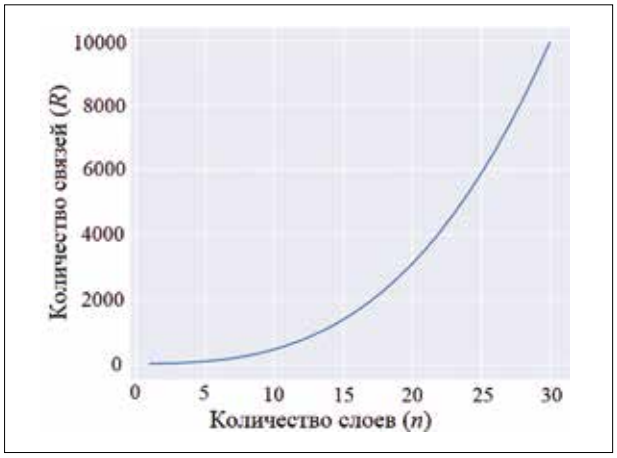

Рисунок 2 – График зависимости скорости роста числа связей от количества слоев

Рисунок 3 – График зависимости скорости роста числа узлов от количества слоев

случай, создает связи между каждым узлом начального слоя и каждым узлом следующего и имеет вид, представленный в формуле 3:

$$L_{max} = (k, m, n) = km \quad (3)$$

Формула (4) определяет линейную (l) скорость роста количества критериев поиска, которая на каждом следующем слое добавляет ровно один узел:

$$G_l(t, n) = t + 1 \quad (4)$$

Тогда скорость роста числа связей $R(1, n, L_{max}, Gl)$ и количества узлов $E(1, n, G_l)$ на n уровнях будет иметь вид, представленный на рис. 2 и 3.

Очевидно, что даже при небольшом количестве слоев $n \leq 30$, число узлов и связей довольно сильно различаются порядком, а сама зависимость далеко не линейна и ближе к степенной или экспоненциальной. Из этого следует, что нецелесообразно формировать большое количество слоев обобщений.

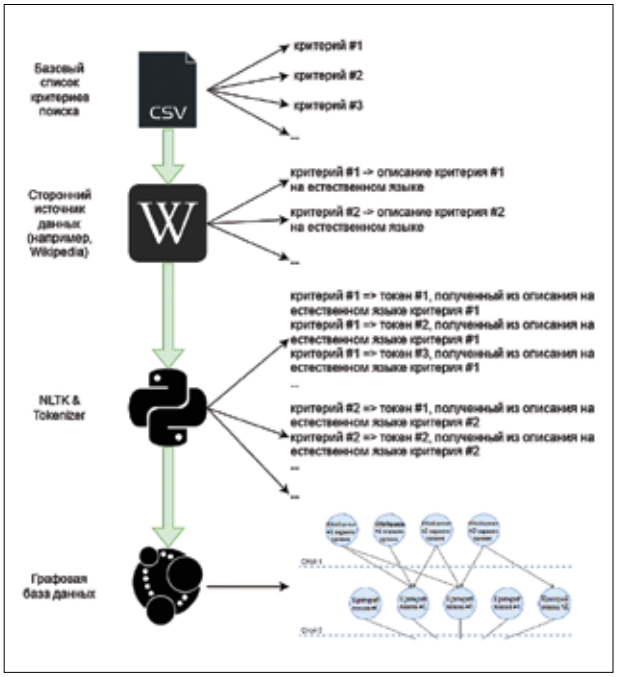

Рисунок 4 – Базовая модель подготовки данных для ИПС

Однако на практике их количество и так едва превысит даже десяток, а предложенная модель позволит эффективно (с точки зрения быстродействия) обработать довольно существенный объем данных и обобщений. В случае необходимости увеличения числа обобщений следует использовать кластерную систему вершин и партиционирование данных.

Модель подготовки и обработки данных ИПС. Помимо задачи непосредственно поиска данных, первостепенным является их подготовка, обработка и загрузка в графовую базу данных. Входными данными будут поисковые документы и список критериев их поиска. Они должны быть максимально конкретными, чтобы алгоритму обучения базы данных (формирования обобщений критериев поиска) было проще выявить нужные обобщения и классы, к которым данные критерии относятся. Выходными данными являются сами критерии поиска, а также список обобщений, к которым данные критерии относятся. Для формирования таких выходных данных на первом этапе необходимо дополнить каждый входной критерий каким-либо его определением или характеристикой на естественном языке. На втором этапе полученное из стороннего источника данных описание на естественном языке необходимо разбить на отдельные части и выбрать из них ключевые лексемы (токенизировать).

Процесс формирования обобщений и обучения базы данных должен быть автоматизирован, не быть привязанным к конкретному типу или схеме источника данных для формирования обобщений и при помощи технологий машинного

обучения самостоятельно выбирать значимые свойства из описания на естественном языке, формировать из них связи и сохранять в графовую базу данных. Причем алгоритм формирования обобщений можно реализовывать неограниченное число раз, каждый раз подставляя результат предыдущего выполнения и на выходе получать все более и более обобщенные данные на основе определений. Данная модель автоматизирует, существенно ускоряет и удешевляет процесс подготовки и загрузки данных в базу данных.

Очевидно, что процесс выявления обобщений и формирования иерархической структуры будет сильно отличаться от задачи к задаче и зависеть от предметной области данных информационно-поисковой системы.

Для машинного обучения базы данных информационно-поисковой системы можно использовать библиотеку обработки естественных текстов NLTK [5].

Базовая модель подготовки данных для ИПС представлена на рис. 4.

Алгоритм токенизации данных. Для получения токенов (ключевых лексем) из описания критериев поиска на естественном языке и связанных с каждым критерием поиска необходимо выполнить следующую последовательность действий:

– шаг 1. Входящее описание разбивается на именные группы (словосочитания). Выбираются словосочетания, в которых имя существительное является вершиной, то есть главным словом, определяющим характеристику всей составляющей;

– шаг 2. Выполнить цикл по всем получившимся именным группа и каждую из них разбить на слова;

– шаг 3. Из полученных слов исключить все «стоп-слова», все знаки пунктуации;

– шаг 4. На заключительном шаге формируется очищенная именная группа, которая и попадет в базу данных как одно из обобщений.

Блок схема алгоритма токенизации представлена на рис. 5.

Поисковый алгоритм. Поисковой запрос может включать в себя основные логические операции,

Рисунок 5 – Блок-схема алгоритма токенизации

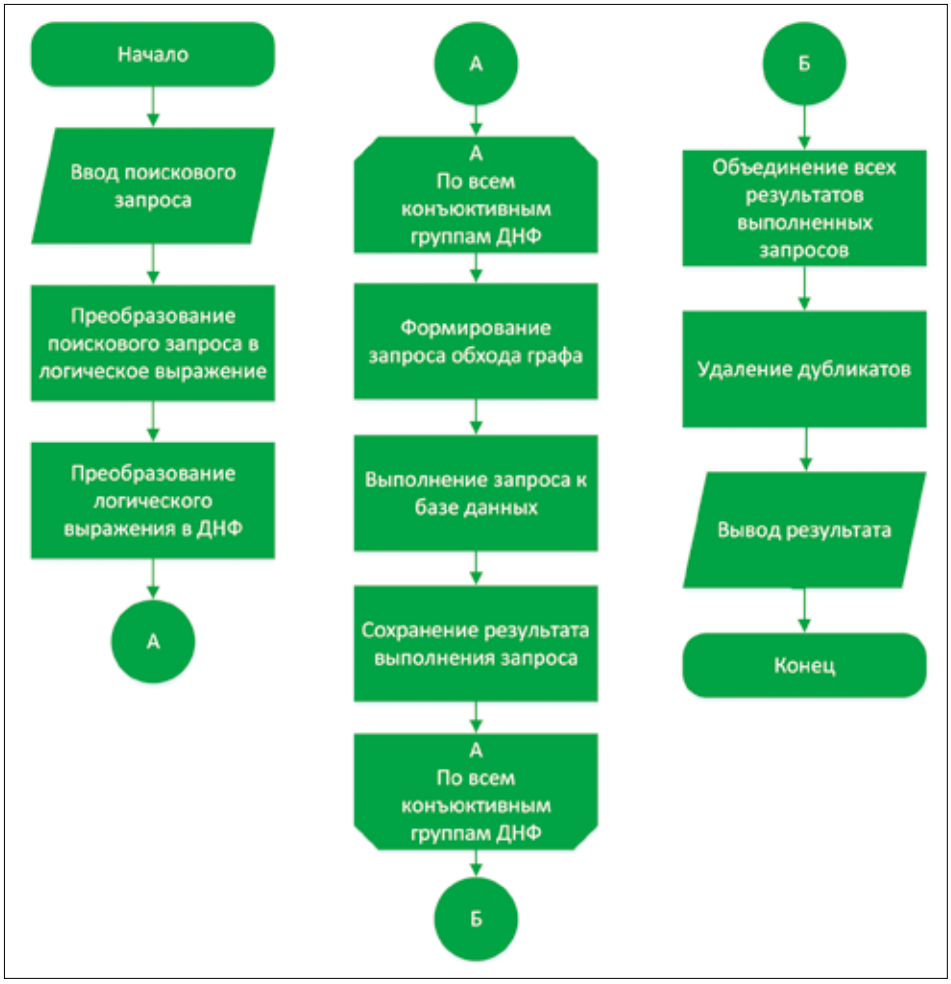

Рисунок 6 – Блок-схема обобщенного поискового алгоритма

такие как И, ИЛИ, НЕ, а также операцию группировки с приоритетом, а сам процесс поиска представляет собой обход по графу.

Однако, прежде чем делать обход по графу, необходимо преобразовать входящее выражение в дизъюнктивную нормальную форму (ДНФ). В качестве логических литералов выступают критерии поиска. ДНФ позволит преобразовать любое входящее логическое выражение в вид дизъюнкции конъюнкций литералов, что позволит произвести минимизацию логического выражения и ускорить алгоритм поиска (за счет уменьшения количества логических литералов) и даст возможность разбить алгоритм на три этапа:

– преобразование исходного выражения в ДНФ. Для этого необходимо закодировать каждый критерий поиска в логический литерал (например, (критерий 1 ∧ критерий 2) ∨ (критерий 3 ∧ критерий 2));

– поиск всех документов, которые должны обладать несколькими критериями одновременно, при этом если критерии находится на более высоком уровне агрегации, то вначале необходимо найти все критерии на самом низком слое связанные с текущим. То есть выполняется цикл по всем конъюнктивным группам и формирование запросов для обхода графа с учетом включения всех критериев поиска;

– объединение всех результатов предыдущего этапа и удаление встречающихся дубликатов.

Для поиска путей на графе используется так называемый двусторонний поиск в ширину. Он запускается одновременно из двух вершин (начальной и конечной) и останавливается в тот момент, когда общая вершина была найдена. Данный способ обхода графа не противоречит предложенной модели (в конечном счете поиск будет осуществляться, как и обозначено, начиная с верхних слоев (обобщений) графа), однако позволяет ускорить этот процесс. Блок-схема обобщенного поискового алгоритма представлена на рис. 6.

Так как входящее выражение от пользователя может быть неоптимальным с точки зрения логики (лишние повторы и комбинации), следует отметить, что в данном случае алгоритмизация минимизации логических выражений является *NP*-полной задачей и существенно зависит от сложности логического выражения, которое можно составить при формировании запроса поиска [6].

Алгоритм ранжирования результатов поиска. Для решения проблемы ранжирования результатов поискового запроса предлагается алгоритм подсказок, который предоставит пользователю дополнительную информацию (основываясь на полученных результатах поиска) для ранжирования после первой итерации поиска. Это позволит пользователю при последующем уточнении поискового запроса применить уже полученную информацию, например выбрать чаще встречающиеся в результате критерии поиска и таким образом получить результат в некотором ранжированном виде.

Для реализации данной идеи в предложенной модели после каждого поискового запроса можно формировать список «подсказок», сформированный из узлов непосредственно связанных с узлами результатов поиска. После чего выполнить подсчет количества каждого из узлов и уже выдать результат пользователю в ранжированном виде. Благодаря предложенной модели, ранжирование результатов можно осуществлять даже с учетом значимости критериев поиска, явно не заданных в поисковом запросе.

Заключение. В результате проведенного исследования были проведен анализ предметной области, обзор и анализ существующих технологий ИПС. Выявлены преимущества и недостатки существующих моделей и алгоритмов. Сформулирована задача по разработке модели и алгоритмов для адаптивного поиска по логическим выражениям, исключающих недостатки аналогичных существующих методов поиска, их программной реализации в виде веб-приложения. Были разработаны модель организации данных ИПС, модель подготовки и обработки данных ИПС. Предложенная модель организации данных обладает рядом преимуществ и исключают некоторые недостатки существующей булевой модели поиска.

Проведен анализ быстродействия предложенной модели организации данных. Рассмотрен граничный случай, при котором поиск будет наиболее неэффективным (менее быстродейственным). Установлена зависимость быстродействия поиска от числа слоев организации данных. Предложенная модель подготовки и обработки данных позволяет автоматизировать, существенно ускорить и удешевить процесс подготовки и загрузки данных в базу данных, в том числе слоев обобщения критериев поиска. Подготовка и импорт обобщений не привязан к конкретной схеме или типу сторонней информационной системы. Разработан алгоритм токенизации, необходимый для формирования слоев обобщений предложенной модели. Алгоритм позволяет автоматически формировать слои обобщений критериев поиска, имеющихся в ИПС, путем разбиения на отдельные токены описания критериев, полученные (с помощью библиотеки обработки естественных текстов NLTK) из сторонних информационных источников любого типа (например, Wikipedia). Разработан поисковый алгоритм предложенной модели, также позволяющий предварительно минимизировать логическое выражение, являющееся поисковым запросом, что позволит увеличить быстродействие алгоритма. Алгоритм ранжирования предоставляет пользователям подсказки о частоте встречаемости критериев поиска в полученных результатах, даже если эти критерии в явном виде отсутствовали в поисковом запросе. Используя эти подсказки, пользователь сам может выбрать значимость каждого критерии и осуществить ранжирование результатов.

Разработанные модели и алгоритмы могут применяться в ИПС, в том числе в адаптивных системах управления обучением, для поиска информации и построения индивидуальных образовательных траекторий обучаемых.

ЛИТЕРАТУРА

1. **Зуенко А. А.** Поисковые запросы на основе операций с логическими векторами / А. А. Зуенко, А. А. Алмаматов // Труды Кольского научного центра РАН. – 2013. – Выпуск № 5 (18). – с. 119–124.
2. **Шоркин, А. П.** Методы и алгоритмы информационного поиска на неточное соответствие / А. П. Шоркин // Доклады БГУИР. – 2011. – № 2 (56). – С. 13–15.
3. **Касекеева, А. Б.** Исследование методов информационного поиска / А. Б. Касекеева // BIG DATA and Advanced Analytics: сборник материалов V Международной научно-практической конференции, Минск, 13–14 марта 2019 г. В 2 ч. Ч. 1 / Белорусский государственный университет информатики и радиоэлектроники; редкол.: В. А. Богуш [и др.]. – Минск, 2019. – С. 324–330.
4. **Паризер Э.** За стеной фильтров. Что Интернет скрывает от вас. – М.: Альпина Бизнес Букс, 2012. – 304 с.
5. Документация по библиотеке обработке естественных текстов NLTK [Электронный ресурс]. – Режим доступа: https://nltk.org. – Дата доступа: 30.12.2021.
6. **Stephen Wolfram**. Undecidability and intractability [Электронный ресурс]. Режим доступа: https://www.wolframscience.com/nks/p768--undecidability-and-intractability. Дата доступа: 30.12.2021.

The article presents an analysis of models and algorithms for information retrieval in information retrieval systems, an analysis of existing problems and approaches to information retrieval, author's models and algorithms for data organization and search by logical expressions in information retrieval systems are proposed, the analysis of efficiency of the offered models and algorithms is made.

Keywords: *adaptive search, search algorithms, data organization model, graph database, search query, logical expressions, relevance, machine learning, information systems.*

Получено 30.01.2022.