Сервисы полнотекстового поиска в информационно-аналитической системе (Часть 2)[1]

Алгоритмы понимания текста

И.В. Соченков, Р.Е. Суворов

Аннотация: Настоящее исследование посвящено разработке структур данных и алгоритмов информационного поиска для обеспечения функций машинного «понимания» текстовой информации при решении поисково-аналитических задач. Статья представляет вторую часть исследования и содержит описание разработанных алгоритмов оценки релевантности и ранжирования результатов информационного поиска. Предлагаемые методы лежат в основе поисково-аналитических сервисов информационно-аналитической системы. Также во второй части рассматриваются вопросы экспериментальной проверки и практического применения созданного алгоритмического обеспечения.

Ключевые слова: информационный поиск, семантический поиск, полнотекстовый поиск, структуры данных для информационного поиска, алгоритмы ранжирования результатов информационного поиска, информационно-аналитическая система, оценка качества информационного поиска.

Введение

Настоящая статья представляет вторую часть исследования, посвященного вопросам реализации сервисов полнотекстового поиска в информационно-аналитической системе (ИАС).

Целью исследования является создание алгоритмического обеспечения сервисов ИАС, позволяющего реализовать качественный и эффективный полнотекстовый поиск и анализ текстов, базирующийся на семантических методах машинного «понимания» текстовой информации. Задачи исследования состоят в разработке алгоритмов и структур данных для полнотекстового поиска, интегрированных в ИАС, а также в экспериментальной проверке предложенных алгоритмов и оценке качества их работы.

Вторая часть исследования посвящена разработке алгоритмов оценки релевантности и ранжирования результатов информационного поиска. Предлагаемые методы лежат в основе поисково-аналитических сервисов ИАС, которые реализуют различные режимы поиска. Также во второй части рассматриваются вопросы экспериментальной проверки и практического применения созданного алгоритмического обеспечения в ИАС.

Основу предлагаемых алгоритмов составляет модель текста, являющаяся развитием реляционно-ситуационной модели Г.С. Осипова - Г.А. Золотовой [1], и метод сравнения текстов, предложенные в работе [2]. Разработанные алгоритмы и структуры данных, предназначенные для обеспе-

[1] Работа выполнена при финансовой поддержке Минобрнауки России по Государственному контракту № 14.514.11.4134 в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007-2013 годы».

чения функций информационного поиска, учитывают лингвистическую информацию, выделяемую из текстов автоматически на различных уровнях анализа: лексико-морфологическом, синтаксическом и семантическом.

Описываемое в статье алгоритмическое обеспечение ИАС для реализации сервисов полнотекстового поиска является результатом многолетних исследований И.В. Соченкова в области методов построения поисковых индексов и ранжирования результатов поиска. Приводимые результаты получены в коллективе исследователей проекта Exactus [1, 3, 4] под руководством Г.С. Осипова. Экспериментальная проверка исследуемых алгоритмов и оценка качества результатов поиска выполнена Р.Е. Суворовым.

1. Представление поискового запроса

Поисковый запрос пользователя ИАС определяет зону поиска в виде ограничений на метаинформацию, связанную с искомыми документами, а также содержит формулировку информационной потребности на естественном языке (ЕЯ). В ИАС научно-технических документов зона поиска содержит следующую метаинформацию:

1) название документа, содержащее слово, фразу или предложение;

2) ФИО авторов документа;

3) тематические рубрики (предметную область или отрасль науки) документа;

4) дату опубликования документа или дату загрузки в систему;

5) источник документа (сайт или иной ресурс, из которого был загружен документ);

6) тип документа (диссертация, научная статья, автореферат, тезисы доклада, монография, курсовая работа и т.д.);

7) формат документа (PDF, MS Word, HTML и т.д.).

Название, ФИО авторов документа наряду с самим поисковым запросом являются текстовыми данными. Эти данные подвергаются машинному лингвистическому анализу, в результате которого строится индекс текста запроса (ИТЗ). Каждому вхождению слова запроса в ИТЗ соответствует элемент данных (ЭД), содержащий следующую информацию:

1. Идентификатор нормальной формы лексемы (ИНФЛ), соответствующей вхождению слова в текст запроса (word_id).

2. Тег метаинформации, к которой относится вхождение слова запросе (tag_id). Слова запроса, которые должны присутствовать в заголовке искомых документов помечаются соответствующим тегом. Аналогичным образом помечаются слова, относящиеся к ФИО авторов. Слова запроса могут быть также помечены произвольными тегами языка гипертекстовой разметки (например, для поиска документов, которые содержат слова запроса в гипертекстовых ссылках на другие документы).

3. Идентификатор формы (ИФ) вхождения слова в текст запроса (form).

4. Порядковый номер вхождения слова в текст запроса (word_no). В случае коротких запросов в ходе лингвистического анализа может быть не снята омонимия нормальных форм: если у омонимов различаются нормальные формы, то в ИТЗ добавляются дублирующие ЭД, отличающиеся только word_id и, возможно, ИФ.

5. Порядковый номер ЭД в ИТЗ.

6. Порядковый номер предложения в тексте запроса, в которое входит слово.

7. Порядковый номер ЭД, представляющего вхождение слова в текст запроса, которое является главным элементом синтаксической группы (ГЭСГ), содержащей текущее вхождение слова (имеет пустое значение, если ЭД соответствует главному слову синтаксической группы).

8. Множество значений синтаксем (если эти значения приписаны вхождению слова в текст).

9. Множество значений синтаксем, связанных семантической связью в тексте запроса с текущим вхождением слова в том же предложении.

10. Порядковый номер фразы, в которую входит слово (concept_id). Под фразой здесь понимается совокупность слов запроса в пределах одного предложения, объединенных с помощью фигурных скобок «{…}» [2].

11. Служебные флаги.

Для каждой фразы задается идентификатор, который позволяет при поиске объединять фразы, связанные в запросе тезаурусными отношениями (например, при пополнении исходного запроса терминами из тезауруса) (phrase_id). Этот идентификатор приписывается каждому

ЭД, и на его основе соотносятся вхождения слов, относящиеся к различным фразам.

Таким образом, ИТЗ представляет собой упорядоченное (по номеру вхождения слова) множество ЭД. ИТЗ совместно с нетекстовыми параметрами, задающими зону поиска, составляют поисковый запрос.

Каждый ЭД ИТЗ содержит дополнительно поле «вес» вхождения слова (weight). Изначально вес вхождения слова определяется на основе теговой разметки запроса: с помощью назначения весов тегов метаданных можно управлять значимостью различных частей запроса (например, слова в поисковом поле «название документа» важнее фамилий авторов, заданных в поисковом поле «авторы»).

Служебные флаги представляют информацию о помеченных (с помощью языка запросов) вхождениях слов в поисковый запрос и могут содержать следующие значения *приоритета слова*:

– Исключаемое слово (t0) – найденные документы не должны содержать вхождения этого слова (возможно, с учетом конкретной словоформы) в тексте (либо в метаданных, если исключаемое слово помечено соответствующим тегом).

– Исключаемое слово предложения (t1) – найденные документы не должны содержать вхождения этого слова (возможно, с учетом конкретной формы этого слова) в тех предложениях, которые релевантны остальному запросу (либо в метаданных, если исключаемое слово помечено соответствующим тегом).

– Обязательное слово (t2) – слово должно присутствовать в текстах найденных документов (возможно, с учетом конкретной формы этого слова).

– Обычное слово (t3) – слово может отсутствовать или присутствовать в текстах найденных документов (при многословных запросах).

– Факультативное слово (t4) – может отсутствовать в текстах найденных документов (при многословных запросах). Этот флаг устанавливается для малозначимых слов, например, выступающих в роли распространяющих членов предложения, определений и т.п.

– Факультативное слово предложения (t5) – может отсутствовать в найденных предложениях текстов документов (при многословных запросах). Этот флаг устанавливается для малозначимых слов, например, выступающих в роли распространяющих членов предложения, определений и т.п.

Процедура поиска и оценки релевантности начинается с формирования системного представления поискового запроса. Для сформированного ИТЗ уточняются веса слов в соответствии с принципами, изложенными в работе [2]. Запрос может быть расширен, например, за счет синонимов путем вставки ЭД в ИТЗ, соответствующих добавляемым словам.

2. Алгоритмы оценки релевантности и ранжирования результатов информационного поиска

Входными данными процедуры информационного поиска является ИТЗ и ограничения на нетекстовые данные искомых документов, заданные пользователем.

Общий алгоритм оценки релевантности и ранжирования результатов информационного поиска содержит следующие шаги:

1. Формирование области поиска на основе заданных в поисковом запросе ограничений на нетекстовые данные искомых документов.

2. Выборка и фильтрация информации из поисковых индексов.

3. Предварительная оценка релевантности найденных документов словам запроса.

4. Итоговое ранжирование и представление результатов.

Рассмотрим эти этапы работы алгоритма.

2.1. Формирование области поиска

Под формированием области поиска подразумевается определение множества документов, удовлетворяющих заданным ограничениям, среди которых будет выполняться текстовый поиск. Для этой цели применяется вспомогательная структура данных, формируемая на этапе индексирования текстовой коллекции и содержащая нетекстовые метаданные документов коллекции – реестр метаданных.

Для каждого документа ЭД реестра метаданных содержит следующие поля данных:

1. Идентификатор сайта документа (site_id) или другого ресурса, из которого был загружен документ.

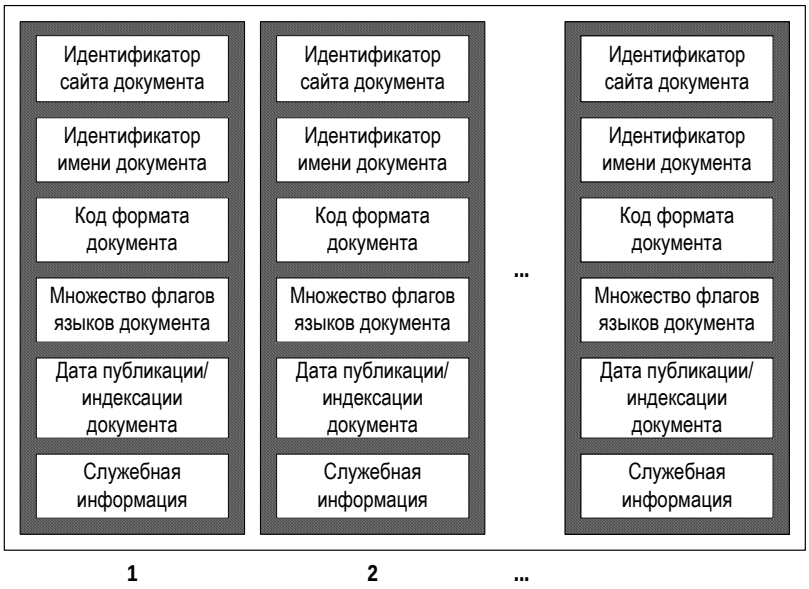

Рис. 1. Структура реестра метаданных документов

2. Идентификатор имени документа (doc_id), определяющий имя документа на сайте.

3. Код формата документа (MS Word, PDF, HTML и др.).

4. Множество языков документа (в виде набора бинарных флагов), которые были обнаружены лингвистическим анализатором в тексте.

5. Дата публикации / индексации документа.

Код документа, идентифицирующий документ в ИПИ, является порядковым номером соответствующего ЭД в реестре метаданных, а сам реестр представляет собой массив в оперативной памяти (Рис 1.).

Путем последовательного просмотра реестра метаданных осуществляется сопоставление содержащейся в нем информации с зоной поиска. В результате формируется массив кодов документов, удовлетворяющих условиям поиска по нетекстовым метаданным. Этот массив представляет область поиска.

2.2. Выборка и фильтрация информации

Выборка информации из инвертированного поискового индекса (ИПИ) (по ИНФЛ в качестве ключа) позволяет получить информацию о вхождениях слов в тексты коллекции документов в виде σ-последовательностей [5]. Выбранные σ-последовательности подвергаются первичной фильтрации: из σ-последовательностей исключаются те σ-ЭД, которые относятся к документам, не вошедшим в область поиска. Блок-схема алгоритма первичной фильтрации представлена на Рис.2.

Этот алгоритм реализует операцию «пересечения» σ-последовательности с областью поиска: в результате в σ-последовательности остаются только те ЭД, которые относятся к документам, входящим в область поиска. Предполагается, что область поиска (doc_id_seq) непустая. В противном случае в коллекции текстов отсутствуют документы, удовлетворяющие зоне поиска, и поиск завершен.

Заметим, что если область поиска содержит больше половины документов коллекции, то оптимальным с точки зрения вычислительной сложности будет следующий алгоритм первичной фильтрации:

1. «Инвертировать» область поиска: на основе реестра метаданных построить массив кодов документов, не удовлетворяющих зоне поиска, – исключаемые документы.

2. Применить алгоритм фильтрации σ-последовательностей, описанный в работе [5], удаляющий из σ-последовательности σ-ЭД, соответствующие исключаемым документам.

Если зона поиска, заданная пользователем, не содержит ограничений на нетекстовые метаданные, то формирование области поиска не производятся, а первичная фильтрация не выполняется.

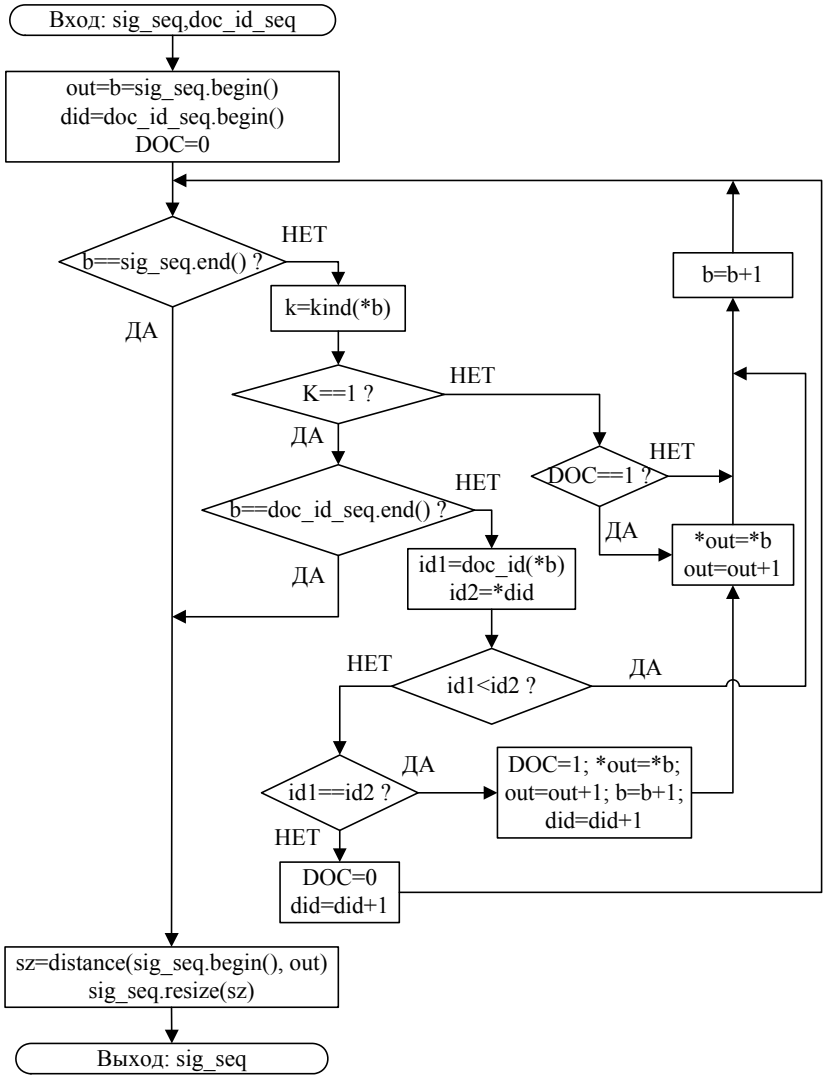

Рис. 2. Блок-схема алгоритма первичной фильтрации σ–последовательностей

Следующим этапом после выборки и первичной фильтрации информации является фильтрация σ-последовательностей по словам запроса в соответствии со следующими критериями:

1. Из σ-последовательности исключаются σ-цепочки, соответствующие тем документам, в которые рассматриваемое слово запроса входит с малым общим суммарным весом:

$$v(w) = \frac{\log\left(1 + \sum\limits_{w' \in \{w'' \in W \mid \delta(w) = \delta(w'')\}} \omega(\tau(w'))\right)}{\log\left(1 + \sum\limits_{w' \in W} \omega(\tau(w'))\right)} \le v_{\min}$$

– в обозначениях, принятых в работе [2]. Этот критерий необходимо применять для лексем, часто встречающихся в текстах коллекции. В результате из рассмотрения исключаются априорно нерелевантные документы, и сокращается время оценки релевантности.

2. Из σ-последовательности исключаются σ-ЭД, соответствующие вхождениям рассматриваемого слова в тексты документов коллекции, и при этом для этих вхождений выполнено хотя бы одно из следующих условий:

2.1. У слова в тексте запроса и в тексте документа не совпадает тег разметки метаданных (tag_id).

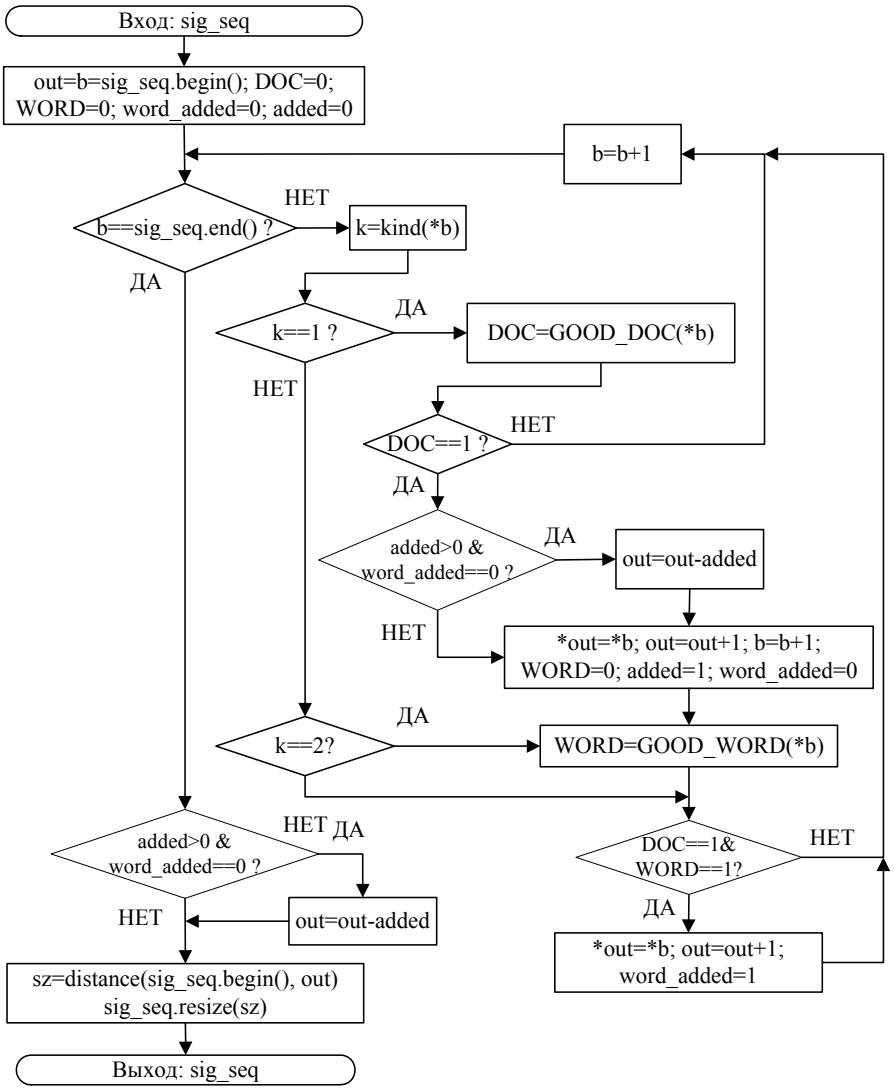

Рис. 3. Блок-схема алгоритма фильтрации σ–последовательности по слову запроса

2.2. У слова в тексте запроса и в тексте документа не совпадают словоформы (form_id) – при поиске по точному совпадению.

Блок-схема алгоритма фильтрации σ-последовательности по слову запроса представлена на Рис. 3.

Отдельно отметим, что фильтрация (и последующие преобразования) σ-последовательностей выполняется отдельно для каждого вхождения слова в текст запроса (а не для множества ИНФЛ запроса). Это позволяет учесть при поиске тот факт, что запрос может состоять из нескольких предложений, причем предложения могут относиться к различным типам метаданных и содержать повторяющиеся слова.

После выборки и фильтрации данных ИПИ каждому вхождению слова в текст запроса сопоставлена σ-последовательность, содержащая информацию о документах, потенциально релевантных запросу пользователя. На следующем этапе информация в σ-ЭД этих последовательностей сопоставляется с информацией в ЭД ИТЗ, соответствующей словам запроса, – выполняется предварительная оценка релевантности.

2.3. Предварительная оценка релевантности найденных документов словам запроса

Предварительная оценка релевантности найденных документов словам запроса заключается

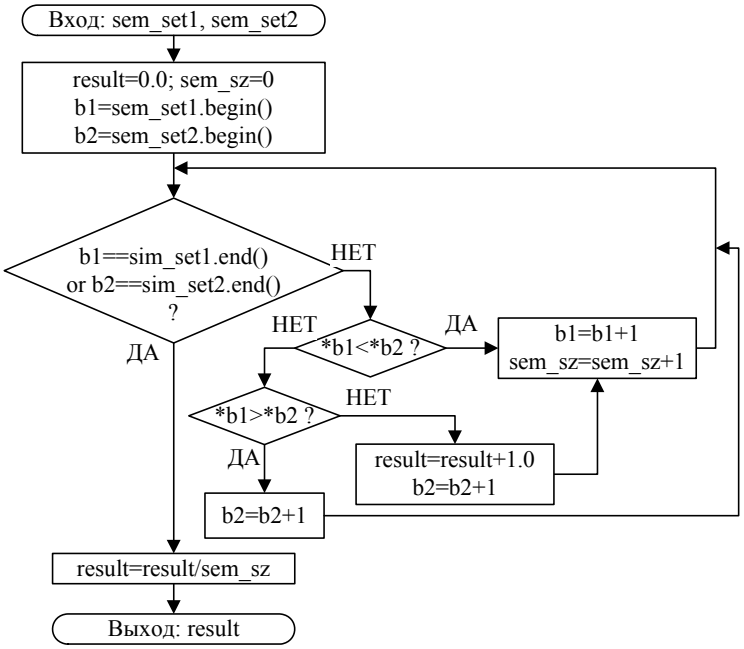

Рис. 4. Алгоритм сравнения двух множеств семантических значений

в преобразовании каждой σ-последовательности в соответствующую ρ-последовательность. ρ-последовательность – массив ρ-ЭД, содержащих следующий набор полей данных:

1) код документа (doc_id);

2) номер предложения, в которое входит найденное вхождение слова в тексте документа (sent_no);

3) порядковый номер вхождения слова в предложении найденного документа (sofs);

4) порядковый номер ЭД в ИТЗ, соответствующего σ-последовательности (word_idx);

5) разность между позицией слова, представляемого ЭД, и позицией слова, являющегося ГЭСГ, в которую входит рассматриваемое вхождение слова в тексте найденного документа (mwp) [5];

6) номер предложения в ИТЗ, в которое входит слово, соответствующее найденной σ-последовательности (rsent);

7) вес найденного вхождения слова в тексте документа (tf);

8) флаг, совпадения формы найденного вхождения слова в тексте документа с формой соответствующего вхождения слова в тексте запроса (match_form);

9) оценки релевантности найденного вхождения слова в документе, вычисленные на основе сопоставления множеств семантических значений и семантических связей в соответствии с формулами (8) и (9) из работы [2] (roles_w, rels_w).

Поля 1–3, 5,7 заполняются на основе информации, представленной в соответствующих σ-ЭД, поля 4, 6 – на основе соответственного ЭД ИТЗ. Поля 8, 9 вычисляются на основе сравнения формы слова и множеств значений синтаксем и семантических связей, представленных ЭД ИТЗ и σ-ЭД. Отметим, что по соображениям вычислительной эффективности множества семантических значений (МСЗ) упорядочены по возрастанию числовых идентификаторов семантических значений, составляющих эти множества. В этом случае операция пересечения множеств выполняется за время, пропорциональное сумме мощностей этих множеств. Таким образом, сложность сравнения двух МСЗ в условиях выбранной схемы представления семантической информации составляет [5]: 3+3=6 и 5+5=10 итераций цикла при расчете величин roles_w и rels_w, соответственно. Алгоритм сравнения двух МСЗ приведен на Рис. 4.

Алгоритм преобразования σ-последовательности в соответствующую ρ-последовательность линеен (Рис. 5), количество ρ-ЭД

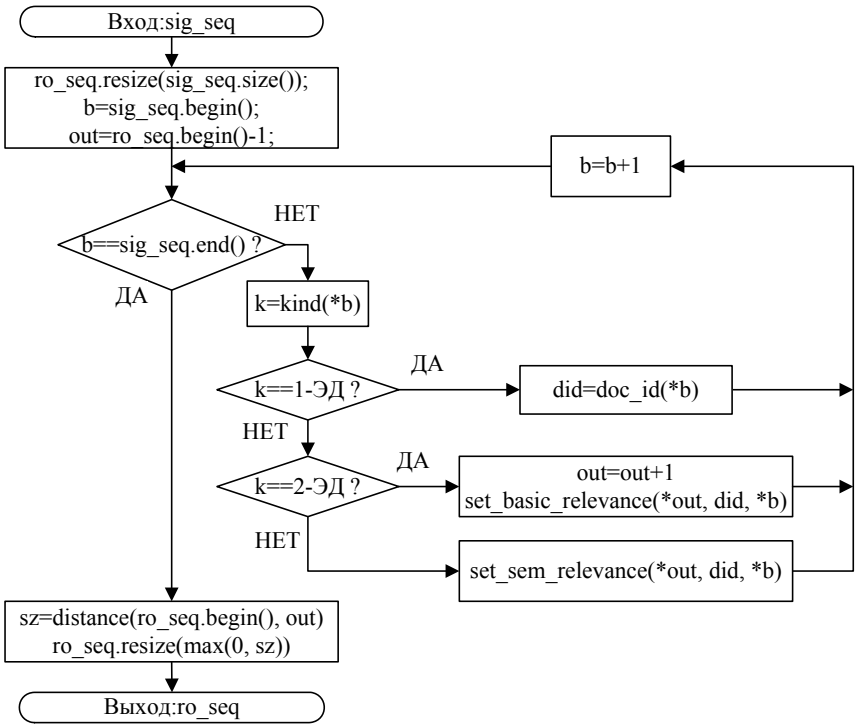

Рис. 5. Алгоритм преобразования σ-последовательности в ρ-последовательность

в результирующей ρ-последовательности не превосходит количества σ-ЭД в исходной σ-последовательности минус один.

В силу того, что указанный алгоритм обрабатывает входные данные последовательно, в результирующей ρ-последовательности сохраняется линейный порядок на множестве ρ-ЭД, заданный следующим правилом (ρ-*правило)*. ρ-ЭД r_1 предшествует другому ρ-ЭД r_2 тогда и только тогда, когда

1. Код документа, представляемый ρ-ЭД r_1 меньше кода документа, представляемого ρ-ЭД r_2: $doc\_id(r_1)<doc\_id(r_2)$.

2. Код документа, представляемый ρ-ЭД r_1 совпадает с кодом документа, представляемым ρ-ЭД r_2: $doc\_id(r_1)==doc\_id(r_2)$, и при этом:

2.1. Номер предложения в тексте запроса, соответствующий вхождению слова в найденном тексте, представляемого ρ-ЭД r_1, меньше номера предложения в тексте запроса, соответствующего вхождению слова в найденном тексте, представляемого ρ-ЭД r_2: $rsent(r_1)<rsent(r_2)$.

2.2. Номера предложений в тексте запроса, соответствующие вхождениям слов в найденном тексте, представляемых ρ-ЭД r_1 и r_2, совпадают $rsent(r_1)==rsent(r_2)$, и при этом

2.2.1. Номер предложения в тексте документа, соответствующий вхождению слова в найденном тексте, представляемого ρ-ЭД r_1, меньше номера предложения в тексте документа, соответствующего вхождению слова в найденном тексте, представляемого ρ-ЭД r_2: $sent\_no(r_1)<sent\_no(r_2)$.

2.2.2. Номера предложений в тексте документа, соответствующие вхождениям слов в найденном тексте, представляемых ρ-ЭД r_1 и r_2, совпадают $sent\_no(r_1)==sent\_no\ (r_2)$, и при этом

2.2.2.1. Смещение (от начала предложения) вхождения слова в найденном тексте, представляемого ρ-ЭД r_1, меньше смещения слова, представляемого ρ-ЭД r_2: $sofs(r_1)<sofs(r_2)$.

Фактически, ρ-ЭД в ρ-последовательности упорядочены сначала по коду документа, далее – по номеру предложения в тексте запроса, в котором находится слово, на основе нормальной формы которого найден соответствующий

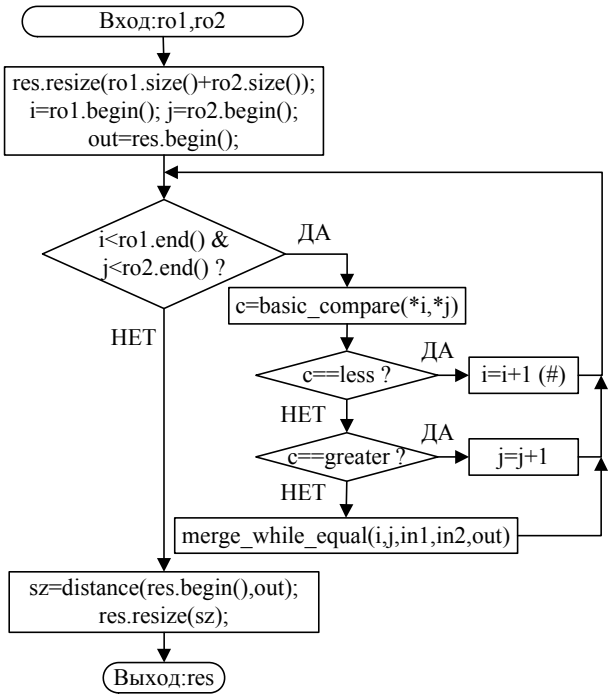

Рис.6. Блок-схема алгоритма условного соединения ρ-последовательность

ρ-ЭД, затем – по позиции словоупотребления в найденном тексте.

В результате каждому словоупотреблению в запросе (представляемому ЭД в ИТЗ) сопоставлена ρ-последовательность, содержащая сведения об встречаемости лексемы, соответствующей указанному словоупотреблению, в текстах коллекции документов.

Для формирования множества документов, потенциально релевантных запросу в целом, на классе ρ-последовательностей реализуются операции слияния, условного соединения, а также одностороннего условного соединения и условного исключения элементов. При реализации этих операций используется ρ-*правило* в качестве отношения линейного порядка на множестве ρ-ЭД.

Операция слияния $join(\rho_1,\rho_2)$ реализуется с помощью классического алгоритма слияния (merge) и дает в результате обычное объединение двух ρ-последовательностей с сохранением отношения линейного порядка: $join(\rho_1,\rho_2)=\rho_1\cup\rho_2$.

Операция условного соединения ρ-последовательностей позволяет получить ρ-последовательность ρ_3, содержащую только те элементы исходных ρ-последовательностей, которые удовлетворяют условию: $\rho_3 = cond\_join(\rho_1,\rho_2) =$

$$\left\{r\in\rho_1\cup\rho_2 \mid \forall r_1\in\rho_i \exists r_2\in\rho_j,\ i=\{1,2\}, i\neq j : P(r_1,r_2)=1\right\},$$

где $P(\rho_1,\rho_2)$ некоторый двухместный предикат на ρ-ЭД. Эта операция симметрична тогда и только тогда, когда симметричен предикат P. Например, если требуется, чтобы найденные словоупотребления в документах коллекции встречались в пределах одного предложения, то предикат $P(\rho_1,\rho_2)$ имеет следующий вид: $P_1(\rho_1,\rho_2):=doc\_id(r_1)==doc\_id(r_2)\&rsent(r_1)=rsent(r_2)$ $\&\ sent\_no(r_1)==sent\_no(r_2)$.

Операция условного соединения ρ-последовательностей реализуется с помощью алгоритма, блок-схема которого приведена на Рис 6. В качестве операции *basic_compare* в этом алгоритме используются условия 1 – 2.2.2 ρ-*правила* (кроме 2.2.2.1). В алгоритме слияния частей ρ-последовательностей, удовлетворяющих заданному условию, (*merge_while_equal*), блок-схема которого приведена на Рис. 7, в качестве условия *compare* выступает ρ-*правило*, а в качестве *is_equal* – $P(\rho_1,\rho_2)$. Блок-схема алгоритма копирования ρ-ЭД, эквивалентных в смысле заданного условия, (*copy_equal_tail*) приведена на Рис. 8.

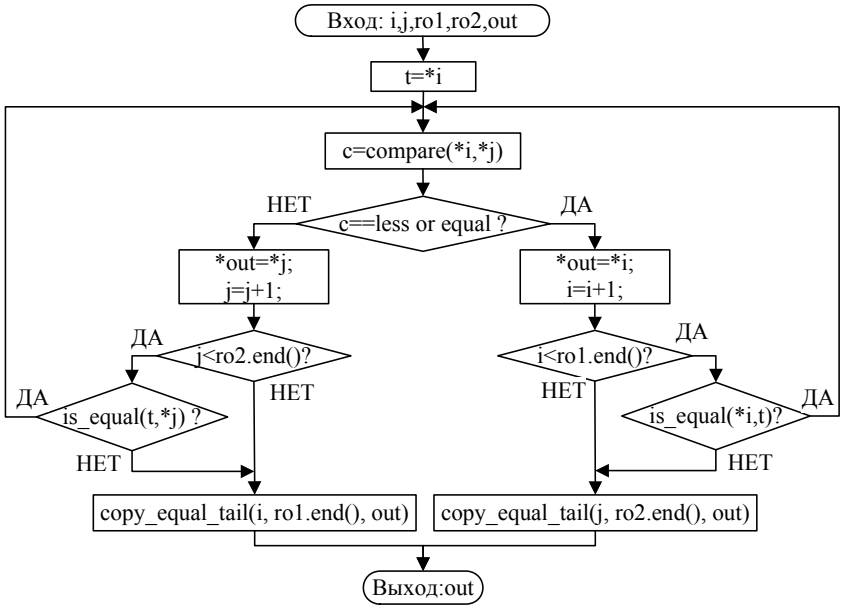

Рис. 7. Блок-схема алгоритма слияния частей ρ-последовательностей, удовлетворяющих заданному условию

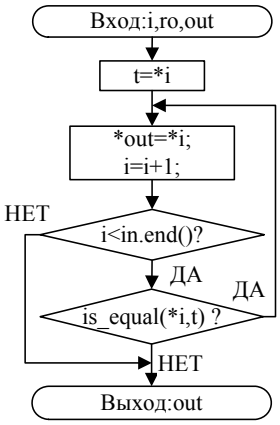

Рис. 8. Блок-схема алгоритма копирования ρ-ЭД, эквивалентных в смысле заданного условия

Если требуется, чтобы какие-либо слова запроса обязательно присутствовали в документах совместно (но не обязательно в пределах одного предложения), то в вышеприведенных алгоритмах в качестве операции *basic_compare* используется условие 1 ρ-*правила*, а условие эквивалентности *is_equal* (предикат $P(\rho_1,\rho_2)$) представляется в виде: $P_2(\rho_1,\rho_2):= doc\_id(r_1)=doc\_id(r_2)$.

Операция одностороннего условного слияния ρ-последовательностей ρ_1 и ρ_2 позволяет пополнить ρ-последовательность ρ_1, ЭД из ρ_2, удовлетворяющих условиям, описанным выше:

$$\rho_3 = cond\_left\_join(\rho_1,\rho_2) =$$
$$\{r_1 \in \rho_1 \vee r_2 \in \rho_2 \mid \forall r_2 \in \rho_2 \exists r' \in \rho_1, \quad : P(r', r_2) = 1\}.$$

В общем случае эта операция несимметрична. Отметим, что операция одностороннего условного слияния ρ-последовательностей реализуется аналогично алгоритму условного соединения ρ-последовательностей (Рис. 6), с той лишь разницей, что на шаге, помеченным знаком (#), выполняются следующие 3 операции: **out=*i; out=out+1; i=i+1*.

Операция условного исключения элементов ρ-последовательности ρ_2 из ρ_1 с позволяет получить ρ-последовательность ρ_3, ЭД которой удовлетворяют следующему условию $\rho_3 = \{r_1 \in \rho_1 \mid \neg\exists r_2 \in \rho_2 : \quad P(r_1, r_2) = 1\}$. Блок-схема алгоритма условного исключения элементов ρ-последовательности ρ_2 из ρ_1 приведена

на Рис. 9. Результатом работы алгоритма является преобразованная ρ-последовательность ρ_1. В качестве предиката $P(\rho_1,\rho_2)$ могут выступать P_1 или P_2, рассмотренные выше.

Заметим, что при фразовом поиске к ρ-последовательностям, которые соответствуют словоупотреблениям запроса, составляющим фразу, попарно применяется операция *cond_join*, а затем над полученной ρ-последовательностью выполняется алгоритм фразовой фильтрации, который проверяет наличие соответствующих синтаксических связей между найденными вхождениями слов в тексты документов – Рис. 10. В этом алгоритме (а также на более поздних стадиях обработки – при оценке релевантности предложений) используется алго-

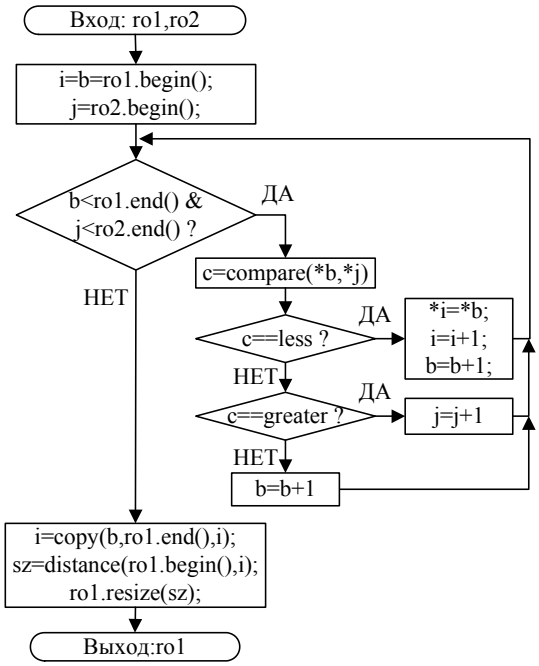

Рис. 9. Блок-схема алгоритма условного исключения элементов ρ-последовательностей

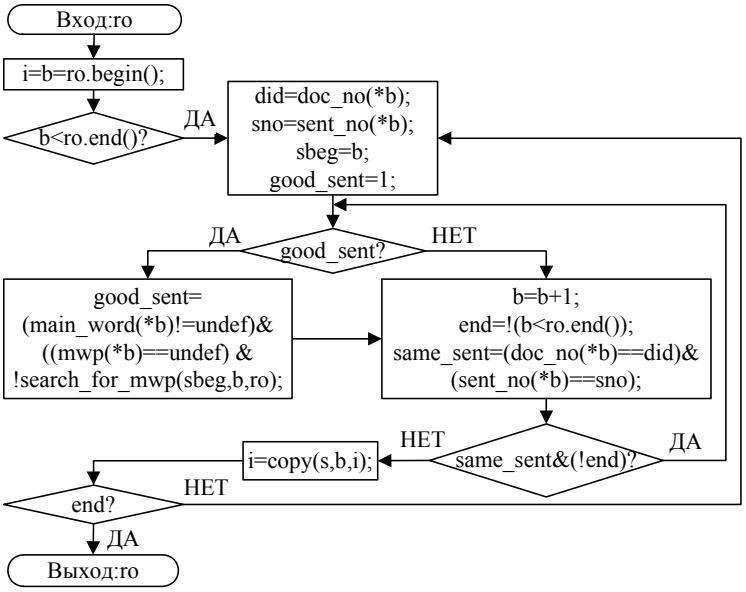

Рис. 10. Блок-схема алгоритма фразовой фильтрации ρ-последовательности

ритм поиска синтаксических связей в ρ-последовательности – Рис. 11. Он состоит в поиске в пределах предложения документа такого ρ-ЭД r_1, что этот r_1 связан синтаксической связью с некоторым другим ρ-ЭД r_2, и при этом словоупотребления в запросе, соответствующие r_1 и r_2, также синтаксически связаны.

С учетом вышеописанных алгоритмов формирование множества потенциально релевантных документов включает следующие стадии обработки ρ-последовательностей:

1. Слияние ρ-последовательностей, соотнесенных с ЭД ИТЗ, представляющих омонимичные варианты нормализации некоторого вхождения слова в тексте поискового запроса. В результате каждому словоупотреблению запроса соответствует одна ρ-последовательность: ρ_i^W.

2. Слияние полученных ρ-последовательностей ρ_i^W «по фразам»: попарное применение операции условного соединения к ρ-последовательностям, соотнесенным с вхождениями слов в запрос, составляющим фразу. К результирующей ρ-последовательности применяется алгоритм фразовой фильтрации. В результате каждая полученная ρ-последовательность ρ_i^P соответствует некоторой фразе запроса и характеризуется *приоритетом фразы*, который определяется как минимальное значение *приоритета слова* (t2-t5) из всех слов, составляющих фразу.

3. Формирование ρ-последовательности ρ^e, характеризующей множество найденных документов, в которых встречаются те слова запроса, которые должны быть исключены из результатов поиска.

4. Формирование ρ-последовательностей ρ_i^e, характеризующих множество предложений найденных документов, в которых встречаются те слова запроса, которые должны быть исключены из результатов поиска.

5. Слияние ρ-последовательностей ρ_i^P «по предложениям» с помощью операций соединения, условного слияния и одностороннего условного слияния. Предварительно к исходным ρ-последовательностям ρ_i^P, соответствующим фразам запроса, входящим в одно предложение,

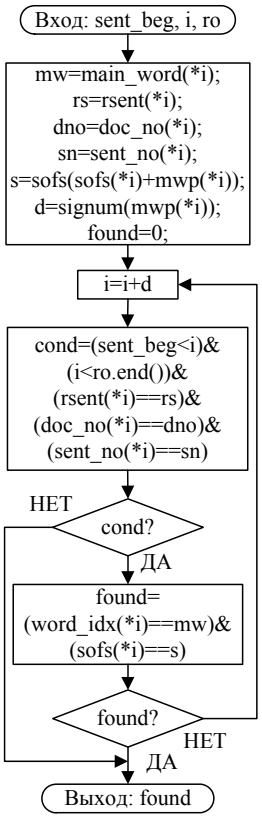

Рис. 11. Блок-схема алгоритма поиска синтаксических связей в ρ-последовательности

применяется операция условного исключения элементов (из ρ_i^P исключаются ρ^e, а затем ρ_i^e). В результате формируется множество таких ρ-последовательностей $R^S = \{\rho_i^S\}$, что каждая ρ-последовательность ρ_i^S соответствует некоторому предложению текста запроса и характеризуется *приоритетом предложения* (минимальное значение *приоритетов фраз* (t2-t5), составляющих предложение).

6. Слияние ρ-последовательностей ρ_i^S «по тегам»: с помощью операций соединения, условного слияния и одностороннего условного слияния, конкретный выбор которых зависит от *приоритета предложения*, формируется множество таких ρ-последовательностей $R^T = \{\rho_i^T\}$, что каждая ρ-последовательность ρ_i^T соответствует определенному тегу метаданных документа, который характеризуется

приоритетом тега (минимальное значение *приоритетов предложений* (t2-t5), помеченных соответствующим тегом).

7. Слияние ρ-последовательностей ρ_i^T аналогично п.6 в одну ρ-последовательность.

В результате формируется ρ-последовательность ρ^F, представляющая информацию об употреблениях слов запроса в текстах найденных документов, потенциально релевантных запросу.

2.4. Итоговое ранжирование результатов поиска

Полученная в ходе предварительной оценки релевантности найденных документов словам запроса ρ-последовательность ρ^F позволяет выполнить итоговое ранжирование результатов поиска. Эта процедура выполняется в 2 этапа.

На первом этапе за один просмотр ρ-последовательности формируется множество предварительных оценок релевантности предложений. Эти оценки вычисляются на основе критерия 1 из работы [2] и дополнительно содержат позиции диапазонов ρ-ЭД в ρ-последовательности ρ^F, относящихся к соответствующим предложениям. Это позволяет исключить из рассмотрения те предложения и те документы, которые содержат минимальное количество слов запроса. Вследствие этого повышается вычислительная эффективность алгоритма ранжирования, поскольку полный расчет всех оценок релевантности для всех предложений не производится.

На втором этапе процедуры ранжирования результатов поиска выполняется итоговая оценка соответствия найденных документов запросу пользователя. Она основывается на тех предварительных оценках релевантности предложений, которые превышают минимальный порог. За один просмотр множества предварительных оценок релевантности предложений на основе критериев 1-5 и формулы (12) из работы [2] вычисляются итоговые оценки соответствия предложений найденных текстов предложениям запроса. Итоговая оценка релевантности найденного документа вычисляется на основе формулы (13) из работы [2].

В ходе вычисления итоговых оценок релевантности найденных документов сведения о них совместно с сопутствующей служебной информацией, необходимой для построения поисковых сниппетов, заносятся в структуру данных типа куча. Куча организуется над массивом фиксированной длины; в ее вершине – ЭД, соответствующий документу с наименьшим рейтингом. Таким образом, в результате формируется множество, состоящее из K документов с наивысшим рейтингом, которое впоследствии сортируется за линейное время и подвергается процедуре построения текстовых аннотаций – поисковых сниппетов. На этом работа процедуры ранжирования результатов поиска завершается.

3. Экспериментальная оценка качества информационного поиска

3.1. Прототип поисково-аналитической системы

Представленные алгоритмы индексирования и поиска текстовой информации реализованы в ИАС Exactus Expert [3, 4, 7].

Коллекции документов в системе Exactus Expert формируются автоматизировано на основе научно-технических данных, свободно доступных в Интернете на русском и английском языках.

Для преобразования электронных документов во внутрисистемное представление применяются анализаторы собственной разработки, основанные на свободном программном обеспечении, поддерживающем конвертацию популярных форматов текстовых документов в текстовое и гипертекстовое представление. В системе учитывается имеющаяся гипертекстовая разметка документов, которая пополняется расширенными тегами метаданных (применяемыми, например, для выделения фрагментов документов, таких, как аннотации, списки литературы, фрагменты текстов, содержащие формулировки результатов работ, вводимые термины и определения и др.). На этапе индексации к тексту документа добавляются текстовые метаданные с соответствующей теговой разметкой: заголовок, сведения об авторах, дате публикации, источнике документа и др. [8, 9].

Это позволяет осуществлять полнотекстовый поиск по различным критериям, как в исходном тексте, так и в связанной с ним текстовой мета-информации.

Для проведения морфологического и синтаксического анализа применяются системы FreeLing [10-12] для английского языка и АОТ [13, 14] для русского языка. На основе результатов работы этих систем устанавливается семантическая информация с помощью лингвистического анализатора собственной разработки [3, 4].

Для проведения экспериментальной оценки качества поиска была развернута тестовая версия системы, включающая следующие коллекции:

– российские научные журналы из списка ВАК (около 28 тыс. документов);

– иностранные журналы (более 320 тыс. документов);

– авторефераты диссертаций на соискание ученой степени доктора наук (около 12 тыс. документов);

– зарубежные и российские патенты (около 400 тыс. документов).

3.2. Методика оценки качества информационного поиска

Экспериментальная оценка качества поиска проводилась с привлечением экспертов для оценки результатов поиска в соответствии со следующей методикой, являющейся модификацией метода общего котла [15, 16].

1. Было сформировано множество поисковых запросов, направленных на поиск научно-технической информации в предметной области экспертов. Запросы были подобраны таким образом, чтобы по оценкам эксперта в поисковой выдаче ИАС среди первых X результатов отсутствовали релевантные ($X=10$ – глубина пула результатов, подвергаемых экспертной оценке). Для этого эксперт просматривал первые X результатов поиска (включая полные тексты) и принимал решение о нерелевантности найденных результатов. При этом требовалось, чтобы выдача по этим запросам была непустая и содержала, как минимум, X результатов. Всего таким образом было подобрано $Y=14$ запросов. Наличие таких запросов объясняется искусственно созданной неполнотой тестовых коллекций.

2. Чтобы вычислить общепринятые метрики качества информационного поиска для каждого из отобранных запросов эксперты сформировали множество релевантных документов. Это было сделано на основе имеющихся у эксперта подборок документов в предметной области, содержание которых хорошо известно эксперту, а также с применением поисковых машин Интернета. Для каждого запроса было отобрано P релевантных документов. Всего были получены оценки для $XxY=140$ документов. При этом эксперт ранжировал документы в соответствии со следующей шкалой от 1 до 5: 1 – совсем не релевантен, 5 – абсолютно релевантен. Таким образом, были сформированы таблицы релевантности.

3. Релевантные документы были проиндексированы в соответствующие коллекции без каких-либо дополнительных корректировок и поправок.

4. Автоматически был выполнен поиск по Y отобранным запросам и рассчитаны показатели качества информационного поиска на основе сформированных таблиц релевантности.

Рассмотренная модификация метода общего котла была предложена в силу следующих причин.

1. Необходимо снизить объем работы эксперта по оценке документов. Поскольку использовалась пятизначная шкала оценки, то для объективного решения о релевантности того или иного документа эксперту потребовалось бы провести глубокий анализ содержания не знакомого ему документа, что потребовало бы длительного времени даже для десятка документов, т.к. научные статьи и авторефераты весьма объемны. Вместе с тем, решение о нерелевантности того или иного документа запросу возможно принять на основе беглого анализа содержания.

2. При оценке результатов информационного поиска, полученных одним алгоритмом ранжирования, методом общего котла невозможно оценить метрику полноты. Для этого эксперту требуется проанализировать всю выдачу поисковой машины, что требует больших временных затрат.

В ходе эксперимента вычислялись следующие метрики качества поиска:

– средняя точность (*Precision*) на глубине пула X [16];

– полнота (*Recall*; классическая формула) на глубине пула X (в предлагаемой методике совпадает со значением точности);

– 11-точечный график полноты/точности по методике TREC;

– normalized discounted cumulative gain (*nDCG)* [17];

– и обобщенная полнота (*GR*).

Формулы для расчета *nDCG* и *GR* приведены далее.

$$nDCG_{Q,X} = \frac{REL_{1,Q} + \sum_{i=2}^{X} \frac{REL_{i,Q}}{\ln(i)}}{iDCG_{Q,X}},$$

$$GR_{Q,X} = \frac{\sum_{i=1}^{X} REL_{i,Q}}{REL_Q},$$

где $nDCG_{Q,X}$ – normalized discounted cumulative gain для запроса Q, на глубине пула X; $iDCG_{Q,S}$ – *ideal DCG* – наибольшее возможное значение *DCG* для запроса Q на глубине пула X; $REL_{i,Q}$ – экспертно оцененная релевантность документа, находящегося в поисковой выдаче по запросу Q на позиции i; $GR_{Q,X}$ – обобщенная оценка полноты для запроса Q на глубине пула X, REL_Q – суммарная величина оценок релевантности документов в таблице релевантности для запроса Q.

При расчете средней точности и построении 11-точечного графика полноты/точности релевантными считались документы, имеющие экспертную оценку 2 и выше. Метрики рассчитывались для каждого запроса по отдельности, а затем применялась процедура макроусреднения [16].

3.3. Результаты экспериментальной оценки качества информационного поиска

В результате эксперимента были получены значения метрик качества информационного поиска, приведенные в таблице. 11-точечный график полноты/точности представлен на Рис.12.

Как видно из представленных данных, предложенный алгоритм позволяет формировать поисковую выдачу, которая в целом отражает представление эксперта о релевантных документах. На первых позициях в результатах поиска, как правило, находятся документы, имеющие высокие оценки релевантности, по мнению эксперта.

Параметры экспериментов и значения метрик качества информационного поиска

Количество документов в пуле для одного запроса, шт.	Количество тестовых запросов, шт.	Общее количество релевантных документов в таблице релевантности, шт.	Средняя точность и полнота на глубине пула, %	nDCG, %	GR, %
10	14	140	81.24	94.71	79.42

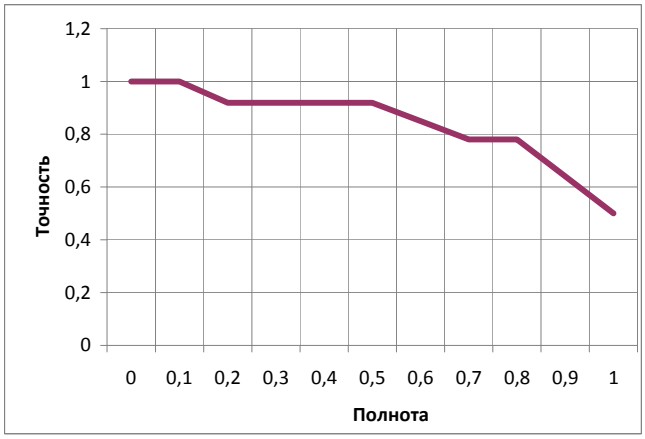

Рис. 12. 11-точечный график зависимости точности от полноты

Наряду с описанными экспериментами были выполнены измерения количественных параметров функционирования системы индексирования и поиска. Была создана масштабная коллекция, содержащая 958 тыс. документов (статьи русскоязычной web-энциклопедии Wikipedia по состоянию на апрель 2013 г.). В ходе нагрузочного тестирования один выделенный сервер-индексатор построил ИПИ этой коллекции за 6,5 часов. Общий объем текста на русском языке в коллекции составил более 8,5 Гб (в кодировке UTF-8). Средняя скорость индексирования составила около 23 Мб текста в минуту. При этом использована следующая конфигурация средств ИАС:

– количество запущенных потоков лингвистического анализа для построения ИТД – 17 (распределены между серверами ИАС: Intel Core i7 CPU 3.1 GHz, 16 Gb RAM, 4xHDD-500 Gb 7200rpm, RAID 1);

– количество модулей построения и агрегации ИПИ – 1;

– количество потоков агрегации ИПИ – 8.

Размер ИПИ (без учета вспомогательных структур) составил менее 7,2 Гб. Вспомогательная информация, необходимая для реализации поисковых и других функций ИАС, занимает около 400 Мб. Среднее время выполнения процедуры поиска в масштабной коллекции (без учета времени построения поисковых сниппетов) составляет менее 0,1 с. на один поисковый запрос (параллельные запросы отсутствуют). При нагрузке в 60 запросов, выполняемых параллельно, среднее время получения результата процедуры поиска возрастает до 0,5 с на один поисковый запрос.

Заключение

В статье представлены результаты исследования в области алгоритмов информационного поиска. Разработанные структуры данных и алгоритмы оценки релевантности и ранжирования результатов информационного поиска позволяют решать задачи оценки лингвистической релевантности и сопоставления текстов на основе лингвистических критериев с учетом лексико-морфологической, синтаксической и семантической информации. Они составляют важную часть алгоритмического и программного обеспечения ИАС Exactus Expert.

Проведенные экспериментальные исследования разработанных алгоритмов и структур данных показали их практическую применимость. Полученные оценки качества информационного поиска свидетельствуют о том, что результаты поиска в коллекции научно-технических документов в значительной степени отвечают информационным потребностям экспертов. Характеристики программной реализации также соответствуют современным требованиям к скорости работы и объемам занимаемой памяти информационно-поисковых и поисково-аналитических систем.

В заключение заметим, что представленное алгоритмическое и программное обеспечение составляет не только основу сервисов поиска информации по запросу, но также играет важную роль при решении следующих задач:

– выявления текстовых заимствований в коллекциях документов и оценке оригинальности содержания научных публикаций;

– сопоставлении публикаций по изложенным в них результатам;

– поиске библиографических ссылок и цитирований;

– идентификации введенной в публикациях терминологии.

Литература

1. Осипов Г.С., Смирнов И.В., Тихомиров И.А. Реляционно-ситуационный метод поиска и анализа текстов и его приложения // Искусственный интеллект и принятие решений. М.: ИСА РАН – №2, 2008. С. 3-10.
2. И.В. Соченков. Метод сравнения текстов для решения поисково-аналитических задач // Искусственный интеллект и принятие решений. М.: ИСА РАН, №2, 2013. С.95-106.
3. Смирнов И.В., Соченков И.В., Муравьев В.В., Тихомиров И. А. Результаты и перспективы поискового алгоритма Exactus. // Труды российского семинара по оценке методов информационного поиска РОМИП'2007-2008. Санкт-Петербург: НУ ЦСИ, 2008. - С. 66-76.
4. Osipov, G.; Smirnov, I.; Tikhomirov, I. and Shelmanov, A. Relational-Situational Method for Intelligent Search and Analysis of Scientific Publications. // In Proceedings of the Workshop on Integrating IR technologies for Professional Search Moscow, Russian Federation, March 24, 2013, p.57-64. [Электронный ресурс] URL: http://ceur-

ws.org/Vol-968/irps_10.pdf (дата обращения 23.03.2013).
5. И.В. Соченков, Р.Е. Суворов. Сервисы полнотекстового поиска в информационно-аналитической системе (Часть 1) // Информационные технологии и вычислительные системы. – М.:ИСА РАН. №2, 2013. С. 69–78.
6. D. E. Knuth, The Art of Computer Programming, Volume 3: Sorting and Searching, Addison-Wesley, (1973), 722 pages.
7. Тихомиров И.А., Смирнов И.В., Соченков И.В., Девяткин Д.А., Шелманов А.О., Зубарев Д.В., Швец А.В., Лешкин А.В., Суворов Р.Е. Exactus Expert: Поисково-аналитическая система поддержки научно-технической деятельности // Труды тринадцатой национальной конференции по искусственному интеллекту с международным участием КИИ-2012. Б.: БГТУ, 2012. т. 4. - С. 100-108.
8. Назаренко Г.И., Плотникова В.А., Смирнов И.В., Соченков И.В., Тихомиров И.А. Программные средства создания и наполнения полнотекстовых электронных библиотек // «Электронные библиотеки: перспективные методы и технологии, электронные коллекции: XII Всероссийская научная конференция RCDL' 2010, Казань, Россия - 2010. – С38-42.
9. Завьялова О.С., Киселев А.А., Осипов Г.С., Смирнов И.В., Тихомиров И.А., Соченков И.В. Система интеллектуального поиска и анализа информации Exactus на РОМИП-2010 // Труды российского семинара по оценке методов информационного поиска РОМИП'2010. - Казань: Казан. ун-т, 2010. С49-69.
10. Semantic services in freeling 2.1: Wordnet and ukb / Lluis Padro, Samuel Reese, Eneko Agirre, Aitor Soroa // Principles, Construction, and Application of Multilingual Wordnets / Ed. by Pushpak Bhattacharyya, Christiane Fellbaum, Piek Vossen ; Global Wordnet Conference 2010. — Mumbai, India : Narosa Publishing House, 2010. — February. — P. 99–105.
11. J. Atserias, B. Casas, E. Comelles, M. González, L. Padró, and M. Padró. Freeling 1.3: Syntactic and semantic services in an open-source nlp library. In Proc. LREC, 2006.
12. Carreras, X., I. Chao, L. Padró And M. Padró. FreeLing: An Open-Source Suite of Language Analyzers. // In M.T. Lino, M. F. Xavier, F. Ferreira, R. Costa, R. Silva, ed., Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04). Lisbon, Portugal. 2004.
13. А. Сокирко. Семантические словари в автоматической обработке текста (по материалам системы ДИАЛИНГ) // Дисс канд.т.н. [Электронный ресурс] URL: http://www.aot.ru/docs/sokirko/sokirko-candid-1.html, (дата обращения 23.03.2013).
14. Автоматическая обработка текста. 2013. // [Электронный ресурс] — URL: http://www.aot.ru/ (дата обращения 23.03.2013).
15. И. Некрестьянов, М. Некрестьянова, А. Нозик. К вопросу об эффективности метода "общего котла". [Электронный ресурс] – Режим доступа: http://rcdl.ru/doc/2005/sek9_1_paper.pdf, свободный. Проверено 14.01.2013.
16. М. Агеев, И. Кураленок, И.Некрестьянов. Официальные метрики РОМИП 2006 [Электронный ресурс] – Режим доступа: http://romip.ru/romip2006/appendix_a_metrics.pdf, свободный. Проверено 14.01.2013.
17. Kalervo Jarvelin, Jaana Kekalainen: Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems №20(4), 2002. P.p.422–446.

Соченков Илья Владимирович. Инженер-исследователь ИСА РАН, научный сотрудник ООО «Технологии системного анализа», ассистент кафедры информационных технологий Российского университета дружбы народов. Окончил Российский университет дружбы народов в 2009 году. Автор 30 научных работ. Область научных интересов: интеллектуальные методы поиска и анализа информации, обработка больших массивов данных, защита сетей, контентная фильтрация, компьютерная лингвистика. E-mail: sochenkov@isa.ru

Суворов Роман Евгеньевич. Аспирант ИСА РАН, инженер-исследователь ООО «Технологии системного анализа». Окончил Рыбинский государственный авиационный технический университет им. П.А. Соловьева в 2012 году. Автор 5 научных работ. Область научных интересов: интеллектуальный анализ текстовой информации, контентная фильтрация, интеллектуальные динамические системы. E-mail: rsuvorov@isa.ru