ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ НАУКИ
ИНСТИТУТ СИСТЕМНОГО АНАЛИЗА
РОССИЙСКОЙ АКАДЕМИИ НАУК

На правах рукописи

Соченков Илья Владимирович

РЕЛЯЦИОННО-СИТУАЦИОННЫЕ СТРУКТУРЫ ДАННЫХ, МЕТОДЫ И АЛГОРИТМЫ РЕШЕНИЯ ПОИСКОВО-АНАЛИТИЧЕСКИХ ЗАДАЧ

Специальность: 05.13.17 – Теоретические основы информатики

Диссертация на соискание ученой степени кандидата физико-математических наук

Научный руководитель –
доктор физико-математических наук,
профессор Г.С. Осипов

Москва, 2014

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Актуальность темы исследования. Развитие Интернета привело к росту объёмов доступной информации, которая может быть использована при решении важных задач в ходе научно-исследовательской и экспертной деятельности, для поддержки принятия решений в научно-технической, социальной и других сферах. Анализ этой информации и её использование при принятии стратегических решений даёт преимущество в развитии экономики, науки и технологий. Поисково-аналитическая обработка информации в условиях динамично растущего Интернета не может быть выполнена без автоматизированных информационных систем.

В ранних информационных системах поиск не был полнотекстовым и осуществлялся только по наиболее важным элементам метаданных документов, а аналитические функции сводились к ручной рубрикации и построению тематических каталогов. Такое решение было недостаточно эффективным и качественным и в целом не соответствовало потребностям пользователей, т.к. не позволяло решать более сложные задачи.

Для реализации сервисов текстового поиска в 1960-е годы исследователями были созданы алгоритмы булева поиска (boolean search) [1, 2]. Примерно в то же время были разработаны представление текстовой информации в виде векторов в пространстве ключевых слов, а также алгоритмы ранжирования результатов поиска с учётом статистических закономерностей распределения слов [3–6].

Развитие Интернета связано с ростом количества доступной информации и увеличением числа пользователей, заинтересованных в получении этой информации. Для поиска информации и навигации в Интернете были созданы поисковые машины: Yandex, Yahoo, Rambler, Google, Bing. Это привело к развитию технологий поиска информации в Интернете, использующих представление web-страниц в виде векторов признаков, статистические методы и методы машинного обучения. В качестве признаков, влияющих на результат

ранжирования результатов поиска, выступают как статистические величины, характеризующие частоты отдельных слов, так и другие факторы, учитывающие, например, ссылочную структуру Интернета [7, 8], а также географическое положение, историю запросов и поведение пользователя [9–13]. Результатом активных исследований в этой области стало эффективное решение задачи поиска по ключевым словам в Интернете.

Функции современных информационно-аналитических систем (ИАС) гораздо шире поиска по заданному набору слов. Современные ИАС включают в себя инструменты решения следующих задач:

- полнотекстовый поиск документов, релевантных информационной потребности пользователя, сформулированной в виде набора ключевых слов, осмысленной фразы, предложения или вопроса на естественном языке (ЕЯ) с дополнительными ограничениями на метаданные, которые могут задаваться как в текстовом, так и нетекстовом виде [1, 2, 14];
- реферирование результатов полнотекстового поиска, а также отдельных документов [15];
- поиск текстовых заимствований в коллекциях документов [16, 17];
- поиск содержательно и тематически близких документов, включая тематическую кластеризацию и классификацию [18, 19];
- извлечение из текстов ЕЯ структурированных данных и фактов, установление зависимостей и связей между ними, например, выявление отзывов о товарах и услугах и анализ тональности высказанных мнений [20–22].

В центре внимания пользователей ИАС находится именно текстовая информация и её содержание. При этом многие факторы, успешно используемые при ранжировании результатов поиска в поисковых машинах Интернета, не применимы в ИАС. Поэтому современные исследования в области информационно-аналитической обработки текстов направлены на развитие методов, основанных на анализе лингвистической информации [23 –

28]. При этом текст и составляющие его элементы характеризуется лексико-морфологическими синтаксическими и семантическими признаками.

В настоящее время созданы методы лингвистического анализа текстов и разработаны программные системы, позволяющие автоматически выполнять морфологический, синтаксический и семантический анализ предложений текста: AOT [29, 30], Solarix [31], NLTK [32, 33], FreeLing [34] и другие. Вычислительная эффективность этих систем и уровень качества лингвистического анализа позволяют применять их для обработки больших коллекций текстов. Однако в существующих ИАС, как правило, не применяются наукоёмкие методы лингвистического анализа текстов, поскольку в настоящее время отсутствуют эффективные методы хранения и обработки лингвистической информации, необходимой для решении поисково-аналитических задач, например, морфологических признаков, синтаксических связей, категориально-семантических значений (ролей) и семантических отношений на структурах текстов. Реляционные базы данных и стандартные инвертированные индексы не позволяют эффективно хранить и совместно использовать эту лингвистическую информацию, в то время как использование этой информации обеспечивает повышение качества решения задач аналитической обработки больших коллекций текстовых документов. В настоящей диссертационной работе предложен метод оценки сходства текстов с использованием лексико-морфологической, синтаксической и семантической информации, а также структуры данных и алгоритмы информационного поиска на основе этого метода, обладающие большей эффективностью и обеспечивающие более высокое качество результатов, нежели существующие методы решения поисково-аналитических задач в ИАС, что свидетельствует об **актуальности темы исследования**.

Предмет исследования – методы оценки сходства текстов с использованием лексико-морфологической, синтаксической и семантической информации; метод поиска информации на основе оценки сходства текстов, а также структуры данных и алгоритмы, реализующие указанные методы.

Целью исследования является повышение качества (полноты и точности) и вычислительной эффективности решения поисково-аналитических за счёт разработки и применения метода оценки сходства текстов, учитывающего лексико-морфологическую, синтаксическую и семантическую информацию, и создания структур данных и алгоритмов информационного поиска, реализующих этот метод.

Задачи исследования:

1. Разработка лексико-морфологических, синтаксических и семантических критериев оценки сходства текстов, а также метода оценки сходства текстов на основе этих критериев.
2. Разработка модели данных, предназначенной для исследования свойств структур данных и алгоритмов поисково-аналитической обработки текстовой информации.
3. Разработка и программная реализация структур данных, необходимых для решения задачи оценки сходства текстов и предназначенных для представления, хранения и обработки лексико-морфологической, синтаксической и семантической информации, являющейся результатом компьютерного лингвистического анализа текстов.
4. Разработка и программная реализация алгоритмов информационного поиска на основе многокритериальной оценки сходства текстов.
5. Экспериментальные исследования разработанных структур данных, алгоритмов формирования инвертированных индексов и многокритериальной оценки сходства текстов.

Для решения поставленных задач применены следующие **методы исследования**:

1. Логико-алгебраическое моделирование;
2. Методы теории множеств и алгебры логики.

3. Методы объектно-ориентированного проектирования программного обеспечения.
4. Общепринятые методы исследования качества результатов информационного поиска.

В ходе решения поставленных задач получены следующие **новые научные результаты:**

1. Разработаны лексико-морфологические, синтаксические и семантические критерии оценки сходства текстов, а также метод многокритериальной оценки сходства текстов.
2. Предложена и исследована модель данных, предназначенная для анализа свойств структур данных и алгоритмов поисково-аналитической обработки текстовой информации.
3. Разработаны структуры данных инвертированного поискового индекса, предназначенные для хранения и обработки лексико-морфологической, синтаксической и семантической информации, являющейся результатом компьютерного лингвистического анализа текстов документов. Эти структуры данных применяются для эффективного решения задач информационного поиска, в частности, многокритериальной оценки сходства текстов.
4. Разработаны и исследованы следующие алгоритмы поисково-аналитической обработки текстовой информации:
   - алгоритм построения инвертированного поискового индекса коллекций документов,
   - алгоритм поиска информации по запросу, основанный на разработанном методе многокритериальной оценки сходства текстов, реализующий следующие режимы поиска с учётом метаданных документов: поиск по ключевым словам, фразовый поиск, семантический и вопросно-ответный поиск.
5. Теоретически исследованы свойства разработанных структур данных и алгоритмов поисково-аналитической обработки текстовой

информации, в том числе получены оценки вычислительной сложности и доказаны утверждения, обосновывающие корректность указанных алгоритмов.

На защиту выносятся следующие положения диссертационной работы:

1. Новые синтаксические и семантические критерии оценки сходства текстов.
2. Новый метод многокритериальной оценки сходства текстов.
3. Новая модель данных, предназначенная для описания и исследования свойств структур данных и алгоритмов поисково-аналитической обработки текстовой информации.
4. Структуры данных предложенного в работе инвертированного поискового индекса, обеспечивающие эффективное хранение и обработку лексико-морфологической, синтаксической и семантической информации.
5. Алгоритмы поисково-аналитической обработки текстовой информации:
   - алгоритм построения инвертированного поискового индекса коллекций документов,
   - алгоритм поиска информации на основе метода многокритериальной оценки сходства текстов, реализующий поиск по ключевым словам, фразовый, семантический и вопросно-ответный поиск с учётом метаданных документов.
6. Программная реализация структур данных и алгоритмов, предназначенных для решения поисково-аналитических задач с помощью метода многокритериальной оценки сходства текстов.
7. Результаты экспериментального исследования разработанных структур данных и алгоритмов поисково-аналитической обработки текстовой информации, демонстрирующие, что эти алгоритмы обладают большей эффективностью и обеспечивают более

качественное решение задач информационного поиска в ИАС, нежели известные методы.

Теоретическая значимость. Разработанные метод многокритериальной оценки сходства текстов, представление текстовой информации, алгоритмы и структуры данных информационного поиска служат основой решения ряда поисково-аналитических задач. Методы семантического аннотирования текстов и поиска потенциально некорректных заимствований (рассмотрение которых выходит за рамки диссертационной работы) опираются на разработанный метод многокритериальной оценки сходства текстов. Алгоритмы, реализующие указанные методы, используют алгоритмы и структуры данных, предложенные и исследованные в настоящей работе.

Практическая значимость. Разработанный метод многокритериальной оценки сходства текстов, в основе которого лежит сопоставление лексико-морфологической, синтаксической, семантической информации, а также алгоритмы и структуры данных для решения поисково-аналитических задач нашли применение в ИАС. Программная реализация указанных алгоритмов и структур данных ориентирована на обработку больших коллекций текстовых документов в ИАС для информационной поддержки аналитической деятельности в научно-технической сфере.

Результаты исследований по теме диссертационной работы использованы при выполнении научно-исследовательских работ по следующим проектам Минобрнауки РФ, программам ОНИТ РАН и грантам РФФИ:

1. «Создание методов и программных средств выявления перспективных направлений научных исследований в России и за рубежом по данным из открытых источников на основе потребностей реального сектора экономики и обеспечения конкурентных позиций отечественных производителей на перспективных рынках инновационных товаров и услуг и созданных научно-технических заделов» (в рамках ФЦП «Научные и научно-педагогические кадры инновационной России», ГК № *16.740.11.0753*, 2011–2013 г.г.).

2. «Создание программного комплекса информационно-аналитической поддержки научно-технической деятельности на основе вычислительного семантического поиска и анализа неструктурированной текстовой информации» (в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007—2013 годы», ГК № *07.551.11.4003*, 2011–2013 г.г.).
3. «Разработка вычислительных методов объективной оценки качества научно-технических документов на естественных языках» (в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007—2013 годы», ГК № *14.514.11.4018*, 2011–2013 г.г.).
4. «Исследование и разработка методов и алгоритмов анализа связанности сложно-структурированных данных в научно-технической сфере» (в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007—2013 годы», ГК № *14.514.11.4024*, 2011–2013 г.г.).
5. «Исследование и разработка программного обеспечения понимания неструктурированной текстовой информации на русском и английском языках на базе создания методов компьютерного полного лингвистического анализа» (в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007—2013 годы», ГК № *07.514.11.4134*, 2011–2013 г.г.).
6. «Развитие методов анализа полуструктурированной информации и моделирования целенаправленного поведения» (в рамках ФЦП «Научные и научно-педагогические кадры инновационной России», ГК № *П952*, 2009–2011 г.г.).
7. Проект «Система высокоточного интеллектуального поиска, индексации и анализа информации для поддержки принятия решений» (проект *ПР4*

«Исследование и разработка параллельных алгоритмов анализа больших объемов текстовой информации из глобальной сети и алгоритмов принятия решений на основе когнитивных методов» программы «ТРИАДА», 2006–2007 г.г.).

8. «Развитие методов и программных средств многоязычного семантического поиска» (в рамках проекта *2.6* ОНИТ РАН 2009–2011 г.г.).
9. «Развитие методов и технологии семантического поиска и анализа научных публикаций Exactus Expert» (в рамках проекта *2.9* ОНИТ РАН 2012–2013 г.г.).
10. «Разработка и исследование структур данных и алгоритмов поисково-аналитической обработки текстовой информации» (в рамках проекта *14-07-31149\14 мол_а* РФФИ 2014–2015 г.г.)

Созданное программное обеспечение (ПО), включающее программную реализацию структур данных и алгоритмов поисково-аналитической обработки текстовой информации, внедрено в следующих системах:

- электронная библиотека международных клинических руководств в Медицинском центре Банка России;
- портал «Руконт» – национальный цифровой ресурс [35] в виде информационно-поисковых сервисов портала;
- информационно-аналитическая система Exactus Expert [36, 37] и поисковая машина Exactus [46, 49, 50].

Достоверность результатов подтверждена строгой математической формализацией основных положений диссертационного исследования и доказательствами теоретических утверждений, а также результатами экспериментальных исследований разработанных программных средств, реализующих предложенные методы, структуры данных и алгоритмы.

Апробация результатов исследования. Основные положения диссертации докладывались и обсуждались на следующих конференциях и семинарах:

- XIII национальная конференция по искусственному интеллекту с международным участием (КИИ: Россия, Белгород, Белгородский государственный технологический университет, 2012 г.);
- European Intelligence and Security Informatics Conference (IEEE EISIC: 2011 (Greece, Athens);
- Информационно-телекоммуникационные технологии и математическое моделирование высокотехнологичных систем – всероссийская конференция с международным участием в 2010, 2011 г.г. (Россия, Москва, Российский университет дружбы народов)
- XII Всероссийская научная конференция RCDL’ 2010: «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (Россия, Казань, 2010 г.)
- Российский семинар по оценке методов информационного поиска в 2008–2010 г.г.;
- Семнадцатая международная Конференция "Крым 2010" (Россия, Геленджик, 2010 г.).

Информационные системы, содержащие в своём составе программную реализацию разработанных структур данных и алгоритмов поисково-аналитической обработки текстовой информации, представлены на российских и международных выставках программного обеспечения и информационных технологий SofTool (в 2010–2013 г.г.) и CeBIT (в 2008–2014 г.г.).

Публикации. Всего по теме исследования опубликовано 14 работ: 6 из них в рецензируемых журналах из списка ВАК РФ [38–43], 8 в материалах российских и международных конференций [37, 44 – 50]. Получен патент Российской Федерации на изобретение [51] и 3 свидетельства о государственной регистрации программ для ЭВМ [52 – 54]. Опубликованные в этих работах результаты, относящиеся к методам оценки сходства текстов, представлению текстовой информации, а также структурам и алгоритмам информационного поиска, получены лично автором.

Структура и объем работы. Диссертация состоит из введения, четырёх глав, заключения, списка сокращений и условных обозначений, а также списка использованной литературы. Диссертация содержит 148 страниц, 25 рисунков, 2 таблицы, 152 источника в списке используемой литературы.

В первой главе приведен обзор методов компьютерного анализа и информационного поиска текстовой информации, относящихся к теме исследования. Обзор включает в себя рассмотрение различных моделей, предназначенных для представления и анализа текстовой ЕЯ информации в системах информационного поиска. В первом параграфе рассматриваются методы компьютерного анализа текстов, нашедшие применение в информационных системах. Второй параграф посвящён рассмотрению структур данных для представления текстовой информации в задачах информационного поиска, а также изучению методов ранжирования результатов информационного поиска. В заключительной части приведены основные выводы, обосновывающие актуальность исследования, поставлена цель исследования и сформулированы задачи исследования.

Вторая глава посвящена методу многокритериальной оценки сходства текстов на основе лексико-морфологической, синтаксической и семантической информации. В первом параграфе представлено разработанное представление текстовой информации для решения задачи многокритериальной оценки сходства текстов. В следующем параграфе введены критерии оценки сходства текстов и описан метод многокритериальной оценки сходства текстов. Изучены свойства введённых критериев, введена общая оценка сходства текстов и доказаны утверждения о её свойствах, важные для практического применения метода оценки сходства текстов. В заключительном параграфе второй главы изучены вопросы применения разработанного метода оценки сходства текстов для решения задач полнотекстового поиска по ключевым словам и словосочетаниям, а также фразового, семантического и вопросно-ответного поиска.

Третья глава посвящена разработке и исследованию модели данных, структурам данных и алгоритмам для решения поисково-аналитических задач. В первых параграфах главы представлены разработанные структуры данных, необходимые для представления информации о текстовых документах в ИАС, структуры инвертированных индексов, а также алгоритмы их построения на этапе индексирования документов. Предложенные структуры данных и алгоритмы описаны и формализованы с помощью методов логико-алгебраического моделирования. Созданная модель данных служит формальным инструментом для исследования свойств разработанных алгоритмов и структур данных. В заключительных параграфах третьей главы представлены разработанные структуры данных, необходимые для представления поисковых запросов и алгоритмы оценки релевантности и ранжирования результатов информационного поиска, реализующие метод многокритериальной оценки сходства текстов. Для всех представленных алгоритмов получены оценки вычислительной сложности и определены границы применимости (доказано соответствующее утверждение).

Четвёртая глава посвящена вопросам реализации и экспериментального исследования метода оценки сходства текстов, структур данных и алгоритмов информационного поиска. В первых параграфах описана программная реализация метода оценки сходства текстов, алгоритмов и структур данных для решения поисково-аналитических задач. Во втором параграфе главы проведено экспериментальное исследование метода оценки сходства текстов, структур данных и алгоритмов информационного поиска. Предложена методика оценки разработанного метода оценки сходства текстов и его реализации в алгоритмах и структурах данных, получены оценки качества информационного поиска. Приведены результаты оценки разработанных программных средств по объёму дисковой памяти, необходимой для хранения индексных структур, показавшие их эффективность в сравнении с известными структурами данных для решения задачи поиска информации.

В заключении приведены основные результаты и выводы диссертационной работы, а также рассмотрены направления дальнейших исследований.

ГЛАВА 1. МЕТОДЫ КОМПЬЮТЕРНОГО АНАЛИЗА И ИНФОРМАЦИОННОГО ПОИСКА ТЕКСТОВОЙ ИНФОРМАЦИИ

Настоящая глава посвящена изучению моделей, предназначенных для представления и анализа текстовой ЕЯ информации в системах информационного поиска. В первом параграфе рассматриваются методы компьютерного анализа текстов, применяемые в информационных системах. Второй параграф посвящён рассмотрению структур данных для представления текстовой информации в системах информационного поиска. В заключительных параграфах представлены выводы, сделанные по итогам выполненного обзора, а также сформулирована цель и поставлены задачи исследования.

1.1 Обзор методов компьютерного анализа текстов для решения задач информационного поиска

Идея автоматизированной обработки текстовой информации с помощью электронно-вычислительных машин возникла в начале XX века. Развитие компьютерной лингвистики способствовало интеграции методов математики (прежде всего, статистики и дискретной математики) и лингвистики для решения прикладных задач анализа текстовой информации. В простых задачах (распознавание языка документа, составление частотного словаря) в компьютерных программах не требуется использовать сложные модели, отражающие содержание, смысл текста. Для решения более сложных задач (компьютерный перевод, автоматическое реферирование и другие задачи аналитической обработки текстовой информации) требуется использовать ряд методов лингвистического анализа текстов, которые позволяют выявлять понятия, ключевую лексику и определять различные связи между ними.

Изначально модели текстовой информации (*statistical language model*) [55] основывались на анализе статистических закономерностей в

распределениях слов в текстах, а сами слова рассматривались как цепочки символов. С ростом вычислительных возможностей и объёмов памяти ЭВМ развивались и применялись методы компьютерного анализа текстовой информации, рассматривающие текст как сложную систему с внутренней многоуровневой структурой.

Современный подход к решению задачи компьютерного анализа текста предполагает, что компьютерный лингвистический анализ не является монолитной одноэтапной процедурой. Процедура компьютерного анализа текста включает несколько уровней обработки. Результаты анализа текста обработчиком некоторого уровня передаются на вход обработчику следующего уровня. Таким образом, обработчики образуют цепочку, каждое звено которой отвечает за определённый этап обработки текстовой информации. Состав обработчиков лингвистического процессора определяется задачами, для которых он предназначен. В области информационного поиска и задач аналитической обработки текстовой информации интерес представляют следующие уровни обработки текста:

- Токенизация (*tokenization*) – разбиение текста на слова и разделители и выделение границ предложений (*splitting*).
- Морфологический анализ (*part-of-speech tagging, POS tagging*) – лемматизация (определение леммы[1] по словоформе некоторой лексемы[2]) и установление морфологических характеристик словоформы, включая разрешение омонимии.
- Поверхностный синтаксический анализ (*shallow parsing*), заключающийся в выделении синтаксических групп (например, именных или глагольных), зачастую без установления связей внутри этих групп (*chunking*).
- Синтаксический анализ (*parsing*) – построение полных синтаксических графов предложений (в том числе сложных).

[1] Лемма – каноническая (словарная) форма слова
[2] Лексема – совокупность парадигматических форм одного слова, как элемента лексики естественного языка. Одной лексеме, как правило, соответствует одна статья в словаре.

– Семантический анализ.

Разбиение текста на слова и предложения является первой задачей. Она обычно решается с помощью регулярных выражений и конечных автоматов с применением эвристик [56 – 58].

Морфологический анализ включает в себя следующие этапы:

1. Определение множества возможных морфологических интерпретаций словоупотребления в тексте (леммы и морфологических характеристик словоформы);
2. Разрешение омонимии – выделение из множества возможных морфологических интерпретаций такой интерпретации, которая соответствует контексту словоупотребления в тексте.

Для морфологического анализа широко используется подход, при котором словоформы языка помещаются в специальный словарь, где вместе со словоформой хранится ее лемма, морфологические характеристики, а также другая информация (например, частота встречаемости в тексте). Анализ сводится к поиску заданной словоформы в этом словаре. Этот подход реализован в системах компьютерного анализа текстов АОТ [29] и Freeling [34]. Словарь лексем не может охватить все многообразие лексики ЕЯ. Поэтому для словоформ (и соответствующих лексем), отсутствующих в словаре, используют методы, которые позволяют предсказать лемму и морфологические характеристики по аффиксу, например, анализатор TnT [59]. В некоторых случаях используют только бессловарный подход – стемминг (stemming), при котором предсказание признаков осуществляется с помощью таблиц аффиксов, а нормализация – путём усечения аффиксов до возможной основы слова [60, 61].

Для разрешения омонимии, возникающей в результате морфологического анализа, применяют скрытые Марковские модели [63, 64] и системы правил (формируемых, например, с помощью машинного обучения), которые используют контекст словоупотреблений для определения морфологических характеристик слов [65].

Существуют реализации вышеописанных методов компьютерного анализа текстов, которые работают с высокой полнотой и точностью (95–99% в зависимости от тестовой коллекции) и находят применение в поисковых машинах, системах обнаружения заимствований, корпусных менеджерах и других приложениях.

Более сложным и ресурсоёмким (в силу неоднозначности ЕЯ) этапом обработки текста является синтаксический анализ. В зависимости от задач в лингвистических процессорах используют поверхностный (*shallow*) или глубокий (*deep*) синтаксический анализ (parsing) [66]. Под поверхностным анализом понимают выделение простых, рекурсивно невложенных синтаксических групп, этот подход в зарубежной литературе имеет термин «*chunking*». Эта наиболее простая задача синтаксического анализа.

В общем случае задачей синтаксического анализа является эксплицитное описание синтаксической структуры предложений текста. Большинство представлений синтаксической структуры предложений опираются либо на грамматику зависимостей [67], либо на грамматику непосредственно-составляющих [68], представляющих предложения текста в виде деревьев (соответственно, связанных отношениями подчинения словоупотреблений или составляющих). Существуют модели синтаксической структуры, не относящиеся ни к одному из перечисленных выше типов. Примером такой модели является грамматика связей [69], которая применяется для анализа текстов на английском [70] и русском языках [71, 72]. В рамках этой модели текст представляется не в виде дерева, а в виде сети, в которой отсутствует корневой элемент.

Для поверхностного анализа, в котором предполагается построение связного синтаксического дерева, широко применяются как грамматики, так и статистические методы, среди них наиболее известны методы, основанные на построении стохастической контекстно-свободной грамматики [73].

Глубокий синтаксический анализ основывается на применении систем правил, с привлечением экстралингвистической знаний: семантических

словарей, тезаурусов, таксономий [74]. К системам глубокого синтаксического анализа русского языка относятся ЭТАП-3 [73], Abby Compreno [75]. Для английского и некоторых других европейских языков методы глубокого синтаксического анализа реализованы в системе Xerox XLE [76], проекте RASP [77] и ENJU [78].

Современные методы синтаксического анализа показывают весьма высокую точность и полноту установления синтаксических связей: в среднем от 75% до 90% в зависимости от языка [66].

Задача семантического анализа является наиболее трудоёмкой в компьютерной реализации. Существуют различные трактовки понятия «семантика»[79]. Как правило, под семантическим анализом понимают установление семантических значений («ролей») слов и понятий в текстах (semantic role labeling, или SLR) и разрешение кореференции. Именно в таком контексте будет пониматься семантический анализ текста в настоящей работе.

Идеи о значении высказываний в языке и способах его выражения были впервые систематизированы в работах Ч. Филлмора [80, 81] и базировались на теории структурного синтаксиса, сформулированной Л. Теньером [82]. Филлмором было введено понятие семантического падежа, который выражает роль – семантическое содержание аргумента в предложении при предикате: «кто сделал нечто», «с кем нечто случились», «что подверглось некоему изменению» [81]. Эти идеи положили начало направлению исследований, посвященных установлению семантических значений слов и словосочетаний. Результатом этих исследований стали ресурсы Frame Net [84, 85] (для английского языка) и PropBank (мультиязычный) [86] –, а также теория коммуникативной грамматики Золотовой [87] и её расширение в виде реляционно-ситуационной модели (РСМ) для решения задач компьютерного семантического анализа флективных языков (в частности, русского) [88 – 90]. Эти исследования развивают подход, предложенный Ч. Филлмором, и используют различные множества семантических ролей (значений). В основе этих подходов к формализации семантики значений лежит понятие действия

(ситуации), описываемого простым предложением (клаузой в составе предложения). Конкретное действие выражается некоторым предикатным словом (или группой синонимичных в некотором контексте предикатных слов), а члены предложения выполняют роль аргументов, вовлечённых в действие (ситуацию).

Для английского языка наибольшее распространение получили методы семантического анализа, основанные на машинном обучении с учителем [83]. Эти методы требуют репрезентативных масштабных лингвистически размеченных корпусов для проведения обучения (таких, как корпуса, созданные в проектах Frame Net и PropBank). Создание подобных корпусов является трудоёмкой задачей, требующей привлечения специалистов-лингвистов для разметки текста. Для многих языков, в том числе русского, подобные семантически аннотированные корпуса текстов в настоящее время отсутствуют. Поэтому ряд исследований направлен на создание методов семантического анализа текстов без учителя или с частичным привлечением учителя [66].

Для реализации семантического анализа с использованием РСМ на практике используется словарь предикатных слов [91, 92] и система автоматически формируемых правил [93]. В основе теории коммуникативной грамматики и основанной на ней РСМ лежит понятие синтаксемы – минимальной синтаксической единицы. Синтаксемам приписываются семантические значения, а сами синтаксемы связываются с другими синтаксемами семантическими отношениями [94].

Следующей задачей, относящейся к семантическому анализу текста, является разрешение кореференции. Методы решения этой задачи также по большей части опираются на машинное обучение по размеченным корпусам. При этом оптимизируется функция, которая определяет оценку уверенности в том, что две сущности в тексте кореферентны [95, 96]. В качестве признаков кореферентности рассматривают согласование по числу и роду, расстояние между сущностями, близость по предложениям, принадлежность к некоторому

классу (персон, компаний, стран и т.п.) синтаксические характеристики, семантические отношения между сущностями и др. Кроме того, применяют статистику совместной встречаемости слов – анализ колокаций в пределах предложений: при разрешении местоименной анафоры для выбора антецедента используют словарь сочетаемости глаголов с потенциальными антецедентами [97]. Широко применяются также различные эвристики.

Качество работы методов семантического анализа зависит от множества факторов: от выбранной модели семантики, количества выбранных семантических значений, обучающих и тестовых корпусов, а также от качества работы методов синтаксического и морфологического анализа, к результатам которых применяется семантический анализатор. При этом точность и полнота сильно варьируются в диапазонах от 50% до 95% [97–101].

Совокупный результат работы методов компьютерного анализа текстовой информации можно представить в виде семантической сети. Вершинами сети являются словоупотребления, имеющие ряд атрибутов (лемма, морфологические признаки, семантические значения), связанные синтаксически и семантически. В зависимости от конкретного приложения эти сети могут сравниваться друг с другом на основе совпадения по различным критериям. Одним из таких приложений является, например, вопросно-ответный семантический поиск информации [23]. Уровень качества лингвистического анализа текста с применением существующих методов вполне достаточен для использования этих методов в промышленных ИАС, обрабатывающих большие объёмы текстовой информации. В следующем параграфе настоящей главы рассматриваются вопросы организации данных в поисковых и информационно-аналитических системах и исследуются возможности хранения и использования лингвистической информации текстов при реализации функций указанных систем.

1.2 Обзор индексных структур данных и методов ранжирования результатов поиска в информационно-аналитических и поисковых системах

Для реализации сервисов текстового поиска в 1960-е годы исследователями были разработаны алгоритмы булева поиска (Boolean search) [1, 2], появление которых связано с развитием теории реляционных баз данных. Примерно в это же время были разработаны представление текстовой информации в виде векторов в пространстве ключевых слов, а также алгоритмы ранжирования результатов поиска с учётом статистических закономерностей распределения слов [1–6, 102]. Тексты документа и запроса представляются в виде векторов признаков. Пространство признаков составляют нормализованные (приведённые к словарной форме) слова ЕЯ, а значениями признаков являются различные числовые величины, характеризующие эти слова с точки зрения их встречаемости как в отдельном тексте, так и в коллекции текстов в целом.

В простейшем случае (для реализации булева поиска) в качестве значений признаков выступают бинарные величины (1– встречается в тексте, 0 – не встречается). Для построения более сложных методов ранжирования в качестве значений признаков используются величины TF-IDF и их модификации [1,2, 102–104]. В качестве оценки сходства текста запроса и текста документа выступают различные метрики в пространстве векторов признаков: косинусное расстояние, расстояние Евклида, манхэттенское расстояние и др. [1,2].

При таком подходе не учитывается контекст словоупотреблений, их совместная встречаемость, а значит, запрос и найденный текст могут совпадать по используемой лексике, но при этом их содержание и смысл могут принципиально отличаться. Кроме того, фразовый, вопросно-ответный и семантический поиск не могут быть реализованы с использованием описанных способов представления текстовой информации. Поэтому векторные

представления текстовой информации были существенно расширены для реализации функций поиска по ключевым словам, фразового, вопросно-ответного и других типов поиска.

В современных информационно-поисковых системах при поиске по ключевым словам дополнительно учитывается взаимное расположение словоупотреблений в найденных текстах, расстояние между ними. Ранние работы в области методов вопросно-ответного поиска и методов построения поисковых сниппетов также базировались на этих признаках [105]. Однако впоследствии было замечено, что точный вопросно-ответный поиск не может основываться лишь на сопоставлении запроса и искомых ответов только по лексике [106–108]. Поэтому ряд более поздних работ был направлен на изучение методов оценки сходства текстов путём сопоставления синтаксических и семантических структур предложений [23, 108, 109].

Несмотря на это, векторное представление текстов было положено в основу методов информационного поиска в Интернете. Их дальнейшее развитие привело к созданию и применению в поисковых машинах Интернета методов ранжирования результатов поиска, которые опираются на «внетекстовые» характеристики информации. К числу таких методов относятся ссылочное ранжирование (PageRank) [7, 8], ранжирование на основе предпочтений пользователей с использованием истории поиска и посещения web-страниц [8, 9], анализ поведения пользователей [10], ранжирование с учётом географического положения пользователей [11–13], ранжирование с учётом типов информационных потребностей пользователей [1, 2, 9]. В результате дальнейшего развития методов интернет-поиска количество значимых факторов ранжирования, непосредственно не связанных с текстовым содержанием документов, превысило десятки и сотни [11]. Однако при решении задач, возникающих в ИАС, факторы, нашедшие применение в области классического поиска в Интернете, оказываются незначимыми. В силу этого в ИАС именно тексты документов становятся основным источником информации.

Классические методы информационного текстового поиска (основанные на статистических законах TF-IDF [110], мультиномиальной модели [111], модели правдоподобия [112]) опираются на предположения о независимом распределении словоупотреблений в текстах (униграммная модель) и относительно просто реализуются в информационных системах. Эти методы характеризуются приемлемым качеством поиска по ключевым словам [1, 2], что в сочетании с другими факторами ранжирования результатов позволяет удовлетворить информационные потребности значительной части аудитории поисковых машин Интернета [113]. Однако реализация функций информационного поиска в ИАС, основанная на этих методах, не позволяет решать задачи, стоящих перед пользователями ИАС с необходимым качеством. Причиной этого является упрощённое представление текста набором слов, и в результате теряется важная информация (синтаксическая, семантическая), которая может быть использована для более полного и точного решения поисково-аналитических задач.

В противоположность классическим подходам методы решения поисково-аналитических задач, основанные на компьютерном лингвистическом анализе текстов, как правило, обладают высокой трудоёмкостью реализации и большей вычислительной сложностью. Поэтому зачастую лингвистические методы противопоставляются классическим и рассматриваются отдельно и независимо от них. Однако в работах [1, 2, 36] отмечено, что повышение качества решения поисково-аналитических задач возможно путём интеграции статистических и лингвистических методов анализа текстов и их адаптации к промышленному применению в крупных ИАС. Кроме того, как было отмечено выше, задачи фразового, вопросно-ответного, семантического поиска и связанные задачи аналитической обработки текстов (семантическое реферирование, поиск некорректных заимствований) не могут быть решены исключительно на основе оценки лексического сходства текстов без учёта других факторов.

От выбранного представления текстовой информации и методов информационного поиска (лингвистического анализа текстов, оценки сходства текстов, ранжирования результатов поиска) зависят структуры данных для хранения результатов анализа текстовой информации. Поиск и, в общем случае, любая выборка релевантной информации в больших коллекциях текстов не могут быть эффективно реализованы прямым сопоставлением текста запроса и текста каждого документа. Поэтому для эффективного поиска информации в коллекциях документов применяется инвертированный поисковый индекс (ИПИ) [114]. ИПИ позволяет для каждого словоупотребления запроса по нормальной форме лексемы, соответствующей этому словоупотреблению, получить список употреблений лексемы в текстах документов коллекции. ИПИ предоставляет также лингвистическую информацию, сопутствующую найденным словоупотреблениям. ИПИ обычно реализуется в виде хеш-таблицы [114]. В качестве ключа ИПИ выступает нормальная (словарная) форма слова (или её числовой идентификатор). В классической реализации [115] каждому ключу соответствует список однотипных элементов данных (ЭД) – пост-лист. Все ЭД в пост-листах имеют одинаковый размер и хранят информацию о словоупотреблениях в текстах документов.

В работах [116, 117] предлагается выделять в пост-листах последовательности ЭД, представляющие информацию о словоупотреблениях, их позициях и частотах в текстах документов. В этих работах рассматриваются вопросы построения дополнительных ИПИ для реализации функций фразового поиска. В книге [118, с. 96] отмечается, что для реализации различных типов поиска формируют ИПИ различных типов – для поиска по ключевым словам, концептного и фразового [119] поиска, а также для поиска по метаинформации. Зачастую это приводит к увеличению объёмов занимаемой памяти и снижению быстродействия – см., например, работу [120].

Отметим, что существующие технологии формирования ИПИ, ориентированные на применение в промышленных поисковых и информационно-аналитических системах (например, Indri Lemur [121])

накладывают существенные ограничения на объёмы памяти, занимаемой индексными структурами. Дополнительными требованиями являются инкрементальное построение ИПИ (возможность добавления новых документов без перестроения базы данных ИПИ целиком), высокая скорость индексирования добавляемого документа. Процесс построения базы данных ИПИ должен выполняться параллельно с обработкой поисковых запросов [122, 123].

Аналогичным образом устроены ИПИ в других поисковых машинах (Xapian [124], Sphinx [125], The Apache Lucene [126]). Они ориентированы на хранение базовой информации, характеризующей словоупотребления текста (словоформы, позиции слов, разбиения на предложения, теги метаданных), но не поддерживают эффективного хранения синтаксической и семантической информации. С точки зрения алгоритмического обеспечения эти системы ориентированы лишь на полнотекстовый поиск по ключевым словам (с поддержкой фразового поиска по точному соответствию текстов запроса и документа), но не поддерживают других режимов поиска и не реализуют связанных с ними функций ИАС.

1.3 Выводы

Решение большинства поисково-аналитических задач в современных ИАС невозможно без применения методов информационного поиска, в которых совместно учитываются такие факторы, как частоты словоупотреблений (лексем), совместная встречаемость словоупотреблений (колокации), а также лингвистические признаки, значения которых определяются с помощью методов компьютерного морфологического, синтаксического и семантического анализа.

В настоящее время создан ряд методов лингвистического анализа текстовой информации, позволяющих выполнять автоматический морфологический, синтаксический и семантический анализ. При этом разработаны программные средства (многие из которых распространяются по

лицензии свободного программного обеспечения) для автоматического анализа текстов ЕЯ.

Несмотря на это, в промышленных ИАС лингвистическая информация текстов (лексическая, морфологическая, синтаксическая и семантическая) не используется в комплексе. В известных приложениях либо применяют простейшие представления текстовой информации в виде векторов лексики со статистическими характеристиками, либо решают частные задачи (например, вопросно-ответного или фразового поиска) с использованием собственных моделей текста (например, учитывающих только синтаксические структуры или только семантические значения).

В ИАС запросы пользователей ориентированы на различные типы поиска информации: наряду с поиском по ключевым словам может использоваться фразовый поиск, семантический и вопросно-ответный поиск. Т.е. в ИАС должно обеспечиваться качественное решение всех перечисленных задач. Эта цель может быть достигнута использованием лингвистической информации при решении поисково-аналитических задач. Однако в настоящее время отсутствует представление текстовой информации для решения, которое объединяло бы результаты лексико-морфологического, синтаксического и семантического анализа и позволяло решать поисково-аналитические задачи на основе оценки сходства текстов с использованием этих результатов.

В силу вышесказанного, перспективным является исследование методов решения поисково-аналитических задач на основе многокритериальной оценки сходства текстов. В зависимости от потребности пользователя и типа поиска, текст запроса и тексты документов коллекции могут сопоставляться по различным критериям. При этом вклад каждого из критериев в функцию ранжирования результатов поиска также может различаться в зависимости от типа поиска.

До настоящего времени не созданы эффективные методы организации индексных структур данных для решения задач информационного поиска в масштабных коллекциях текстов с использованием лингвистической

информации. Существующие структуры данных ИПИ не позволяют эффективно хранить и использовать при поиске лексико-морфологическую, синтаксическую и семантическую информацию текстов, полученную в результате компьютерного лингвистического анализа. Известные универсальные решения, например, SQL базы данных, не способны поддерживать объёмы информации, с которыми должны работать промышленные ИАС: при сохранении результатов анализа 1 млн документов в одной базе данных (на отдельном сервере ИАС), количество ЭД с лингвистической информацией, связанной со словоупотреблениями, составляет до десятков миллиардов.

Вышесказанное свидетельствует об актуальности настоящего диссертационного исследования. На основе вышеизложенного сформулируем цель и поставим задачи диссертационного исследования.

1.4 Цель и задачи исследования

Целью исследования является повышение качества и эффективности решения поисково-аналитических задач за счёт разработки метода сопоставления текстов с учётом лексико-морфологической, синтаксической и семантической информации, а также реализации этого метода в структурах данных и алгоритмах информационного поиска.

Для достижения цели исследования должны быть решены следующие задачи:

1. Разработка лексико-морфологических, синтаксических и семантических критериев оценки сходства текстов, а также метода оценки сходства текстов на основе перечисленных критериев.
2. Разработка модели данных, предназначенной для формализации и исследования свойств структур данных и алгоритмов поисково-аналитической обработки текстовой информации.
3. Разработка и программная реализация структур данных, применяемых для решения задачи оценки сходства текстов и

предназначенных для представления, хранения и обработки лексико-морфологической, синтаксической и семантической информации, получаемой в результате компьютерного лингвистического анализа текстов.

4. Разработка и программная реализация алгоритмов формирования инвертированных индексов и многокритериальной оценки сходства текстов, предназначенных для решения поисково-аналитических задач.
5. Проведение экспериментальных исследований разработанных структур данных, алгоритмов формирования инвертированных индексов и многокритериальной оценки сходства текстов.

Структуры данных, применяемые для решения задачи оценки сходства текстов, должны быть эффективны с точки зрения объёмов занимаемой памяти. Алгоритмы их построения должны обеспечивать инкрементальное пополнение ИПИ и обладать вычислительной сложностью, близкой к линейной (по количеству обрабатываемых ЭД). Алгоритмы выборки информации из ИПИ и алгоритмы, реализующие метод оценки сходства текстов, также должны обладать высоким быстродействием (а значит, и вычислительной сложностью, близкой к линейной по количеству обрабатываемых ЭД).

Показатели качества информационного поиска с применением разработанных алгоритмов оценки сходства текстов должно быть выше показателей качества известных алгоритмов.

Успешное решение вышеперечисленных задач свидетельствует о достижении цели исследования.

ГЛАВА 2. МЕТОД МНОГОКРИТЕРИАЛЬНОЙ ОЦЕНКИ СХОДСТВА ТЕКСТОВ НА ОСНОВЕ ЛЕКСИКО-МОРФОЛОГИЧЕСКОЙ, СИНТАКСИЧЕСКОЙ И СЕМАНТИЧЕСКОЙ ИНФОРМАЦИИ

2.1 Представление текстовой информации в задаче многокритериальной оценки сходства текстов

Формализацию метода оценки сходства текстов начнём с описания представления текстовой информации, на котором базируется предложенный метод.

Пусть задано некоторое универсальное множество D лемм – словарь нормальных форм всех лексем ЕЯ. Пусть также Φ – множество всевозможных форм (словоформ) всех лексем ЕЯ.

Рассмотрим множество текстов $E = \{\tau\}$. и произвольный текст $\tau \in E$. Он содержит конечное множество словоупотреблений $W^\tau = \{w_i\}$. Определим:

- Бинарное отношение δ^τ на множестве словоупотреблений W^τ текста τ и множестве D нормальных форм лексем: $\delta^\tau \subseteq W^\tau \times D$, сопоставляющее каждому словоупотреблению его нормальную форму: $\forall w \in W^\tau \exists d \in D :< w, d > \in \delta^\tau$. При этом допустимо, что у каких-либо словоупотреблений имеется более одного варианта нормализации (для некоторых $w \in W^\tau \quad \exists d' \in D, d'' \in D\, d' \neq d'' \& < w, d' > \in \delta^\tau \& < w, d'' > \in \delta^\tau$), т.е. δ^τ не является функциональным в общем случае.
- Бинарное отношение ψ^τ на множестве словоупотреблений W^τ текста τ и множестве Φ всевозможных форм различных лексем: $\psi^\tau \subseteq W^\tau \times \Phi$. Это отношение определяет форму каждого словоупотребления в тексте. Каждому словоупотреблению соответствует единственная форма: $\forall w \in W^\tau \exists g \in \Phi :< w, g > \in \psi \quad \& \left(\exists g' \in \Phi :< w, g' > \in \psi \rightarrow g = g'\right)$, т.е. отношение ψ^τ функционально по определению.

– Конечное множество меток (тегов) $T=\{t_i\}$, и бинарное отношение $\theta^\tau \subseteq W^\tau \times T$, сопоставляющее каждому словоупотреблению некоторую метку (тег гипертекстовой разметки или разметки по метаданным, например, для словоупотреблений в заголовке текста, в списке авторов и т.п.). Каждому словоупотреблению соответствует единственный тег: $\forall w \in W^\tau \exists t \in T :< w,t >\in \theta^\tau \ \&\left(\exists t' \in T :< w,t' >\in \theta^\tau \to t = t'\right)$, т.е. отношение θ^τ функционально по определению.
– Числовую функцию, задающую вес («значимость») меток: $\omega(t)$.
– Числовую функцию, задающую вес словоупотребления в тексте: $v(w^\tau)$.

Введём разбиение множества словоупотреблений на предложения (возможно, сложные), заданное в виде множества $S^\tau = \left\{s_i\right\}$, где $\mathrm{s}_k = \left\{w_{i_k},...,w_{j_k} \mid \forall\, w_x \in W^\tau\right\}$, где $i_k,...,j_k$ – последовательность индексов словоупотреблений в тексте, входящих в k-е предложение, а j_k-i_k – длина k-го предложения. Т.е. $S^\tau \subset 2^{W^\tau}$ (где 2^{W^τ} – булеан множества W^τ). Условимся считать, что для $\forall m \neq n, s_m \in S^\tau, s_n \in S^\tau$ имеет место $s_m \cap s_n = \varnothing$ и $\forall w \exists k \exists l : s_k \in S^\tau, s_l \in S^\tau$ выполнено $\left(w \in s_k \ \&\ w \in s_l\right) \to \left(k = l\right)$). Таким образом, S^τ индуцирует разбиение на множестве словоупотреблений.

Для учёта синтаксических зависимостей между словоупотреблениями в тексте рассмотрим следующие объекты:

– Конечное множество типов синтаксических связей: *SR* [30, 127].
– Бинарное отношение на множестве словоупотреблений: $\Sigma^\tau \subseteq W^\tau \times W^\tau$, представляющее всевозможные синтаксические связи между словоупотреблениями согласно работам [29, 30, 70, 127]. Поскольку в этих работах синтаксические структуры рассматриваются в виде деревьев, введённое представление корректно, и мы будем считать, что в паре элементов $< w_i , w_j >$ первый элемент – $w_i \in W^\tau$ – соответствует

главному словоупотреблению (ГС), а второй элемент – $w_j \in W^\tau$ – зависимому (подчинённому) словоупотреблению.

– Семейство отношений $Syntax^\tau = \left\{ \Sigma_z^\tau \subseteq \Sigma^\tau \mid z \in SR^\tau, \bigcup_{z \in SR} \Sigma_z^\tau = \Sigma^\tau \right\}$ – разбиение множества Σ^τ, где каждое из отношений Σ_z^τ определяет тип синтаксической связи $z \in SR$ для пар словоупотреблений $< w_i, w_j > \in \Sigma_z^\tau$.

Для представления семантической информации в текстах рассмотрим:

– Конечное множество категориально-семантических значений синтаксем *Roles* в соответствии с работами [87, 90]. Для краткости в ходе дальнейшего изложения будем называть их семантическими значениями.
– Бинарное отношение $SemRoles^\tau \subseteq W^\tau \times Roles$, сопоставляющее словоупотреблениям текста семантические значения соответствующих синтаксем[3] [85, 94, 108]. Таким образом, каждое словоупотребление может иметь 0 и более семантических значений в тексте.
– Множество типов отношений R на множестве значений синтаксем (в соответствии с работой [89]).
– Бинарное отношение $\Omega^\tau \subseteq W^\tau \times W^\tau$, определяющее семантически связанные словоупотребления в тексте.
– Семейство отношений $SemRels^\tau = \left\{ \Omega_x^\tau \subseteq \Omega^\tau \mid x \in R, \bigcup_{x \in R} \Omega_x^\tau = \Omega^\tau \right\}$ – разбиение множества Ω^τ, где каждое отношение Ω_x^τ определяет тип семантической связи $x \in R$ для пар словоупотреблений $< w_i, w_j > \in \Omega_x^\tau$, где $w_i \in W^\tau$ и $w_j \in W^\tau$.

[3] В общем случае в тексте синтаксема является некоторой синтаксической конструкцией. Например, синтаксема может быть представлена синтаксической группой (предложной или именной). При решении задачи информационного поиска будем связывать семантическое значение не с синтаксемой в целом, а с главным словом синтаксической конструкции, представляющей синтаксему.

Таким образом, в предложенном представлении текст описывается множествами различных признаков:

- внетекстовых (теговая разметка, веса вхождений слов);
- лексико-морфологических (нормальные формы лексем, формы словоупотреблений);
- синтаксических (словосочетания, представленные именными и глагольными группами);
- семантических (значения синтаксем и семантические связи).

2.2 Метод оценки сходства текстов

Пусть задан эталонный текст $\varepsilon \in E$ и сопоставляемый с ним текст $\tau \in E$. Определим функционал, оценивающий сходство текстов ε и τ в рамках описанного выше представления текстовой информации.

Оценку близости текстов будем проводить по множествам предложений этих текстов: S^{ε} и S^{τ} соответственно. Выберем два произвольных предложения $\mathrm{s}^{\varepsilon} \in S^{\varepsilon}$ и $\mathrm{s}^{\tau} \in S^{\tau}$. Рассмотрим множество $N\left(\mathrm{s}^{\varepsilon}, \mathrm{s}^{\tau}\right) = \left\{ < w^{\varepsilon}, w^{\tau} > \in W^{\varepsilon} \times W^{\tau} \mid w^{\varepsilon} \in \mathrm{s}^{\varepsilon} \exists w^{\tau} \in \mathrm{s}^{\tau} \exists d \in D : < w^{\varepsilon}, d > \in \delta^{\varepsilon} \,\&\, < w^{\tau}, d > \in \delta^{\tau} \right\}$ – суть множество пар словоупотреблений, таких что первый элемент пары входит в состав предложения эталонного текста s^{ε}, а второй элемент входит в предложение s сопоставляемого текста, и при этом оба эти словоупотребления совпадают по нормальной форме *d* лексемы. Такие словоупотребления будем называть *соответственными*. Сопоставление предложений s^{ε} и s^{τ} производится путём рассмотрения соответственных словоупотреблений, составляющих множество $N\left(\mathrm{s}^{\varepsilon}, \mathrm{s}^{\tau}\right)$.

Рассмотрим критерии оценки сходства предложений s^{ε} и s^{τ}.

1. ***Покрытие предложения-эталона*** s^{ε} ***предложением*** s^{τ} ***сопоставляемого текста*** τ:

$$I_1\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right)=\sum_{<w^{\varepsilon},w^{\tau}>\in N\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right)} v(w^{\varepsilon}). \qquad (2.1)$$

В качестве функции для определения весов $v(w^{\varepsilon})$ может применяться классическая оценка информационной значимости на основе inverse document Frequency (IDF) [1, 2, 104] или характеристика тематической значимости (ХТЗ), описанная в работах[41, 43, 128]. Применение ХТЗ возможно, если для эталонного текста s^{ε} известна его тематическая принадлежность.

В качестве ограничения потребуем, чтобы сумма весов словоупотреблений текста-эталона была равна единице:

$$\sum_{w^{\varepsilon}\in W^{\varepsilon}} v(w^{\varepsilon})=1. \qquad (2.2)$$

2. ***Общая оценка информационной значимости слов предложения-эталона*** s^{ε} ***в предложении*** s^{τ} ***сопоставляемого текста*** τ определяется формулой:

$$I_2\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right)=\sum_{<w^{\varepsilon},w^{\tau}>\in N\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right)} f(w^{\varepsilon},w^{\tau})v(w^{\varepsilon})v'(w^{\tau}). \qquad (2.3)$$

В этой формуле функционал $f(w^{\varepsilon},w^{\tau})$ соответствует «штрафу» за несовпадение форм словоупотреблений w^{ε}, w^{τ}:

$$f(w^{\varepsilon},w^{\tau})=\begin{cases}1, & \exists \mathrm{g}\in\Phi :< w^{\varepsilon},g>\in\psi^{\varepsilon}\ \&\ < w^{\tau},g>\in\psi^{\tau},\\ f_0, & \text{в противном случае,}\end{cases} \qquad (2.4)$$

где $0\le f_0\le 1$ – параметр метода.

В качестве весов $v(w^{\varepsilon})$, аналогично предыдущему пункту, могут использоваться IDF или ХТЗ (если известна тематическая принадлежность текстов) с сохранением условия нормировки (2.2). В качестве функции для определения весов $v'(w^{\tau})$ может быть выбрана классическая оценка term frequency (TF) [1, 2] или её модификации [129–131]. Дополнительным ограничением на величину $v'(w^{\tau})$ является следующее условие:

$$0 \le v'(w^{\tau}) \le 1. \tag{2.5}$$

Пусть $< w^{\tau}, t > \in \theta^{\tau}$ словоупотребление и соответствующий ему тег разметки. В качестве $v'(w^{\tau})$ может рассматриваться модификация предложенной в работе [43] логарифмической оценки lTF:

$$v'(w^{\tau}) = \omega(t) \frac{\log\left(1 + \sum\limits_{<w',t'> \in \left\{<w,t''> \in \theta^{\tau} \mid t'' \in T \,\&\, \exists d \in D: <w^{\tau}, d> \in \delta^{\tau} \,\&\, <w,d> \in \delta^{\tau}\right\}} \omega(t')\right)}{\log\left(1 + \sum\limits_{<w',t'> \in \left\{<w,t''> \in \theta^{\tau}\right\}} \omega(t')\right)}, \tag{2.6}$$

Логарифмическая оценка веса словоупотребления w^{τ} равна произведению тегового веса, связанного с этим словоупотреблением – $\omega(t)$, на величину, пропорциональную логарифму суммарного тегового веса всех словоупотреблений w лексемы $d :< w, d > \in \delta^{\tau}$ в тексте τ. В отличие от работы [43] предлагается оценивать эту величину не прямым подсчётом количества повторений лексемы в тексте, а суммировать теговые веса $\omega(t)$ словоупотреблений этой лексемы – формула (2.6).

Заметим, что из условий (2.2) и (2.5) следует, что выполняется соотношение:

$$0 \le I_2\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right) \le 1. \qquad (2.7)$$

3. ***Оценку сходства предложения-эталона*** s^{ε} ***и предложения*** s^{τ} ***сопоставляемого текста на основе совпадения синтаксических структу***р зададим формулой:

$$I_3\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right) = \frac{\sum\limits_{<w^{\varepsilon},w^{\tau}>\in N_{Syn}\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right)} v(w^{\varepsilon})}{\sum\limits_{w^{\varepsilon}\in\left\{w\in W^{\varepsilon}\,|\,\exists w'\in W^{\varepsilon}:<w,w'>\in\Sigma^{\varepsilon}\right\}} v(w^{\varepsilon})}. \qquad (2.8)$$

В этой формуле $N_{Syn}\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right) = \left\{ < w^{\varepsilon}, w^{\tau} > \in N\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right) \,|\, \exists \tilde{w}^{\varepsilon} \in W^{\varepsilon}, \exists \tilde{w}^{\tau} \in W, \exists z \in SR : < \tilde{w}^{\varepsilon}, \tilde{w}^{\tau} > \in N\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right) \,\&\, < w^{\varepsilon}, \tilde{w}^{\varepsilon} > \in \Sigma_z^{\varepsilon} \,\&\, < w^{\tau}, \tilde{w}^{\tau} > \in \Sigma_z^{\tau} \right\}$ представляет собой множество пар соответственных словоупотреблений в эталонном предложении s^{ε} и сопоставляемом предложении s^{τ}, для которых совпадают (по нормальным формам лексем) главные $< w^{\varepsilon}, w^{\tau} > \in N\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right)$ и зависимые $< w^{\varepsilon}, w^{\tau} > \in N\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right)$ слова, а сами словоупотребления связаны в контексте эталонного и сопоставляемого предложения однотипными синтаксическими связями: $< w^{\varepsilon}, \tilde{w}^{\varepsilon} > \in \Sigma_z^{\varepsilon} \,\&\, < w^{\tau}, \tilde{w}^{\tau} > \in \Sigma_z^{\tau}$. Знаменатель формулы (2.8) – есть совокупный вес словоупотреблений, которые являются главными элементами в синтаксически связанных парах словоупотреблений в эталонном тексте. Величина $I_3\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right)$ – суть отношение суммарного веса словоупотреблений, у которых в сопоставляемых текстах совпадают синтаксические связи, к

общему весу словоупотреблений, участвующих в синтаксических связях в эталонном тексте.

По построению величина $I_3\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right)$ удовлетворяет условию $0 \le I_3\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right) \le 1$.

Заметим, что знаменатель формулы (2.8) может обратиться в нуль, если набор словоупотреблений W^{ε} не образует синтаксически связанного текста. В таком случае этот критерий не имеет смысла рассматривать и следует положить $I_3\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right) \equiv 0$.

4. Рассмотрим множество таких словоупотреблений с семантическими значениями в предложении эталонного текста, что для них в сопоставляемом тексте имеются соответственные словоупотребления, которым приписаны такие же семантические значения: $\rho\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right) = \left\{ < w^{\varepsilon}, a > \in SemRoles^{\varepsilon} \mid w^{\varepsilon} \in \mathrm{s}^{\varepsilon} \;\&\; a \in Roles \;\&\; \exists w^{\tau} \in \mathrm{s}^{\tau} < w^{\varepsilon}, w^{\tau} > \in N\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right) \;\&\; < w^{\tau}, a > \in SemRoles^{\tau} \right\}$. Тогда ***сходство предложения-эталона*** s^{ε} ***и предложения*** s^{τ} ***сопоставляемого текста*** τ ***на основе совпадения семантических значений*** определяется формулой:

$$I_4\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right) = \frac{\left|\rho\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right)\right|}{\left|SemRoles^{\varepsilon}\right|}. \qquad (2.9)$$

Числитель в формуле (2.9) выражает количество совпавших семантических значений у соответственных словоупотреблений в предложении эталона и в предложении сопоставляемого текста. Знаменатель формулы (2.9) задаёт условие нормировки на 1 по всем словоупотреблениям, имеющим семантические значения в тексте эталона: $0 \le I_4\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right) \le 1$.

Заметим, что знаменатель формулы (2.9) может обратиться в нуль, если в предложениях текста ε не установлены семантические значения. В таком случае этот критерий не имеет смысла рассматривать и следует считать $I_4\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right)\equiv 0$.

5. Для оценки сходства предложений на основе совпадения семантических связей определим множество $SemR_w^{\tau}=\left\{a\in Roles\,|\;\exists w'\in W^{\tau},\exists x\in R:\quad <w,w'>\in\Omega_x^{\tau}\;\&\quad <w',a>\in SemRoles^{\tau}\right\}$. Это – множество значений синтаксем, приписанных в тексте τ тем словоупотреблениям *w'*, которые связаны в этом тексте со словоупотреблением *w* всевозможными семантическими связями. ***Сходство предложения-эталона*** s^{ε} ***и предложения*** s^{τ} ***сопоставляемого текста*** τ ***на основе семантических связей*** зададим формулой:

$$I_5\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right)=\frac{\sum\limits_{<w^{\varepsilon},w^{\tau}>\in N\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right)}\left|SemR_{w^{\varepsilon}}^{\varepsilon}\cap SemR_{w^{\tau}}^{\tau}\right|}{\left|SemRoles^{\varepsilon}\right|}. \qquad (2.10)$$

Аналогично формуле (2.9) числитель в формуле (2.10) выражает количество совпавших семантических связей у соответственных словоупотреблений в предложении эталона и в предложении сопоставляемого текста. Знаменатель формулы задаёт условие нормировки на 1 по всем словоупотреблениям, имеющим семантические связи в тексте эталона: $0\le I_5\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right)\le 1$.

Заметим, что знаменатель формулы (2.10) может обратиться в нуль, если в предложениях текста ε не установлены семантические связи. В таком случае этот критерий не имеет смысла рассматривать и следует положить $I_5\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right)\equiv 0$.

Общая оценка сходства предложения s^{ε} текста-эталона и предложения s^{τ} сопоставляемого текста определяется взвешенной суммой вышеописанных величин:

$$Sim\left(s^{\varepsilon}, s^{\tau}\right) = \sum_{n=1}^{5} \lambda_n I_n \left(s^{\varepsilon}, s^{\tau}\right). \tag{2.11}$$

В формуле (2.11) набор $L=\{\lambda_i | i=1..5\}$ задаёт набор параметров метода, определяющих вклад каждого из критериев оценки сходства в итоговую величину. Зададим ограничение на набор параметров L:

$$\sum_{n=1}^{5} \lambda_n = 1. \tag{2.12}$$

Теперь из всех возможных предложений s^{τ} в сопоставляемом тексте τ выберем те, которые наилучшим образом (в смысле максимизации оценки (2.11)) соответствуют предложению эталона s^{ε}. Величина, определяющая наибольшие оценки близости предложений найденного текста к предложению эталона s^{ε}, выражается следующей формулой:

$$Y\left(s^{\varepsilon}, \tau\right) = \max_{s^{\tau} \in S^{\tau}} \left\{ Sim\left(s^{\varepsilon}, s^{\tau}\right) \right\}. \tag{2.13}$$

С учётом введённых выше величин общая оценка сходства текста-эталона ε по отношению к тексту τ выражается следующей формулой:

$$X\left(\varepsilon, \tau\right) = \sum_{s^{\varepsilon} \in S^{\varepsilon}} Y\left(s^{\varepsilon}, \tau\right). \tag{2.14}$$

В завершение настоящего параграфа исследуем свойства величин, заданных формулами (2.11) и (2.14), представляющие интерес с точки зрения их практического применения для оценки сходства текстов в ИАС.

Утверждение 2.1. $0 \le Sim\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right) \le 1$.

Доказательство. Выберем произвольные предложение s^{ε} текста-эталона и предложение s^{τ} сопоставляемого текста.

А) Неотрицательность: $Sim\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right) \ge 1$. Величина (2.11) представляет собой сумму произведений неотрицательных величин: $\forall n = \overline{1,5}\ \lambda_n I_n\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right) \ge 0$, следовательно, эта сумма неотрицательна.

Б) Нормировка: $Sim\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right) \le 1$.

Умножая соответствующий член λ_n суммы (2.12) на $I_n\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right) \le 1$, получаем слагаемые суммы (2.12). При этом $\forall n = \overline{1,5}\ \lambda_n I_n\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right) \le \lambda_n$, а значит,

$$Sim\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right) \le \sum_{n=1}^{5} \lambda_n = 1.$$

Утверждение 2.1 доказано.

Утверждение 2.2. $0 \le X\left(\varepsilon,\tau\right) \le 1$.

Доказательство.

А) Неотрицательность: $X\left(\varepsilon,\tau\right) \ge 0$. По определению (2.13) и в силу доказанного утверждения 2.1 имеет место неравенство $Y\left(\mathrm{s}^{\varepsilon},\tau\right) \ge 0$, поскольку функция выбора максимума из множества неотрицательных величин не уменьшает их.

Б) Нормировка: $X\left(\varepsilon,\tau\right) \le 1$. Из определения (2.14) путём подстановки в него соотношения (2.13) получим следующее соотношение:

$$X\left(\varepsilon,\tau\right) = \sum_{\mathrm{s}^{\varepsilon} \in S^{\varepsilon}} \max_{\mathrm{s}^{\tau} \in S^{\tau}} \left\{ Sim\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right) \right\}. \qquad (2.15)$$

Рассмотрим множество предложений сопоставляемого текста, на которых достигаются соответствующие максимумы: $\arg\max\left(\mathrm{s}^{\varepsilon}, S^{\tau}\right) = \left\{ s^{\tau} \in S^{\tau} \mid Sim\left(\mathrm{s}^{\varepsilon}, \mathrm{s}^{\tau}\right) = \max_{\tilde{\mathrm{s}}^{\tau} \in S^{\tau}} \left\{ Sim\left(\mathrm{s}^{\varepsilon}, \tilde{\mathrm{s}}^{\tau}\right) \right\} \right\}$. С учётом определения (2.11), полагая везде далее $\tilde{s}^{\tau} \in \arg\max(s^{\varepsilon}, S^{\tau})$, преобразуем формулу (2.15):

$$X\left(\varepsilon,\tau\right) = \sum_{\mathrm{s}^{\varepsilon} \in S^{\varepsilon}} \sum_{n=1}^{5} \lambda_n I_n\left(\mathrm{s}^{\varepsilon}, \tilde{\mathrm{s}}^{\tau}\right) = \sum_{n=1}^{5} \lambda_n \sum_{\mathrm{s}^{\varepsilon} \in S^{\varepsilon}} I_n\left(\mathrm{s}^{\varepsilon}, \tilde{\mathrm{s}}^{\tau}\right). \qquad (2.16)$$

С учётом ограничения (2.12) формула (2.16) будет удовлетворять условию нормировки на 1, если выражение, включающее вторую сумму $\chi_n\left(\varepsilon,\tau\right) = \sum_{\tilde{\mathrm{s}}^{\tau} \in S^{\tau}} I_n\left(\mathrm{s}^{\varepsilon}, \tilde{\mathrm{s}}^{\tau}\right)$, будет удовлетворять условию $\chi_n\left(\varepsilon,\tau\right) \le 1$ для всех $n = \overline{1,5}$.

Покажем, что это условие выполнено для каждого из критериев 1–5. Для этого предварительно заметим, что, как было указано выше, множество S^{ε} предложений текста ε индуцирует разбиение множества словоупотреблений, входящих в этот текст (т.к. всякое словоупотребление относится к одному и только к одному предложению). В силу этого имеем:

1. Из определения (2.1) получаем: $\chi_1\left(\varepsilon,\tau\right) = \sum_{\mathrm{s}^{\varepsilon} \in S^{\varepsilon}} I_1\left(\mathrm{s}^{\varepsilon}, \tilde{\mathrm{s}}^{\tau}\right) = \sum_{\mathrm{s}^{\varepsilon} \in S^{\varepsilon}} \sum_{< w^{\varepsilon}, w^{\tau} > \in N\left(\mathrm{s}^{\varepsilon}, \tilde{\mathrm{s}}^{\tau}\right)} v(w^{\varepsilon})$. Поскольку в этом соотношении двойное суммирование осуществляется по разбиению множества словоупотреблений на предложения, то в этой сумме словоупотребления эталонного текста учитываются не более одного раза: $\bigcup_{\mathrm{s}^{\varepsilon} \in S^{\varepsilon}} \left\{ < w^{\varepsilon}, w^{\tau} > \mid < w^{\varepsilon}, w^{\tau} > \in N\left(\mathrm{s}^{\varepsilon}, \tilde{\mathrm{s}}^{\tau}\right) \right\} \subseteq W^{\varepsilon}$. Отсюда с учётом соотношения (2.2) следует, что $\sum_{\mathrm{s}^{\varepsilon} \in S^{\varepsilon}} \sum_{< w^{\varepsilon}, w^{\tau} > \in N\left(\mathrm{s}^{\varepsilon}, \tilde{\mathrm{s}}^{\tau}\right)} v(w^{\varepsilon}) \le \sum_{w^{\varepsilon} \in W^{\varepsilon}} v(w^{\varepsilon}) = 1$.

Причём равенство здесь достигается, тогда и только тогда, когда в сопоставляемом тексте содержатся предложения, содержащие все словоупотребления эталонного текста.

2. Из условий (2.4), (2.5) получаем, что $f(w^{\varepsilon}, w^{\tau})v(w^{\tau})v(w^{\varepsilon}) \le v(w^{\varepsilon})$, откуда аналогично предыдущему пункту доказательства с учётом определения (2.3) следует, что: $\chi_2(\varepsilon,\tau) = \sum_{s^{\varepsilon} \in S^{\varepsilon}} \sum_{<w^{\varepsilon}, w^{\tau}> \in N\left(s^{\varepsilon}, \tilde{s}^{\tau}\right)} f(w^{\varepsilon}, w^{\tau})v(w^{\tau})v(w^{\varepsilon}) \le \chi_1(\varepsilon,\tau) \le 1.$

3. С учётом определения (2.8) получаем: $\chi_3(\varepsilon,\tau) = \sum_{s^{\varepsilon} \in S^{\varepsilon}} \frac{\sum_{<w^{\varepsilon}, w^{\tau}> \in N_{Syn}\left(s^{\varepsilon}, \tilde{s}^{\tau}\right)} v(w^{\varepsilon})}{\sum_{w^{\varepsilon} \in \left\{w \in W^{\varepsilon} \mid \exists w' \in W^{\varepsilon} : <w, w'> \in \Sigma^{\varepsilon}\right\}} v(w^{\varepsilon})} = \frac{\sum_{s^{\varepsilon} \in S^{\varepsilon}} \sum_{<w^{\varepsilon}, w^{\tau}> \in N_{Syn}\left(s^{\varepsilon}, \tilde{s}^{\tau}\right)} v(w^{\varepsilon})}{\sum_{w^{\varepsilon} \in \left\{w \in W^{\varepsilon} \mid \exists w' \in W^{\varepsilon} : <w, w'> \in \Sigma^{\varepsilon}\right\}} v(w^{\varepsilon})}$. По построению множества $N_{Syn}\left(s^{\varepsilon}, \tilde{s}^{\tau}\right)$ с учётом разбиения текста на предложения заключаем, что $\bigcup_{s^{\varepsilon} \in S^{\varepsilon}} \left\{ < w^{\varepsilon}, w^{\tau} > \in N_{Syn}\left(s^{\varepsilon}, \tilde{s}^{\tau}\right)\right\} \subseteq \left\{ w \in W^{\varepsilon} \mid \exists w' \in W^{\varepsilon} : < w, w' > \in \Sigma^{\varepsilon} \right\}$, т.е. каждая пара синтаксически связанных словоупотреблений в тексте эталона учитывается в сумме в числителе не более одного раза. Это означает, что справедливо неравенство $\sum_{s^{\varepsilon} \in S^{\varepsilon}} \sum_{<w^{\varepsilon}, w^{\tau}> \in N_{Syn}\left(s^{\varepsilon}, \tilde{s}^{\tau}\right)} v(w^{\varepsilon}) \le \sum_{w^{\varepsilon} \in \left\{w \in W^{\varepsilon} \mid \exists w' \in W^{\varepsilon} : <w, w'> \in \Sigma^{\varepsilon}\right\}} v(w^{\varepsilon})$. Равенство здесь достигается тогда и только тогда, когда все пары словоупотреблений текста эталона, связанные синтаксически, присутствуют в сопоставляемом тексте, и в этих парах присутствуют синтаксические связи того же типа, что и в эталонном тексте. Из вышесказанного следует, что $\chi_3(\varepsilon,\tau) \le 1$.

4. Из определения (2.9) с учётом введённых обозначений получаем, что $\chi_4(\varepsilon,\tau)=\sum_{s^\varepsilon\in S^\varepsilon}\frac{\left|\rho\left(s^\varepsilon,\tilde{s}^\tau\right)\right|}{\left|SemRoles^\varepsilon\right|}=\frac{\sum_{s^\varepsilon\in S^\varepsilon}\left|\rho\left(s^\varepsilon,\tilde{s}^\tau\right)\right|}{\left|SemRoles^\varepsilon\right|}$. Откуда по построению множеств $\rho\left(s^\varepsilon,\tilde{s}^\tau\right)$ и $SemRoles^\varepsilon$ с учётом разбиения текста на предложения заключаем, что $\bigcup_{s^\varepsilon\in S^\varepsilon}\rho\left(s^\varepsilon,\tilde{s}^\tau\right)\subseteq SemRoles^\varepsilon$. В силу этого числитель не превосходит знаменателя, следовательно, $\chi_4\left(\varepsilon,\tau\right)\le 1$. Равенство достигается тогда и только тогда, когда у соответственных словоупотреблений в тексте эталона и в сопоставляемом тексте совпадают все семантические значения.
5. Рассматривая определение (2.10) с учётом определений множеств $SemR_w^\tau$ и $N\left(s^\varepsilon,\tilde{s}^\tau\right)$, заключаем, что $\bigcup_{<w^\varepsilon,w^\tau>\in N\left(s^\varepsilon,\tilde{s}^\tau\right)}SemR_{w^\varepsilon}^\varepsilon\cap SemR_{w^\tau}^\tau\subseteq\bigcup_{<w^\varepsilon,w^\tau>\in N\left(s^\varepsilon,\tilde{s}^\tau\right)}SemR_{w^\varepsilon}^\varepsilon\subseteq SemRoles^\varepsilon$. Отсюда следует, что в определении (2.10) числитель не превосходит знаменателя, т.е. $\chi_5\left(s^\varepsilon,\tilde{s}^\tau\right)\le 1$. Равенство здесь достигается тогда и только тогда, когда у соответственных словоупотреблений в тексте эталона и в сопоставляемом тексте совпадают все значения у всех семантически связанных слов.

Утверждение 2.2 доказано.

В заключение настоящего параграфа отметим, что предложенная оценка сходства двух текстов (2.14) в общем случае несимметрична. Однако этот факт не является недостатком на практике при решении задач полнотекстового поиска информации по запросу или заданному эталонному тексту.

2.3 Применение разработанного метода оценки сходства текстов для решения поисково-аналитических задач

Представленный в работе метод оценки сходства текстов предназначен для реализации следующих функций поиска и анализа научно-технической информации в ИАС:

- полнотекстовый поиск по ключевым словам и словосочетаниям;
- фразовый поиск;
- семантический поиск по запросу на ЕЯ, включая вопросно-ответный поиск.

Вышеперечисленные функции поиска должны поддерживать возможность задания ограничений на метаданные документов, которые также могут быть сформулированы на ЕЯ.

Рассмотрим аспекты практической реализации функций полнотекстового поиска и оценки релевантности его результатов на основе разработанного метода оценки сходства текстов.

2.3.1 Применение разработанного метода оценки сходства текстов для решения задач полнотекстового поиска по ключевым словам и словосочетаниям

При поиске запрос пользователя на ЕЯ рассматривается в качестве эталонного текста. С применением методов лингвистического анализа к поисковому запросу формируется представление текста запроса. Отметим, что запрос может состоять как из нескольких предложений или фраз, так и из единственного словоупотребления. По нормальным формам словоупотреблений выполняется выборка данных из специальной структуры данных, содержащей информацию о словоупотреблениях в текстах коллекции документов, в которой осуществляется поиск. Эта структура данных содержит информацию о текстах документов в соответствии с представлением текста, описанным в разделе 2.1. Для наполнения этой структуры тексты документов подвергнуты предварительному лингвистическому анализу.

Отметим, что в случае коротких запросов морфологический анализатор не способен разрешить омонимию (на уровне нормальных форм) и предоставляет несколько вариантов нормализации. Поэтому, чтобы обеспечить полноту и не потерять релевантные результаты поиска, выборка информации должна производиться с учётом омонимов нормальных форм. Этот факт учтён при определении отношений δ^{τ} и $N\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right)$. Если морфологический анализатор предлагает варианты нормализации словоупотреблений с некоторыми весами (например, вероятностными оценками), то они могут быть дополнительно учтены при расчёте весов $X\left(\varepsilon,\tau\right)$.

При ранжировании результатов поиска документы упорядочиваются по убыванию величины $X\left(\varepsilon,\tau\right)$. По умолчанию считается, что запросу релевантны все документы, содержащие хотя бы одно слово запроса. В булевом поиске этому принципу соответствует объединение слов запроса через оператор «ИЛИ» [1, 2]. Однако нерелевантные результаты можно исключить из выдачи по заданному порогу минимального сходства X_{min}. Результат считается нерелевантным и исключается из выдачи, если $X\left(\varepsilon,\tau\right) < X_{\min}$. Утверждение 2.2, доказанное в п. 2.2, определяет границы, в которых целесообразно задавать этот параметр: $0 \le X_{\min} \le 1$.

Далее, в главе, посвящённой разработке алгоритмов информационного поиска будет показано, как в предложенном методе оценки сходства текстов реализуется поддержка языка пользовательской разметки запросов, который позволяет:

- исключать из результатов поиска документы, содержащие выбранные слова запроса (оператор «\»);
- исключать из рассмотрения отдельные предложения документов, содержащие выбранные слова запроса (оператор «~»);
- задавать слова запроса, которые обязательно должны присутствовать в результатах поиска (оператор «+»), в том числе в определённой форме (оператор «&»).

Реализация функций полнотекстового поиска должна поддерживать поиск с учётом ограничений на метаданные документов. Эти ограничения могут выражаться в нетекстовой форме (фильтрация по году опубликования/индексации документа, ресурсу-источнику, тематической рубрике и т.п.). Также ограничения на метаданные искомых документов могут быть выражены на естественном языке, если пользователя интересуют документы, содержащие определённые слова или фразы в заголовке, аннотации, авторах, гипертекстовых ссылках и др. полях текстовых метаданных.

В общем случае в задаче поиска с учётом метаданных требуется находить тексты, в которых определённые фрагменты (предложения) поискового запроса выделены специальными тегами гипертекстовой разметки либо тегами разметки метаданных. Для решения этой задачи в определение соответственных словоупотреблений дополнительно добавим условие совпадения тегов у соответственных вхождений слов, т.е. положим при задании множества $N\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right)$:

$$\begin{aligned} N\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right)=\{&<w^{\varepsilon},w^{\tau}>\in W^{\varepsilon}\times W^{\tau}\mid w^{\varepsilon}\in \mathrm{s}^{\varepsilon}\,\exists w^{\tau}\in \mathrm{s}^{\tau}\,\exists d\in D\,\exists t\in T: \\ &<w^{\varepsilon},d>\in\delta^{\varepsilon}\,\&<w^{\tau},d>\in\delta^{\tau}\,\&<w^{\varepsilon},t>\in\theta^{\varepsilon}\,\&<w^{\tau},t>\in\theta^{\tau}\} \end{aligned} \tag{2.17}$$

Таким образом, формула (2.17) выражаёт условие фильтрации соответственных словоупотреблений по тегам метаданных.

2.3.2 Применение разработанного метода оценки сходства текстов для решения задач фразового поиска

Под фразой будем понимать некоторое подмножество словоупотреблений, связанных друг с другом синтаксически, в составе предложения. В качестве фразы может выступать термин предметной области («*интеллектуальные методы поиска*»), ФИО персоны («*Пётр Струве*») или организации, а также предложение целиком («*лиса ест манную кашу*») – любые

конструкции, в которых синтаксический анализатор устанавливает связи между входящими в них словами. Таким образом, фразовый поиск позволяет находить предложения текстов документов, содержащие лексемы запроса, связанные теми же синтаксическими связями, что присутствуют между соответствующими словоупотреблениями в запросе. При решении задач информационного поиска в настоящей работе мы ограничимся синтаксическими связями, в которых в качестве главного слова выступает существительное – таблица 2.1.

Таблица 2.1 – Примеры синтаксических связей между словоупотреблениями

ЧАСТЬ РЕЧИ		**ПРИМЕРЫ**
ГЛАВНОГО СЛОВА	**ЗАВИСИМОГО СЛОВА**	
существительное	прилагательное	*файловая система,* *preemptive multitasking*
существительное	причастие	*вытесняющая многозадачность,* *finished task*
существительное	существительное	*менеджер памяти,* *управление рисками,* *data mining*
существительное	предлог + существительное	*поездка в Сибирь,* *journey to Siberia*

Этот выбор обоснован тем, что фразовый поиск направлен на обнаружение упоминаний конкретных терминов и понятий в текстах. При этом в русском и наиболее распространённых европейских языках термины чаще всего выражаются словосочетаниями, имеющими выбранную синтаксическую структуру (см., например работы [132–135]). Без каких-либо ограничений в предложенном представлении текста могут учитываться связи «существительное – предлог – существительное», где каждому предлогу

соответствует свой тип синтаксической связи. При этом предлог выступает в роли атрибута связи – Рисунок 2.1.

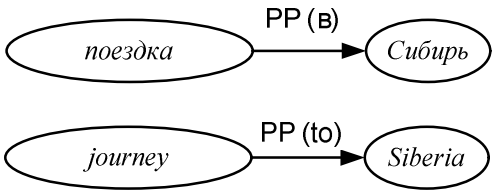

Рисунок 2.1 – Примеры синтаксических связей «существительное – предлог – существительное»

В отличие от функции поиска по точному соответствию, реализованной в большинстве поисковых машин [1, 2], при предлагаемом подходе к решению задачи фразового поиска будут найдены предложения, соответствующие запросу лексически и синтаксически, но, возможно, отличающиеся формой и порядком словоупотреблений. Для примеров, приведённых выше, в текстах документов будут найдены следующие фразы: «*интеллектуальный метод информационного поиска*», «*Пётр Бернгардович Струве*», «*кашу манную лисы едят с удовольствием*». Соответствующие синтаксические деревья приведены на рисунке 2.2.

При этом нерелевантные результаты («*инструменты поиска интеллектуальных методов анализа данных*») будут ранжированы ниже релевантных, поскольку им соответствуют синтаксические деревья другой структуры. Эти результаты могут быть полностью исключены из выдачи.

Для формализации метода фразового поиска выделим в тексте множество фраз $\Pi^{\tau} = \left\{ \pi_{p,m}^{\tau} \right\}$, где $\pi_{p,m}^{\tau} = \left\{ \mathrm{w}_{\mathrm{i}_m}, ..., \mathrm{w}_{\mathrm{j}_m} \right\}$, $\{i_m, ..., j_m\}$ – произвольный непустой набор индексов словоупотреблений в предложении s_p^{τ} текста τ, входящих в m-ю фразу. Фразы выделяются внутри предложений; каждая фраза соответствует связному подграфу в дереве синтаксического разбора предложения, в ней выделяется ГС. ГС $MW(\pi_{p,m}^{\tau})$ фразы $\pi_{p,m}^{\tau}$ – это такое словоупотребление в составе фразы, которое не является зависимым ни от какого-либо другого словоупотребления в составе этой фразы:

$MW(\pi^{\tau}_{p,m}) = w \Leftrightarrow w \in \pi^{\tau}_{p,m} \,\&\, \left(\neg\exists z \in SR\right) \& \left(\neg\exists w' \in \pi^{p}_{m}\right) :< w, w' > \in \Sigma^{\tau}_{z}$. В тривиальном случае каждая фраза содержит в точности одно словоупотребление.

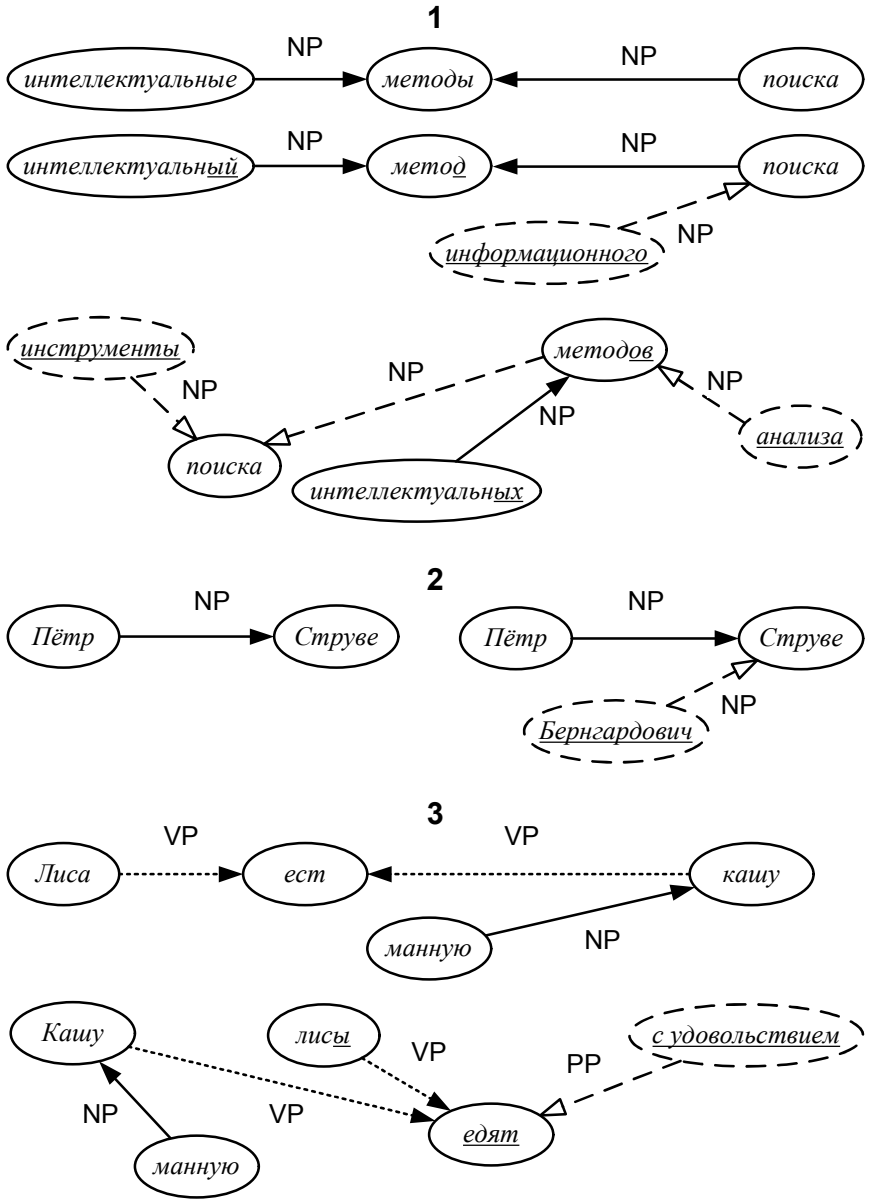

Рисунок 2.2 – Примеры синтаксических деревьев, сопоставляемых при фразовом поиске

При фразовом поиске предложения найденных документов сопоставляются с предложением запроса таким образом, чтобы в них совпала хотя бы одна из фраз, присутствующих в предложении запроса. Под совпадением фраз здесь подразумевается наличие всех синтаксических связей у

соответствующих словоупотреблений в составе фраз в предложениях эталона и найденных предложениях. Это условие выражается в следующем виде:

$$N'_{Syn}\left(\pi^{\varepsilon}_{p,m},\pi^{\tau}_{q,n}\right)=\left\{<w^{\varepsilon},w^{\tau}>\in N\left(s^{\varepsilon}_{p},s^{\tau}_{q}\right)\right|$$
$$\exists z\in SR\,\exists\tilde{w}^{\varepsilon}\in W^{\varepsilon}:<w^{\varepsilon},\tilde{w}^{\varepsilon}>\in\Sigma^{\varepsilon}_{z}\ \&\ \exists\tilde{w}^{\tau}:<w^{\tau},\tilde{w}^{\tau}>\in\Sigma^{\tau}_{z}\ \& \tag{2.18}$$
$$<\tilde{w}^{\varepsilon},\tilde{w}^{\tau}>\in N\left(s^{\varepsilon}_{p},s^{\tau}_{q}\right)\Big\}.$$

Тогда предложение s^{τ}_{q} найденного текста содержит фразу $\pi^{\tau}_{q,n}$, соответствующую фразе $\pi^{\varepsilon}_{p,m}$ текста эталона тогда и только тогда, когда выполнено условие:

$$\forall w^{\varepsilon}\in\pi^{\varepsilon}_{p,m}\left(\exists z\in SR,\exists\tilde{w}^{\varepsilon}\in\pi^{\varepsilon}_{p,m}<w^{\varepsilon},\tilde{w}^{\varepsilon}>\in\Sigma^{\varepsilon}_{z}\right)\Rightarrow$$
$$\Rightarrow\left(\exists w^{\tau}\in\pi^{\tau}_{q,n}:<w^{\varepsilon},w^{\tau}>\in N'_{Syn}\left(\pi^{\varepsilon}_{p,m},\pi^{\tau}_{q,n}\right)\right) \tag{2.19}$$

Это условие означает, что для каждого словоупотребления w^{ε} в составе фразы $\pi^{\varepsilon}_{p,m}$, имеющего синтаксически связанное словоупотребление $\tilde{w}^{\varepsilon}$ (в качестве зависимого словоупотребления), должно найтись соответственное словоупотребление w^{τ} в предложении s^{τ}_{q}, имеющее в этом предложении синтаксически связанное словоупотребление $\tilde{w}^{\tau}$, причём словоупотребления $\tilde{w}^{\varepsilon}$ и $\tilde{w}^{\tau}$ также должны быть соответственными.

В языке разметки запросов для фразового поиска определён способ объединения нескольких словоупотреблений в группу: они заключаются в фигурные скобки «{…}». Внутри группы в соответствии с введёнными выше определениями выделяются фразы. Далее выполняется поиск документов, в которых присутствуют все фразы, входящие в состав группы. К группам/фразам также применимы вышеперечисленные операции языка запросов.

2.3.3 Применение разработанного метода оценки сходства текстов для решения задач семантического и вопросно-ответного поиска

Вопросно-ответный и семантический поиск являются дальнейшим развитием фразового поиска. Если поисковый запрос пользователя представляет собой осмысленное предложение или вопрос, то в нём определяется семантическая информация, которая позволяет находить предложения, близкие по смыслу к исходному, или ответ на поставленный вопрос. Пример оценки сходства вопроса пользователя («*кто уничтожил Троцкого?*») и найденного по слову «*Троцкий*» ответа («*Этим ледорубом Рамон Меркадер убил Льва Троцкого*») приведён на рисунке 2.3. При поиске вопросительное слово исключается из рассмотрения, и поиск проводится только по оставшимся словам с учётом семантической информации.

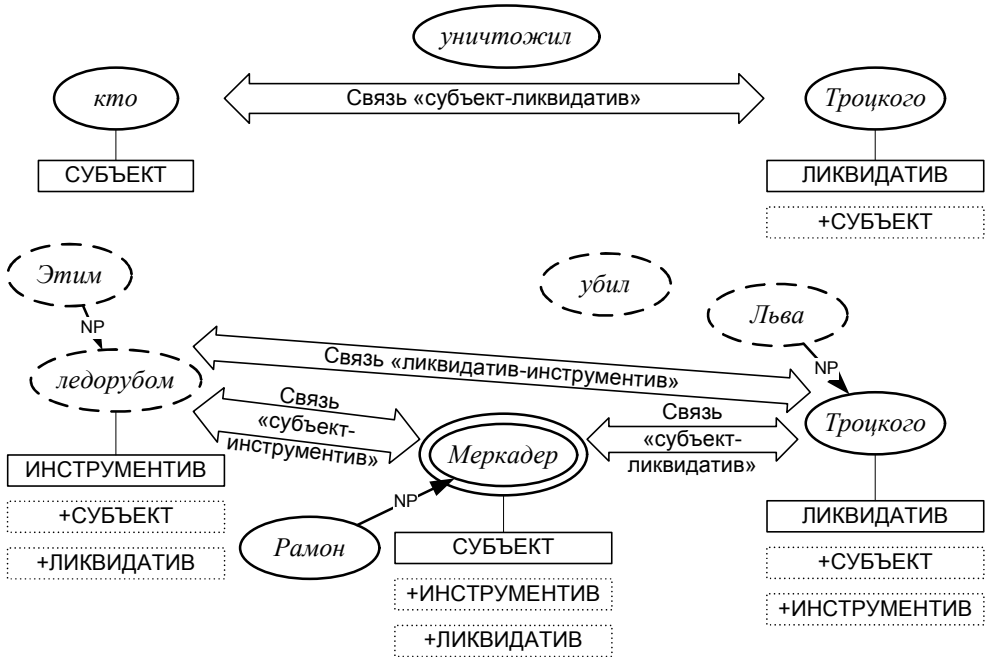

Рисунок 2.3 – Пример оценки сходства семантико-синтаксических структур вопроса и найденного ответа

Критерии 4, 5 из п. 2.2 позволяют находить ответ на поставленный вопрос, если он содержится в текстах коллекции документов. При этом в ряде приложений информационного поиска необходимо учитывать тезаурусные отношения между словами или фразами ЕЯ, например, синонимичные или родовидовые отношения для поиска близких по смыслу формулировок. С этой целью выполняется перифраз исходного текста. Для этого во множество

словоупотреблений W^{ε} текста запроса (в соответствующие предложения) добавляются, например, слова-синонимы. В общем случае синоним подбирается не к отдельному слову, а к словосочетанию, составляющему фразу (соответствующую некоторому термину предметной области, например, «*поисково-аналитическая система*» – «*система поиска и анализа информации*»).

Для формализации пополнения представления текста запроса введём в рассмотрение универсальное множество всевозможных фраз Π и бинарное отношение на множестве фраз $P \subset \Pi \times \Pi$. В паре фраз $<\pi^{\varepsilon}_{p,m}, \tilde{\pi}> \in P$ $\pi^{\varepsilon}_{p,m}$ будем называть эталонной фразой, а $\tilde{\pi}$ – связанной. В случае пополнения текста запроса концептами тезауруса связанная фраза $\tilde{\pi}$ является концептом тезауруса, соотнесённым с фразой запроса $\pi^{\varepsilon}_{p,m}$ некоторым тезаурусным отношением. Текст запроса может быть пополнен множеством фраз $\tilde{\Pi}^{\varepsilon}_{p,m} = \left\{ \tilde{\pi} \mid <\pi^{\varepsilon}_{p,m}, \tilde{\pi}> \in P \right\}$, где $\tilde{\pi} = \left\{ \tilde{w}_1, ..., \tilde{w}_j \right\}$ – множество словоупотреблений, составляющих фразу в тезаурусе. Для пополнения представления текста запроса необходимо:

1. Добавить во множество словоупотреблений текста все словоупотребления, составляющие фразы $\tilde{\pi} \in \tilde{\Pi}^{\varepsilon}_{p,m}$: $W^{\varepsilon}{}' = W^{\varepsilon} \cup \bigcup_{<\pi^{\varepsilon}_{p,m}, \tilde{\pi}> \in P} \tilde{\pi}$. При этом для всех $<\pi^{\varepsilon}_{p,m}, \tilde{\pi}> \in P$ разбиение текста на предложения доопределяется таким образом, чтобы $s^{\varepsilon}_{p}{}' = s^{\varepsilon}_{p} \cup \bigcup_{\tilde{\pi} \in \tilde{\Pi}^{\varepsilon}_{p,m}} \tilde{\pi}$. Т.е. добавляемые фразы из тезауруса соотносятся с эталонными фразами запроса и добавляются в точности в те же самые предложения, в которые входят эталонные фразы. Словоупотребления добавленных фраз не изменяют условий нормировки весов исходного текста и, возможно, рассматриваются с уменьшенным весом, т.е.

требуется, чтобы для всех $<\pi^{\varepsilon}_{p,m},\tilde{\pi}>\in P$ выполнялось соотношение: $\sum_{w\in\tilde{\pi}} v(w) \le \sum_{w\in\pi^{\varepsilon}_{p,m}} v(w)$.

2. Добавить во множество синтаксически связанных словоупотреблений Σ^{ε}_{z} текста соответствующие пары словоупотреблений $\tilde{w}_i,\tilde{w}_j\in\tilde{\pi}$: $<\tilde{w}_i,\tilde{w}_j>\in\tilde{\Sigma}^{\tilde{\pi}}_{z}$, связанных синтаксически в составе добавляемой фразы тезауруса для всех $<\pi^{\varepsilon}_{p,m},\tilde{\pi}>\in P$: $\Sigma^{\varepsilon}_{z}{}' = \Sigma^{\varepsilon}_{z} \cup \bigcup_{<\pi^{\varepsilon}_{p,m},\tilde{\pi}>\in P} \tilde{\Sigma}^{\tilde{\pi}}_{z}$.

3. ГС добавленной фразы $\tilde{\pi}$ целесообразно назначить те же семантические значения и связи, что имеются у фразы $\pi^{\varepsilon}_{p,m}$ в исходном тексте. Для этого необходимо пополнить отношение $SemRoles^{\varepsilon}$ следующим образом: $SemRoles^{\varepsilon}{}' = SemRoles^{\varepsilon} \cup \bigcup_{<\pi^{\varepsilon}_{p,m},\tilde{\pi}>\in P} \left\{ <MW(\tilde{\pi}),a> \mid \exists a\in Roles : <MW(\pi^{\varepsilon}_{p,m}),a>\in SemRoles^{\varepsilon} \right\}$.

   Отношение $SemRels^{\tau}$ при этом необходимо пополнить следующим образом:

$$\Omega^{\tau}_{x}{}' = \Omega^{\tau}_{x} \cup \bigcup_{<\pi^{\varepsilon}_{p,m},\tilde{\pi}>\in P} \left\{ <MW(\pi),w> \mid \exists w\in W^{\tau}\ \exists x\in R < MW(\pi^{\varepsilon}_{p,m}),w>\in\Omega^{\tau}_{x} \right\}$$
$$\cup \bigcup_{<\pi^{\varepsilon}_{p,m},\tilde{\pi}>\in P} \left\{ <w,MW(\pi)> \mid \exists w\in W^{\tau}\ \exists x\in R < w,MW(\pi^{\varepsilon}_{p,m})>\in\Omega^{\tau}_{x} \right\}.$$

Аналогичным образом при поиске информации учитывается кореферентность в текстах документов коллекции. Пусть на этапе лингвистического анализа устанавливаются анафорические связи между фразами, входящими в текст. Рассмотрим ещё одно бинарное отношение на универсальном множестве фраз: $P^{\tau}\subset\Pi\times\Pi$ – это отношение, связующее фразы текста τ анафорическими связями. В паре фраз $<\pi^{\tau}_{p,m},\pi^{\tau}_{q,n}>\in P^{\tau}$ первый элемент $\pi^{\tau}_{p,m}\in s^{\tau}_{p}$ соответствует антецеденту, а $\pi^{\tau}_{q,n}\in s^{\tau}_{q}$ – анафорическому элементу.

Для всех таких пар фраз, что $<\pi^{\tau}_{p,m},\pi^{\tau}_{q,n}>\in P^{\tau}$ и $p \neq q$, т.е. фраза-антецедент и фраза-анафор входят в различные предложения реализуем процедуру пополнения текста следующим образом:

1. Добавим в предложения, содержащие фразы-анафоры $\pi^{\tau}_{q,n}$, реплики соответствующих им фраз-антецедентов. Для этого для всех фраз-анафоров $\pi^{\tau}_{q,n}$, таких, что $\exists \pi^{\tau}_{p,m}: <\pi^{\tau}_{p,m},\pi^{\tau}_{q,n}>\in P^{\tau}$ рассмотрим множество реплик фраз-антецедентов $\hat{\Pi}^{\tau}_{q,n} = \left\{\hat{\pi} \in \Pi \mid \hat{\pi} \cap W^{\tau} = \varnothing\right\}$. При этом потребуем, чтобы каждая реплика $\hat{\pi}$ некоторой фразы-антецедента π обладала следующими свойствами:
   - $|\hat{\pi}| = |\pi| \,\&\, \left(\forall \hat{w}' \in \hat{\pi}\, \exists w \in \pi < \hat{w}', w > \in N(\hat{\pi},\pi)\right) \,\&$
     $\left(\forall w \in \pi\, \exists \hat{w} \in \hat{\pi} < \hat{w}', w > \in N(\hat{\pi},\pi)\right)$ (фраза $\hat{\pi}$ содержит в точности те же самые лексемы, что и исходная фраза π);
   - $\left(\forall \hat{w}' \in \hat{\pi}\, \exists w \in \pi\, \exists g \in \Phi < \hat{w}', g > \in \hat{\psi} \,\&\, < w, g > \in \psi^{\tau}\right) \,\&$
     $\left(\forall w \in \pi\, \exists \hat{w} \in \hat{\pi}\, \exists g \in \Phi \,\&\, < \hat{w}', g > \in \hat{\psi} \,\&\, < w, g > \in \psi^{\tau}\right)$ (формы словоупотреблений фразы $\hat{\pi}$ совпадают с формами словоупотреблений исходной фразы π)
   - $\forall w \in \pi \left(\exists z \in SR \exists w' \in \pi < w, w' > \in \Sigma^{\tau}_{z} \leftrightarrow \exists \hat{w} \in \hat{\pi} < w, \hat{w} > \in N_{Syn}(\pi,\hat{\pi})\right)$ (синтаксическая структура фразы $\hat{\pi}$ повторяет структуру исходной фразы π).
2. Для пополнения представления текста τ фразами-антецедентами положим: $W^{\tau}{}' = W^{\tau} \cup \bigcup\limits_{<\pi,\pi'>\in P} \hat{\pi}$, и для всех пар фраз $<\pi,\pi^{\tau}_{q,n}>\in P^{\tau}$ $s^{\tau}_{q}{}' = s^{\tau}_{q} \cup \bigcup\limits_{\hat{\pi}\in\hat{\Pi}^{\tau}_{q,n}} \hat{\pi}$. Т.е. словоупотребления фраз-антецедентов дублируются и добавляются в предложения, содержащие фразы-анафоры.

3. Добавим во множество синтаксически связанных словоупотреблений Σ_z^τ текста соответствующие пары словоупотреблений $\hat{w}_i, \hat{w}_j \in \hat{\pi}$: $< \hat{w}_i, \hat{w}_j > \in \Sigma_z^{\hat{\pi}}$, связанных синтаксически в составе добавляемой фразы тезауруса для всех $< \pi, \pi_{q,n}^\tau > \in P^\tau$: $\Sigma_z^\tau{}' = \Sigma_z^\tau \cup \bigcup\limits_{<\pi,\pi'>\in P^\tau} \Sigma_z^{\hat{\pi}}$.

– ГС каждой добавленной фразы $\hat{\pi}$ назначим те же семантические значения и связи, что имеются у фразы-анафора $\pi_{q,n}^\tau$ в исходном тексте. Для этого пополним отношение $SemRoles^\tau$: $SemRoles^\tau{}' = SemRoles^\tau \cup \bigcup\limits_{<\pi,\pi'>\in P^\tau} \left\{ < MW(\pi), a > | \exists a \in Roles : < MW(\pi'), a > \in SemRoles^\tau \right\}$. Отношение $SemRels^\tau$ пополняем следующим образом:

$$\Omega_x^\tau{}' = \Omega_x^\tau \cup \bigcup_{<\pi,\pi'>\in P^\tau} \left\{ < MW(\pi), w > | \exists w \in W^\tau \; \exists x \in R < MW(\pi'), w > \in \Omega_x^\tau \right\}$$
$$\cup \bigcup_{<\pi,\pi'>\in P^\tau} \left\{ < w, MW(\pi) > | \exists w \in W^\tau \; \exists x \in R < w, MW(\pi') > \in \Omega_x^\tau \right\}$$

Как и в случае с пополнением текста фразами из тезауруса, необходимо соблюсти условие нормировки весов (2.2). На практике это означает, что итоговый расчёт весов словоупотреблений следует производить после реализации процедуры пополнения текста фразами-антецедентами.

2.4 Выводы

Представленный в настоящей главе метод оценки сходства текстов позволяет проводить многокритериальное сопоставление текстов с учётом различных лингвистических признаков, устанавливаемых на различных этапах компьютерного анализа текстов. Сформулированные критерии оценки сходства предложений текстов опираются на лексико-морфологическую информацию, а также синтаксическую и семантические структуры предложений. Введённая величина общей оценки сходства текста-эталона по отношению к произвольному тексту – формула (2.14) – агрегирует вышеопределённые критерии оценки сходства предложений текстов в вещественный функционал.

Утверждение 2.2, доказанное в настоящей главе, определяет диапазон значений, в котором варьируется величина, заданная формулой (2.14). Это позволяет рассматривать общую оценку сходства текстов (ОСТ) как оценку соответствия запроса пользователя (как эталонного текста) найденным текстам. Таким образом, ОСТ применима для решения задачи ранжирования результатов текстового поиска по запросу на ЕЯ. Причём, если ОСТ не превышает минимального порога $0<X_{min}\leq 1$ для некоторого текста, то этот текст можно считать нерелевантным запросу. Сравнение значения ОСТ с пороговым значением может быть применено и при решении аналогичных задач сопоставления произвольных текстов.

В зависимости от решаемой задачи (классический или фразовый поиск, семантический поиск близких ситуаций или вопросно-ответный поиск) значимость критериев и их вклад в итоговую оценку сходства текстов может варьироваться. Вклад каждого из критериев определяется конкретными значениями в наборе параметров (L, $f_{0,}$ X_{min} и др.). Настройка метода под каждую конкретную задачу должна проводиться на основе контрольных выборок с подбором таких параметров, которые обеспечивают наилучшее качество решения соответствующей задачи.

В заключительном параграфе настоящей главы рассмотрены аспекты применения разработанного метода оценки сходства текстов для решения поисково-аналитических задач. Показано, как в рамках введённого представления текстовой информации решается задача полнотекстового поиска по ключевым словам и словосочетаниям. В заключительном параграфе также исследованы особенности использования метаинформации при полнотекстовом поиске и предложен способ учёта возможной омонимии словоупотреблений в текстах. Кроме того, раскрыты и исследованы особенности применения разработанного метода оценки сходства текстов для решения задач полнотекстового поиска по ключевым словам и словосочетаниям, фразового поиска, а также семантического и вопросно-ответного поиска. При этом процедура семантического поиска предусматривает не только учёт

семантических значений словоупотреблений и связей между словоупотреблениями, но также пополнение текста-эталона фразами из тезаурусов и учёт кореферентности в текстах документов коллекции.

Предложенная формализация представления текстовой информации и метода оценки сходства текстов основана на теории множеств. Разработанное представление текстовой информации может рассматриваться как неоднородная семантическая сеть (НСС) [25], в качестве объектов которой выступают словоупотребления текста, и другие вспомогательные объекты (нормальные формы лексем, теги разметки по метаданным и т.п.). Словоупотребления связаны отношениями различных типов (определяются синтаксическими и семантическими связями) и имеют атрибуты различных типов. Эти атрибуты определяются с помощью функционалов (веса словоупотреблений, тегов) или бинарных отношений на множествах словоупотреблений и вспомогательных объектов (так задаются нормальные формы словоупотреблений, теги метаданных словоупотреблений, разбиение текста на предложения и т.п.).

Для количественной оценки сходства двух текстов устанавливается соответствие между словоупотреблениями в предложениях сопоставляемых текстов с помощью отношения $N\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right)$ – совпадения словоупотреблений по нормальным формам лексем. Это отношение и ряд производных от него отношений позволяют получить из двух НСС сопоставляемых текстов новую НСС, на основе которой производится расчёт ОСТ в соответствии с критериями, сформулированными в п. 2.2.

Построение отношения $N\left(\mathrm{s}^{\varepsilon},\mathrm{s}^{\tau}\right)$ и других отношений в рамках предложенного представления текстовой информации, а также многие математические операции (выбор максимального значения из множества, пересечение, разность множеств и т.п.) требуют специальных приёмов для снижения вычислительной сложности при реализации соответствующих алгоритмов на практике. Предложенное представление текстовой информации

значительно отличается от принятого и наиболее распространённого в системах информационного поиска представления текстовой информации в виде векторов признаков. Это означает, что классические алгоритмы работы с текстами, как с векторами признаков, и известные методы индексирования текстовой информации не применимы для реализации предложенного метода оценки сходства текстов. Поэтому предложенный в настоящей работе метод оценки сходства текстов требует разработки специализированных структур данных и алгоритмов для эффективного представления и обработки данных. Решению этой задачи посвящена следующая глава настоящего диссертационного исследования.

ГЛАВА 3. МОДЕЛЬ ДАННЫХ, СТРУКТУРЫ ДАННЫХ И АЛГОРИТМЫ РЕШЕНИЯ ПОИСКОВО-АНАЛИТИЧЕСКИХ ЗАДАЧ

В современных ИАС задачи информационного поиска решаются путём обработки коллекций текстовых документов, построения специализированных индексных структур (индексирования) и обращения к этим структурам при поступлении поисковых запросов. Этот принцип позволяет выполнить наиболее вычислительно трудоёмкие операции над текстами документов на этапе индексирования; при этом данные в индексных структурах организуют таким образом, чтобы время ответа на поисковый запрос было минимальным. В случае распределённых поисковых систем реализация вышеуказанного принципа позволяет индексировать десятки миллиардов документов [136] и минимизирует время ответа системы на поисковый запрос.

В первых параграфах настоящей главы представлены разработанные структуры данных, необходимые для представления информации о текстовых документах в ИАС, структуры инвертированных индексов, а также алгоритмы их построения на этапе индексирования документов. Разработанные структуры данных и алгоритмы описываются и формализуются с помощью методов логико-алгебраического моделирования. Созданная таким образом модель данных служит формальным инструментом для исследования свойств разработанных алгоритмов и структур данных.

В заключительных параграфах настоящей главы представлены разработанные структуры данных, необходимые для представления поисковых запросов и алгоритмы оценки релевантности и ранжирования результатов информационного поиска, реализующие метод многокритериальной оценки сходства текстов, описанный в предыдущей главе.

3.1 Структуры данных поисковых индексов

3.1.1 Представление текстовых документов

Для представления информации об отдельных документах в подсистеме полнотекстового поиска ИАС рассмотрим структуру данных, содержащую:

- нетекстовую метаинформацию документов (например, идентификатор документа, дату публикации, коды тематических классов, формат документа и др.)
- индекс текста документа (ИТД), построенный с помощью лингвистического анализа.

ИТД представляет собой последовательность ЭД, в которой каждый ЭД однозначным образом соответствует словоупотреблению в тексте документа. Будем считать, что ЭД в ИТД пронумерованы и следуют друг за другом в том же порядке, в каком следуют в тексте соответственные им словоупотребления. Каждый ЭД содержит следующий набор полей данных:

1. Числовой идентификатор нормальной формы лексемы (ИНФЛ) (word_id), соответствующей словоупотреблению в тексте документа.
2. Числовой код тега метаинформации или гипертекстовой разметки, к которой относится словоупотребление в тексте документа (tag_id).
3. Числовой идентификатор формы (ИФ) словоупотребления в тексте документа (form).
4. Разность между номером словоупотребления, представляемого ЭД, и номером словоупотребления, являющегося главным по отношению к рассматриваемому словоупотреблению в контексте некоторой синтаксической связи. Поле имеет нулевое значение, если у словоупотребления, представляемого ЭД, нет ГС в контексте некоторой синтаксической связи. Выбранный способ хранения информации о синтаксических связях в предложениях подходит для различных моделей синтаксической структуры предложения (например, дерева подчинения [137] и дерева зависимостей [138]).

5. Множество семантических значений (если эти значения приписаны словоупотреблению в предложении) в виде множества числовых идентификаторов.
6. Множество семантических значений, каждое из которых приписано некоторому словоупотреблению, связанному семантическим отношением в предложении со словоупотреблением, представляемым ЭД. Поле задаётся в виде множества числовых идентификаторов. В реляционно-ситуационной модели простого предложения семантические отношения между словоупотреблениями (представляющими множество синтаксем предложения) устанавливаются на основе сопоставления пар их семантических значений [36, 89]. Для решения задач информационного поиска мы будем использовать (и хранить в ЭД) только значения синтаксем, связанных с рассматриваемой синтаксемой, и при том только те значения, которые участвовали в установлении соответствующей семантической связи между синтаксемами (см. рисунок 2.3 – семантические значения связанных синтаксем заключены в прямоугольные рамки с точечной границей).
7. Вес вхождения – lTF в соответствии с формулой (2.6).
8. Признак наличия у вхождения слова зависимых слов в тексте, относящихся к синтаксической категории определения (атрибута).

Предложенные структуры данных представляют лексико-морфологическую, синтаксическую и семантическую информацию текста: в ИТД хранится не только информация, характеризующая отдельные словоупотребления в тексте, но и синтаксические и семантические связи между ними. Заметим, что в реляционно-ситуационной модели семантические значения и связи назначаются не всем синтаксическим структурам, а только именным синтаксемам [139]. Поэтому ЭД, содержащие поля данных с семантическими значениями и связями, присутствуют в ИТД только для таких словоупотреблений, которые входят в состав именных синтаксем, имеющих семантические значения и участвующих в семантических связях. В то же время

у некоторого словоупотребления может быть несколько семантических значений, и оно может участвовать в нескольких семантических связях в контексте некоторого сложного предложения в соответствии с выбранным представлением текстовой информации. Эти факторы необходимо учитывать при проектировании индексных структур данных в ИАС.

3.1.2 Инвертированный поисковый индекс

Для эффективной реализации процедуры поиска разработан ИПИ, ориентированный на хранение информации о текстах масштабных коллекций текстовых документов. Отличительной особенностью разработанного ИПИ является возможность поиска с учётом синтаксической и семантической информации.

ИПИ реализуется в виде хеш-таблицы. В качестве ключа ИПИ выступает ИНФЛ. Каждому ключу сопоставлена последовательность ЭД – пост-лист. Все ЭД в пост-листах имеют одинаковый размер и хранят информацию о словоупотреблениях в текстах документов. Предлагаемый далее подход является развитием подхода, предложенного в работах [116, 117], и отличается от него способом организации структур данных и набором полей данных, включающих информацию о синтаксической и семантической структурах текста.

В предлагаемом подходе к организации ИПИ пост-лист употреблений некоторой лексемы представляет собой массив ЭД различных видов (обозначим их σ-ЭД). Каждый σ-ЭД занимает фиксированный размер в памяти и содержит набор полей данных, различающийся у σ-ЭД в зависимости от вида.

Рассмотрим различные виды σ-ЭД:

1. ЭД метаинформации документа (1-ЭД) содержит следующие поля данных:
    1) вид ЭД (значение поля равно 1);
    2) числовой код (идентификатор) документа в индексной базе, предназначенный для идентификации документа в ИПИ (doc_id);

3) суммарный вес всех вхождений слова в соответствующий документ (вещественное число).

2. σ-ЭД вхождения слова (2-ЭД) содержит информацию, которой характеризуется каждое из словоупотреблений:
   1) вид ЭД (значение поля равно 2);
   2) поля 2–4 и 7, 8 ЭД ИТД.
3. Семантический σ-ЭД (3-ЭД) содержит информацию, которой характеризуются лишь некоторые словоупотребления (семантическая информация, как было отмечено выше, может отсутствовать у отдельных словоупотреблений). Семантический σ-ЭД содержит следующие поля:
   1) вид ЭД (значение поля равно 3);
   2) поля 5, 6 ЭД ИТД.

Рассмотрим более подробно предложенную схему представления информации в ИПИ (рисунок 3.1).

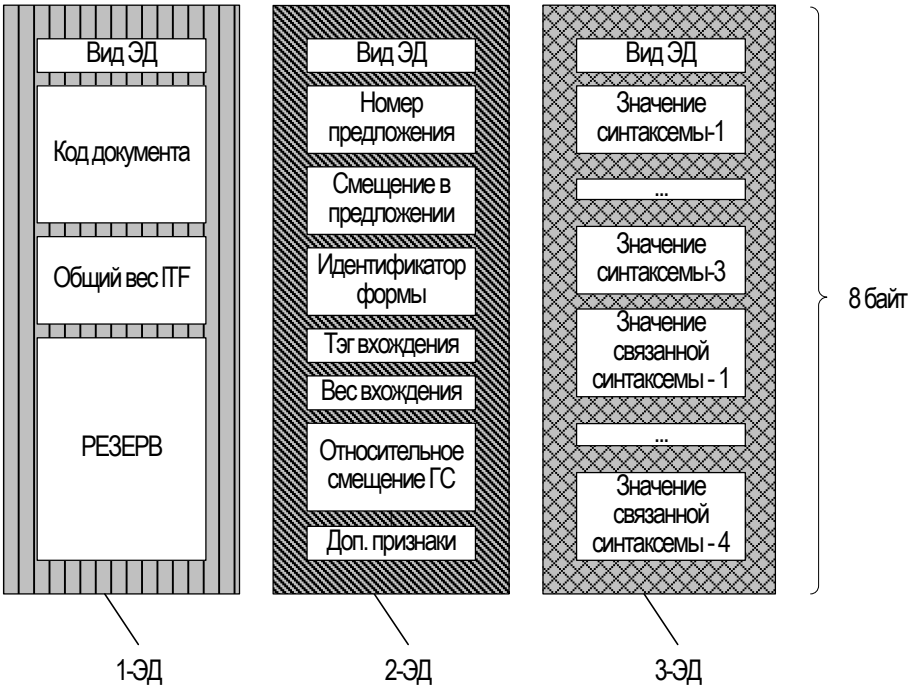

Рисунок 3.1 – Схема элементов данных разработанного инвертированного поискового индекса

Каждый σ-ЭД занимает в памяти ЭВМ 8 байт, что позволяет эффективно реализовать представление σ-последовательностей в памяти ЭВМ в виде массивов. Семантическая информация, связанная со словоупотреблением в

тексте, размещается в 3-ЭД; при этом каждое значение синтаксемы представляется числовым идентификатором, размер которого составляет 7 бит. При таком подходе для каждого словоупотребления сохраняется до 3 семантических значений и до 5 значений семантически связанных словоупотреблений ((*3+5*)·*7=56* бит) в сложном предложении.

По сравнению с классической схемой хранения информации о вхождениях слов, в которой на каждое вхождение приходится отдельный ЭД (рисунок 3.2), содержащий полный набор полей данных, предложенная схема представляет информацию о последовательности употреблений некоторой лексемы в тексте последовательностью σ-ЭД.

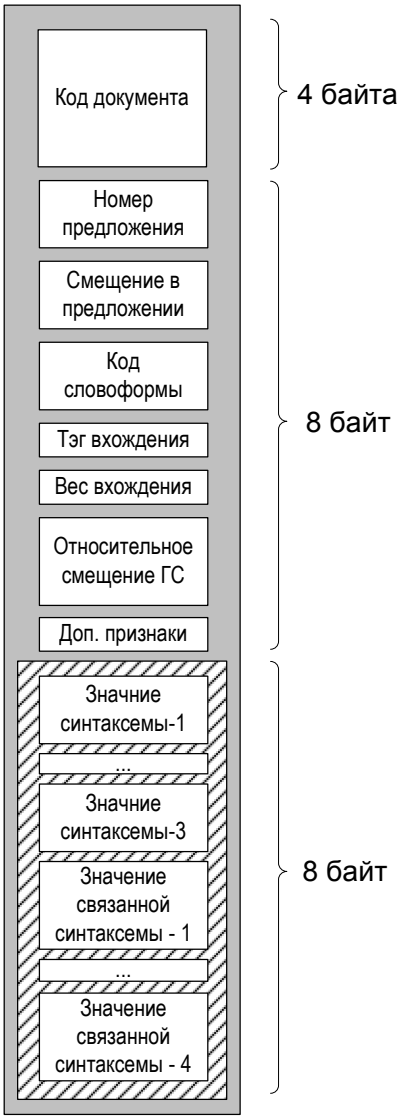

Рисунок 3.2 – Схема элемента данных классического инвертированного поискового индекса

Выбор такого подхода обоснован тем, что семантические значения и связи назначаются не всем словоупотреблениям в тексте. Хранение семантической информации в обычных пост-листах привело бы к тому, что многие поля данных в ЭД этих пост-листов были инициализированы пустыми значениями (значениями «по умолчанию»).

В σ-последовательностях эта информация сохраняется в 2-ЭД, которые могут отсутствовать для тех словоупотреблений, которые не характеризуются семантическими значениями и связями. Как будет показано в следующей главе, на практике это позволяет значительно уменьшить объёмы памяти, требуемой для хранения информации в ИПИ.

Для формального описания разработанной модели данных ИПИ введём следующие определения.

Определение 3.1. Последовательность σ-ЭД, относящихся к употреблениям некоторой лексемы в тексте отдельно взятого документа назовём *σ-цепочкой*; σ-цепочка строится по следующим правилам:

1. В начале цепочки размещается 1-ЭД.
2. За ним следует 2-ЭД, соответствующий первому (по порядку) употреблению лексемы в тексте.
3. За ним опционально следует 3-ЭД, если с рассматриваемым употреблением лексемы связана некоторая семантическая информация.
4. Далее следуют чередующиеся 2-ЭД и 3-ЭД (опционально), соответствующие остальным употреблениям рассматриваемой лексемы в тексте.

В соответствии с этим определением заметим, что σ-цепочка $\sigma = \left\{ m_0, m_1, ..., m_{|\sigma|-1}, \right\}$ является нижней полурешёткой, поскольку она представляет собой частично упорядоченное множество (по построению) и при этом для каждой пары ЭД σ-цепочки ($\forall m', m'' \in \sigma$) определена точная нижняя грань ($\inf \left\{ m', m'' \right\} = m_0$).

Определение 3.2. Для представления информации обо всех употреблениях некоторой лексемы всех документах коллекции применяется последовательность σ-цепочек – *σ-последовательность*. σ-последовательность формируется с помощью конкатенации σ-цепочек по следующему правилу: σ-цепочка σ_1 предшествует σ_2 тогда и только тогда, когда код документа, представляемого σ-цепочкой σ_1 меньше кода документа, представляемого σ-цепочкой σ_2.

Таким образом на классе σ-цепочек индуцируется линейный порядок (*σ-правило-1*): σ-цепочка $\sigma^1 = \left\{m_0^1, m_1^1, ..., m_{|\sigma^1|-1}^1, \right\}$, в начале которой находится 1-ЭД m_0^1 ($m_0^1 = \inf(\sigma^1)$) с кодом документа d_1 ($d_1 = doc\_id(m_0^1)$) предшествует σ-цепочке $\sigma^2 = \left\{m_0^2, m_1^2, ..., m_{|\sigma^2|-1}^2, \right\}$, в начале которой находится 1-ЭД m_0^2 ($m_0^2 = \inf(\sigma^2)$) с кодом документа d_2, тогда и только тогда, когда $d_1 < d_2$.

Аналогичным образом определяется частичный порядок на классе σ-ЭД внутри σ цепочки (*σ-правило-2)*:

1. 2-ЭД $m' \in \sigma$ предшествует другому 2-ЭД $m'' \in \sigma$ тогда и только тогда, когда:
   a. Номер предложения в тексте, в котором содержится словоупотребление, представленное 2-ЭД m', меньше номера предложения в тексте, в котором содержится словоупотребление, представленное 2-ЭД m'': $sent(m') < sent(m'')$.
   b. Вышеуказанные номера предложений совпадают ($sent(m') = sent(m'')$), а смещение (от начала предложения) словоупотребления, представленного 2-ЭД m', меньше смещения словоупотребления, представленного 2- ЭД m'': $sofs(m') < sofs(m'')$.
2. 2-ЭД m' предшествует 3-ЭД $m''' \in \sigma$, содержащему семантическую информацию о том же словоупотреблении, что и 2-ЭД m'.

3. В общем случае 2-ЭД и 3-ЭД, соответствующие разным словоупотреблениям слов, не сравнимы.

В совокупности *σ-правило-1* и *σ-правило-2* вводят отношение частичного порядка на классе σ-ЭД в σ-последовательностях – рисунок 3.3.

Рисунок 3.3 – Пример упорядочения σ-ЭД в σ-последовательностях

Одним из основных алгоритмов, применяемых к σ-последовательностям является алгоритм слияния (merge) [114]. Однако введённое отношение частичного порядка (*σ-правило-2*) не позволяет подвергать слиянию произвольные σ-последовательности. Причина этого состоит в том, что при одновременном просмотре ЭД двух σ-последовательностей слева направо и сравнении «текущих» σ-ЭД возникают ситуации, которые не регламентированы *σ-правилом-2*.

Для устранения этого недостатка введём следующее *σ-правило-3*: из двух сравниваемых σ-ЭД m_1 и m_2, входящих в некоторую σ-последовательность ($m_1, m_2 \in \sigma$), ЭД m_1 предшествует m_2 ($m_1 < m_2$) тогда и только тогда, когда:

1. m_1 является 3-ЭД, а m_2 – 2-ЭД.
2. m_1 является 3-ЭД, а m_2 – 1-ЭД.
3. m_1 является 2-ЭД, а m_2 – 1-ЭД.
4. m_1 и m_2 являются 1-ЭД, и при этом код документа в m_1 меньше кода документа в m_2: $doc\_id(m_1) < doc\_id(m_2)$;
5. m_1 и m_2 являются 2-ЭД, и при этом (аналогично *σ-правилу-2*)
    a. Номер предложения в тексте документа, в котором содержится словоупотребление, представленное 2-ЭД m_1, меньше номера предложения в тексте, содержащего словоупотребление, представленное 2-ЭД m_2: $sent(m_1) < sent(m_2)$;

b. Номера вышеуказанных предложений совпадают ($sent(m_1) = sent(m_2)$), а смещение (от начала предложения) словоупотребления, представленного 2-ЭД m_1, меньше смещения словоупотребления, представленного 2- ЭД m_2: $sofs(m_1) < sofs(m_2)$.

Отметим, что *σ-правило-3* также задаёт отношение частичного порядка на ЭД σ-последовательностей: рассмотрены 5 вариантов сравнения σ-ЭД друг с другом из 6 возможных, причём два 3-ЭД не сравнимы между собой.

Введённое *σ-правило-3* позволяет применять классический алгоритм слияния к σ-последовательностям, а также ряд других важных алгоритмов на этапе индексирования и поиска. Это утверждение будет обосновано в последующих параграфах. Пока же для этой цели сделаем несколько предварительных замечаний.

Определение 3.3. Объект, указывающий на текущий обрабатываемый ЭД в последовательности элементов, назовём итератором.

В простейшем случае итератор – порядковый номер текущего обрабатываемого ЭД в массиве элементов [140]. В дальнейшем для его обозначения будем использовать строчные буквы *i*, *j*. Доступ к ЭД последовательности *X* через итератор будем обозначать «*X*[*i*]» (или для краткости «**i*»), а диапазон обрабатываемых элементов последовательности через *i=X.begin*(); *j=X.end*(). Запись *i=i+1* будет означать перемещение итератора к следующему элементу.

Определим далее класс алгоритмов *S*, обладающих следующими свойствами. Всякий алгоритм *A* класса *S* по определению:

1. Обрабатывает две σ-последовательности слева направо в цикле (инициализируя итераторы по каждой σ-последовательности в их начале).
2. Использует *σ-правило-3* в качестве отношения порядка для сравнения σ-ЭД.
3. Сравнивает σ-ЭД друг с другом только для перемещения итераторов по σ-последовательностям.

4. В ходе своей работы перемещает к следующему ЭД тот итератор, который указывает на меньший ЭД из двух сравниваемых (в смысле *σ-правила-3*) – рисунок 3.4.
5. Завершает свою работу, если хотя бы один из двух итераторов (по первой или по второй σ-последовательности) достигает конца соответствующей σ-последовательности.

Схематический пример обработки σ-последовательностей алгоритмом класса *S* приведён на рисунке 3.4.

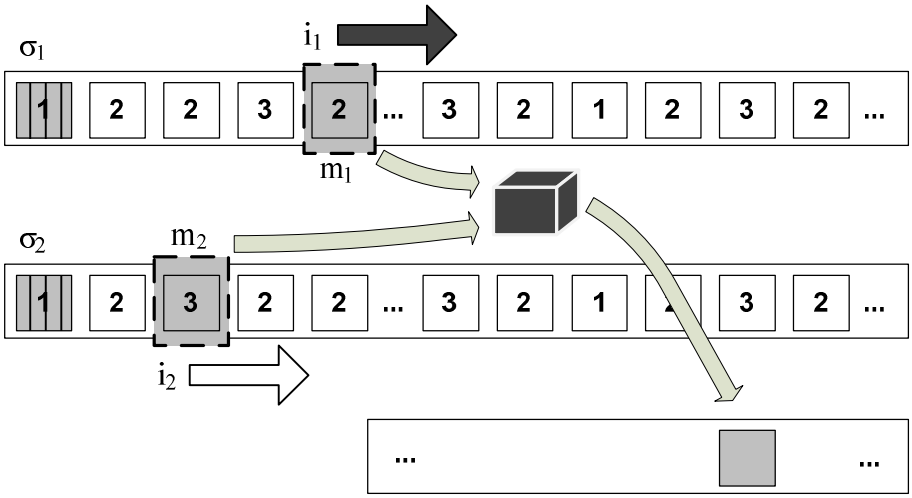

Рисунок 3.4 – Пример обработки σ-последовательностей алгоритмом класса *S*

Заметим, что к классу алгоритмов *S* относится, в частности, алгоритм слияния σ-последовательностей.

Для всякого алгоритма *A* класса *S* справедливо следующее утверждение.

Утверждение 3.1. При обработке σ-последовательностей σ_1 и σ_2 любым алгоритмом *A* класса *S* не происходит сравнения 3-ЭД между собой.

Доказательство. Предположим противоположное. Пусть в какой-то момент (такт *t*) работы алгоритма *A* из указанного класса *S* итераторы i_1 и i_2 по σ-последовательностям σ_1 и σ_2 указывают на ЭД m_1 и m_2 соответственно ($m_1 = *i_1 \in \sigma_1$, $m_2 = *i_2 \in \sigma_2$). Пусть m_1 и m_2 оба являются 3-ЭД. Тогда они находятся не в начале σ-последовательностей σ_1 и σ_2 (поскольку в противном случае последовательности σ_1 и σ_2 не удовлетворяли бы определению σ-

последовательности, что противоречит условию). Это означает, что итераторы i_1 и i_2 были перемещены в ходе предыдущих шагов (тактов) работы алгоритма A, т.е. $t \neq t_0 \,\&\, t_0 < t-1 < t$. Тогда на предыдущем шаге (t-1) работы алгоритма был перемещён, например, итератор по $\sigma_1 - i_1$. Это означает, что на шаге (t-1) итератор по σ_1 указывал на (i_1-1), и сравнивались σ-ЭД m=*(i_1-1) и m_2. По определению σ-последовательности предыдущий элемент – m является 2-ЭД (т.к. перед 3-ЭД всегда присутствует 2-ЭД). Это означает, что на шаге (t-1) работы алгоритма A итератор последовательности σ_1 указывал на 2-ЭД (m), предшествующий 3-ЭД (m_1). Значит, на шаге (t-1) сравнивались 2-ЭД и 3-ЭД. Но по п.1 *σ-правила-3* 3-ЭД предшествует 2-ЭД, т.е. $m_2<m$. По п.4 определения алгоритмов класса S в этом случае должен был быть перемещён итератор по σ_2, т.е. i_2. Причём в этом случае по определению σ-последовательности на шаге t итератор i_2 указывал бы на следующий за 3-ЭД 2-ЭД или 1-ЭД), и неопределённое сравнение также не потребовалось. Полученное противоречие доказывает утверждение.

Утверждение 3.1 доказано.

Заметим, что существует альтернативное конструктивное доказательство утверждения 3.1, основанное на построении конечного автомата (с 6 состояниями), моделирующего всевозможные переходы между состояниями, которые соответствуют парам сравниваемых σ-ЭД. При этом при подаче на вход всевозможных цепочек элементов, удовлетворяющих определению σ-последовательности, с учётом *σ-правила-3* и ограничения п.4 определения алгоритмов класса S, автомат никогда не перейдёт в состояние, в котором требуется сравнение двух 3-ЭД друг с другом.

Как будет показано в следующем разделе, введённые отношения частичных порядков позволяют реализовать алгоритмы построения σ-последовательностей для представления ИТД (используя *σ-правило-2)* и слияния σ-последовательностей (с помощью *σ-правила-3*). Поскольку эти алгоритмы относятся к классу S, в соответствии с утверждением 3.1 в них не

возникает ситуации (не определённой в *σ-правиле-2* и *σ-правиле-3*), требующей сравнения двух 3-ЭД между собой.

3.2 Алгоритмы формирования поисковых индексов

Процедура формирования поисковых индексов заключается в лингвистическом анализе текстов, составляющих коллекцию, построении ИТД и помещении ИТД в ИПИ. На этапе лингвистического анализа автоматически устанавливается информация, связанная со словоупотреблениями. Построенные ИТД передаются в систему наполнения ИПИ. Поскольку текст документа может содержать значительное количество различных лексем (от нескольких сотен до десятков тысяч), добавление ИТД к ИПИ требует обновления большого количества σ-последовательностей, соответствующих этим лексемам.

Обновление σ-последовательности состоит в слиянии исходных данных σ-последовательности с σ-цепочкой, содержащей информацию о вхождениях слова в текст обрабатываемого документа. В простейшем случае, если документы не удаляются из ИПИ, их коды строго возрастают, и ИПИ целиком размещается в оперативной памяти сервера ИАС, то слияние заключается в конкатенации исходной σ-последовательности с добавляемой σ-цепочкой (путём дописывания σ-цепочки в конец σ-последовательности). Эта операция корректна (не нарушает порядка в σ-последовательности) и на практике достаточно эффективна, если реализовать σ-последовательность как массив ЭД с предварительным резервированием.

В более сложном случае, ИПИ размещается в файловой системе и представляет собой хранилище данных «ключ-значение» (key-value storage) [141]. При этом может потребоваться удалять документы из ИПИ (например, по причине их устаревания). В этом случае алгоритм обновления включает в себя стадии считывания с диска и восстановления σ-последовательности из сжатого образа, слияния с σ-цепочкой, сжатия и записи на диск. Эти операции достаточно трудоёмки и при больших объёмах информации в ИПИ могут

выполняться несколько секунд. Общее время обработки одного ИТД может достигать десятков минут. Поэтому для эффективной обработки поступающих ИТД выполняется буферизация в оперативной памяти.

Структуру данных для промежуточного накопления ИТД назовём буфером ξ-последовательностей. Буфер ξ-последовательностей представляет собой массив ξ-последовательностей. Каждый ЭД ξ-последовательности (ξ-ЭД) представляет собой пару *<ИНФЛ, σ-ЭД>* – рисунок 3.5. ЭД в ξ-последовательности частично упорядочены (*ξ-правило*): ξ-ЭД-1 предшествует ξ-ЭД-2 тогда и только тогда, когда:

1. Идентификатор лексемы ξ-ЭД-1 меньше идентификатора лексемы ξ-ЭД-2.
2. Идентификаторы лексем ξ-ЭД-1 и ξ-ЭД-2 совпадают, а σ-ЭД из ξ-ЭД-1 предшествует σ-ЭД из ξ-ЭД-2 (в смысле *σ-правила-3*).

Рисунок 3.5 – Пример упорядочения ЭД в ξ-последовательностях

Поскольку ξ-ЭД отличается от σ-ЭД только наличием ИНФЛ в качестве дополнительного поля, то с учётом ξ-правила заключаем, что всякая ξ-последовательность может рассматриваться как линейно упорядоченное множество σ-последовательностей.

При добавлении нового документа в буфер ξ-последовательностей ИТД сортируется в соответствии со следующим правилом: ЭД ИТД μ_1 предшествует μ_2 тогда и только тогда, когда:

1. ИНФЛ μ_1 меньше ИНФЛ μ_1 ($word\_id(\mu_1) < word\_id(\mu_2)$);
2. ИНФЛ μ_1 совпадает с ИНФЛ μ_1 ($word\_id(\mu_1) = word\_id(\mu_2)$), но при этом выполнено:

a. Номер предложения в тексте, в которое входит словоупотребление, представленное ЭД μ_1, меньше номера предложения в тексте, в которое входит словоупотребление, представленное ЭД μ_2: $sent(\mu_1) < sent(\mu_2)$.
b. При совпадении вышеупомянутых номеров предложений $sent(\mu_1) = sent(\mu_1)$ μ_1 смещение (от начала предложения) словоупотребления, представляемого μ_1, меньше смещения словоупотребления, представляемого μ_2: $sofs(\mu_1) < sofs(\mu_2)$.

На следующем этапе отсортированный ИТД преобразуется в ξ-последовательность. Этак процедура заключается в последовательном просмотре ИТД слева направо, формированию и добавлению пар <*ИНФЛ*, *2-ЭД*>, <*ИНФЛ*, *3-ЭД*>, а также вставке пар <*ИНФЛ*, *1-ЭД*> при нарушении условия равенства $word\_id(\mu_{prev}) < word\_id(\mu_{current})$ перед ЭД $\mu_{current}$. Пара <*ИНФЛ*, *3-ЭД*> добавляется только в том случае, если для соответствующего словоупотребления установлены семантические значения или связи.

В результате формируются σ-подпоследовательности в составе ξ-последовательности. Для ИТД, содержащего N ЭД, представляющих информацию о словоупотреблениях в тексте некоторого документа, в предположении, что в документе содержится M различных лексем, длина результирующей ξ-последовательности составляет не более $(2 \cdot N + M)$ ЭД.

При добавлении полученной ξ-последовательности в буфер, она подвергается слиянию с ξ-последовательностью из буфера, имеющей наименьший размер. Процедура слияния двух произвольных ξ-последовательностей выполняется с помощью классического алгоритма merge [114] (блок-схема которого приведена на рисунке 3.6) с отношением порядка (<), введённым ξ-правилом. Таким образом, подобно алгоритмам класса *S*, обеспечивается корректность применения алгоритма слияния к ξ-последовательностям. Отметим, что результирующая последовательность также является ξ-последовательностью, поскольку в двух ξ-последовательностях, подвергаемых процедуре слияния, отсутствуют

конфликтные ЭД (т.е. такие ЭД μ_1, μ_2, что при применении алгоритма из класса S они являются эквивалентными, т.е. для них одновременно не выполнено $\mu_1 < \mu_2$ и $\mu_2 < \mu_1$ в смысле выбранного отношения порядка).

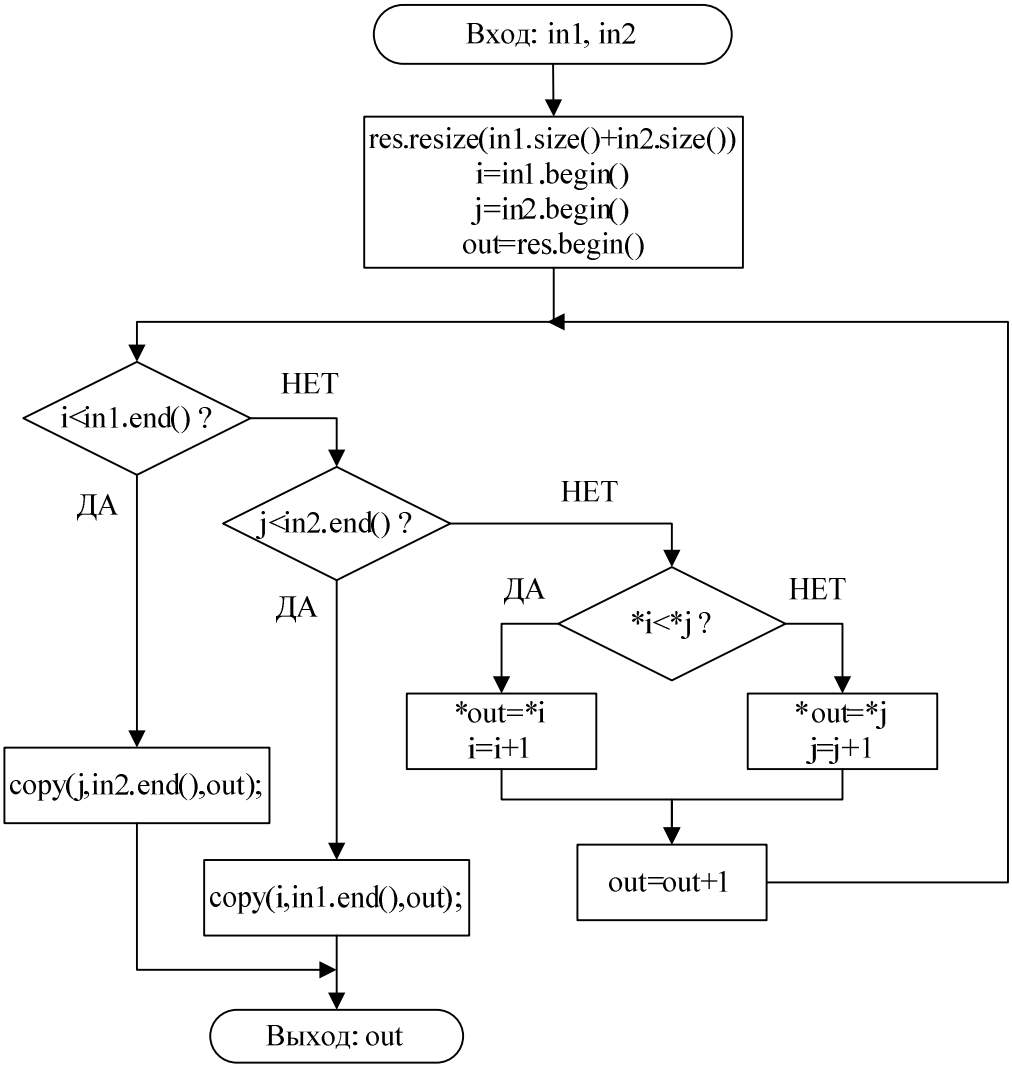

Рисунок 3.6 – Блок-схема алгоритма слияния последовательностей

Оценим сложность алгоритма помещения ИТД в буфер.

Утверждение 3.2. Оценка сложности алгоритма помещения ИТД в буфер ИПИ относительно количества словоупотреблений (N), количества различных лексем в тексте (M) и количества ЭД в самой короткой ξ-последовательности в буфере (K) составляет не более $O(N \cdot log^2(N) + N + M + K)$.

Доказательство. Алгоритм помещения ИТД в буфер ИПИ содержит 3 последовательные стадии:

- Сортировка. Используется алгоритм quicksort, реализация которого имеет сложность, не более $O(N \cdot log^2(N))$ в худшем случае ($O(N \cdot log(N))$ в среднем) [142].
- Преобразование ИТД в ξ-последовательность. Алгоритм последовательно просматривает ИТД и даёт в результате ξ-последовательность длины не более $(2 \cdot N + M)$ ЭД. Оценка сложности равна $O(N)$.
- Слияние ξ-последовательностей. Используется классический алгоритм слияния сортированных массивов (см. рисунок 3.6), имеющий линейную сложность относительно длин входных последовательностей. В силу этого оценка сложности составляет: $O(2xN + M + K)$.

Суммируя оценки сложности алгоритмов, получаем, что общая оценка сложности алгоритма помещения ИТД в буфер ИПИ не превышает $O(N \cdot log^2(N) + N + M + K)$.

Утверждение 3.2 доказано.

Если после обработки очередного ИТД количество ЭД в буфере ξ-последовательностей превышает максимально заданное значение, выполняется объединение данных буфера с ИПИ. Для этого сначала выполняется слияние всех ξ-последовательностей в одну с помощью модифицированного алгоритма сбалансированного n-путевого слияния на основе кучи (n-way merge) [143] с отношением порядка, заданным *ξ-правилом*. Затем с помощью последовательного прохода по полученной ξ-последовательности в ней выделяются σ-последовательности (см. рисунок 3.5), которые подвергаются слиянию с σ-последовательностями из ИПИ с соответственными ИНФЛ. Процедура слияния выполняется с помощью классического алгоритма merge с *σ-правилом-3* в качестве отношения порядка, как это было описано в заключительной части параграфа 3.1.

При необходимости из σ-последовательности, полученной из ИПИ, могут быть исключены ЭД, соответствующие документам, коды которых присутствуют в списке удалённых документов. При этом удаляются все σ-цепочки, соответствующие удалённым документам. Предполагается, что

список кодов удалённых документов отсортирован по возрастанию кодов. Блок-схема алгоритма фильтрации σ-последовательности путём удаления σ-цепочек представлена на рисунке 3.7. Указанная процедура фильтрации позволяет исключить из σ-последовательностей конфликтные элементы, поэтому в результате их слияния получается σ-последовательность.

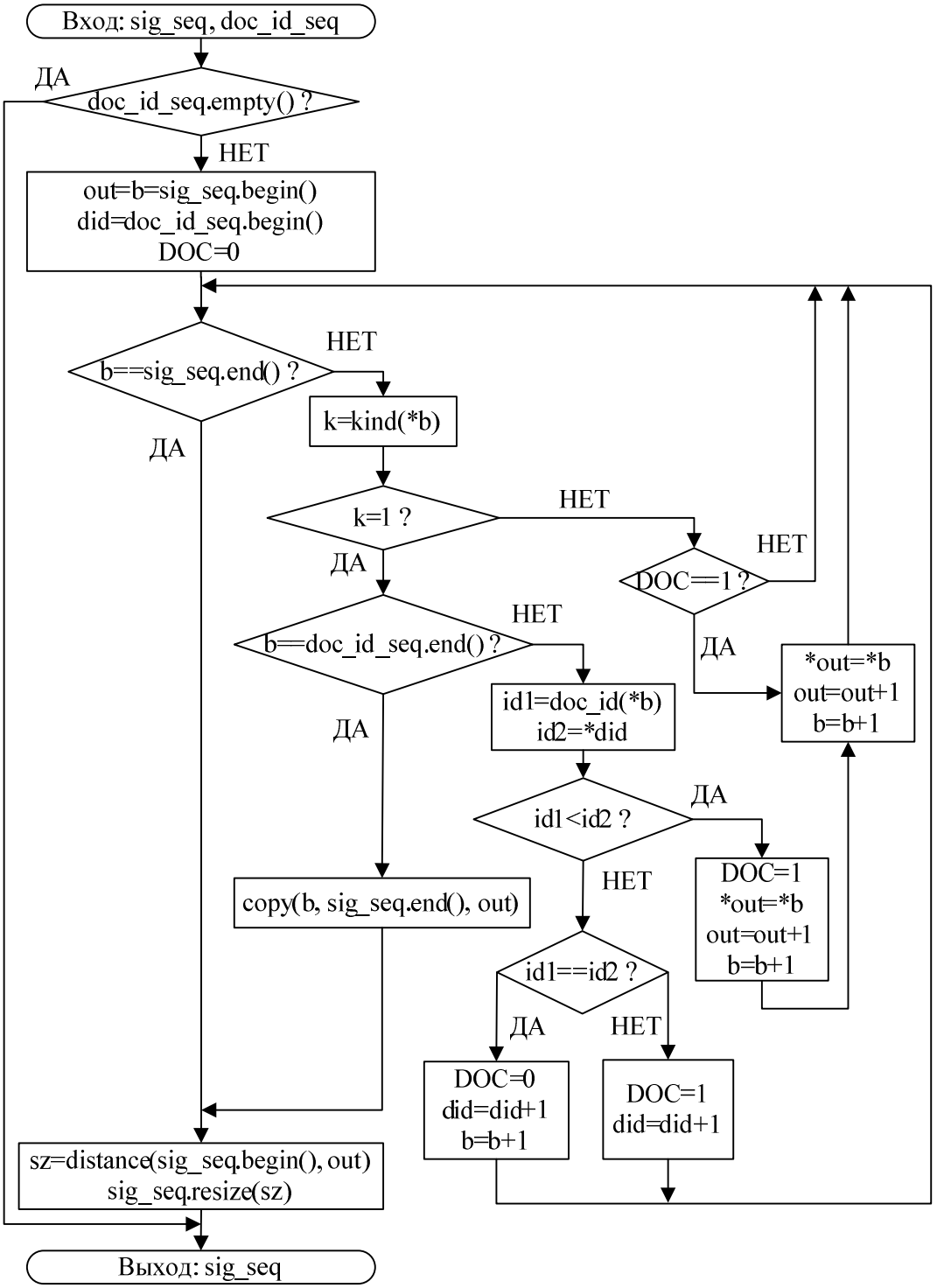

Рисунок 3.7 – Блок-схема алгоритма фильтрации σ-последовательности

Утверждение 3.3. Алгоритм фильтрации σ-последовательности имеет оценку сложности $O(M + D))$, где M – длина обрабатываемой σ-последовательности, D – длина списка кодов удаляемых документов.

Доказательство. Обработка входной σ-последовательности и упорядоченного списка кодов удаляемых документов аналогична алгоритму слияния двух упорядоченных массивов. За счёт перемещения итератора по списку кодов удаляемых документов будет обработано, как минимум, D ЭД σ-последовательности (что потребует в худшем случае не более 14 операций на каждый обрабатываемый σ-ЭД – см. рисунок 3.7). Оставшаяся часть σ-последовательности будет скопирована: $(M - D)$ ЭД. Операция копирования имеет линейную сложность относительно длины входных данных.

В худшем случае обработка потребует M перемещений итераторов по σ-последовательности и D перемещений итераторов по списку кодов удаляемых документов (не более 14 операций в теле цикла на одно перемещение каждого из итераторов – см. рисунок 3.7).

Таким образом, оценка сложности алгоритма фильтрации σ-последовательности не хуже $O(M + D)$.

Утверждение 3.3 доказано.

Утверждение 3.4. Алгоритм обновления ИПИ имеет оценку сложности $O(K \cdot log(N) + P \cdot (L + D + M))$, где

- K – количество ξ-последовательностей в буфере,
- N – количество ξ-ЭД в буфере,
- P – количество различных лексем с различными ИНФЛ в ИПИ,
- L – средняя длина σ-последовательности в ИПИ,
- D – длина списка кодов удаляемых документов,
- M – средняя длина σ-последовательности в буфере.

Доказательство. Для шагов представленного алгоритма обновления ИПИ оценка сложности составляет:

- n-путевое слияние: $O(K \cdot log(N))$;

– фильтрация σ-последовательностей: $O(P \cdot (L + D))$;
– слияние σ-последовательностей: $O(P \cdot (M + L))$.

Суммируя оценки сложности алгоритмов, получаем, что общая оценка сложности алгоритма обновления ИПИ не превышает $O(K \cdot log(N) + P(L + D + M))$.

Утверждение 3.4 доказано.

Отметим, что разработанный подход к наполнению ИПИ позволяет реализовать параллельную обработку ИТД (на вычислительных ядрах процессоров на одном вычислительном сервере в составе распределённой ИАС).

Таким образом, в параграфах 3.1, 3.2 описаны алгоритмы, структуры и модель данных для решения задачи индексирования текстов и построения ИПИ. В следующих параграфах рассматриваются разработанные алгоритмы и структуры данных, применяемые на этапе поиска информации с использованием ИПИ.

3.3 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. 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 лингвистическому анализу, в результате которого строится индекс текста запроса (ИТЗ). В ИТЗ каждому словоупотреблению текста запроса соответствует ЭД, содержащий следующий набор полей данных:

1. Числовой ИНФЛ, соответствующей словоупотреблению (word_id).
2. Числовой идентификатор тега метаинформации, которым помечено словоупотребление в запросе (tag_id). Словоупотребления, которые должны присутствовать в заголовке искомых документов помечаются соответствующим тегом. Аналогичным образом помечаются слова, относящиеся к ФИО авторов и другим текстовым полям метаданных. Предполагается, что разбиение текста запроса на предложения или фразы происходит внутри соответствующих полей метаданных, т.е. словоупотребления в составе одного предложения запроса не могут относиться к нескольким различным молям метаданных. Словоупотребления запроса также могут быть помечены произвольными тегами языка гипертекстовой разметки (например, для поиска документов, которые содержат слова запроса в гипертекстовых ссылках на другие документы).
3. Числовой ИФ словоупотребления в тексте запроса (form).
4. Порядковый номер словоупотребления в тексте запроса (word_no). Как было отмечено в главе 2, в случае коротких запросов в ходе лингвистического анализа может быть не снята омонимия нормальных форм. Для учёта омонимов, если у них различаются нормальные формы, в ИТЗ добавляются «дублирующие» ЭД с одинаковым порядковым номером, отличающиеся только ИНФЛ и, возможно, ИФ.
5. Порядковый номер ЭД в ИТЗ.

6. Порядковый номер предложения в тексте запроса, содержащего словоупотребление.
7. Порядковый номер ЭД, представляющего словоупотребление в тексте запроса, которое является главным по отношению к рассматриваемому словоупотреблению в смысле синтаксической связи. Если словоупотребление не зависит от других словоупотреблений в составе предложения, то поле имеет пустое значение.
8. Множество семантических значений (если эти значения приписаны словоупотреблению в тексте запроса).
9. Множество семантических значений синтаксем, связанных семантической связью в тексте запроса с рассматриваемым словоупотреблением в том же предложении.
10. Порядковый номер фразы, в которую входит словоупотрбление (concept_id). Под фразой здесь понимается совокупность слов запроса в пределах одного предложения, объединённых с помощью фигурных скобок «{…}» (см. главу 2).
11. Числовой идентификатор фразы (phrase_id), который позволяет при поиске объединять фразы, связанные в запросе тезаурусными отношениями (например, при пополнении исходного запроса терминами из тезауруса). Этот идентификатор приписывается каждому ЭД, и на его основе соотносятся вхождения слов, относящиеся к различным фразам.
12. 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»).
13. Служебные флаги.

Служебные флаги представляют информацию о помеченных (с помощью языка запросов – см. раздел 2.3.1) словоупотреблениях в поисковом запросе и могут содержать следующие значения *приоритета слова*:

- Исключаемое словоупотребление (t0) – найденные документы не должны содержать словоупотреблений соответствующей лексемы (возможно, с учётом конкретной словоформы) в тексте (либо в метаданных документов, если исключаемое словоупотребление помечено соответствующим тегом).
- Исключаемое словоупотребление предложения (t1) – найденные документы не должны содержать словоупотреблений соответствующей лексемы (возможно, с учётом конкретной словоформы) в тех предложениях текстов, которые релевантны остальному запросу (либо в метаданных документов, если исключаемое словоупотребление помечено соответствующим тегом).
- Обязательное словоупотребление (t2) – должно присутствовать в текстах найденных документов (возможно, с учётом конкретной формы этого слова) либо в метаданных документов, если словоупотребление помечено соответствующим тегом.
- Обычное словоупотребление (t3) – может отсутствовать или присутствовать в текстах найденных документов (при многословных запросах).
- Факультативное словоупотребление (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 словоупотребление 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.

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

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

Рассмотрим далее разработанные алгоритмы оценки релевантности и ранжирования результатов информационного поиска по запросу на ЕЯ.

3.4 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 каждый из этапов работы алгоритма.

3.4.1 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. Числовой идентификатор сайта документа (site_id) или другого ресурса, из которого был загружен документ.
2. Числовой идентификатор имени документа (doc_id), определяющий имя документа на сайте.
3. Числовой 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 – рисунок 3.8.

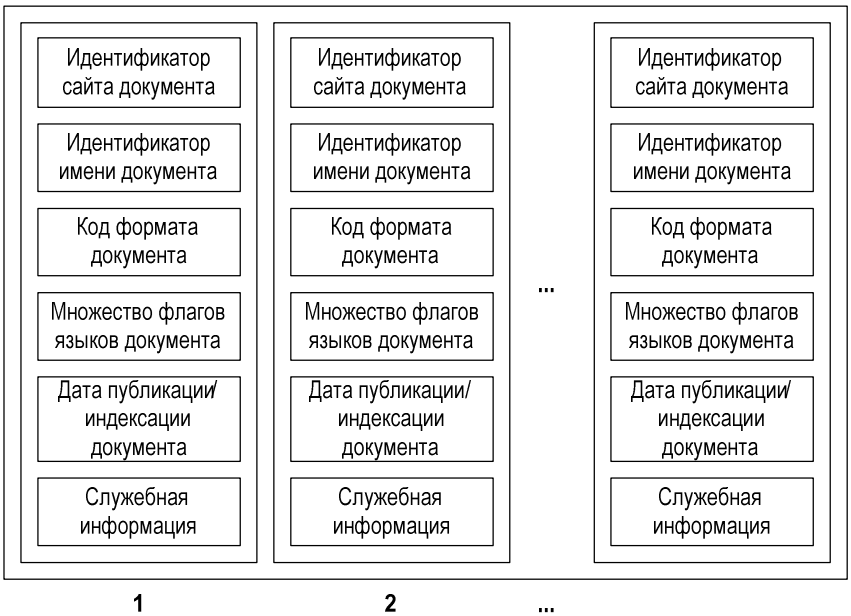

Рисунок 3.8 – Структура реестра метаданных документов

Путём последовательного просмотра реестра метаданных осуществляется сопоставление содержащейся в нём информации с зоной поиска. В результате формируется массив кодов документов, удовлетворяющих условиям поиска по

нетекстовым метаданным. Этот массив представляет область поиска. Сложность его построения – линейна по числу проиндексированных документов в коллекции.

3.4.2 Выборка и фильтрация информации

Выборка информации из ИПИ (по ИНФЛ в качестве ключа) позволяет получить информацию о словоупотреблениях в текстах коллекции документов в виде σ-последовательностей. Выбранные σ-последовательности подвергаются первичной фильтрации: из σ-последовательностей исключаются те σ-ЭД (образующие σ-цепочки), которые относятся к документам, не вошедшим в область поиска. Блок-схема алгоритма первичной фильтрации представлена на рисунке 3.9. 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. При применении алгоритма первичной фильтрации σ-последовательностей предполагается, что область поиска (doc_id_seq) непустая. В противном случае в коллекции текстов отсутствуют документы, удовлетворяющие зоне поиска, и поиск завершён.

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

Утверждение 3.5. Алгоритм первичной фильтрации σ-последовательностей имеет оценку сложности $O(B+C)$, где B – длина входной σ-последовательности, C – количество документов, входящих в область поиска.

Доказательство. Обработка входной σ-последовательности и упорядоченного списка кодов документов, входящих в область поиска, аналогична алгоритму слияния двух упорядоченных массивов. За счёт перемещения итератора по списку кодов документов, входящих в область поиска, будет обработано, как минимум, C ЭД σ-последовательности (что потребует в худшем случае не более 16 операций на каждый обрабатываемый

σ-ЭД – см. рисунок 3.9). Оставшаяся часть σ-последовательности будет скопирована: (*B* - *C*) ЭД. Операция копирования имеет линейную сложность относительно длины входных данных.

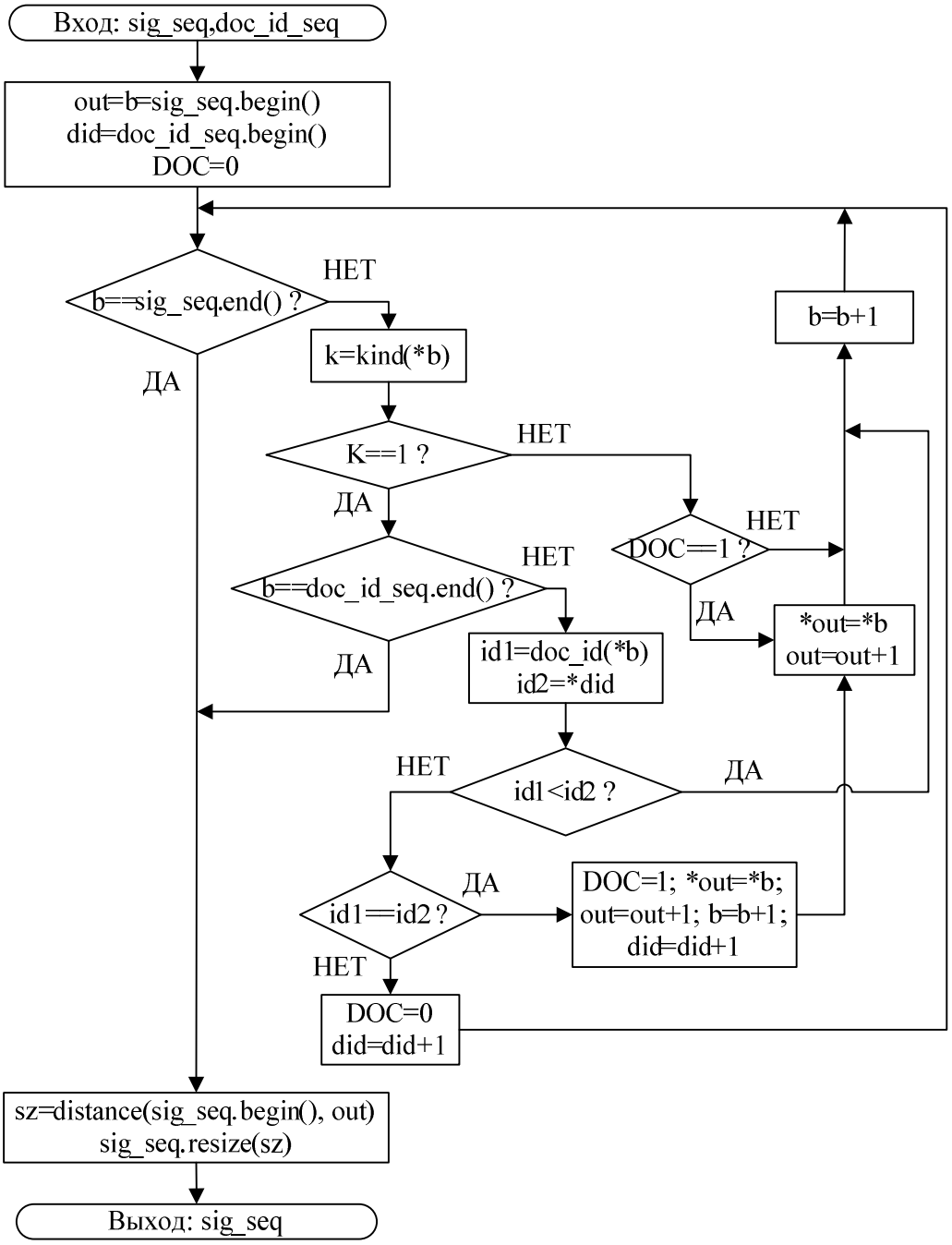

Рисунок 3.9 – Блок-схема алгоритма первичной фильтрации σ-последовательностей

В худшем случае обработка потребует *B* перемещений итераторов по σ-последовательности и *C* перемещений итераторов по списку кодов документов (не более 16 операций в теле цикла на одно перемещение каждого из итераторов – см. рисунок 3.9).

Таким образом, оценка сложности алгоритма фильтрации σ-последовательности не хуже $O(B + C)$.

Утверждение 3.5 доказано.

Заметим, что если область поиска содержит больше половины документов коллекции, то оптимальным с точки зрения вычислительной сложности будет следующий алгоритм первичной фильтрации (за счёт уменьшения размерности входных данных – списка кодов документов):

1. «Инвертировать» область поиска: на основе реестра метаданных построить массив кодов документов, не удовлетворяющих зоне поиска, – исключаемые документы.
2. Применить алгоритм фильтрации σ-последовательностей из п.3.2, удаляющий из σ-последовательности σ-ЭД, соответствующие исключаемым документам (см. рисунок 3.7).

Следующим этапом после выборки и первичной фильтрации информации является фильтрация σ-последовательностей по словоупотреблениям 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, рассчитанным по формуле (2.6): $v(w) \le v_{\min}$. 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:
    a. У слова в тексте запроса и в тексте документа не совпадает тег разметки метаданных (tag_id).

b. 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 словоупотреблению запроса представлена на рисунке 3.10.

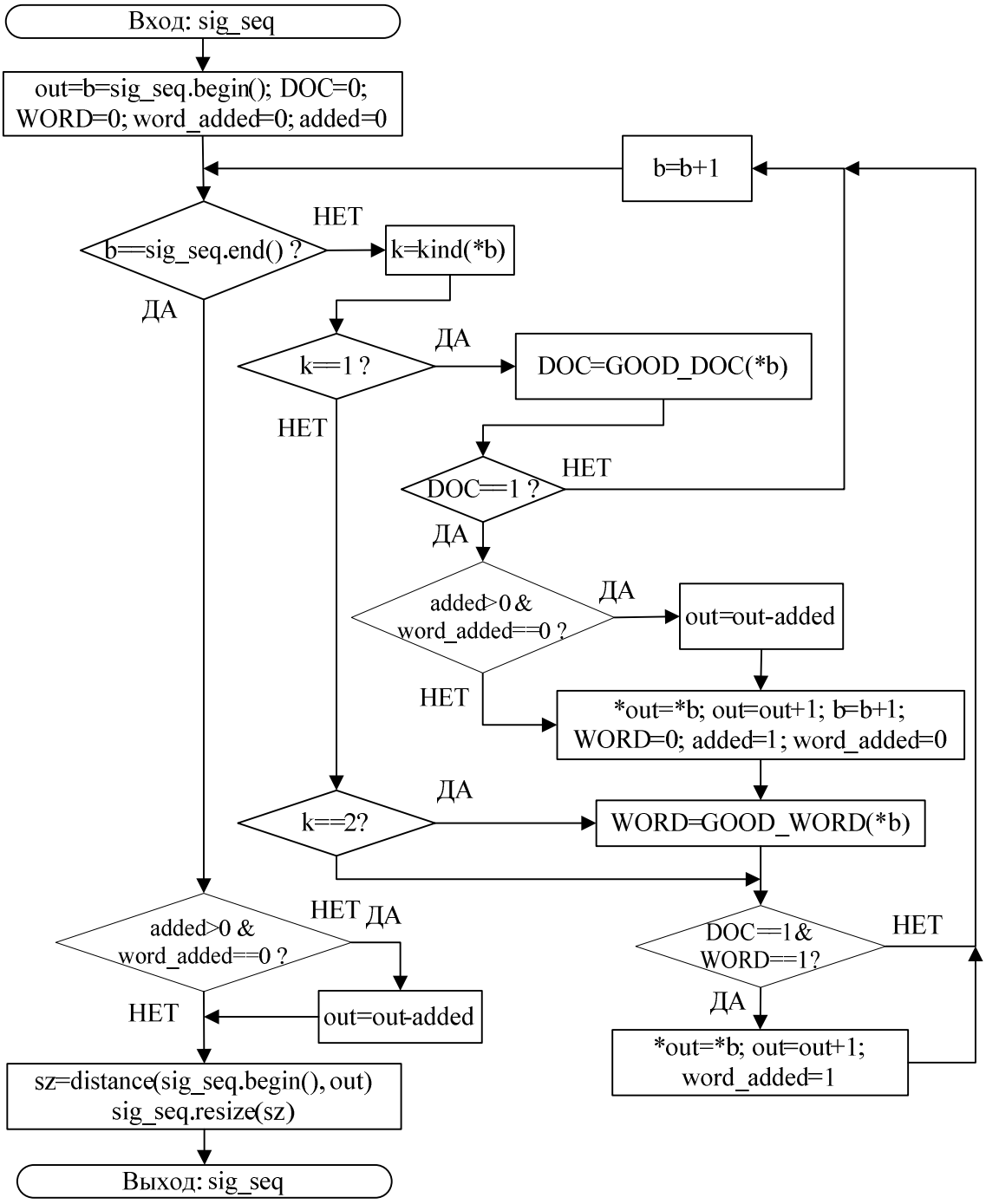

Рисунок 3.10 – Блок-схема алгоритма фильтрации σ-последовательности по словоупотреблению запроса

Утверждение 3.6. Оценка сложности алгоритма фильтрации σ-последовательности по словоупотреблению запроса линейна относительно размера исходной σ-последовательности.

Доказательство. Пусть N – количество ЭД в обрабатываемой σ-последовательности. Алгоритм фильтрации σ-последовательности по словоупотреблению запроса реализует цикл обработки ЭД. Тело цикла будет выполнено в точности N раз. В худшем случае в теле цикла будет выполнено не более 21 операции (см. рисунок 3.10).

Таким образом, в среднем, оценка сложности алгоритма фильтрации σ-последовательности по словоупотреблению запроса составляет $O(N)$.

Утверждение 3.6 доказано.

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, причём предложения могут относиться к различным типам метаданных и содержать повторяющиеся употребления некоторой лексемы.

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.

3.4.3 Предварительная оценка соответствия найденных документов словоупотреблениям запроса

Предварительная оценка соответствия найденных документов словоупотреблениям запроса заключается в преобразовании каждой σ-последовательности в соответствующую ρ-последовательность.

Определение 3.4. ρ-последовательность – массив ρ-ЭД, содержащих следующий набор полей данных:

1. Код документа (doc_id).
2. Номер предложения, в которое входит найденное словоупотребления в тексте документа (sent_no).
3. Порядковый номер словоупотребления в предложении найденного документа (sofs).
4. Порядковый номер ЭД в ИТЗ, соответствующего σ-последовательности (word_idx).
5. Разность между позицией словоупотребления, представляемого ЭД, и позицией словоупотребления, являющегося главным (в смысле синтаксической связи) по отношению к рассматриваемому словоупотреблению в тексте найденного документа (mwp).
6. Номер предложения в ИТЗ, в которое входит словоупотребление, соответствующее найденной σ-последовательности (rsent).
7. Вес найденного словоупотребления в тексте документа (tf).
8. Флаг, совпадения формы найденного словоупотребления в тексте документа с формой соответствующего словоупотребления 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 (2.9) и (2.10) (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. Поле 8 вычисляется вычисляются на основе сравнения словоформ словоупотребления текста запроса и словоупотребления найденного текста, представленных ЭД ИТЗ и σ-ЭД. Поле 9 вычисляется на основе оценки сходства множеств значений синтаксем и семантических связей, в 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 оценки сходства двух МСЗ в условиях выбранной схемы представления семантической информации составляет не более: 3+3=6 и 5+5=10 итераций цикла при расчёте величин roles_w и rels_w соответственно. Алгоритм сравнения двух МСЗ приведён на рисунке 3.11. Его сложность мы будем полагать константной.

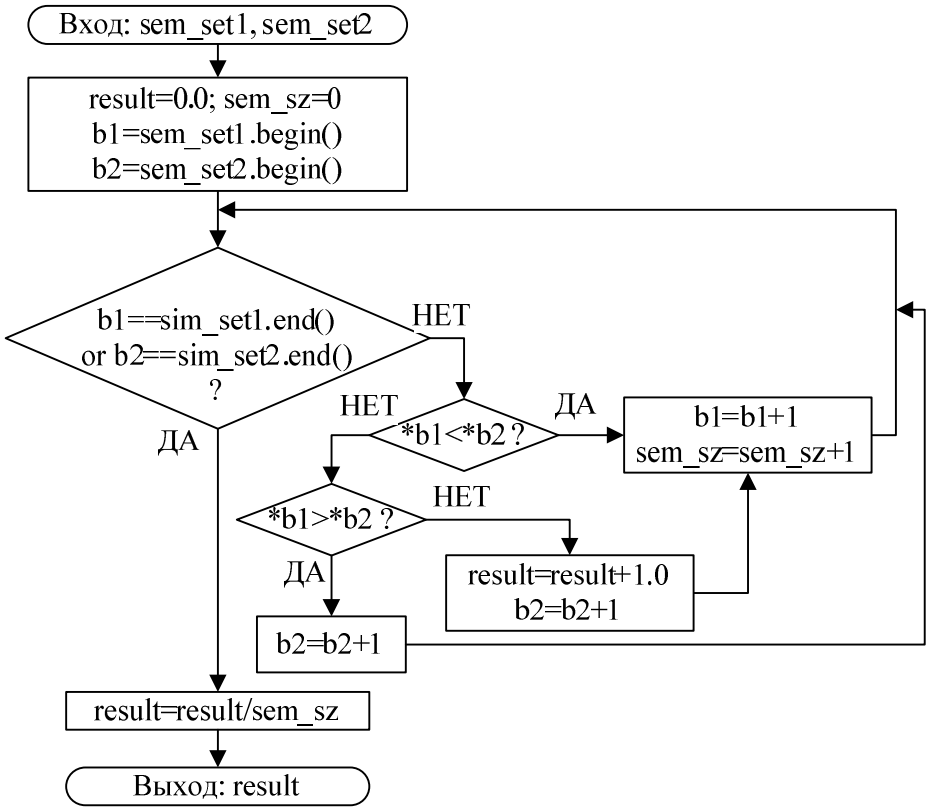

Рисунок 3.11 – Алгоритм оценки сходства двух множеств семантических значений

Алгоритм преобразования σ-последовательности в соответствующую ρ-последовательность линеен (рисунок 3.12), количество ρ-ЭД в результирующей ρ-последовательности не превосходит количества σ-ЭД в исходной σ-последовательности за вычетом единицы. В результате работы алгоритма каждому словоупотреблению в запросе (представляемому ЭД в ИТЗ) сопоставляется ρ-последовательность, содержащая сведения об встречаемости лексемы, соответствующей указанному словоупотреблению, в текстах коллекции документов.

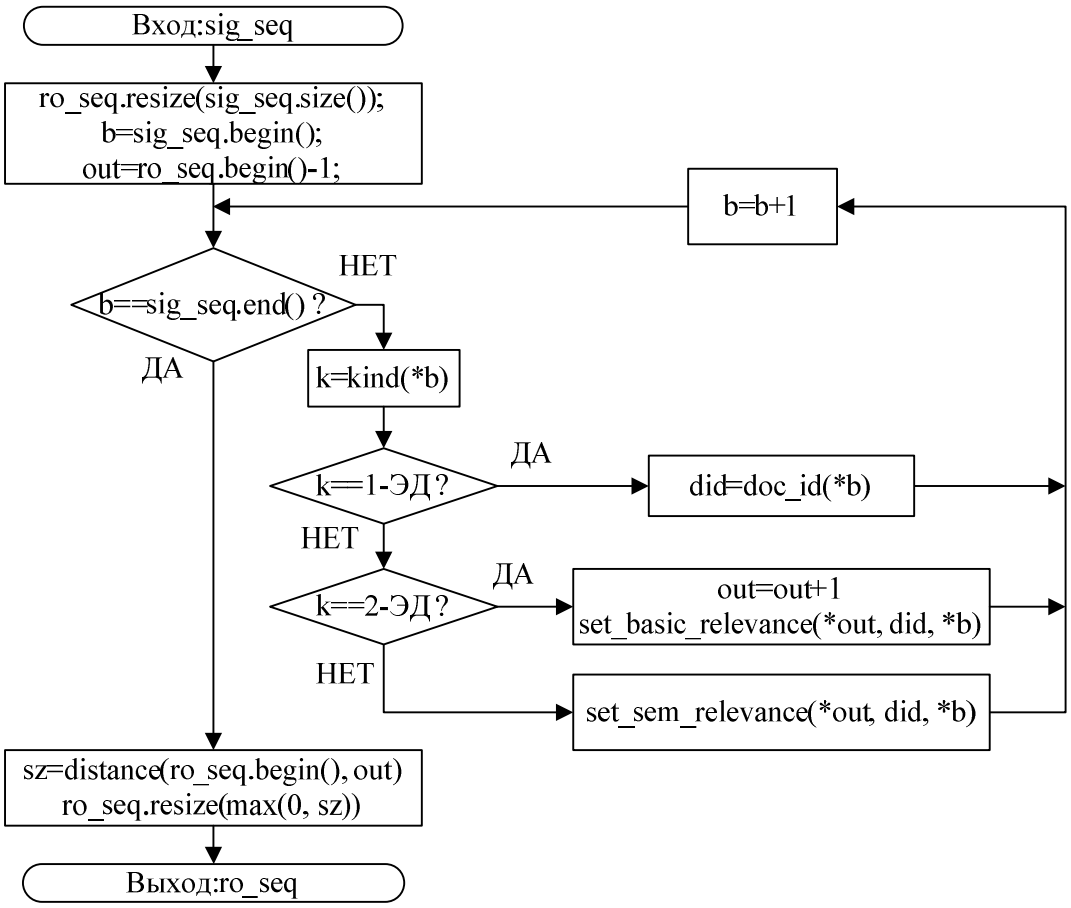

Рисунок 3.12 – Алгоритм преобразования σ-последовательности в ρ-последовательность

Утверждение 3.7. Оценка сложности алгоритма преобразования σ-последовательности в ρ-последовательность линейна относительно размера исходной σ-последовательности.

Доказательство. Пусть N – количество ЭД в обрабатываемой σ-последовательности. Алгоритм преобразования σ-последовательности в ρ-последовательность реализует цикл обработки ЭД. Тело цикла будет выполнено в точности N раз. В худшем случае в теле цикла будет выполнено не более 7 операций (см. рисунок 3.12).

Таким образом, в среднем, оценка сложности алгоритма преобразования σ-последовательности в ρ-последовательность составляет $O(N)$.

Утверждение 3.7 доказано.

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 текстового 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 текстового документа, представляемого ρ-ЭД r_2: $doc\_id(r_1)=doc\_id(r_2)$, и при этом:

2.1 Номер предложения в тексте запроса, в которое входит словоупотребление найденного текста, представляемое ρ-ЭД r_1, меньше номера предложения в тексте запроса, в которое входит словоупотребление найденного текста, представляемое ρ-ЭД r_2: $rsent(r_1)<rsent(r_2)$.

2.2 Номера предложений в тексте запроса, в которые входят словоупотребления найденного текста, представляемые 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, в которое входит найденное словоупотребление, представляемое ρ-ЭД r_1, меньше номера предложения, в которое входит найденное словоупотребление, представляемое ρ-ЭД r_2: $sent\_no(r_1)<sent\_no(r_2)$.

2.2.2 Номера предложений в тексте документа, в которые входят найденные словоупотребления, представляемые 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 в найденном тексте словоупотребления, представляемого ρ-ЭД r_1, меньше смещения от начала предложения в найденном тексте словоупотребления, 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 словоупотребление, на основе нормальной формы которого найден

соответствующий ρ-ЭД, затем – по позиции словоупотребления в найденном тексте. Такой порядок является естественным при преобразовании σ-последовательности в ρ-последовательность, поскольку в соответствии с ним упорядочены σ-цепочки в σ-последовательности, а также σ-ЭД в σ-цепочках. Введённые по сравнению с σ-правилом-3 условия упорядочения 2.1, 2.2 по полю rsent обеспечивают группировку найденных словоупотреблений по предложениям запроса, содержащим словоупотребления запроса, по которым эти словоупотребления текстов документов были найдены.

Поскольку ЭД ρ-последовательностей ρ_1, и ρ_2 ($\rho_1 \neq \rho_2$) соответствуют различным словоупотреблениям в текстах документов (найденным по различным словоупотреблениям текста запроса), то на практике можно считать, что для ρ-последовательностей ρ_1, и ρ_2 выполнено следующее условие:

$$\rho_1 \cap \rho_2 = \varnothing . \qquad (3.1)$$

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 даёт в результате обычное объединение двух ρ-последовательностей с сохранением отношения линейного порядка: $join(\rho_1, \rho_2) = \rho_1 \cup \rho_2$. Вычислительная сложность этого алгоритма линейна относительно размеров ρ_1 и ρ_2 (сложность слияния двух линейно-упорядоченных множеств) и составляет $O(|\rho_1| + |\rho_2|)$.

Операция условного слияния ρ-последовательностей ρ_1, и ρ_2 позволяет получить ρ-последовательность ρ_3, содержащую только те элементы исходных ρ-последовательностей, которые удовлетворяют некоторому условию:

$\rho_3 = cond\_join(\rho_1, \rho_2) =$

$\{r \in \rho_1 \mid \exists r' \in \rho_2 : P(r, r') = 1\} \cup \{r \in \rho_2 \mid \exists r' \in \rho_1 : P(r', r) = 1\}$, где $P(r_1,r_2)$ некоторый двухместный предикат на ρ-ЭД. Эта операция является обобщённым аналогом операции пересечения множеств. В отличие от операции пересечения двух множеств (результирующее множество входят те и только те элементы, которые принадлежали каждому из пересекаемых множеств), в операции условного слияния ρ-последовательностей ρ_1, и ρ_2 в результирующую ρ-последовательность входят только такие ρ-ЭД r_1 *и* r_2 из ρ_1, и ρ_2, на которых предикат $P(r_1,r_2)$ принимает истинное значение.

Утверждение 3.8. С учётом ограничения, заданного формулой (3.1), операция условного слияния ρ-последовательностей симметрична тогда и только тогда, когда предикат $P(r_1,r_2)$ симметричен.

Доказательство.

А) Необходимость. Пусть $cond\_join(\rho_1, \rho_2) = cond\_join(\rho_2, \rho_1)$. Тогда по определению имеет место равенство:

$$\begin{aligned} &\{r \in \rho_1 \mid \exists r' \in \rho_2 : P(r, r') = 1\} \cup \{r \in \rho_2 \mid \exists r' \in \rho_1 : P(r', r) = 1\} = \\ &\{r \in \rho_2 \mid \exists r' \in \rho_1 : P(r, r') = 1\} \cup \{r \in \rho_1 \mid \exists r' \in \rho_2 : P(r', r) = 1\}. \end{aligned} \tag{3.2}$$

Или, вводя для краткости обозначения аргументов, входящих в объединения в формуле (3.2): $A \cup B = C \cup D$.

. Поскольку для исходных ρ-последовательностей ρ_1, и ρ_2 выполнено условие (3.1), то также верно, что $A \cap B = \varnothing$ и $C \cap D = \varnothing$. С учётом этого, сравнивая аргументы операции объединения в формуле (3.2), заключаем, что $A = D$ и $B = C$, т.е. $\{r \in \rho_1 \mid \exists r' \in \rho_2 : P(r, r') = 1\} = \{r \in \rho_1 \mid \exists r' \in \rho_2 : P(r', r) = 1\}$ и $\{r \in \rho_2 \mid \exists r' \in \rho_1 : P(r', r) = 1\} = \{r \in \rho_2 \mid \exists r' \in \rho_1 : P(r, r') = 1\}$. Откуда заключаем, что $P(r', r) = P(r, r')$.

Б) Достаточность. Пусть $P(r', r) = P(r, r')$. Тогда в силу симметричности предиката $P(r,r')$ получаем

$cond\_join(\rho_1,\rho_2)=\{r\in\rho_1\mid\exists r'\in\rho_2:P(r,r')=1\}\cup\{r\in\rho_2\mid\exists r'\in\rho_1:P(r',r)=1\}=$ $\{r\in\rho_1\mid\exists r'\in\rho_2:P(r',r)=1\}\cup\{r\in\rho_2\mid\exists r'\in\rho_1:P(r,r')=1\}$. Меняя местами аргументы в операции объедания множеств, заключаем, что $cond\_join(\rho_1,\rho_2)=cond\_join(\rho_2,\rho_1)$.

Утверждение 3.8 доказано.

С помощью введённой операции условного слияния ρ-последовательностей из них синтезируется новая ρ-последовательность, ЭД которой попарно связаны отношением, заданным с помощью предиката $P(r_1,r_2)$. Например, если требуется, чтобы найденные словоупотребления в документах коллекции встречались в пределах одного предложения, то в качестве предиката $P(\rho_1,\rho_2)$ можно использовать следующий предикат: $P_1(\rho_1,\rho_2)=(doc\_id(r_1)=doc\_id(r_2)\ \&\ rsent(r_1)=rsent(r_2)\ \&\ sent\_no(r_1)=sent\_no(r_2))$.

Операция условного слияния ρ-последовательностей реализуется с помощью алгоритма, блок-схема которого приведена на рисунке 3.13.

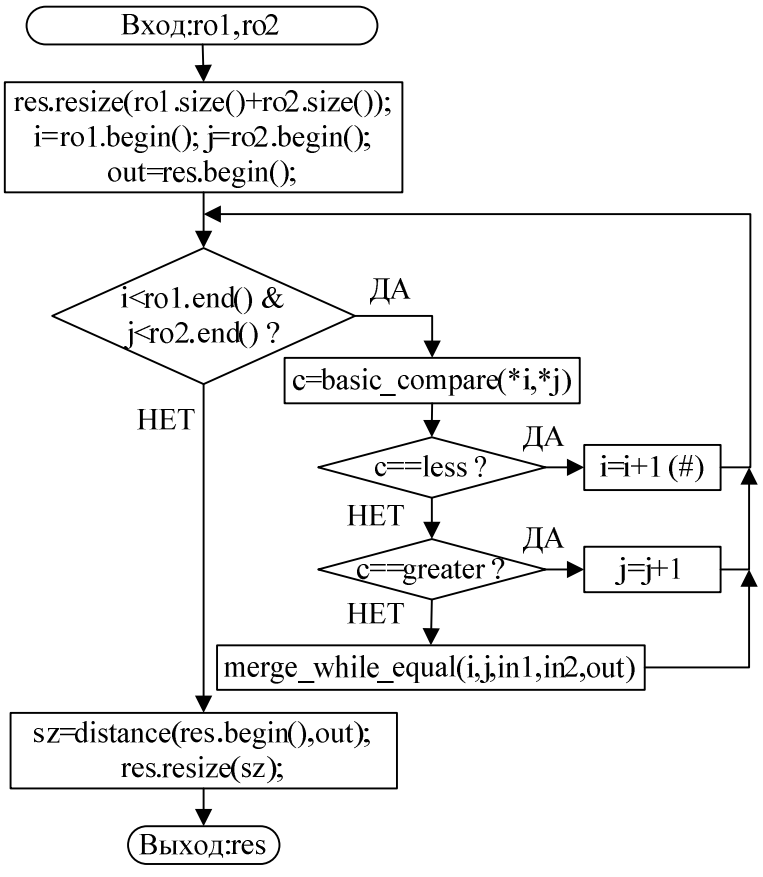

Рисунок 3.13 – Блок-схема алгоритма условного слияния ρ-последовательностей с предикатом P_1

В качестве операции *basic_compare* в этом алгоритме используются условия 1–2.2.2 ρ-*правила* (кроме 2.2.2.1): сравнение ЭД по doc_id, при их равенстве – по rsent, а при равенстве последних – по sent_no. Алгоритм *merge_while_equal* (блок-схема приведена на рисунке 3.14) осуществляет слияние частей ρ-последовательностей, удовлетворяющих заданному условию: в качестве условия *compare* выступает ρ-*правило* целиком, а в качестве предиката *is_equal* – $P(\rho_1,\rho_2)$. Блок-схема алгоритма *copy_equal_tail*, выполняющего копирование ρ-ЭД, эквивалентных в смысле заданного условия, приведена на рисунке 3.15.

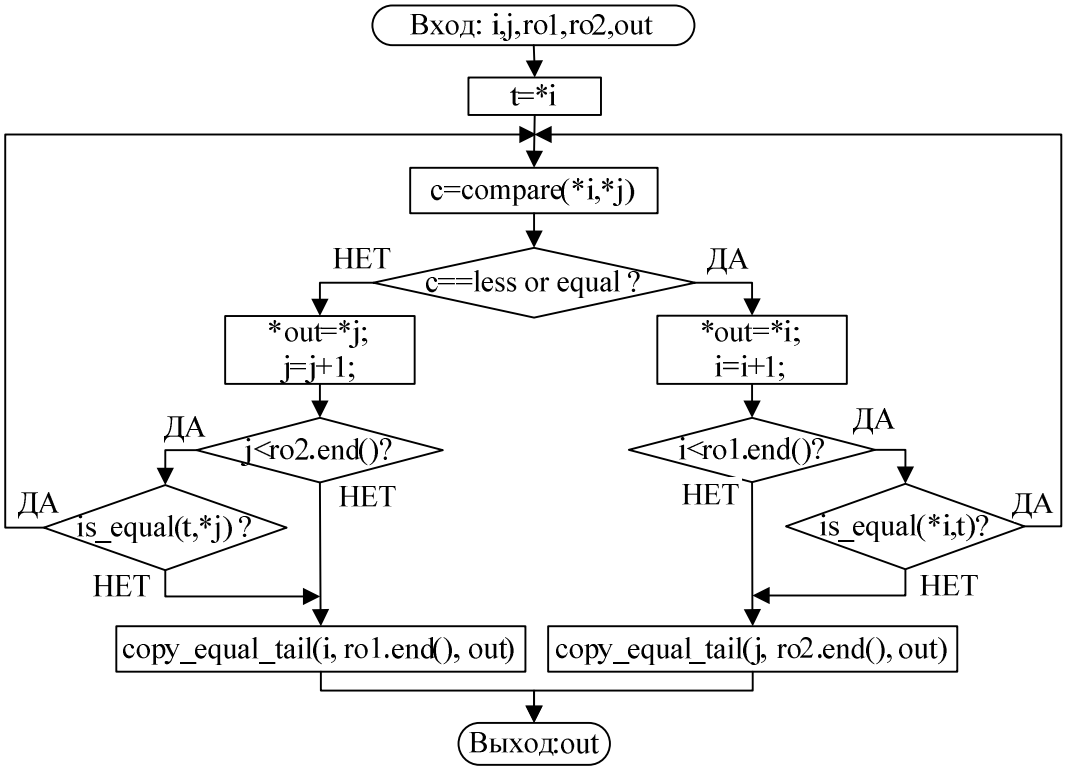

Рисунок 3.14 – Блок-схема алгоритма слияния частей ρ-последовательностей, удовлетворяющих заданному условию

Если требуется, чтобы какие-либо словоупотребления запроса обязательно присутствовали в документах совместно (но не обязательно в пределах одного предложения), то в вышеприведённых 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: $P_2(\rho_1,\rho_2)=(doc\_id(r_1)=doc\_id(r_2))$.

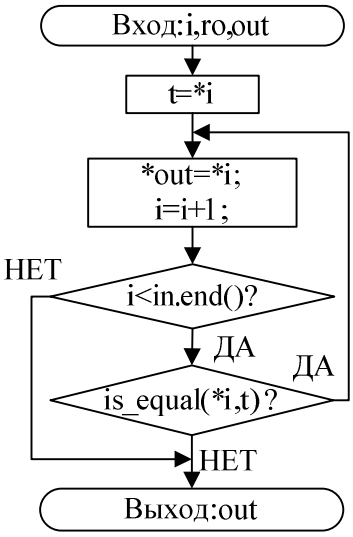

Рисунок 3.15 – Блок-схема алгоритма копирования ρ-ЭД, эквивалентных в смысле заданного условия

Утверждение 3.9. Вычислительная сложность алгоритма условного слияния ρ-последовательностей линейно зависит от размеров входных последовательностей ρ_1 и ρ_2 и составляет в худшем случае не более $O(|\rho_1| + |\rho_2|)$.

Доказательство. Алгоритм условного слияния ρ-последовательностей реализует цикл обработки ЭД двух ρ-последовательностей, перемещая итераторы *i* и *j* по первой и второй ρ-последовательности соответственно. Всего возможно не более $|\rho_1| + |\rho_2|$ вызовов тела цикла. В самом деле, если внутри основного цикла алгоритм *merge_while_equal*, выполняющий слияние частей ρ-последовательностей, удовлетворяющих заданному условию, не вызывается ни разу, то все $|\rho_1| + |\rho_2|$ итераций придутся на первые две ветви алгоритма (см. рисунок 3.13). Если же в каких-то случаях алгоритм *merge_while_equal* был вызван, то в ходе обработки входных данных он также реализует цикл обработки ЭД двух ρ-последовательностей, перемещая те же самые итераторы *i* и *j*. В теле цикла (см. рисунок 3.14) на одну итерацию выполняется не более 6 операций (в худшем случае). При этом обработка линейна, и всего возможно не более $|\rho_1| + |\rho_2|$ итераций. Если в завершение алгоритма *merge_while_equal* вызван алгоритм *copy_equal_tail*, выполняющего копирование ρ-ЭД, эквивалентных в смысле заданного условия, то обработка остаётся линейной, т.к. алгоритм *copy_equal_tail* копирует не более $|\rho_1|$ или $|\rho_2|$ ЭД (в зависимости

от исходных данных) – см. рисунок 3.15. При этом соответственно, перемещаются итераторы *i* или *j*, а в теле цикла в ходе каждой итерации выполняется не более 4 операций (в худшем случае). Это означает, что, каким бы образом не вёлся процесс обработки входных ρ-последовательностей ρ_1 и ρ_2, всего будет выполнено не более $|\rho_1| + |\rho_2|$ итераций, некоторым образом распределённых между тремя циклами (в зависимости от выполнения условий и ветвлений в алгоритмах). При этом в худшем случае в ходе каждой итерации будет выполнено не более 6 операций. Таким образом, вычислительная сложность алгоритма условного слияния ρ-последовательностей составляет в худшем случае не более $O(|\rho_1| + |\rho_2|)$.

Утверждение 3.9 доказано.

Операция одностороннего условного слияния ρ-последовательностей ρ_1 и ρ_2 позволяет пополнить ρ-последовательность ρ_1 удовлетворяющими некоторым условиям ЭД из ρ_2: $\rho_3 = cond\_left\_join(\rho_1, \rho_2) = \rho_1 \cup \{r \in \rho_2 \mid \exists r' \in \rho_1 : P(r', r) = 1\}$. Очевидно, что в общем случае эта операция несимметрична.

Операция одностороннего условного слияния ρ-последовательностей реализуется аналогично алгоритму условного соединения ρ-последовательностей, с той лишь разницей, что на шаге, помеченным знаком (#), выполняются следующие 3 операции (блок-схема на рисунке 3.13): **out=*i; out=out+1; i=i+1*. Вычислительная сложность в этом случае также линейно зависит от размеров входных последовательностей ρ_1 и ρ_2 и составляет $O(|\rho_1| + |\rho_2|)$.

Операция условного исключения элементов ρ-последовательности ρ_2 из ρ_1 позволяет получить ρ-последовательность ρ_3, ЭД которой удовлетворяют следующему условию $\rho_3 = \{r_1 \in \rho_1 \mid \neg\exists r_2 \in \rho_2 : \quad P(r_1, r_2) = 1\}$, где $P(r_1, r_2)$ – некоторый предикат на ρ-ЭД. Блок-схема алгоритма условного исключения элементов ρ-последовательности ρ_2 из ρ_1 приведена на рисунке 3.16. Результатом работы алгоритма является преобразованная ρ-последовательность

ρ_1. В качестве предиката $P(r_1,r_2)$ могут выступать предикаты P_1 или P_2, введённые выше. В этих случаях в качестве операции *compare* используется соответственно:

- условия 1–2.2.2 ρ-правила (кроме 2.2.2.1) для P1: сравнение ЭД по doc_id, при их равенстве – по rsent, а при равенстве последних – по sent_no;
- условие 1 ρ-правила для P2: сравнение ЭД по doc_id.

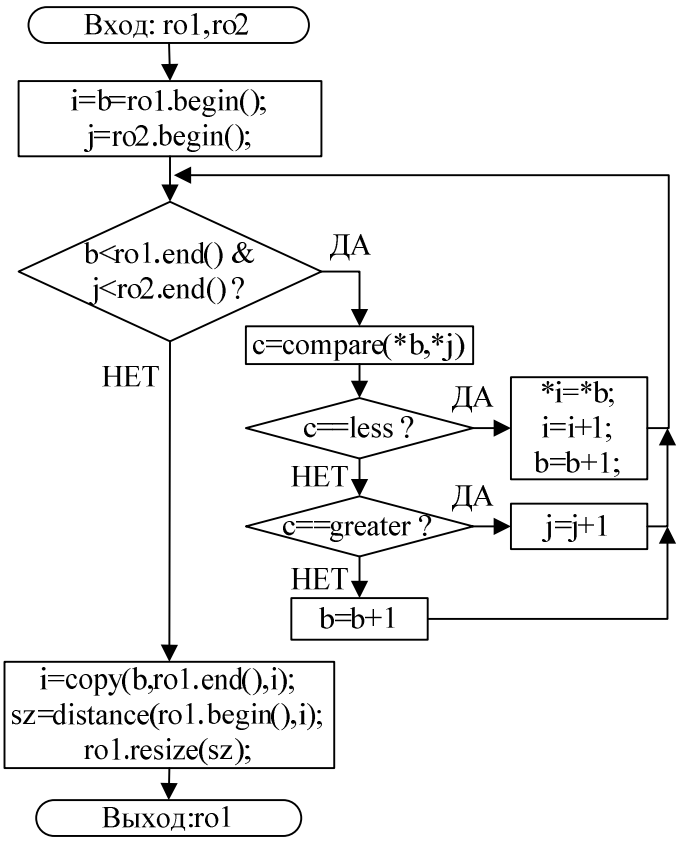

Рисунок 3.16 – Блок-схема алгоритма условного исключения элементов ρ-последовательностей

Утверждение 3.10. Вычислительная сложность алгоритма условного исключения элементов ρ-последовательности ρ_2 из ρ_1 линейно зависит от размеров входных последовательностей ρ_1 и ρ_2 и составляет в худшем случае не более $O(|\rho_1| + |\rho_2|)$.

Доказательство. Алгоритм условного исключения элементов ρ-последовательности ρ_2 из ρ_1 реализует цикл обработки ЭД двух ρ-последовательностей, перемещая итераторы i и j по первой и второй ρ-последовательности соответственно. Всего возможно не более $|\rho_1| + |\rho_2|$ вызовов

тела цикла. На одну итерацию выполняется не более 7 операций (в худшем случае) – см. рисунок 3.16. Это означает, что оценка вычислительной сложности алгоритма составляет в худшем случае не более $O(|\rho_1| + |\rho_2|)$.

Утверждение 3.10 доказано.

С помощью введённых операций фразовый поиск реализуется следующим образом. К ρ-последовательностям $\rho_1,\ldots,\rho_Q$, которые соответствуют словоупотреблениям запроса, составляющим некоторую фразу, попарно применяется операция *cond_join* с предикатом P_1: $\rho = cond\_join(\rho_Q, cond\_join(\ldots, cond\_join(\rho_2, \rho_1)\ldots))$. Затем над полученной ρ-последовательностью выполняется алгоритм фразовой фильтрации, который проверяет наличие соответствующих синтаксических связей между найденными вхождениями слов в тексты документов – рисунок 3.17.

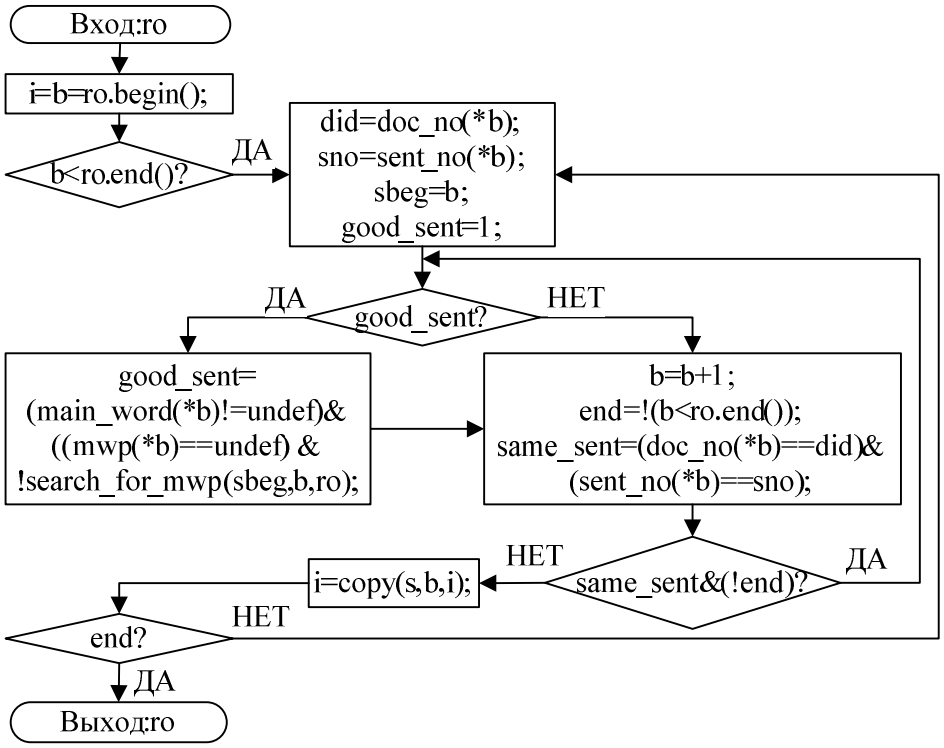

Рисунок 3.17 – Блок-схема алгоритма фразовой фильтрации ρ-последовательности

В этом алгоритме (а также на более поздних стадиях обработки – при оценке релевантности предложений) используется алгоритм поиска синтаксических связей в ρ-последовательности – рисунок 3.18. Он состоит в поиске в пределах предложения документа такого ρ-ЭД r_1, что этот r_1 связан

синтаксически с некоторым другим ρ-ЭД r_2, при условии, что словоупотребления, соответствующие r_1 и r_2 в запросе, также синтаксически связаны.

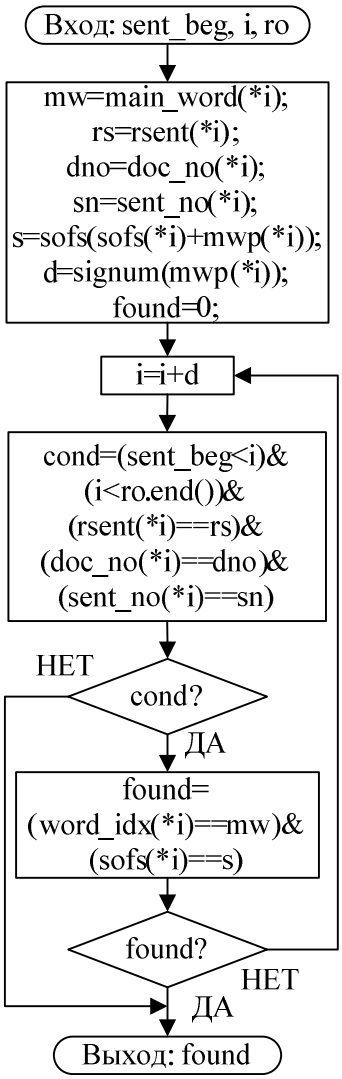

Рисунок 3.18 – Блок-схема алгоритма поиска синтаксических связей в ρ-последовательности

Утверждение 3.11. Оценка вычислительной сложности алгоритма поиска синтаксических связей в ρ-последовательности в пределах одного предложения найденного текста составляет $O(B)$, где B – количество словоупотреблений в составе фразы в поисковом запросе, для которой проверяется наличие синтаксических связей.

Доказательство. Алгоритм поиска синтаксических связей в ρ-последовательности в пределах одного предложения найденного текста

реализует цикл обработки ЭД ρ-последовательности. Условиями выхода из цикла являются:

- обнаружение синтаксической связи – условие *found*;
- переход на ρ-ЭД, относящиеся к другому предложению (текущего или следующего документа) – условие *cond*;
- достижение конца ρ-последовательности.

Худший случай входных данных для алгоритма возникает, если все слова, составляющие фразу, находятся в некотором предложении найденного текста, однако между ними отсутствуют синтаксические связи, т.е. они не образуют фразу. В этом случае будет выполнено не более B итераций цикла, в ходе каждой из которых будет выполнено не более 10 операций (см. рисунок 3.18). Таким образом, оценка вычислительной сложности алгоритма в худшем случае – не более $O(B)$.

Утверждение 3.11 доказано.

Утверждение 3.12. Оценка вычислительной сложности алгоритма фразовой фильтрации ρ-последовательности составляет не более $O(|\rho|\cdot B)$, где B – общее количество словоупотреблений в поисковом запросе.

Доказательство. Алгоритм фразовой фильтрации ρ-последовательности реализует цикл обработки ЭД ρ-последовательности ρ. Всего выполняется не более $|\rho|$ итераций. При этом для каждого словоупотребления, найденного по некоторому словоупотреблению запроса (которое имеет синтаксическую связь в составе некоторой фразы в запросе) выполняется алгоритм поиска синтаксических связей в ρ-последовательности. В худшем случае (не возникает на практике на реальных данных) эта проверка будет выполнена на каждой итерации, т.е. для каждого ρ-ЭД ρ-последовательности ρ. Будем считать, что все B словоупотреблений поискового запроса образуют одну фразу. С учётом *утверждения 3.11* это означает, что оценка вычислительной сложности алгоритма в худшем случае – не более $O(|\rho|\cdot B)$.

Утверждение 3.12 доказано.

Заметим, что в среднем оценка вычислительной сложности алгоритма фразовой фильтрации ρ-последовательности составляет $O(|\rho|\cdot L\cdot C)$, где

- L – среднее количество ρ-ЭД, представляющих найденные словоупотребления текстов, входящие в состав одного предложения в этих текстах ($L\sim 1$);
- C – среднее количество словоупотреблений, образующих одну фразу, в поисковом запросе.

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

1. Слияние ρ-последовательностей (*join*), соотнесённых 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 (*cond_join*) к ρ-последовательностям, соотнесённым со словоупотреблениями в запросе, составляющими 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 словоупотребления запроса, которые должны быть исключены из результатов поиска.

4. Формирование ρ-последовательностей ρ_i^e, характеризующих множество предложений найденных документов, в которых встречаются те словоупотребления запроса, которые должны быть исключены из найденных предложений в результатах 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, применяется операция условного исключения элементов (из ρ_i^P исключаются ρ^e, а затем ρ_i^e). В результате формируется множество таких ρ-последовательностей $R^S = \left\{ \rho_i^S \right\}$, 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 $R^T = \left\{ \rho_i^T \right\}$, что каждая ρ-последовательность ρ_i^T соответствует определённому 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 ρ^F.

Алгоритмы, применяемые на каждом из вышеописанных шагов, имеют линейные оценки вычислительной сложности относительно длины своих входных данных. Таким образом, общая оценка вычислительной сложности

также линейна относительно количества σ-ЭД в σ-последовательностях, извлечённых из ИПИ для всех словоупотреблений, составляющих поисковый запрос.

В результате выполнения шагов 1-7 вышеизложенного алгоритма формируется ρ-последовательность ρ^F, представляющая информацию об употреблениях слов запроса в текстах найденных документов, потенциально релевантных запросу. На основе ρ-последовательности ρ^F вычисляются итоговые оценки соответствия найденных документов запросу пользователя.

3.4.4 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 оценки сходства текстов из п.2.2 (формула 2.1) и дополнительно содержат позиции диапазонов ρ-ЭД в ρ-последовательности ρ^F, относящихся к соответствующим предложениям в текстах найденных документов. Это позволяет исключить из рассмотрения те предложения и те документы, которые содержат минимальное количество слов запроса. Вследствие этого повышается общая вычислительная эффективность алгоритма ранжирования, поскольку полный расчёт 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 из п.2.2. формулы (2.13) вычисляются итоговые оценки соответствия предложений найденных текстов предложениям запроса.

Для оценки синтаксического соответствия предложений запроса и найденных текстов применяется алгоритм поиска синтаксических связей в ρ-последовательности в пределах одного предложения (см. рисунок 3.18).

Итоговая оценка релевантности найденного документа вычисляется на основе формулы (2.14).

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 [114]. Куча организуется над массивом фиксированной длины; в её 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.

3.5 Выводы

В настоящей главе представлены разработанные структуры данных и алгоритмы решения поисково-аналитических задач, реализующие метод оценки сходства текстов, описанный во второй главе диссертационного исследования. Для формального описания разработанных алгоритмов и структур данных построена модель данных, базирующаяся на определениях σ-ЭД, ρ-ЭД, σ- и ρ-последовательностей. В качестве операции над данными в разработанной модели выступают:

- слияние σ-последовательностей,
- фильтрация σ-последовательностей,
- преобразования σ-ЭД в ρ-ЭД,
- преобразования σ- последовательности в ρ- последовательность,

- слияние ρ-последовательностей,
- условное слияние ρ-последовательностей,
- одностороннее условное слияние ρ-последовательностей,
- условное исключение элементов ρ-последовательности.

Для реализации всех указанных операций на практике разработаны соответствующие алгоритмы. Для этих алгоритмов получены оценки вычислительной сложности. Также сформулированы требования к алгоритмам, реализующим операции над последовательностями ЭД и определены границы применимости этих алгоритмов (доказано Утверждение 3.1).

Задача поиска информации с использованием разработанных структур данных и алгоритмов решается с использованием классического подхода и включает в себя два этапа:

- построение индексных структур (индексирование документов),
- извлечение информации из индексных структур по запросу с последующей оценкой соответствия найденной информации этому запросу.

На первом этапе решается задача наполнения ИПИ, структура которого ориентирована на эффективное хранение и обработку информации, связанной со словоупотреблениями в текстах индексируемой коллекции документов: в последовательностях σ-ЭД поддерживается введённый порядок элементов, а сами σ-ЭД имеют фиксированный размер в памяти, что позволяет вычислительно эффективно реализовать операции обработки σ-последовательностей на ЭВМ, построенных на современных аппаратных архитектурах. Алгоритмы, применяемые на этом этапе, обладают линейно-логарифмической вычислительной сложностью относительно входных данных размера N: $N \times log(N)$.

На втором этапе производится выборка необходимой информации из ИПИ и применяется ряд алгоритмов, фильтрующих и преобразующих σ-последовательности в ρ-последовательности, в которых сохраняется введённый

порядок следования ЭД. ρ-последовательности поэтапно «объединяются» в результирующую последовательность, на основе которой рассчитываются итоговые оценки соответствия найденных документов запросу пользователя. Все алгоритмы, применяемые на этом этапе, обладают линейной сложностью относительно длины входных данных. Эти алгоритмы эффективно реализуются на современных ЭВМ за счёт использования кэш-памяти центрального процессора при последовательной обработке массивов ЭД.

Вопросам реализации описанных структур данных и алгоритмов в ИАС, а также количественным оценкам их эффективности посвящена следующая глава.

ГЛАВА 4. РЕАЛИЗАЦИЯ И ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ МЕТОДА ОЦЕНКИ СХОДСТВА ТЕКСТОВ, СТРУКТУР ДАННЫХ И АЛГОРИТМОВ ИНФОРМАЦИОННОГО ПОИСКА

4.1 Программная реализация метода оценки сходства текстов, алгоритмов и структур данных информационного поиска

Разработанные метод оценки сходства текстов, а также структуры данных и алгоритмы информационного поиска реализованы в системе семантического поиска Exactus [46, 49, 50, 91, 94] и ИАС Exactus Expert [36, 37].

В системе Exactus разработанные алгоритмы лежат в основе функций поиска информации в собственном индексе информационных ресурсов, а также метапоиска по внешним информационным источникам (поисковым системам).

В системе Exactus Expert разработанные алгоритмы и структуры данных реализуют сервисы информационно-аналитической поддержки исследовательской и экспертной деятельности в научно-технической сфере. Коллекции документов в системе Exactus Expert формируются автоматизировано на основе научно-технических данных, свободно доступных в Интернете на русском и английском языках.

Метод оценки сходства текстов, структуры данных и алгоритмы информационного поиска в виде программных компонентов входят в алгоритмическое и программное обеспечение ИАС, предоставляющее следующие информационно аналитические сервисы:

- полнотекстовый семантический поиск по запросу на естественном языке (с учётом особенностей, описанных в главе 2);
- аннотирование результатов информационного поиска (построение поисковых сниппетов);

- реферирование заданных документов на основе ключевой лексики для быстрого ознакомления с содержанием этих документов;
- поиск библиографических ссылок на заданные работы в текстах;
- поиск публикаций, имеющих сходные формулировки полученных результатов;
- поиск по терминологии и определениям, используемым в публикациях;
- поиск публикаций по формулировкам изложенных в них научно-технических результатов;
- анализ количественной динамики употреблений лексики и терминологии во временном разрезе в различных информационных источниках.

Для преобразования электронных документов во внутрисистемное представление в Exactus Expert 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. [37, 47, 48]. Это позволяет осуществлять полнотекстовый поиск по различным критериям, как в исходном тексте, так и в связанной с ним текстовой метаинформации.

Для проведения морфологического и синтаксического анализа применяются системы FreeLing [34, 144, 145] для английского языка и АОТ [29, 30] для русского языка. На основе результатов работы этих систем

устанавливается семантическая информация с помощью лингвистического анализатора собственной разработки [36, 93, 94].

Сервисы ИАС Exactus Expert функционируют в распределённой вычислительной среде и поддерживают динамическое масштабирование и балансировку нагрузки. Реализация модулей ИАС, отвечающих за индексирование и поиск информации в ИПИ, позволяет проводить эти операции одновременно, а сам ИПИ является динамически пополняемым (инкрементальным).

Описываемые в дальнейших разделах настоящей главы экспериментальные исследования выполнялись на экспериментальном стенде с развёрнутым программным обеспечением ИАС Exactus Expert в распределённой вычислительной среде.

4.2 Экспериментальное исследование метода оценки сходства текстов, структур данных и алгоритмов информационного поиска

4.2.1 Программно-аппаратный состав экспериментального стенда

Аппаратная часть экспериментального стенда включала следующее основное оборудование:

- три вычислительных сервера;
- маршуризатор Gigabit Ethernet 3com 3C16478 Unmanaged Switch 2816, 16-port 10/100/1000 Mbps;
- Источник бесперебойного питания APC Black Smart-UPS 3000VA/2700W, RackMount, 2U.

Основные параметры конфигурация вычислительного сервера:

- процессор: Intel Core i7 CPU 3.1 ГГц (64-разрядный, четырёхъядерный с поддержкой Hyper-Threading);
- оперативная память: 16 Гб;
- дисковые накопители: 4 диска объёмом 500 Гб каждый, с частотой вращения шпинделя 7200 об/мин.;

– аппаратный контроллер массива дисков (RAID): Adaptec 6405 (сконфигурирован для поддержки массива RAID уровня 1).

Сервера функционировали под управлением операционной системы Debian GNU/Linux 7.

4.2.2 Исходные данные для проведения экспериментов

Для проверки работоспособности, отладки и тестирования поисково-аналитических функций, реализуемых разработанным программным обеспечением в составе ИАС Exactus Expert, на экспериментальном стенде была развёрнута тестовая версия системы, включающая следующие коллекции документов:

– российские научные журналы из списка ВАК (около 28 тыс. документов);
– иностранные журналы (более 320 тыс. документов);
– авторефераты диссертаций на соискание учёной степени доктора наук (около 12 тыс. документов);
– зарубежные и российские патенты (около 400 тыс. документов).

Эти коллекции были сформированы с применением средств автоматизированного наполнения информационных баз ИАС Exactus Expert [47, 48] из открытых информационных источников Интернета.

Помимо этого для тестирования производительности системы (скорости индексирования и поиска по запросу) средствами ИАС Exactus Expert также сформирована и проиндексирована коллекция статей свободной энциклопедии «Википедия» [146], содержащая 958 тыс. документов.

Для сравнительной независимой экспертной оценки качества информационного поиска проиндексирована коллекция нормативных документов (300 тыс. документов Законодательства Российской Федерации, Москвы и Санкт-Петербурга по состоянию на 2 неделю декабря 2006 года), предоставленная организаторами Российского семинара по оценке методов информационного поиска (РОМИП) [147]. Эта коллекция была выбрана в силу

того, что для неё существуют репрезентативные таблицы релевантности, содержащие оценки документов по информационным текстовым запросам [148]. Эти запросы, как правило, представляют собой развёрнутую формулировку потребности на естественном языке и содержат синтаксическую и семантическую информацию.

Всего было проведено три группы экспериментов с использованием вышеописанных данных:

- оценка общей работоспособности ИАС с внедрёнными в неё сервисами, использующими разработанное программное обеспечение, которое реализует метод оценки сходства текстов, структуры данных и алгоритмы, предложенные и исследованные в настоящей работе;
- оценка эффективности разработанных структур данных по занимаемой памяти и быстродействию алгоритмов индексирования и поиска информации;
- оценка качества информационного поиска.

4.2.3 Оценка общей работоспособности ИАС

Оценка общей работоспособности ИАС состояла в обнаружении ошибок, возникающих на этапе индексирования информации, а также на этапе функционирования информационно-аналитических сервисов.

Ошибки, возникающие на этапе индексирования информации, выявлялись путём анализа журналов работы соответствующих системных компонентов. Они включали в себя сообщения о возникших в процессе функционирования модулей некорректных ситуациях, отклонениях от ожидаемых результатов.

Ошибки, возникающие на этапе функционирования информационно-аналитических сервисов, выявлялись пользователями, тестирующими поведение системы. Также ошибки, возникающие в ходе тестирования, регистрировались в журналах работы соответствующих системных компонентов.

На основе анализа обнаруженных ошибок и отклонений в функционировании системы были выполнены необходимые корректировки настроек программных компонентов ИАС. В результате компоненты ИАС функционировали без сбоев в соответствии с ожидаемым поведением, на основании чего был сделан вывод о её работоспособности. После этого были выполнены серии экспериментов.

4.2.4 Оценка эффективности разработанных структур данных по занимаемой памяти и скорости обработки информации

В реализованной в ИАС схеме представления данных каждый σ-ЭД (независимо от вида) занимает в памяти 8 байт (см. Рисунок 3.1 в п. 3.1.2). При классической схеме представления списков вхождений в ИПИ («одно вхождение – один ЭД») каждый ЭД требует 20 байт для хранения всей требуемой информации: код документа – 4 байта, 8 байт – общая информация, связанная с вхождением слова, 8 байт – семантическая информация (см. Рисунок 3.2 в п. 3.1.2). На практике в текстах проиндексированных коллекций документов каждая лексема встречается в тексте в среднем около 3 раз, а доля словоупотреблений с семантической информацией составляет 25–40%. Из этих данных следует, что для хранения информации обо всех вхождениях некоторой лексемы в документ потребуется:

- в классической схеме организации ИПИ без учёта семантической информации – 3x12=36 байт (где 12 байт – размер ЭД без учёта полей данных для хранения семантической информации);
- в классической схеме организации ИПИ с учётом семантической информации – 3x20=60 байт;
- в разработанном ИПИ без учёта семантической информации – 8+3x8=32 байта (8 байт на 1-ЭД, далее следуют 3 2-ЭД);
- в разработанном ИПИ с учётом семантической информации (из расчёта, что она присутствует у 40% словоупотреблений) – 8+3x(8+8x0.4)=41.6 байт.

Соотношение размеров ИПИ при различном способе представлении информации приведено на диаграмме – Рисунок 4.1.

По сравнению с наиболее распространённой схемой представления информации в ИПИ (левый столбец) учёт семантических связей и значений увеличивает размер памяти, необходимой для хранения ИПИ, на 16% (в среднем). По сравнению с наиболее эффективной схемой организации ИПИ без учёта семантической информации (третий слева столбец) увеличение размера ИПИ составляет 30%. Использование классической схемы организации ИПИ с учётом семантической информации приводит к увеличению затрат памяти на 67% (второй столбец слева). Преимущество разработанного ИПИ перед таким вариантом организации ИПИ составляет 51% (в среднем).

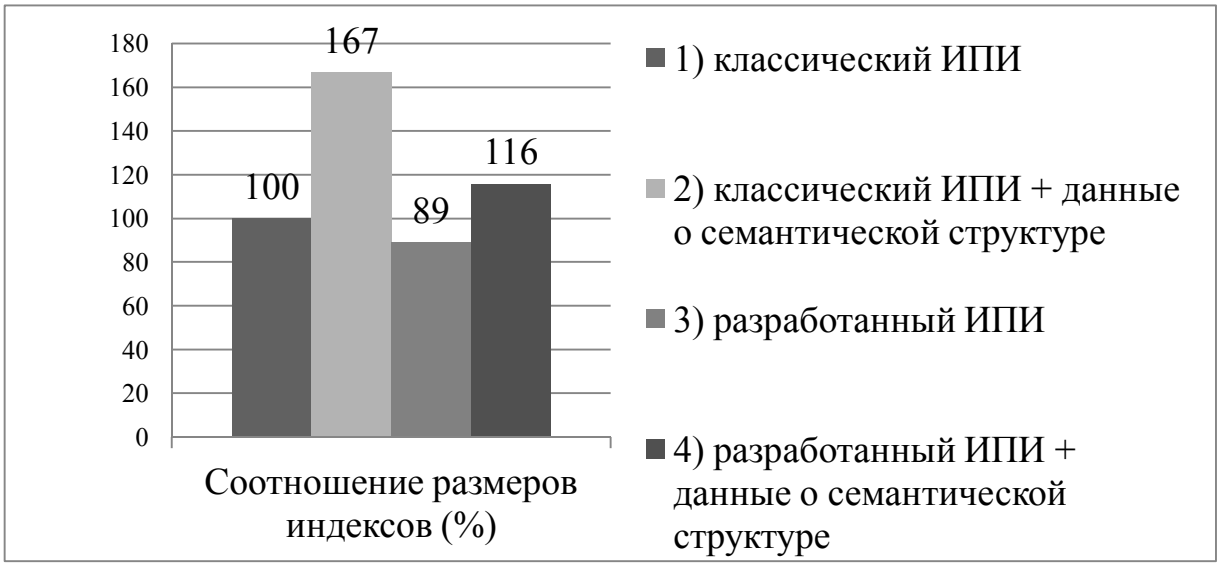

Рисунок 4.1 – Соотношение размеров инвертированных поисковых индексов при различных схемах представления информации

Из приведённого сравнения следует, что разработанный ИПИ эффективно представляет информацию о словоупотреблениях в текстах документов коллекции. Отметим, что итоговый выигрыш предложенной схемы организации ИПИ зависит от среднего количества словоупотреблений одной лексемы из расчёта на один текст (ИПИ тем эффективнее, чем больше эта величина, которая, как показывают эксперименты, пропорциональна длине документа).

Для получения абсолютных значений размеров ИПИ и оценки скорости обработки информации с применением разработанных структур данных и алгоритмов была проиндексирована коллекция «Википедия». Общий объём текста на русском языке в этой коллекции (без учёта гипертекстовой и другой вспомогательной разметки) составил более 8,5 Гб (в кодировке UTF-8), которые распределены по 958 тыс. документов.

В ходе нагрузочного тестирования процесса индексирования на экспериментальном стенде ИАС функционировало 17 потоков анализа текстов. Формирование ИПИ осуществлялась одним модулем-индексатором в 8 потоков агрегации ИТД. В результате коллекция «Википедия» была целиком обработана за 6,5 часов. Средняя скорость индексирования составила около 23 Мб текста в минуту (следует принять во внимание тот факт, что помимо ИПИ формировались и другие индексы ИАС, отвечающие за выполнение других поисково-аналитических функций). Общий размер ИПИ составил менее 7,2 Гб.

Среднее время выполнения процедуры поиска в ИПИ коллекции «Википедия» (без учёта времени построения поисковых сниппетов) составляет менее 0,1 с на один поисковый запрос (параллельные запросы отсутствуют). При нагрузке в 60 запросов, выполняемых одновременно (эмулируемой программно параллельными запросами), среднее время получения результата процедуры поиска возрастает до 0,5 с на один поисковый запрос.

Как видно из представленных экспериментальных данных, реализация разработанного метода оценки сходства текстов в виде поисковых сервисов ИАС соответствует современным требованиям к времени построения поисковых индексов и времени поиска информации. При этом разработанное программное обеспечение, реализующее предложенные алгоритмы и структуры данных, не требует специальных аппаратных ресурсов и позволяет обрабатывать масштабные коллекции текстов на современном стандартном оборудовании.

На основании вышесказанного, разработанные метод оценки сходства текстов, структуры данных и алгоритмы информационного поиска в составе

программного обеспечения ИАС эффективны по критериям занимаемой памяти и быстродействия. Перейдём к оценке качества информационного поиска, реализованного на основе разработанных алгоритмов.

4.2.5 Методика оценки качества информационного поиска

При экспериментальной оценке качества поиска необходимо определить, насколько предложенный метод оценки сходства текстов и его реализация в виде информационно-поисковых сервисов способны удовлетворить информационные потребности эксперта – пользователя ИАС. Для этой цели была разработана методика оценки качества информационного поиска, в рамках которой реализованы две серии экспериментов.

В первой серии экспериментов требовалось получить абсолютные оценки качества решения задачи информационного поиска в ИАС для конкретного пользователя-эксперта. Во второй серии экспериментов требовалось сравнить разработанный алгоритм информационного поиска с другими методами, используя независимые экспертные оценки нескольких асессоров.

При выполнении ***экспериментов первой серии*** были привлечены эксперты для оценки результатов поиска в соответствии со следующей методикой, которая является модификацией метода общего котла [149, 150] и описана в работе [40]. В качестве тестовой коллекции использовалась коллекция российских научных журналов из списка ВАК РФ. Методика оценки качества информационного поиска при выполнении экспериментов первой серии включала в себя следующие шаги.

1. Было сформировано множество поисковых запросов, направленных на поиск научно-технической информации в предметной области экспертов.
2. Запросы были подобраны таким образом, чтобы выдача по этим запросам была непустая и содержала, как минимум, X результатов ($X=10$ – глубина пула результатов, подвергаемых экспертной оценке).
3. Экспертной оценке подвергались первые X результатов в поисковой выдаче. Эксперт просматривал первые X результатов поиска (включая

полные тексты) и принимал решение о релевантности найденных результатов. Если эксперт определял некоторый документ как релевантный некоторому запросу, то он отбирал его для последующего анализа. В противном случае документ получал оценку «нерелевантен».

4. Для каждого из отобранных запросов эксперты сформировали расширенные множества релевантных документов. В них вошли документы, отобранные на предыдущем шаге, как релевантные соответствующим запросам. Затем множества релевантных документов, были пополнены документами из имеющихся у эксперта подборок документов в предметной области, содержание которых хорошо известно эксперту. Также множества релевантных документов были пополнены документами, которые были найдены экспертами с применением поисковых машин Интернета. Для каждого запроса было отобрано P релевантных документов.
5. Эксперт ранжировал релевантные документы в соответствии со шкалой оценок релевантности от 1 до 5, в которой значение 1 соответствует нерелевантному документу, а значение 5 – абсолютно релевантному. Таким образом, были сформированы таблицы релевантности.

Всего, таким образом, было подобрано $Y=14$ запросов. На выбранной глубине пула были получены оценки для Xx$Y=140$ документов. Выбор вышеописанной методики объясняется тем фактом, что задача экспертной оценки релевантности весьма трудоёмка: эксперт должен прочитать документ и осмыслить его содержание, а потом сопоставить его с формулировкой запроса, чтобы выставить оценку по шкале от 1 до 5. При этом эксперту проще принять решение в бинарной шкале («релевантный – нерелевантный». В предложенном способе формирования таблиц релевантности от эксперта требуется оценить выдачу ИАС сначала по бинарной шкале, а затем, используя свои знания в предметной области, сформировать множество релевантных документов с оценками по шкале от 1 до 5. В результате гарантируется, что для каждого

запроса на глубине пула *X* будет *X* релевантных документов (с оценкой >2 по шкале от 1 до 5).

Выбор вышеописанного способа оценки объясняется также тем фактом, что в первой серии экспериментов участвовал только один алгоритм ранжирования. В этом случае классическим методом общего котла невозможно оценить метрику полноты. Для этого эксперту потребовалось бы проанализировать всю выдачу поисковой машины (на глубине пула, равной количеству документов в выдаче поисковой машины), что на практике не возможно.

Сформированные множества релевантных документов были добавлены в коллекцию без каких-либо дополнительных корректировок и поправок.

Вторая серия экспериментов состояла в оценке качества информационного поиска по заранее сформированным таблицам релевантности для коллекции нормативных документов [147]. Таблицы релевантности РОМИП 2008 года для этой коллекции содержат *Y=95* запросов, для которых совокупно оценены *D=12165* документов. Для проведения экспериментов была использована OR-оценка для двух экспертов-асессоров с бинарной шкалой (релевантен, нерелевантен). В имеющихся таблицах релевантности количество релевантных документов $D_{rel}=3601$; количество нерелевантных D_{notrel}=8561.

Качество информационного поиска оценивалось с помощью 11-точечного графика «полнота-точность», построенного по методике TREC [151]. Дополнительно для каждого запроса рассчитывалось значение метрики *nDCG* [152], а затем выполнялось её усреднение на множествах запросов первой и второй серии экспериментов. 11-точечный график «полнота-точность» позволяет наглядно представить оценку качества работы нескольких алгоритмов ранжирования: График показывает, насколько точны результаты поиска в зависимости от достигаемого уровня полноты. Чем выше график, тем больше средняя точность работы алгоритма.

Метрика *nDCG* характеризует алгоритм ранжирования сходным образом. Она показывает, насколько близки результаты поиска к «идеальной» (с точки

зрения эксперта-асессора) поисковой выдаче, в которой вначале идут наиболее релевантные (по оценкам) документы, затем – менее релевантные, а в конце – нерелевантные документы. Большему значению метрики соответствует лучший алгоритм с точки зрения эксперта-асессора.

4.2.6 Результаты экспериментальной оценки качества информационного поиска

Для экспериментальной оценки качества информационного поиска на тестовой версии системы были проведены две серии экспериментов в соответствии с вышеизложенной методикой. Поскольку метод оценки сходства текстов, лежащий в основе тестируемого алгоритма ранжирования результатов информационного поиска, имеет набор параметров, определяющих численные значения критериев оценки сходства и их вклад в итоговую оценку сходства, то в ходе экспериментов требовалось подобрать такой набор параметров алгоритма ранжирования (профиль ранжирования), который максимизировал бы значения метрик качества ранжирования результатов поиска: 11-точечный график «полнота-точность» и *nDCG*. Было сформировано множество профилей ранжирования, где каждый параметр изменялся по заданной дискретной решётке числовых значений таким образом, чтобы для профиля в целом сохранялось условие нормировки (2.12).

Поскольку выбранные метрики рассчитываются на основе усреднения значений, полученных по каждому запросу в отдельности, было сделано предположение о том, что малое (в пределах одного порядка) изменение параметров профиля приводит к малому изменению результатов ранжирования и соответствующих им значений метрик качества. Существенных различий между поисковыми выдачами можно достичь, если варьировать параметры алгоритма таким образом, чтобы они отличались друг от друга на порядки (с соблюдениями ограничений, сформулированных в главе 2). Эта гипотеза была подтверждена в ходе экспериментов.

В обеих сериях экспериментов для всех запросов был выполнен поиск документов в соответствующих коллекциях. Затем выполнен расчёт значений

метрик качества информационного поиска. При этом документы поисковой выдаче, для которых отсутствовали оценки в таблицах релевантности, считались нерелевантными.

Значения метрик в первой серии экспериментов представлены на рисунке 4.2.

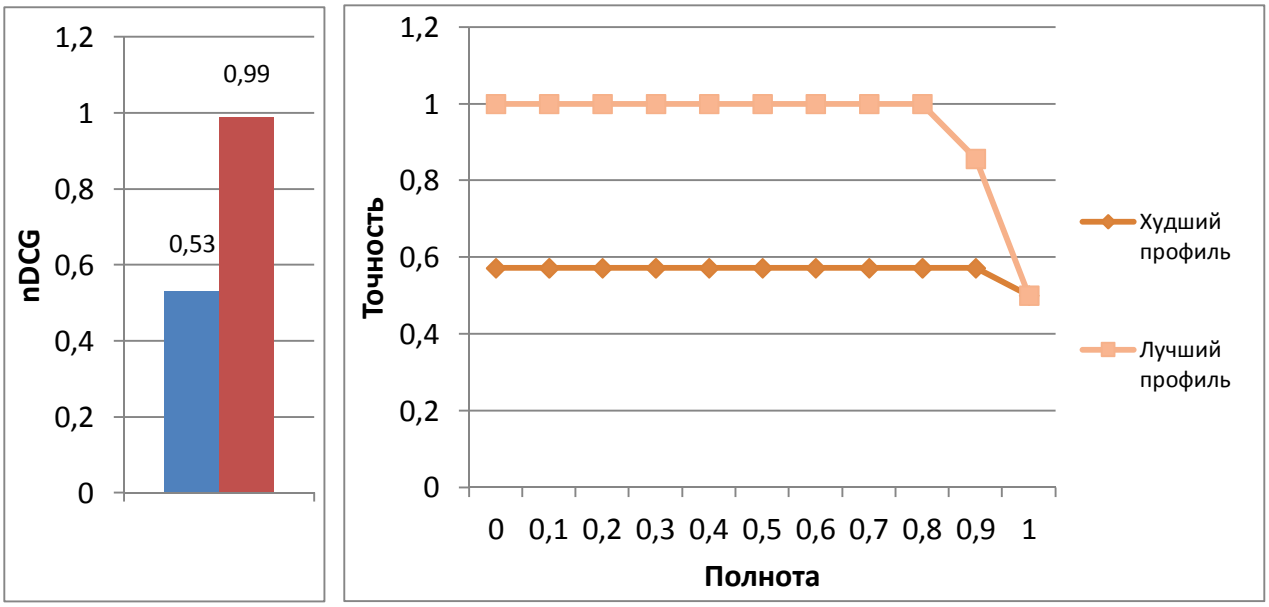

Рисунок 4.2 – Значения метрик качества ранжирования результатов информационного поиска в первой серии экспериментов

Лучшие средние значения метрики *nDCG* (левая диаграмма) и лучший 11-точечный график «полнота-точность» в первой серии экспериментов достигнуты на одном и том же профиле ранжирования. Из представленного графика и диаграммы видно, что качество ранжирования результатов информационного поиска сильно зависит от выбранного профиля. При этом лучший профиль обеспечивает ранжирование, близкое к «идеальной» поисковой выдаче с точки зрения эксперта-асессора.

Значения метрик во второй серии экспериментов представлены на рисунке 4.3. Также на этом рисунке представлен результат участников 2008 года: лучший результат и результат алгоритма Exactus-2008.

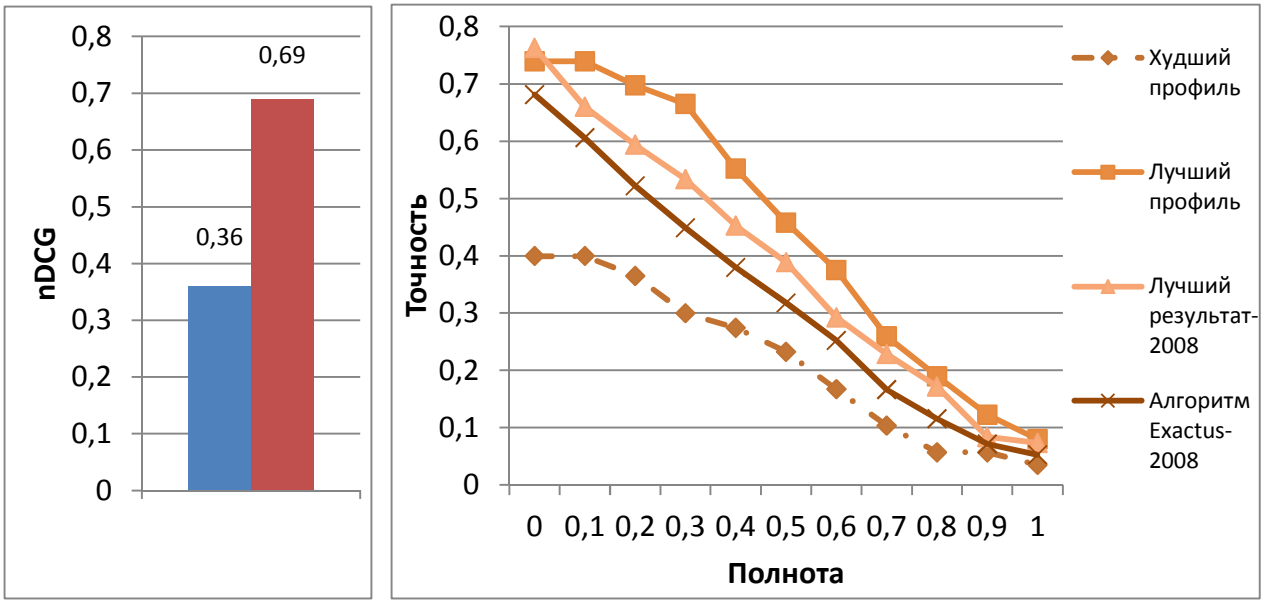

Рисунок 4.3 – Значения метрик качества ранжирования результатов информационного поиска во второй серии экспериментов

Средние значения метрики *nDCG* (левая диаграмма) приведены для лучшего и худшего профиля (в рамках семинара РОМИП-2008 значения этих метрик не рассчитывались). Лучшее среднее значение метрики *nDCG* и лучший 11-точечный график «полнота-точность» во второй серии экспериментов достигнуты на разных профилях ранжирования (которые отличаются незначительно). Это объяснимо тем, что в методике РОМИП документы имеют бинарную шкалу оценок («релевантный–нерелевантный»), и в силу этого значения метрики *nDCG* для поисковых выдач, отличающихся порядком следования релевантных документов, не различаются.

Сравнение оценок качества ранжирования с результатами участников РОМИП 2008 по средней точности (площадь фигуры, ограниченной осями координат и 11-точечным графиком «полнота-точность») приведено на рисунке 4.4. На верхней диаграмме представлены абсолютные значения, а на нижней – изменение средней точности относительно лучшего результата 2008 г. Из представленных диаграмм видно, что относительное увеличение средней точности по сравнению с лучшим значением участников 2008 года составило 17%.

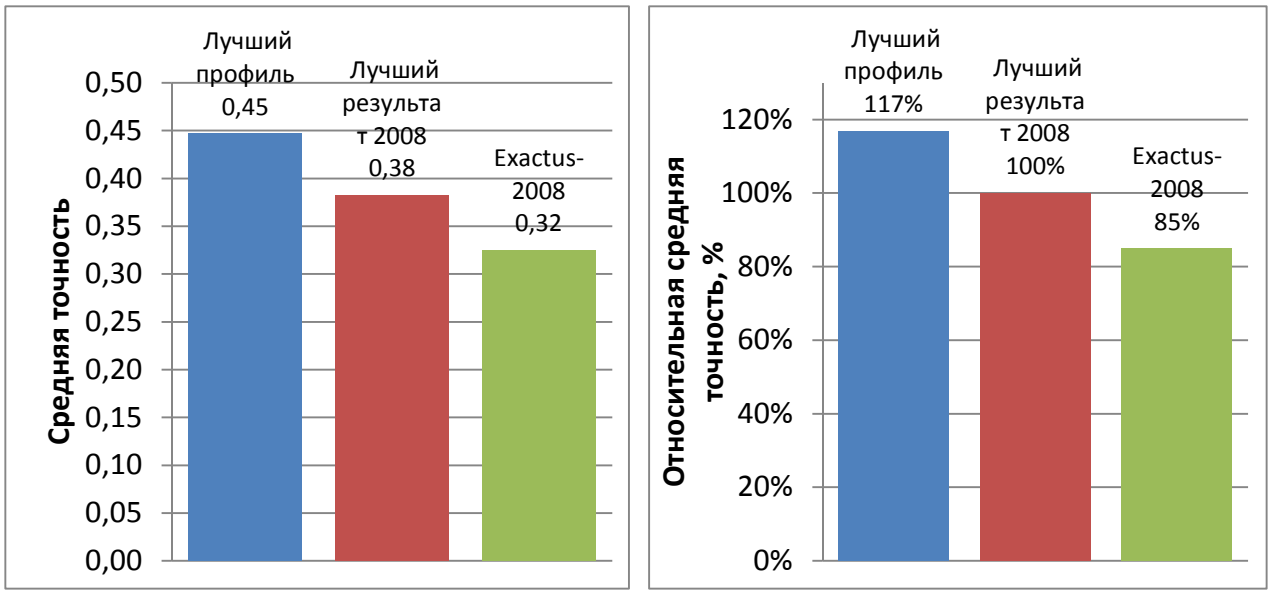

Рисунок 4.4 – Средняя точность результатов информационного поиска

Сводные данные по экспериментам второй серии приведены в таблице 4.1.

Таблица 4.1 – Результаты экспериментов второй серии

ХАРАКТЕРИСТИКА	ЗНАЧЕНИЕ
Количество оцененных запросов	*95*
Количество оценённых документов (в таблицах релевантности РОМИП)	*12165*
Количество релевантных документов (в таблицах релевантности РОМИП): D_{rel}	*3601*
Количество нерелевантных (в таблицах релевантности РОМИП): D_{notrel}	*8561*
Глубина пула оцениваемых документов (для каждого запроса)	*100*
Найдено релевантных документов: F_{rel}	*1769*
Найдено неоценённых документов: $F_{n/a}$	*5079*
Всего найдено документов: N_F	*9100*
$F_{n/a} / N_F$	*0,56*
$F_{rel} / F_{n/a}$	*0,35*
Обобщённая полнота на глубине пула: D_{rel} / F_{rel}	*0,49*
Средняя точность	*0,45*

ХАРАКТЕРИСТИКА	ЗНАЧЕНИЕ
nDCG	*0,69*

Как видно из представленных данных, доля найденных документов, для которых отсутствует оценка в таблице релевантности, по отношению к общему количеству найденных документов составляет 56%. Это означает, что полученные значения метрик качества информационного поиска для разработанного алгоритма ниже их реального значения, поскольку среди неоценённых документов (которые считались априорно нерелевантными) могут присутствовать релевантные. Однако уже достигнутое повышение качества информационного поиска (на имеющихся таблицах релевантности) по сравнению с лучшими результатами участников 2008 г. позволяет говорить об успешном решении задачи исследования.

4.3 Выводы

В настоящей главе описаны особенности реализации метода оценки сходства текстов в виде алгоритмов и структур данных для решения поисково-аналитических задач. Также в главе представлены результаты экспериментального исследования метода оценки сходства текстов, структур данных и алгоритмов информационного поиска в составе ИАС Exactus Expert [36, 37, 39, 40].

Программная реализация разработанных алгоритмов и структур данных была протестирована, отлажена и внедрена в ИАС Exactus Expert.

Выполненные серии экспериментов подтвердили работоспособность разработанных алгоритмов и структур данных. Оценка разработанных программных средств по объёму дисковой памяти, необходимой для хранения индексных структур, показывает их эффективность в сравнении с известными структурами данных для решения задачи поиска информации. Быстродействие разработанных программных средств соответствует современным требованиям к скорости обработки информации в ИАС как на этапе индексирования документов, так и на этапе поиска по запросу. При этом созданные структуры

данных хранят синтаксическую и семантическую информацию текста, а разработанный алгоритм ранжирования результатов информационного поиска, реализующий метод оценки сходства текстов, обеспечивает лучшее качество поиска, нежели алгоритмы без учёта этой информации.

Проведённые 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 объёмам занимаемой памяти информационно-поисковых и поисково-аналитических систем.

ЗАКЛЮЧЕНИЕ

Подводя итоги настоящего диссертационного исследования, перечислим основные полученные результаты:

1. Разработаны лексико-морфологические, синтаксические и семантические критерии оценки сходства текстов, а также метод многокритериальной оценки сходства текстов.
2. С помощью методов логико-алгебраического моделирования разработана и исследована модель данных, предназначенная для исследования свойств структур данных и алгоритмов поисково-аналитической обработки текстовой информации.
3. Разработаны структуры данных инвертированного поискового индекса, предназначенные для хранения и обработки лексико-морфологической, синтаксической и семантической информации, являющейся результатом компьютерного лингвистического анализа текстов. Указанные структуры данных и алгоритмы использованы для эффективного решения задач информационного поиска, в частности, многокритериальной оценки сходства текстов.
4. Разработаны и исследованы следующие алгоритмы поисково-аналитической обработки текстовой информации:
   - алгоритм построения инвертированного поискового индекса коллекций документов,
   - алгоритм поиска информации по запросу, основанный на разработанном методе многокритериальной оценки сходства текстов, реализующий следующие режимы поиска с учётом метаданных документов: поиск по ключевым словам, фразовый поиск, семантический и вопросно-ответный поиск.
5. Теоретически исследованы свойства разработанных структур данных и алгоритмов поисково-аналитической обработки текстовой информации, в том числе получены оценки вычислительной

сложности и доказаны утверждения, обосновывающие корректность указанных алгоритмов.

6. Разработанные структуры данных и алгоритмы программно реализованы и экспериментально проверены. Получены оценки параметров качества информационного поиска. Экспериментальное исследование разработанных структур данных и алгоритмов поисково-аналитической обработки текстовой информации показало, что они обеспечивают более эффективное и качественное решение задач информационного поиска, нежели известные методы, за счёт применения многокритериальной оценки сходства текстов на основе лингвистической информации. Это свидетельствует об успешном решении поставленных задач и достижении цели исследования.

Областью применения разработанного метода оценки сходства текстов, реализованного в структурах данных и алгоритмах, являются информационно-поисковые и информационно-аналитические системы. Программная реализация разработанных алгоритмов и структур данных внедрена в ИАС Exactus Expert [36, 37, 39, 40].

В качестве направления дальнейших исследований отметим разработку метода обнаружения потенциально некорректных заимствований в текстах, а также метода семантического резюмирования, основанных на разработанном методе оценки сходства текстов. В дополнение к уже сформулированным критериям могут быть предложены новые критерии сопоставления текста, не вошедшие в настоящую работу. Также практический интерес представляет разработка эффективного метода подбора параметров профиля ранжирования, обеспечивающего лучшие результаты информационного поиска.

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

ИАС	информационно-аналитическая система
ЕЯ	естественный язык
TF	term frequency
IDF	inverse document frequency
NLTK	Natural Language ToolKit
ФЦП	Федеральная целевая программа
ОНИТ	Отделение нано- и информационных технологий
РАН	Российская академия наук
РФФИ	Российский фонд фундаментальных исследований
ПО	программное обеспечение
КИИ	Конференция по искусственному интеллекту
IEEE	Institute of Electrical and Electronics Engineers
EISIC	Intelligence and Security Informatics Conference
ВАК	Высшая аттестационная комиссия
РФ	Российская Федерация
ЭВМ	электронно-вычислительная машина
POS	part-of-speech
SLR	semantic role labeling
РСМ	реляционно-ситуационная модель
ИПИ	инвертированный поисковый индекс
ЭД	элемент данных
SQL	Structured Query Language
ГС	главное словоупотребление
ХТЗ	характеристика тематической значимости
ФИО	фамилия, имя, отчество
ОСТ	оценка сходства текстов
НСС	неоднородная семантическая сеть
ИТД	индекс текста документа

ИНФЛ	идентификатор нормальной формы лексемы
ИФ	идентификатор формы
ИТЗ	индекс текста запроса
МСЗ	множество семантических значений
РОМИП	Российский семинара по оценке методов информационного поиска
TREC	Text REtrieval Conference
nDCG	normalized discounted cumulative gain

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

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

2 Маннинг К., Рагхаван П., Шютце Х. Введение в информационный поиск. — Вильямс, 2011. — ISBN 978-5-8459-1623-5

3 Zipf, G. K. Selected studies of the principle of relative frequencies of language / Cambridge, Massachusetts: Harvard Unive, 1932.

4 Montemurro, M. A. Beyond the Zipf-Mandelbrot law in quantitative linguistics. / Physica A Statistical Mechanics and its Applications, 2001, №300, P.P. 567-578.

5 Пиотровский Р.Г. Текст, машина, человек. - Л.: Наука, 1975. - 327 с

6 Gerard Salton, Edward A. Fox, and Harry Wu. Extended Boolean information retrieval. / Commun. ACM 26, 11 (November 1983), P.P. 1022-1036. DOI=10.1145/182.358466

7 Brin, S. and Page, L. The anatomy of a large-scale hypertextual Web search engine. Computer networks and ISDN systems, Elsevier, 1998, №30, P.P. 107-117.

8 Page, L.; Brin, S.; Motwani, R. and Winograd, T. The PageRank citation ranking: bringing order to the web. Stanford InfoLab, 1999.

9 С.Протасов Новое ранжирование Рамблера, Почему мы отказались от MatrixNet, Ноябрь 2010 / [Электронный ресурс] URL: **http://slashzone.ru/parser/Protasov-Rambler-NR2010.pdf** (дата обращения 8.03.2013)

10 Kromer, P.; Snasel, V.; Platos, J.; Owais, S.S.J. Implicit User Modelling for Web Search Improvement. / Intelligent Systems Design and Applications, 2007. ISDA 2007. Seventh International Conference on. pp.309-314. doi: 10.1109/ISDA.2007.5.

11 Компания Яндекс – Матрикснет / 2009 [Электронный ресурс] URL: **http://company.yandex.ru/technologies/matrixnet/index.xml** (дата обращения 29.10.2012)

12 И. Зябрев, О. Пожарков. Жадные алгоритмы в Яндексе. / 2010 [Электронный ресурс] URL: **http://www.altertrader.com/publications20.html** (дата обращения 23.03.2013)

13 Paananen, A. Comparative Analysis of Yandex and Google Search Engines. / Master's thesis. Metropolia Ammattikorkeakoulu, 2012 // [Электронный ресурс] URL: **http://publications.theseus.fi/handle/10024/46483** (дата обращения 23.03.2013)

14 Ермаков А.Е. Полнотекстовый поиск: проблемы и их решение. / Мир ПК, № 5, 2001

15 А. Салтыков, С. Куротченко, Р. Дорохин Алгоритм контекстно-зависимого аннотирования / Труды российского семинара по оценке методов информационного поиска РОМИП'2010. - Казань: Казан. ун-т, 2010 // [Электронный ресурс] URL: **http://romip.ru/romip2010/08_rookie_annotation.pdf** (дата обращения 23.03.2013)

16 Carrol, J. A handbook for deterring plagiarism in higher education. Oxford: The Oxford Centre for Staff and Learning Development, Oxford Brookes University, 2002. 96 p. ISBN 1873576560

17 Bao, Jun-Peng; Malcolm, James A., "Text similarity in academic conference papers", 2nd International Plagiarism Conference Proceedings, Northumbria University Press, 2006 // [Электронный ресурс] URL: **http://www.plagiarismadvice.org/research-papers?task=callelement&format=raw&item_id=209&element=f85c494b-2b32-4109-b8c1-083cca2b7db6&method=download&args[0]=e5341b7da69a78afbb46548780fbde78** (дата обращения 23.03.2013)

18 Fabrizio Sebastiani. Machine learning in automated text categorization. ACM Computing Surveys, 34(1):1-47, 2002 [Электронный ресурс] URL: **http://nmis.isti.cnr.it/sebastiani/Publications/ACMCS02.pdf** (дата обращения 23.03.2013)

19 P. Berkhin, Survey of Clustering Data Mining Techniques, Accrue Software, 2002

20 Киселев С.Л., Ермаков А.Е., Плешко В.В. Поиск фактов в тексте естественного языка на основе сетевых описаний // Компьютерная лингвистика и интеллектуальные технологии: труды Международной конференции Диалог'2004. — М.: Наука, 2004

21 Bo Pang; Lillian Lee. "Opinion Mining and Sentiment Analysis". 2008, №2: P.p.1–135 / [Электронный ресурс] URL: **http://www.cs.cornell.edu/home/llee/omsa/omsa.pdf** (дата обращения 23.03.2013)

22 Prabowo R. and Thelwall M. «Sentiment analysis: A combined approach.» Journal of Informetrics, 2009 №3(2): P.p. 143-157. // [Электронный ресурс] URL: **http://citeseer.ist.psu.edu/viewdoc/download;jsessionid=21C7C04158501E322A1371EAEF97B405?doi=10.1.1.102.7125&rep=rep1&type=pdf** (дата обращения 23.03.2013)

23 Shen, Dan, and Mirella Lapata. Using Semantic Roles to Improve Question Answering / Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). 2007 // [Электронный ресурс] URL: **http://www.aclweb.org/anthology/D/D07/D07-1002** (дата обращения 23.03.2013).

24 Stenchikova, S.; Hakkani-Tur, D.and Tur, G. QASR: Spoken Question Answering Using Semantic Role Labeling. // ASRU-2005, 9th biannual IEEE workshop on Automatic Speech Recognition and Understanding, 2005.

25 G. Osipov. Methods for extracting semantic types of natural language statements from texts. // In 10th IEEE International Symposium on Intelligent Control, Monterey, California, USA, 1995.

26 Osipov, G.; Smirnov, I.; Tikhomirov, I.; Zavjalova O. Application of linguistic knowledge to search precision improvement. // InProceedings of 4th International IEEE conference on Intelligent Systems. Vol. 2, pp. 17-2–17-5, 2008.

27 Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno Coppen. Using syntactic information for improving why-question answering. / In Proceedings of the 22nd International Conference on Computational Linguistics (COLING '08), Vol. 1., 2008. Association for Computational Linguistics, Stroudsburg, PA, USA, P.P. 953-960.

28 Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. Question answering passage retrieval using dependency relations. / In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '05). ACM, 2005. NY, USA. P.P. 400-407. DOI=10.1145/1076034.1076103

29 Автоматическая Обработка Текста (АОТ). / [Электронный ресурс] URL: **http://www.aot.ru** (дата обращения 23.01.2013)

30 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 23.01.2013)

31 Solarix: Компьютерная лингвистика. / [Электронный ресурс] URL: **http://www.solarix.ru/** (дата обращения 05.03.2013)

32 Natural Language Toolkit. / [Электронный ресурс] URL: **http://nltk.org/** (дата обращения 05.03.2013).

33 Bird S. Natural Language Processing with Python. / O'Reilly Media Inc, 2009

34 Freeling: An Open Source Suite Of Language Analyzers. / [Электронный ресурс] URL: **http://nlp.lsi.upc.edu/freeling/** (дата обращения 05.03.2013).

35 Национальный цифровой ресурс Руконт - межотраслевая электронная библиотека на базе технологии Контекстум / [Электронный ресурс] URL: **http://www.rucont.ru/** (дата обращения 16.02.2014)

36 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)

37 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

38 И.В. Соченков. Метод сравнения текстов для решения поисково-аналитических задач // Искусственный интеллект и принятие решений. М.: ИСА РАН, 2013, №2, с.95-106.

39 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.

40 И. В. Соченков, Р. Е. Суворов. Сервисы полнотекстового поиска в информационно-аналитической системе (Часть 2) // Информационные технологии и вычислительные системы. М.: ИСА РАН, №3, 2013 , С. 71-87.

41 Р.Е. Суворов, И.В. Соченков. Определение связанности научно-технических документов на основе характеристики тематической значимости. // Искусственный интеллект и принятие решений. М.: ИСА РАН, №1, 2013. С.33-40.

42 Д.А. Девяткин, Р.Е. Суворов, И.В. Соченков. Метод тематической кластеризации масштабных коллекций научно-технических документов. // Информационные технологии и вычислительные системы. М.: ИСА РАН, №1, 2013 , С.33-42.

43 Э. Мбайкоджи, А.А. Драль, И.В. Соченков "Метод автоматической классификации коротких текстовых сообщений" // Информационные технологии и вычислительные системы. М.: ИСА РАН №3, 2012. С. 93-102.

44 O. Vybornova, I. Smirnov, I. Sochenkov, A. Kiselyov, I. Tikhomirov, N. Chudova, Y. Kuznetsova and G. Osipov Social Tension Detection and Intention Recognition Using Natural Language Semantic Analysis (on the material of

Russian-speaking social networks and web forums) // In: Proceedings of the European Intelligence and Security Informatics Conference (IEEE EISIC 2011), September 12-14; Athens, Greece, 2011

45 Соченков И. В., Мбайкоджи Э. Модель представления текста для решения задач машинного анализа естественно-языковой информации // Информационно-телекоммуникационные технологии и математическое моделирование высокотехнологичных систем: Тезисы докладов Всероссийской конференции с международным участием. 18-22 апреля 2011г. – М.: Изд-во РУДН, 2011. Стр. 138–140.

46 Завьялова О.С., Киселёв 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.

47 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.

48 Осипов Г.С., Смирнов И.В., Соченков И.В., Тихомиров И.А. Полнотекстовые электронные библиотеки с сервисами автоматического наполнения и высокоточного поиска / Семнадцатая международная Конференция "Крым 2010".

49 Смирнов И.В., Соченков И.В., Тихомиров И.А. Система интеллектуального поиска и анализа информации «Exactus» на РОМИП-2009 // Российский семинар по Оценке Методов Информационного Поиска. Труды РОМИП 2009. Россия, Санкт-Петербург: НУ ЦСИ, 2009. Стр. 41-52.

50 Смирнов И.В., Муравьев В.В., Тихомиров И.А. Соченков И.В. 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

51 Осипов Г.С., Тихомиров И.А., Соченков И.В., Смирнов И.В. СПОСОБ И СИСТЕМА СЕМАНТИЧЕСКОГО ПОИСКА ЭЛЕКТРОННЫХ ДОКУМЕНТОВ. / Патент РФ на изобретение №2473119, дата отсчета срока действия патента: 05.08.2011

52 Соченков И.В. «Программа определения степени релевантности запросу пользователя». / Свидетельство о государственной регистрации программ для ЭВМ. № 2013613284, 2013 г.

53 Девяткин Д.А., Соченков И.В., Суворов Р.Е. «Программа объективной вычислительной оценки соответствия научно-технического документа заданному набору тематик». / Свидетельство о государственной регистрации программы для ЭВМ. №2013613412, 2013 г.

54 Зубарев Д.В., Соченков И.В. «Программа хранения научно-технических документов на естественном языке». / Свидетельство о государственной регистрации программы для ЭВМ. №2013613410, 2013 г.

55 F. Song and W. B. Croft. A General Language Model for Information Retrieval. / Research and Development in Information Retrieval, 1999. P.P. 279–280

56 Cunningham H. GATE, a general architecture for text engineering. / Computers and the Humanities. 2002.Vol. 36, №2. P.P. 223–254.

57 David D. Palmer , Marti A. Hearst. Adaptive sentence boundary disambiguation. / Proceedings of the fourth conference on Applied natural language processing, October 13-15, 1994, Stuttgart, Germany. DOI:10.3115/974358.974376

58 Урюпина О. Автоматическое разбиение текста на предложения для русского языка. // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог» (4–8 июня 2008 г.). Вып. 7 (14). — М.: РГГУ, 2008.

59 Brants T. TnT: a statistical part-of-speech tagger / Proceedings of the sixth conference on Applied natural language processing. Association for Computational Linguistics. 2000. P.P. 224–231

60 M.F. Porter. An algorithm for suffix stripping, Program. №14(3), 1980. P.P. 130−137.

61 Болховитянов А.В., Чеповский А.М. Методы автоматического анализа словоформ / Информационные технологии. М.: 2011. № 4 (176). С. 24-29.

62 Equations for part-of-speech tagging / Eugene Charniak, Curtis Hendrickson, Neil Jacobson, Mike Perkowitz // Proceedings of the National Conference on Artificial Intelligence. 1993. P. 784–784.

63 Сокирко А. В., Толдова С. Ю. Сравнение эффективности двух методик снятия лексической и морфологической неоднозначности для русского языка (скрытая модель маркова и синтаксический анализатор именных групп) // Сборник работ стипендиатов Yandex. — 2005.

64 Зеленков Ю.Г., Сегалович И.В., Титов В.А. Вероятностная модель снятия морфологической омонимии на основе нормализующих подстановок и позиций соседних слов. // Компьютерная лингвистика и интеллектуальные технологии. Труды международного семинара Диалог'2005. – М., 2005

65 Brill E. A simple rule-based part of speech tagger / Proceedings of the workshop on Speech and Natural Language // Association for Computational Linguistics. — 1992. — P. 112–116.

66 Смирнов И. В., Шелманов А. О. Семантико-синтаксический анализ естественных языков. часть I. Обзор методов синтаксического и семантического анализа текстов // Искусственный интеллект и принятие решений. — 2013. — Т. 1. — С. 41–54

67 Tesnière L. Elements de syntaxe structurale. — Editions Klincksieck, 1959.

68 Chomsky N. Syntactic structures. — Mouton, The Hague, 1957. — P. 117

69 Sleator D. D., Temperley D. Parsing english with a link grammar. — 1991.

70 Link grammar. / 2013 // [Электронный ресурс] URL: **http://www.abisource.com/projects/link-grammar/** (дата обращения 23.03.2013)

71 Протасов С. Преимущества грамматики связей для русского языка // Труды международной конференции «Диалог 2005». — 2005.

72 Протасов С. Обучение с нуля грамматики связей русского языка // X Национальная конференция по искусственному интеллекту с международным участием «КИИ-06». — 2006. — С. 515–524

73 Л.Л. Иомдин, В. В. Петроченков, В. Г. Сизов, Л. Л. Цинман . Синтаксический анализатор системы ЭТАП: современное состояние. / Papers from the Annual International Conference "Dialogue" (2012). — 2012.

74 Federici S., Montemagni S., Pirrelli V. Shallow parsing and text chunking: a view on underspecification in syntax // Cognitive science research paper-university of Sussex CSRP. — 1996. — P. 35–44

75 Syntactic and semantic parser based on ABBYY Compreno linguistic technologies / K. V. Anisimovich, K. Ju. Druzhkin, F. R. Minlos et al. // Papers from the Annual International Conference "Dialogue" (2012). — Vol. 2. — 2012. — P. 91–103

76 Speed and accuracy in shallow and deep stochastic parsing / Ronald M. Kaplan, Stefan Riezler, Tracy H. King et al. // In proceedings of HLT-NAACL'04. — 2004.

77 Briscoe T., Carroll J. Robust accurate statistical annotation of general text. — 2002

78 Miyao Y., Tsujii J. Feature forest models for probabilistic hpsg parsing // Comput. Linguist. — 2008. — Vol. 34, no. 1. — P. 35–80

79 Апресян Ю.Д. Лексическая семантика // Ю.Д.Апресян. Избранные труды. — М., 1995. — Т. 1. — С. 3-69

80 Charles J. F. The case for case. In Universals in Linguistic Theory. — 1968

81 Филлмор Ч. Дело о падеже. // Новое в зарубежной лингвистике. — 1981. — № 10. — С. 400–444

82 Л. Теньер. Основы структурного синтаксиса. / Пер. с франц. Вступ. ст. и общ. ред. В. Г. Гака. М.: Прогресс, 1988. — 656 с.

83 Gildea D., Jurafsky D. Automatic labeling of semantic roles // Comput. Linguist. — 2002. — Vol. 28, no. 3. — P. 245–288

84 Fillmore, Charles J. and Collin F. Baker. FrameNet: Frame semantics meets the corpus. / In Poster presentation, 74th Annual Meeting of the Linguistic Society of America. 2000.

85 Johnson, Christopher R., Charles J. Fillmore, Esther J. Wood, Josef Ruppenhofer, Margaret Urban, Miriam R. L. Petruk, and Collin F. Baker. The

FrameNet project: Tools for lexicon building. Version 0.7. 2001 // [Электронный ресурс] URL: http://www.icsi.berkeley.edu/~framenet/book.html (дата обращения 23.03.2013)

86 Johansson R., Nugues P. Dependency-based syntactic-semantic analysis with PropBank and NomBank / Proceedings of the Twelfth Conference on Computational Natural Language Learning // Association for Computational Linguistics. 2008. P.P. 183–187

87 Золотова Г.А., Онипенко Н.К., Сидорова М.Ю. Коммуникативная грамматика русского языка. – М. 2004. – 544 с.

88 G. Osipov. Methods for extracting semantic types of natural language statements from texts. // In 10th IEEE International Symposium on Intelligent Control, Monterey, California, USA, 1995

89 Осипов Г.С. Приобретение знаний интеллектуальными системами. // М.: Наука. Физматлит, 1997.

90 Осипов Г.С. Методы искусственного интеллекта. – М.: ФИЗМАТЛИТ, 2011. – 296 с.

91 Smirnov I., Tikhomirov I. Heterogeneous semantic networks for text representation in intelligent search engine EXACTUS // Proceedings of workshop SENSE’09 - conceptual Structures for Extracting Natural language SEmantics, The 17th International Conference on Conceptual Structures (ICCS’09). — Moscow, Russia, 2009. — July. — P. 1–9

92 Завьялова О. С. О принципах построения словаря глаголов для задач автоматического анализа текста // Материалы международной конференции по компьютерной лингвистике "Диалог 2004". — 2004.

93 Смирнов И. В. Порождение правил установления значений минимальных семантико-синтаксических единиц текста // Научно-техническая информация. Сер. 2. Информационные процессы и системы. — 2007. — Т. 6. — С. 1–6.

94 Осипов Г. С., Тихомиров И. А., Смирнов И. В. «Семантический поиск в сети интернет средствами поисковой машины Exactus». // Труды одиннадцатой национальной конференции по искусственному интеллекту

с международным участием КИИ-2008. — Т. 3. — М. : ЛЕНАНД, 2008. — С. 323–328

95 Bengtson E., Roth D. «Understanding the value of features for coreference resolution» // Proceedings of the Conference on Empirical Methods in Natural Language Processing / Association for Computational Linguistics. — 2008. — P. 294–303

96 Aron Culotta, Michael Wick, Robert Hall, Andrew McCallum. «First-order probabilistic models for coreference resolution». / HLT/NAACL. — 2007. — P.P. 81–88.

97 Толпегин П. «Новые методы и алгоритмы автоматического разрешения референции местоимений третьего лица русскоязычных текстов». — М. : КомКнига, 2006

98 Mooney R. J. Learning for semantic parsing // Computational Linguistics and Intelligent Text Processing: Proceedings of the 8th International Conference (CICLing 2007) / Ed. by A. Gelbukh. — Mexico City, Mexico : Springer: Berlin, Germany, 2007. — February. — P. 311–324. — Invited paper

99 .Kate R. J., Mooney R. J. Semi-supervised learning for semantic parsing using support vector machines // Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, Short Papers (NAACL/HLT-2007). — Rochester, NY, 2007. — April. — P. 81–84

100 Learning semantic lexicons from a part-of-speech and semantically tagged corpus using inductive logic programming / Vincent Claveau, Pascale Sébillot, Cécile Fabre, Pierrette Bouillon // J. Mach. Learn. Res. — 2003. — Vol. 4. — P. 493–525

101 Toutanova K., Haghighi A., Manning C. Joint learning improves semantic role labeling // Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics / Association for Computational Linguistics. — 2005. — P. 589–596

102 Zhai, C., Notes on the Lemur TFIDF model. Available at / [Электронный ресурс] URL: http://www.cs.cmu.edu/~lemur/1.0/tfidf.ps (дата обращения 20.02.2014).

103 M. E. Maron and J. L. Kuhns. On relevance, probabilistic indexing and information retrieval. J. ACM, 7(3), 1960

104 S.E. Robertson, C.J. van Rijsbergen and P.W. Williams. «Probabilistic models of indexing and searching». / In R.N. Oddy Information Retrieval Research, pp. 35-56, London, 1981. Butterworths. // [Электронный ресурс] URL: **http://www.soi.city.ac.uk/~ser/papers/Robertson_vanRijsbergen_Porter.pdf** (дата обращения 23.03.2013)

105 Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. «Quantitative evaluation of passage retrieval algorithms for question answering». / In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '03). ACM, New York, NY, USA, 2003. P.P. 41-47. DOI=10.1145/860435.860445

106 Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. «Question answering passage retrieval using dependency relations» / In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR ’05. New York, USA. ACM. 2005. P.P. 400–407

107 Verberne, S., Boves, L., Oostdijk, N., & Coppen, P. «What is not in the bag of words for why-QA?» / Computational Linguistics, № 32(2), 2005. P.P. 229–245.

108 Осипов Г. С., Смирнов И. В., Тихомиров И. Реляционно-ситуационный метод поиска и анализа текстов и его приложения // Искусственный интеллект и принятие решений. — 2008. — № 2. — С. 3–10.

109 Suzan Verberne, Lou Boves, Nelleke Oostdijk, Peter-Arno Coppen. «Using Syntactic Information for Improving Why-Question Answering». / COLING 2008. P.P. 953-960

110 Montemurro, M. A. Beyond the Zipf-Mandelbrot law in quantitative linguistics. / Physica A Statistical Mechanics and its Applications, 2001, №300, P.P. 567-578. // [Электронный ресурс] URL: **http://www.kamalnigam.com/papers/multinomial-aaaiws98.pdf** (дата обращения 23.03.2013)

111 Andrew McCallum and Kamal Nigam. «A Comparison of Event Models for Naive Bayes Text Classification». / In AAAI/ICML-98 Workshop on Learning

for Text Categorization. Technical Report WS-98-05. AAAI Press. 1998. P.P. 41-48

112 V. Lavrenko and W. B. Croft. «Relevance-based language models». / Proceedings of the ACM SIGIR 2001, P.P. 120-127

113 Golliher, S.A. «Search Engine Ranking Variables and Algorithms». / semj.org, 2008, №1, Supplemental Issue.

114 D. E. Knuth. «The Art of Computer Programming». / Volume 3: Sorting and Searching, Addison-Wesley. 1973. 722 pages.

115 Frakes, W. and Baeza-Yates, R., Eds. «Information Retrieval: Data Structures and Algorithms». / Prentice-Hall, Englewood Cliffs, New Jersey, 1992.

116 H. E. Williams, J. Zobel, and P. Anderson. «What's next? Index structures for efficient phrase querying». / In J. Roddick, editor, Proceedings of the Australasian Data-base Conference. Auckland, New Zealand. 1999. P.P. 141-152.

117 Justin Zobel, Alistair Moffat. «Inverted files for text search engines». /ACM Computing Surveys (CSUR). Vol. 38, №2, 2006. Article 6. doi:10.1145/1132956.1132959

118 Mihai Lupu, Katja Mayer, John Tait and Anthony J. Trippe. «Current Challenges in Patent Information Retrieval». / Springer, The Information Retrieval Series, Vol. 29, 2011. 415 pages. ISBN 978-3-642-19230-2

119 Hugh E. Williams, Justin Zobel , Dirk Bahle. «Fast phrase querying with combined indexes». / ACM Transactions on Information Systems (TOIS). Vol. 22 №4, 2004. P.P.573-594 [doi:10.1145/1028099.1028102]

120 Губин М.В. «Модели и методы представления текстового документа в системах информационного поиска». / Автореферат дисс. канд. физ.-мат. наук. СПб., 2005. - 16 с. // [Электронный ресурс] URL: **http://maxgubin.com/articles/thesis.pdf** (дата обращения 23.02.2014)

121 T. Strohman, D. Metzler, H. Turtle, and W.B. Croft. «Indri: A language model-based search engine for complex queries». / In Proceedings of the Interna-tional Conference on Intelligent Analysis. Vol. 2, 2005. P.P. 2–6

122 T. Strohman. «Dynamic Collections in Indri». / CIIR Technical Report, 2005 // [Электронный ресурс] URL: **http://ciir.cs.umass.edu/pubfiles/ir-426.pdf** (дата обращения 23.02.2014)

123 Strohman, T. and Croft, W. B. «Low latency index maintenance in Indri». / In Proceedings of the Open Source Information Retrieval Workshop (OSIR'06). 2006. P.P. 7–11. // [Электронный ресурс] URL: **http://ciir.cs.umass.edu/pubfiles/ir-503.pdf** (дата обращения 23.02.2014)

124 Xapian: an Open Source Probabilistic Information Retrieval library. / // [Электронный ресурс] **URL:http://www.xapian.org** (дата обращения 23.02.2014)

125 Aksyonoff A. «Introduction to Search with Sphinx: From Installation to Relevance Tuning». / O'Reilly Media: Sebastopol, California, 2011.

126 Wan, J., Pan, S. «Performance evaluation of compressed inverted index in Lucene». / In: International Conference on Research Challenges in Computer Science. 2009. P.P. 178–181

127 Head-Driven Phrase Structure Grammar. / [Электронный ресурс] URL: **http://hpsg.stanford.edu/** (дата обращения 23.03.2013).

128 Тихомиров И.А., Соченков И.В. Метод динамической контентной фильтрации сетевого трафика на основе анализа текстов на естественном языке. // Вестник Новосибирского государственного университета. Серия: Информационные технологии. 2008. Т. 6. № 2. С. 94-100

129 Joachims, T. «A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization». / DTIC Document, 1996.

130 Robertson, S. E.; Walker, S.; Jones, S.; Hancock-Beaulieu, M. and Gatford, M. «Okapi at TREC-3» / Proceedings of the Third Text REtrieval Conference TREC, 1994.

131 Joaquin Perez-Iglesias. «Integrating the Probabilistic Model BM25: BM25F into Lucene», 2009 / [Электронный ресурс] URL: **http://nlp.uned.es/~jperezi/Lucene-BM25/** (дата обращения 23.03.2013)

132 Браславский П.И., Соколов Е.А. «Сравнение пяти методов извлечения терминов произвольной длины». / Материалы международной конференции «Диалог 2008», Москва. С.67-75

133 Baroni M., Bernardini S. «BootCaT: Bootstrapping Corpora and Terms from the Web». / Proceedings of LREC 2004. Lisbon: ELDA, 2004. P.P. 1313–1316.

134 Bourigault D. «Surface Grammatical Analysis for the Extraction of Terminological Noun Phrases». / Proc. Of COLING-92, Nantes, France, August 23-28, 1992. P. 977–981.

135 Frantzi K., Ananiadou S., Mima H. «Automatic recognition of multi-word terms: the C-value/NC-value method» / Int J Digit Libr. №3, 2000. P.P. 115–130.

136 The size of the World Wide Web (The Internet) / 2013 [Электронный ресурс] URL: **http://www.worldwidewebsize.com/** (дата обращения 23.02.2014)

137 Апресян Ю. Д. Непосредственно составляющих метод // Лингвистический энциклопедический словарь / Под ред. В. Н. Ярцевой. — М.: Советская энциклопедия, 1990. — ISBN 5-85270-031-2

138 Теньер Л. Основы структурного синтаксиса. — М.: Прогресс, 1988. — 656 с.

139 Тихомиров И.А., Смирнов И.В. «Применение методов лингвистической семантики и машинного обучения для повышения точности и полноты поиска в поисковой машине Exactus». / Труды международной конференции Диалог'2009. - С. 483-487.

140 Николас А. Солтер, Скотт Дж. Клепер. «C++ для профессионалов» = «Professional C++». — Диалектика, Вильямс, 2006. — С. 637-639. — 912 с. — ISBN 5-8459-1065-X

141 M. Seeger. «Key-Value stores: a practical overview». / Media, 2009 P.P. 1–21 // [Электронный ресурс] — URL: **http://blog.marc-seeger.de/assets/papers/Ultra_Large_Sites_SS09-Seeger_Key_Value_Stores.pdf** (дата обращения 23.03.2013)

142 Кормен, Т., Лейзерсон, Ч., Ривест, Р., Штайн, К. Глава 7. Быстрая сортировка / Алгоритмы: построение и анализ = Introduction to Algorithms

// Под ред. И. В. Красикова. — 2-е изд. — М.: Вильямс, 2005. — С. 198-219. — ISBN 5-8459-0857-4

143 Седжвик Р. Фундаментальные алгоритмы на C++. Анализ/Структуры данных/Сортировка/Поиск: Пер. с англ./ Роберт Сэджвик. – К.: Издательство “ДиаСофт”, 2001. – 688 с.

144 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.

145 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, eds., Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC’04). Lisbon, Portugal. 2004.

146 Википедия – свободная энциклопедия / [Электронный ресурс] URL: **http://ru.wikipedia.org/** (дата обращения 23.02.2024)

147 Коллекция нормативных документов 2007. / 2007 // [Электронный ресурс] URL: **http://romip.ru/ru/collections/legal07.html** (дата обращения 24.02.2014)

148 Агеев М.С., Добров Б.В., Лукашевич Н.В., Штернов С.В. УИС РОССИЯ в РОМИП 2008: поиск и классификация нормативных документов. / Российский семинар по Оценке Методов Информационного Поиска. Труды РОМИП 2007-2008: Семинар в рамках Всероссийской науч. конф. RCDL'2008. 9 окт. 2008 г., Дубна. СПб.: НУ ЦСИ, 2008. С.44-58. [Электронный ресурс] URL: **http://www.cir.ru/docs/ips/publications/2008_romip_uis.pdf** (дата обращения 23.04.2013)

149 И. Некрестьянов, М. Некрестьянова, А. Нозик. К вопросу об эффективности метода «общего котла». / Труды RCDL'2005. – Ярославль, 2005 // [Электронный ресурс] – URL: **http://rcdl.ru/doc/2005/sek9_1_paper.pdf** (дата обращения 14.01.2013).

150 Агеев М.С., Кураленок И.Б. Официальные метрики РОМИП. — В кн.: Труды третьего российского семинара РОМИП'2005 (Ярославль, 6 октября 2005 г.) - Санкт-Петербург: НИИ Химии СПбГУ, 2005. -224 с.

151 The Twelfth Text Retrieval Conference (TREC 2003). Appendix 1. Common Evaluation Measures. / [Электронный ресурс] – URL: **http://trec.nist.gov/pubs/trec12/appendices/measures.ps** (дата обращения 14.01.2013)

152 Kalervo Jarvelin, Jaana Kekalainen. Cumulated gain-based evaluation of IR techniques. / ACM Transactions on Information Systems №20(4), 2002. P.P.422–446