РОССИЙСКАЯ АКАДЕМИЯ НАУК

ВСЕРОССИЙСКИЙ ИНСТИТУТ НАУЧНОЙ И ТЕХНИЧЕСКОЙ ИНФОРМАЦИИ
(ВИНИТИ)

НАУЧНО • ТЕХНИЧЕСКАЯ
ИНФОРМАЦИЯ

Серия 2. ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ И СИСТЕМЫ

ЕЖЕМЕСЯЧНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ СБОРНИК

Издается с 1961 г. № 12 Москва 2014

ИНФОРМАЦИОННЫЙ АНАЛИЗ

УДК [001.102 : 004.7] (048.8)

С. А. Бельков, С. Л. Гольдштейн

Основные компоненты сетевой информационной навигации (литературно-аналитический обзор)

Представлен обзор средств, используемых при поиске и анализе информации. Рассмотрены традиционные лингвистические подходы и возможности существующих программных средств. Дана классификация используемых при анализе информации формализмов. Приведены основные составляющие сетевой информационной навигации и выдвинуты предложения по расширению традиционной парадигмы.

Ключевые слова: *поисковая машина, структура поискового запроса, анализ документов, эффективность поиска, методы поиска*

ВВЕДЕНИЕ

В наше время способы использования сети Интернет весьма разнообразны. Однако связанные с этим алгоритмы не формализованы и, следовательно, есть много неразрешенных проблем.

Например, существующие в Интернете поисковые системы выдают, как правило, огромное количество документов. Просмотреть все эти документы зачастую практически невозможно. Кроме того, большинство из полученных ссылок малопригодны с точки зрения целей, с которыми пользователь начал поиск, т.е. мы имеем дело с большим количеством информационного шума. Возникает задача ограничения множества найденных в сети документов до релевантного множества тех, которые более точно соответствуют действительным целям поиска. Таким образом, возможность улучшения этой ситуации продолжает оставаться актуальной.

В рамках настоящей статьи сначала определим перечень лингвистических проблем, традиционно возникающих в задачах поиска и анализа текстов. За-

тем приведем основные возможности существующих лингвистических программ, а также создадим классификацию используемых при этом теоретических и математических формализмов. Далее рассмотрим проблемы, возникающие при сетевой информационной навигации, и выделим их основные компоненты. После этого определим некоторые дополнительные подходы и введем несколько полезных формализмов.

ТРАДИЦИОННЫЕ ЛИНГВИСТИЧЕСКИЕ ПОДХОДЫ

Проблема автоматической обработки текста имеет многолетнюю историю. Первоначально, при отсутствии мощных компьютеров, рассматривали преимущественно теоретические аспекты.

В работах [1–3] исследованы следующие вопросы.

1. Базовые понятия:

- семиозис (включает означаемое, означающее и контекст);
- знак (например, символ или слово);
- передача информации (процесс, предполагающий источник – отправитель сообщения, виды каналов передачи, приемник сообщения, интерпретацию, реакцию на распознанное и интерпретированное сообщение, стороннего метанаблюдателя);
- основные составляющие лингвистического знака (слова, словоформы, словосочетания, предложения) – имя (материальный носитель информации), денотат (предмет, явление действительности, обозначаемое именем), десигнат или концепт (смысл, понятие предмета или явления), коннотат (дополнительные экспрессивно-оценочные, а также эстетические значения).

2. Виды информации:

- прагматическая – ценность сообщения с точки зрения заданной цели;
- семантическая – оценивание отношений, возникающих в ходе семиозиса, между означающим и десигнатом знака;
- синтагматическая – характеристика отношения означаемого и денотата знака;
- аффективная – экспрессивно-оценочное и эстетическое восприятие знака, обобщенное в его коннотате;
- синтаксическая – оценка ограничений, накладываемых на комбинаторику и частоту употребления знаков.

3. Уровни интерпретации сообщения:

- интерпретационная способность растений и низших животных и их ответ в виде простейших реакций;
- адаптивная обучаемость – информационная деятельность высших животных, способных принимать не только синтаксическую и аффективную информацию, но и перерабатывать сигматическую (комплексную) информацию, передаваемую сложными знаками, а также обучаться;
- десизивный уровень – информационное, в том числе и речевое поведение человека, опирающееся на прагматическую информацию, обобщающее все остальные виды информации, при котором человек способен к принятию осмысленных решений.

4. Уровни восприятия текста как результата взаимодействия порождающей системы языка и независимой от языка внешней ситуации (последняя подразумевает описываемое текстом событие и общий ситуативный контекст, таким образом, в порождающей системе языка предполагается знание системы языка и знакомство с ситуацией):

- полное, или глобальное распознавание смысла;
- лингвистическое (более ограниченное) распознавание смысла.

5. Уровни лингвистического восприятия и распознавания сообщения:

- низкий (грамматический и словарный);
- продвинутый (фразеологический);
- высокий (лексико-грамматический);
- семантико-синтаксический (полная языковая компетенция).

6. Индивидуальный и социальный аспекты лингвистических знаков.

7. Качественные требования к лингвистическому автомату и порождающим грамматикам.

8. Особенности коммуникативной системы человек – машина – человек.

9. Статистическое моделирование текста:

- создание частотных словарей слов для конкретного естественного языка с возможностью упорядочения по алфавиту и частоте;
- алгоритмическое выявление устойчивых словосочетаний и построение их частотных словарей, возможно, с учетом предметной области.

10. Основные информационные характеристики текста, а также способы угадывания (предсказывания) дальнейших фрагментов текста.

11. Способы измерения смысловой информации.

12. Прочие релевантные теме вопросы.

ВОЗМОЖНОСТИ СУЩЕСТВУЮЩИХ ЛИНГВИСТИЧЕСКИХ ПРОГРАММ

Одна из современных классификаций [4] лингвистических компьютерных средств содержат разделы.

1. Программы лингвистического анализа и обработки текстов, которые включают следующие возможности (реализованные для многих европейских языков):

- парсинг (parsing) – синтаксическая разбивка сложного текста на отдельные конструкции, в простом случае – на слова;
- морфологический (пословный) анализ текста с учетом соответствующей части речи (существительное, прилагательное, глагол), окончаний и т.п. (для русскоязычных текстов используется словарь основных морфологических форм Зализняка);
- лемматизация (приведение слов к нормальной форме);
- более расширенный синтаксический анализ текста (компоненты для грамматического разбора, морфологического анализа, склонения слов и словосочетаний, лемматизации слов и словосочетаний, выделения отдельных частей предложения и т.д.);
- лингвистический анализ текста, включая средства для учета мультиязычности, построения конкорданса, статистического анализа, решения вопросов

автоматического перевода, использования различных словарей и тезаурусов;

- математический анализ структуры текста;
- получение упорядоченных по алфавиту или частоте списков и/или индексов слов в загруженном документе; учет формата, в котором представлен текст; создание набора утилит для построения и обработки словарных частотных индексов и т.д.;
- исследование поведения слов в текстах, построение списка типовых фразеологических оборотов, кластеров слов, упорядоченных по алфавиту и частоте, создание конкордансов и списка ключевых слов с настройкой на тип языка;
- анализ и обработка текста, что позволяет извлекать необходимые сведения из большого объема данных (Text Mining);
- разметка текста, анализ распределения слов в тексте по длине и частоте;
- автоматическая классификация функционального стиля текста, например, на основе спектров длин слов;
- специальные функции: графематический анализ текста, автоматическое уничтожение омонимии, первоначальный семантический анализ текста, лингвистический поиск, полнотекстовый поиск, создание различных тезаурусов и словников;
- основные функции для анализа и классификации текстов, автоматического реферирования;
- семантический анализ текста;
- создание семантической сети понятий путем построения иерархического дерева смысловых тем или подтем текста, а также средств для реализации возможности реферирования текста;
- различные нетрадиционные и креативные подходы, например:

а) графическое изображение связи слов текста в виде галактики, в которой слова играют роль звезд и часто встречающиеся слова светятся ярко, а редкие – вовсе не видны;

б) идентификация в больших корпусах текстов словесных n-грамм с использованием стандартных статистических критериев, таких как тест Фишера на равенство, отношение логарифма вероятности и тест Пирсона хи-квадрат;

в) поиск рифм с фонетическим сравнением слогов с учетом ударения;

г) нахождение для заданного слова синонимов и антонимов;

д) парсер русского языка, использующий при разборе предложений «грамматику связей» (linkgrammar), результатом работы которого является граф, где слова предложения соединены между собой связями (позволяют корректно определить морфологические признаки слов в предложении и разрешить возникающую омонимию);

е) генерация синтаксически корректных предложений;

ж) генерация билингва-текста (текст из двух синхронных половинок на разных языках);

з) создание систем автоматического аннотирования, классифицирования, поиска и морфологической обработки текстовой информации;

и) обработка текстов на основе методов машинного обучения, что включает средства рекурсии, выделения предложений, разметки частей речи, выделения имен собственных, разбора текста и разрешения перекрестных ссылок.

2. Средства преобразования текста из одного формата в другой.

Различные тексты представлены в различных форматах (dos, Ansi, Unicode, Utf8, html, XML, Word-форматы и др.), и для программных систем это является большой проблемой. Поэтому задачи, связанные с преобразованием текстов из одного формата в другой, приведения всех текстовых источников к единому формату или анализа текстов с учетом их разных форматов представления, весьма актуальны.

3. Психолингвистические программы, которые предполагают:

- поиск вложенных слов в тексте, т.е. слов, «спрятанных» внутри и на переходах между словами;
- поиск повторяющихся фрагментов текста при анализе «автоматического письма» (такие тексты пишутся с целью анализа текущих подсознательных процессов);
- синтез подсознательного компонента текста;
- лексический и контент-анализ текстов, с задачами: прогноза эффекта неосознаваемого воздействия текста на массовую аудиторию, анализа текстов с точки зрения такого воздействия, генерации текста с заданным вектором воздействия, выявления личностно-психологических качеств автора текста;
- консультирование с целью помочь пользователю при написании различных текстов, когда пользователь выбирает ряд параметров, характеризующих желаемый результат, а программа выдает ему рекомендации по написанию текста и иллюстрирует их примерами стиля писателей-классиков, современных журналистов и политиков;
- психологические тесты, справочники и базы данных, тренинги и игры, программы для наблюдения биоритмов, психолингвистические программы;
- профессиональные психодиагностические программы.

4. Различного рода генераторы текстов:

- генерация русскоязычных стихоподобных текстов («инструмент поэта»). Программа способна конструировать русские неологизмы на основе заданного словаря с лексико-статистической информацией;
- генераторы случайных текстов на основе заданной грамматики (например, на английском языке), псевдоосмысленных текстов заданной длины, псевдофилософских текстов, текстов мистической тематики, письменных жалоб с генерацией текста жалобы на заданную персону или организацию и т.п.;
- всевозможные программные «боты» (болтуны, виртуальные собеседники), существующие во всемирной сети. В продвинутых случаях этот виртуальный собеседник обладает зачатками искусственного разума, благодаря чему он может реагировать на реплику пользователя своей репликой, которая кажется вполне осмысленной;
- говорящие программы, начиная с классики (например, всемирно известная система ALICE) и

кончая самыми последними разработками с использованием в тексте специализированного языка разметки для искусственного интеллекта (AIML);

- виртуальные анимированные дикторы, например, симпатичная виртуальная девушка, синтезированным голосом рассказывающая о последних новостях;
- интеллектуальная программа естественно-языкового общения с возможностью использования анимации, показывающей на экране монитора образ очаровательной девушки. База знаний программы не слишком обширна, но теоретически ее можно улучшать и дополнять (используется язык разметки AIML).

5. Поисковые машины, конкретные реализации которых имеют хотя бы одну из следующих возможностей:

- полнотекстовый поиск информации по содержимому сайта в пределах корпоративной сети или на отдельном компьютере;
- встраивание в разрабатываемые приложения функции полнотекстового поиска и морфологического анализа текстов (поддерживаются практически все европейские языки, включая русский);
- нечеткий и смысловой поиск документов;
- генерация аннотации для найденного документа;
- поиск и анализ информации в текстовых массивах, реализуемые с использованием технологии нейронных сетей;
- полнотекстовая индексация и поиск с учетом морфологии английского или русского языков;
- поиск и анализ информации с использованием запросов на естественном языке;
- нахождение слова в разных формах и падежах;
- поддержка работы с документами в разных форматах;
- кластеризация результатов поиска и их визуализация в виде семантической сети;
- поиск по большому количеству документов в сети, возможно, со способностью кластеризации полученных результатов по рубрикам, что значительно облегчает и ускоряет поиск нужной информации;
- полнотекстовый семантический анализ документов и извлечение из них знаний (прецедентов, примеров, фактов, решений и прогнозов), интересующих пользователя (при поиске может использоваться семантическая сеть понятий, редактируемая пользователем);
- индексация сообщений почтовых клиентов и клиентов для мгновенного обмена сообщениями;
- решение проблемы поиска для сторонних разработчиков через прикладные интерфейсы (API – Application Interface), интегрируемые в другие приложения для организации поиска по любым источникам данных.

6. Лингвистические словари и тезаурусы различного назначения.

Некоторые вопросы, связанные с компьютерной реализацией словарей, рассмотрены в [1, 5].

7. Системы обработки вводимых предложений естественного языка и машинного перевода. Основные проблемы систем, связанных с компьютерным переводом, подробно рассмотрены в [1, 6].

КЛАССИФИКАЦИЯ ИСПОЛЬЗУЕМЫХ ФОРМАЛИЗМОВ

В работе [7] представлено содержательное описание и краткое математическое описание формализмов, используемых в концепции Text Mining.

Концепция Text Mining опирается на контент-анализ в части: классификации, кластеризации, извлечения фактов, понятий, реферирования, ответов на запросы, тематического индексирования, поиска по ключевым словам, обработки эмпирических данных, оценки энтропии и количества информации, выявления дублирования информации, выявления новых событий.

Классификация информации осуществляется следующими методами: ранжирование и четкая классификация, линейный метод, метод Rocchio, регрессия, днф-классификация, нейронные сети, баейсовский подход, опорные вектора. Существуют также методы оценки качества классификации.

К элементам кластерного анализа относятся: латентно-семантический анализ (матричный и вероятностный), метод κ-средних, иерархическое группирование–объединение, суффиксные деревья, гибридные методы, ранжирование результатов поиска (алгоритмы HITS, PageRank, Salsa, ранжирование по Хиршу).

Поиск по ключевым словам осуществляется по следующим моделям: классическая булева, расширенная булева, нечеткая, векторно-пространственная, вероятностная, модель поиска в пиринговых сетях.

Алгоритмы поиска в пиринговых сетях – это: поиск ресурсов по ключам, широкий первичный, случайный широкий первичный, интеллектуальный, получение большинства результатов по прошлой эвристике, случайные блуждания.

Обработка эмпирических данных включает следующие способы: анализ эмпирических данных (метод Парето, законы Ципфа, закономерность Бредфорда, закон Хипса), анализ степенных распределений случайных величин, использование однородных функций и скейлинга (масштабирования), учет параметра порядка и фазовых переходов.

Может иметь место оценка энтропии: по Шеннону, условная, непрерывного источника информации.

В работе [7] рассмотрены также основы теории сложных сетей, теория перколяции, модели информационных потоков, фрактальный анализ.

Основы теории сложных сетей включают следующие проблемы: оценка параметров сложных сетей (для сети в целом, отдельных узлов сети, распределения степеней узлов, путей между узлами, коэффициента кластерности, степени посредничества узла в пути, эластичности сети, структуры сообщества сети), модели слабых связей и малых миров, рассмотрение WWW как сложной сети (топология, сетевая структура новостных веб-серверов), проблемы визуализации сложных сетей.

Теория перколяции (просачивания, протекания информации в сетях) представляет: описание проблемы, характеристики перколяционных сетей, сети с экспоненциально-широким распределением, диодные перколяционные сети, перколяцию на случайных сетях (графах), использование теории для моделирования атак на сети.

Моделями информационных потоков являются следующие: линейная, экспоненциальная, логистическая, модель диффузии информации, самоорганизованной критичности.

Элементы фрактального анализа включают: абстрактные модели, фрактальное описание информационного пространства, фракталы во временных рядах (метод DFA, корреляционный анализ, фактор Фано, показатель Херста), мультифрактальный анализ рядов измерений.

СОСТАВЛЯЮЩИЕ СЕТЕВОЙ ИНФОРМАЦИОННОЙ НАВИГАЦИИ

Проблемы, связанные с информационной навигацией, предполагается разделить на три основные компоненты:

1) поиск информации (некоторых документов или текстов);

2) анализ найденных документов;

3) работа с сетевыми ресурсами, такими как словари, тезаурусы и онтологии.

Эти проблемы и возможные источники получения информации представлены в таблице.

Компоненты информационной навигации

Вид	Источники
Поиск информации (текста)	Книги, журналы, сборники СМИ Сетевые ресурсы («веб»): сайты и социальные сети Методы поиска
Анализ информации (текста)	Способы представления документов Методы получения множества релевантных текстов Методы анализа
Использование семантических справочников	Словари и справочники Автономные тезаурусы Сетевые тезаурусы Онтологии

Проблема поиска информации

Традиционный процесс поиска (*SP*) документов в сети Интернет можно представить тройкой

$$SP = <Q, SS, DOC; R> , \quad (1)$$

где Q – множество запросов; SS – множество поисковых систем; DOC – получаемые в результате поиска ссылки на документы (далее просто документы), R – матрица связи. Запрос q, как правило, включает список простых ключевых слов и/или словосочетаний, из которых формируется дизъюнкция конъюнктов или дизъюнктивная нормальная форма. В более сложном случае (с учетом связок *и, или, не*, а также скобок) можно говорить о некотором графе запроса.

Для поиска используют известные системы (Yandex, Google, Rambler, Aport, Altavista и др.).

Полученные документы могут быть представлены в разных форматах (txt, doc, pdf, ps, djvu, html, XML и др). Проблема сведения в единый массив разноформатных текстов далеко не тривиальна. Множества документов, полученных разными поисковыми системами в ответ на один и тот же запрос, могут различаться.

При этом возникают следующие задачи:

- выбор наиболее эффективной (с точки зрения цели поиска) поисковой системы;
- оптимизация структуры запроса;
- выбор из множества полученных документов только тех, которые наиболее полно отвечают цели поиска.

Существующие поисковые системы используют различные алгоритмы поиска, детали которых в большинстве случаев неизвестны пользователю, поэтому выбор из них наиболее эффективной далеко не прост. Задачи, связанные с оптимизацией структуры запроса, и ограничения множества найденных документов обычно выходят за рамки возможностей поисковых систем.

Проблема анализа информации

В процессе анализа множества найденных документов, возможно, понадобится решение следующих задач:

а) определить те документы, которые наиболее близки к цели поиска, например, случайным образом, взяв некоторое количество документов с начала множества (для некоторых поисковых систем они обычно наиболее адекватно отвечают цели запроса), или с использованием более специальных процедур (к примеру, взяв документы одного формата представления);

б) разбить множество документов на группы (например, так: маловажные документы, документы средней важности и документы высокой важности), области или кластеры.

При этом используется набор ключевых слов или словосочетаний (термов), которые и представлены в документах [7]. Часть из этих термов присутствует также в запросе q. Характеризующий документ набор его ключевых слов есть образ этого документа.

Для предметной области (ПО) должен иметься словарь Dict, состоящий из термов t_i. В более сложных случаях словарь превращается в онтологию [8]. Также здесь могут использоваться подходы, использующие паттерны [9].

Полученные образы документов дают возможность перейти к решению проблемы классификации (когда есть эталонные образы документов и обучение с учителем) или кластеризации (когда эталонных образов нет и обучение без учителя) документов.

Для определения степени близости двух документов применим математический аппарат следующих

моделей: булевская, расширенная булевская, векторная, с нечеткой логикой, вероятностная. Тем не менее непосредственное сравнение этих методов затруднительно, здесь требуется разработка дополнительного математического аппарата.

Множество документов, выбранных для анализа, обозначим через D.

$$D = \{ d_j \}, j=1,n ,$$

где n – количество документов.

Множество соответствующих документам из D ключевых слов (основных терминов какой-либо предметной области, в информационно-поисковых системах их еще часто называют термами) есть некоторый словарь T, который не совпадает с более обширным словарем Dict предметной области:

$$T = \{ t_i \}, i=1,K ,$$

где K – количество терминов, встретившихся во всех документах из D.

Отдельный документ характеризуется его образом, включающим множество используемых в нем терминов $d_j = \{ t_{i,j} \}$, где $i = 1...m_j$ – количество терминов в документе d_j. Часто просто перечня терминов недостаточно, необходимо учитывать еще и частоту использования термина в документе, т.е. пару $t_{i,j}$, $freq_{i,j}$, где $freq_{i,j}$ – показатель того, сколько раз i-й термин встречается в j-м документе. При подсчете этого показателя необходимо учитывать также данные морфологического анализа, поскольку простое автоматическое разбиение текста на словоформы может дать несколько падежных форм для одного и того же термина. Таким образом, для множества документов получаем некий частотный словарь терминов.

От показателя $freq_{i,j}$ можно перейти к весу термина $w_{i,j}$.

При этом образ документа получает представление в виде вектора весов терминов:

$$d_j = \{ w_{i,j} \}.$$

Способы получения веса термина могут быть различны и зависят от модели анализа и характера дальнейших вычислений.

Так, в булевой модели [7] значение веса $w_i^{(j)} = \{0,1\}$, т.е. фиксируется только наличие или отсутствие термина в документе.

Для расширенной булевой модели уже используются частоты:

$$x = f_i \cdot \frac{idf_i}{\max(idf)} ,$$

где f_i = нормализованная частота терма; idf_i – величина, обратная нормированному количеству документов из D (инверсная частота), содержащих терм t_i; max берется по всем документам.

Для векторно-пространственной модели предлагается следующий способ получения веса термина:

1) определяется частота термина

$$tf_{i,j} = \frac{freq_{i,j}}{\max(freq_{i,j})} ,$$

где $j = 1, n$, т.е. максимум берется по всем документам;

2) вес документа рассчитывается по формуле

$$w_{i,j} = tf_{i,j} \cdot \log\frac{n}{n_i} ,$$

где n_i – количество документов, в которых используется терм t_i.

В других моделях место весов могут занимать функции принадлежности (модель нечеткого поиска) или вероятности (вероятностная модель).

Полученные матрицы попарной близости документов позволяют перейти к их классификации или кластеризации. При этом возможно решение следующих задач:

- отсеивание малоинформативных (с точки зрения цели поиска) документов (шума);
- устранение дублирующих документов;
- разбиение (классификация) множества документов на две (важные, неважные) или три (малой, средней и высокой степени важности) основных категории;
- собственно кластеризация как разбиение множества документов на группы в соответствии со свойствами их образов (векторов признаков).

Важен вопрос, связанный с оценкой качества:

- классификации;
- более общей системы SP, показанной на рисунке (см. далее).

Качество классификации зависит от выбранного метода классификации.

Качество работы SP в первую очередь учитывает следующие коэффициенты:

1) полноты

$$r = \frac{a}{(a+c)} ,$$

где a – выданные релевантные документы; c – не выданные релевантные документы;

2) точности

$$p = \frac{a}{(a+b)} ,$$

где b – выданные, однако не релевантные документы.

Анализ множества терминов

При работе с терминами появляются отдельные классы задач:

- выявление в документе ключевых слов и словосочетаний. Автоматическая обработка текста выдаст для документа список слов (и их частот), но не все из них являются частью более сложных конструкций – словосочетаний и не все из них пополнят список ключевых. Попытки наметить математические и алгоритмические подходы, связанные с формированием ключевых словосочетаний, частично рассмотрены в [7];
- сопоставление словаря, полученного на основе анализа множества документов D, и обычно более широкого словаря ПО Dict (проблема согласования различных множеств);

- переход от множества понятий ПО к графу предметной области как совокупности взаимосвязанных понятий;
- переход от более простой задачи формирования словаря ПО к построению соответствующей онтологии, что предполагает, как минимум, добавление к словарю множеств отношений между понятиями и интерпретаций понятий, когда одни понятия определяются через другие;
- следующим шагом может быть переход от понятий ПО к более сложным структурам – паттернам или фреймам, т.е. построение онтологии, состоящей из паттернов [9].

Взаимосвязи понятий определяются следующим образом.

Подмножество документов из D, содержащих термин t_i , обозначим P_i.

При этом $e_{i,j}$ – признак соответствия понятия (термина) документу:

$$e_{i,j} = \begin{cases} 1, & d_j \in P_i \\ 0, & d_j \notin P_i \end{cases} .$$

Определим уровень связи терминов t_i и t_k :

$$v_{i,k} = \sum_{j=1}^{n} e_{i,j} e_{k,j} \quad .$$

Так получили таблицу взаимосвязей TVP1.

Для учета контекстной близости используем определение профайла для термина t_i как множества терминов из документов, соответствующих этому термину:

$$IP(t_i) = \bigcup_{d_j \in P_i} W_j \quad ,$$

где $W_j = \{ w_{i,j} \}$ – множество ключевых слов, входящих в документ d_j.

Тогда словарю T терминов из D соответствует вектор

$$E_j = < e'_{1,j}, ..., e'_{K,j} >$$

с элементами, подходящими профайлу термина:

$$e'_{i,j} = \begin{cases} 1, & t_i \in IP(t_i), \quad i = 1, K \\ 0, & t_i \notin IP(t_i), \quad i = 1, K \end{cases} .$$

В этом случае уровень взаимосвязи понятий t_i и t_k определяется следующим образом:

$$v'_{i,k} = \sum_{j=1}^{K} e'_{i,j} e'_{k,j} \quad .$$

Так получили таблицу взаимосвязей TVP2.

Таблица взаимосвязей TVP1 всегда отражает взаимосвязи терминов точнее, чем таблица взаимосвязей TVP2, однако таблица TVP2 учитывает взаимосвязи более полно за счет подключения контекста. Таблицы TVP1 или TVP2 можно получить в автоматическом режиме.

На основании таблиц взаимосвязей можно построить граф связей терминов

$$G = \{ V, E \},$$

где множество V соответствует терминам, а E отражает в виде ребер взаимосвязи терминов, при этом каждая вершина графа имеет вес w_i , а каждое ребро – вес $v_{i,k}$. Кроме того, при построении такого графа для $v_{i,k}$ можно определить некоторый предел, для значений ниже которого соответствующее ребро в граф не включается. Это позволит исключить из графа G слабосвязанные понятия.

Множество терминов соответствует множеству терминов, заложенных в связанную с предметной областью онтологию, анализ и группирование на множестве связей (отношений) терминов позволяют сформировать в этой онтологии множество отношений. Используемые при этом процедуры составляют функциональную компоненту онтологии. Проблема сравнения результатов анализа множества документов D с уже имеющейся онтологией требует дополнительных исследований.

Переходя далее от множества терминов к отдельным документам и их взаимосвязям, получаем сеть, в которой одни документы содержат отсылки к другим. Кроме того, множество отдельных документов можно разделить на несколько подмножеств (групп или кластеров), например, следующим образом: центральное ядро (область сильной связности); отправные (ведут в ядро); конечные (ведут из ядра); изолированные (не связанные с ядром).

Используя аналогичные методы, можно определить матрицы взаимосвязей отдельных документов (групп или кластеров) и построить граф взаимосвязей документов (групп или кластеров).

Построение графа взаимосвязей документов позволяет выделить еще один класс задач, связанный с исследованием характеристик сложных сетей из взаимосвязанных документов или их групп.

Отметим также работы (в том числе с участием авторов данной статьи), в которых проблемы поиска, анализа и использования информации исследуются с позиций системного анализа. В них, в частности, рассматриваются следующие вопросы: система управления когнитивностью мультимедийных гипертекстов [10], структура и технологии системного интеллектуального подсказчика по разрешению проблемных ситуаций [11], концептуальные модели запроса/ответа [12], системно-интеграционный взгляд на информацию [13], задача об оценке качества поиска информации [14].

ПРЕДЛОЖЕНИЯ ПО РАЗВИТИЮ ПАРАДИГМЫ

Развитие кортежной модели

Вводя в традиционную схему поиска обратные связи, дополним ее следующими блоками (см. рисунок):

- выбор из множества полученных документов подмножества для анализа (число найденных документов часто велико);
- анализ выбранного подмножества;
- выбор или смена поисковой системы;
- оптимизация структуры запроса.

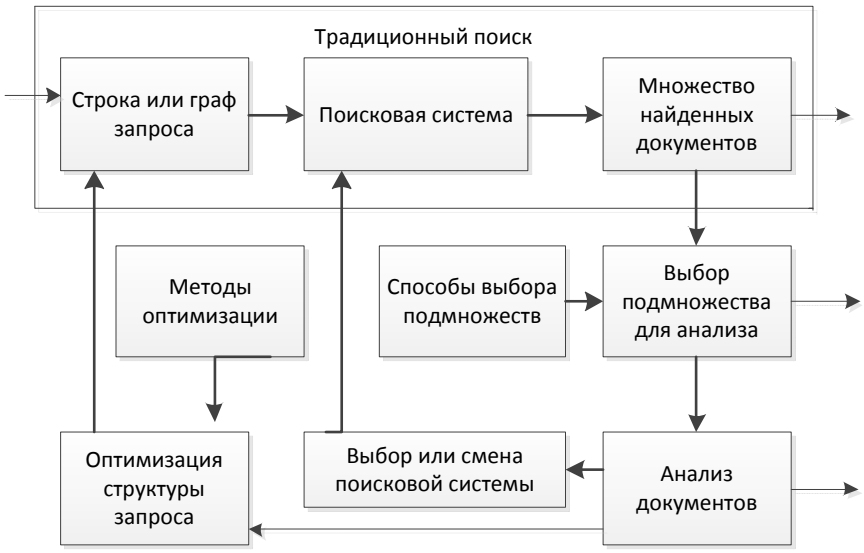


Схема поиска с обратными связями (верхние три блока являются традиционными)

Таким образом, кортеж (1) приобретает вид:

$$SP = <Q, SS, DOC, SPV, V,$$

$$MA, A, Sel, MO, Opt, DO; R2> , \qquad (2)$$

где *SPV* – способы выбора подмножества для анализа; *V* – процедура выбора; *MA* – способы анализа выбранного подмножества; *A* – процедура анализа; *Sel* – выбор или смена поисковой системы; *MO* – методы оптимизации структуры запроса; *Opt* – процедура оптимизации; *DO* – оптимизированное множество документов, *R2* – матрица связи.

Для решения подобных задач применимы методы системного анализа, принятия решений и оптимального управления.

Полезные формализмы

Рассмотрим некоторые из перечисленных компонентов отдельно.

Серьезной проблемой для анализа может стать большая размерность множества документов на выходе поисковой системы. Ограничение анализируемой выборки может происходить случайным выбором документов, привлечением экспертов либо требовать разработки дополнительных процедур.

Для выбора конкретной поисковой системы запишем:

$$SS_k = F_{sel}\,(SS, C_{sel}) ,$$

где F_{sel} – функция выбора; *SS* – множество доступных поисковых систем; *C* – критерии выбора.

Результат анализа множества полученных документов, полученных применением *k*-й поисковой системы следующий:

$$R_k = F_a(DOC_k , C_a , M_a) ,$$

где F_a – функция анализа; DOC_k – множество полученных документов; C_a – критерии анализа; M_a – методы анализа.

Введем также понятие оптимального запроса:

$$Q_{opt} = F_{opt}(R_k) ,$$

где F_{opt} – функция оптимизации структуры (графа связей между ключевыми словами) запроса.

Часто множество найденных документов DOC_k слишком велико (обычно десятки тысяч). Поэтому одним из критериев оптимальности запроса является сокращение количества найденных документов. Другими критериями могут быть адекватность множества цели поиска и полнота рассмотрения темы.

ЗАКЛЮЧЕНИЕ

Мы рассмотрели существующие в настоящее время поисковые системы, на вход которых подается некоторый, возможно, достаточно сложный запрос, а на выходе получается множество найденных документов, многие из которых мы либо не сможем просмотреть физически, либо они дублируют другие документы, либо не пригодятся.

После анализа возникающих при этом задач мы предложили дополнить существующие поисковые системы рядом дополнительных блоков, в частности, помогающих оптимизировать структуру запроса и ограничить множество релевантных документов.

СПИСОК ЛИТЕРАТУРЫ

1. Пиотровский Р. Г. Текст, машина, человек. – Л.: Наука, 1975.
2. Пиотровский Р. Г., Бектаев К. Б., Пиотровская А. А. Математическая лингвистика. – М.: Высшая школа, 1977.
3. Налимов В. В. Вероятностная модель языка. О соотношении естественных и искусственных языков. – М.: Наука, 1979.

4. Логичев С. В. Каталог лингвистических программ и ресурсов в сети. – 2006. – URL: http://www.rvb.ru/soft/catalogue/index.html (дата обращения: 12.12.2013).
5. Пиотровский Р. Г. Компьютеризация преподавания языков: учебное пособие по спецкурсу. – Л.: ЛГПИ, 1988.
6. Искусственный интеллект. Кн. 1. Системы общения и экспертные системы: справочник / под ред. Э. В. Попова. – М.: Радио и связь, 1990.
7. Ландэ Д. В., Снарский А. А., Безсуднов И. В. Интернетика: навигация в сложных сетях. – М.: Либроком, 2009.
8. Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем. – СПб.: Питер, 2001.
9. Бельков С. А., Гольдштейн С. Л. Представление материала текстовых и гипертекстовых источников сетью паттернов // Информационные технологии. – 2010. – № 1. С. 29–34.
10. Гольдштейн С. Л., Бельков С. А., Звонарев С. В. Система управления когнитивностью мультимедийных гипертекстов // Сборник научных трудов «Интеллектика, логистика, системология». Вып. 17. – Челябинск: ЧНЦ РАЕН, 2006. – С. 22–41.
11. Гольдштейн С. Л., Кудрявцев А. Г. Структура и технологии системного интеллектуального подсказчика по разрешению проблемных ситуаций // Сборник научных трудов «Наука и производство». – Челябинск: ЧНЦ РАЕН, 2007. – С. 236–255.
12. Гольдштейн С. Л., Джмухадзе Е. С. Концептуальные модели запроса-ответа // Сборник научных трудов «Интеллектика, логистика, системология». – Челябинск: ЧНЦ РАЕН, 2007. – С. 139–147.
13. Гольдштейн С. Л., Джмухадзе Е. С. Системно-интеграционный взгляд на информацию // Сборник научных трудов «Наука и производство». – Челябинск: ЧНЦ РАЕН, 2007. – С. 213–235.
14. Гольдштейн С. Л., Джмухадзе Е. С. Задача об оценке качества поиска информации // Сборник научных трудов «Экономика и производство». – Челябинск: ЧНЦ РАЕН, 2007. – С. 148–167.

Материал поступил в редакцию 22.05.14.

Сведения об авторах

БЕЛЬКОВ Сергей Александрович – кандидат технических наук, доцент кафедры Интеллектуальных информационных технологий Уральского Федерального университета, г. Екатеринбург
e-mail: srgb@mail.ru

ГОЛЬДШТЕЙН Сергей Людвигович – доктор технических наук, профессор, заведующий кафедрой Вычислительной техники Уральского Федерального университета, г. Екатеринбург
e-mail: s.l.goldshtein@urfu.ru

ИНФОРМАЦИОННЫЙ ПОИСК

УДК [025.4.036 : 004] – 047.58

В.Б. Барахнин, А.М. Федотов

Построение моделей документального и фактографического поиска в электронных библиотеках*

Рассматриваются построение моделей документального и фактографического поиска в электронных библиотеках, работающих с документами достаточно произвольной структуры, а также разработка технологии извлечения фактографической информации из научных документов достаточно произвольной структуры. Предложена модель классификации документов электронной библиотеки, основанная на использовании отношения толерантности, учитывающая возможное отсутствие априорно заданных классификаторов. Показано, что при создании фактографических систем целесообразно следующее понимание факта: содержащаяся в тексте и метаданных документа совокупность связей между сущностями, описываемыми в онтологии информационной системы. Предложена простейшая модель онтологии фактографической системы.

Ключевые слова: *интеллектуальные системы, документальный поиск, факт, фактографический поиск*

ВВЕДЕНИЕ

В классической монографии [1], изданной ВИНИТИ РАН и содержащей подробный обзор теоретических проблем фактографического поиска, на основе выделения двух типов информационных потребностей: потребности в сведениях об источниках необходимой научной информации и потребности в самóй необходимой научной информации – говорится, что для удовлетворения информационных потребностей первого предназначены информационные системы, получившие название *документальных*, второго типа – *фактографических*. В настоящее время наиболее востребованным средством информационного обеспечения научной деятельности становятся *интеллектуальные системы* (ИнтС), сочетающие возможности информационных систем обоих названных типов и позволяющие удовлетворять информационные потребности квалифицированного пользователя в соответствии со схемой «документ – факт – рассуждение» [2, 3]. В дальнейшем мы будем использовать термин «фактографические системы» в широком смысле, включающем и интеллектуальные системы.

Важным этапом процесса функционирования фактографических систем является извлечение из текстов документов содержащихся в них *фактов*, т. е., в наиболее общем смысле, «особого рода предложений, фиксирующих эмпирическое знание» [4].

К сожалению, указанная задача далека не только от сколько-нибудь удовлетворительного решения, но и от достаточно общей постановки. Одна из основных причин этого заключается в том, что с появлением в конце 1970-х гг. персональных компьютеров появились мощные средства визуализации информации, вследствие чего были почти остановлены научные изыскания в области теории создания информационно-поисковых систем. Другой причиной приостановки развития новых алгоритмов обработки фактографической информации стало развитие в начале 1980-х гг. в Японии проекта так называемых «компьютеров пятого поколения», который активно подхватили США, СССР, Великобритания и структуры Европейского сообщества. В процессе реализации этого проекта предполагалось, в частности, разработать технологии логических заключений для обработки знаний, способные делать логические выводы из представленных фактов, хранящихся в сверхбольших базах данных и базах знаний, при этом предусматривалась параллельная обработка данных. Доступ к данным должен был осуществляться с помощью языка логического программирования Пролог. Кроме того, планировалось реализовать поиск характерных признаков в массивах данных автома-

* Работа выполнена при частичной поддержке РФФИ (проекты 12-07-00472, 13-07-00258), и президентской программы «Ведущие научные школы РФ» (грант 5006.2014.9) и интеграционных проектов СО РАН.

тическое реферирование текстов на естественном языке и т.п. Требуемое для решения поставленных задач резкое увеличение производительности предполагалось достигнуть путем замены программных решений на аппаратные, что означало приостановку теоретических исследований в области фактографического поиска. Однако в 1992 г. проект завершился, не достигнув цели. Среди множества имевшихся причин провала проекта мы остановимся лишь на тех, которые связаны с разработкой программного обеспечения. Прежде всего, возможности решения задач в области искусственного интеллекта были переоценены, разработчики питали ничем не обоснованную надежду на то, что возможно создание системы искусственного интеллекта, реализованной на компьютере достаточно большой мощности, способной к самоорганизации, проявляющейся, в частности, в самостоятельном (не зависящим от человека) изменении внутренних правил и параметров системы. Эта идея оказалась непродуктивной: система, которой было позволено «самоорганизовываться», быстро утрачивала целостность и начинала проявлять неадекватную реакцию. Ошибочным был и выбор языка логического программирования Пролог: программы, написанные на нем, плохо отлаживались и не распараллеливались. Наконец, сделанная в процессе реализации проекта ставка на развитие преимущественно аппаратных решений в ущерб программным оказалась ошибочной: аппаратные средства неоправданно усложнялись, а развитие и совершенствование алгоритмов резко затормозилось. Но окончательно похоронило «японский проект компьютеров пятого поколения» появление Интернета, приведшее к возникновению принципиально новой парадигмы распределения и хранения данных. Таким образом, научные изыскания в области теории создания информационно-поисковых систем возобновились лишь в середине 1990-х гг. в связи с развитием информационных технологий сети Интернет и перехода к распределенному хранению информации.

К настоящему моменту в указанной области получены важные теоретические результаты, а также сделан ряд практических шагов по их реализации (см.. например, [5, 6]). Эти разработки обычно опираются на неявное предположение о возможности широкого распространения более или менее подробной стандартизации представления информации, например на основе словарей, как это сделано в рамках концепции Semantic Web консорциума W3 [7].

Однако при попытке автоматизировать процесс извлечения фактографической информации из реальных массивов документов, например, размещенных в сети Интернет, использование концепции Semantic Web неизбежно порождает серьезные проблемы, поскольку наработки консорциума W3 носят лишь рекомендательный характер, а объявить их стандартами могут только организации, имеющие соответствующий статус, например ISO, ГОСТ или ANSI. Ввиду этого реальное развитие большинства ресурсов Интернета, в том числе научной направленности, идет без учета подобных необязательных рекомендаций. Более того, свободный характер размещения материалов в сети Интернет превращает требование соблюдения даже обязательных стандартов представления информации всего лишь в благое пожелание (особенно это касается российской части Интернета). Разумеется, сказанное относится еще в большей степени к электронным документам, не размещенным в Интернете и полученным создателями ИнтС для обработки посредством локального доступа.

Таким образом, возникает необходимость разработки моделей документального и фактографического поиска в электронных библиотеках (ЭБ), работающих с документами достаточно произвольной структуры. Настоящая статья посвящена построению таких моделей.

МОДЕЛЬ КЛАССИФИКАЦИИ ДОКУМЕНТОВ В ЭЛЕКТРОННЫХ БИБЛИОТЕКАХ

Так как задачи поиска и классификации информации взаимно-обратны, то нам достаточно рассмотреть модель классификации документов, наиболее адекватно отражающую особенности работы с электронной библиотекой, в частности, возможное отсутствие априорно заданных классификаторов.

Наиболее распространенным вариантом классификации библиографических ресурсов является фасетная классификация, теория построения которой формализована индийским библиотековедом Ш.Р. Ранганатаном [8]. Объекты классифицируются одновременно по нескольким независимым друг от друга признакам (фасетам). Применительно к электронным библиотекам (и электронным ресурсам вообще) в качестве фасетов выступают элементы метаданных.

Важно отметить, что при создании научно-образовательных ЭБ, для которых библиографические признаки документов гораздо менее важны по сравнению с обычными электронными библиотеками, подмножества множеств значений библиографических метаданных, образующих значения фасетов, как правило, более широки. Так, ссылки на различные переиздания одного и того же документа с точки зрения научно-образовательных электронных библиотек целесообразно считать эквивалентными.

Простейшая формальная модель классификации документов с использованием структурированных метаданных документов выглядит следующим образом [9]. Пусть в справочно-поисковом аппарате ЭБ хранится информация о документах d_i. При этом любой документ d_i. представляется как $d_i = < m_i^{j,k} >$, где $m_i^{j,k}$ – значения элементов метаданных M^j, k – количество значений (с учетом повторений) соответствующего элемента метаданных в описании документа. Рассмотрим подмножество метаданных M_C, определяющее набор классификационных признаков документов, используемых для составления поискового предписания (с учетом заданных логических операций). Для фиксированного элемента метаданных M^j, где $M^j \subset M_C$, заранее определяются подмножества $M_i{}^j$ множества значений этого элемента метаданных (указанные подмножества могут, вообще говоря, пересекаться).

Будем считать два документа *толерантными* (напомним, что толерантность – отношение, которое

обладает свойствами рефлексивности и симметричности, но, вообще говоря, может не обладать, в отличие от отношения эквивалентности, свойством транзитивности; подробно свойства этого отношения исследованы в [10]), если у них значения некоторого элемента метаданных входят в одно и то же подмножество $M_i{}^j$, при этом если значения рассматриваемого элемента метаданных могут повторяться, то документы считаются толерантными при совпадении хотя бы одного из значений. Каждое такое подмножество порождает на множестве документов электронной библиотеки предкласс толерантности, который обозначим $K_i{}^j$.

Более того, в большинстве случаев такие предклассы максимальны, т.е. это –классы толерантности. Предкласс $K_k{}^i$ является классом, если не существует отличного от него (т.е. порожденного другим набором элементов метаданных) предкласса $K_l{}^j$, такого, что $K_k{}^i \subset K_l{}^j$, в противном случае $K_k{}^i$ классом не является.

Выясним в каких случаях предклассы не являются классами (это необходимо, например, для описываемого ниже определения базиса пространства толерантности). Прежде всего, если $M_l{}^i \subset M_k{}^i$, то $K_k{}^i \subseteq K_l{}^i$, и поэтому $K_k{}^i$ классом не является, за исключением конкретного подбора документов, когда $K_k{}^i = K_l{}^i$, но и в этом случае, очевидно, нет смысла рассматривать $K_k{}^i$ в качестве отдельного класса. С содержательной точки зрения этой ситуации соответствует вхождение некоторого раздела классификатора ЭБ в раздел более высокого уровня, когда оба этих раздела учитываются при описании пространства толерантности (разумеется, можно и не учитывать раздел более низкого уровня при определении толерантных элементов, но тогда мы будем иметь дело с пространством толерантности, отличным от первоначального). В описанной ситуации предклассы, не являющиеся классами, определяются априори.

Однако возможна и ситуация, когда $K_k{}^i \subset K_l{}^j$ из-за конкретных особенностей документов ЭБ. Например, в электронной библиотеке по истории математики все документы, имеющие географический признак *Египет*, имеют хронологической признак *до новой эры*, при этом указанный хронологический признаки имеют и документы, относящиеся к другим регионам. Ясно, что в этом случае все документы с признаком *Египет* попарно толерантны не только в силу географического, но и в силу хронологического признака, однако, появление в ЭБ хотя бы одного документа с признаком *Египет*, датируемого *новой эрой*, изменит эту ситуацию. Тем самым в рассматриваемой ситуации предкласс $K_k{}^i$ целесообразно рассматривать (например, при построении базиса) в качестве класса.

Совокупность всех классов толерантности (включая предклассы, рассматриваемые в соответствии со сказанным выше в качестве классов) будем обозначать через H.

Укажем далее, как устроен базис описываемого пространства толерантности (некоторая совокупность H_B классов толерантности называется базисом, если для всякой толерантной пары документов существует класс из H_B, содержащий оба этих документа, а удаление из H_B хотя бы одного класса приводит к потере этого свойства). Очевидно, что множество классов толерантности H_M (включающее по нашему построению, в том числе, и предклассы, рассматриваемые в качестве классов), порожденных всей совокупностью подмножеств $M_i{}^j$, содержит базис. Утверждать, что H_M в точности является базисом нельзя, потому что входящие в него предклассы, не являющиеся классами, могут быть удалены без потери первого свойства из определения базиса. Однако, поскольку добавление в ЭБ даже одного документа может сделать предкласс классом и, стало быть, «полноценным» элементом базиса, постольку рассмотрение таких предклассов в качестве элементов базиса целесообразно с точки зрения организации классификации и поиска документов в электронной библиотеке.

Описание классов толерантности для ЭБ имеет большое практическое значение. Прежде всего, рассмотрим множество всех документов, для которых существует такая совокупность классов (включая предклассы, рассматриваемые в качестве классов) из H, что каждый из этих документов входит в эти и только эти классы. Такое множество представляет собой ядро толерантности, а множество всех ядер толерантности задает отношение эквивалентности на множестве документов ЭБ. При этом для построения ядер толерантности достаточно рассматривать лишь классы (и предклассы) из базиса H_M [10].

Таким образом, поисковое предписание, содержащее подмножество метаданных, определяющее набор классификационных признаков, с указанием сочетаний значений этих метаданных при помощи логических операций, определяет конкретное ядро толерантности на множестве документов, которое и выдается пользователю в качестве ответа на его информационный запрос.

Кроме того, на множестве классов толерантности также можно, в свою очередь, ввести отношение толерантности, при этом толерантными считаются классы, имеющие хотя бы один общий документ. Такая конструкция оказывается полезной, например, для организации поиска документов «по аналогии».

Формализм, основанный на использовании отношения толерантности, оказывается более удобным при создании ЭБ, поскольку в отличие от обычных библиотек, в которых классификаторы заданы априорно, при работе с электронной библиотекой нередко приходится использовать те или иные алгоритмы кластеризации документов (см., например, [3]), а уже потом, исходя из результатов кластеризации, устанавливать подмножества множеств значений элементов метаданных, выступающих в качестве значений фасетов.

УТОЧНЕНИЕ ПОНЯТИЯ «ФАКТ»

Прежде чем обсуждать проблемы работы с фактографической информацией, следует уточнить, какое именно содержание мы будет вкладывать в понятие «факт».

К сожалению, в официальных документах: ГОСТ 7.73–96 «Поиск и распространение информации» и

ГОСТ 7.74–96 «Информационно-поисковые языки» – этот термин практически не формализован. Так, в ГОСТе 7.74–96 дано лишь косвенное, причем не слишком содержательное, определение факта: «7.7. **фактографическое индексирование:** Индексирование, предусматривающее отражение в поисковом образе документа конкретных сведений (фактов)». Интересно отметить, что иноязычные эквиваленты терминов, относящихся к фактографическому поиску (в отличие от подавляющего большинства прочих терминов), в указанном ГОСТе отсутствуют. Что же касается ГОСТа 7.73–96, то интересующее нас понятие косвенно раскрывается в следующем определении: «3.3.7. **база первичных данных; фактографическая база данных:** База данных, содержащая информацию, относящуюся непосредственно к предметной области».

Подробный анализ значения термина «факт» и его производных, основанный на соответствующих статьях «Философской энциклопедии» и «Словаря современного русского литературного языка», был проведен в монографии [1]. В итоге были выявлены следующие признаки фактов:

1. Факты следует отличать от *данных*, фиксирующих специфику объекта, условия наблюдения и т. п. Понятие же научного факта «предполагает элиминирование такой информации, т. е. требует определенного *обобщения* непосредственных данных». Однако при этом отмечается, что четкого различия между указанными понятиями в «Словаре современного русского литературного языка» не приводится.

2. Фактом можно назвать лишь знание, выдержавшее критическую проверку, т. е. полученное в результате обобщения и переработки данных абстрактно-логическим мышлением (разумеется, при этом надо отдавать отчет в том, что достижение абсолютно достоверного знания является лишь идеалом развития науки, практически недостижимым).

3. Любой факт, прежде чем стать объектом научной коммуникации, должен быть преобразован в текст или изображение, получив форму научного документа или его части. Более того, «объектом сбора, хранения, поиска и выдачи в так называемых фактографических информационно-поисковых системах… могут быть лишь соответствующие тексты или документы, описывающие некоторые данные или факты, если под документом понимать… любой фрагмент такого текста» [1].

Нетрудно видеть, что сформулированные признаки весьма расплывчаты. Прежде всего, признаки 1 и 2 предполагают обобщение и оценку перерабатываемых данных. Поэтому жесткое соблюдение требований, вытекающих из указанных признаков, выводит работу с фактами за рамки собственно научно-информационной деятельности, поскольку в той или иной степени требует использования теорий и методик конкретных научных дисциплин, к которым относятся данные.

К тому же, как уже отмечалось, очень трудно провести четкую границу между фактами и непосредственно данными. Это касается следующих типов сущностей, описывающих тот или иной объект исследования: имена собственные, хронологические сведения, различные характеристики объектов и т.п. Например, даже такой, казалось бы, бесспорный факт: «Температура кипения воды равна 100°С» – неявно предполагает указание на условия наблюдения, например химическую чистоту воды и давление в 1 атм, причем последнее условие нельзя заменить на более абстрактное: «стандартное атмосферное давление», поскольку в химии таковым согласно решению Международного союза теоретической и прикладной химии (ИЮПАК) считается давление 100 кПа, меньшее 1 атм., и при «стандартном давлении» температура кипения воды несколько меньше 100°С.

Еще больше проблем возникает в области гуманитарных наук, в частности истории, где некое утверждение, снабженное ссылкой на источник информации, нередко становится новым утверждением, являющимся предметом изучения источниковедения. При этом если исходное высказывание может быть спорным и не являться историческим фактом (например, «Император Александр Первый и старец Фёдор Кузьмич – одно и то же лицо»; о том, что данное высказывание отнюдь не относится к «лженаучным», а заслуживает, по крайней мере, серьезного обсуждения, см. монографию [11]), то утверждение со ссылкой может являться фактом источниковедения («Князь Н.С.Голицын опубликовал версию о том, что император Александр Первый и старец Фёдор Кузьмич – одно и то же лицо, в журнале «Русская старина», 11 книга, 1880 г.»).

Наконец, рассмотрение в качестве фактов имен собственных предполагает, как показано в [1], наличие связей имен собственных с информацией о конкретных носителях этих имен, ибо в противном случае имя несет лишь назывную, но не информационную функцию.

Сказанное объясняет наметившуюся тенденцию стирания граней между понятиями «данные» и «факты», которая отчетливо проявилась в более современной монографии [2], также изданной ВИНИТИ РАН. *Данные* понимаются в ней как факты и идеи, представленные в символьной форме, позволяющей производить их передачу, обработку и интерпретацию, а *информация* – как смысл, приписываемый данным на основании известных правил представления фактов и идей. Структурированная (связанная причинно-следственными и иными отношениями) информация, образующая систему, составляет *знания*.

Для уточнения смысла, вкладываемого в термин «факт» применительно к той области информатики, которая изучает процессы взаимных преобразований данных, информации и знаний в процессе функционирования ИнтС, представляется целесообразным использование семиотического подхода. Понятие «факт» является центральным в «Логико-философском трактате» Л.Витгенштейна [12], одним из источников которого, как отметил Витгенштейн в предисловии трактата, стали работы Г.Фреге – основателя семиотики. Процитируем основные положения трактата, касающиеся фактов:

«…1.1. Мир есть совокупность фактов, а не вещей.

...

1.2. Мир распадается на факты.

1.21. Любой факт может иметь место или не иметь места, а все остальное останется тем же самым.

....

2. То, что имеет место, что является фактом, – это существование атомарных фактов.

2.01. Атомарный факт есть соединение объектов (вещей, предметов).

2.011. Для предмета существенно то, что он может быть составной частью атомарного факта.

…

2.034. Структура факта состоит из структур атомарных фактов.

2.04. Совокупность всех существующих атомарных фактов есть мир.

2.05. Совокупность всех существующих атомарных фактов определяет также, какие атомарные факты не существуют.

2.06. Существование или несуществование атомарных фактов есть действительность. (Существование атомарных фактов мы также называем положительным фактом, несуществование – отрицательным.)

2.061. Атомарные факты независимы друг от друга.

2.062. Из существования или несуществования какого-либо одного атомарного факта нельзя заключать о существовании или несуществовании другого атомарного факта.

…

4.21. Простейшее предложение, элементарное предложение, утверждает существование атомарного факта.

…

4.22. Элементарное предложение состоит из имен. Оно есть связь, сцепление имен».

Положения, выдвинутые в «Логико-философском трактате», имеют большое значение для семиотики, в частности, потому, что в нем устанавливается полное соответствие между онтологическими и семантическими понятиями [13]. Кроме того, Витгенштейн не исключает ложные (или, если угодно, представляющиеся на данном уровне познания ложными) утверждения из числа атомарных фактов, а называет такие факты несуществующими.

Нетрудно заметить, что процитированные положения «Логико-философского трактата» (прежде всего, ключевые определения из раздела 2.01: «**Атомарный факт есть соединение объектов (вещей, предметов)... Структура факта состоит из структур атомарных фактов**») практически полностью воспроизводятся в модели данных «сущность-связь» [14], являющейся основой для унификации различных представлений данных (при этом следует отметить, что в статье [14] для обозначения связи между сущностями не используется термин «факт», а в ее библиографическом списке отсутствует ссылка на «Логико-философский трактат»).

Для единообразия определения понятия «факт» удобно использовать модификацию модели данных «сущность-связь» из той же статьи, называемую моделью множества сущностей. Ее отличительные особенности заключаются в том, что, во-первых, в ней всё трактуется как объекты (в том числе, например, цвет, в то время как в модели «сущность-связь» цвет обычно трактуется как «значение», а согласно «Логико-философскому трактату» «2.0251. Пространство, время и цвет (цветность) есть формы объектов») а, во-вторых, все связи в этой модели – бинарные. Связи между объектами в модели множества сущностей также рассматриваются как объекты, связанные, в свою очередь, с объектами – атрибутами связей.

Важно подчеркнуть, что создание фактографических систем подразумевает извлечение фактов не только непосредственно из текста документа, но и из его метаданных. Это следует, например, из традиционного понимания научно-информационного процесса [15], второй этап которого (аналитико-синтетическая переработка документальной информации) предусматривает как извлечение сведений о содержании документа (индексирование, аннотирование и т.п.), так и обработку его библиографических данных.

Более того, в некоторых случаях целесообразно извлекать и факты, касающиеся не только семантического, но и синтаксического уровня сообщения. В частности, при анализе поэтических текстов [16] исследуются их метрические, ритмические и фонетические характеристики. При этом они могут представлять не только непосредственный интерес, но и использоваться для установления фактов, касающихся, например, авторства документов. Так, Д.С. Самойлов [17], проанализировав особенности рифм одной из версий продолжения X главы «Евгения Онегина», полностью исключил авторство Пушкина, поскольку в этом тексте процент рифм с совпадающими опорными согласными в несколько раз превышает этот показатель в произведениях Пушкина.

Однако всякий ли факт, содержащийся в тексте или метаданных документа, обрабатываемого ИнтС с целью извлечения из него фактов, представляет интерес с точки зрения создателей и пользователей данной системы? Чтобы ответить на этот вопрос, формализуем введенное понятие факта подобно тому, как это было сделано в нашей работе [18] для терминов «информация», «знание», «тезаурус», «онтология». В этой работе, в частности, показано, что данные соответствуют синтаксическому уровню сообщения (в том числе документа), информация (в узком смысле!) – семантическому, а знания – прагматическому. Отсюда вытекает, что функционирование интеллектуальной информационной системы основано на двух противоположных процессах: при пополнении ИнтС новыми сведениями происходит преобразование семантической информации в данные, однако непосредственно потребности пользователя удовлетворяет обратный процесс – извлечение из данных нужной пользователю информации и знаний.

Следовательно, в качестве «первичного» факта рассматривается некоторая информация (как правило, семантическая; примеры возможных исключений приведены выше), но в справочно-информационный фонд ИнтС факт заносится в качестве совокупности элементов данных, описывающих сущности и связи между ними, что соответствует уже упоминавшемуся соотнесению данных и фактов из монографии [2].

Но какого рода информация может быть занесена в справочно-информационный фонд системы в виде данных? Ведь сами по себе данные не несут никакой информационной ценности без соответствующих моделей: например, А.Н.Колмогоров неоднократно отмечал, что данные представляют информационную ценность лишь тогда, когда они являются составной часть некоторой модели реального мира и связаны с другими данными [19, 20]. Таким образом, применение информационных технологий должно основываться на использовании различных моделей (феноменологических, информационных, математических и др.). Как подчеркивал А.А.Ляпунов (см., например, [21]): «нет модели – нет информации».

В качестве модели предметной области обычно выступает ее *онтология* (какой именно смысл мы вкладываем в это весьма широко трактуемое понятие – будет уточнено в следующем разделе).

Таким образом, при создании фактографических информационных систем разумно следующее понимание факта*:* **содержащаяся в тексте и метаданных документа совокупность связей между сущностями, описываемыми в онтологии информационной системы**.

Отсюда, в частности, вытекает следующее важное замечание: именно онтология фактографической системы определяет, что́ будет считаться фактом в рамках этой системы. Здесь мы имеем дело с ситуацией, столь характерной для естественных наук, о которой говорил, например, А.Эйнштейн в своей известной беседе с В.Гейзенбергом: «Только теория решает, что можно наблюдать» [22].

ОСОБЕННОСТИ ОНТОЛОГИЙ ДЛЯ ФАКТОГРАФИЧЕСКИХ СИСТЕМ

Прежде всего, уточним, какого именно понимания термина «онтология» мы будем придерживаться в настоящей работе.

В [18] нами было проведено (применительно к рассматриваемой предметной области) установление определенности в понимании и разграничении использования терминов «тезаурус» и «онтология». Более или менее однозначное трактование термина «тезаурус» сложилось еще в конце 1960-х гг. [23]: это «словарь-справочник, содержащий все лексические единицы информационно-поискового языка – дескрипторы (вместе с ключевыми словами, которые в пределах данной информационно-поисковой системы считаются синонимами этих дескрипторов), причем дескрипторы в словаре должны быть систематизированы по смыслу, а смысловые связи между ними эксплицитно выражены».

Что же касается термина «онтология», в настоящее время, как отмечено в [24], под онтологией нередко стали понимать широкий спектр структур, представляющих знания о той или иной предметной области с разной степенью формализации [25]:

1) словарь с определениями;
2) простая таксономия;
3) тезаурус (таксономия с терминами);
4) модель с произвольным набором отношений;
5) таксономия и произвольный набор отношений;
6) полностью аксиоматизированная теория.

Нами было показано [18], что тезаурус становится онтологией тогда, когда связи между дескрипторами не просто эксплицированы (как это предусмотрено в классическом определении тезауруса), но и классифицированы универсальными зависимостями типа «общее – частное», «часть – целое», «причина – следствие» и т.п. (см., например, [26]). Разумеется, это – лишь «нижняя граница» сложности онтологии. Для эффективной работы с фактами следует, чтобы сущности, относящиеся к предметной области, были представлены не только обозначающими их терминами, но и достаточно широким набором атрибутов, т.е. речь идет об онтологии, обладающей известными признаками модели предметной области.

Разумеется, на первоначальном этапе создания интеллектуальной системы речь, как правило, идет о создании лишь каркаса онтологии, содержащего только краткие сведения о сущностях, а их более подробное описание будет происходить в процессе функционирования ИнтС посредством извлечения из документов соответствующих фактов, выступающих в качестве тех или иных атрибутов сущностей. При этом следует хранить и библиографическую ссылку на информационный источник, из которого был извлечен данный факт.

Поскольку, как уже отмечалось выше, «объектом сбора, хранения, поиска и выдачи в так называемых фактографических информационно-поисковых системах… могут быть лишь соответствующие тексты или документы, описывающие некоторые данные или факты, если под документом понимать… любой фрагмент такого текста» [1], постольку в роли онтологии – модели предметной области – может выступать та или иная модель интеллектуальной информационной системы, например предложенная нами в работе [27]. Эта модель, записанная в качестве модели предметной области, имеет вид

$$S = \langle K, M, M^j \langle K_i, K_{i'} \rangle \rangle,$$

где K – классы сущностей, M – множество используемых атрибутов сущностей, $M^j \langle K_i, K_{i'} \rangle$ – типы возможных связей между классами сущностей, когда сущность из класса $K_{i'}$ может входить в качестве значения атрибута M^j сущности из класса K_i. Тем самым любая сущность s_i представляется как

$$d_i = \langle m_i^{j,k} \rangle,$$

где $m_i^{j,k}$ – значения атрибутов сущности, k – количество значений (с учетом повторений) j-го атрибута в описании сущности.

При создании информационной системы сущности будут представлены в виде описывающих их документов, а атрибуты сущностей будут представлять собой элементы метаданных.

Предложенная модель онтологии полностью соответствует введенному нами пониманию факта, что делает ее наиболее пригодной для создания фактографической системы. Разумеется, пользуясь знания-

ми о предметной области, возможно и целесообразно накладывать различные ограничения (морфологические, синтаксические, семантические, структурно-текстовые) на характеристики сущностей, входящих в те или иные классы (подробно принципы установления ограничений описаны в [28]).

Отметим, что применительно к фактографическим информационным системам, создаваемым в рамках концепции Semantic Web, довольно близкий подход был предложен в работе [5]. Речь идет об использовании модели, в которой сущности внешнего мира представляются атрибутированными информационными единицами, а отношения между сущностями реализуются либо в виде прямых ссылок, либо в виде составных конструкций определенного вида, при этом спецификация такой модели воплощается в виде онтологии.

АВТОМАТИЗИРОВАННОЕ ИЗВЛЕЧЕНИЕ ФАКТОВ ИЗ ДОКУМЕНТОВ

Разработка методик автоматизированного извлечения фактов из документов представляет собой наиболее сложную проблему, возникающую при создании фактографических систем. Это было подчеркнуто еще в [1]: «не существует сколько-нибудь значительных различий в теории и методике построения документальных и фактографических информационно-поисковых систем, если фактографический поиск понимать лишь как процесс отыскания уже готовых данных и фактов, ранее введенных в фактографическую систему… Однако под фактографическим поиском можно понимать и нечто принципиально иное, а именно отыскание машиной требуемых данных и фактов в текстах научных документов, написанных на одном или нескольких разных естественных языках, …[что] требует оперирования со смыслом текстов, его анализа и синтеза, т.е. моделирования достаточно сложных мыслительных процессов».

Собственно говоря, в середине 1970-х гг. возможности компьютеров были явно недостаточными для сколько-нибудь полноценного практического решения поставленной задачи. К настоящему моменту рост мощности компьютеров позволил создавать разнообразные алгоритмы для извлечения данных и фактов из документов на естественных языках. Выбор конкретного алгоритма (или, точнее, даже типа алгоритмов) зависит от того, насколько структурированы (и структурированы ли вообще) данные и факты, содержащиеся в конкретном документе.

1. Табличные данные. Они могут выступать, согласно [1], в качестве фактов, если являются, например, характеристиками предметов, географических объектов и т.п. Для их извлечения из документов существуют разнообразные, весьма надежные алгоритмы (см., в частности, [29], включая библиографический обзор).

2. Массивы однородных слабоструктурированных текстовых документов. Нередко первоначальный этап создания онтологий удобно проводить, занося факты, содержащиеся в массивах однородных документов, описывающих предметную область: биографических справочниках, геологических, ботанических или зоологических каталогах и т.п. В таких случаях наиболее целесообразно использовать алгоритмы, учитывающие информацию о закономерностях их текстовой структуры (например, общих для всех документов массива синтаксических и семантических конструкциях), а также о гипертекстовой разметке обрабатываемых документов (при наличии таковой). Такой алгоритм, извлекающий факты (метаданные) о библиографии документов, подробно описан, например, в нашей монографии [3]. Он может быть легко адаптирован к фактографической информации произвольного характера, содержащейся в массивах документах, имеющих более или менее однородную текстовую структуру.

3. Тексты произвольного характера. Задача извлечения фактов из произвольных текстов на естественном языке до сих пор, по-видимому, не имеет сколько-нибудь общего решения, поскольку построение такого решения предполагает, в частности, достаточно точное моделирование когнитивной деятельности человека, а также наличие мощных средств как синтаксического, так и семантического анализа текстов, включая подробнейшие онтологии, тезаурусы которых учитывают, например, всё богатство синонимии естественного языка (не столько даже в части научной лексики, сколько в части лексики общеупотребительной).

«Частное решение» этой задачи применительно к той или иной предметной области предполагает, прежде всего, построение онтологии, тезаурус которой включает, наряду с описанием сущностей предметной области, по крайней мере, те пласты общеупотребительной лексики (разумеется, с учетом синонимии), которые наиболее характерны для этой области.

Непосредственная работа по извлечению фактов из текста может опираться на совокупное применение методов синтаксического и семантического анализа. Например, общедоступным средством анализа текстов является стеммер (морфологический анализатор) компании «Яндекс» (http://company.yandex.ru/technologies/mystem/), позволяющий извлекать словосочетания заданной структуры, например, (*прилагательное)* + *(существительное)* или (*существительное)* + *(существительное в родительном падеже)*, т.е. проводить не только морфологический, но и синтаксический анализ. Для семантического анализа текстов может быть применен подробно описанный в [3] алгоритм выявления в тексте терминов, в том числе и составных, входящих в словарь онтологии данной предметной области. Само же извлечение факта, относящегося к тому или иному упоминаемому в тексте субъекту, описанному в онтологии, состоит в определении значения предиката, связанного с этим субъектом (описание подробностей конкретной реализации алгоритмов синтаксического и семантического анализа выходит за рамки данной статьи).

О ВЗАИМОДЕЙСТВИИ ФАКТОГРАФИЧЕСКИХ СИСТЕМ С ПОЛЬЗОВАТЕЛЯМИ

Факты, извлеченные из текстов документов, и занесенные в фактографическую информационную систему, могут быть использованы как для дальнейшего получения новых знаний (что, собственно, и характеризует интеллектуальные системы), так и для

непосредственного поиска пользователем системы. При этом нередко в качестве чуть ли непременного атрибута качественной фактографической системы называют возможность формулировки запроса на естественном языке. Однако из изложенного выше, на наш взгляд, вытекает вывод о том, что такая функция не дает пользователям специализированных систем каких-то принципиальных удобств. Действительно, коль скоро мы рассматриваем в качестве фактов характеристики сущностей, описанных в онтологии, то весьма несложный интерфейс, позволяющий просматривать онтологию посредством использования последовательности гиперссылок (или даже посредством таблицы), сможет предоставить пользователю возможность без труда найти нужный факт или, по крайней мере, убедиться в том, что этот факт не занесен в систему. Однако задача «понимания» системой запросов на естественном языке практически эквивалентна задаче извлечения фактов из текстов на естественном языке, о трудностях в решении которой нами сказано выше. При этом следует учесть, что далеко не все пользователи (пусть даже являющиеся высококвалифицированными специалистами в своей предметной области) способны формулировать свой вопрос так четко и недвусмысленно, как, согласно стихотворению проф. А.С.Компанейца, это умел делать на своем знаменитом семинаре в Институте физических проблем АН СССР Л.Д.Ландау (цит. по [30]):

С первых слов, как Вельзевул во плоти,
Навалился Дау на него:
«Лучше вы скажите, ***что в работе***
Ищется как функция чего?»

Слишком же расплывчатая постановка вопроса, «не распознанная» информационной системой, может привести к тому, что у пользователя сложится ошибочное мнение, будто бы система не располагает необходимой ему информацией. Таким образом, непосредственный просмотр онтологии представляется наиболее надежным путем получения конкретной фактографической информации.

Разумеется, возможна и усложненная постановка задачи, когда пользователю требуются не только (или даже не столько) сами факты, но и их анализ, обобщение и т.п. Для решения этой задачи необходимы такие компоненты ИнтС [2], как рассуждающая информационная система, формализующая правила логического вывода, и интеллектуальный интерфейс (диалог, графика и т.д.).

Таким образом, функционирование фактографических информационных систем как частного случая интеллектуальных систем основано на двух противоположных процессах: при пополнении фактографической системы новыми фактами происходит преобразование семантической информации в данные, однако непосредственно потребности пользователя удовлетворяет обратный процесс – извлечение из данных нужной пользователю информации и знаний.

ЗАКЛЮЧЕНИЕ

В настоящей статье изложены модели документального и фактографического поиска в электронных библиотеках, работающих с документами достаточно произвольной структуры. Предложена модель классификации документов электронной библиотеки, основанная на использовании отношения толерантности, учитывающая возможное отсутствие априорно заданных классификаторов. Показано, что при создании фактографических информационных систем целесообразно следующее понимание факта: **содержащаяся в тексте и метаданных документа совокупность связей между сущностями, описываемыми в онтологии информационной системы**. Предложена простейшая модель онтологии фактографической системы.

Важным этапом практической реализации предлагаемых в статье подходов должна стать реализация алгоритмов синтаксического и семантического анализа текстов с целью извлечения фактов.

Примером практического использования фактографических систем может служить проверка в научных издательствах и редакциях журналов достоверности сведений, содержащихся в рукописях, имеющих биографический, научно-публицистический, обзорный и т.п. характер. Факты, извлекаемые из текста рукописей, подвергаются сравнению с «эталонными» фактами из онтологии информационной системы, и в случае расхождения редакция просит автора уточнить правильность приведенных им сведений.

* * *

Авторы выражают признательность Ю.В.Леоновой, обратившей внимание на определение факта в «Логико-философском трактате» Л. Витгенштейна.

СПИСОК ЛИТЕРАТУРЫ

1. Михайлов А.И., Черный А.И., Гиляревский Р.С. Научные коммуникации и информатика. – М: Наука, 1976.
2. Арский Ю.М., Гиляревский Р.С., Туров И.С., Черный А.И. Инфосфера: Информационные структуры, системы и процессы в науке и обществе. – М.: ВИНИТИ, 1996.
3. Шокин Ю.И., Федотов А.М., Барахнин В.Б. Проблемы поиска информации. – Новосибирск: Наука, 2010.
4. Ракитов А. Факт // Философская энциклопедия. Т. 5. – М: Советская энциклопедия, 1970. – С. 298.
5. Марчук А.Г. О распределенных фактографических системах // Труды Десятой Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (RCDL'2008). Дубна, 7-11 октября 2008 г. – С.93-102.
6. Марчук А.Г., Марчук П.А. Архивная фактографическая система// Труды Одиннадцатой Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (RCDL'2009). Петрозаводск, 17-21 сентября 2009 г. С. 177-185.

7. Berners-Lee T., Hendler J., Lassila O. The Semantic Web // Scientific American. – 2001. – Vol. 284(5). – P. 34-43.
8. Ранганатан Ш.Р. Классификация двоеточием. Основная классификация / пер. с англ. – М.: ГПТНБ СССР, 1970.
9. Федотов А.М., Барахнин В.Б. Проблемы поиска информации: история и технологии // Вестник НГУ. Серия: Информационные технологии. – 2009. – Т. 7, Вып. 2. – С.3-17.
10. Шрейдер Ю.А. Равенство, сходство, порядок. – М.: Наука, 1971.
11. Сахаров А.Н. Александр I. – М: Наука, 1998.
12. Wittgenstein L. Logisch-Philosophische Abhandlung // Annalen der Naturphilosophie. Vol. XIV. Parts 3/4. – Leipzig: Verlag Unesma, 1921. – P..185-262 / пер. Витгенштейн Л. Логико-философский трактат. М.: Издательство иностранной литературы, 1958.
13. Грязнов А.Ф. Витгенштейн // Новая философская энциклопедия. Т.1. – М: Мысль, 2000. – С. 406-408.
14. Chen P.P. The entity-relational model. Toward a unified view of data // ACM TODS. 1976. № 1. P. 9-36. / пер. Чен П. П.-Ш. Модель «сущность–связь» – шаг к единому представлению данных // СУБД. – 1995. – № 3. – С.137-158.
15. Михайлов А.И., Черный А.И., Гиляревский Р.С. Основы информатики. – М.: Наука, 1968.
16. Барахнин В.Б., Кожемякина О.Ю. Об автоматизации комплексного анализа русского поэтического текста // Труды Четырнадцатой Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (RCDL'2012). Переславль-Залесский, 15-18 октября 2012 г. – С. 213-217.
17. Самойлов Д. С. Книга о русской рифме. – М.: Художественная литература, 1982.
18. Барахнин В.Б., Федотов А.М. Уточнение терминологии, используемой при описании интеллектуальных информационных систем, на основе семиотического подхода // Известия вузов. Проблемы полиграфии и издательского дела. – 2008. – № 6. – С.73-81.
19. Колмогоров А.Н. Три подхода к определению понятия «количество информации» // Проблемы передачи информации. – 1965. – Т. 1, Вып. 1. – С.3-11.
20. Колмогоров А.Н. Теория информации и теория алгоритмов. – М.: Наука, 1987.
21. Ляпунов А.А. О соотношении понятий материя, энергия и информация // В кн.: А.А. Ляпунов Проблемы теоретической и прикладной кибернетики. – Новосибирск: Наука, 1980. – С. .320-323.
22. Heisenberg W. Der Teil und das Ganze. Gespräche im Umkreis der Atomphysik. – München, 1976.
23. Михайлов А.И., Черный А.И., Гиляревский Р.С. Основы информатики. – М.: Наука, 1968.
24. Добров Б.В., Лукашевич Н.В., Синицын М.Н., Шапкин В.Н. Разработка лингвистической онтологии по естественным наукам для решения задач информационного поиска // Труды Седьмой Всероссийской научной конференции «Электронные библиотеки: перспективные ме-тоды и технологии, электронные коллекции» (RCDL'2005). – Ярославль, 2005. – С. 70-79.
25. Welty C., McGuinness D., Uschold M., Gruninger M., Lehmann F. Ontologies: Expert Systems all over again // AAAI-1999 Invited Panel Presentation. – 1999.
26. Нариньяни А.С. Кентавр по имени ТЕОН: Тезаурус + Онтология // Труды международного семинара Диалог'2001 по компьютерной лингвистике и ее приложениям. Т. I. – Аксаково, 2001. – С. 184-188.
27. Барахнин В.Б., Леонова Ю.В., Федотов А.М. К вопросу о формулировке требований для построения информационных систем научно-организационной направленности // Вычислительные технологии. – 2006. – Т. 11. Специальный выпуск. – С. 52-58.
28. Сидорова Е.А. Онтологический подход к представлению знаний для задачи анализа текстовых ресурсов // Материалы Всероссийской конференции с международным участием «Знания – Онтологии – Теории» (ЗОНТ–07). Т. 1. – Новосибирск, 2007. – С. 221-228.
29. Бычков И.В., Ружников Г.М., Хмельнов А.Е., Шигаров А.О. Эвристический метод обнаружения таблиц в разноформатных докумен-тах // Вычислительные технологии. – 2009. – Т. 14, № 2. – С. 58-73.
30. Горобец Б.С. Советские физики шутят… Хотя бывало не до шуток. – М: Книжный дом «ЛИБРОКОМ», 2010.

Материал поступил в редакцию 24.09.14.

Сведения об авторах

БАРАХНИН ВЛАДИМИР БОРИСОВИЧ – доктор технических наук, доцент, старший научный сотрудник Института вычислительных технологий СО РАН, зав. кафедрой информационных технологий Высшего колледжа информатики Новосибирского государственного университета.
e-mail: bar@ict.nsc.ru

ФЕДОТОВ АНАТОЛИЙ МИХАЙЛОВИЧ – доктор физико-математических наук, профессор, член-корреспондент РАН, декан Факультета информационных технологий Новосибирского государственного университета, главный научный сотрудник Института вычислительных технологий СО РАН, г. Новосибирск.
e-mail: fedotov@sbras.ru

УДК [004.65 : 54] ВИНИТИ

О.М. Нефедов, С.В. Трепалин, Л.М. Королева, Ю.Е. Бессонов, Н.И. Чуракова

База структурных данных по химии ВИНИТИ РАН: проблемы поиска по фрагменту структуры*

Описан новый алгоритм поиска в Базе структурных данных по химии ВИНИТИ РАН как интеграция существующих алгоритмов релаксации и поиска с возвратом. Предложены новые фильтры для предварительной фильтрации данных, позволяющие реализовать поиск стереохимической информации в оптимизированной базе данных, хранящейся на Microsoft SQL Server.

Ключевые слова: *база структурных данных по химии, ВИНИТИ РАН, кодирование информации, информационный поиск, алгоритм поиска, информационный запрос, стереохимия, пользователи, программное обеспечение, изоморфизм подграфов, фильтрация данных*

ВВЕДЕНИЕ

База структурных данных по химии ВИНИТИ РАН (далее База СД) генерируется с 1975 г. За годы эксплуатации Базы СД сформировался программно-технологический комплекс, математическое, лингвистическое и информационное обеспечение. Значительным этапом в развитии Базы СД стало введение в эксплуатацию режима графической обработки структурной химической информации с помощью программного комплекса CBASE32. В результате была создана специальная система обработки, поиска, хранения, распознавания, использования химической информации [1, 2]. Программное обеспечение этой системы реализует на практике уникальные информационные модели координационных соединений, хиральных органических и элементоорганических соединений, которые адекватны современным химическим представлениям и во многом превосходят принятые в мировой практике стандарты. Внедрение процесса автоматического распознавания и определения хиральности углеродных атомов в сложных органических соединениях, содержащих до десяти асимметрических центров, обеспечивает точность и высокое качество поиска пертинентной информации.

Модернизация Базы СД предполагает создание эффективных механизмов поиска на основе разделения объектов химических структур на всех этапах обработки запроса пользователя.

В химических базах данных возможны различные виды поиска, выбор которых определяется задачами, стоящими перед пользователем (поиск по химической структуре, названиям, регистрационным номерам и т.д.). В основе представления химических веществ в виде химических структур лежит метод валентных схем. Согласно этому методу каждый атом имеет валентности, с помощью которых он может соединяться с другими атомами, образуя связи. Связи бывают одинарные, двойные, тройные, а также координационные. При создании запросов часто используются ароматические связи, но в точной структуре они представляются альтернацией одинарных и двойных связей. Не все химические соединения подчиняются методу валентных схем – исключения составляют, например, карбораны, металлы и сплавы, ионные кристаллы. Такие соединения не могут быть представлены в виде химической структуры. Для представления химической структуры белков, ДНК, РНК и других биологических объектов с большим молекулярным весом используются специальные методы, основанные, например, на повторяемости небольшого числа фрагментов (аминокислот, азотистых оснований), из которых состоят такие объекты.

Практика информационного обслуживания специалистов показывает, что к большинству низкомолекулярных химических соединений применимо понятие химическая структура. Принято различать следующие виды поиска по химической структуре: поиск по точной химической структуре; по фрагменту структуры; поиск по сходству; поиск по формуле Маркуша [3]; специальные поиски, например, поиск максимально общего фрагмента, поиск атом-центрированных свойств [4], таких как химические сдвиги в ЯМР спектрах [5]. Для реализации этих видов поиска используются соответствующие модели, схемы, алгоритмы.

АЛГОРИТМЫ ПОИСКА ХИМИЧЕСКИХ СТРУКТУР ПО ГРАФУ

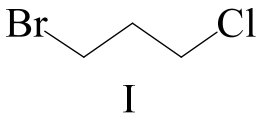
Представляется интересным рассмотреть возможность реализации поиска по фрагменту химической структуры. История осуществления такого поиска

* Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований, проект № 13-07-00488.

относится к 1959 г. [6], когда он был реализован Chemical Abstracts Service (CAS). Поиск по фрагменту структуры представляется в виде логической переменной, принимающей значение True, если заданный фрагмент входит в структуру, иначе False. Такая задача является NP-полной [7], и время ее решения растет в худшем случае экспоненциально с ростом сложности.

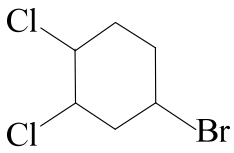
Основой всех известных алгоритмов решения задачи изоморфизма подграфу является в том или ином виде перебор с возвратом, а для сокращения числа вариантов перебора используются различные методы, включающие процедуры релаксации, предварительной сортировки элементов запроса, построения вспомогательных графов (модульное произведение, граф соответствий) [8–10].

Впервые алгоритм перебора с возвратом был опубликован Дж. Ульманом [11], а релаксационный алгоритм предложил Л. Китчен [12]. Многочисленные модификации этих алгоритмов с целью повышения эффективности и точности информационного поиска описаны в публикациях [13–21]. Использование алгоритма перебора с возвратом для поиска структурной химической информации по фрагменту 1-бром-3-хлорпропан (**I**) представлено на рис. 1–5. На первом этапе выбирается стартовый атом во фрагменте **I** (атом Cl) и аналогичный атом в содержащихся в базе данных структурах, в нашем примере в структуре 1-бром-3,4-дихлорциклогексан (**II**), такие атомы называются совмещенными (см. рис. 1 и 2).



I

Рис. 1. Заданный для поиска фрагмент структуры



II

Рис. 2. Структура, содержащая искомый фрагмент

Затем выбирается химическая связь во фрагменте **I** при только что совмещенном атоме и находится связь того же типа в структуре **II.** Если данная связь и во фрагменте и в структуре образована с идентичными атомами, то эти связи также считаются совмещенными (рис. 3 а). И такие действия далее последовательно повторяются для всех связей во фрагменте и в структуре (рис. 3 б, в). Предположим, что на некотором этапе не удается совместить связь и атом фрагмента со связью и атомом в структуре (рис. 3 г). В этом случае происходит последовательный возврат к предыдущему совмещенному атому (backtracking) (рис.3 д) и уже вторую связь при предыдущем совмещенном атоме в структуре пытаются совместить с фрагментом. При невозможности совмещения происходит отступ еще на одну связь (рис.4 а). Таким образом перебираются все возможные комбинации совмещений (рис. 4 б – д).

Если в результате поиска всех возможных комбинаций не удалось найти подходящих совмещений, то осуществляется перенос отнесения первых атомов во фрагменте и структуре (рис. 5 а) и процедура повторяется. Поиск считается выполненным успешно, если соотнесены все атомы фрагмента и структуры (рис. 5 в – е). Поиск считается безуспешным, если исчерпаны отнесения первых атомов фрагмента и структуры и не найдено ни одного полного отнесения.

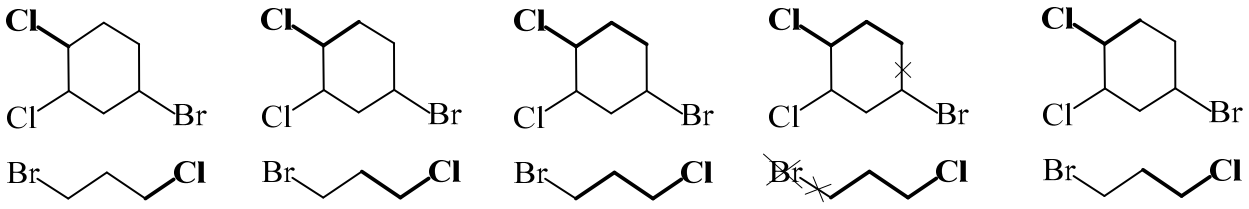


Рис. 3. Последовательность поиска совмещенных атомов и связей во фрагменте и структуре при использовании алгоритма перебора с возвратом. Первый этап. (Текущее отнесение атомов и связей выделено жирным шрифтом, а тупиковые ветви перечеркнуты)

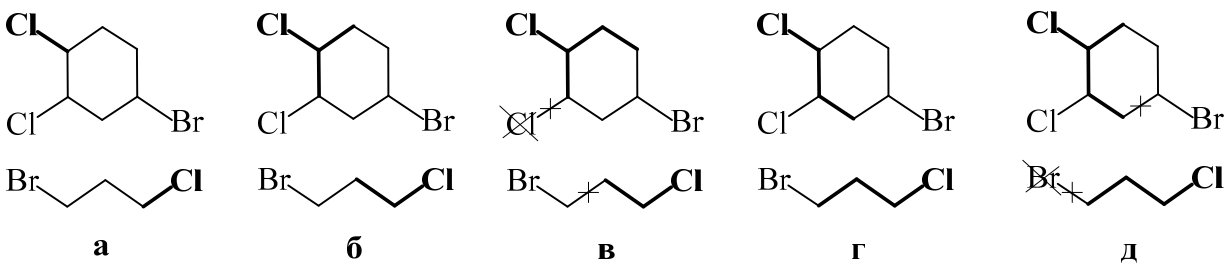


Рис. 4. Последовательность поиска совмещенных атомов и связей во фрагменте и структуре при использовании алгоритма перебора с возвратом. Второй и третий этапы. (Текущее отнесение атомов и связей выделено жирным шрифтом, а тупиковые ветви перечеркнуты)

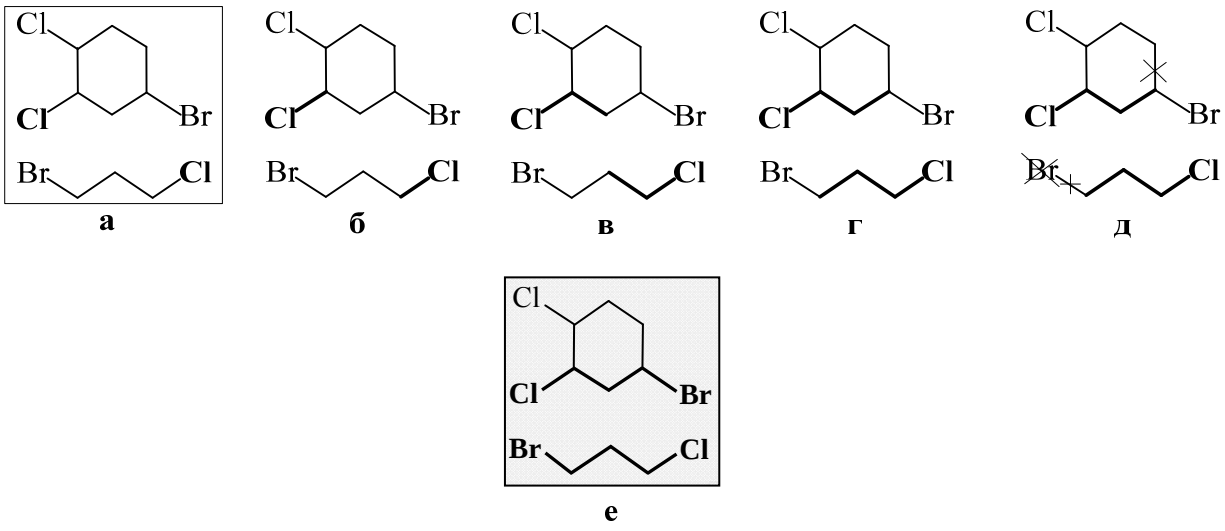


Рис. 5. Последовательность поиска совмещенных атомов и связей во фрагменте и структуре при использовании алгоритма перебора с возвратом. Заключительные этапы. (Текущее отнесение атомов и связей выделено жирным шрифтом, а тупиковые ветви перечеркнуты)

Несмотря на очевидное преимущество этого алгоритма, состоящего в том, что он позволяет определить соответствие атомов и связей во фрагменте и структуре (atom-atom mapping), он характеризуется значительными затратами времени и других ресурсов.

При поиске химической структуры по фрагменту на практике применяются также релаксационные алгоритмы. Детально рассмотрим механизм реализации релаксационного алгоритма на примере поиска той же структуры **II** по фрагменту **I** (рис. 6).

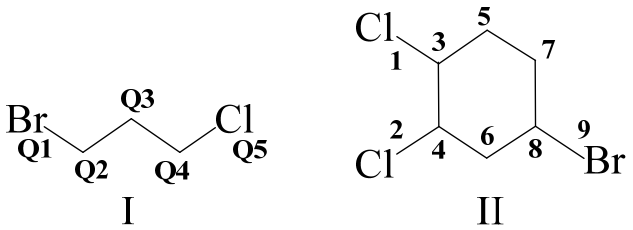


Рис. 6. Фрагмент **I** и структура **II** для иллюстрации работы релаксационного алгоритма (схема работы релаксационного алгоритма для примера на рис. 6 показана в табл. 1)

Как видно из рис. 6, для каждого атома во фрагменте **I**, например Br (Q1), формируется список атомов структуры **II**, которые могут быть отнесены к выбранному атому фрагмента (табл. 1, итерация № 0 (ноль)). Далее для каждого атома оценивается его ближайшее окружение. В данном примере имеется однозначное отнесение – единственный атом брома во фрагменте и структуре. Во фрагменте **I** атом брома (Q1) связан с единственным атомом углерода Q2, а в структуре **II,** соответственно, бром (9) связан с атомом углерода (8). Поэтому в столбце Q2 удаляются все возможные варианты, кроме атома 8. Поскольку атом 8 структуры **II** имеет однозначное отнесение, его исключают из отнесения к другим атомам фрагмента **I**. Хлор (Q5) во фрагменте **I** связан с углеродом Q4. В структуре **II** присутствуют два атома хлора – 1 и 2. Они связаны с атомами углерода 3 и 4, соответственно. Поэтому для атома углерода Q4 фрагмента **I** необходимо удалить позиции всех углеродов структуры **II**, кроме атомов, связанных с хлором – 3 и 4 (табл. 1, итерация № 1).

При последующем рассмотрении атомов второго окружения удаляется возможное отнесение атомов 5 и 7 структуры **II** к атому Q3 фрагмента **I**, поскольку, как атом хлора, так и атом брома во фрагменте **I** находятся на расстоянии двух связей от атома углерода Q3. При отнесении же атома углерода 5 структуры **II** к атому Q3 запроса получаем расстояние до атома брома от атома углерода 5 равное трем связям, а при отнесении атома 7 структуры к атому Q3 уже расстояние до атома хлора будет равно трем связям. Таким образом, полностью соответствует атому Q3 искомого фрагмента атом углерода 6 структуры **II.** Так генерируется итерация № 2 табл. 1. Учет третьего окружения атомов исключает отнесение атома 3 структуры к атому Q4 запроса, так как на расстоянии трех связей от атома Q4 должен находиться атом брома, а расстояние между атомом 3 структуры и атомом брома равно четырем связям. Таким образом, атом 4 структуры однозначно относится к Q4 и его необходимо исключить из отнесения к Q3 – итерация 3. Поэтапно повторяя дальнейшее рассмотрение все более удаленных соседей для атомов в структуре **I,** которые являются аналогами атома углерода Q4 и атома хлора Q5 фрагмента **I,** для нашего примера можно найти однозначное соответствие (табл. 1, итерация № 4). Если же при поиске соответствия какой-то из столбцов таблицы остается пустым, т. е. для какого-то атома запроса не находится соответствия, то поиск считается неудачным и структуры, содержащие искомый фрагмент, отсутствуют.

Таблица 1

Последовательность отнесения атомов запрос-структура в релаксационном алгоритме

№ итерации	Номера атомов фрагмента и списки отнесенных номеров атомов структуры				
	Q1	Q2	Q3	Q4	Q5
0	9	3,4,5,6,7,8	3,4,5,6,7,8	3,4,5,6,7,8	1,2
1	9	8	3,4,5,6,7,8	3,4	1,2
2	9	8	4,6	3,4	1,2
3	9	8	6	4	1,2
4	9	8	6	4	2

Релаксационный алгоритм позволяет значительно ускорить информационный поиск. Тем не менее, в некоторых случаях он не позволяет произвести однозначное соотнесение атомов структур с искомым фрагментом (пример приведен на рис. 7 и в табл. 2).

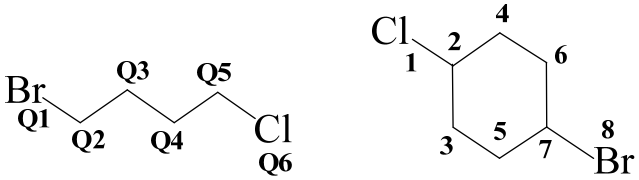


Рис. 7. Пример структуры для иллюстрации неоднозначного соотнесения Q3 и Q4 искомого фрагмента.

Таблица 2

Фрагмент неоднозначного соотнесения Q3 и Q4 и невозможности получить соответствие между атомами при использовании релаксационного алгоритма

№ итерации	Номера атомов фрагмента					
	Q1	Q2	Q3	Q4	Q5	Q6
5	8	7	5,6	3,4	2	1

Релаксационный алгоритм правильно определит, что к атому запроса Q3 может быть отнесен один из пары атомов 5 и 6 структуры, а к атому запроса Q4 – один из пары атомов 3 и 4, соответственно. Но однозначного ответа в этом случае не будет. Алгоритм с возвратом при соотнесении с парой атомов Q3+Q4 выберет случайным образом либо пару атомов 5+3 структуры, либо пару 6+4, которые являются равноценными, и аналогично определит соответствие атомов (atom-atom mapping).

Кроме этого, бывают случаи, когда неоднозначность проявляется при поиске циклических фрагментов, и в результате в ответе на такие запросы получаются циклы большего размера. На рис. 8 приведен пример, когда по запросу (рис.8а), релаксационный алгоритм находит структуру (рис.8б) [15].

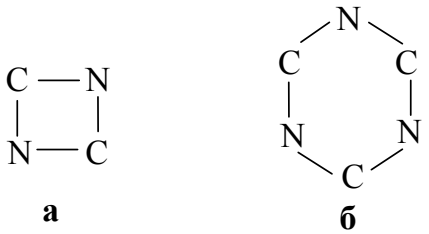


Рис.8. Пример некорректной работы релаксационного алгоритма: структура (а) находится как фрагмент структуры (б)

Для реализации так называемых классических задач определения изоморфного вложения (результат только True или False) релаксацию можно рассматривать как метод, позволяющий значительно упростить необходимый этап перебора.

Рассмотренные алгоритмы, наряду с преимуществами, характеризуются рядом недостатков, в том числе значительными временными затратами (алгоритм перебора с возвратом) или неоднозначностью при соотнесении атомов структуры и фрагмента (релаксация).

ПОИСК ПО ФРАГМЕНТУ СТРУКТУРЫ С ИСПОЛЬЗОВАНИЕМ СПЕЦИАЛЬНЫХ ФИЛЬТРОВ

На практике используют так называемые фильтры, которые позволяют оптимизировать информационный поиск. Например, если в запросе фрагмент содержит атом N (азота), то сразу производится процесс отсеивания всех структур в поисковом массиве, которые не содержат атом азота. Наибольшее распространение получили фильтры, в которых существует предопределенный набор молекулярных фрагментов. Фильтры, состоящие из молекулярных фрагментов, называются скринами (screens). Впервые использование скринов для ускорения поиска по фрагменту структуры было предложено М. Линчем и сотрудниками [22, 23]. В настоящее время наибольшее распространение получили скрины MDL [24]. Скрины хранятся в двоичном представлении в виде нескольких чисел типа целое (integer). Например, если используется 32-разрядная операционная система, то пяти целых чисел достаточно, чтобы сохранить информацию о наличии (или отсутствии) 5*32=160 предопределенных фрагментов. Обязательное усло-

вие при работе с фильтрами – необходимость сохранения скринов в базе данных. Если их генерировать во время поиска, то эффективность фильтров теряется. Поиск по предопределенным фрагментам (генерация скринов) осуществляется для вновь добавляемой записи и при редактировании существующей записи.

Для фильтрации химических структур первоначально выполняется поиск предопределенных фрагментов в запросе и формируется битовая строка запроса. Затем для каждой записи в базе проверяется наличие следующих соответствий:

bitsring(structure) AND
bitstring(query)=bitstring(query),

где bitsring(structure) – битовая строка химической структуры для выбранной записи, bitstring(query) – битовая строка запроса, AND – побитовый оператор “И”.

Если это условие истинно, то далее осуществляется поиск вложения графа запроса в граф этой химической структуры.

Увеличение числа скринов позволяет улучшить фильтрацию данных, но приводит к необходимости хранить большое количество дополнительной информации. В этом случае битовые строки часто сжимаются, т. е. одному биту в строке соответствует хотя бы один из нескольких предопределенных фрагментов. Такие сжатые строки называются фингерпринт (fingerprint). Наиболее распространенными являются DAYLIGHT фингерпринты [25].

Еще один способ фильтрации, описанный в литературе – формирование базы данных в виде дерева [26, 27], вершины которого содержат иерархический список фрагментов. Такой способ фильтрации сразу дает набор отфильтрованных записей, в которых следует выполнять поиск по графам. Но в случае добавления новой записи или изменения существующей необходимо пересчитывать заново все дерево. Кроме этого, хранение инвертированных индексов требует много ресурсов. Этот метод фильтрации применяется, как правило, для поиска информации в небольших, редко изменяемых базах данных.

РАЗРАБОТКА РАЦИОНАЛЬНОГО МЕТОДА ОБРАБОТКИ ПОИСКОВЫХ ЗАПРОСОВ В БАЗЕ СД

Перед началом поиска в базе структурных данных формируется двоичное представление запроса: создаются массивы fAtom и fBond. Каждый элемент массива fAtom содержит позицию атома в Периодической таблице Д.И. Менделеева, заряд атома, признак радикала и 2D-координаты. Они используются как для изображения химической структуры, так и для поисков с учетом стереохимии. Каждый элемент массива связей fBond содержит тип связи: одинарная, двойная, тройная, координационная, Up-, Down-, стерео-неопределенная и индексы пары атомов из массива fAtom, которые образуют связь.

Для ускорения поиска по фрагменту структуры нами было разработано несколько оригинальных способов предварительной обработки запросов.

Мы предлагаем перед началом поиска запрос сортировать, кроме того использовать специальный синтаксис в SQL-запросах. Сортировка состоит в изменении нумерации атомов и связей запроса таким образом, чтобы номера атомов и связей совпадали с очередностью их отнесения при работе backtracking-алгоритма. Для этого в запросе выбирается неуглеродный атом с максимальным числом связей, при отсутствии такового – максимально координированный углерод. Номер этого атома заносится в первый элемент формируемого массива QueryAQTested. Все связи, присоединенные к первому атому, последовательно заносятся в массив BSTESTED, а номера атомов, образующих второй конец связей, заносятся последовательно в QueryAQTested. Затем формируется список соседей для соседей, и этот процесс повторяется до тех пор, пока не возникнет ситуация, когда новых соседей не будет найдено.

При отнесении новой связи возможны два варианта: 1) второй атом, присоединенный к связи, является новым, еще не отнесенным, 2) данная связь образует цикл, и второй атом уже занесен в массив QueryAQTested. Эта информация заносится в массив QueryAGen, который для каждой связи содержит номер второго атома. В случае замыкания цикла этот номер совпадет с номером уже существующего атома в массиве QueryAQTested, иначе он равен нулю.

После этого осуществляется перенумерация массивов fAtom и fBond в запросе следующим образом: если атом имеет индекс K в массиве QueryAQTested, а связь имеет индекс N в массиве BSTESTED, то после перенумерации они будут иметь индекс K в массиве fAtom и индекс N в массиве fBond, соответственно. Далее запрос проверяется на связность. При этом контролируется, все ли атомы вошли в список QueryAQTested. Поиск по несвязным фрагментам требует специального подхода.

В разработанном нами алгоритме формируются данные для поиска фрагментов с определенной конфигурацией заместителей относительно двойной связи. Этот поиск позволяет осуществлять стерео поиск относительно двойной связи, т.е. найти Z- и E- изомеры. Такой поиск реализован введением записи (структуры – C++ термин) QuerySTAS, которая содержит пару связей, присоединенных к двойной связи и их скалярное произведение. При соотнесении этих связей в химической структуре будет проверяться совпадение знаков скалярных произведений соответствующих связей в структуре со значением в записи QuerySTAS.

Модель Базы СД, в которой осуществляется поиск, была описана ранее [1, 28, 29]. Химические структуры хранятся в поле Structure в виде BLOB (Binary Large Object) записей в таблице Compound. Фрагмент списка полей таблицы Compound приведен в табл. 3 (поля, содержащие служебную информацию и не относящиеся к химической структуре не приведены).

Помимо BLOB-записей (поле Structure, тип данных image) таблица Compound содержит ряд вычисляемых полей и их расчет осуществляется автоматически при добавлении (изменении) химической структуры. К ним относятся:

1) InChI key – хэш код структуры [30] используется для поиска по точной структуре, подробнее его использование мы описывали ранее [1];.

2) Screens1, …, Screens5 – 32-битовые целые числа, в которых каждому биту соответствует признак наличия (отсутствия) предопределенного заранее фрагмента. Значения этих полей используются для фильтрации записей (см. далее);

3) Molweight – молекулярный вес. Используется для поиска по молекулярному весу, это довольно частый запрос химиков. Кроме того, поскольку молекулярный вес хранится с 4-5 значащими цифрами после запятой, он представляет собой удобный индекс для поиска по точной молекулярной формуле соединения;

4) NC, NH, NN, NO – число атомов углерода, водорода, азота и кислорода соответственно. Используются для поиска по запросам на элементный состав в диапазонах от – до;

5) FLAGS – набор признаков присутствия (отсутствия) таких элементов, как сера, фосфор, хлор, фтор, бром, йод, кремний, бор, металлы. Используется для поиска по молекулярной формуле;

6) Molformula – молекулярная формула в виде строки. Используется как для показа пользователю, так и для поиска по формуле. Разборка молекулярной формулы при выполнении запроса поиска занимает заметное время и для ускорения этого типа поиска предварительно используются описанные выше индексы NC, NH, NN, NO, FLAGS;

7) ScreenCount – число ненулевых бит в индексах Screen1, ..., Screen5. Используется для поиска по сходству.

Таблица 3

Фрагмент таблицы Compound

№ п/п	Название поля	Тип поля
1	CompoundID	int
2	InChIKey	char(27)
3	Screens1	int
4	Screens2	int
5	Screens3	int
6	Screens4	int
7	Screens5	int
8	Molweight	real
9	NC	smallint
10	NH	smallint
11	NN	smallint
12	NO	smallint
13	FLAGS	int
14	Structure	image
15	Molformula	varchar(50)
16	ScreenCount	smallint

Нами была создана хранимая процедура на MS SQL сервере, которая на вход принимает текст с химической структурой запроса в формате MDL Molfile [31]. Запрос конвертируется в матрицу связности, для чего формируются описанные выше двоичные массивы fAtom и fBond, которые заносятся в BLOB-поле. Перед расчетами индексов осуществляется стандартизация химической структуры, при которой удаляются атомы водорода в явном виде, осуществляется конверсия семиполярных и стереосвязей, очищаются метки изотопов. Далее из хранимой процедуры вызывается COM-объект, который рассчитывает все индексы. В частности, осуществляется поиск по предопределенным фрагментам, которые созданы на основе 166 предопределенных MDL фрагментов и признаков соединений [24].

С целью повышения эффективности фильтрации нами были усовершенствованы правила MDL.

1. Было удалено 11 признаков MDL. Среди удаленных признаков MDL: 103 < ATOMIC NO. < 256 (N2), QAAA@1 (N8), GROUP IIIA (B...) (N18), OTHER(N44), CHARGE(N49), QAAAA@1(N83), QAAAAA@1(N98), FRAGMENTS(N166).

2. Объединены следующие признаки: ACTINIDE, GROUP IIIB, IVB (Sc...), LANTHANIDE, GROUP VIII (Fe...) (N4 N5 N6 N9), а также ACH2AAACH2A и ACH2AACH2A (N128, N129).

3. Было добавлено 5 новых признаков. Структуры новых фрагментов-признаков представлены на рис. 9.

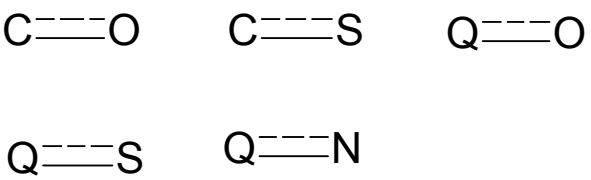


Рис. 9. Добавленные фрагменты-признаки для CBASE. Все связи ароматические, Q – гетероатом

Добавление этих фрагментов (см. рис. 9) потребовалось с целью однозначного представления данных для 5-ти членных циклов с двумя двойными связями и гетероатомом, например, индазола. В программе ISIS (MDL), для которой были разработаны упомянутые выше 166 признаков [24], такие фрагменты рассматриваются как чередование одинарных и двойных связей, вследствие чего их нельзя найти по запросу «Ароматическая связь». В CBASE [32] такие фрагменты в соответствии с правилом Хюккеля – 4n+2 рассматриваются как ароматические соединения. Структуры, аналогичные приведенным на рис. 10, программа ISIS считает разными, а в CBASE они считаются одинаковыми, и это находится в согласии с экспериментальными химическими данными, поскольку изомеров индазол не имеет.

Рис. 10. Два представления структуры индазола, которые ISIS различает как два разных соединения. Первая структура – 1H-индазол, вторая структура – 2H-индазол

Таким образом, для фильтрации при поиске по фрагменту структуры нами используются 160 при-

знаков и фрагментов. Уменьшение числа признаков слабо влияет на эффективность фильтрации, зато они упаковываются в пять 32-битовых целочисленных значений @ScreenData1, ..., @ScreenData5.

Далее для фильтрации выполняется следующий SQL запрос:

```
SET @cursorcreate=' DECLARE search_cursor CUR-
SOR READ_ONLY FOR SELECT Com-
pound.CompoundID, Compound.Structure FROM Com-
pound WHERE ((Compound.Screens1 &
'+convert(varchar(16),@ScreenData1)+') =
'+convert(varchar(16),@ScreenData1)+') and ((Com-
pound.Screens2 &
'+convert(varchar(16),@ScreenData2)+') =
'+convert(varchar(16),@ScreenData2)+') and ((Com-
pound.Screens3 &
'+convert(varchar(16),@ScreenData3)+') =
'+convert(varchar(16),@ScreenData3)+') and ((Com-
pound.Screens4 &
'+convert(varchar(16),@ScreenData4)+') =
'+convert(varchar(16),@ScreenData4)+') and ((Com-
pound.Screens5 &
'+convert(varchar(16),@ScreenData5)+') =
'+convert(varchar(16),@ScreenData5)+')
```

Это означает, что выполняется пять раз операция побитового (bitwise) AND.

При использовании этого SQL запроса потребуется перебор всей базы данных. Поэтому нами была предложена следующая модификация этого SQL запроса, а именно – в секцию WHERE добавляется следующее выражение:

```
'…WHERE (Compound.Screens1 >=
'+convert(varchar(16),@ScreenData1)+') AND  (Com-
pound.Screens2 >=
'+convert(varchar(16),@ScreenData2)+') AND (Com-
pound.Screens3 >= '+convert(varchar(16),
@ScreenData3)+') AND (Compound.Screens4 >=
'+convert(varchar(16),@ScreenData4)+') AND (Com-
pound.Screens5 >=
'+convert(varchar(16),@ScreenData5)+')…
```

Поля Screen1, ..., Screen5 декларированы нами как индексные (не уникальные). Индексные поля сортируются MS SQL сервером и если позволяют ресурсы, то целиком загружаются в оперативную память. Сортировка означает, что для поиска по индексным полям используется быстрый алгоритм бисекций со скоростью сходимости, пропорциональной $\log_2 N$, где N - число записей в базе – т. е. практически [33] не зависит от размера базы данных. При этом фильтрация частично выполняется сравнением индексных полей, а уже далее при успешном сравнении происходит операция побитового AND. После фильтрации для найденных записей необходимо осуществить процедуру поиска подграфа в графе.

Разработанный нами алгоритм представляет собой сочетание алгоритмов релаксации и перебора с возвратом, что позволяет обеспечить более эффективный информационный поиск по фрагменту структуры. Для этого первоначально формируется пара логических матриц: AEQ матрица эквивалентности атомов и BEQ матрица эквивалентностей связей, которые содержат True, если данный атом (связь) структуры может быть совмещен с данным атомом (связью) запроса, иначе False. Затем используется алгоритм релаксации первого порядка. Если одинарная связь в запросе связывает атомы C и Cl, а одинарная связь в структуре связывает атомы C и C, то в матрице BEQ ставится значение False, несмотря на совпадение типов связей. Если какой-либо атом запроса (или структуры) имеет единственно возможное отнесение, то всем элементам матрицы AEQ для всех остальных атомов присваивается значение False. И, наконец, проверяется, чтобы все столбцы и строки матрицы имели как минимум один True элемент, иначе искать бесполезно. Таким образом, реализуется часть алгоритма релаксации.

Далее используется алгоритм перебора с возвратом (backtracking). Для этого формируется массив, содержащий список атомов в структуре, которые могут быть отнесены к первому атому запроса, прошедшего предварительный процесс обработки – «сортировки запроса». Как было указано выше, обычно первым атомом в отсортированном запросе является максимально координированный неуглеродный атом. Затем выбираем первый атом структуры в сформированном массиве и принимаем, что он соответствует первому атому запроса. После этого используем метод DirectBondAssignment, который осуществляет операции соотнесения последовательно для первой, второй и последующих BNQ связей в запросе и, соответственно, для связанных с этими связями атомов (если есть новые) или проверяет замыкание цикла (если при соотнесении связи в запросе новый атом не образуется, а происходит замыкание). В процессе соотнесения связей, соседних с двойной связью, и при этом, если эта двойная связь содержит метку учета геометрии, вызывается метод SProduct, который проверяет совпадения знаков скалярных произведений уже соотнесенных связей.

При успешном соотнесении связи метод Direct BondAssignment возвращает True и вновь вызывается из fragmentSearch, но уже с требованием соотнести следующую связь. Если же связь и связанный атом соотнести не удается, то возвращается False и это означает, что атом, из которого исходит связь BNQ, соотнесен неверно. Все его атрибуты инициализируются, и вновь вызывается DirectBondAssignment, но уже для связи BNQ-1 с требованием найти другое соотнесение.

Если в результате выполнения этих процедур DirectBondAssignment все связи запроса будут успешно соотнесены со связями химической структуры, то это будет означать, что найдена структура, содержащая искомый фрагмент. Если же счетчик BNQ вернется к первой связи, то это будет значить, что первый атом запроса и первый атом структуры соотнесены неверно. В этом случае совмещаются следующий атом структуры с первым атомом в запросе. Процедура повторяется до тех пор, пока либо не будут исчерпаны все варианты соотнесения первого атома и счетчик вернется к первой связи (неуспешный поиск), либо пока число соотнесенных связей станет равным числу связей в запросе и DirectBondAssignment вернет True – успешный поиск. Поскольку в разработанном нами алгоритме

поиска везде фиксируется, какая именно связь в запросе соотнесена с какой связью в структуре и какой именно атом в запросе соотнесен с каким атомом в структуре, то после успешного окончания поиска становится доступным соответствие атомов и связей (atom-atom mapping).

Последовательность создания и обработки требования пользователя представлена на рис. 11.

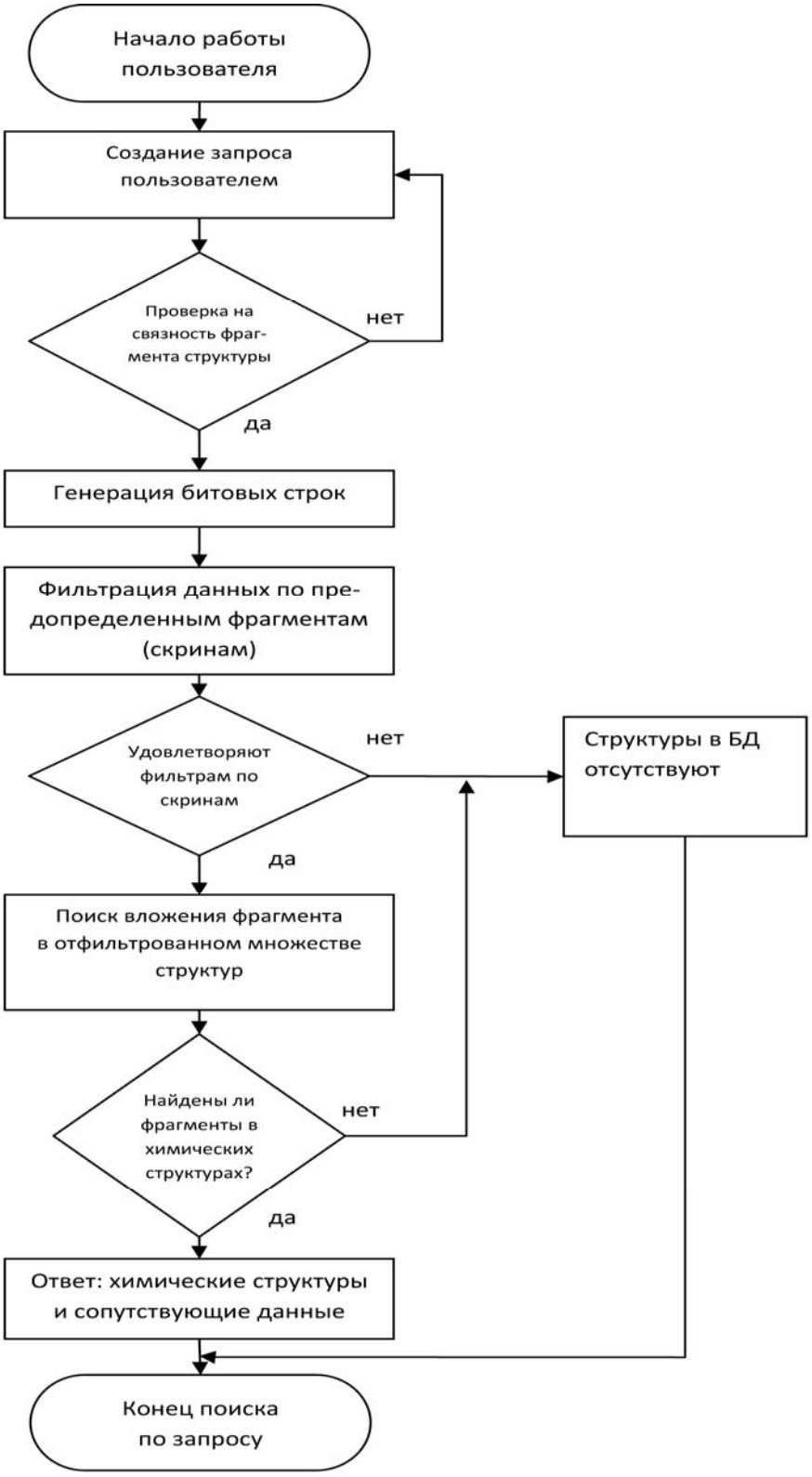


Рис. 11. Алгоритм обработки запроса пользователя при поиске по фрагменту химической структуры в Базе структурных данных по химии

С целью реализации разработанного в ВИНИТИ РАН алгоритма поиска химической информации в Базе структурных данных по химии визуальный интерфейс создания запроса был дополнен опциями поиска по фрагменту структур. Для создания интерфейса пользователя ранее нами был использован JME редактор структур [34]. Однако, в феврале 2014 г. компания Oracle [35] – владелец языка интернет-программирования Java в новом релизе запретил работу неподписанных Java Applets. В результате JME редактор структур перестал работать, и мы с любезного разрешения Peter Ertl заменили его на JSME-редактор [36] – аналогичный продукт, написанный на языке Java Script.

Интерактивный доступ к Базе структурных данных по химии ВИНИТИ РАН осуществляется через ресурс http://chem.viniti.ru/

ОСОБЕННОСТИ ПОИСКА СТЕРЕОХИМИЧЕСКОЙ ИНФОРМАЦИИ

Задача поиска структурной химической информации значительно усложняется, если химическая структура содержит стереохимические метки, отражающие особенности пространственного строения молекулы. В разработанной нами программе реализован поиск стереохимической информации для следующих запросов пользователя:

- задание в запросе стереоконфигурации для двойных связей;
- задание в запросе стереоконфигурации хиральных атомов с Up- и Down-связями [37].

Поиск стереоконфигурации относительно двойной связи осуществляется путем анализа скалярного произведения SProduct направляющих векторов связей, присоединенных к двойной связи.

Для учета стереохимических конфигураций атомов со стереосвязями, мы воспользовались понятием атомных хиральностей [38]. Традиционно хиральности описываются R- или S-конфигурациями, которые рассчитываются по правилу старшинства заместителей Кох-Ингольд-Прелога (CIP) [39]. При описании атомных хиральностей в нашем случае старшинство заместителей при хиральном атоме, в отличие от правила CIP, определяется порядковым номером атома в массиве fAtom. Чем больше номер – тем старше атом. В ряде случаев может возникнуть парадоксальная ситуация: при изменении внутренней нумерации атомов атомная хиральность может меняться на противоположную. Решение этого парадокса заключается в том, что для расчета атомной хиральности в структуре используется нумерация атомов в запросе, которая через соответствие атомов (atom-atom mapping) переносится на структуру.

Предлагаемый в литературе [38] подход к определению атомной хиральности позволяет решить главную проблему поиска по фрагменту структуры: невозможность использовать правила CIP для определения стереоконфигурации запроса. Если запрос содержит лишь фрагмент структуры, то во многих случаях невозможно провести расчет хиральностей с использованием правила CIP, поскольку невозможно вычислить старшинство заместителей. Кроме того, так как выбранная нумерация атомов в молекуле уникальна, т. е. любые два атома имеют различающиеся номера, то если у атома есть стереосвязь, его хиральность всегда отличается от нуля, даже если группы, связанные с ним, являются симметричными. При использовании правил CIP в этом случае получается нулевая хиральность. Наличие всегда ненулевой хиральности позволяет успешно выполнять поиск стереоизомеров, например, таких соединений, как транс-1,4-диметилциклогексан (рис. 12).

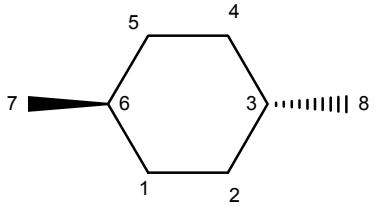


Рис. 12. транс-1,4-Диметилциклогексан
(Показана внутренняя нумерация атомов, которая не связана с нумерацией по IUPAC номенклатуре)

На рис. 12 представлена нумерация атомов в массиве fAtom. Как видно все они различаются, следовательно, атомы со стереосвязями имеют ненулевую атомную хиральность. Таким образом, порядковая нумерация всех атомов в структуре (внутренняя перенумерация атомов) и формирование рабочего массива fAtom позволяют избежать потерь исходной информации.

Это привело к необходимости произвести следующие изменения в алгоритме поиска стереохимической информации. Если запрос содержит стереосвязи, то при формировании матрицы эквивалентности атомов AEQ таким атомам запроса ставятся в соответствие только те атомы структуры, которые тоже имеют стереосвязи. Это позволяет значительно улучшить фильтрацию на этапе работы алгоритма релаксации. Далее в методе DirectBondAssignment после соотнесения последней связи атома запроса, для него вычисляется атомная хиральность и номера атомов запроса заменяются на номера атомов химической структуры. Это легко достигается, поскольку имеется соответствие между атомами (atom-atom mapping). Атомная хиральность запроса, вычисленная таким образом, должна совпадать с хиральностью этого атома в структуре. В противном случае соотнесение последней связи считается неудачным и происходит возврат к началу поиска.

Разработанный нами поиск стереохимической информации дополнен опцией, учитывающей относительную конфигурацию. При этом реализована следующая процедура: если поиск фрагмента с относительной конфигурацией в структуре был неудачным, то все стереосвязи инвертируются – Up заменяются на Down и наоборот; также предусмотрена опция «неопределенная атомная стереоконфигурация» для поиска с Either стереосвязью. При совмещении такого атома с атомом, содержащим Up- или Down-связь, существует возможность как разрешить такое совмещение, так и запретить его.

ЗАКЛЮЧЕНИЕ

1. Изучено современное состояние проблемы поиска по фрагменту структуры в базе структурных данных по химии на основе обзора литературы по проектированию и эксплуатации баз данных, использующих различные алгоритмы поиска.

2. Разработан алгоритм поиска по фрагменту структуры как комбинация алгоритмов релаксационного и перебора с возвратом. Он позволяет по фрагменту структуры осуществлять поиск стереохимической информации на основе атомных хиральностей.

3. Предложен и описан алгоритм сортировки запроса пользователя в том порядке, как он будет выполняться при работе алгоритма перебора с возвратом. Для ускорения поиска предложено и реализовано использование индексов для полей MS SQL сервера.

4. Для оптимизации информационного поиска химической структурной информации использованы специальные фильтры, состоящие из молекулярных фрагментов. Показана возможность применения специального нового набора скринов для корректного и однозначного описания 5-ти членных гетероциклов с двумя и более атомами азота.

5. На основе рабочего массива информационных и процедурных требований пользователей разработан алгоритм обработки запроса пользователя для релевантного поиска структурной химической информации по фрагменту структуры.

СПИСОК ЛИТЕРАТУРЫ

1. Нефедов О.М., Трепалин С.В., Королева Л.М., Бессонов Ю.Е. Быстрый поиск точных химических структур в больших базах данных с использованием InChI Key кодировки структур // Научно-техническая информация. Сер. 2. – 2013. – № 12. – С.27-33.
2. Королева Л.М., Федоровская М.А., Чуракова Н.И., Фельдман Б.С., Лазарев В.В., Качурина Н.В. Разработка компьютерной программы автоматической проверки систематических названий химических соединений как средство повышения качества формирования Базы структурных данных по химии ВИНИТИ РАН. – М., 2014. – 16 с. – Деп. в ВИНИТИ РАН 01.07.2014, № 183-В2014.
3. Markush E. The process for manufacture of dyes which comprises coupling with a halogen-substituted pyralazone, a diazotized unsulphonated material selected from the group consisting of aniline, homologues of aniline, and halogen substitution products of aniline, patent 1924. – URL: http://www.answers.com/ topic/markush-structure#ixzz35cXX3yMt (дата обращения: 18.08.2014)
4. Bremser W. HOSE – a novel substructure code // Anal. Chim. Acta. – 1978. – Vol. 103. – P. 355-365.
5. Trepalin S.V., Yarkov A.V., Dolmatova L.M., Zefirov N.S., Finch S.E. WinDat: An NMR Database Compilation Tool, User Interface and Spectrum Libraries for Personal Computers // J.Chem. Inf. Comput. Sci. – 1995. – Vol. 35. – P. 405-412.
6. Dyson G. M. Research Expansion at the Chemical Abstracts Service // Chem. Eng. News. – 1959. – Vol. 37, № 36. – P.128-131.
7. Cook S. The complexity of theorem-proving procedures // Proceedings of the 3rd Annual ACM Symposium on Theory of Computing. – 1971. – P. 151-158.
8. Визинг В.Г. Сведение проблемы изоморфизма и изоморфного вхождения к задаче нахождения неплотности // Труды III Всесоюз. конф. по проблемам теоретической кибернетики. – Новосибирск, 1974.– С. 124.
9. Бессонов Ю.Е. О решении задачи поиска наибольших пересечений графов на основе анализа проекций модульного произведения // Алгоритмический анализ структурной информации (Вычислительные системы). – 1985. – Вып. 112. – С. 3-22.
10. Бессонов Ю.Е. Использование свойств решеточно полных графов для поиска общих подструктур. – М., 2014. – 13 с. – Деп. в ВИНИТИ РАН №32-В2014.
11. Ullmann J.R. An algorithm for subgraph isomorphism // Journal of the ACM . – 1976. – Vol. 23, № 1. – P. 31-42.
12. Kitchen L. Relaxation Applied to Matching Quantitative Relational Structures // Trans. Systems, Man and Cybernetics. – 1980. – Vol.10. – P. 96-101.
13. Dittmar P. G., Farmer N. A. , Fisanick W., Haines v, Mockus J. The CAS ONLINE Search System. 1. General System Design and Selection, Generation, and Use of Search Screens // J. Chem. Inf. Comput. Sci. – 1983. – Vol.23. – P. 93-102.
14. Attias R. DARC Substructure Search System: A New Approach to Chemical Information // J. Chem. Inf . Comput. Sci. – 1983. – Vol. 23. – P. 102-108.
15. Scolley A.V. A Relaxation Algorithm for Generic Chemical Structure Screening // J. Chem. Inf. Comput. Sci. – 1984. – Vol. 24. – P. 235-241.
16. Hicks M.G., Jochum K. Substructure Search Systems. 1. Performance Comparison of the MACCS, DARC, HTSS, CAS Registry MVSSS, and S4 Substructure Search Systems // J. Chem. Inf. Comput. Sci. – 1990. – Vol. 30. – P. 191-199.
17. Hagadone T.R. Molecular Substructure Similarity Searching: Efficient Retrieval in Two-Dimensional Structure Databases // J. Chem. Inf. Comput. Sci. – 1992. – Vol. 32. – P. 515-521.
18. Bartmann A., Maier H., Walkowiak D., Roth B., Hicks M.G. Substructure Searching on Very Large Files by Using Multiple Storage Techniques // J. Chem. Inf. Comput. Sci. – 1993. – Vol. 33. – P. 539–541.
19. Xu J. GMA: A Generic Match Algorithm for Structural Homomorphism, Isomorphism, and Maximal Common Substructure Match and Its Applications // J. Chem. Inf. Comput. Sci. – 1996. – Vol. 36. – P. 25-34.

20. Ozawa K., Yasuda T., Fujita S. Substructure Search with Tree-Structured Data // J. Chem. Inf. Comput. Sci. – 1997. – Vol. 37. – P. 688-695.

21. Vogt J., Vogt N., Kramer R. Visualization and Substructure Retrieval Tools in the MOGADOC Database (Molecular Gasphase Documentation) // J. Chem. Inf. Comput. Sci. – 2003. – Vol. 43. – P. 357-361.

22. Crowe J. E., Lynch M. F., Town W. G. Analysis of structural characteristics of chemical compounds in a large computer-based file. Part I. Non-cyclic fragments // J. Chem. Soc. C. – 1970. – P. 990-996.

23. Adamson G. W., Cowell J., Lynch M. F., McLure A. H. W., Town W. G., Yapp A. M. Strategic Considerations in the Design of a Screening System for Substructure Searches of Chemical Structure Files // J. Chem. Doc. – 1973. – Vol. 13. – P. 153-157.

24. Durant J.L, Leland B.A., Henry D.R., Nourse J.G. Reoptimization of MDL Keys for Use in Drug Discovery // J. Chem. Inf. Comput. Sci. – 2002. – Vol. 42. – P. 1273-1280.

25. Daylight Theory: fingerprints. – URL: http://www.daylight.com/dayhtml/doc/theory/theory.finger.html (дата обращения: 18.08.2014)

26. Nagy M.Z., Kozics S., Veszpremi T., Bruck P. Substructure search on very large files using tree structured data bases, Chemical Structure: The International Language of Chemistry, Wendy Warr (Ed.). – Springer-Verlag, 1988. – P. 127-130.

27. Smellie A. Compressed Binary Bit Trees: A New Data Structure For Accelerating Database Searching // J. Chem. Inf. Model . – 2009. – Vol. 49. – P. 257-262.

28. Воронежева Н.И., Трепалин С.В., Чуракова Н.И. Нечаева К.С., Королева Л.М. Глоссарий как элемент стандартизации ввода данных в программном комплексе CBASE32. // Научно-техническая информация. Сер. 2. –2007. – № 6. – С. 19-24.

29. Королева Л.М., Чуракова Н.И., Федоровская М.А., Бессонов Ю.Е., Кирьянова Н.С., Фельдман Б.С., Трепалин С.В. Использование АРМ «Администратор Глоссария» для актуализации базы структурных данных по химии ВИНИТИ РАН. – М., 2014. – 15 с. – Деп. в ВИНИТИ РАН 14.04.2014, № 95-В2014.

30. Stein S. E., Heller S. R, Tchekhovskoi D. An Open Standard for Chemical Structure Representation: The IUPAC Chemical Identifier // Proceedings of the 2003 International Chemical Information Conference, Nimes, France, 19 – 22 October 2003. – Infonortics, 2003. – P. 131-143.

31. Dalby A., Hourse J. G., Hounshell W. D., Gurchurst A.K.I., Grier D. L., Leland B. A., Laufer J. Description of several chemical structure file formats used by computer programs developed at Molecular Design Limited // J. Chem. Inf. Comput. Sci. – 1992. – V. 32. – P. 244-255.

32. Алфимов М.В., Авакян В.Г., Трепалин С.В., Воронежева Н.И., Чуракова Н.И. Универсальная программная оболочка для создания баз данных химических соединений и реакций // Доклады РАН. – 1999. – Т.366, № 5. – С.639-642.

33. Burden R L., Faires J. D. The Bisection Algorithm. Numerical Analysis (3rd ed.). – PWS Publishers, 1985. – 676 p.

34. JME Molecular Editor Applet. – URL: http://www.molinspiration.com/jme/index.html (дата обращения: 18.08.2014)

35. Oracle. Hardware and Software Engineering to Work Together. – URL: http://www.oracle.com/index.html (дата обращения: 18.08.2014)

36. Bienfait B., Ertl P. JSME: a free molecule editor in JavaScript // Journal of Cheminformatics. – 2013. – Vol.5, № 24. – P. 1-6.

37. Немировская И.Б., Трепалин С.В., Королева Л.М. Представление стереохимической информации в Базе структурных данных ВИНИТИ РАН. // Научно-техническая информация. Сер. 2. –2006. – № 4. – С. 1-6.

38. Moreau G. Atomic Chirality, a Quantitative Measure of the Chirality of the Environment of an Atom // J. Chem. Inf. Comput. Sci. – 1997. – Vol. 37. – P. 929-938.

39. Cahn R.S., Ingold C.K., Prelog V. Specification of Molecular Chirality // Angew. Chem. Int. Ed. – 1966. – Vol. 5, № 4. – P. 385–415.

Материал поступил в редакцию 03.09.14.

Сведения об авторах

НЕФЕДОВ Олег Матвеевич – академик РАН, доктор химических наук, советник Президиума РАН, Москва
e-mail: onefedov@ras.ru

ТРЕПАЛИН Сергей Владимирович – кандидат химических наук, ведущий научный сотрудник ФГБУН Институт физиологически активных веществ РАН
e-mail: trep@chemical-block.com

КОРОЛЕВА Любовь Михайловна – кандидат химических наук, зав. отделением научной информации по проблемам химии и наук о материалах ВИНИТИ РАН
e-mail: lkorol@viniti.ru

БЕССОНОВ Юрий Ефимович – кандидат технических наук, зав. Отделом программного обеспечения и сопровождения информационных систем по химии ВИНИТИ РАН
e-mail: bessonov-ye@rambler.ru

ЧУРАКОВА Наталия Исааковна – кандидат химических наук, зав. сектором структурной информации по биоорганической химии ВИНИТИ РАН
e-mail: nichurak@rambler.ru

Указатель статей, опубликованных в сборнике «Научно-техническая информация», и Авторский указатель за 2014 год*

Указатель статей

* 5 означает номер сборника, (1) – серию, 3 – страницу

ДОКУМЕНТАЛЬНЫЕ ИСТОЧНИКИ ИНФОРМАЦИИ

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

ИНФОРМАЦИОННЫЕ СИСТЕМЫ

ИНФОРМАЦИОННЫЙ ПОИСК

ИНФОРМАЦИОННЫЕ ЯЗЫКИ

ИНФОРМАЦИОННЫЙ АНАЛИЗ

АВТОМАТИЗАЦИЯ ОБРАБОТКИ ТЕКСТА

СПРАВОЧНО-ИНФОРМАЦИОННЫЙ РАЗДЕЛ

Авторский указатель

Федеральное государственное бюджетное учреждение науки
ВСЕРОССИЙСКИЙ ИНСТИТУТ НАУЧНОЙ И ТЕХНИЧЕСКОЙ ИНФОРМАЦИИ РОССИЙСКОЙ АКАДЕМИИ НАУК

предлагает научным работникам, аспирантам и другим специалистам в области естественных, точных и технических наук, желающим быстро и эффективно опубликовать результаты своей научной и научно-производственной деятельности, использовать способ публикации своих работ через *систему депонирования.*

«Депонирование (передача на хранение) – особый метод публикации научных работ (отдельных статей, обзоров, монографий, сборников научных трудов, материалов научных конференций, симпозиумов, съездов, семинаров) узкоспециального профиля, разрешенных в установленном порядке к открытому опубликованию, широкое тиражирование которых, как правило, в силу их узкой специализации, не считается целесообразным, а также работ широкого профиля, срочная информация о которых необходима для утверждения их приоритета. Депонирование предусматривает прием, учет, регистрацию, хранение научных работ и обязательное размещение информации о них в специальных информационных изданиях».

Подготовка и передача на депонирование научных работ происходит в соответствии с «Инструкцией о порядке депонирования научных работ по естественным, техническим, социальным и гуманитарным наукам» (М., 2013).

Депонированные научные работы находятся на хранении в депозитарии ВИНИТИ РАН, копии работ предоставляются заинтересованным организациям и специалистам на бумажном и электронном носителях и являются официальной публикацией.

Информация о депонированных научных работах включается в информационные издания ВИНИТИ РАН, в РЖ ВИНИТИ РАН и БД ВИНИТИ РАН и аннотированный библиографический указатель «Депонированные научные работы».

Подать научную работу на депонирование можно, обратившись в Отдел депонирования ВИНИТИ РАН по адресу:

125190, Москва, ул. Усиевича, 20.
ВИНИТИ РАН, Отдел депонирования научных работ.
Тел.: 8 (499) 155-43-28, Факс: 8 (499) 943-00-60.
e-mail: dep@viniti.ru

С инструкцией о порядке депонирования можно ознакомиться на сайте ВИНИТИ РАН: **http://www.viniti.ru**