

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М. В. ЛОМОНОСОВА
МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

На правах рукописи

Шачнев Дмитрий Алексеевич

**Семантические модели классификации и анализа
данных в больших информационно-аналитических
системах**

05.13.17 – Теоретические основы информатики

ДИССЕРТАЦИЯ

на соискание ученой степени

кандидата физико-математических наук

Научный руководитель

д. ф.-м. н., проф.

Васенин Валерий Александрович

Москва – 2021

Оглавление

Введение	4
Глава 1. Требования к моделям, алгоритмам и программному обеспечению. Используемые обозначения	21
1.1. Требования к использованию данных	21
1.2. Функциональные требования	22
1.3. Требования по производительности работы	23
1.4. Требования по корректности работы	23
1.5. Требования по безопасности	24
1.6. Требования по сопровождаемости кода	25
1.7. Список сокращений и условных обозначений	26
Глава 2. Тематические портреты	27
2.1. Математическая модель тематического портрета	30
2.2. Коэффициент схожести токенов	31
2.3. Коэффициент схожести тематических портретов	37
2.4. Пример вычисления коэффициентов схожести	39
2.5. Оценка количества операций и способы повышения производительности	42
2.6. Тестирование метрики сравнения ключевых слов и рубрик	46
2.7. Программная реализация функции схожести токенов на языке SQL	51
2.8. Выводы	53
Глава 3. Алгоритмы построения тематических портретов	55
3.1. Модель информационно-аналитической системы	55
3.2. Построение портрета по связям в информационно-аналитической системе	58
3.3. Примеры построенных тематических портретов	63

3.4. Выводы	66
Глава 4. Вычисление коэффициента значимости вершин	68
4.1. Модель для задания правил вычисления коэффициента	69
4.2. Алгоритм для вычисления коэффициента в реляционной СУБД . .	77
4.3. Выводы	85
Глава 5. Алгоритм ранжирования при поиске экспертов	87
5.1. Описание шагов алгоритма	87
5.2. Выявление потенциального конфликта интересов	91
5.3. Описание программной реализации алгоритма поиска экспертов .	93
5.4. Выводы	100
Заключение	102
Список литературы	105

Введение

Актуальность темы исследования. В настоящее время одним из важных социально и экономически значимых направлений исследований в мире и в России является разработка математических моделей, алгоритмов и реализующих их программных средств для поиска и систематизации информации о результатах деятельности субъектов (отдельных работников, коллективов) в разных предметных (проблемных) областях. Результаты таких исследований нужны для оценки её эффективности (ранжирования) на основе интеллектуального анализа данных, которые хранятся в больших, перманентно изменяющихся коллекциях (базах), содержащих сведения о деятельности каждого из субъектов. Задачи в подобной постановке на настоящее время актуальны во всех секторах экономической и хозяйственной деятельности. На их решение в Российской Федерации направлен Национальный проект «Цифровая экономика Российской Федерации» [11], положения которого приняты и повсеместно реализуются в стране.

Цифровизация в разных областях деятельности человека призвана способствовать не только более активному привлечению специалистов к работе на стратегически важных, востребованных национальной экономикой направлениях, но и к адекватной оценке их деятельности, стимулирующей повышение её эффективности в национальном масштабе.

Результатом внимания государства к вопросам цифровизации национальной экономики с этих позиций в последние годы стало появление большого числа платформ в разных её областях — от медицины и производства товаров массового потребления до научно-технической и технологической деятельности. Каждая из таких платформ, как правило, представляет собой интерактивную систему больших данных, пользователи которой перманентно пополняют её сведениями о результатах деятельности.

Экономическое положение любого государства в мире в ближайшие годы будет определяться его успехами не только на направлении цифровизации и ав-

томатизации производственных (бизнес) процессов, но и, в первую очередь, на путях его перевода на современные технологии. Модели и средства управления на основе цифровых технологий должны способствовать существенному повышению эффективности деятельности (результативности, производительности) субъектов (отдельных работников, коллективов, предприятий и организаций) во всех секторах национальной экономики. Эффективность самих механизмов управления, в свою очередь, зависит от точности (адекватности, достоверности) и оперативности данных, на основе анализа которых управленческие решения готовятся и принимаются. Важное место в таком анализе, как правило, отводится работникам, имеющим хорошие знания и богатый практический опыт.

С учетом представленных выше соображений фигура специалиста, активно и эффективно работающего в отдельной предметной области, является ключевым элементом (звеном) в системе управления этой областью, определяющим не только её настоящее, но и будущее. Таких специалистов в контексте данной диссертации будем именовать *экспертами*. Их знания, навыки используются не только для получения адекватной оценки текущего состояния сферы деятельности, но и для формирования планов, определяющих её будущее. Сравнительный анализ результатов деятельности специалистов с результатами таких экспертов является главным (определяющим) фактором в системе управления любым сектором экономики, отдельным предприятием или организацией в этом секторе.

С этих позиций решение задачи анализа данных, аккумулируемых в базах данных отмеченных выше платформ, с целью поиска и ранжирования результатов работ, выполняемых специалистами — экспертами, является актуальным и в экономическом, и в социальном плане. Методы и средства такого ранжирования на разных направлениях деятельности со временем могут и должны изменяться. Они зависят от национальных, отраслевых и региональных приоритетов и, как следствие, от целей, которые установлены в отдельных областях экономической и хозяйственной деятельности.

Результаты исследований будут изложены применительно к сфере научно-ис-

следовательской, опытно-конструкторской и технологической деятельности. Такая сфера деятельности является одной из самых сложных с точки зрения бизнес-логики и многопараметричности её описания. Это обстоятельство позволяет с высокой степенью вероятности утверждать, что полученные в этой области решения могут с успехом использоваться в других областях экономической и хозяйственной деятельности.

Большой объём данных в рассматриваемой сфере деятельности связан с тем, что в научных организациях в последние годы стали широко использоваться системы анализа текущих исследований (Current Research Information Systems, *CRIS-системы*), в которых хранятся метаданные о результатах деятельности специалистов: научной, инновационной, преподавательской, публицистической и т.п. Примерами таких результатов деятельности являются научные публикации, выступления на конференциях, разработанные и прочитанные учебные курсы, научное руководство, членство в редакционных коллегиях научно-технических журналов и сборников. Кроме того, объекты различных типов в таких системах связаны между собой. Например, у записи о научной статье может быть ссылка на журнал или сборник, в котором она опубликована, у сборника докладов — ссылка на серию, в которую он входит или на конференцию, материалы которой в нём опубликованы. В дальнейшем под термином «информационно-аналитическая система» будем иметь в виду граф связанных между собой объектов; более чёткое определение будет дано в разделе 3.1.

Важное место в процессе анализа научно-исследовательских, опытно-конструкторских и технологических работ (НИОКТР) имеют индикаторы и показатели их значимости в той или иной системе оценивания. Например, одним из механизмов определения значимости публикации научно-технического содержания является её цитируемость — количество других публикаций, ссылающихся на неё. На цитируемости основаны более сложные показатели. Например, импакт-фактор журнала рассчитывается как $A \div B$, где A — число цитирований в определённый год публикаций, вышедших в данном журнале за предшествующие несколько лет

(например, 2 года или 5 лет), B — общее число публикаций в журнале за эти предшествующие годы. Индекс Хирша для учёного определён как наибольшее число $h \in \mathbb{N}$, такое что h его публикаций были процитированы как минимум по h раз каждая. В большинстве CRIS-систем подобные показатели доступны, и, как следствие, возникает возможность *поиска экспертов в заданной предметной области по их результатам деятельности и с учётом показателей этих результатов*.

Многие индикаторы не являются универсальными для всех дисциплин и предметных областей, или имеют другие недостатки. Например, импакт-факторы журналов основаны на числе цитирований за определённый промежуток времени, но практика цитирования и среднее время появления цитирований сильно отличаются в разных предметных областях [12; 13]. Исследования, результаты которых представлены в диссертации, предполагают соблюдение принципов построения CRIS-систем, описанных в Лейденском манифесте для наукометрии [14] и нацеленных на рациональное применение индикаторов с учётом их недостатков. К числу таких принципов относится необходимость учитывать экспертную оценку наряду с численными индикаторами, подбирать индикаторы в зависимости от предметной области и от исследовательских задач, сохранять сбор данных и аналитические процессы открытыми и прозрачными. Заметим, что создание новых наукометрических показателей выходит за рамки диссертационной работы. В случае, если такие показатели будут разработаны, они могут быть учтены в рамках предложенной далее модели определения значимости.

В случае, если доступных показателей для оценки значимости несколько, возникает вопрос, какие именно показатели целесообразно использовать, и как их необходимо учитывать при вычислении коэффициента значимости результата деятельности. Например, если научный журнал индексируется в базах Web of Science и Scopus, то для него могут быть вычислены обычный и пятилетний импакт-фактор в первой базе и ранг журнала в системе Scimago (SJR, Scimago Journal Rank) по данным второй базы. Кроме того, эти показатели вычисляются для конкретного года, поэтому они будут меняться с течением времени. Поскольку

требования к показателям результатов деятельности могут меняться в зависимости от задачи, в рамках решения которой оценивается их значимость, возникает необходимость разработать модель, которая позволила бы пользователям системы описывать *сложные критерии отбора и оценки результатов* в формализованном виде, и механизмы для учёта этих критериев.

Таким образом, в диссертации представлены модели, алгоритмы и программные механизмы анализа данных в больших информационно-аналитических системах с целью выявления экспертов, активно работающих в заданной предметной области, и их ранжирования с учётом динамически изменяемых критериев.

Для учёта тематики в диссертационном исследовании предлагается понятие *тематических портретов* — наборов данных, адекватно описывающих некоторую предметную область. Такой тематический портрет может быть сопоставлен каждому результату деятельности или объекту другого типа в CRIS-системе (например, научному журналу или конференции). Кроме того, поисковый запрос описывает предметную область, экспертов в которой требуется найти, то есть также является тематическим портретом. Таким образом, задача определения релевантности результата или эксперта поисковому запросу может быть сведена к определению схожести двух портретов. В диссертации представлены метрики, разработанные для решения подобной задачи автором.

Степень разработанности темы исследования. Приведём библиографический обзор имеющихся в мировой литературе подходов к поиску экспертов в предметной области. Более общий обзор методов анализа текстовых данных, которые можно использовать для решения этой и других задач, будет приведён в главе 2.

Задача поиска экспертов актуальна и востребована в научно-технической сфере для экспертизы тем и результатов исследований [15]; в кадровой сфере [16] и в области средств массовой информации [17].

Процесс поиска экспертов с организационной точки зрения подробно описан

в книге «Сетевая экспертиза» под ред. Д. А. Новикова и А. Н. Райкова [18]. Задача формирования реестра потенциальных экспертов описана в работе П. Б. Мельника [19]. Её решения положены в основу Федерального реестра экспертов научно-технической сферы ФГБНУ НИИ РИНКЦЭ. В диссертации будет исследоваться только задача выбора наиболее подходящего эксперта из некоторого заранее заданного множества. Например, это может быть множество всех учёных, данные о результатах деятельности которых внесены в CRIS-систему.

Подходы к поиску экспертов, средства и системы, реализующие эти подходы, исследуются с конца 1980-х годов. К этому времени появились и стали внедряться в практику информационно-аналитические механизмы анализа больших объёмов текстовых данных. В ранних исследованиях на этом направлении, которые, например, представлены в работе М. Марона и др. [20], предполагается, что имеется конечное множество экспертов, для каждого из которых указаны области его интересов, и по этим областям выполняется поиск. Недостатком такой архитектуры является необходимость вручную поддерживать профили экспертов в актуальном состоянии, и отсутствие возможности установить, является ли заданная предметная область основной сферой интересов эксперта или второстепенной. В некоторых работах предполагается определять компетенции экспертов по косвенным данным, таким как их переписка по электронной почте (как, например, в работе К. Кэмпбелла и др. [21]), данные документов внутренней сети организации (как в системе P@NOPTIC Expert, описанной Н. Красвеллом и др. [22]) или по совокупности данных, например информации на домашних страницах работников, почтовой переписке и по информации о совместном участии в проектах (как в работе Р. Д'Амора [23]). В работе Б. Алеман-Мезы и др. [24] предлагается добавлять на персональные страницы экспертов структурированные данные с использованием технологий семантической паутины, которые потом смогут быть использованы для поиска. В работе Д. Имам-Сейда и А. Кобсы [25] проводится обзор некоторых существующих информационных систем, реализующих поиск экспертов, и предлагается модульная архитектура, к которой можно динамически подключать

новые источники данных.

В новых работах, в дополнение к анализу ключевых слов и текстовому поиску по документам, предлагается использовать методы машинного обучения для классификации документов по предметным областям. В работах К. Балоба с соавторами [26; 27] для этой цели используются генеративные модели линейного классификатора, в работе И. Фанга и др. [28] — дискриминативные модели. В работе Дж. Танга и др. [29] описывается система AMiner, извлекающая данные об учёных из разнородных данных в сети Интернет (в первую очередь, из их персональных страниц) и использующая генеративную вероятностную модель для реализации различных механизмов, в том числе поиска экспертов. В работе А. Синхи и др. [30] описывается система Microsoft Academic Graph. Данные в этой системе получены из интернета и из базы поискового сервиса Bing, формирование списка предметных областей и классификация данных выполнялись с использованием методов машинного обучения.

В книге Б. Гантера и Р. Вилле [31] описывается метод анализа формальных понятий. Понятием называется пара (множество объектов, множество признаков), где каждый объект из первого множества обладает каждым признаком из второго, и множества максимальны. Такие понятия, упорядоченные по вложению, образуют решётку понятий. На основе анализа формальных понятий можно решать различные задачи информационного поиска и анализа данных. В работе В. Боевой и др. [32] описаны приложения к поиску экспертов.

В упомянутых выше публикациях предлагаются методы и средства, предназначенные для анализа полуструктурированных данных, которые не содержат представления результатов деятельности работников в виде, удобном для их многокомпонентного интеллектуального целевого анализа. Таким недостатком обладают, например, корпоративные информационные системы. Кроме того, многие из предлагаемых методов не используют в полной мере информацию о связях между объектами системы. Одним из основных отличий подхода, применяемого в диссертационном исследовании, является использование данных CRIS-систем, которые

обеспечивают целевой сбор и хранение информации о результатах деятельности исследователей в структурированном виде. Это обстоятельство позволяет учесть как тематику результатов, так и их значимость.

Цели и задачи диссертационной работы. Целью диссертационной работы является исследование и разработка математических моделей, алгоритмов и программных средств для поиска экспертов в предметной области путём анализа данных в информационно-аналитических системах о результатах деятельности, и ранжирования экспертов с учётом показателей значимости результатов.

Такая деятельность соответствует областям исследования, отмеченным в пунктах 1, 2, 5 и 9 Паспорта специальности 05.13.17 «Теоретические основы информатики»:

1. Исследование, в том числе с помощью средств вычислительной техники, информационных процессов, информационных потребностей коллективных и индивидуальных пользователей.
2. Исследование информационных структур, разработка и анализ моделей информационных процессов и структур.
5. Разработка и исследование моделей и алгоритмов анализа данных, обнаружения закономерностей в данных и их извлечениях, разработка и исследование методов и алгоритмов анализа текста, устной речи и изображений.
9. Разработка новых интернет-технологий, включая средства поиска, анализа и фильтрации информации, средства приобретения знаний и создания онтологии, средства интеллектуализации бизнес-процессов.

Для достижения поставленной цели решаются следующие задачи:

1. создание модели тематического портрета, который может быть использован для описания предметной области, результата деятельности или эксперта;

2. создание метрики схожести тематических портретов и отдельных составляющих их элементов;
3. разработка алгоритмов для составления тематических портретов объектов в информационно-аналитических системах по связям этих объектов с другими;
4. создание модели, позволяющей динамически определять критерии для отбора результатов деятельности и назначения им коэффициентов значимости в соответствии с их показателями, а также алгоритма для формирования запросов к реляционным СУБД на основе этой модели;
5. разработка алгоритма поиска экспертов по данным об их результатах деятельности, с учётом релевантности результатов запросу и их значимости.

Научная новизна. Важным элементом новизны подхода, представленного в диссертационной работе, является использование для поиска экспертов отдельных результатов их деятельности, каждый из которых имеет метаданные различных типов, на основе которых можно сделать вывод о тематической принадлежности и степени значимости отдельного результата. Таким образом можно получить не просто перечень тематических направлений для каждого потенциального эксперта, но и определить количественный вес каждого направления, что даст более полное представление об области интересов эксперта и о том, как она менялась с течением времени. Этот подход отражён в алгоритме составления тематического портрета объекта по его связям с другими объектами в информационно-аналитической системе, описанном в главе 3.

Другой ключевой особенностью модели, представленной в диссертации, является возможность отбора результатов деятельности, используемых для построения тематических портретов экспертов, по заданным правилам и назначения этим результатам коэффициентов значимости, учитывающих их наукометрические показатели. При этом такие правила не являются фиксированными и могут

динамически меняться пользователем информационно-аналитической системы в зависимости от решаемых им задач. Например, такие правила могут определять, что при поиске экспертов должны учитываться только публикации, индексируемые в международных базах «Сеть науки» (Web of Science) и Scopus, и присваивать им коэффициент на основе числа их цитирований или импакт-фактора научного издания в этих системах. Примером правил, определяющих весовой коэффициент публикации в зависимости от её наукометрических показателей, является методика расчёта комплексного балла публикационной результативности, утверждённая Министерством науки и высшего образования РФ. Для некоторых задач могут потребоваться совсем другие правила, например, учёт не публикаций, а участия в научно-исследовательских работах и результатов интеллектуальной деятельности в других направлениях. Для участия в научно-исследовательских и опытно-конструкторских работах примером численного показателя, который может быть учтён при поиске экспертов, является объём финансирования, умноженный на коэффициент трудового участия исполнителя.

Предполагается, что поисковым запросом является предметная область, описание которой включает в себя ключевые слова, рубрики классификатора, результаты деятельности или комбинацию этих данных. Для случаев, когда в запросе фигурирует результат деятельности (например, заявка на проведение исследования), в модели предусмотрены механизмы, позволяющие исключить наличие у эксперта конфликта интересов с авторами данного результата (заявки). Это достигается за счёт анализа графа соавторств результатов и других данных, если они доступны, например, данных о наличии общих мест работы.

Теоретическая и практическая значимость. Изложенные в диссертации модели, методы и алгоритмы могут быть востребованы в информационно-аналитических системах в различных сферах деятельности. В научно-технической сфере результаты диссертации могут быть использованы для решения задач, направленных на

- поиск специалистов для экспертизы заявок на проведение научно-исследовательских, опытно-конструкторских и технологических работ и их результатов;
- определение тематики деятельности отдельных исследователей или коллективов исполнителей, структурных подразделений и организаций в целом;
- выявление научных коллективов и научных связей между различными подразделениями и организациями при проведении междисциплинарных исследований;
- определение наиболее активно развивающихся научных областей;
- формирование рейтинговых показателей отдельных учёных, структурных подразделений или организаций целиком, в том числе, с привязкой к заданной предметной области;
- формирование отчётных и статистических материалов.

Заметим, что для построения информационно-аналитической системы, в которой можно использовать разработанные алгоритмы, не обязательно собирать информацию о результатах деятельности специалистов путём их ручного ввода. Такие данные во многих случаях могут быть импортированы из внешних источников. Например, метаданные публикаций, которым присвоен цифровой идентификатор объекта (DOI), можно получить автоматизированным способом из системы CrossRef — официального регистрационного агентства.

Теоретическая значимость диссертации заключается в разработке метрики схожести тематических портретов с учётом взаимных семантических связей между составляющими их элементами, алгоритма для построения тематического портрета вершины в графе информационно-аналитической системы по цепочкам связей с другими вершинами, а также модели для определения правил отбора результатов и вычисления коэффициентов значимости.

Методология исследования включает следующие характеризующие её аспекты.

- Для построения модели тематического портрета используются ключевые слова и рубрики, которые анализируются с использованием аппарата и методов теории графов. Кроме естественного графа классификатора, в который организованы рубрики, используется вспомогательный граф совместной встречаемости ключевых слов и рубрик.
- Алгоритм, реализующий формирование тематического портрета, присваивает каждому ключевому слову и рубрике вещественный весовой коэффициент, определяющий степень соответствия этого слова или рубрики описываемому объекту.
- Для расчёта коэффициентов схожести между портретами и отдельными элементами, составляющими их, используются методы алгебры и математической статистики. Проведено сравнение разработанных функций схожести с другими известными метриками.
- Для формализации понятия информационно-аналитической системы используется графовая модель данных и элементы онтологии, а также реляционная модель данных.
- При разработке программной реализации моделей и алгоритмов использовался язык программирования Python и инструментальное средство Django. Для контроля качества исходного кода он проверялся различными средствами статического анализа, а также были написаны автоматические тесты, обеспечивающие полное покрытие кодовой базы. Для тестирования использовалась система управления базами данных PostgreSQL.

Положения, выносимые на защиту. На защиту выносятся: обоснование актуальности, научная новизна, теоретическая и практическая значимость работы,

а также следующие положения, которые подтверждаются результатами исследования, представленными далее в разделе Заключение.

1. Модель тематических портретов для описания заданной предметной области, алгоритм построения тематических портретов результатов деятельности и их авторов на основе совокупности информации о ключевых словах и элементах рубрикаторов, связанных с ними в графе информационно-аналитической системы.
2. Метрики для определения степени схожести тематических портретов и составляющих их элементов, на основе анализа данных об их совместном использовании.
3. Модель и алгоритм, позволяющие динамически определять критерии для отбора результатов деятельности и назначать таким результатам коэффициенты значимости в соответствии с их наукометрическими показателями.
4. Программная реализация системы интеллектуального анализа данных для поиска и ранжирования экспертов.

Степень достоверности и апробация результатов. Представленные в диссертации материалы докладывались на следующих конференциях.

- Всероссийская конференция «Научный сервис в сети Интернет», Новороссийск, 2016, доклад «Использование онтологического представления структуры реляционной базы для агрегации наукометрических данных».
- Международная Ершовская конференция по информатике, Москва, 2017, доклад «Using the Subject Area Ontology for Automating Learning Processes and Scientific Investigation».
- Всероссийская конференция с международным участием «Знания – Онтологии – Теории», Новосибирск, 2017, доклад «Онтология предметной области

для научных исследований и автоматизации учебных процессов: методы реализации и алгоритмы использования».

- Международная конференция «Актуальные проблемы системной и программной инженерии», Москва, 2017, доклад «Web-редактор онтологий: архитектура и способы применения».
- Конференция «Ломоносовские чтения», Москва, 2018, доклад «Программные механизмы автоматической генерации SQL-запросов в ИАС «ИСТИНА»: особенности и варианты использования».
- Конференция «Ломоносовские чтения», Москва, 2019, доклад «Методы и средства тематического анализа данных в больших системах на основе ключевых слов и косвенных связей между ними».
- Конференция «Ломоносовские чтения», Москва, 2020, доклад «Новая версия rmodel — генератора SQL-запросов, основанного на онтологическом представлении структуры БД информационной системы».

Кроме того, материалы были доложены на научных семинарах: «Проблемы современных информационно-вычислительных систем» под руководством д. ф.-м. н., проф. В. А. Васенина (1–3) и «Теоретические проблемы программирования» под руководством д. ф.-м. н., проф. В. А. Захарова (4).

1. 13 декабря 2016 г., доклад «Механизмы агрегации наукометрических данных в ИАС ИСТИНА».
2. 17 апреля 2018 г., доклад «Тематический классификатор наукометрических данных на примере анализа проектов в ИАС ИСТИНА».
3. 13 ноября 2018 г., доклад «Методы определения тематической схожести документов, на примере поиска схожих диссертационных советов МГУ».

4. 19 мая 2021 г., доклад «Метод поиска объектов в информационных системах с учётом динамически задаваемых критериев отбора и ранжирования».

Апробация результатов была проведена в информационно-аналитической системе «ИСТИНА», в разработке которой автор принимает участие. Алгоритмы, изложенные в диссертации, легли в основу отдельного модуля поиска экспертов в этой ИАС, а также нашли применение в подсистемах расчёта персональных рейтингов, проведения конкурсов, формирования отчётных материалов. По результатам апробации модели и реализующих её программных средств получен акт о внедрении.

По состоянию на сентябрь 2021 года в системе «ИСТИНА» зарегистрировано 140 тысяч пользователей, и система используется в 29 организациях: в МГУ имени М. В. Ломоносова, ряде других вузов и институтов Российской академии наук, научно-технических центрах Министерства здравоохранения РФ. В системе организовано хранение мета-информации о результатах деятельности различных типов, в числе которых:

- статьи в журналах и сборниках, тезисы докладов (около 990 тысяч записей),
- доклады на конференциях (около 300 тысяч),
- книги (монографии, учебные пособия и другие — около 75 тысяч),
- кандидатские и докторские диссертации (около 30 тысяч),
- патенты и свидетельства о регистрации прав на ПО,
- участие в научных проектах (НИР, НИОКР, гранты),
- членство в редколлегиях журналов и сборников, в оргкомитетах и программных комитетах конференций,
- научное руководство студентами и аспирантами,
- читаемые учебные курсы и авторства курсов (около 100 тысяч записей),
- членство в диссертационных советах.

Информация, которая хранится в системе, вносится самими авторами результатов деятельности. Такой подход позволяет обеспечить целостность профи-

лей пользователей и разделить профили учёных с одинаковыми именами друг от друга. Такие качества сложно обеспечить при получении данных другим способом, например, от издательств или из международных библиографических систем. При этом организована дополнительная проверка данных ответственными за сопровождение информации от организаций. Неподтверждённые результаты деятельности показываются в профилях пользователей, но могут не учитываться в отчётных материалах или при поиске экспертов.

Методологические принципы, на которых основана ИАС «ИСТИНА», более подробно описаны в монографии под редакцией академика В. А. Садовниченко [33], а также в статье В. А. Садовниченко и В. А. Васенина [34].

Публикации. Материалы диссертации опубликованы в 10 печатных работах, из них 2 статьи в журнале, индексируемом в системе RSCI и входящем в перечень ВАК [1; 2], 1 статья в российском журнале, переводная версия которого индексируется в системах WoS и Scopus [3], 4 статьи в сборниках трудов конференций [4–7] и 3 тезиса докладов [8–10].

Личный вклад автора. Все результаты, представленные в диссертации, получены лично автором. Предшествующие научные работы, использованные в диссертации, отмечены библиографическими ссылками.

Структура и объем диссертации. Диссертация состоит из введения, 5 глав и заключения. Полный объем диссертации составляет 112 страниц, включая 7 рисунков и 7 таблиц. Список литературы содержит 54 наименования.

В главе 1 приведены используемые термины и требования к моделям, алгоритмам и их программной реализации. Следующие две главы посвящены определению релевантности результатов деятельности поисковому запросу. В главе 2 формализовано понятие тематического портрета и приведён алгоритм вычисления коэффициента схожести двух портретов. Далее, в главе 3 описан способ построения тематического портрета вершины в информационно-аналитической системе по данным о ключевых словах и рубриках, приписанных вершинам, связанным с данной цепочками рёбер. Глава 4 посвящена определению значимости резуль-

татов и механизмам для задания правил вычисления коэффициента значимости. Наконец, в главе 5 результаты предыдущих глав соединяются в алгоритм ранжирования при поиске экспертов, а также приводится описание и исследование производительности программной реализации.

Глава 1

Требования к моделям, алгоритмам и программному обеспечению. Используемые обозначения

Прежде чем перейти к описанию предлагаемых в диссертации моделей, алгоритмов и реализующих их программных средств, перечислим требования к ним, которые позволяют решать поставленные задачи. Такие требования содержат характеристики и показатели, предъявляемые в соответствии со стандартом ГОСТ Р ИСО/МЭК 9126-93 к разрабатываемым программным реализациям, которые подлежат тестированию на реальных данных. Они определяют исходные положения, которыми руководствовался автор в ходе диссертационного исследования.

1.1. Требования к использованию данных

При разработке моделей и алгоритмов должны учитываться следующие положения.

- Информационно-аналитические системы, данные в которых вносятся самими пользователями, могут иметь недостаточно информации для применения распространённых средств интеллектуального анализа, таких как методы машинного обучения. В частности, при работе с результатами научной деятельности следует исходить из недоступности полных текстов результатов.
- Доступная для анализа информация включает в себя информацию о типе результатов деятельности, мета-данные результатов, в том числе ключевые слова и рубрики различных классификаторов, информацию о связях результата с другими объектами.
- Ключевые слова могут быть записаны на различных естественных языках и с использованием различных грамматических форм, например, в един-

ственном и множественном числе. Соответствие между ключевыми словами и понятиями не является взаимно-однозначным: одно понятие может быть выражено различными ключевыми словами, и одно слово может обозначать различные понятия. В связи с этим обстоятельством, ключевые слова нельзя считать полностью независимыми друг от друга.

- Любые численные показатели, полученные в результате работы алгоритмов, должны быть наглядны для пользователей. А именно, должно быть показано, каким образом показатели получены из исходных данных в СУБД.

1.2. Функциональные требования

Программная реализация должна предоставлять следующие функциональные возможности.

- Интерфейс для поиска отдельных результатов деятельности по ключевым словам, рубрикам и заданному набору правил.
- Интерфейс для поиска экспертов в предметной области, заданной ключевыми словами и рубриками, по заданному набору правил. Выдача должна быть представлена в виде списка, в котором для каждого потенциального эксперта указан список результатов деятельности, по которым он найден, со ссылками на страницу каждого результата, а также должны быть указаны ранги для экспертов и отдельных результатов.
- Интерфейс для ввода наборов правил, позволяющий для каждого правила выбрать тип и задать вычисляемое выражение, а также поддерживающий редактирование, перемещение и удаление правил в наборе.
- Интерфейс для проверки соответствия результата деятельности набору правил. Должна быть реализована возможность выбрать результат из списка результатов заданного типа, автором которых является пользователь, или

проверить результат по его URL. Для каждого правила, соответствующего типу результата, должен быть указан список несоответствий, либо сообщение об их отсутствии.

- Интерфейс для применения набора правил к результатам заданной группы пользователей, с возможностью просмотра результатов в виде сводной таблицы либо в виде гистограммы с распределением ранга.

1.3. Требования по производительности работы

Программная реализация должна удовлетворять следующим положениям, которые обеспечивают возможность её практического использования.

- Вычислительная сложность алгоритмов должна быть достаточно мала для их практического использования при обработке входящих запросов в режиме реального времени. Это означает, что в информационно-аналитических системах среднего размера (работающих с миллионами и десятками миллионов объектов) максимальное время обработки запроса сервером не должно превышать 5 минут.
- Лишь для небольшого числа ресурсоёмких задач может быть предусмотрена возможность работы в асинхронном (неблокирующем) режиме: постановка задачи в очередь, обработка задач в очереди по порядку, уведомление автора запроса об успешном выполнении задачи.
- Число выполняемых запросов к базе данных не должно зависеть от количества входных данных.

1.4. Требования по корректности работы

Реализация должна иметь автоматические тесты, запускаемые при каждом изменении программного кода. Не менее 70 % строк кода должны быть задейство-

ваны в тестах. В том числе должны присутствовать тесты, перечисленные далее.

- Тесты, проверяющие корректность генерации и работоспособность выражений на языке SQL, соответствующих каждому предикату, доступному для каждого типа объектов.
- Юнит-тесты, проверяющие работоспособность отдельных функций в программном коде.
- Функциональные тесты, проверяющие возможность сервера обрабатывать поступающие к нему запросы.

1.5. Требования по безопасности

Реализация должна удовлетворять следующим требованиям по безопасности.

- SQL-запросы, вносящие изменения в базу данных, должны выполняться только в рамках обработки HTTP-запросов, тип которых это предусматривает. Например, допустимы типы запросов POST, PUT и DELETE, но не GET или HEAD. Исключение возможно для запросов, вносящих изменения в таблицы, содержащие кэшированные данные.
- При обработке запросов POST, PUT и DELETE должен проверяться секретный ключ для защиты от межсайтовых подделок запросов (CSRF-токен), который клиент мог получить в ответе на предыдущий запрос.
- Страницы, при посещении которых могут быть внесены изменения в базу данных, должны быть доступны только зарегистрированным пользователям системы.
- В реализации должна использоваться модель логического разграничения доступа (МЛРД).

- Автоматически сформированные SQL-запросы должны выполняться на отдельной реплике базы данных, доступ к которой предоставляется только на чтение.
- Все литеральные значения, в том числе заданные пользователями системы, должны передаваться в автоматически формируемые SQL-запросы как связанные параметры, а не подставляться непосредственно в текст запроса.
- Для всех используемых в коде системы внешних программных модулей должен проводиться мониторинг наличия в них известных уязвимостей. В случае их обнаружения зависимости должны обновляться до тех версий, в которых эти уязвимости устранены.

1.6. Требования по сопровождаемости кода

Исходный код программной реализации должен удовлетворять следующим требованиям.

- Код должен быть разбит на модули, каждый из которых решает отдельную задачу. Пример: модуль генерации SQL-запросов, модуль выполнения запросов и обработки результатов; модуль контроля доступа; модуль построения агрегированных данных.
- Исходный код должен проверяться средствами статического анализа при каждом изменении. Примеры средств для языка Python: pylint, pyflakes.
- Исходный код должен соответствовать рекомендациям сообщества Python по стилю оформления (PEP 8).
- Все программные модули, классы и их методы, функции, должны иметь строки документации, описывающие их предназначение, список аргументов и их типы, тип выходного значения.

1.7. Список сокращений и условных обозначений

λ	— Ранг результата деятельности или эксперта
obj	— Объект
$pred$	— Предикат
$subj$	— Субъект
$\omega(t, t')$	— Вес связи между соседними токенами
Ω	— Вес ребра и вес цепочки при построении тематического портрета
ρ	— Коэффициент пути доступа при выявлении конфликта интересов
σ	— Коэффициент набора при построении тематического портрета
E	— Множество рёбер (триплетов) информационно-аналитической системы
$h(r)$	— Высота рубрики
$I(r)$	— Значимость рубрики
L	— Множество литеральных значений
N	— Множество основных вершин информационно-аналитической системы
$P(t)$	— Множество портретов из P_{train} , содержащих токен t
$P(t, t')$	— Множество портретов из P_{train} , содержащих токен t и токен t'
p	— Тематический портрет
P_{train}	— Обучающая выборка — множество тематических портретов
q	— Поисковый запрос (тематический портрет)
$R(r)$	— Множество дочерних рубрик
R	— Множество рубрик (рубрикатор, классификатор)
r	— Рубрика
$s(\cdot, \cdot)$	— Функция схожести токенов или тематических портретов
T	— Множество всех токенов ($W \cup R$)
t	— Токен (ключевое слово или рубрика)
T_p	— Множество токенов портрета p
W	— Множество возможных ключевых слов
w	— Ключевое слово

Глава 2

Тематические портреты

Под тематикой объекта, в контексте настоящей диссертации, будем понимать проблемную (предметную) область такого объекта. Проблемную область можно связать с объектами различных типов. Например, в CRIS-системах это могут быть специалисты, результаты их деятельности, научные журналы, конференции, организации и другие объекты. Кроме того, в задаче поиска экспертов в упомянутом ранее понимании поисковый запрос также задаёт некоторую предметную область.

Существует множество моделей для формализованного описания предметных областей. Для целей настоящей диссертации была выбрана модель, основанная на двух понятиях — ключевых словах и рубриках.

Определение 1. *Ключевым словом* называется одно или несколько слов на естественном языке, описывающие тематику объекта [35].

Рубрикой называется вершина древовидного классификатора (рубрикатора). Примерами таких классификаторов являются УДК (универсальная десятичная классификация), ГРНТИ (Государственный рубрикатор научно-технической информации) и классификатор Российского фонда фундаментальных исследований (РФФИ), используемый для проведения конкурсов этого фонда. Некоторые классификаторы не имеют единой корневой вершины; в таких случаях добавляется «виртуальная» вершина, потомками которой являются рубрики верхнего уровня классификатора.

Для формализованного описания предметной области будем использовать множества ключевых слов и рубрик с присвоенными им весовыми коэффициентами. Коэффициенты определяют, насколько значимым является данное слово или рубрика в описании области. Такие множества будем называть *тематическими портретами* или просто *портретами* (формальное определение будет дано ниже в разделе 2.1). Указание различных весовых коэффициентов в портрете позво-

ляет отделить наиболее близкие слова и рубрики от тех, которые не определяют его тематическую направленность, а лишь дополняют её. Таким образом, предлагаемая модель является обобщением известной концепции «мешка слов» (bag of words) [36]. Множества ключевых слов и рубрик с весовыми коэффициентами естественным образом возникают при решении различных задач. Например, в программном интерфейсе (API), предоставляемом издательством Springer Nature, для поискового запроса выдаётся список наиболее частотных ключевых слов и рубрик из публикаций, соответствующих запросу, с указанием их частотности.

Обоснуем, почему выбрана модель тематического портрета, основанная на ключевых словах и рубриках, а не другие распространённые модели. При выборе учитывались требования к использованию данных, сформулированные выше в разделе 1.1. Среди распространённых моделей, в первую очередь, следует выделить модель, представляющую тематику в виде вектора в пространстве $\mathbb{R}^n$ [37]. В частности, такая модель используется в алгоритмах машинного обучения, основанных на нейронных сетях, например, в алгоритме word2vec [38]. Основным преимуществом этой модели является простота вычисления коэффициентов схожести различных объектов — для этого достаточно взять скалярное произведение векторов. Однако, результаты операций над векторами сложно представить в наглядной форме. При этом задача поиска экспертов может быть использована в бизнес-процессах, где требуется наглядное объяснение степени соответствия того или иного портрета заданной предметной области (требование 4).

Кроме того, при использовании векторной модели в качестве источников данных обычно используют полные тексты документов. Однако, в CRIS-системах обычно хранятся только метаданные результатов деятельности. Полные тексты могут быть доступны в виде прикреплённых файлов, однако при отсутствии единых требований к форматам таких файлов извлечение из них текстов является отдельной трудоёмкой задачей (требование 1). Другой особенностью векторной модели является то обстоятельство, что при составлении векторов по текстам в качестве базиса пространства обычно берутся все возможные слова. Это подра-

зумевают независимость слов друг от друга и не позволяет учесть связи между словами, такие как синонимия (требование 3). Для устранения этого недостатка, а также для понижения размерности пространства, были разработаны алгоритмы, такие как латентно-семантический анализ (LSI) [39], однако они имеют довольно большую вычислительную сложность и делают представление ещё менее наглядным.

Портреты, основанные на ключевых словах и рубриках, напротив, понятны человеку, даже если они созданы алгоритмически, а не взяты из данных информационно-аналитической системы, то есть такая модель соответствует набирающей влияние концепции объяснимого (доказуемого) искусственного интеллекта [40].

Другое направление тематического анализа связано с построением онтологий предметных областей, то есть выделением классов, объектов в каждом классе и связей между объектами. Однако на данный момент методы автоматизированного построения онтологий на основе текстов далеки от совершенства и позволяют выявлять лишь небольшое число связей. С учётом этого обстоятельства, для построения полноценной онтологии необходим большой объём ручной работы [41].

Таким образом, подход на основе ключевых слов и рубрик не требует ручной работы, в отличие от онтологий, и является наглядным, в отличие от векторного представления. Дополнительно можно отметить, что ключевые слова и рубрики являются естественным атрибутом научных публикаций и научно-исследовательских работ, во многих случаях их указание является обязательным (требование 2). А именно эти типы результатов деятельности будут являться основным источником данных для работы алгоритма поиска экспертов в CRIS-системах.

При этом для создания тематических портретов по текстам на естественном языке могут быть использованы методы машинного обучения [42]. Некоторые методы позволяют также определять, какие ключевые слова являются характерными для разных предметных областей на основе массивов текстов (например, метод “Topical N-Grams” [43]), что позволяет строить тематические портреты таких областей.

2.1. Математическая модель тематического портрета

Составим формализованное определение тематического портрета, основанного на ключевых словах и рубриках, с учётом изложенных выше соображений.

Пусть W — множество всех возможных ключевых слов, записанных на любом естественном языке и относящихся к любой предметной области (это множество может быть бесконечным). Пусть также задан некоторый древовидный классификатор, то есть множество рубрик R , заданная корневая рубрика $r_0 \in R$ и функция, определённая для остальных рубрик и сопоставляющая им их родительские рубрики: $\text{par}: R \setminus \{r_0\} \rightarrow R$. Такая функция при этом не должна образовывать циклов, то есть $\nexists r: (\text{par} \circ \dots \circ \text{par})(r) = r$.

Определение 2. *Тематическим портретом* будем называть набор ключевых слов и рубрик, каждой из которых присвоен некоторый вещественный коэффициент от 0 до 1:

$$p = \{(w_1, c_{w_1}), \dots, (w_n, c_{w_n}), (r_1, c_{r_1}), \dots, (r_m, c_{r_m})\}, \quad (2.1)$$

где $n + m > 0$, $w_i \in W$, $r_j \in R$, $0 < c_{w_i}, c_{r_j} \leq 1$.

Как было указано выше, тематические портреты могут быть сопоставлены объектам различных типов, данные о которых хранятся в CRIS-системах. При этом коэффициенты c_{w_i}, c_{r_j} отражают степень тематического соответствия объекта ключевому слову или рубрике, то есть показывают, насколько данное слово или рубрика подходит для описания тематики объекта.

В дальнейшем будет удобно использовать общее название и для ключевых слов, и для рубрик, там где эти понятия будут равноправны. Назовём их *токенами*¹. Через T обозначим множество всех токенов $W \cup R$.

Определение 3. Для портрета p из (2.1) множество $\{w_1, \dots, w_n, r_1, \dots, r_m\}$ будем называть *множеством токенов портрета p* и обозначать T_p .

¹ Обозначение не связано с термином «токен» из лексического анализа.

2.2. Коэффициент схожести токенов

Под сравнением двух тематических портретов будем подразумевать вычисление коэффициента схожести этих портретов — то есть числа от 0 до 1, отражающего степень тематической близости двух портретов друг к другу.

Перед тем, как приступить к алгоритму определения такого коэффициента, необходимо научиться решать задачи более простого порядка, а именно — вычислять коэффициент схожести: двух ключевых слов; двух рубрик классификатора; ключевого слова и рубрики. Не имея данных о том, в каких контекстах встречаются данные слова и рубрики, эти задачи можно решить лишь частично. Например, при помощи морфологического анализа можно выявлять однокоренные слова или формы одного слова, но будет невозможно определить наличие синонимической связи между словами, не являющимися родственными. Данные о контекстах можно получить при наличии заранее заданного множества тематических портретов. Такое множество будем называть обучающей выборкой и обозначать P_{train} .

2.2.1. Коэффициент схожести двух ключевых слов

Для вычисления коэффициента схожести двух ключевых слов за основу примем модель, описанную в [7; 44], внеся в неё изменения, учитывающие весовые коэффициенты в портретах (в формуле (2.2)) и позволяющие добиться коэффициента схожести 1 ключевого слова с самим собой. Идея этой модели основана на следующих двух соображениях.

1. (Дистрибутивная гипотеза [45].) Ключевые слова, наиболее близкие по смыслу (например, синонимы) будут редко встречаться в одном портрете. Например, в случае научных публикаций, если какое-то понятие может быть выражено различными ключевыми словами, то как правило авторы публикации указывают только одно из этих слов. Однако, такие слова будут встречаться в схожих контекстах, то есть совместно с рядом других ключевых слов или рубрик. Таким образом, если составить граф, в котором вершины будут

соответствовать токенам, и две вершины будут соединены ребром, если соответствующие им токены встречаются совместно в каком-то портрете из обучающей выборки, то коэффициент схожести должен быть больше у тех пар вершин, которые имеют больше общих соседей в этом графе.

2. Тематические портреты, в которых много токенов, описывают более широкую предметную область. Следовательно, чем больше токенов в портрете, тем меньший вклад должен вносить этот портрет в коэффициенты схожести своих элементов с другими токенами.

На основе этих соображений будем использовать следующий алгоритм расчёта тематической близости двух ключевых слов. Через $P(t)$ будем обозначать множество портретов из обучающей выборки, содержащих токен t :

$$P(t) = \{p \in P_{\text{train}} : t \in T_p\}.$$

Через $P(t_k, t_l)$ будем обозначать множество портретов, содержащих два токена t_k и t_l одновременно: $P(t_k, t_l) = P(t_k) \cap P(t_l)$.

Определим вес связи между t_k и t_l следующей формулой, учитывающей второе соображение:

$$\omega(t_k, t_l) = \sum_{p \in P(t_k, t_l)} \frac{c_{t_l}}{\sum_{t \in T_p} c_t}. \quad (2.2)$$

Функция ω не является симметричной, так как слагаемые отражают долю t_l , а не t_k в каждом портрете p . Однако, это необходимо для достижения коэффициента схожести 1 ключевого слова с самим собой.

Пусть теперь w_i и w_j — два ключевых слова, $T_{i,j} = \{t \in T : |P(w_i, t)| \geq 1, |P(w_j, t)| \geq 1\}$ — множество токенов, встречающихся совместно с обоими словами в обучающей выборке.

За первоначальную версию коэффициента схожести между w_i и w_j примем следующее значение:

$$s_0(w_i, w_j) = \sum_{t \in T_{i,j}} (\omega(w_i, t) + \omega(w_j, t)). \quad (2.3)$$

В отличие от ω , эта функция является симметричной, так как оба ключевых слова w_i и w_j подставляются в ω как левый аргумент.

Недостатком данной формулы является то обстоятельство, что она не учитывает частоту встречаемости ключевого слова во всей обучающей выборке. Из-за этого коэффициент схожести между двумя словами может оказаться высоким только из-за того, что эти слова достаточно общие и часто встречаются в выборке. Для исправления этого недостатка будем учитывать обратную частоту встречаемости двух слов, что является разновидностью известной меры “inverse document frequency” [46]:

$$s(w_i, w_j) = \frac{s_0(w_i, w_j)}{|P(w_i)| + |P(w_j)|}. \quad (2.4)$$

Построенная функция для коэффициента схожести принимает значения от 0 до 1, что следует из следующей теоремы, сформулированной и доказанной автором.

Теорема 1. *Значение коэффициента схожести $s(w_i, w_j)$ равно 1 тогда и только тогда, когда $\bigcup_{p \in P(w_i)} T_p = \bigcup_{p \in P(w_j)} T_p$, то есть все токены, являющиеся соседними для одного слова, являются также соседними и для другого. В частности, коэффициент схожести слова с самим собой равен 1. В остальных случаях $s(w_i, w_j) < 1$.*

Доказательство. Рассмотрим сначала случай, когда множества соседних токенов совпадают. Тогда множество $T_{i,j}$ также совпадает с ними, и каждый токен каждого портрета, содержащего w_i или w_j , входит в $T_{i,j}$. Некоторые токены могут входить в несколько таких портретов одновременно, тогда их вклад в $\omega(w_i, t)$ или $\omega(w_j, t)$ будет состоять из нескольких компонент, соответствующих каждому портрету. Поэтому можно говорить о вкладе пары (портрет p , токен $t \in T_p$).

Разобьём $s_0(w_i, w_j)$ на две суммы: $\sum_{t \in T_{i,j}} \omega(w_i, t)$ и $\sum_{t \in T_{i,j}} \omega(w_j, t)$. Заметим, что для каждого портрета $p \in P(w_i)$ сумма вкладов пар (p, t) , соответствующих

этому портрету, в первую сумму равна

$$\sum_{t \in T_p} \frac{c_t}{\sum_{t' \in T_p} c_{t'}} = 1.$$

Поэтому сумма вкладов всех пар в первую сумму равна $\sum_{p \in P(w_i)} 1 = |P(w_i)|$. Аналогично сумма вкладов всех пар во вторую сумму равна $|P(w_j)|$. Следовательно, $s_0(w_i, w_j) = |P(w_i)| + |P(w_j)|$. Но это в точности знаменатель в формуле (2.4), поэтому $s(w_i, w_j) = 1$.

Если множества соседних токенов не совпадают, значит существует портрет, вклад соответствующих которому пар в одну из сумм в числителе меньше 1. Тогда либо первая сумма будет меньше левой части знаменателя, либо вторая сумма — меньше правой части, либо и то, и другое. Поэтому получим $s(w_i, w_j) < 1$. $\square$

Определение 4. Два ключевых слова w_1 и w_2 будем называть *эквивалентными*, если $s(w_1, w_2) = 1$.

Симметричность такого отношения следует из симметричности функции $s(w_1, w_2)$. Рефлексивность и транзитивность следуют из Теоремы 1. Следовательно, такое отношение действительно является отношением эквивалентности.

2.2.2. Коэффициент схожести двух произвольных токенов

Определение 5. Множеством дочерних рубрик рубрики r назовём множество $R(r) = \{r' : \text{par}(r') = r\}$.

Для вычисления коэффициента схожести между ключевым словом и рубрикой, а также между двумя рубриками будем пользоваться формулой, основанной на (2.4), но с изменениями, учитывающими древовидную структуру рубрикатора. Эти изменения будут основаны на следующих соображениях.

- Если на основе анализа информации о совместной встречаемости известно о схожести токена t и рубрики r , то можно сделать вывод и о схожести t с родительской рубрикой $\text{par}(r)$, а также со всеми дочерними рубриками $r' \in R(r)$. Разумеется, коэффициент схожести должен быть при этом уменьшен.

- Связи между родительскими и дочерними рубриками, расположенные выше в иерархии, то есть ближе к корню дерева, должны учитываться с меньшим весом, так как чем выше рубрики, тем более широкие предметные области они описывают.

Определение 6. Предком n -го уровня рубрики r будем называть рубрику

$$\text{par}^n(r) = \begin{cases} r & \text{если } n = 0, \\ \text{par}(\text{par}^{n-1}(r)) & \text{если } n \in \mathbb{N}. \end{cases}$$

Замечание: предок n -го уровня определён не для всех рубрик.

Определение 7. Высотой рубрики r будем называть высоту поддерева, соответствующего этой рубрике:

$$h(r) = \max\{n \in \mathbb{Z}_{\geq 0} : \exists r' : r = \text{par}^n(r')\}.$$

Введём понятие *значимости рубрики* — величины, которая будет равномерно убывать от 1 для рубрик-листьев до 0 для корня дерева:

$$I(r) = \frac{h(r_0) - h(r)}{h(r_0)}.$$

Теперь можно определить новую функцию схожести, учитывающую появившиеся дополнительные соображения для рубрик. Через $\hat{s}$ будем обозначать «старую» функцию схожести из (2.4), применённую к произвольным токенам как к ключевым словам, без учёта иерархической структуры дерева рубрик.

Для двух ключевых слов s будет совпадать с $\hat{s}$:

$$s(w_k, w_l) = \hat{s}(w_k, w_l).$$

Для ключевого слова и рубрики определим её следующим образом:

$$s(w_k, r_l) = \max \left\{ \hat{s}(w_k, r_l), \frac{I(\text{par}(r_l))}{|R(\text{par}(r_l))|} \hat{s}(w_k, \text{par}(r_l)), \frac{I(r_l)}{|R(r_l)|} \sum_{r' \in R(r_l)} \hat{s}(w_k, r') \right\}. \quad (2.5)$$

В этой формуле для каждой связи между родительской рубрикой и дочерней вводится понижающий коэффициент, обратно пропорциональный общему количеству дочерних рубрик. Это позволяет избежать чрезмерного увеличения веса связи для рубрик, имеющих много дочерних. В частности, при таком построении, суммарный вклад всех дочерних рубрик не может оказаться больше 1. Отметим, что благодаря умножению на значимость рубрики, связи, расположенные ближе к корню дерева, учитываются с меньшим весом, что соответствует изложенному выше второму соображению.

Коэффициент схожести двух рубрик из одного классификатора можно было бы посчитать и без использования данных о контексте — например, на основе длины кратчайшего пути в дереве между этими рубриками (это аналог метрики WordNet similarity [47] для рубрик). Однако, для обеспечения возможности рассчитывать схожесть между рубриками разных классификаторов и для единообразия с предыдущими двумя функциями схожести определим коэффициент схожести рубрик аналогично формуле (2.5):

$$s(r_k, r_l) = \max \left\{ \hat{s}(r_k, r_l), \frac{I(\text{par}(r_k))}{|R(\text{par}(r_k))|} \hat{s}(\text{par}(r_k), r_l), \frac{I(r_k)}{|R(r_k)|} \sum_{r' \in R(r_k)} \hat{s}(r', r_l), \right. \\ \left. \frac{I(\text{par}(r_l))}{|R(\text{par}(r_l))|} \hat{s}(r_k, \text{par}(r_l)), \frac{I(r_l)}{|R(r_l)|} \sum_{r' \in R(r_l)} \hat{s}(r_k, r') \right\}. \quad (2.6)$$

В подразделе 2.6.2 будет проведено сравнение значений данной формулы с длиной пути в дереве.

Как вытекает из следующего утверждения, коэффициенты схожести между ключевым словом и рубрикой и между двумя рубриками также имеют диапазон значений $[0, 1]$ и по своим свойствам похожи на коэффициент между двумя ключевыми словами.

Утверждение 1. *Теорема 1 остаётся действительной для произвольных токенов.*

Доказательство. Докажем, что в формулах (2.5) и (2.6) все составляющие максимума, кроме первой, меньше 1. Из этого будет следовать, что $s(w_k, w_l) = 1$ тогда и только тогда, когда $\hat{s}(w_k, w_l) = 1$, то есть добавление учёта иерархической структуры дерева рубрик не порождает новых эквивалентных токенов.

Заметим, что в формуле (2.5) $I(\text{par}(r_l)) < 1$, так как у $\text{par}(r_l)$ есть как минимум одна дочерняя рубрика — сама r_l . Следовательно, для второй составляющей максимума выполнено:

$$\frac{I(\text{par}(r_l))}{|R(\text{par}(r_l))|} \hat{s}(w_k, \text{par}(r_l)) < \frac{1}{|R(\text{par}(r_l))|} \cdot 1 \leq 1.$$

Теперь рассмотрим третью составляющую максимума. Если у рубрики r_l нет дочерних, то эта составляющая равна 0. Если дочерние рубрики есть, то $I(r_l) < 1$. В этом случае

$$\frac{I(r_l)}{|R(r_l)|} \sum_{r' \in R(r_l)} \hat{s}(w_k, r') \leq \frac{I(r_l)}{|R(r_l)|} \sum_{r' \in R(r_l)} 1 = I(r_l) < 1.$$

Для формулы (2.6) доказательство аналогично. □

Следствие. *Отношение, введённое в определении 4, остаётся отношением эквивалентности для произвольных токенов.*

Итак, даны определения для коэффициента схожести двух токенов. Теперь можно определить и коэффициент схожести двух тематических портретов.

2.3. Коэффициент схожести тематических портретов

Для определения коэффициента схожести тематических портретов обычно используются представления отдельных токенов и портретов в виде векторов в евклидовом пространстве, базисом в котором являются отдельные токены. В этом случае естественной мерой близости портретов является косинус угла между соответствующими им векторами. Однако если между некоторыми токенами существует ненулевой коэффициент схожести, то их набор нельзя считать ортогональным базисом. Для учёта коэффициентов тематической схожести между парами

токенов было разработано обобщение косинусной меры — «мягкая» косинусная мера близости [48]. Классическим вариантом такой меры является следующее выражение:

$$\text{soft_cosine}_1(p, p') = \frac{\sum_{(t_i, c_i) \in p, (t'_j, c'_j) \in p'} s(t_i, t'_j) c_i c'_j}{\sqrt{\sum_{(t_i, c_i), (t_j, c_j) \in p} s(t_i, t_j) c_i c_j} \sqrt{\sum_{(t'_i, c'_i), (t'_j, c'_j) \in p'} s(t'_i, t'_j) c'_i c'_j}}.$$

Недостатком такой меры является то обстоятельство, что она не всегда принимает значения в диапазоне $[0, 1]$. Авторы показывают, что такое ограничение достигается, если $\forall t \sum_{t' \neq t} s(t, t') < 1$, но вероятность выполнения этого условия становится мала при большом количестве токенов.

Поэтому будем использовать разновидность этой меры, которая также описана в [48] и обозначается soft_cosine_2 .

Пусть p и p' — два тематических портрета. Дополним каждый портрет недостающими токенами из портретов обучающей выборки, с нулевыми весами. Тогда эти портреты можно представить в виде $\{(t_i, c_i) \mid 1 \leq i \leq n\}$ и $\{(t_i, c'_i) \mid 1 \leq i \leq n\}$. Введём следующие вспомогательные величины:

$$\begin{aligned} c_{ij} &= \sqrt{s(t_i, t_j)} \cdot \frac{c_i + c_j}{2}; \\ c'_{ij} &= \sqrt{s(t_i, t_j)} \cdot \frac{c'_i + c'_j}{2}. \end{aligned}$$

Теперь можно определить коэффициент схожести двух портретов на основе меры soft_cosine_2 следующим образом:

$$s(p, p') = \frac{\sum_{i,j} c_{ij} c'_{ij}}{\sqrt{\sum_{i,j} c_{ij}^2} \sqrt{\sum_{i,j} c'^2_{ij}}}. \quad (2.7)$$

Заметим, что знаменатель всегда больше 0, так как $c_{ii} = c_i$, $c'_{ii} = c'_i$, и в обеих суммах хотя бы одна из таких величин не равна 0.

Данный коэффициент схожести имеет значения в диапазоне от 0 до 1, как и все описанные ранее коэффициенты. Случаи, при которых он равен 1, описываются следующим утверждением.

Утверждение 2. Коэффициент схожести двух портретов равен 1 тогда и только тогда, когда они состоят из одних и тех же токенов, с точностью до замены на токены, имеющие те же коэффициенты схожести с другими токенами обоих портретов, и веса токенов во втором портрете пропорциональны весам в первом портрете ($\forall i \ c'_i = \alpha c_i$). В остальных случаях коэффициент схожести меньше 1.

Доказательство. Сначала заметим, что если при замене токена t на другой значения $s(t_i, t)$ не меняются, то не меняются и значения c_{ij}, c'_{ij} .

Далее, из неравенства Коши–Буняковского следует, что $s(p, p') \leq 1$, причём равенство достигается тогда и только тогда, когда наборы c_{ij} и c'_{ij} пропорциональны. Это эквивалентно тому, что $\forall i, j \ c'_i + c'_j = \alpha(c_i + c_j)$. При $j = i$ получаем $c'_i = \alpha c_i$, в обратную сторону утверждение также очевидно. $\square$

Для ключевых слов частным случаем замены, допускаемой формулировкой теоремы, является замена слова w_k на слово w_l , входящее в точности в те же портреты, то есть такое, что $P(w_k) = P(w_l)$. Если данное условие выполнено, то $\forall t \ P(w_k, t) = P(w_k) \cap P(t) = P(w_l) \cap P(t) = P(w_l, t)$, следовательно и $\omega(w_1, t) = \omega(w_2, t)$. Далее, $s_0(w_i, w_k) = \sum_{t \in T_{i,k}} (\omega(w_i, t) + \omega(w_k, t))$, но множества $T_{i,k}$ и $T_{i,l}$ совпадают, и для любого t из этих множеств $\omega(w_k, t) = \omega(w_l, t)$, следовательно $s_0(w_i, w_k) = s_0(w_i, w_l)$ и $s(w_i, w_k) = s(w_i, w_l)$.

2.4. Пример вычисления коэффициентов схожести

Продemonстрируем вычисление коэффициентов схожести ключевых слов и тематических портретов на следующем простом примере, представленном на рис. 2.1. В данном примере w_1, w_2, w_3, w_4 — ключевые слова, $p_{1,2}, p_{2,3}, p_{3,4}, p_{1,4}$ — содержащие их тематические портреты, все весовые коэффициенты слов в портретах равны 1. В качестве обучающей выборки будем использовать эти же четыре портрета.

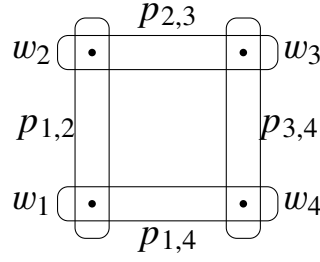


Рис. 2.1. Пример тематических портретов для расчёта коэффициентов схожести

Сначала определим значения функции ω для различных пар слов:

- для слова с самим собой: $\omega(w_1, w_1) = \frac{1}{2} + \frac{1}{2} = 1$;
- для соседних слов: $\omega(w_1, w_2) = \frac{1}{2}$;
- для противоположных слов: $\omega(w_1, w_3) = 0$, так как у них нет ни одного общего портрета.

Коэффициент схожести слова с самим собой равен 1 по Теореме 1. Найдём коэффициент схожести двух соседних слов:

$$s(w_1, w_2) = \frac{\omega(w_1, w_1) + \omega(w_2, w_1) + \omega(w_1, w_2) + \omega(w_2, w_2)}{|P(w_1)| + |P(w_2)|} = \frac{1 + \frac{1}{2} + \frac{1}{2} + 1}{2 + 2} = \frac{3}{4}.$$

Коэффициент схожести двух противоположных слов:

$$s(w_1, w_3) = \frac{\omega(w_1, w_2) + \omega(w_3, w_2) + \omega(w_1, w_4) + \omega(w_3, w_4)}{|P(w_1)| + |P(w_3)|} = \frac{\frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \frac{1}{2}}{2 + 2} = \frac{1}{2}.$$

Вычислим теперь коэффициенты схожести тематических портретов. Сначала рассмотрим пары портретов с одним общим ключевым словом. Пусть $p = p_{1,2}$, $p' = p_{2,3}$. Тогда

$$\begin{aligned} c_{11} &= c_{22} = c'_{22} = c'_{33} = \sqrt{1} \cdot \frac{1+1}{2} = 1, \\ c_{12} &= c_{21} = c'_{23} = c'_{32} = \sqrt{\frac{3}{4}} \cdot \frac{1+1}{2} = \frac{\sqrt{3}}{2}, \\ c_{13} &= c_{31} = c_{24} = c_{42} = c'_{13} = c'_{31} = c'_{24} = c'_{42} = \sqrt{\frac{1}{2}} \cdot \frac{1+0}{2} = \frac{1}{2\sqrt{2}}, \\ c_{14} &= c_{41} = c_{23} = c_{32} = c'_{12} = c'_{21} = c'_{34} = c'_{43} = \sqrt{\frac{3}{4}} \cdot \frac{1+0}{2} = \frac{\sqrt{3}}{4}, \end{aligned}$$

остальные c_{ij} , c'_{ij} равны 0. Группируя равные члены, получаем:

$$\begin{aligned}
 s(p_{1,2}, p_{2,3}) &= \frac{c_{22}c'_{22} + 2c_{12}c'_{12} + 2c_{13}c'_{13} + 2c_{23}c'_{23} + 2c_{24}c'_{24}}{\sqrt{c_{11}^2 + c_{22}^2 + 2c_{12}^2 + 4c_{13}^2 + 4c_{23}^2} \sqrt{c'_{22}^2 + c'_{33}^2 + 4c'_{12}^2 + 4c'_{13}^2 + 2c'_{23}^2}} = \\
 &= \frac{1 + 2 \cdot \frac{\sqrt{3}}{2} \cdot \frac{\sqrt{3}}{4} + 2 \cdot \frac{1}{2\sqrt{2}} \cdot \frac{1}{2\sqrt{2}} + 2 \cdot \frac{\sqrt{3}}{4} \cdot \frac{\sqrt{3}}{2} + 2 \cdot \frac{1}{2\sqrt{2}} \cdot \frac{1}{2\sqrt{2}}}{\sqrt{1 + 1 + 2 \cdot \frac{3}{4} + 4 \cdot \frac{1}{8} + 4 \cdot \frac{3}{16}} \sqrt{1 + 1 + 4 \cdot \frac{3}{16} + 4 \cdot \frac{1}{8} + 2 \cdot \frac{3}{4}}} = \\
 &= \frac{1 + \frac{3}{4} + \frac{1}{4} + \frac{3}{4} + \frac{1}{4}}{\sqrt{2 + \frac{3}{2} + \frac{1}{2} + \frac{3}{4}} \sqrt{2 + \frac{3}{4} + \frac{1}{2} + \frac{3}{2}}} = 3 \div \frac{19}{4} = \frac{12}{19}.
 \end{aligned}$$

Теперь вычислим коэффициент схожести для портретов без общих ключевых слов: $p = p_{1,2}$, $p' = p_{3,4}$. Для них имеем:

$$\begin{aligned}
 c_{11} &= c_{22} = c'_{33} = c'_{44} = \sqrt{1} \cdot \frac{1+1}{2} = 1, \\
 c_{12} &= c_{21} = c'_{34} = c'_{43} = \sqrt{\frac{3}{4}} \cdot \frac{1+1}{2} = \frac{\sqrt{3}}{2}, \\
 c_{13} &= c_{31} = c_{24} = c_{42} = c'_{13} = c'_{31} = c'_{24} = c'_{42} = \sqrt{\frac{1}{2}} \cdot \frac{1+0}{2} = \frac{1}{2\sqrt{2}}, \\
 c_{14} &= c_{41} = c_{23} = c_{32} = c'_{14} = c'_{41} = c'_{23} = c'_{32} = \sqrt{\frac{3}{4}} \cdot \frac{1+0}{2} = \frac{\sqrt{3}}{4},
 \end{aligned}$$

остальные c_{ij} , c'_{ij} равны 0. Следовательно, коэффициент схожести равен

$$\begin{aligned}
 s(p_{1,2}, p_{3,4}) &= \frac{2c_{13}c'_{13} + 2c_{14}c'_{14} + 2c_{23}c'_{23} + 2c_{24}c'_{24}}{\sqrt{c_{11}^2 + c_{22}^2 + 2c_{12}^2 + 4c_{13}^2 + 4c_{14}^2} \sqrt{c'_{33}^2 + c'_{44}^2 + 2c'_{34}^2 + 4c'_{13}^2 + 4c'_{14}^2}} = \\
 &= \frac{2 \cdot \frac{1}{2\sqrt{2}} \cdot \frac{1}{2\sqrt{2}} + 2 \cdot \frac{\sqrt{3}}{4} \cdot \frac{\sqrt{3}}{4} + 2 \cdot \frac{\sqrt{3}}{4} \cdot \frac{\sqrt{3}}{4} + 2 \cdot \frac{1}{2\sqrt{2}} \cdot \frac{1}{2\sqrt{2}}}{\sqrt{1 + 1 + 2 \cdot \frac{3}{4} + 4 \cdot \frac{1}{8} + 4 \cdot \frac{3}{16}} \sqrt{1 + 1 + 2 \cdot \frac{3}{4} + 4 \cdot \frac{1}{8} + 4 \cdot \frac{3}{16}}} = \\
 &= \frac{\frac{1}{4} + \frac{3}{8} + \frac{3}{8} + \frac{1}{4}}{\sqrt{2 + \frac{3}{2} + \frac{1}{2} + \frac{3}{4}} \sqrt{2 + \frac{3}{2} + \frac{1}{2} + \frac{3}{4}}} = \frac{5}{4} \div \frac{19}{4} = \frac{5}{19}.
 \end{aligned}$$

2.5. Оценка количества операций и способы повышения производительности

Исследование вычислительной сложности разработанных методов является актуальным в связи с требованиями по производительности, описанными выше в разделе 1.3.

Пусть в информационно-аналитической системе всего k различных токенов, в обучающей выборке m тематических портретов, в среднем каждый портрет состоит из n токенов ($n \ll k$). Естественным является предположение, что по мере добавления новых данных в CRIS-систему и пропорционального увеличения обучающей выборки величина m растёт быстрее всего, величина k растёт несколько медленнее, а величина n ограничена сверху, и её порядок не меняется. Например, в ИАС «ИСТИНА» $n \approx 13$ для портретов статей и НИР.

В этих обозначениях вероятность того, что случайно выбранный токен имеется в случайном портрете, равна $\frac{n}{k}$. Следовательно, каждый токен будет встречаться в среднем в $\frac{mn}{k}$ портретах. Определение списка портретов, содержащих конкретный токен, можно осуществить за $O(\log k)$ операций при использовании двоичного дерева поиска.

Далее, вероятность того, что два токена окажутся соседними, равна

$$1 - \left(1 - \frac{n^2}{k^2}\right)^m = O\left(\frac{mn^2}{k^2}\right).$$

Исходя из этого, будем предполагать, что $\frac{mn^2}{k^2} = O(1)$. Будем также предполагать, что все тематические портреты нормированы так, что для каждого портрета $\sum_{i,j} c_{ij}^2 = 1$, то есть знаменатель в формуле (2.7) равен 1. Нормировку можно выполнить путём деления всех c_i на квадратный корень из исходного значения этой суммы. По Утверждению 2 значения коэффициентов схожести с другими портретами при этом не изменятся. Оценим теперь сложность вычисления коэффициентов схожести.

Утверждение 3. Для вычисления коэффициента схожести двух ключевых слов требуется в среднем $O(\frac{mn^2}{k})$ операций. Для вычисления коэффициента схожести двух тематических портретов — $O(\frac{mn^4}{k})$.

Доказательство. Рассмотрим сначала коэффициент схожести двух ключевых слов. Для его вычисления требуется для каждого из общих соседей вычислить значения функции ω , заданные формулой (2.2). Для этого достаточно перебрать все портреты, содержащие первое из исходных слов, и для каждого портрета p перебрать все токены t_i в нём, добавляя $c_{t_i} \div \sum_{t \in T_p} c_t$ к величине, приписанной токenu. Таких пар (p, t_i) в среднем $\frac{mn}{k} \cdot n = \frac{mn^2}{k}$. Далее аналогично выполняется перебор всех пар (портрет, токен) для второго из исходных слов, и если у токена имелась приписанная ему величина, то сумма её и новой аналогичной величины добавляется в итоговое значение s_0 . Отсюда и вытекает требуемая оценка.

Для вычисления коэффициента схожести двух тематических портретов требуется вычислить числитель в формуле (2.7). Ненулевыми в нём являются такие слагаемые c_{ij} , что либо t_i входит в один из портретов, а t_j — в другой, либо один из токенов входит сразу в оба портрета, а другой токен имеет ненулевой коэффициент схожести с первым. Пар первого типа в среднем $O(n^2)$, и их коэффициенты схожести можно вычислить за $O(n^2 \cdot \frac{mn^2}{k}) = O(\frac{mn^4}{k})$ операций.

Токенов, входящих сразу в оба портрета, в среднем $k \cdot \frac{n}{k} \cdot \frac{n}{k} = \frac{n^2}{k}$. Чтобы вычислить для одного из таких токенов (обозначим его t_i) все ненулевые коэффициенты схожести, можно использовать следующий алгоритм. Сначала, как в предыдущем случае, приписываем каждому соседнему токenu t_j значение $\omega(t_i, t_j)$. Далее по циклу рассматриваются все соседние токены. Для каждого токена t_j все токены, соседние для него, сначала помечаются как непройденные, и им приписывается значение 0 (отдельно от значения, возможно приписанного ранее). Далее перебираются все портреты, содержащие t_j , и все токены в них. Для каждой такой пары $(p, t_l \in T_p)$ к величине, приписанной t_l , прибавляется $c_{t_j} \div \sum_{t \in T_p} c_t$. Если t_l был помечен как непройденный, то прибавляется также величина, которая была

приписана t_j на первом шаге. После этого t_l помечается как пройденный. После перебора всех таких пар значение величины, приписанной каждому токenu t_l , будет равно $s_0(t_i, t_l)$. Последним действием является деление каждой такой величины на $|P(t_i) + P(t_l)|$. Поскольку на всех шагах алгоритма перебирались лишь соседние вершины первого и второго порядка для t_i , то количество операций для вычисления всех ненулевых коэффициентов схожести t_i равно $O(\frac{mn^2}{k} \cdot \frac{mn^2}{k}) = O(\frac{m^2n^4}{k^2})$. Умножая на число различных токенов t_i , получаем $O(\frac{m^2n^6}{k^3})$. Но так как $\frac{mn^2}{k^2} = O(1)$ по описанному выше предположению, то $O(\frac{m^2n^6}{k^3}) = O(\frac{mn^4}{k})$. $\square$

Полученные оценки являются оценками в среднем, а не в худшем случае. Однако, за такое же количество операций можно получить приблизительное значение коэффициента схожести в худшем случае, если зафиксировать величину n_0 и в каждом портрете оставить не более n_0 токенов с наибольшими весовыми коэффициентами.

Кроме того, количество операций можно уменьшить, если при вычислении коэффициента схожести двух портретов принять $c_{ij} = 0$ для тех пар (i, j) , для которых $s(t_i, t_j)$ меньше некоторого заранее заданного значения ε .

Для поиска портретов, имеющих наибольшие коэффициенты схожести с заданным портретом p , методом прямого перебора всех портретов, необходимо выполнить в среднем $O(\frac{mn^4}{k} \cdot m) = O(\frac{m^2n^4}{k})$ операций, без учёта сортировки списка результатов. Здесь предполагается, что число всех портретов в CRIS-системе пропорционально числу портретов в обучающей выборке, то есть m . На практике можно получить приближённый список кандидатов за меньшее количество операций. Опишем метод, основанный на поиске портретов, включающих токены, имеющие как можно больше общих соседей с токенами из исходного портрета. Для этого фиксируются некоторые числа $A, B \in \mathbb{N}$. Метод состоит из нескольких шагов.

1. Составляется список всех соседей токенов исходного портрета, и для каждого токена t из соседей вычисляется значение $\sum_{t' \in T_p} \omega(t', t)$. После этого

вычисляется A -я порядковая статистика по убыванию этих значений и составляется список из A соседей, для которых значение не меньше этой статистики. Такой список можно составить в среднем за $O(\frac{mn^3}{k})$ операций.

2. Далее составляется аналогичный список соседей этих соседей, и для каждого соседа t вычисляются приблизительные коэффициенты схожести со всеми токенами p , отличающиеся от точных учётом только оставленных на предыдущем шаге общих соседей, и на их основе — коэффициент схожести портрета $\{(t, 1)\}$ с p . Для этого необходимо ещё $O(n \cdot A \cdot \frac{mn^2}{k}) = O(\frac{Amn^3}{k})$ операций. Из этого списка оставляются только B токенов, для которых такие коэффициенты схожести наибольшие. После этого за $O(\frac{Bmn^3}{k})$ операций вычисляются точные значения попарных коэффициентов схожести между этими B токенами и всеми токенами p .
3. На третьем шаге происходит обход портретов, содержащих токены из шага 2 (таких портретов $O(\frac{Bmn}{k})$), и для каждого портрета p' вычисляется приблизительное значение $s(p, p')$, отличающееся учётом только токенов, оставленных после шага 2. Поскольку попарные коэффициенты схожести токенов уже вычислены, то для вычисления всех коэффициентов схожести портретов необходимо $O(\frac{Bmn^3}{k})$ действий.
4. При необходимости можно оставить из полученного списка фиксированное число портретов с наибольшими приблизительными коэффициентами схожести с p и выполнить расчёт точных коэффициентов и сортировку по ним.

Без учёта последнего шага общая сложность описанного метода поиска составляет $O(\max\{A, B\} \cdot \frac{mn^3}{k})$ операций.

На практике можно принять $n = \text{const}$, поэтому полученные оценки означают возможность осуществлять сравнение двух тематических портретов и поиск портретов, похожих на заданный, за время, пропорциональное $\frac{m}{k}$. За счёт использова-

ния кэширования, индексирования и удаления несущественных компонент портретов можно добиться ещё большей производительности. Кроме того, поскольку в алгоритмах выполняются однотипные операции одновременно для разных токенов, такие операции можно выполнять параллельно на разных вычислительных узлах.

2.6. Тестирование метрики сравнения ключевых слов и рубрик

Исследование построенных функций схожести ключевых слов и рубрик было проведено на данных информационно-аналитической системы «ИСТИНА». Сведения об этой системе ранее были представлены во Введении.

2.6.1. Сравнение схожести ключевых слов с долей общих авторов

Для тестирования метрики сравнения ключевых слов были взяты ключевые слова, привязанные к публикациям и научно-исследовательским работам в ИАС «ИСТИНА». Всего в системе имеются данные о 750 тысячах привязок ключевых слов, из них 145 тысяч уникальных ключевых слов. Доли русскоязычных и англоязычных ключевых слов примерно одинаковы. В таблице 2.1 приведены примеры пар русскоязычных ключевых слов с различными коэффициентами схожести.

Косвенным показателем схожести для двух ключевых слов является доля общих авторов среди авторов всех результатов деятельности, содержащих какое-либо ключевое слово из пары. Например, похожие показатели описаны в [49]. Поэтому разработанную метрику схожести для ключевых слов можно верифицировать, сопоставив её значения с долей общих авторов.

Долю общих авторов будем вычислять как меру Жаккара для множеств авторов публикаций, использующих каждое из двух слов. Для удобства вычислений будем использовать формулу $\frac{n_1+n_2}{n_{1,2}} - 1$, где n_1 — число авторов, в публикациях которых встречается первое ключевое слово, n_2 — второе, $n_{1,2}$ — одно из двух. При

Ключевое слово 1	Ключевое слово 2	$s(w_1, w_2)$
аномалии снежности	лавинообразующие снегопады	0,9
геомагнетизм	аэроэлектричество	0,8
хронология	берестяные грамоты	0,7
эпистемология	нарративная онтология	0,6
вечная мерзлота	литосфера полярных районов	0,5
история литературы	писательские объединения	0,4
сопряженные полимеры	пористые электроды	0,3
пластовые льды	геоэкологическая безопасность	0,2

Таблица 2.1. Примеры пар ключевых слов с различной степенью схожести

таком способе подсчёта нет необходимости считать число авторов, в публикациях которых встречаются оба слова.

Сравнение приведено в таблице 2.2. Поскольку подавляющее большинство пар ключевых слов имеют низкую долю общих авторов, и лишь у примерно каждой двадцатой пары такая доля насчитывает десятки процентов, то все пары ключевых слов были разбиты на группы по логарифмической шкале для доли общих авторов. Как видно из таблицы, коэффициент схожести примерно совпадает с долей общих авторов, незначительно превышая её для последних трёх групп.

Для расчёта функции схожести двух ключевых слов использовалась программная реализация формулы (2.4) на языке запросов SQL, фрагменты которой приведены ниже в разделе 2.7.

2.6.2. Сравнение схожести рубрик с длиной пути в классификаторе

Для функции схожести между двумя рубриками было проведено сравнение с другим естественным показателем схожести — длиной пути между двумя рубриками в дереве классификатора.

Рубрики в ИАС «ИСТИНА» привязываются к нескольким типам объектов,

Доля общих авторов	Среднее значение $s(w_1, w_2)$	Число пар
> 50 %	0,611	127 030
20–50 %	0,245	203 885
10–20 %	0,145	259 230
5–10 %	0,095	356 918
2–5 %	0,058	564 886
1–2 %	0,036	381 217
< 1 %	0,034	9 734 712

Таблица 2.2. Значения схожести пар ключевых слов с различными долями общих авторов

в первую очередь к журналам (около 100 тысяч привязок) и к научно-исследовательским работам (около 170 тысяч привязок). НИР привязываются к различным рубрикам, распределение по состоянию на сентябрь 2021 года показано в таблице 2.3 (данные отсортированы по убыванию количества привязок).

Рубрикатор	Число привязок	Число рубрик
ГРНТИ	45061	7795
Социально-экономические цели	18105	24
Scopus	17435	339
УДК	16905	118902
Приоритетные направления развития МГУ до 2020 года	9281	10
Номенклатура специальностей научных работников	5866	494
Критические технологии Российской Федерации	3273	27
Рубрикатор ОЭСР	2295	300

Таблица 2.3. Количество привязок НИР к рубрикам различных рубрикаторов

Для изучения были взяты привязки к рубрикам Государственного рубрикатора научно-технической информации (ГРНТИ), так как таких привязок больше всего. Рубрикатор при этом имеет достаточно большое число рубрик (второе после

УДК).

ГРНТИ представляет собой трёхуровневый рубрикатор, где рубрики верхнего уровня имеют код АВ.00.00, среднего уровня — АВ.СD.00, нижнего уровня — АВ.СD.ЕF. Последняя версия ГРНТИ, опубликованная в 2020 году, насчитывает 8837 рубрик, но в ИАС «ИСТИНА» рубрик меньше, поскольку используется более старая версия. При этом из 7795 рубрик лишь менее половины (3628) имеют привязки к НИР, поэтому не для всех пар рубрик удалось установить коэффициент схожести.

Все возможные пары рубрик были разбиты на классы в зависимости от длины пути в дереве между ними — от 0 (если рубрики в паре совпадают) до 6 (если путь от одной рубрики до другой проходит через корень дерева). В каждом классе были вычислены коэффициенты схожести рубрик по формуле, не учитывающей связи в дереве рубрикатора (функция $\hat{s}$).

На рис. 2.2 показаны средние значения коэффициента схожести внутри каждого класса. Учитывались те пары рубрик, для которых имелось достаточное количество данных для вычисления коэффициента. Как видно, среднее значение убывает практически равномерно при увеличении длины пути.

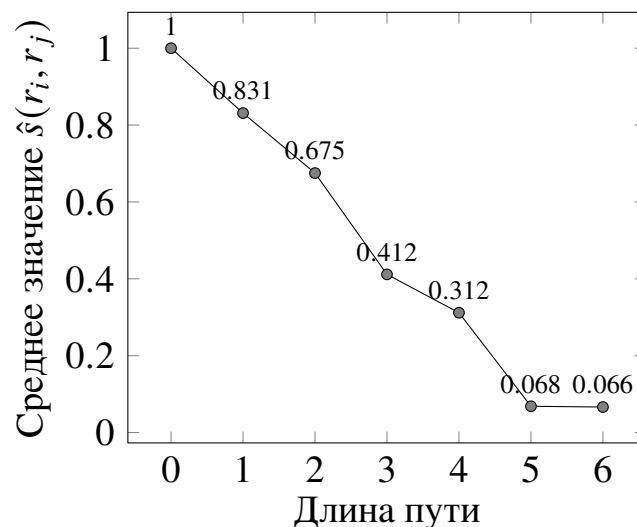


Рис. 2.2. Связь коэффициента схожести с длиной пути

2.6.3. Выявление переводов ключевых слов

Подавляющее большинство ключевых слов в данных системы «ИСТИНА» относится к одному из двух языков: русскому или английскому. Для многих слов на одном языке доступен и перевод на другой язык. Слово и его перевод являются семантически схожими, поэтому естественно ожидать, что в разработанной метрике перевод слова будет встречаться среди наиболее близких к нему токенов.

Для исследования возможности метрики выявлять переводы был проведён следующий эксперимент. Из базы данных выбиралось произвольное слово w , по набору используемых символов определялся его язык, далее составлялся список из трёх ключевых слов на другом языке, обладающих наибольшими коэффициентами схожести. Для целей сравнения также сохранялись три ключевых слова, являющихся соседними для данного и имеющих максимальное значение $\omega(w, w')$.

Приведём пример полученных данных:

- исходное ключевое слово: «полистирол-дивинилбензол»;
- первый список (по метрике схожести): «**polystyrene-divinylbenzene**», «hydrophilic interaction chromatography», «determination of anions»;
- второй список (соседние слова): «hydrophilic interaction chromatography», «ion chromatography», «**polystyrene-divinylbenzene**».

Процедура была повторена 100 раз. Среди исходных слов было 66 русскоязычных и 34 англоязычных. Далее полученные списки слов были вручную проверены на наличие переводов. Результат оказался следующий:

- В 54 % случаев перевод ключевого слова присутствовал среди трёх слов, имеющих наибольший коэффициент схожести. В 26 % случаев перевод был первым из таких слов. В некоторых случаях нашлось несколько вариантов перевода: например, «numerical modeling» и «numerical modelling».
- В 62 % случаев перевод присутствовал среди трёх слов, являющихся соседними. Такой результат является ожидаемым, поскольку в данных о статьях

и НИР часто указываются ключевые слова и на русском, и на английском, следовательно они попадают в один и тот же портрет. При этом метрика схожести, основанная на соседях второго порядка, способна определять перевод, даже если общих портретов нет. Это подтверждается тем, что в 14 % случаев метрика схожести показала лучший результат: перевод встретился в первом списке, но не во втором, либо его позиция в первом списке была выше.

- В 71 % случаев перевод встретился или в первом списке, или во втором. Таким образом, совместное использование двух метрик позволяет увеличить процент определённых переводов.

Для оставшихся 29 % случаев (29 слов) была вручную проведена проверка наличия в системе переводов. В 14 случаях перевода не имелось ни в одном портрете, следовательно и обнаружить его было невозможно. Среди таких слов были латинское название семейства («*chenopodiaceae*»), англоязычные аббревиатуры («*watem/sedem*», «*fmdv 2a*») и низкочастотные ключевые слова («лазерная корреляционная спектроскопия»).

На основе полученных данных можно утверждать, что разработанная метрика схожести показывает хорошую способность выявлять переводы ключевых слов. Это может свидетельствовать и о возможности определять другие типы семантических связей.

2.7. Программная реализация функции схожести токенов на языке SQL

Приведём пример программного кода реализации функции схожести токенов $\hat{s}(r_i, r_j)$ на языке запросов SQL. В коде будет использоваться таблица `PORTRAITTOKEN`, хранящая данные о вхождении токенов в портреты и имеющая следующие поля:

- PORTRAIT_ID — ключ тематического портрета p ;
- TOKEN — токен $t \in T_p$;
- COEFFICIENT — весовой коэффициент c_t .

Для промежуточных вычислений в реализации используются общие табличные выражения (common table expressions, CTE). Первое выражение используется для суммирования весовых коэффициентов в портрете:

```
with SUM_PORTRAIT_COEFFICIENTS as (
    select PORTRAIT_ID, sum(COEFFICIENT) CSUM
    from PORTRAITTOKEN
    group by PORTRAIT_ID),
```

Второе вспомогательное выражение используется для подсчёта числа портретов, содержащих токен:

```
NUM_TOKEN_PORTRAITS as (
    select TOKEN, count(*) CNT
    from PORTRAITTOKEN
    group by TOKEN),
```

Далее реализована функция $\omega(r_i, r_j)$:

```
OMEGA as (
    select PT1.TOKEN TOKEN1,
           PT2.TOKEN TOKEN2,
           sum(PT2.COEFFICIENT / SUM_PORTRAIT_COEFFICIENTS.CSUM)
           as VALUE
    from PORTRAITTOKEN PT1
    join PORTRAITTOKEN PT2 on PT2.PORTRAIT_ID = PT1.PORTRAIT_ID
    join SUM_PORTRAIT_COEFFICIENTS on
           SUM_PORTRAIT_COEFFICIENTS.PORTRAIT_ID = PT1.PORTRAIT_ID
    group by PT1.TOKEN, PT2.TOKEN),
```

Для увеличения производительности этот подзапрос можно превратить в материализованное представление и создать для него индексы по ключам TOKEN1 и TOKEN2.

Затем реализована функция $s_0(r_i, r_j)$:

```
S0 as (
  select OMEGA1.TOKEN1 TOKEN1,
         OMEGA2.TOKEN1 TOKEN2,
         sum(OMEGA1.VALUE + OMEGA2.VALUE) as VALUE
  from OMEGA OMEGA1
  join OMEGA OMEGA2 on OMEGA2.TOKEN2 = OMEGA1.TOKEN2
  group by OMEGA1.TOKEN1, OMEGA2.TOKEN1),
```

Наконец, итоговый коэффициент схожести (функция $\hat{s}(r_i, r_j)$) реализован следующим образом:

```
S1 as (
  select S0.TOKEN1,
         S0.TOKEN2,
         S0.VALUE / (NUM_PORTRAITS1.CNT + NUM_PORTRAITS2.CNT)
         as VALUE
  from S0
  join NUM_TOKEN_PORTRAITS NUM_TOKEN_PORTRAITS1 on
        NUM_TOKEN_PORTRAITS1.TOKEN = S0.TOKEN1
  join NUM_TOKEN_PORTRAITS NUM_TOKEN_PORTRAITS2 on
        NUM_TOKEN_PORTRAITS2.TOKEN = S0.TOKEN2)
```

2.8. Выводы

Проведён сравнительный анализ моделей для формализованного описания предметных областей. На основе такого анализа и с учётом требований, изло-

женных в разделе 1.1, введено понятие тематического портрета, состоящего из токенов — ключевых слов или рубрик классификатора, с присвоенными им весовыми коэффициентами (2.1).

Рассмотрены вопросы вычисления коэффициента, отражающего смысловую схожесть отдельных токенов и портретов целиком. Для сравнения токенов предложена модель, основанная на учёте совместного вхождения токенов в тематические портреты обучающей выборки. На основе этой модели введены функции схожести двух ключевых слов (2.4), ключевого слова и рубрики (2.5) и двух рубрик (2.6). Наконец, на основе этих функций схожести определена функция схожести двух портретов $s(p, p')$ (2.7). Показано, что эта функция имеет значения в диапазоне $[0, 1]$ и описаны случаи, когда её значение равно 1. Даны оценки вычислительной сложности и описан способ ускоренного поиска портретов, схожих с заданным. На данных наукометрической системы проведено сравнение значений функции схожести ключевых слов с долей общих авторов публикаций, включающих эти ключевые слова, а значений функции схожести рубрик — с длиной пути между рубриками в графе классификатора. Был проведён эксперимент, показавший возможность выявлять переводы ключевых слов при помощи разработанной метрики схожести. Приведён пример программной реализации функции схожести токенов на языке SQL.

Результаты этой главы будут использованы при составлении алгоритма поиска экспертов далее в главе 5.

Глава 3

Алгоритмы построения тематических портретов

В настоящей главе рассматривается вопрос построения тематического портрета объекта в информационно-аналитической системе. В простейшем случае токены (ключевые слова или рубрики) могут быть сопоставлены объекту непосредственно. Однако в CRIS-системах со сложной структурой, то есть с большим количеством типов объектов и связей между объектами, для некоторых типов токены можно получить только при помощи информации из других объектов, связанных с данным прямо или косвенно. Например, тематический портрет научно-технического журнала можно построить по данным о публикуемых в нём статьях, а тематический портрет учёного — по данным о всех результатах его деятельности.

В данной главе будет предложена общая математическая модель, при помощи которой предоставляется возможность указать связи или цепочки связей, по которым требуется перейти для построения тематического портрета, и весовые коэффициенты таких связей и цепочек. Такая модель подходит для использования как в простых, так и в сложно устроенных системах.

По связям объектов можно также расширять и уточнять существующие тематические портреты. Например, если имеется статья, портрет которой состоит из небольшого количества ключевых слов, то этот портрет можно расширить за счёт портретов публикаций, на которые данная статья ссылается. Такие статьи могут быть частым явлением в CRIS-системах, например, в ИАС «ИСТИНА» для половины статей указано 7 или менее ключевых слов.

3.1. Модель информационно-аналитической системы

Современные информационно-аналитические системы используют для хранения данных различные системы управления базами данных, которые различают-

ся по модели данных, по организации их хранения, по степени распределённости данных и по способу доступа к ним для чтения либо для внесения изменений. Приведём наиболее распространённые модели данных.

- Модель «Ключ — значение»: самая простая из возможных моделей, для получения доступа к объекту необходимо знать лишь его ключ. Данная модель, как правило, используется для временного хранения данных, к которым требуется быстрый доступ (кэш).
- Реляционная модель данных: при создании базы данных задаётся её схема, каждый хранимый объект должен быть отнесён к какому-либо из заданных в схеме отношений (таблиц), и представляется в виде кортежа из значений, соответствующих полям данного отношения (столбцам таблицы). В качестве языка для доступа к данным как правило используется язык запросов SQL. Существует стандартизированная версия SQL, но различные реализации имеют свои собственные диалекты.
- Семейство столбцов: как и в реляционной модели, используется табличная схема данных, однако предполагается, что большинство ячеек таблицы не имеют данных, поэтому данные хранятся в виде разреженной матрицы. Данная модель эффективна для построения индексов и для задач, требующих обработки транзакций в реальном времени.
- Иерархическая модель: фиксированной схемы данных нет; данные представляются в виде дерева, для доступа к элементу дерева достаточно знать путь к нему от вершины дерева. Примером такой модели являются базы, данные в которых представимы в виде документа в формате XML или JSON. В таком случае для задания путей могут быть использованы XPath и JSONPath, соответственно. Для получения данных о большом количестве вершин используются более сложные языки запросов, такие как GraphQL.

- Сетевая модель: обобщение иерархической модели; вместо дерева рассматривается ориентированный граф, где каждая вершина может иметь несколько входящих рёбер, однако не допускаются циклические пути.
- Графовые модели данных: данные представляются в виде ориентированного графа, при этом циклические пути разрешены. Наиболее распространённой моделью является RDF [50], в которой единицей данных является триплет (субъект, предикат, объект). Для доступа к данным в модели RDF разработан язык запросов SPARQL.

Иерархическая и сетевая модели являются частными случаями графовой модели. Поэтому в настоящем разделе для описания абстрактной информационно-аналитической системы будем использовать графовую модель данных. При этом, такое описание будет также применимо и к системам, использующим реляционную модель, так как данные в реляционной модели можно представить в виде графа специального типа [51].

Определение 8. *Множеством литеральных значений L будем называть некоторое заранее заданное множество (возможно, бесконечное), не привязанное к конкретной информационно-аналитической системе. Например, элементами L могут являться числа или строки — упорядоченные наборы символов из заранее определённого алфавита.*

В частности, в L входят ключевые слова и рубрики.

Определение 9. *Информационно-аналитической системой в контексте её описания в настоящей работе будем называть следующую пару множеств.*

- Множество N основных вершин графа (nodes).
- Множество E рёбер графа (edges), где каждое ребро является триплетом (тройкой) $e = (subj, pred, obj)$, где $subj, pred \in N$, $obj \in N \cup L$.¹ Элементы

¹ Более короткие обозначения s, p, o не используем, чтобы избежать совпадения обозначений для субъекта и функции схожести; для предиката и тематического портрета.

триплетов будем называть *субъект*, *предикат* и *объект*, соответственно.

В графовом представлении вершины графа будут элементами множеств N и L , а каждый триплет будет соответствовать направленному ребру: $subj \xrightarrow{pred} obj$. Элементы множества L при этом будут листовыми вершинами, то есть для них не будет исходящих рёбер.

Будем также предполагать, что каждая вершина относится к определённому *типу*. Например, в CRIS-системах типами вершин могут являться «статья», «автор», «конференция» и т. п. Информацию о типах можно хранить непосредственно в графе — для этого нужно добавить вершины, соответствующие типам, и ввести специальный предикат «имеет тип» (в модели RDF такой предикат называется `rdf:type`).

Определение 10. Областью значений предиката $pred$ будем называть множество $range(pred) = \{obj \in N \cup L \mid \exists subj : (subj, pred, obj) \in E\}$.

3.2. Построение портрета по связям в информационно-аналитической системе

Будем считать, что токены (ключевые слова и рубрики) являются литеральными значениями, которые могут быть сопоставлены вершинам при помощи различных предикатов (то есть для таких вершин существуют триплеты $(subj, pred, t)$, где t — токен). В простейшем случае для построения портрета необходимо указать один или несколько предикатов, чтобы получить список токенов для вершины, и весовые коэффициенты для каждого предиката, чтобы построить на их основе тематический портрет. Однако, если для вершины нет токенов, соединённых с ней всего одним предикатом, то возникает необходимость учёта токенов, соединённых с рассматриваемой вершиной цепочкой рёбер.

Заметим, что рёбра в каждой цепочке необязательно направлены в одном направлении, то есть возможны цепочки, в которых есть рёбра с направлением как

в сторону рассматриваемой вершины (портрет которой строится), так и в сторону токена. Будет удобно привести все рёбра к единому направлению. Для этого для каждого предиката $pred$ дополним множество N обратным предикатом $pred^{-1}$ и для каждого триплета $(subj, pred, obj)$, такого что $obj \in N$, дополним множество E триплетом $(obj, pred^{-1}, subj)$.

После этого любую такую цепочку, если в ней для каждой связи с направлением к рассматриваемой вершине заменить $pred_i$ на $pred_i^{-1}$, можно представить в виде последовательности рёбер, направленных от рассматриваемой вершины к токenu:

$$subj \xrightarrow{pred_1} obj_1 \xrightarrow{pred_2} \dots \xrightarrow{pred_m} t,$$

где $subj$ — рассматриваемая вершина, t — токен.

Однако, для решаемой задачи больший интерес будет представлять не направление ребра, а тип связи: сколько существует вершин, связанных с данной связью того же типа и того же направления.

Определение 11. *Образом вершины $subj \in N$ при предикате $pred$ будем называть множество $pred(subj) = \{obj \in N \cup L : (subj, pred, obj) \in E\}$.*

Прообразом вершины $obj \in N \cup L$ при предикате $pred$ будем называть множество $pred^{-1}(obj) = \{subj \in N : (subj, pred, obj) \in E\}$.

Заметим, что при $obj \in N$ это определение согласуется с обозначением для обратного предиката $pred^{-1}$, то есть образ вершины при предикате $pred^{-1}$ является её прообразом при предикате $pred$, и наоборот.

Для каждого триплета $(subj, pred, obj)$, в зависимости от того, состоят ли $pred(subj)$ и $pred^{-1}(obj)$ из одного элемента или из нескольких, можно отнести связь к виду «один к одному», «один ко многим», «многие к одному» либо «многие ко многим». В CRIS-системах встречаются связи всех четырёх видов, например:

- связь между конференцией и сборником её трудов является связью «один к одному»;

- связь между журналом и опубликованными в нём статьями является связью «один ко многим»;
- связь между докладом на конференции и самой конференцией является связью «многие к одному»;
- связь между статьями и авторами является связью «многие ко многим», так как у статьи может быть несколько авторов, а у автора может быть несколько статей.

Для построения тематического портрета вершины *subj* требуется, чтобы было заранее задано множество упорядоченных наборов предикатов (наборы будем записывать в виде $(pred_1, \dots, pred_m)$), которые могут быть использованы для «передачи» сопоставленных токенов от одной вершины к другой. Каждый предикат из набора «передает» токен от какой-то вершины к её прообразу при соответствующем этому элементу предикате, то есть в направлении, противоположном ребру. Кроме того, для каждого набора должен быть задан весовой коэффициент $\sigma(pred_1, \dots, pred_m)$, и сумма коэффициентов всех наборов должна быть равна 1.

Список наборов составляется с учётом специфики конкретной CRIS-системы. В простейшем случае можно взять все возможные наборы предикатов, где m меньше некоторого заранее определённого значения.

Будем исходить из того, что для видов связей «многие к одному» и «многие ко многим» при передаче токена от вершины ко множеству вершин, являющемуся её прообразом, вес токена распределяется равномерно между этими вершинами (и для связей «многие к одному» сумма весов токена у новых вершин равна исходному весу токена). Кроме того, для связей «один ко многим» и «многие ко многим» вес токена у очередной вершины в цепочке должен быть тем больше, чем больше вершин в её образе имеют этот токен (и для связей «один ко многим» вес токена у новой вершины равен сумме весов этого токена у вершин образа). Исходный вес токена у вершины, соответствующей ему, будем считать равным 1. На основании этих соображений определим *вес ребра* $subj \xrightarrow{pred} obj$ следующим

образом:

$$\Omega(subj, pred, obj) = \frac{1}{|pred(subj)| \cdot |pred^{-1}(obj)|}. \quad (3.1)$$

Дальнейший алгоритм для построения тематического портрета вершины $subj$ по цепочкам рёбер следующий.

1. Для каждого набора предикатов $pred_1, \dots, pred_m$ составляются все возможные цепочки вершин $obj_1, \dots, obj_m$, где $obj_i \in pred_i(obj_{i-1})$ для $1 \leq i \leq m$, $obj_0 = subj$, $obj_m \in L$ — некоторый токен.

Для каждой цепочки l через $t(l)$ будем обозначать токен, которым она заканчивается. Через T_{subj} обозначим множество всех таких токенов: $T_{subj} = \{t_1, \dots, t_n\}$.

2. Для каждой цепочки вершин определим её вес как коэффициент соответствующего набора предикатов, умноженный на произведение весов всех входящих в цепочку рёбер:

$$\begin{aligned} \Omega(l: obj_0 \xrightarrow{pred_1} obj_1 \xrightarrow{pred_2} \dots \xrightarrow{pred_m} obj_m) = \\ = \sigma(pred_1, \dots, pred_m) \cdot \prod_{i=1}^m \Omega(obj_{i-1}, pred_i, obj_i). \end{aligned}$$

3. Для каждого токена $t_j \in T_{subj}$ определим его коэффициент в портрете как сумму весов всех цепочек, заканчивающихся этим токеном:

$$c_j = \sum_{l: t(l)=t_j} \Omega(l).$$

4. В качестве тематического портрета $subj$ возьмём множество

$$p(subj) = \{(t_1, c_1), \dots, (t_n, c_n)\}. \quad (3.2)$$

Покажем, что полученный таким образом портрет удовлетворяет определению 2. Для этого требуется, чтобы все c_j были не больше 1. Докажем более сильное утверждение.

Теорема 2. Пусть тематический портрет $p(subj)$ построен в соответствии с алгоритмом, описанным выше. Тогда в формуле (3.2) $\sum_{j=1}^n c_j \leq 1$.

Доказательство.

$$\sum_{j=1}^n c_j = \sum_{j=1}^n \sum_{l: t(l)=t_j} \left(\sigma(pred_1^l, \dots, pred_m^l) \cdot \prod_{i=1}^m \frac{1}{|pred_i^l(obj_{i-1}^l)| \cdot |(pred_i^l)^{-1}(obj_i^l)|} \right).$$

Рассмотрим вклад всех цепочек с конкретными предикатами $pred_1, \dots, pred_m$ в эту сумму и покажем, что он не превосходит $\sigma(pred_1, \dots, pred_m)$. Этот вклад равен

$$\begin{aligned} \sum_l \left(\sigma(pred_1, \dots, pred_m) \cdot \prod_{i=1}^m \frac{1}{|pred_i(obj_{i-1}^l)| \cdot |pred_i^{-1}(obj_i^l)|} \right) = \\ = \sigma(pred_1, \dots, pred_m) \cdot \sum_l \prod_{i=1}^m \frac{1}{|pred_i(obj_{i-1}^l)| \cdot |pred_i^{-1}(obj_i^l)|}, \end{aligned}$$

где суммирование производится по всем цепочкам l с данными предикатами, начинающимся в $subj$. Рассмотрим второй множитель. Так как $|pred_i^{-1}(obj_i^l)| \geq 1$, то для него справедливо неравенство

$$\begin{aligned} \sum_l \prod_{i=1}^m \frac{1}{|pred_i(obj_{i-1}^l)| \cdot |pred_i^{-1}(obj_i^l)|} &\leq \sum_l \prod_{i=1}^m \frac{1}{|pred_i(obj_{i-1}^l)|} = \\ &= \sum_{obj_1 \in pred_1(obj_0)} \left(\frac{1}{|pred_1(obj_0)|} \cdots \left(\sum_{obj_m \in pred_m(obj_{m-1})} \frac{1}{|pred_m(obj_{m-1})|} \right) \cdots \right). \quad (3.3) \end{aligned}$$

Рассмотрим выражение в самых внутренних скобках. Если $pred_m(obj_{m-1}) = \emptyset$, то оно равно 0, а в противном случае оно равно

$$\begin{aligned} \sum_{obj_m \in pred_m(obj_{m-1})} \frac{1}{|pred_m(obj_{m-1})|} &= \frac{1}{|pred_m(obj_{m-1})|} \cdot \sum_{obj_m \in pred_m(obj_{m-1})} 1 = \\ &= \frac{1}{|pred_m(obj_{m-1})|} \cdot |pred_m(obj_{m-1})| = 1. \end{aligned}$$

Таким образом, умножение на это выражение можно удалить, и общая сумма от этого не уменьшится. После удаления в самых внутренних скобках окажется

выражение

$$\sum_{obj_{m-1} \in pred_{m-1}(obj_{m-2})} \frac{1}{|pred_{m-1}(obj_{m-2})|},$$

которое аналогично равно 0 либо 1. Продолжая такие упрощения, при каждом из которых второй множитель не уменьшится, в итоге получим, что весь этот множитель равен 0 (если ни одной подходящей цепочки не существует) либо 1. Следовательно, изначально он не превосходил 1. Таким образом, вклад всех цепочек с данными предикатами не превосходит $\sigma(pred_1, \dots, pred_m)$. Поэтому итоговая сумма $\sum_{j=1}^n c_j$, равная сумме вкладов всех цепочек, не превосходит

$$\sum_l \sigma(pred_1^l, \dots, pred_m^l),$$

но это выражение равно 1 по построению. $\square$

Справедливо также следующее симметричное утверждение.

Теорема 3. *Сумма вкладов произвольного токена в тематические портреты, построенные по общему множеству наборов предикатов, не превосходит 1.*

Доказательство. Теорема доказывается аналогично Теореме 2, только при построении неравенства в (3.3) из знаменателя удаляется не второй множитель, а первый, и сумма произведений разлагается в цепочку сумм не в прямом порядке, а в обратном (самое внешнее суммирование будет по $obj_{m-1} \in pred_m^{-1}(obj_m)$, самое внутреннее — по $obj_0 \in pred_1^{-1}(obj_1)$). $\square$

3.3. Примеры построенных тематических портретов

Будем строить портреты диссертационных советов по наборам предикатов, которые схематично представлены на рис. 3.1. Двухнаправленными стрелками обозначены связи типа «многие ко многим».

Всего в данной схеме используется 6 наборов. Приведём один из них в явной форме: «является членом»⁻¹, «является автором», «имеет ключевое слово». Коэф-



Рис. 3.1. Наборы предикатов, используемые для построения тематических профилей диссертационных советов

фициенты двух наборов, содержащих предикат «является руководителем», положим равными 0,1, остальных четырёх наборов — 0,2.

По приведённым наборам предикатов с использованием данных ИАС «ИСТИНА» были построены тематические портреты диссертационных советов, функционирующих в МГУ имени М. В. Ломоносова. В таблицах 3.1, 3.2 и 3.3 приведены фрагменты портретов нескольких диссертационных советов, соответствующие десяти ключевым словам с наибольшими весовыми коэффициентами.

Для каждой пары диссертационных советов МГУ был также вычислен коэффициент схожести между их тематическими портретами. Наибольший коэффициент схожести оказался у пары МГУ.23.01 (специальности: политическая психология; конфликтология) и МГУ.23.02 (политические институты, процессы и технологии; политическая регионалистика, этнополитика) — он составляет 0,414. Для совета МГУ.05.01 (см. табл. 3.1) самым близким оказался совет МГУ.01.09 (дифференциальные уравнения, динамические системы и оптимальное управление; математическое моделирование, численные методы и комплексы программ) с коэффициентом схожести 0,087.

Ключевое слово	Весовой коэффициент
конечный автомат	0,0154
математика	0,011
сложность	0,0076
функциональная система	0,0076
software engineering	0,0062
приближённые вычисления	0,0061
numerical algorithm	0,006
базы данных и знаний	0,0059
теория кодирования	0,0056
численные методы	0,0053

Таблица 3.1. Фрагмент портрета для диссертационного совета МГУ.05.01 (специальности: методы и системы защиты информации, информационная безопасность; теоретические основы информатики; вычислительная математика; дискретная математика и математическая кибернетика).

Ключевое слово	Весовой коэффициент
логика	0,0217
понятие общества	0,0175
социальная теория	0,0155
философия истории	0,0113
национальные ценности	0,0112
цивилизационное развитие	0,0112
социальная философия	0,0107
философия политики	0,0095
философия	0,007
философия науки	0,0067

Таблица 3.2. Фрагмент портрета для диссертационного совета МГУ.09.01 (специальности: онтология и теория познания; логика; философия науки и техники; социальная философия).

Ключевое слово	Весовой коэффициент
романские языки	0,0175
германские языки	0,0153
шведский язык	0,0139
французский язык	0,0129
языкознание	0,0128
индоевропейские языки	0,0115
преподавание и методика изучения языка	0,0101
history of germanic and celtic languages	0,0095
западногерманские языки	0,0084
английский язык	0,008

Таблица 3.3. Фрагмент портрета для диссертационного совета МГУ.10.07 (специальности: германские языки; романские языки).

3.4. Выводы

В главе 3 представлено математически формализованное описание информационно-аналитической системы, основанное на графовой модели данных (Определение 9). Такая модель соответствует требованиям к использованию данных, изложенным в разделе 1.1. В терминах этой модели сформулирована задача построения тематического портрета для вершины на основе токенов, находящихся в графе на некотором расстоянии от заданной вершины.

Предложен алгоритм, решающий эту задачу при заданном множестве упорядоченных наборов предикатов с весовыми коэффициентами, на основе которых строятся все возможные цепочки рёбер от заданной вершины к токенам. При этом весовой коэффициент токена принимается равным 1 в конце цепочки и при движении к её началу уменьшается в зависимости от количества «соседних» рёбер с таким же типом связи, как у рёбер цепочки (3.1). Доказано, что благодаря этому, сумма весовых коэффициентов в построенном портрете (3.2) не превосходит 1

(Теорема 2), и суммарный вклад одного токена в тематические портреты вершин также не превосходит 1 (Теорема 3).

В качестве иллюстрации приведены построенные тематические портреты диссертационных советов МГУ, а также рассчитаны коэффициенты схожести между ними.

Глава 4

Вычисление коэффициента значимости вершин

Как было изложено во Введении, при решении различных задач в CRIS-системах целесообразно учитывать значимость тех или иных объектов. При этом должна предоставляться возможность динамического указания правил для вычисления *коэффициентов значимости* в соответствии с требованиями, приведёнными в разделе 1.2. Основное внимание в настоящей главе будет посвящено разработке модели, которая позволит представить такие правила в формализованном и доступном для редактирования виде. При формировании правил в рамках такой модели будет возможность учёта значений предикатов для самой вершины и для вершин, соединённых с ней путём определённого вида в графе.

В применении к задаче поиска экспертов коэффициенты значимости будут определять, должен ли учитываться соответствующий вершине результат деятельности при поиске, а также влиять на его вклад в ранги потенциальных экспертов в поисковой выдаче. Коэффициенты значимости вершин могут быть основаны на наукометрических и других численных показателях соответствующих результатов деятельности.

Коэффициенты значимости важны для поиска экспертов, так как позволяют по-разному учитывать результаты, для получения которых необходимы серьёзные знания и высокая квалификация в заданной предметной области, и результаты, которые можно получить и при более низкой квалификации. Например, как правило, чем более авторитетным считается журнал в своей предметной области, тем больших знаний и, соответственно, трудозатрат требуется для публикации в нём статьи. Авторитетность журнала нельзя измерить напрямую, но можно оценить при помощи вычисляемых показателей, таких как импакт-фактор и позиция (квартиль) по импакт-фактору внутри заданной предметной области. Кроме того, в некоторых случаях коэффициенты позволяют учесть вклад автора в создание

результата деятельности и распределить балл за результат между несколькими авторами в соответствии с их вкладом. Подчеркнём, что в рамках настоящей работы нет цели разработать новые наукометрические показатели. Разрабатываемая модель не будет зависеть от специфики конкретных показателей, поэтому её можно будет использовать с теми показателями, данные о которых имеются в конкретной информационно-аналитической системе.

Помимо разработки модели, в данной главе будет предложен алгоритм для практического вычисления коэффициентов значимости вершин в соответствии с заданными правилами в информационно-аналитических системах, использующих СУБД, основанные на реляционной модели данных.

4.1. Модель для задания правил вычисления коэффициента

В настоящем разделе приведена математическая формализация правил, определяющих критерии отбора вершин в информационно-аналитической системе и их коэффициенты значимости. Ключевыми элементами правил являются *вычисляемые выражения* — функции, основанные на значениях предикатов для различных вершин графа. Каждое такое выражение будет несложно представить в виде функции на одном из языков запросов, например SQL. При этом такая модель позволит представить широкий класс функций, в том числе востребованных в рамках решения задачи поиска экспертов.

Как и в предыдущей главе, будем считать, что каждая вершина информационно-аналитической системы относится к некоторому типу. При этом тип будем отождествлять с множеством вершин, относящихся к нему, то есть будем использовать обозначения вида $obj \in \tau$, где obj — вершина, τ — тип. Типы могут образовывать иерархию — множества, соответствующие им, могут быть вложены друг в друга.

Далее через 2^M будем обозначать множество подмножеств множества M . Одновременно введём два вида выражений, тесно связанных друг с другом.

- *Пути доступа* — выражения, задающие функции, сопоставляющие вершине типа τ либо конечное множество других вершин из N ($f: \tau \rightarrow 2^N$), либо некоторое (также конечное) множество литеральных значений ($f: \tau \rightarrow 2^L$). Путь доступа будем называть *однозначным*, если такие множества состоят не более, чем из одного элемента: $\forall obj \in \tau \ |f(obj)| \leq 1$.
- *Вычисляемые выражения* — выражения, задающие функции, сопоставляющие вершине типа τ некоторое вещественное число: $f: \tau \rightarrow \mathbb{R}$.

Определение 12. Путь доступа определим рекурсивно следующим образом.

1. Предикат, определённый для вершин типа τ — путь доступа (как и в главе 3, предполагаем, что для каждого предиката $pred$ имеется обратный предикат $pred^{-1}$).
2. Пусть f_1 и f_2 — два пути доступа. Тогда композиция $f' = f_2 \circ f_1$, определённая как $f'(subj) = \bigcup_{obj \in f_1(subj)} f_2(obj)$, — путь доступа.
3. Пусть $f_1: \tau \rightarrow 2^M$ — путь доступа, сопоставляющий вершинам некоторые литеральные значения из подмножества $M \subseteq L$, $f_2: M \rightarrow L$ — функция, определённая для всех таких значений. Тогда композиция $f' = f_2 \circ f_1$, определённая как $f'(subj) = \{f_2(obj) \mid obj \in f_1(subj)\}$, — путь доступа.
4. Пусть f_1 — путь доступа, f_2 — вычисляемое выражение, имеющее булев тип, то есть принимающее значения 0 и 1. Тогда выражение $f': f'(subj) = \{obj \in f_1(subj) \mid f_2(obj) = 1\}$ — путь доступа (выражение f_2 в этом случае будем называть *выражением фильтрации*).

Определение 13. Вычисляемое выражение определим рекурсивно следующим образом.

1. Константа $a \in \mathbb{R}$ — вычисляемое выражение.

2. Пусть $f_1: \tau \rightarrow 2^{\mathbb{R}}$ — путь доступа со значениями в $\mathbb{R}$, а также задана функция $f_2: 2^{\mathbb{R}} \rightarrow \mathbb{R}$, определённая для конечных подмножеств. Тогда композиция $f' = f_2 \circ f_1$ является вычислимым выражением для типа τ . Функцию f_2 в этом случае будем называть *агрегатной функцией*. Примеры агрегатных функций:

- $f_{\text{count}}(\{a_1, \dots, a_n\}) = n$;
- $f_{\text{sum}}(\{a_1, \dots, a_n\}) = \sum_{i=1}^n a_i$; $f_{\text{sum}}(\emptyset) = 0$;
- $f_{\text{max}}(\{a_1, \dots, a_n\}) = \max_{1 \leq i \leq n} a_i$; $f_{\text{max}}(\emptyset) = 0$;
- $f_{\text{min}}(\{a_1, \dots, a_n\}) = \min_{1 \leq i \leq n} a_i$; $f_{\text{min}}(\emptyset) = 0$.

В частности, однозначный путь доступа со значениями в $\mathbb{R}$ сам по себе является вычислимым выражением, так как в качестве агрегатной функции можно взять f_{sum} .

3. Пусть f_1 и f_2 — два вычисляемых выражения для одного типа, g — некоторый бинарный оператор ($g: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$). Тогда функция $f' : f'(\text{subj}) = g(f_1(\text{subj}), f_2(\text{subj}))$ — вычисляемое выражение. Примеры бинарных операторов:

- арифметические операторы: $+$, $-$, $\times$, $\div$ (при условии, что функция-делитель не принимает значение 0);
- логические операторы, применимые к вычислимым выражениям, имеющим булев тип: $\wedge$, $\vee$;
- операторы сравнения (сами имеют булев тип: принимают значение 1, если сравнение истинно, и 0 в противном случае): $=$, $<$, $\leq$, $>$, $\geq$;
- операторы взятия большего или меньшего значения из двух ($\max$, $\min$).

4. Пусть f_1 — вычисляемое выражение, f_2 — путь доступа для того же типа со значениями в $2^{\mathbb{R}}$. Тогда следующая функция f' является вычислимым

выражением (имеющим булев тип):

$$f'(subj) = \begin{cases} 1 & \text{если } f_1(subj) \in f_2(subj), \\ 0 & \text{иначе.} \end{cases}$$

Благодаря введённым вспомогательным понятиям стало возможным определить сами правила.

Определение 14. *Правилом* будем называть пару (τ, f) , где:

- τ — некоторый тип вершин информационно-аналитической системы;
- $f: \tau \rightarrow \mathbb{R}$ — вычисляемое выражение, значения которого будут коэффициентами значимости вершин.

Будем называть правило (τ, f) *применимым к вершине* obj , если $obj \in \tau$. *Результатом применения* правила к вершине будем называть число $f(obj)$.

Правила могут быть объединены в наборы. Внутри набора каждому типу вершин может соответствовать одно или несколько правил. В случае, если типу вершины соответствует несколько правил из набора, то в качестве коэффициента значимости данной вершины берётся максимум из значений вычисляемых выражений этих правил. В применении к поиску экспертов, равенство коэффициента значимости вершины нулю будет означать, что соответствующий ей результат деятельности не следует учитывать при поиске.

4.1.1. Оценки количества операций

Рассмотрим вычислительную сложность применения вычисляемого выражения к вершине, что является основной составляющей для вычисления её коэффициента значимости. Исследование сложности является актуальным в связи с требованиями, описанными выше в разделе 1.3.

Среди всех операций выделим отдельно операцию взятия образа вершины при предикате. Пути доступа являются обобщением цепочек предикатов, которые

можно получить, используя лишь первые два способа построения путей, а для таких цепочек данная операция будет единственной используемой. Кроме того, можно предположить, что эта операция будет более затратна, чем арифметические операции: например, если данные о вершинах (множество N) и о триплетах (множество E) хранятся раздельно, то при использовании двоичного дерева поиска стоимость поиска исходящих из вершины рёбер будет равна $O(\log |N|)$. На практике количество операций будет определяться используемыми в СУБД моделью данных, индексами и оптимизациями. В реляционной модели данных операция взятия образа вершины при предикате будет соответствовать операции соединения (JOIN).

За $p(f)$ обозначим среднее (по вершинам $obj \in N$) число взятий образа вершины, необходимое для вычисления пути доступа или вычислимого выражения f , за $q(f)$ обозначим среднюю мощность получившегося в результате вычисления пути доступа множества. Тогда справедливо следующее утверждение, вытекающее из построений.

Утверждение 4. *Для путей доступа, построенных описанным выше первым способом, $p(f) = 1$, третьим способом — $p(f) = p(f_1)$, вторым и четвёртым способом — $p(f) = p(f_1) + q(f_1) \cdot p(f_2)$ (в предположении, что f_1 и f_2 независимы, то есть значение $p(f_2)$ при расчёте по вершинам из образа такое же, как и по всем вершинам из N).*

Для вычисляемых выражений, построенных первым способом, $p(f) = 0$, вторым — $p(f) = p(f_1)$, третьим и четвёртым — $p(f) = p(f_1) + p(f_2)$.

4.1.2. Возможности использования вычисляемых выражений для задачи поиска экспертов

Вычисляемые выражения представляют собой инструментарий для сопоставления вершинам чисел, учитывающих значения предикатов для самих вершин и их соседей в графе. Предлагаемая ниже схема может быть использована для постро-

ения таких выражений, которые будут эффективны при решении задачи поиска экспертов.

1. Взятие в качестве начального коэффициента значимости для вершин данного типа некоторой вещественной константы.
2. Умножение на вычислимое выражение, характеризующее значимость результата, если такое выражение существует для данного типа. Это может быть число цитирований для статей, импакт-фактор для журналов, объём финансирования для участия в НИР и т. п.
3. В случае, если часть вершин данного типа необходимо вообще не учитывать при поиске экспертов, это может быть достигнуто путём умножения на одно или несколько вычисляемых выражений, принимающих значения 0 и 1. Такие выражения будем называть *ограничивающими*.

Например, если следует исключить все публикации, число страниц которых меньше n , то ограничивающее выражение можно получить путём применения оператора « $\geq$ » к однозначному пути доступа, соответствующему предикату «число страниц», и константе n .

Можно также определить множество не исключённых, а включённых вершин: например, задать условие, что журнал, в котором опубликована статья, должен входить в заданный список или индекс цитирования (например, список журналов, рекомендованных ВАК, или индекс RSCI Web of Science).

4. Учёт типа авторства и вклада автора для объектов, имеющих несколько авторов. Учёт вклада в простых случаях может быть достигнут путём деления на число соавторов, то есть на результат применения агрегатной функции f_{count} к пути доступа, соответствующему предикату, обратному к «является автором».

В некоторых информационно-аналитических системах связь между результатом деятельности и авторами задана не простым предикатом, а при помо-

щи промежуточных вершин, соответствующих авторствам. Для таких вершин может быть указана дополнительная информация, такая как порядковый номер автора, тип участия или доля участия автора в создании результата. Например, для книг помимо участия в качестве автора возможно участие в роли редактора, рецензента или переводчика. Для НИР возможно участие в качестве руководителя, исполнителя либо ответственного исполнителя, а также для каждого исполнителя может быть указан коэффициент трудового участия.

Такую информацию можно учесть, если использовать правила, определённые для типов вершин, соответствующих не результатам, а авторствам. Пути доступа для результатов можно превратить в пути доступа для авторств, взяв их композиции с предикатом, соответствующим переходу от авторства к результату. При этом появляется дополнительная возможность использовать предикаты для авторств, например, дать первому автору больший балл, чем остальным, либо умножить вычисляемое выражение на долю участия автора.

5. Если требуется, чтобы коэффициент значимости одного результата деятельности был ограничен сверху (или снизу) константой n независимо от его наукометрических показателей, то такое ограничение может быть достигнуто путём применения оператора $\min$ ($\max$) к литеральному значению n и исходному вычисляемому выражению.

При помощи ограничивающих выражений представляется возможным разделить некоторый тип на подтипы, для каждого из которых будет использоваться отдельное вычисляемое выражение. Таким образом можно, например, использовать различные выражения для кандидатских и докторских диссертаций или для участия в НИР в качестве руководителя, исполнителя либо ответственного исполнителя.

Приведём пример вычисляемого выражения для вершин типа «статья в журнале», которое построено по описанной схеме и может быть использовано в рамках

задачи поиска экспертов: «импакт-фактор журнала за 2020 год, разделённый на число соавторов статьи, при условии, что объём статьи не менее 5 страниц».

Такое выражение можно представить в виде $A \div B \times C$, где A , B и C — следующие вычислимы выражения:

- A — путь доступа, являющийся композицией предикатов «(статья) опубликована в журнале» и «(журнал) имеет импакт-фактор», к которому применено выражение фильтрации, построенное при помощи применения оператора « $=$ » к однозначному пути доступа «год импакт-фактора» и константе 2020;
- B — результат применения агрегатной функции f_{count} к пути доступа, соответствующему предикату, обратному к «является автором»;
- C — результат применения оператора « $\geq$ » к однозначному пути доступа, соответствующему предикату «число страниц», и константе 5.

Приведём список свойств статей в журналах, которые могут быть использованы в вычисляемых выражениях в ИАС «ИСТИНА». Часть из этих свойств являются предикатами, определёнными для журналов, остальные — предикатами для самих статей.

- год публикации;
- число страниц, число печатных листов;
- число соавторов, порядковый номер автора;
- типы статьи (научная статья, научная хроника, научно-популярная статья, рецензия, краткое сообщение, тезисы);
- число цитирований статьи;
- наличие индексации статьи в Web of Science, Scopus;
- язык журнала (русский, иностранный);
- вхождение журнала в перечень рекомендованных ВАК;
- импакт-фактор журнала (Web of Science, SJR, РИНЦ);

- позиция журнала по импакт-фактору в его предметной области (Web of Science, SJR);
- наличие у статьи DOI, наличие у журнала ISSN;
- является ли статья переводом, имеет ли переводы.

4.1.3. Дополнительные атрибуты набора правил

Набор правил может дополнительно содержать ограничивающие выражения, общие для всех правил. Можно выделить следующие далее два таких ограничения, полезных для задачи поиска экспертов.

- Ограничение на временной диапазон результатов деятельности, например, включение результатов только за последние 5 лет. Это устранил ситуацию, при которой эксперты, начавшие научную деятельность раньше, получают заведомое преимущество.
- Требование наличия верификации результатов деятельности. В некоторых CRIS-системах, например в ИАС «ИСТИНА», результаты деятельности могут быть подтверждены либо отклонены ответственными от структурных подразделений организации за сопровождение данных. Требование наличия таких подтверждений позволит гарантировать достоверность информации, используемой при поиске экспертов.

4.2. Алгоритм для вычисления коэффициента в реляционной СУБД

В данном разделе будет представлен предлагаемый автором подход к тому, как графовая модель данных и конструкции, описанные в её терминах, могут быть применены в CRIS-системах, работающих с системами управления базами данных (СУБД), использующими реляционную модель и язык запросов SQL. Отметим базовые положения такого соответствия:

- отношения (таблицы) соответствуют типам в графовой модели;
- каждый кортеж отношения (строка таблицы) соответствует некоторой вершине графа;
- каждый атрибут отношения (столбец таблицы) соответствует некоторому предикату. В случае, если атрибут является внешним ключом (то есть ссылкой на другое отношение), то область значений предиката *pred* будет подмножеством типа, соответствующему этому отношению, иначе область значений будет подмножеством *L*.

Однако, для достижения взаимно-однозначного соответствия между кортежами и вершинами не каждому типу должно соответствовать отношение. А именно, если типы образуют иерархию с несколькими уровнями, то для любого пути в дереве типов от корня к листовой вершине отношение должно соответствовать ровно одному элементу пути. На рис. 4.1 представлен пример дерева типов, удовлетворяющего такому условию. Закрашены типы, которым соответствует отношение в реляционной модели.

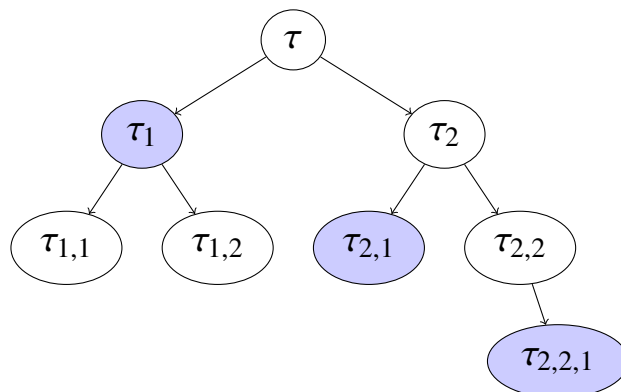


Рис. 4.1. Пример соответствия типов отношениям в реляционной модели

Будем считать, что для типов, соответствующих отношениям, в графе информационно-аналитической системы есть информация о названии отношения, и для каждого предиката, определённого для такого типа, есть информация о названии атрибута. Такую информацию можно добавить при помощи специальных предикатов *dbRelation* и *dbAttribute*.

В реляционных СУБД, как правило, используется язык запросов SQL. Текст запроса на получение данных включает в себя три основных раздела: SELECT . . . FROM . . . WHERE Пути доступа могут быть представлены в виде SQL-запросов, раздел SELECT которых состоит из одного выражения (то есть выбирающих один столбец). Вычисляемые выражения могут быть представлены в виде SQL-выражений — конструкции языка SQL, которую можно использовать как элемент раздела SELECT. Запрос, выбирающий 1 столбец и не более 1 строки, является частным случаем выражения.

Опишем, какие конструкции языка SQL соответствуют различным вариантам построения путей доступа, описанным в разделе 4.1.

1. Предикат *pred* для типа τ — в разделе FROM указывается *dbRelation*(τ), в разделе SELECT указывается *dbAttribute*(*pred*), раздел WHERE пустой.
2. Композиция двух путей доступа — за основу берётся запрос, соответствующий второму пути доступа, но в разделе FROM исходное название отношения заменяется на вложенный запрос, соответствующий первому пути доступа.
3. Композиция пути доступа и функции — за основу берётся запрос, соответствующий пути доступа, но SQL-выражение в разделе SELECT заменяется на значение заданной функции от этого выражения.
4. Применение функции фильтрации к пути доступа — выражение в разделе WHERE заменяется конъюнкцией исходного выражения (либо 1, если его не было) и выражения, соответствующего функции фильтрации.

Опишем, какие конструкции языка SQL соответствуют различным вариантам построения вычисляемых выражений.

1. Константа записывается тривиальным образом.

2. Применение агрегатной функции к пути доступа — в запросе для пути доступа выражение в разделе `SELECT` заменяется на значение агрегатной функции от этого выражения. Приведённые в предыдущем разделе примеры агрегатных функций являются встроенными функциями языка SQL: `COUNT`, `SUM`, `MAX`, `MIN`.
3. Бинарные операторы либо непосредственно соответствуют операторам языка SQL, либо заменяются на встроенные функции от двух аргументов (`max` заменяется на `GREATEST`, `min` — на `LEAST`).
4. Для проверки вхождения вычислимого выражения в путь доступа используется оператор `IN`.

При необходимости, для приведения булева выражения к численному типу используется конструкция «`CASE WHEN  $f$  THEN 1 ELSE 0 END`».

Вместо генерации непосредственно текста SQL-запроса, можно использовать промежуточное представление запроса в виде абстрактного синтаксического дерева (abstract syntax tree). В этом случае операции с путями доступа и вычислимыми выражениями сводятся к объединению деревьев или замене части дерева. Частным случаем такого подхода является использование модулей для различных языков программирования, позволяющих представлять синтаксическое дерево запроса в виде конструкций этих языков. Такие модули, например, используются в инструментах объектно-реляционного отображения (ORM). Предлагаемая автором программная реализация использует механизмы генерации запросов из состава инструментального средства Django.

В листинге 1 приведён пример реального SQL-запроса, соответствующего примеру правила из подраздела 4.1.2. В таблице `ARTICLE` хранится информация о статьях, в `AUTHORA` — об авторах статей, в `JOURNALRANG` — об импакт-факторах журналов.


```

select (select JOURNALRANG.F_JOURNALRANG_VAL from JOURNALRANG
       where JOURNALRANG.F_JOURNAL_ID = ARTICLE.F_JOURNAL_ID
       and JOURNALRANG.F_JOURNALRANG_YEAR = 2020)
/ (select count(AUTHORA.F_AUTHORA_ID) from AUTHORA
   where AUTHORA.F_ARTICLE_ID = ARTICLE.F_ARTICLE_ID)
from ARTICLE
where ARTICLE.F_ARTICLE_NUMPAGES >= 5

```

Листинг 1. Пример SQL-запроса, соответствующего правилу

4.2.1. Проверка применимости ограничивающих выражений к заданной вершине

В случае, если вычисляемое выражение содержит несколько ограничивающих выражений, объединённых логическими операциями (конъюнкциями, дизъюнкциями), то возникает побочная задача. Её суть в том, чтобы по заданной вершине определить, будет ли она учтена алгоритмом, а если не будет, то какие из ограничивающих выражений являются причиной (то есть принимают значение 0). Данная задача вытекает из пункта 4 функциональных требований, приведённых выше в разделе 1.2. Наличие в информационно-аналитической системе такого сервиса позволит пользователю понять, почему конкретный результат деятельности получает тот или иной коэффициент значимости, и можно ли его улучшить, внося недостающие данные.

Для решения сформулированной вспомогательной задачи запрос можно изменить следующим образом.

- Каждое из ограничивающих выражений помещается в раздел SELECT как отдельный выбираемый столбец. В случае, если ограничивающее выражение представляло собой численный оператор сравнения, в качестве столбца выбирается неконстантный операнд.
- Присоединения таблиц из обязательных (JOIN) становятся опциональными

(LEFT JOIN).

- К запросу добавляется фильтр на идентификатор проверяемой вершины.

Таким образом, если исходный запрос имел следующую структуру:

```
select ID, ... from TABLE
join JOINED_TABLE on ...
where JOINED_TABLE.A and not B and C > C0
```

то новый запрос будет выглядеть следующим образом:

```
select JOINED_TABLE.A, B, C from TABLE
left join JOINED_TABLE on ...
where ID = ...
```

Структуру правила можно расширить, добавив туда текстовое описание каждого ограничивающего выражения либо действие, которое необходимо выполнить, чтобы выражение стало принимать значение 1 (например: «Объём статьи должен быть не менее 5 страниц»). В таком случае пользователю системы для каждого результата его деятельности может быть показан список правил, определённых для типа данного результата, и для каждого правила даны значения входящих в него ограничивающих выражений и их описания. Пример такого пользовательского интерфейса приведён на рис. 4.2.

4.2.2. Представление наборов правил в структурированном виде

Покажем, что вычисляемые выражения и сами правила могут быть представлены при помощи графовой модели данных, то есть в виде набора вершин и триплетов.

Сначала добавим в множество N следующие вспомогательные предикаты: *argument*, *left*, *right*, *path*, *expression*. Далее, добавим в N типы, соответствующие простым функциям, поддержку которых требуется обеспечить: это функции

Правило	Найденные проблемы
Из списка ВАК Статьи → Журналы → В России → Из списка ВАК	• Журнал не входит в «Список ВАК» • Значение параметра «Импакт-фактор РИНЦ» не установлено • Значение параметра «Число соавторов» равно 3, что больше верхней границы 2.0 • Результат должен подтвердить ответственный
Международные Статьи → Журналы → Международные	• Журнал не входит в «Scopus», «PubMed», «Science Citation Index», «Science Citation Index Expanded» или «Emerging Sources Citation Index» • Статья не имеет аффилиации организации • Значение параметра «Импакт-фактор Web of Science» не установлено • Результат должен подтвердить ответственный

Рис. 4.2. Пользовательский интерфейс для проверки соответствия результата деятельности набору правил

$L \rightarrow L$, агрегатные функции, бинарные операторы. Вершины, имеющие тип одного из операторов, будут иметь исходящие рёбра с предикатами *left* и *right*, а вершины с типами остальных функций — исходящее ребро с предикатом *argument*. Кроме того, добавим в N типы, соответствующие механизмам построения путей доступа и вычисляемых выражений: *Composition* (механизмы 2, 3 для путей), *Filter* (механизм 4 для путей) и *SetMembershipCheck* (механизм 4 для вычисляемых выражений). Вершины с типом *Composition* и бинарные операторы будут иметь исходящие рёбра с предикатами *left* и *right*, с оставшимися двумя типами — исходящие рёбра с предикатами *path* и *expression*. Благодаря этим предикатам и типам можно описать вычисляемые выражения. Корневым элементом является вершина с типом, соответствующим механизму построения, которая соединена рёбрами указанных предикатов с подграфами, соответствующими частям выражения. Предикаты, являющиеся основными единицами для построения путей доступа (механизм 1), уже есть в множестве N .

Для задания наборов правил осталось добавить два типа: для отдельных правил и для наборов, а также предикаты: *type* (при помощи которого и уже введённого предиката *expression* для правила будут указываться его тип и вычисляемое выражение), и *rule* (при помощи которого для набора правил будут указываться отдельные правила). Дополнительные атрибуты набора правил могут быть указаны при помощи предиката *expression* для вершины набора.

Таким образом, наборы правил могут быть представлены в рамках модели RDF. Для хранения и передачи данных можно использовать различные форматы: формат RDF/XML, определённый спецификацией RDF [50], или альтернативные форматы, такие как Turtle [52], JSON [53] и синтаксис в функциональном стиле [54].

В листинге 2 приведён пример представления правила, описанного выше в подразделе 4.1.2, в формате JSON. Пространство имён `pmodel`: указывает на классы и предикаты, связанные с построением вычислимых выражений и наборов правил, а пространство имён `cris`: — на классы и предикаты, описывающие структуру CRIS-системы.

```
{
  "@type": "pmodel:Rule",
  "pmodel:type": "cris:JournalArticle",
  "pmodel:expression": {
    "@type": "pmodel:Multiply",
    "pmodel:left": {
      "@type": "pmodel:Divide",
      "pmodel:left": {
        "@type": "pmodel:Filter",
        "pmodel:path": {
          "@type": "pmodel:Composition",
          "pmodel:left": "cris:publishedInJournal",
          "pmodel:right": "cris:hasImpactFactor"
        },
        "pmodel:expression": {
          "@type": "pmodel:Equal",
          "pmodel:left": "cris:publishedInYear",
          "pmodel:right": 2020
        }
      },
      "pmodel:right": {
        "@type": "pmodel:CountAggregate",
```

```

        "pmodel:argument": "cris:hasAuthor"
    },
    "pmodel:right": {
        "@type": "pmodel:GreaterOrEqual",
        "pmodel:left": "cris:numberOfPages",
        "pmodel:right": 5
    }
}

```

Листинг 2. Пример представления правила в формате JSON

4.3. Выводы

В настоящей главе введены понятия путей доступа (которые являются обобщением цепочек предикатов) и вычислимых выражений, позволяющих получать различные численные показатели для вершин на основе значений предикатов для самой вершины и вершин, связанных с ней. Даны рекурсивные определения этих двух понятий. Приведены примеры использования вычислимых выражений для получения наукометрических показателей результатов деятельности. Описаны возможности и приведены примеры построения вычислимых выражений, применимых в рамках задачи поиска экспертов. Описаны правила, задающие тип и вычисляемое выражение для этого типа, и наборы правил.

Во втором разделе главы описано, как вычисляемые выражения отображаются в язык запросов SQL, используемый в реляционных системах управления базами данных. Описана вспомогательная задача — сопоставление вершине списка правил с указанием выполненных и невыполненных для неё ограничений, устанавливаемых каждым правилом, и метод решения такой задачи. Таким образом, представленные модели и алгоритмы позволяют реализовать функциональность, описанную в пунктах 3, 4 и 5 функциональных требований (см. раздел 1.2). Опи-

саны возможные форматы хранения наборов правил в информационно-аналитической системе.

Результаты данной главы имеют собственную значимость и вне задачи поиска экспертов. Наборы правил могут быть использованы в CRIS-системах для автоматизированного формирования различных отчётов, рейтингов, статистики, а также для определения политик доступа к данным в моделях логического разграничения доступа.

Глава 5

Алгоритм ранжирования при поиске экспертов

Как отмечалось во Введении, эксперт — это специалист, активно и эффективно работающий в отдельной предметной области, что должно подтверждаться результатами его деятельности. Следовательно, именно по таким результатам и следует организовать поиск экспертов.

В рамках описанной ранее модели поисковым запросом является тематический портрет. Однако, поиск можно также осуществить, если входные данные представлены в следующем далее виде.

- Набор ключевых слов или рубрик (токенов) без весовых коэффициентов. В этом случае тематический портрет можно построить, либо распределив весовые коэффициенты равномерно, либо дав больший коэффициент токенам, которые в списке идут раньше. Например, в наборе из n токенов i -му можно присвоить коэффициент $\frac{1}{n} + \frac{n-2i+1}{n-1} \cdot \varepsilon$, где ε — коэффициент, показывающий, насколько значим порядок токенов ($0 \leq \varepsilon \leq \frac{1}{n}$).
- Вершина информационно-аналитической системы (результат деятельности). Тематический портрет вершины строится по алгоритму, описанному в главе 3. При этом сама вершина может быть использована для выявления потенциального конфликта интересов при помощи алгоритма, который будет описан ниже.

5.1. Описание шагов алгоритма

Алгоритм состоит из нескольких основных шагов. Сначала выполняется поиск результатов деятельности по заданной предметной области. Далее для каждой пары (результат, автор результата) вычисляется ранг, учитывающий коэффициент схожести с запросом и коэффициент значимости. Заметим, что коэффициент

значимости может зависеть от вклада конкретного автора, если их несколько. Наконец, для каждого из авторов найденных результатов вычисляется его ранг, и составляется список потенциальных экспертов.

Опишем каждый шаг такого алгоритма подробнее.

Поиск результатов деятельности по заданной предметной области. Среди всех результатов деятельности выбираются те, тематический портрет которых имеет ненулевой коэффициент схожести с тематическим портретом запроса. Чтобы исключить перебор всех результатов деятельности, можно рассматривать только те результаты, в портретах которых есть токены, имеющие общих соседей с токенами запроса в портретах из обучающей выборки. К их числу относятся такие токены, для которых в графе совместной встречаемости, описанном в главе 2, есть путь из двух рёбер до токенов из запроса. Чтобы уменьшить количество вычислений, тематические портреты для тех результатов деятельности, для которых они заданы неявно (с использованием алгоритмов главы 3), необходимо построить заранее.

Для каждого результата деятельности составляется множество его авторов — прообраз соответствующей ему вершины при предикате «является автором» (будем использовать обозначение *authorOf*). Списком потенциальных экспертов становится объединение таких множеств для всех результатов, имеющих ненулевой коэффициент схожести с запросом.

Вычисление рангов результатов. Ранги, которые результаты деятельности получают при поиске, должны учитывать как их коэффициенты релевантности, то есть схожести с поисковым запросом, так и коэффициенты значимости. Естественным методом учёта обоих показателей в равной степени является взятие произведения двух коэффициентов. В дальнейшем в формулах ранжирования будет использоваться именно произведение, хотя его можно заменить на другую функцию от двух переменных, монотонно возрастающую по каждой из них.

Коэффициенты значимости рассчитываются на основе динамически задаваемых наборов правил по алгоритму, описанному в разделе 4.1.

Как было описано в подразделе 4.1.2, в CRIS-системах информация об авторстве результата деятельности может быть выражена либо предикатом *authorOf*, либо при помощи промежуточной вершины для авторства. В зависимости от этого, вычисляемые выражения правил принимают на вход либо вершину, соответствующую результату, либо такую промежуточную вершину. Пусть для промежуточных вершин *author* — предикат, являющийся ссылкой на автора, *result* — ссылкой на результат.

Пусть $(\tau_1, f_1), \dots, (\tau_k, f_k)$ — набор правил, f_0 — общее для набора ограничивающее выражение (см. подраздел 4.1.3). Сначала для каждого правила (τ_i, f_i) введём следующую вспомогательную функцию, определённую для всех вершин информационно-аналитической системы:

$$f'_i(obj) = \begin{cases} f_i(obj) \cdot f_0(obj) & \text{если } obj \in \tau_i, \\ 0 & \text{если } obj \notin \tau_i. \end{cases}$$

Пусть q — поисковый запрос, $subj$ — потенциальный эксперт, obj — результат его деятельности. В случае, если для этого результата нет вершины авторства, определим его ранг как

$$\lambda(obj, subj) = s(p(obj), q) \cdot \max_{1 \leq i \leq k} f'_i(obj). \quad (5.1)$$

Рассмотрим теперь случай, когда используются вершины авторства. Сначала определим множество вершин авторства, соответствующих результату $subj$ и автору $subj$: $N_A(obj, subj) = result^{-1}(obj) \cap author^{-1}(subj)$. Ранг определим следующим образом:

$$\lambda(obj, subj) = s(p(obj), q) \cdot \max_{\substack{obj_A \in N_A(obj, subj) \\ 1 \leq i \leq k}} f'_i(obj_A). \quad (5.2)$$

Вычисление рангов экспертов. Рангом каждого потенциального эксперта будем называть следующее выражение:

$$\lambda(subj) = \sum_{obj \in authorOf(subj)} \lambda(obj, subj). \quad (5.3)$$

Построенный список потенциальных экспертов ранжируется по убыванию величины $\lambda(subj)$. Программную реализацию можно настроить таким образом, чтобы число экспертов в поисковой выдаче было ограничено сверху, либо чтобы не показывались кандидаты, для которых $\lambda(subj)$ меньше некоторого заданного значения.

Оптимизация вычисления $\lambda(subj)$ за счёт объединения правил. В некоторых случаях количество операций вычисления вычисляемых выражений можно уменьшить. В информационно-аналитических системах с реляционной СУБД таким образом будет уменьшено количество запросов к базе данных (см. раздел 1.3).

Определение 15. Два правила, определённые для одного и того же типа τ и имеющие вычисляемые выражения f_1 и f_2 , будем называть *дизъюнктными*, если $\forall obj \in \tau \ f_1(obj) \cdot f_2(obj) = 0$.

В частности, дизъюнктными являются такие правила, для которых вычисляемое выражение одного из них содержит ограничивающее выражение, отрицание которого содержится в вычисляемом выражении другого:

$$\begin{cases} f_1(obj) = f_0(obj) \cdot f'_1(obj), \\ f_2(obj) = (1 - f_0(obj)) \cdot f'_2(obj), \end{cases}$$

где $f_0(obj) \in \{0, 1\}$.

Утверждение 5. Значение $\lambda(obj, subj)$ останется неизменным, если два дизъюнктных правила с вычисляемыми выражениями f_1 и f_2 заменить на одно правило с вычисляемым выражением $f : f(obj) = f_1(obj) + f_2(obj)$.

Доказательство. Для каждой пары $(obj, subj)$ произведение значений вычисляемых выражений двух исходных правил равно 0, следовательно хотя бы одно из этих значений равно 0. Тогда значение для оставшегося правила будет совпадать со значением вычислимого выражения нового правила. Следовательно, значения максимума в формулах (5.1) и (5.2) не меняются при такой замене. $\square$

Такой подход позволяет объединять не только два, но и несколько дизъюнктивных правил. Всем объединённым правилам будет соответствовать один запрос к СУБД.

На практике объединение правил может быть использовано, если требуется разбить результаты деятельности какого-то типа на мелкие группы. Примером такого разбиения является деление статей в научных журналах, входящих в международные индексирующие системы, на группы в зависимости от квартиля журнала: Q1, Q2, Q3 или Q4. Набор правил может включать в себя четыре правила, соответствующие этим квартилям (например, с разными начальными коэффициентами). Но поскольку эти правила будут являться дизъюнктивными, то их можно объединить в общее правило.

5.2. Выявление потенциального конфликта интересов

Конфликтом интересов является ситуация, при которой поиск экспертов производится для экспертизы конкретного набора данных, информация о котором есть в информационно-аналитической системе, и при этом результат поиска содержит кандидатуры экспертов, связанных с этими данными напрямую или косвенно. Например, таким набором данных может быть заявка на экспертизу темы научно-исследовательской работы, а кандидатами на экспертизу — сами авторы заявки, их текущие либо бывшие коллеги, авторы совместных публикаций с заявителями. Такие кандидаты могут попасть в поисковую выдачу естественным образом: поисковым запросом является тематический портрет, связанный с данными, которые подлежат экспертизе. Однако при этом у заявителей или их соавторов могут

иметься публикации и другие результаты деятельности в этом же тематическом направлении, которые будут включены в результат поиска экспертов благодаря высокому коэффициенту схожести с тематикой заявки.

Вся необходимая информация для выявления конфликта интересов может быть получена из графа информационно-аналитической системы. Данные, которые подлежат экспертизе, соответствуют некоторому подграфу, который может включать вершины, соответствующие этапам, участию авторов и другие. Среди этих вершин имеется главная, соответствующая непосредственно рассматриваемым данным. Зафиксируем её и назовём obj_0 .

Заметим, что вершины, соответствующие непосредственным авторам заявки, связаны с obj_0 путём в графе (возможно, помимо двух конечных вершин проходящим через некоторые вершины описанного подграфа). Таким образом, эксперты, имеющие потенциальный конфликт интересов, также соединены с obj_0 путём, состоящим из определённых предикатов. Следовательно, можно определить один или несколько путей доступа, сопоставляющих вершине obj_0 вершины, соответствующие экспертам с конфликтом интересов. Например, путь доступа, сопоставляющий НИР множество соавторов её участников по публикациям, является композицией: пути, сопоставляющего НИР её участников, предиката $authorOf$ и предиката $authorOf^{-1}$.

Имеет смысл предоставить возможность не просто составить список таких путей доступа, но и сопоставить каждому пути некоторый коэффициент от 0 до 1. Коэффициент, равный 0, будет означать, что эксперта, соответствующего пути доступа, необходимо полностью исключить из выдачи. Если значение больше 0, то это будет означать, что ранг эксперта $\lambda(subj)$ необходимо умножить на данное значение.

С учётом изложенного выше, пусть задано множество пар $\{(g_1, \rho_1), \dots, (g_m, \rho_m)\}$, где g_i — пути доступа, сопоставляющие вершине obj_0 некоторое множество вершин, соответствующих экспертам, имеющим потенциальный конфликт интересов, $0 \leq \rho_i \leq 1$. Для каждой такой пары определим

функцию

$$\mu_i(subj) = \begin{cases} 1 & \text{если } subj \notin g_i(obj_0), \\ \rho_i & \text{если } subj \in g_i(obj_0). \end{cases}$$

Тогда в качестве финального ранга для эксперта $subj$ можно взять, например, выражение

$$\lambda'(subj) = \lambda(subj) \cdot \min_{1 \leq i \leq m} \mu_i(subj),$$

либо

$$\lambda''(subj) = \lambda(subj) \cdot \prod_{1 \leq i \leq m} \mu_i(subj).$$

Помимо понижения ранга таких потенциальных экспертов, они могут быть снабжены в поисковой выдаче специальной пометкой, сообщающей о необходимости проверить наличие конфликта интересов.

5.3. Описание программной реализации алгоритма поиска экспертов

Сервис поиска экспертов реализован в виде модуля информационно-аналитической системы «ИСТИНА». Программный код модуля написан на языке программирования Python (3.x) с использованием инструментального средства Django (в настоящий момент в ИАС «ИСТИНА» используется Django версии 2.2). Данные хранятся в СУБД PostgreSQL. Для построения пользовательского интерфейса используется набор инструментов для вёрстки Bootstrap.

При поиске учитываются следующие типы результатов деятельности: статьи в журналах и сборниках, участие в научно-исследовательских работах, защищённые кандидатские и докторские диссертации.

Пользовательский интерфейс позволяет ввести одно или несколько ключевых слов в строку поиска. Для разделения ключевых слов можно использовать запятую или клавишу Enter.

Поиск экспертов по ключевым словам и рубрикам

Введите ключевые слова через запятую:

× нейросети

× машинное обучение

▶ Выбрать рубрики ГРНТИ

▼ Выбрать рубрики ОЭСР

▶ ☐ Естественные и точные науки

▶ ☒ Техника и технологии

▶ ☐ Гуманитарные науки

▶ ☐ Медицинские науки и общественное здравоохранение

▶ ☐ Сельскохозяйственные науки

▶ ☐ Социальные науки

Учитывать результаты с

2000

↓

↑

года

☒ Учитывать переводы ключевых слов

▶ Настройки весовых коэффициентов результатов деятельности

Поиск

Рис. 5.1. Пользовательский интерфейс для ввода ключевых слов и рубрик

Предоставляется также возможность выбрать рубрики из двух рубрикаторов: ГРНТИ и рубрикатора ОЭСР (Организации экономического сотрудничества и развития). Рубрики в каждом рубрикаторе представлены в виде дерева, при выборе нелистовой вершины автоматически выбираются все дочерние рубрики. Для показа дерева используется библиотека Fancytree. Пользовательский интерфейс для ввода ключевых слов и рубрик показан на рис. 5.1.

Для обеспечения учёта результатов как на русском, так и на английском языке, выполняется автоматизированный перевод русскоязычных ключевых слов на английский язык. Для этого используется имеющаяся в системе таблица переводов ключевых слов (TRANSLATE), а также программные интерфейсы (API) двух внешних сервисов: Abbyu Lingvo и Википедии. Для перевода через сервис Abbyu Lingvo используется метод API, возвращающий мини-карточку (краткий перевод слова / фразы). Для перевода через Википедию проверяется наличие русскоязычной статьи, названием которой является заданное ключевое слово, и если есть такая статья и ссылка на связанную статью в английской Википедии, не являющуюся страницей разрешения неоднозначности (disambiguation page), то название связанной статьи используется в качестве перевода. Переведённые ключевые сло-

ва добавляются в тематический портрет с весом, равным 0,8 от веса исходного ключевого слова.

Для каждого поддерживаемого типа результатов деятельности реализован класс, производный от общего класса BaseFinder и определяющий следующие параметры:

- класс с описанием таблицы и её полей в СУБД, в терминах объектно-реляционного отображения (ORM) Django — «модель»;
- в случае, если результаты данного типа составляют подмножество всех объектов данной модели — условие фильтрации (например, для результатов типа «статья в журнале» таким условием будет «F_JOURNAL_ID is not null», отделяющее их от статей в сборнике);
- список наборов предикатов, используемых для построения тематического портрета результата, и их весов;
- максимальное количество результатов данного типа, которое будет получено из базы данных;
- список возможных фильтров и показателей, и соответствующие им вычисляемые выражения, описанные в терминах Django (для каждого фильтра указывается также его текстовое описание);
- фрагмент формы, показываемой пользователю, в виде класса форм Django (форма используется для указания весового коэффициента типа и различных параметров, которые могут учитываться в вычисляемых выражениях, например, поле BooleanField, определяющее, следует ли делить на число авторов).

В листинге 3 приведены фрагменты исходного кода класса, описывающего тип диссертаций.

```

class DissertationFinder(BaseFinder):
    """Класс для поиска диссертаций."""
    model = 'dissertations.Dissertation'
    keywords_paths = [('keywords__word', 0.5)]
    rubrics_paths = [('specialty', 0.5)]
    form_prefix = 'dissertation'
    worker_attr = 'author'
    constraints = [
        {
            'expression': Q(status='defended'),
            'error_description': 'Диссертация не является защищённой'
        }, {
            'expression': Q(phd='Y'),
            'error_description': 'Диссертация не является кандидатской',
            'when': lambda form: form.cleaned_data['is_candidate']
        }, {
            'expression': Q(phd='N'),
            'error_description': 'Диссертация не является докторской',
            'when': lambda form: form.cleaned_data['is_doctor']
        }
    ]
    parameters = [
        'description': {
            'expression': F('title')
        },
        'year': {
            'expression': F('year')
        }
    ]

class FormFragment(Form):
    """Фрагмент формы для поиска диссертаций."""
    is_candidate = BooleanField(label='Кандидатская диссертация')
    is_doctor = BooleanField(label='Докторская диссертация')

```


В классе BaseFinder реализована общая логика для формирования QuerySet (структурированного запроса Django) с учётом всех фильтров и сортировки по весовому коэффициенту результата, а также логика по присваиванию всем найденным результатам дополнительных атрибутов, указывающих, по какому правилу и по каким токенам они нашлись. Эта логика может быть изменена или расширена в производных классах.

Метод класса, выдающий список результатов, соответствующих запросу, является генератором, то есть выдаёт результаты по одному, не используя дополнительную память для создания списка. В основном классе для поиска экспертов по каждому правилу из набора определяется класс Finder для соответствующего типа, с использованием метода этого класса выполняется поиск, для каждого результата определяются идентификаторы его авторов и заполняется структура, сопоставляющая ID автора список его результатов с весовыми коэффициентами. Далее из базы данных загружается информация об авторах, для каждого автора считается его итоговый ранг и начало списка авторов, отсортированного по убыванию ранга, становится списком потенциальных экспертов.

Козицын Александр Сергеевич (4,0903)

- МГУ имени М.В. Ломоносова, 404 Лаборатория автоматизации экспериментальных исследований (Научно-исследовательский институт механики) ведущий научный сотрудник с 2001 г.
- Кафедра вычислительной математики (Механико-математический факультет) доцент с 2009 г., по совместительству

- **Статья** Система управления информацией о результатах научно-исследовательской и педагогической деятельности ИСТИНА-Наука МГУ: состояние и перспективы Ломоносовские чтения. Тезисы докладов научной конференции. Секция механики. 15 – 26 апреля 2013, Москва, МГУ имени М.В. Ломоносова. – М.: Издательство Московского университета, 2013 (0,711, авторов: 5)
- **Статья** Методы и средства повышения качества данных в Интеллектуальной Системе Тематического Исследования НАучно-технической информации (ИСТИНА) Ломоносовские чтения. Тезисы докладов научной конференции. Секция механики. Апрель 2013, 2013 (0,696, авторов: 4)
- **Статья** Информационная система "ИСТИНА" как Big Data - инструментарий в области управления на основе анализа наукометрических данных 1, 2015 (0,608, авторов: 5)
- **Статья** Формирование и использование настраиваемых отчетов в системе ИСТИНА Ломоносовские чтения. Научная конференция. Секция механики. 14–23 апреля 2014 года. Тезисы докладов, 2014 (0,608, авторов: 3)
- **Статья** Система "ИСТИНА" для подготовки принятия решений на основе анализа наукометрической информации Научный сервис в сети Интернет: Труды XVII Всероссийской научной конференции, 2015 (0,608, авторов: 4)
- **НИР** Развитие и сопровождение системы подготовки принятия решений на основе анализа информации о результатах научно-исследовательской, педагогической и инновационной деятельности ИАС «Наука МГУ» («ИСТИНА») (0,669, участников: 71)

Рис. 5.2. Фрагмент выдачи поиска экспертов

В выдаче сервиса поиска экспертов включена следующая дополнительная информация о местах работы каждого эксперта: организация, подразделение, корневое подразделение, должность, дата начала работы и статус совместительства. Пример фрагмента поисковой выдачи (по запросу «истина»), соответствующий одному из авторов, показан на рис. 5.2.

Реализован поиск экспертов по токенам, связанным с научно-исследовательской работой. На странице НИР в блоке «Работа с НИР» появляется ссылка «Поиск экспертов». Участники НИР в результатах поиска выделяются красным цветом для обозначения наличия конфликта интересов.

В рамках разработки модуля написаны автоматические тесты, проверяющие его работоспособность для поиска результатов деятельности различных типов и учёт опций, переданных через форму. Тесты обеспечивают полное покрытие кода, что обеспечивает выполнение требований по корректности работы, приведённых выше в разделе 1.4. Для проверки работы учёта переводов вместо прямого обращения к внешним сервисам в тестах используется модуль `unittest.mock`, позволяющий подменить настоящий код для перевода тестовым кодом, возвращающим заданные в тесте значения. Помимо тестов, код проверяется статическим анализатором `pylint` на наличие распространённых типов ошибок (требование по сопровождаемости, см. раздел 1.6). Исходный код хранится на сервере, работающем под управлением платформы GitLab. Тесты и анализатор запускаются на сервере при каждом изменении кода в рамках системы непрерывной интеграции.

5.3.1. Исследование производительности программной реализации

Исследование производительности производилось с использованием запущенного локально сервера разработки Django и удалённой базы данных, работающей под управлением СУБД PostgreSQL. Для измерения времени работы алгоритма и подсчёта SQL-запросов использовалось инструментальное средство Django Debug Toolbar.

Каждое измерение было проведено три раза, с разными наборами ключевых слов и/или рубрик. Наборы были подобраны из разных областей науки, так, чтобы i -ый набор состоял из i ключевых слов или рубрик. Наборы ключевых слов использовались следующие:

1. «ряды лорана»;
2. «ускорители заряженных частиц», «тяжёлые ионы»;
3. «полимеры», «высокомолекулярные соединения», «глобула».

Наборы рубрик использовались следующие. Рубрика 01.04.UY взята из рубрикатора ОЭСР, все остальные — из ГРНТИ:

1. 27.27.15 «Функции одного комплексного переменного»;
2. 29.05.49 «Гипотетические частицы и взаимодействия»,
29.05.81 «Методика и техника эксперимента в физике элементарных частиц»;
3. 31.25.01 «Химия высокомолекулярных соединений. Общие вопросы»,
31.25.15 «Структура и свойства природных и синтетических высокомолекулярных соединений»,
01.04.UY «Полимеры».

Вид поиска	Число запросов	Время 1	Время 2	Время 3
Ключевые слова без учёта переводов	16	16,7	19,7	21,1
Ключевые слова с учётом переводов	19	33,2	33,8	47,5
Рубрики	17	7,1	8,4	8,5
Ключевые слова + рубрики	25	20,1	20,3	23,4

Таблица 5.1. Измерения производительности работы поиска экспертов

Время (в секундах) загрузки страницы с результатами поиска для каждого набора в разных конфигурациях указано в таблице 5.1. Таблица показывает,

что быстрее всего выполняется поиск по рубрикам, а поиск по ключевым словам работает примерно в 2,4 раза дольше. При этом при включённом учёте переводов ключевых слов более 10 секунд тратится на обращение к внешним сервисам для поиска переводов. Опыт разработки системы показывает, что время загрузки страницы будет в несколько раз меньше при использовании не удалённой СУБД, а расположенной в той же подсети, что и основной сервер. Именно такая конфигурация используется для рабочих серверов ИАС «ИСТИНА». Измерения показывают, что программная реализация удовлетворяет пункту 1 требований по производительности, приведённых выше в разделе 1.3.

В таблице указано также число выполненных в рамках загрузки страницы SQL-запросов. Число запросов в каждой строке постоянно и не зависит от количества ключевых слов или рубрик в запросе, что соответствует пункту 3 требований по производительности. При этом 6 запросов являются общими для всех страниц системы и не относятся напрямую к поиску. Наиболее долго выполняющимся запросом является вспомогательный запрос на получение информации о местах работы потенциальных экспертов.

5.4. Выводы

Перечислены возможные варианты данных, поступающих на вход алгоритму поиска экспертов. Подробно описаны все шаги алгоритма поиска экспертов. Приведены формулы для рангов результатов деятельности $\lambda(obj, subj)$ двух типов: не использующих промежуточную вершину авторства (5.1) и использующих её (5.2), и формула для рангов самих экспертов $\lambda(subj)$ (5.3). Предложен способ уменьшения количества операций вычисления вычисляемых выражений (Утверждение 5).

Описаны механизмы выявления и учёта конфликта интересов, имеющегося у потенциального эксперта относительно набора данных, являющихся входными для алгоритма поиска экспертов. Для этих целей предложено использование множества путей доступа, сопоставляющих исходной вершине множество вершин

экспертов с потенциальным конфликтом, и присвоенных этим путям коэффициентов от 0 до 1. Ранги экспертов, определённых данными путями доступа, либо становятся равны 0, либо понижаются за счёт умножения на коэффициенты путей.

Приведена структура созданной автором программной реализации алгоритмов и описан её пользовательский интерфейс. Исследована производительность программной реализации.

Заключение

Задача интеллектуального анализа больших данных с целью поиска экспертов востребована в разных сферах деятельности. В зависимости от сферы деятельности, цели поиска, специфики предметной области и от других факторов, к результатам поиска могут предъявляться различные требования. Общими принципами являются необходимость соответствия сферы деятельности потенциального эксперта поисковому запросу (релевантность) и значимость научных и практических результатов, характеризующих работу эксперта на данном направлении. В настоящей диссертации предложен подход, учитывающий оба этих фактора путём составления поисковой выдачи на основе отдельных результатов деятельности (научных публикаций, участия в научно-исследовательских и опытно-конструкторских работах, преподавательской деятельности и других типов), с учётом наукометрических и иных показателей этих результатов. При этом критерии отбора и фильтрации результатов могут задаваться динамически при помощи модели, разработанной для этого автором.

К основным результатам диссертации относятся следующие.

- В рамках модели тематических портретов, основанных на ключевых словах и рубриках, введены функции схожести отдельных слов и рубрик, а также описана основанная на них функция схожести самих портретов. Функция схожести ключевых слов и рубрик основана на ранее описанном в [44] подходе, учитывающем их совместные вхождения в портреты, при этом добавлен учёт весовых коэффициентов ключевых слов и рубрик, иерархических связей между рубриками, и формулы модифицированы таким образом, чтобы схожесть ключевого слова с самим собой была равна 1. Доказательно показано, что значения коэффициентов схожести не превосходят 1, и описаны случаи, когда это максимальное значение достигается (Теорема 1, Утверждения 1 и 2). Приведены оценки вычислительной сложности (Утверждение 3). Проведено сравнение разработанной метрики с другими показателями схо-

жести.

- С использованием графовой модели данных для описания информационно-аналитической системы разработан алгоритм, позволяющий при наличии заранее заданных наборов предикатов составить тематический портрет вершины по ключевым словам и рубрикам, находящимся в графе на небольшом расстоянии от исходной вершины и связанным с ней цепочками рёбер с заданными предикатами. Доказательно показано, что сумма весовых коэффициентов в построенных таким образом портретах не превосходит 1 (Теорема 2), и то же справедливо для суммарного вклада одного токена в построенные портреты (Теорема 3).
- Построена модель, позволяющая динамически определять правила для отбора результатов деятельности, которые будут учтены при поиске, и для сопоставления этим результатам коэффициентов значимости, учитывающих значения предикатов для вершин самих результатов и для вершин, связанных с ними путями заданного вида. Описан практический аспект: метод построения запросов к реляционным СУБД на основе таких правил.
- Описан алгоритм, учитывающий тематическую схожесть и коэффициенты значимости результатов для составления списка потенциальных экспертов и присвоения им рангов. Создана программная реализация этого алгоритма. Разработаны методы выявления потенциального конфликта интересов.

Результаты диссертации успешно внедрены в качестве отдельного модуля в информационно-аналитической системе «ИСТИНА», которая используется в МГУ имени М. В. Ломоносова и в ряде других научных организаций.

К перспективам развития темы диссертации можно отнести учёт морфологических характеристик отдельных слов, входящих в ключевые слова, и определение схожести с учётом данных о формах слов и родственных связях между ними. Другим перспективным направлением является учёт дополнительных данных, кото-

рые могут дать информацию о схожести предметных областей, например данных графа соавторства.

Автор благодарит своего научного руководителя — профессора, доктора физико-математических наук В. А. Васенина, а также доцента, кандидата физико-математических наук С. А. Афонаина за постановку задач и внимание к работе на всех этапах её выполнения.

Список литературы

1. *Афонин С. А., Козицын А. С., Шачнев Д. А.* Программные механизмы агрегации данных, основанные на онтологическом представлении структуры реляционной базы наукометрических данных // Программная инженерия. — Москва, 2016. — т. 7, № 9. — с. 408—413. — ISSN 2220-3397. — DOI: 10.17587/prin.7.408-413. — RSCI (импакт-фактор РИНЦ 2020: 0,459).
2. *Шачнев Д. А.* Searching for Activity Results and Experts in a Given Subject Area, Taking Results Significance into Account // Программная инженерия. — Moscow, 2021. — т. 12, № 5. — с. 260—266. — ISSN 2220-3397. — DOI: 10.17587/prin.12.260-266. — RSCI (импакт-фактор РИНЦ 2020: 0,459).
3. *Shachnev D., Karpenko D.* Using Subject Area Ontology for Automating Processes in Sphere of Scientific Investigation and Education // Programming and Computer Software. — Road Town, United Kingdom, 2018. — Vol. 44, no. 1. — P. 15–22. — ISSN 0361-7688. — DOI: 10.1134/S0361768818010061. — Web of Science (Impact Factor 2020: 0,936).
4. *Шачнев Д. А., Афонин С. А., Козицын А. С.* Использование онтологического представления структуры реляционной базы для агрегации наукометрических данных // Научный сервис в сети Интернет: труды XVIII Всероссийской научной конференции (19–24 сентября 2016 г., г. Новороссийск). — Москва : ИПМ им. М.В. Келдыша, 2016. — с. 58—63. — ISBN 978-5-98354-027-9. — DOI: 10.20948/abrau-2016-1.
5. *Шачнев Д. А.* Онтология предметной области для научных исследований и автоматизации учебного процесса: методы реализации и алгоритмы использования // Материалы Всероссийской конференции с международным участием «Знания-Онтологии-Теории» (ЗОНТ-2017). т. 2. — Новосибирск : ООО «Дигит Про», 2017. — с. 148—155.

6. *Shachnev D.* Web ontology editor: architecture and applications // 5th International Conference on Actual Problems of System and Software Engineering, APSSE 2017. Vol. 1989. — CEUR Workshop Proceedings (CEUR-WS.org), 2017. — P. 342–350.
7. Methods for Intelligent Data Analysis Based on Keywords and Implicit Relations: The Case of “ISTINA” Data Analysis System / D. Shachnev [et al.] // 2019 Actual Problems of Systems and Software Engineering (APSSE). — IEEE, 2019. — P. 157–161. — ISBN 978-1-7281-6061-0. — DOI: 10.1109/APSSE47353.2019.00027.
8. *Шачнев Д. А.* Программные механизмы автоматической генерации SQL-запросов в ИАС «ИСТИНА»: особенности и варианты использования // Ломоносовские чтения. Секция механики. Тезисы докладов. — Москва : Издательство Московского университета, 2018. — с. 196—197. — ISBN 978-5-19-011337-2.
9. Методы и средства тематического анализа данных в больших системах на основе ключевых слов и косвенных связей между ними / Д. А. Шачнев [и др.] // Ломоносовские чтения. Секция механики. Тезисы докладов. — Москва : Издательство Московского университета, 2019. — с. 51—51. — ISBN 978-5-19-011444-7.
10. *Шачнев Д. А.* Новая версия rmodel — генератора SQL-запросов, основанного на онтологическом представлении структуры базы данных информационной системы, с использованием Django ORM // Ломоносовские чтения. Секция механики. Тезисы докладов. — Москва : Издательство Московского университета, 2020. — с. 207—208. — ISBN 978-5-19-011565-9.
11. Национальная программа «Цифровая экономика Российской Федерации» : Паспорт национального проекта / Правительство Российской Федерации. — 2019. — URL: https://digital.ac.gov.ru/upload/iblock/219/NP_Cifrovaya_ekonomika.docx.

12. *Rossner M., Van Epps H., Hill E.* Show me the data // The Journal of cell biology. — 2007. — Vol. 179, no. 6. — P. 1091–1092. — ISSN 0021-9525. — DOI: 10.1083/jcb.200711140.
13. *Callaway E.* Beat it, impact factor! Publishing elite turns against controversial metric // Nature. — 2016. — Vol. 535, no. 7611. — P. 210–211. — ISSN 0028-0836. — DOI: 10.1038/nature.2016.20224.
14. The Leiden Manifesto for research metrics / D. Hicks [et al.] // Nature. — 2015. — Vol. 520. — P. 429–431. — ISSN 0028-0836. — DOI: 10.1038/520429a.
15. *Панкова Л. А., Пронина В. А., Крюков К. В.* Онтологические модели поиска экспертов в системах управления знаниями научных организаций // Проблемы управления. — Москва, 2011. — № 6. — с. 52—60. — ISSN 1819-3161. — URL: http://pu.mtas.ru/upload/pu_611.pdf.
16. *Russell-Rose T., Chamberlain J.* Searching for talent: The information retrieval challenges of recruitment professionals // Business Information Review. — 2016. — Vol. 33, no. 1. — P. 40–48. — ISSN 0266-3821. — DOI: 10.1177/0266382116631849.
17. *Albæk E.* The interaction between experts and journalists in news journalism // Journalism. — 2011. — Vol. 12, no. 3. — P. 335–348. — ISSN 1464-8849. — DOI: 10.1177/1464884910392851.
18. Сетевая экспертиза / Д. А. Губанов [и др.] ; под ред. Д. А. Новиков, А. Н. Райков. — Москва : Эгвес, 2010. — 168 с. — ISBN 978-5-91450-037-2. — URL: <http://www.mtas.ru/upload/library/networkexpertise.pdf>.
19. *Мельник П. Б.* Математическая постановка задачи формирования реестра экспертов // Инноватика и экспертиза. — Москва, 2014. — 2 (13). — с. 69—81. — ISSN 1996-2274. — URL: http://inno-exp.ru/archive/13/innov_13_2014_69-81.pdf.

20. *Maron M. E., Curry S., Thompson P.* An Inductive Search System: Theory, Design, and Implementation // IEEE Transactions on Systems, Man, and Cybernetics. — 1986. — Vol. 16, no. 1. — P. 21–28. — DOI: 10.1109/TSMC.1986.289278.
21. Expertise Identification Using Email Communications / C. S. Campbell [et al.] // Proceedings of the Twelfth International Conference on Information and Knowledge Management (New Orleans, LA, USA). — New York, NY, USA : Association for Computing Machinery, 2003. — P. 528–531. — (CIKM '03). — ISBN 978-1-58113-723-1. — DOI: 10.1145/956863.956965.
22. P@NOPTIC Expert: Searching for Experts not just for Documents / N. Craswell [et al.] // In Ausweb. — 2001. — P. 21–25.
23. *D'Amore R.* Expertise Community Detection // Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (Sheffield, United Kingdom). — New York, NY, USA : Association for Computing Machinery, 2004. — P. 498–499. — (SIGIR '04). — ISBN 978-1-58113-881-8. — DOI: 10.1145/1008992.1009089.
24. Combining RDF Vocabularies for Expert Finding / B. Aleman-Meza [et al.] // The Semantic Web: Research and Applications / ed. by E. Franconi, M. Kifer, W. May. — Berlin, Heidelberg : Springer, 2007. — P. 235–250. — ISBN 978-3-540-72667-8. — DOI: 10.1007/978-3-540-72667-8_18.
25. *Yimam-Seid D., Kobsa A.* Expert-finding systems for organizations: Problem and domain analysis and the DEMOIR approach // Journal of Organizational Computing and Electronic Commerce. — 2003. — Vol. 13, no. 1. — P. 1–24. — DOI: 10.1207/S15327744JOCE1301_1.
26. *Balog K., Azzopardi L., de Rijke M.* Formal Models for Expert Finding in Enterprise Corpora // Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (Seattle, Washington, USA). — New York, NY, USA : Association for Computing Machin-

- ery, 2006. — P. 43–50. — (SIGIR '06). — ISBN 978-1-59593-369-0. — DOI: 10.1145/1148170.1148181.
27. *Balog K., de Rijke M.* Determining Expert Profiles (with an Application to Expert Finding) // Proceedings of the 20th International Joint Conference on Artificial Intelligence (Hyderabad, India). — San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 2007. — P. 2657–2662. — (IJCAI '07). — URL: <https://dl.acm.org/doi/10.5555/1625275.1625703>.
 28. *Fang Y., Si L., Mathur A. P.* Discriminative Models of Integrating Document Evidence and Document-Candidate Associations for Expert Search // Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (Geneva, Switzerland). — New York, NY, USA : Association for Computing Machinery, 2010. — P. 683–690. — (SIGIR '10). — ISBN 978-1-4503-0153-4. — DOI: 10.1145/1835449.1835563.
 29. AMiner: Search and Mining of Academic Social Networks / H. Wan [et al.] // Data Intelligence. — 2019. — Vol. 1, no. 1. — P. 58–76. — DOI: 10.1162/dint_a_00006.
 30. An Overview of Microsoft Academic Service (MAS) and Applications / A. Sinha [et al.] // Proceedings of the 24th International Conference on World Wide Web (Florence, Italy). — New York, NY, USA : Association for Computing Machinery, 2015. — P. 243–246. — (WWW '15 Companion). — ISBN 978-1-4503-3473-0. — DOI: 10.1145/2740908.2742839.
 31. *Ganter B., Wille R.* Formal Concept Analysis : Mathematical Foundations. — Germany : Springer, 1999. — 284 p. — ISBN 978-3-540-62771-5. — DOI: 10.1007/978-3-642-59830-2.
 32. Identifying a group of subject experts using formal concept analysis / V. Boeva [et al.] // 2016 IEEE 8th International Conference on Intelligent Systems (IS) (Sofia,

- Bulgaria). — IEEE, 2016. — P. 464–469. — ISBN 978-1-5090-1354-8. — DOI: 10.1109/IS.2016.7737462.
33. Интеллектуальная система тематического исследования научно-технической информации («ИСТИНА») / С. А. Афонин [и др.] ; под ред. В. А. Садовничий. — Москва : Издательство Московского университета, 2014. — 262 с. — ISBN 978-5-19-011015-9.
 34. Садовничий В. А., Васенин В. А. Интеллектуальная система тематического исследования наукометрических данных: предпосылки создания и методология разработки. Часть 1 // Программная инженерия. — Москва, 2018. — т. 9, № 2. — с. 51—58. — ISSN 2220-3397. — DOI: 10.17587/prin.9.51-58.
 35. Automatic keyword extraction from individual documents / S. Rose [et al.] // Text Mining: Applications and Theory / ed. by M. W. Berry, J. Kogan. — Chichester, United Kingdom : Wiley, 2010. — Chap. 1. P. 3–20. — ISBN 978-0-470-74982-1. — DOI: 10.1002/9780470689646.ch1.
 36. *McTear M., Callejas Z., Griol D.* Spoken Language Understanding // The Conversational Interface. — Cham, Switzerland : Springer International Publishing, 2016. — P. 161–185. — ISBN 978-3-319-32965-9. — DOI: 10.1007/978-3-319-32967-3_8.
 37. *Erk K.* Vector Space Models of Word Meaning and Phrase Meaning: A Survey // Language and Linguistics Compass. — 2012. — Vol. 6, no. 10. — P. 635–653. — DOI: 10.1002/lnco.362.
 38. Efficient Estimation of Word Representations in Vector Space / T. Mikolov [et al.]. — 2013. — arXiv: 1301.3781 [cs.CL].
 39. Indexing by latent semantic analysis / S. Deerwester [et al.] // Journal of the American Society for Information Science. — 1990. — Vol. 41, no. 6. — P. 391–407. — DOI: 10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.

40. *Došilović F. K., Brčić M., Hlupić N.* Explainable artificial intelligence: A survey // 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). — IEEE, 2018. — P. 0210–0215. — ISBN 978-953-233-095-3. — DOI: 10.23919/MIPRO.2018.8400040.
41. *Biemann C.* Ontology Learning from Text: A Survey of Methods // LDV-Forum. — 2005. — Vol. 20, no. 2. — P. 75–93. — URL: https://jlc1.org/content/2-allissues/23-Heft2-2005/Chris_Biemann.pdf.
42. Keyword extraction: Issues and methods / N. Firoozeh [et al.] // Natural Language Engineering. — 2020. — Vol. 26, no. 3. — P. 259–291. — DOI: 10.1017/S1351324919000457.
43. *Wang X., McCallum A.* A Note on Topical N-grams : tech. rep. / University of Massachusetts Amherst, Department of Computer Science. — 2005.
44. *Лунев К. В.* Графовые методы определения семантической близости пары ключевых слов и их применения к задаче кластеризации ключевых слов // Программная инженерия. — Москва, 2018. — т. 9, № 6. — с. 262—271. — ISSN 2220-3397. — DOI: 10.17587/prin.9.262-271.
45. *Sahlgren M.* The distributional hypothesis // Italian Journal of Linguistics. — 2008. — Vol. 20. — P. 33–53. — ISSN 1120-2726. — URL: <http://www.italian-journal-linguistics.com/wp-content/uploads/Sahlgren.pdf>.
46. *Jones K. S.* A Statistical Interpretation of Term Specificity and Its Application in Retrieval // Journal of Documentation. — 1972. — Vol. 28, no. 1. — P. 11–21. — ISSN 0022-0418. — DOI: 10.1108/eb026526.
47. *Leacock C., Chodorow M.* Combining Local Context and WordNet Similarity for Word Sense Identification // WordNet: An Electronic Lexical Database / ed. by C. Fellbaum. — Cambridge : MIT Press, 1998. — Chap. 11. P. 265–283. — ISBN 978-0-262-06197-1. — DOI: 10.7551/mitpress/7287.003.0018.

48. Soft similarity and soft cosine measure: Similarity of features in vector space model / G. Sidorov [et al.] // *Computación y Sistemas*. — 2014. — Vol. 18, no. 3. — P. 491–504. — ISSN 1405-5546. — DOI: 10.13053/CyS-18-3-2043.
49. Evaluating Publication Similarity Measures / S. Bani-Ahmad [et al.] // *IEEE Data Eng. Bull.* — 2005. — Vol. 28, no. 4. — P. 21–28. — URL: <http://sites.computer.org/debull/A05dec/bani-ahmad.pdf>.
50. *Hayes P., Patel-Schneider P.* RDF 1.1 Semantics : W3C Recommendation / W3C. — 2014. — URL: <https://www.w3.org/TR/rdf11-mt/>.
51. A Direct Mapping of Relational Data to RDF : W3C Recommendation / M. Arenas [et al.] ; W3C. — 2012. — URL: <https://www.w3.org/TR/rdb-direct-mapping/>.
52. *Prud'hommeaux E., Carothers G.* RDF 1.1 Turtle : W3C Recommendation / W3C. — 2014. — URL: <https://www.w3.org/TR/turtle/>.
53. *Champin P.-A., Kellogg G., Longley D.* JSON-LD 1.1 : W3C Recommendation / W3C. — 2020. — URL: <https://www.w3.org/TR/json-ld11/>.
54. *Patel-Schneider P., Motik B., Parsia B.* OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax : W3C Recommendation / W3C. — 2012. — URL: https://www.w3.org/TR/owl2-syntax/#Functional-Style_Syntax.