МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
ДОНЕЦКОЙ НАРОДНОЙ РЕСПУБЛИКИ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

На правах рукописи
УДК 004.89:004.048:004.912

Андриевская Наталия Климовна

СОВЕРШЕНСТВОВАНИЕ МОДЕЛЕЙ И АЛГОРИТМОВ ОБРАБОТКИ ИНФОРМАЦИИ В СИСТЕМАХ ОРГАНИЗАЦИОННОГО СОПРОВОЖДЕНИЯ ДЕЯТЕЛЬНОСТИ НАУЧНО-ОБРАЗОВАТЕЛЬНЫХ УЧРЕЖДЕНИЙ

Специальность 05.13.01 – Системный анализ, управление и обработка информации (по отраслям) (технические науки)

Диссертация
на соискание ученой степени
кандидата технических наук

Научный руководитель:
кандидат технических наук, доцент
Секирин А.И.

Идентичность всех экземпляров
ПОДТВЕРЖДАЮ
Ученый секретарь диссертационного
совета Д 01.024.04
кандидат технических наук, доцент Т.В. Завадская

Донецк – 2021

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Актуальность темы исследования. Организационное сопровождение деятельности научно-образовательных учреждений характеризуется непрерывным ростом количества электронных документов и их общедоступности, в том числе и в среде Интернет. Слабая структурированность информационных фондов осложняет управление информацией и работу пользователей с ней, что напрямую относится к потокам информации, с которыми встречаются сотрудники вузов. В настоящее время кафедрами вузов накоплен большой объем знаний и информационных ресурсов (ИР) по различным курсам и результатам научно-методической работы. Однако отсутствие связанности ИР и унифицированного доступа к ним приводят к возникновению проблем поиска, учета и систематизации как существующих знаний, так и новых [1].

В связи с этим возникает необходимость создания современного интеллектуального инструмента, поддерживающего повседневную профессиональную деятельность преподавателя. Для решения этой задачи необходим переход на качественно новый уровень представления и обработки информации – семантический, что позволит учитывать смысл документов, извлекая из них важные для пользователя знания.

Таким образом, совершенствование моделей и алгоритмов обработки информации для реализации в системах организационного сопровождения деятельности научно-образовательных учреждений является актуальной научно-технической задачей, имеющей отраслевое значение.

Связь работы с научными программами, планами, темами.

Работа выполнена в соответствии с тематическим планом ГОУВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»: Н-17-12 «Разработка основ, методов и средств проектирования информационных управляющих систем»; Н-8-18 «Развитие научных основ, методов и средств проектирования информационных систем и технологий»; Н-2020-16 «Методы и средства построения информационных систем с использованием технологий

интеллектуального анализа данных», в которых соискатель являлся исполнителем (справка №06/4-328 от 28.09.2021 г.).

Степень разработанности темы исследования. Среди систем организационного сопровождения деятельности научно-образовательных учреждений в последние годы все чаще встречаются системы управления знаниями (СУЗ), которые явно ориентированы на эффективную работу с ИР и знаниями [2].

Эта тенденция обусловлена главным образом тем, что знания все больше становятся организационным активом, существование которого позволяет повторно использовать знания, избежать «испарения» знаний и поддерживать принятие решений в учреждении [3].

Перспективность данного направления подтверждается результатами исследований таких зарубежных учёных, как D. Ameller and X. Franch [4], M. Bhat [5], P. Kruchten [6], R. Capilla [7], I. Lytra, H. Tran, U. Zdun [8] и др., а также российских ученых: А.Ф. Тузовского [9], В.З. Ямпольского, В.А. Лапшина [10], А.Г. Олейника [11], Т.А. Гавриловой [12] и др.

Таким образом, можно констатировать, что важность проблематики разработки и совершенствования моделей и алгоритмов обработки информационных ресурсов научно-образовательных учреждений, построенных на основных принципах систем управления знаниями, осознается специалистами, занимающимися информационными технологиями и корпоративным управлением. Тем не менее, до сих пор не существует полного набора моделей и алгоритмов, позволяющих поддерживать работу с явным описанием семантики ИР вузов и других научно-образовательных учреждений, что определяет необходимость дальнейших исследований.

Цель и задачи исследования. Целью диссертационной работы является повышение эффективности системы управления ИР научно-образовательных учреждений за счет применения онтологического подхода, разработки новых и усовершенствования существующих моделей и алгоритмов поиска, хранения и классификации данных.

Для достижения цели поставлены и решены следующие задачи.

1. Провести системный анализ процессов с целью формализации исследуемого объекта и обосновать возможность использования онтологического подхода к построению системы управления ИР учреждения.

2. Разработать онтологическую модель объектов знаний – аппарат для описания семантики области профессиональной деятельности сотрудников научно-образовательных учреждений.

3. Усовершенствовать гибридную меру оценки семантической близости (СБ).

4. Модифицировать векторную модель представления текстов на базе известных подходов bag-of-words и bag-of-concepts, улучшив ее за счет использования онтологической модели предметной области.

5. Усовершенствовать модели и алгоритмы поиска, хранения и классификации данных на основе разработанной онтологической модели.

6. Разработать прототипы программных модулей, реализующие предложенные модели и алгоритмы в виде фреймворка, а также выполнить на базе фреймворка реализацию и тестирование программного модуля учета научной деятельности сотрудников учреждения.

Объект исследования – информационные процессы поиска и обработки ИР научно-образовательного учреждения.

Предмет исследования – модели и алгоритмы реализации информационных процессов и концепции информационного поиска на семантическом уровне с использованием онтологий в системах организационного сопровождения деятельности научно-образовательных учреждений.

Научная новизна полученных результатов заключается в следующем:

1. Впервые разработана онтологическая модель научно-образовательной деятельности сотрудников вуза.

2. Усовершенствована гибридная мера определения семантической близости на базе модифицированной N-мерной модели представления знаний RDF-графа, использование которой повысило качество поиска, выраженное F-мерой, на 10.7% по сравнению с мерой «косинусного сходства».

3. Получила дальнейшее развитие векторная модель представления текстов на базе известных подходов bag-of-words и bag-of-concepts, улучшенная за счет применения онтологии и тематической редукции векторного пространства, что позволило при уменьшении размерности пространства с 2250 терминов до 30 терминов повысить скорость выполнения тестируемых алгоритмов более чем на порядок при незначительном снижении меры семантической близости на 6.2%.

4. Усовершенствована модель классификации данных, основанная на применении гибридной меры определения СБ, что привело к повышению качества классификации, выраженного F-мерой, по сравнению с алгоритмами, использующими меру, вычисленную только по онтологии на 45.4%, «косинусную» меру – на 5.3% и «мягкую косинусную» меру – на 9.5%.

Теоретическая значимость работы.

Теоретическая значимость результатов исследований заключается в развитии моделей и алгоритмов обработки ИР научно-образовательных учреждений и переходу к онтологическому и семантическому моделированию.

Практическая значимость работы.

1. На основе проведенных ранее исследований, разработанных моделей и алгоритмов выполнена программная реализация системы управления ИР в рамках кафедры вуза.

2. Использование документированной прикладной онтологии дает возможность разработчикам систем повторно использовать и развивать данную онтологию, а различным ИС – интегрировать данные и обеспечивать обмен данными на основе онтологии.

3. Предложенные подходы и математические модели, разработанные программные модули фреймворка могут быть применены при создании различных СУЗ, а также систем обработки ИР любых учреждений.

4. Разработанные в ходе выполнения диссертационной работы модели и методы использованы в учебном процессе кафедры автоматизированных систем управления ГОУВПО «ДОННТУ» при выполнении курсовых работ и выпускных квалификационных работ студентов.

5. Разработанный программный модуль «Наука» успешно прошел тестирование в ГУ «Автоматгормаш им. В.А. Антипова» (г. Донецк) в условиях отдела систем управления.

Практическая реализация результатов работы подтверждается справкой №30-12/214 от 10.12.2021 г. о внедрении в учебный процесс ГОУВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ», справкой № 12-319 от 9.06.2021 г. о внедрении в ГУ «Автоматгормаш им. В.А. Антипова» (г. Донецк).

Методология и методы исследования. Для решения поставленных задач использованы методы исследования, основанные на методах системного анализа, онтологического инжиниринга, семантического моделирования, теории графов и множеств, теории экспертных оценок, а также теории нечеткой логики и генетических алгоритмов.

Научные положения, выносимые на защиту.

1. Доказано, что применение усовершенствованной гибридной меры определения семантической близости, использующей модифицированную N-мерную модель представления знаний RDF-графа и генетический алгоритм определения весовых коэффициентов базовых мер, позволило повысить точность определения сходства концептов и улучшить качество поиска, выраженного F-мерой, на 10.7% по сравнению с мерой «косинусного сходства».

2. Определено, что использование техники снижения размерности векторного пространства по тематическим векторам предметной онтологии для векторной модели представления текста при размере онтологии, равном 2250 терминов, и длине контекстного вектора в 30 элементов, приводит к снижению вычислительной сложности тестируемых алгоритмов в десятки раз при незначительном снижении коэффициента СБ с 0.83 до 0.778.

3. Установлено, что при классификации текстовых ИР результаты, полученные с помощью усовершенствованной модели классификации, алгоритм которой использует гибридную меру определения СБ, более точны, чем

вычисленные по онтологии, по «косинусной» и «мягкой косинусной» мерам на 45.4%, 5.3% и 9.5% соответственно.

Степень достоверности результатов. Обоснованность и достоверность научных положений, выводов и практических результатов подтверждается полнотой анализа теоретических и практических исследований, разработкой и тестированием программного модуля системы, о чем свидетельствуют справки о внедрении, выполненными публикациями и положительной оценкой на научно-технических конференциях.

По направлению исследований, содержанию научных положений и выводов, существу полученных результатов диссертационная работа соответствует паспорту специальности 05.13.01 – Системный анализ, управление и обработка информации (по отраслям) (технические науки) в частности: п.4 «Разработка методов и алгоритмов решения задач системного анализа, оптимизации, управления, принятия решений и обработки информации»; п.5 «Разработка специального математического и алгоритмического обеспечения систем анализа, оптимизации, управления, принятия решений и обработки информации»; п.8 – «Теоретико-множественный и теоретико-информационный анализ сложных систем».

Апробация. Основные положения диссертационной работы апробированы на научно-технических конференциях: VII Международной научно-технической конференции студентов, аспирантов и молодых ученых «Информатика и компьютерные технологии» г. Донецк, 22-23 ноября 2011 г.; XI Международной научно-технической конференции «Информатика, управляющие системы, математическое и компьютерное моделирование» (ИУСМКМ-2020), г. Донецк, 27-28 мая 2020 г.; 16-й Юбилейной Международной молодежной научно-технической конференции «Современные проблемы радиоэлектроники и телекоммуникаций» (РТ-2020), г. Севастополь, с 12 по 16 октября 2020 г.; III Международной научно-практической конференции «Программная инженерия: методы и технологии разработки информационно-вычислительных систем» (ПИИВС-2020), г. Донецк, 25–26 ноября 2020 г.

Личный вклад соискателя. Все результаты и положения, составляющие основное содержание диссертации, вынесенные на защиту, получены автором самостоятельно. Личный вклад соискателя заключается в обосновании идеи работы и ее реализации, цели и задач работы, в выборе методов и направлений исследований, выполнении теоретических, аналитических и экспериментальных исследований, разработке положений и рекомендаций по использованию результатов работы.

Публикации. По теме диссертационной работы всего было опубликовано 10 научных работ. Из них 6 работ в изданиях, рекомендованных ВАК ДНР, 4 по материалам научно-технических конференций.

Структура и объем диссертации. Диссертационная работа содержит 199 страниц машинописного текста и состоит из введения, четырех разделов, заключения, списка литературы из 141 источника и 4 приложений. Основной текст диссертации иллюстрируется 82 рисунками и содержит 47 таблиц.

РАЗДЕЛ 1
АНАЛИЗ СОСТОЯНИЯ ВОПРОСА, ЦЕЛЬ И ЗАДАЧИ ИССЛЕДОВАНИЯ

1.1 Описание особенностей объекта исследований

Современный мир характеризуется резким ростом объёмов информационных потоков. Предпосылками такого роста является активный переход к электронному ведению документации, повсеместное развитие современных информационных технологий, относительная «дешевизна» и доступность Интернет.

Вследствие этого появилась проблема «информационного насыщения», когда лавинообразные темпы роста количества электронных ресурсов перекрыли информационные возможности человека, находящиеся и так на пределе. Возник конфликт между огромным количеством информационных ресурсов и способностью человека усвоить и обработать информацию. Параллельно возникла проблема «информационного шума», когда существование большого количества избыточной информации затрудняет поиск полезной и релевантной информации, распределенной по всему информационному пространству.

Определенным образом это относится к потокам информации, с которыми встречаются работники различных научно-образовательных учреждений, в том числе и сотрудники вузов.

Согласно Закону РФ «Об информации, информационных технологиях и о защите информации» информация определяется следующим образом: информация – сведения (сообщения, данные) независимо от формы их представления [13].

Данными называется информация, представленная в формализованном виде, пригодном для передачи, интерпретации или обработки с участием человека или автоматическими средствами [14].

Документом называется зафиксированная на материальном носителе информация с реквизитами, позволяющими ее идентифицировать, и предназначенная для ее хранения и передачи.

Знаниями являются выявленные в результате практической деятельности и профессионального опыта закономерности предметной области, а также информация, некоторым образом полученная и упорядоченная. Знания представляют собой:

- знания в памяти человека как результат обучения, воспитания, мышления;
- знания на материальных носителях (инструкции, учебники, методические пособия);
- знания, описанные на языках представления знаний;
- база знаний на машинных носителях информации.

Информационный ресурс – это индивидуальные и коллективные экспертные знания, отдельные документы, отдельные массивы документов, а также документы и их массивы, составляющие базы и банки данных, базы знаний, библиотеки, архивы, фонды, информационные системы и другие системы в определенной предметной тематической области, которые удовлетворяют функциональным потребностям и запросам потребителей информации. Информационные ресурсы представляют собой лишь одну из возможных форм представления знаний.

Информационными ресурсами обладают отдельные люди, коллективы людей, организации, территориальные и национальные образования, города, регионы, страны и в целом весь мир.

Например, в результате своей профессиональной деятельности, преподаватели кафедр вуза выполняют поиск материалов для подготовки лекционных курсов и материалов для самостоятельного изучения студентами, подбирают источники для формирования списка литературы, а также ведут учет и архивирование различной документации.

Сотрудники работают с информационными ресурсами в виде текстовых документов, презентаций, наборов данных и БД различного типа, программных пакетов, изображений, аудио и видеофайлов, обеспечивающих качественную организацию учебного процесса и научно-методической деятельности.

Данные могут быть структурированными (мета-описания, онтологии, метаданные библиотечных систем, БД, анкеты, тесты, семантически размеченные

файлы и т.п.), слабоструктурированными (данные социальных сетей, шаблонные документы, отчеты, ВКР) и неструктурированными (лекции, рефераты, публикации электронных СМИ, электронные письма, сообщения в форумах, информация с сайтов конференций).

В настоящее время кафедрами вузов уже накоплен большой объем знаний и информационных ресурсов по различным курсам, по результатам научной и методической работы. Однако эти данные слабо структурированы, плохо систематизированы, рассредоточены по различным ресурсам, библиотекам и архивам, что существенно ограничивает к ним доступ. Более того, по историческим, техническим и другим причинам, тематически связанные данные сохраняются в разных форматах под управлением различных систем хранения и обработки данных. Такое положение дел приводит к тому, что разнообразные коллекции, базы персоналий и публикаций, даже расположенные на одном физическом сервере, зачастую имеют различные логические входы и представляют собой разрозненные автономные информационные ресурсы. Отсутствие связанности информационных ресурсов и унифицированного доступа к ним приводят к неполноте рассмотрения и учета существующих данных и знаний.

Хранение знаний в организациях необходимо для формирования и наращивания активов знаний (интеллектуального капитала, баз знаний). Ввиду существенного различия явных и неявных знаний различаются и способы их хранения.

Явными знаниями являются текстовые документы, электронные таблицы, базы данных, Web-страницы, чертежи, схемы, почтовые сообщения и т. п., которые хранятся в специально создаваемых для этой цели репозитариях знаний. Поскольку технологическую основу такого рода хранилищ составляют СУБД и современные ИТ, то применяются и соответствующие способы введения и извлечения знаний из репозитариев. Перед помещением явных знаний в репозитарий осуществляется их аннотирование, классификация и систематизация, без чего немыслимо их эффективное хранение с целью обеспечения в дальнейшем эффективного поиска.

Для того чтобы потенциал репозитария знаний использовался в основных бизнес-процессах организации, необходимо осуществлять актуализацию и верификацию хранящихся в нем знаний применительно к новым задачам и меняющимся условиям бизнес-среды.

Неявные знания хранятся, прежде всего, в нейронных структурах головного мозга сотрудников организации. Кроме того, они «хранятся» в группах, в командах, в организационных формах ежедневной деятельности, в правилах выполнения рутинных работ и процессов, которые описаны либо даже не описаны явно.

Существующие подходы к работе с информацией становятся недостаточно эффективными. Совершенствование существующих и разработка новых эффективных подходов к сбору, хранению и обработке информации является неотъемлемой частью процесса развития информационных технологий (ИТ) и информационных систем (ИС).

1.2 Системы управления знаниями научно-образовательных учреждений

Невозможно представить любое научное или научно-образовательное учреждение без развитой комплексной автоматизированной системы управления. Работы в этом направлении активно велись еще, начиная с 80-х годов прошлого столетия и продолжаются до сих пор.

Система управления знаниями (СУЗ) или Knowledge Management Systems (KMS) является корпоративной автоматизированной системой, предназначенной для создания, поиска, распространения, обработки, хранения, извлечения, производства и предоставления для использования знаний по вопросам профессиональной деятельности учреждения.

Приведем несколько определений процесса управления знаниями:

- систематический захват, хранение и повторное использование профессионального опыта и знаний (определение Ernst & Young) [15];
- создание условий, при которых люди смогут получать информацию и знания в нужное для выполнения задач время;

– управленческие действия, направленные на использование информационных ресурсов компании (определение IBM Lotus) [16];
– является средством, а не конечной целью (определение Б. Гейтса) [17].

Как следует из выше сказанного, основной задачей СУЗ становится извлечение структурированной информации из неструктурированной и разно-структурированной, а также кодификация неявных знаний в структурированные БД знаний.

По законодательству РФ информационные ресурсы имеют слишком широкое толкование, но как только к термину «управление информационными ресурсами» добавляется термин «система», эта новая сущность сразу приобретает характер информационной системы в смысле определения, данного в Законе РФ: «Информационная система – совокупность содержащейся в базах данных информации и обеспечивающих ее обработку информационных технологий и технических средств».

В связи с этим следует отметить, что переход к семантическим и онтологическим моделям данных, являющихся каркасом любой СУЗ, позволит учитывать содержание документов и извлекать важные для сотрудников знания и факты.

Знания, как объект управления, в соответствии с классическими принципами теории управления имеет ряд особенностей: знания являются одновременно ресурсом и результатом деятельности, они проявляют себя и на «входе», и на «выходе» системы, они же могут выступать в качестве управляющих воздействий.

Цель управления знаниями в организации – создание таких механизмов, при которых накопленные знания эффективно используются для выполнения важных для учреждения задач. Органом управления, вырабатывающем управляющие сигналы, являются сотрудники организации.

Эффективность системы управления знаниями можно оценить по совокупности трех показателей: результативности, ресурсоемкости и оперативности. Результативность системы управления знаниями организации – это степень достижения целей, ради которых она существует, т.е. степень извлечения

полезных соответствующих запросам знаний. Под ресурсоемкостью понимается совокупность общих затрат вычислительных ресурсов, используемых для получения целевого эффекта системы управления знаниями. Оперативность системы управления знаниями характеризуется расходом времени, необходимого для достижения поставленной цели и отражает ее быстродействие при реагировании на изменение внешних условий.

Существуют два значительно отличающихся подхода к построению систем управления знаниями (СУЗ). Первый можно назвать классическим, когда СУЗ строится на основе комбинирования существующих технологий для поддержки различных процессов работы со знанием (E-mail, доски объявлений, дискуссионные форумы, общие каталоги документов, порталы, метаданные), а также на специфических технологиях, тяготеющих к инструментарию искусственного интеллекта, таких, как автоматическая классификация, автоматическое аннотирование документов, распознавание образов и речи и т.п.

Второй подход правильнее определить, как семантический. Он основан на использовании взаимосвязанного набора методов и технологий по работе со смыслом, семантикой данных, информацией и знаниями. В их числе онтологии предметных областей, технологии их построения и сопровождения, семантические метаданные, семантический поиск, системы логического вывода, и т.п. При этом семантический подход не отвергает классический. Большинство элементов и инструментов классического подхода зачисляется в арсенал средств и, по существу, применяются для повышения качества работы с данными и информацией.

Существенным отличительным признаком СУЗ от корпоративной автоматизированной системы управления (АСУ) следует считать работу с семантикой и использование семантически ориентированных подходов, методов и технологий, онтологий, а также тот факт, что объектом управления является сам информационный ресурс.

1.3 Обзор подобных систем обработки информации

Настоящее время характеризуется резким ростом объемов данных, обусловленных техническим прогрессом и необходимостью единообразного представления данных и знаний. Проблемы разработки и реализации различных СУЗ были широко исследованы как американскими, европейскими, японскими учеными, так и отечественными специалистами, и вопросам разработки систем посвящен целый ряд работ.

Начиная с 60х-70х годов в работах Э. Тоффлера, Д. Белла, М. Маклюэна уделяется внимание понятиям «знания», «информация» и развивается концепция нового информационного общества [18].

В СССР еще в 1970-1980-хх годах начинают разрабатываться информационные основы управления знаниями в трудах В.М. Глушкова, Л. С. Козачкова [19, 20]. В.М. Глушков считал, что последовательное накопление знаний и эффективные способы их обработки, развитие интеллектуальных способностей ЭВМ, обеспечат прорыв в развитии цивилизации и переход к информационному обществу.

Термин «управление знаниями» введен К. Вигом почти два десятилетия назад и активно использовался в его работах [20, 21]. В труде Т. Давенпорта и Л. Прусака был описан способ преобразования данных в информацию и отличия между этими понятиями [22].

Японские исследователи И. Нонака и Х. Такэучи внесли большой вклад в развитие концепции управления знаниями. По их утверждению, между явными и неявными знаниями происходит непрерывный обмен и трансформация. Была предложена модель процесса создания знания учреждения, когда процесс создания организационного (корпоративного) знания начинается с распространения неявных знаний (фаза 1), затем неявное знание преобразуется в явное (фаза 2) [23].

П. Ромер отмечает в [24]: «Знания – единственный неограниченный ресурс, единственный актив, который увеличивается по мере его использования».

К 2000 году исследования зарубежных ученых переходят на промышленный уровень, создан «Институт управления знаниями» – коммерческий исследовательский консорциум. Увеличилось количество проводимых конференций, посвященных управлению знаниями и соответственно количество публикаций.

В России в этот период вышел сборник «Новая постиндустриальная волна на Западе» под редакцией В. Л. Иноземцева [25]. В 2003 году опубликована статья «Управление знаниями: эволюция и революция в организации» первого заместителя Института экономики РАН, доктора экономических наук Б. Мильнера [26].

Большую роль в развитии инженерии знаний сыграли работы российских ученых А.Ф Тузовского, В.З. Ямпольского, в работах которых пространство знаний (интеллектуальное пространство) организации предлагается описывать следующим образом: в качестве системы координат использовать онтологию предметной области, описания же объектов, содержащие знания, задавать в виде их метаописаний, в качестве меры близости объектов использовать семантическую близость их метаописаний.

Г.Б. Клейнер, В.Л. Макаров [27] указали на необходимость применения системного подхода к управлению знаниями. Предприятие и окружающее его социально-экономическое пространство являются взаимодействующими системами, а обмен знаниями между ними – частью общесистемных отношений. Поэтому, применяя системный подход, можно определить сдерживающие и стимулирующие факторы распространения и накопления знаний.

В последние десятилетия для структурирования, формализации и унификации представления знаний с целью их многократного и гибкого использования в информационных системах применяются онтологии.

В ходе изучения материала оказалось, что исследования по использованию онтологий в ИС также достаточно широко распространены и отражены в работах известных российских и зарубежных ученых, например, Т. Груббера [28, 29], Н. Гуарино [30, 31], S. Staab [32], Jos de Bruijn и D. Fensel [33], Т.А. Гавриловой

[34], В.Ф. Хорошевского [35], А.В. Смирнова [36], Г.С. Осипова [37], В.Ш. Рубашкина [38].

Термин «онтология» мигрировал из философии в область компьютерной науки. Онтология, по общепринятому определению, есть спецификация концептуальной модели, формализованное представление основных понятий и связей между ними [39]. Существует и множество других определений. Общим для всех существующих определений является понимание онтологии как модели представления знаний какой-либо предметной области в виде набора понятий этой предметной области и существующих между ними отношений.

Т.А. Гавриловой и Ф.В. Хорошевским были классифицированы тезаурусы и словари, а также приведена взаимосвязь между различными онтологиями формальной модели онтологической системы.

Позже онтологии стали рассматриваться в качестве ключевого элемента в проекте семантической сети – нового этапа развития сети WWW (Word Wide Web). Если существующая Web-сеть – это огромное множество документов, которые связаны перекрестными ссылками, то создаваемая семантическая сеть должна добавить к существующей сети множество онтологий и мета-описаний знаний, содержащихся в документах Web-сети (включая стандарты и программные инструменты) [40].

Об использовании формальной модели онтологии для совокупности объектов некоторой предметной области и определении структуры метаданных (мета-описаний) говорится в работе Стефана Стааба [41].

В работе [42] описываются различные особенности и теоретические преимущества фреймворка knowledge-based framework, основанного на знаниях, введенных экспертом в систему, в интеллектуальном автоматизированном химическом производстве с целью минимизации затрат и максимизации эффективности производственной линии за счет предотвращения сбоев, опасных ситуаций и оптимизации планов производства.

В сфере здравоохранения существует множество АСУ, основанных на онтологии. Так, например, контекстно-ориентированная структура управления

доступом, которая использует онтологию как модель знаний наряду с соответствующим методом вывода на основе Web Ontology Language (OWL) и Semantic Web Rule Language (SWRL) [43].

В статье [44] описаны этапы проектирования, разработки и валидации iOSC3 – онтологической системы для интеллектуального наблюдения и лечения критических пациентов с острыми сердечными расстройствами в отделе интенсивной терапии. Система анализирует состояние пациента и дает рекомендации по лечению, которое необходимо назначить для достижения максимально быстрого выздоровления.

Механизм интерпретации высказываний для интеллектуальных разговорных интерфейсов, основанный на онтологии, был описан в источнике [45]. В этом механизме онтологии используются для синтаксической и семантической интерпретации.

В работе [46] на основе анализа семантики финансовой отчетности предприятия разработана онтологическая модель для области финансовой деятельности организации. Путем объединения онтологической модели финансовой области с алгоритмом интеллектуального анализа правил разработана новая модель бизнес-аналитики с прогнозированием банкротств.

В статье [47] представлена онтологическая модель TSH, которая разработана по модульному принципу и реализована в OWL с использованием Protege2000 с целью использования всего потенциала онтологий для описания предметной области, чтобы обеспечить эффективную базу для разработки, настройки и выполнения программных приложений.

В работе С. Ниренбурга, В. Раскина [48] утверждается о возможности использования онтологий как в системах извлечения знаний, так и при информационном поиске знаний, при аннотировании текстов, при классификации знаний и др.

Для поддержки деятельности инженеров по знаниям и системных архитекторов было предложено несколько подходов, основанных на онтологии. Akerman and Tyree [49] предложили основанный на онтологии подход для

поддержки процесса разработки программного обеспечения. Ameller и Franch [50] представили онтологию с именем Arteon для представления знаний. Эта онтология призвана обеспечить строительные блоки архитектурных представлений, фреймворков и элементов при построении архитектуры программного обеспечения.

В своей основополагающей работе Р. Крухтен [51] предлагает онтологию для управления архитектурными знаниями, вводит таксономию архитектурных решений, атрибуты и связи между такими понятиями, как требования, дефекты, элементы проектирования и реализации. Преимущество такой онтологии заключается в том, что она сохраняет сложные графики взаимосвязанных проектных решений и поддерживает примеры их использования, что позволяет сформировать рекомендации для поддержки архитекторов программного обеспечения в процессе принятия решений.

Солиман и др. предложили механизм поиска для извлечения проектной информации из StackOverflow, онлайн-сообщества разработчиков [52].

В одной из работ [53] автор предлагает моделировать и захватывать термины в онтологии с помощью OWL.

Система восстановления структуры проектного решения и архитектуры TREx [54] использует доменные онтологии для аннотирования текста темами и предопределенными правилами в операторах тегов TREx в качестве проектных решений или проектных структур.

В Российской Федерации и странах СНГ выпущено достаточно большое количество различных монографий, публикаций, статей на Интернет-порталах, посвященных в той или иной форме концепции управления знаниями. Регулярно проводятся конференции, создаются форумы для обсуждения проблем и перспектив внедрения технологии управления знаниями в учреждениях.

Так идеи, используемые в их технологиях и методологиях, оказали непосредственное влияние на реализацию подсистемы проектирования в разработанном инструментарии разработки Web-порталов PORTO (PORTal Ontology) [55].

Концептуальным стержнем системы УЗОР, предназначенной для управления знаниями организации, является система онтологий компании [56]. Сформированы алгоритмы онтологического инжиниринга, направленные на поддержку практического формирования онтологий.

В работе [57] приведен широкий обзор опыта применения систем управления знаниями в различных компаниях нефтеперерабатывающей и нефтедобывающей отраслей, в том числе и таких, как ПАО «Лукойл» и ПАО «Газпром нефть».

Статья [58] посвящена проблеме создания и внедрения системы управления знаниями (СУЗ) в Госкорпорации «Росатом» (ГК «Росатом») и ее организациях, в частности на площадке Акционерного общества «Институт реакторных материалов» (АО «ИРМ»). Опыт АО «ИРМ» в сфере управления знаниями в рамках ключевых функциональных блоков СУЗ ГК «Росатом» показал, что внедрение СУЗ позволяет систематизировать знания организации с точки зрения контента и экспертных сообществ, упорядочить процессы поиска и анализа информации, а также способствует сохранению критически важных знаний и формированию нематериальных активов организации.

Онтологии также используются в сфере образования и электронного обучения. Системы управления знаниями основаны на прочных связях, определенных в метаданных объектов обучения, которые позволяют комбинировать их с другими объектами обучения, чтобы сформировать целостную образовательную программу. В адаптивных образовательных системах, доступных студентам, интеграция таких систем превращается из интересной исследовательской задачи в важную практическую задачу, решение которой базируется на основе онтологий и метаданных [59].

Система ИСТИНА [60] – интеллектуальная система тематического исследования наукометрических данных, предназначенная для перманентного сбора и систематизации, хранения и анализа наукометрической информации в вузах и научных организациях с целью подготовки и принятия управленческих решений. Одной из главных задач является формирование современной информационно-аналитической интегрированной среды для автоматизации

процессов управления научными исследованиями, инновационной и образовательной деятельностью.

Исследование литературных источников по применяемым на практике методам классификации показало, что существует значительно меньше публикаций с описанием методов, основанных на знаниях и онтологиях, чем основанных на методах машинного обучения, а имеющейся информации не всегда достаточно для подробного анализа применяемых методов. Недостаток публикаций связан в первую очередь с тем, что классификация с использованием знаний экспертов специфична для каждой предметной области.

В статье [61] описывается технология автоматического рубрицирования документов, применяемая в большой электронной библиотеке LexisNexis на базе ручного описания рубрик. Для большинства рубрик качество классификации находится на уровне более 90% полноты и точности (к сожалению, в указанной статье методика получения данных оценок не приводится).

Технология классификации, основанная на знаниях и разработанная в рамках проекта УИС РОССИЯ, показала высокую эффективность при создании систем рубрикации для различных текстовых 39 коллекций [62].

Как следует из обзора, построение информационных систем на базе онтологий является весьма перспективным, как и использование онтологий в различных методах и алгоритмах. Но приведенные аналоги обладают избыточностью или недостаточностью требуемого функционала, позиционируются в качестве инструментов для решения несколько других задач.

Таким образом, разработка системы управления информационными ресурсами научно-образовательных учреждений на данный момент является актуальной задачей, требующей решения.

1.4 Обзор существующих моделей и алгоритмов обработки информации в системах организационного сопровождения деятельности научно-образовательных учреждений

1.4.1 Модели представления знаний

Модели представления знаний (МПЗ) – это «язык» и концептуальная схема для описания знаний. На сегодняшний день разработано уже достаточное количество моделей. Каждая из них обладает своими плюсами и минусами, поэтому для каждой конкретной задачи необходимо выбрать именно свою модель. От этого будет зависеть не столько эффективность выполнения поставленной задачи, сколько возможность ее решения вообще.

Отметим, что модели представления знаний относятся к прагматическому направлению исследований в области искусственного интеллекта. Это направление основано на предположении о том, что мыслительная деятельность человека – «черный ящик». При таком подходе не ставится вопрос об адекватности используемых в компьютере моделей представления знаний тем моделям, которыми пользуется в аналогичных ситуациях человек, а рассматривается лишь конечный результат решения конкретных задач.

Рассмотрим наиболее часто используемые и популярные на сегодняшний день модели представления знаний [63].

1. Продукционные модели – модели, основанные на правилах, которые позволяют представить знания в виде предложений типа: «ЕСЛИ условие, ТО действие». Под условием понимается некоторое предложение – образец, по которому осуществляется поиск в базе знаний.

2. Семантические сети – графическое изображение модели, чаще всего в виде графов. Узлы этого графа соответствуют понятиям и объектам, а дуги – отношениям между объектами.

3. Фреймовые модели основываются на таком понятии как фрейм (англ. frame – рамка, каркас). Фрейм – структура данных для представления некоторого концептуального объекта. Информация, относящаяся к фрейму, содержится в составляющих его слотах. Слоты могут быть терминальными либо являться сами фреймами, т.о. образуя целую иерархическую сеть.

4. Модели нечеткой логики базируются на неточных числах, коэффициентах уверенности, вероятности, нечетких множествах. Последние содержат упорядоченные пары, включающие номер элемента множества и функцию степени принадлежности этого элемента множеству.

Продукционные модели можно считать наиболее распространенными моделями представления знаний. Основные достоинства систем, основанных на продукционных моделях: простота представления знаний; легкость организации логических выводов; модульность применения правил; явность и наглядность интерпретации правил; простота механизма вывода; простота модификации базы знаний.

К недостаткам таких систем относятся: низкая эффективность обработки знаний; неоднозначность выбора правил вывода; низкая эффективность и негибкость механизма вывода; неоднозначность учёта взаимосвязи между отдельными продукциями; сложность восприимчивости; сложность оценки целостного представления о предметной области. Реализация этих моделей базируется на языках типа Prolog.

При разработке небольших систем (десятки правил) проявляются в основном положительные стороны продукционных моделей знаний, однако при увеличении объёма знаний более заметными становятся слабые стороны. Продукционная модель обладает тем недостатком, что при накоплении достаточно большого числа (порядка нескольких сотен) продукций они начинают противоречить друг другу.

Основная идея при построении логических моделей знаний заключается в следующем: вся информация, необходимая для решения прикладных задач, рассматривается как совокупность фактов и утверждений, которые представляются как формулы в некоторой логике. Знания отображаются совокупностью таких формул, а получение новых знаний сводится к реализации процедур логического вывода.

Основные достоинства логических моделей знаний: в качестве «фундамента» здесь используется классический аппарат математической логики, методы которой достаточно хорошо изучены и формально обоснованы; существуют достаточно

эффективные процедуры вывода, в том числе реализованные в языке логического программирования Prolog; в базах знаний можно хранить лишь множество аксиом, а все остальные знания получать из них по правилам вывода; высокий уровень формализации; согласованность знаний как единого целого.

Недостатки данной модели: представление знаний в таких моделях не наглядно; из-за того, что факты (формулы) выглядят очень похоже, модель тяжело использовать для конкретной предметной области; из-за отсутствия определённости в некоторых сферах науки, в логическую модель тяжело добавить необходимое количество аксиом для корректной работы будущей системы; вывод, полученный из верных аксиом может не иметь смысла со стороны человеческого разума; каждая аксиома должна иметь строгий вывод, зачастую либо «да», либо «нет»; привязка к языку программирования Prolog.

Однозначное определение семантической сети в настоящее время отсутствует. В инженерии знаний под ней подразумевается граф, отображающий смысл целостного образа. Узлы графа соответствуют понятиям и объектам, а дуги – отношениям между объектами. Семантическая сеть, как модель, наиболее часто используется для представления декларативных знаний. Семантическая сеть позволяет снизить объем хранимых данных, обеспечивает вывод умозаключений по ассоциативным связям.

Достоинства семантических сетей: универсальность; наглядность системы знаний, представленной графически; близость структуры сети, представляющей систему знаний, к семантической структуре фраз на естественном языке.

Недостатки семантических сетей: формирование и модификация семантической модели затруднительны; поиск решения в семантической сети сводится к задаче поиска фрагмента сети, соответствующего подсети, отражающей поставленный запрос; чем больше отношений между понятиями, тем сложнее использовать и модифицировать знания.

Несмотря на недостатки, семантическая сеть, в связи со своей наглядностью и легкостью создания, незаменима, особенно при решении задач в системах

распознавания речи и понимания естественного языка, а также в технологиях Semantic Web для создания глобальной базы знаний на основе Интернет.

Фреймовую модель можно считать более специализированной по отношению к сетевой. Она основана на принципе кластеризации (фрагментации) знаний. Фрейм – это структура для представления знаний, которая при ее заполнении соответствующими значениями превращается в описание конкретного факта, события или ситуации.

К достоинствам фреймовой модели знаний относятся: гибкость, т. е. структурное описание сложных объектов; наглядность, т. е. данные о родовидовых связях хранятся явно; механизм наследования свойств; возможность использования предположений и ожиданий; универсальность за счет существования не только фреймов для обозначения объектов и понятий, но и фреймов-событий, фреймов-ситуаций, фреймов-ролей, фреймов-сценариев и т.п.; возможность легкого перехода к сетевой модели.

Недостатками фреймовой системы являются: высокая сложность систем в целом; отсутствие строгой формализации; трудно внести изменение в иерархию; фреймовую модель представления знаний можно заменить сетевой (семантической).

В последнее время в разработках группы российских ученых для представления знаний используются миварные сети – принципиально новый подход к описанию и формализации любых типов знаний в сочетании с возможностью решать сложные логические задачи на сверхбольших массивах данных.

Миварная технология накопления информации — это способ создания глобальных эволюционных баз данных и правил (знаний) с изменяемой структурой на основе адаптивного дискретного миварного информационного пространства унифицированного представления данных и правил, базирующегося на трех основных понятиях «вещь, свойство, отношение» [64]. К достоинствам технологии можно отнести ее универсальность и простоту логического вывода, к недостаткам– ее недостаточную распространенность.

Так как продукционные и логические модели не представляют данные в удобном и наглядном виде, фреймовые модели сложны в реализации, а миварные сети еще недостаточно распространены, то в качестве базовой модели остановим свой выбор на семантической сети.

В сфере искусственного интеллекта наиболее концептуальные и универсальные понятия моделируемых областей, абстрагированные полностью от их практической реализации, представляют собой онтологии предметных областей. Для графов знаний онтология – семантическая основа представления данных, базирующаяся на логике и включающая терминологический словарь и набор утверждений о моделируемых объектах.

Таким образом, в качестве модели представления знаний целесообразно использовать модель семантической сети с семантическим «каркасом» в виде онтологии.

1.4.2 Модели представления текста

В системах информационного поиска используются различные подходы к построению представлений хранимых документов. От характера используемых представлений документа существенно зависит качество поиска, в том числе точность, полнота и другие параметры.

Модель представления текста – это формализованная математическая структура, построенная по неструктурированному тексту. Существуют следующие модели: векторная, языковая, модель скрытых тематик и др.

В векторной модели Vector Space model (VSM) каждый терм представляет собой вектор в пространстве слов [65]. Суть метода отображения текста в вектор заключается в том, что каждому слову соответствует определенная координата в пространстве признаков или вес в соответствии с выбранной весовой функцией.

Для полного определения векторной модели текста необходимо выбрать саму весовую функцию. Главным образом, оценочные весовые функции строятся

на частотных и вероятностных характеристиках текста, а также на дистрибутивной семантике.

Среди частотных методов используются следующие функции взвешивания: двоичные – вес равен 1, если терм встречается в документе и 0 если не встречается; TF (term frequency) – функция веса вычисляется от количества вхождений терма в документе; TF-IDF (term frequency – inverse document frequency) – функция веса вычисляется как произведение функции TF и функции IDF, благодаря которой можно снизить весомость наиболее широко используемых слов (предлогов, союзов, общих терминов и понятий) [66].

Существуют и другие весовые функции, например, логарифмическая, скорректированных весов Гаусса, вероятностная, которая основана на понятии условной вероятности и теория Байеса, на моделях дистрибутивной семантики [67].

В настоящее время наиболее часто используются векторная модель и модель на основе латентных семантик, а также их модификации.

Векторная модель имеет ряд расширений, среди которых наиболее известна bag-of-words (BOW). В модели BOW текст представляет собой набор неупорядоченных и несвязанных слов. Считается, что тексты, не имеющие общих слов или имеющие их небольшое количество, являются неблизкими по смыслу (семантически и тематически), что в целом неверно.

Для определения подобия между двумя документами используются различные меры сходства между двумя векторами, определяющими эти документы. Наиболее часто рассчитывают меру косинусной меры на основе скалярного произведения векторов, позволяющей провести оценку смысловой близости.

Достоинствами базовой векторной модели BOW является простота, возможность использования операций линейной алгебры для определения сходства между текстами и при ранжировании слов в поисковых запросах.

Недостатками является отсутствие семантики, низкая степень сходства для текстов большой длины, отсутствие учета синонимии и полисемии. Главный недостаток заключается в том, что множество слов проецируется в пространство

высокой размерности и разреженности и появляется эффект так называемого «проклятия размерности», когда сложность вычислений растет экспоненциально из-за увеличения размерности данных.

Для преодоления этих проблем проводится много исследований, по результатам которых написано немало работ. В качестве одного из вариантов решения многие авторы видят в необходимость перехода от традиционного представления текста в разреженном пространстве к новым - семантическим пространствам.

Среди широко известных рассмотрено три подхода: на базе концептов, на базе контекстных векторов, на базе латентных семантических связей [68].

Модель bag-of-concepts (BOC) построена на основе BOW, позволяет учитывать скрытые смысловые связи. Концептами будем считать наборы синонимичных и семантически связанных слов с одинаковыми и похожими смыслами. Каждый концепт представляется не только с помощью определяющих его слов, но и с помощью контекстного вектора, содержащего веса слов в концепте.

Для получения набора концептов, близких к данному тексту, достаточно иметь вектор концепта и вектор текста, содержащий веса этих же слов в тексте. Два вектора сравниваются по какой-либо мере семантической близости и отбираются несколько наиболее близких, которые и образуют после сортировки вектор, представляющий текст в новом пространстве.

В качестве оператора перехода из одного пространства дескрипторов в другое используется матрица семантических связей «термина-на-термины» размерности [nxn], состоящая из контекстных векторов n-мерного пространства.

Пусть текст представлен вектором V(n). Матрица семантических связей S(nxn) позволяет отобразить V(n) – исходное представление текста, в новое представление U(n), уже отражающие семантические связи между словами. Математически модель можно представить в виде выражения:

$$U_n=V_n*S_{nn}. \quad (1.1)$$

В отличие от модели BOC, модель «контекстных векторов» учитывает зависимости между всеми словами текста, а не только концептами, и формирует контекстные вектора по всем словам документа.

Представление текста на основе моделей скрытых тематик – это класс моделей, объединённых общим предположением о существовании скрытых (латентных) тем. Одна из первых тематических моделей – это латентно-семантический анализ (или латентно-семантическая индексация). В результате количественного анализа латентных факторов можно получить новое семантическое представление нового документа, в результате которого веса терминов могут быть скорректированы, и поисковый образ документа станет более адекватным его содержанию. Модель латентного семантического анализа относят к классу векторных моделей, однако другие модели скрытых тем имеют более сложную структуру и уходят от векторного представления текстов.

Каждый набор документов (коллекция) имеет неявную, латентную семантическую структуру, представляющую собой объединение отдельных терминов документов. Формирование представлений осуществляется путем факторизации матрицы «Документы-термины» [69].

Существуют несколько способов определения тематик, например, LSA/LSI использует сингулярное разложение (SVD) в качестве факторизации (Рисунок 1.1).

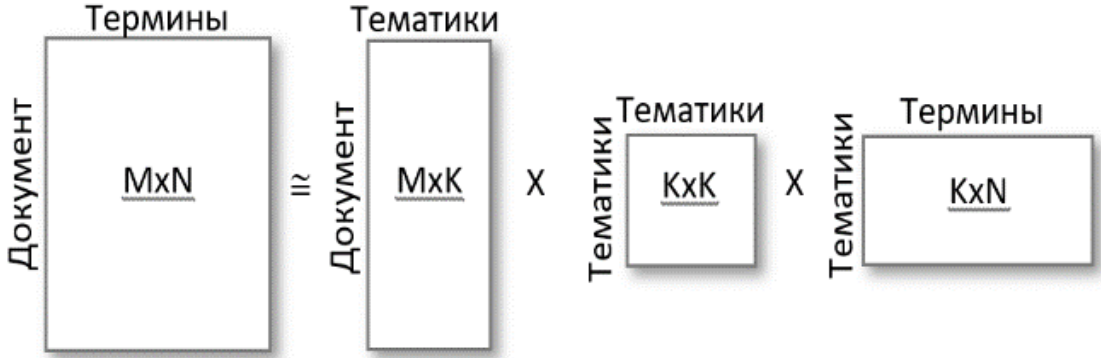


Рисунок 1.1 – SVD разложение для LSA/LSI

Извлечение скрытых тематик также можно произвести на основе метода неотрицательной матричной факторизации (NMF), который вместо SVD использует разложение на две матрицы (Рисунок 1.2) [70].

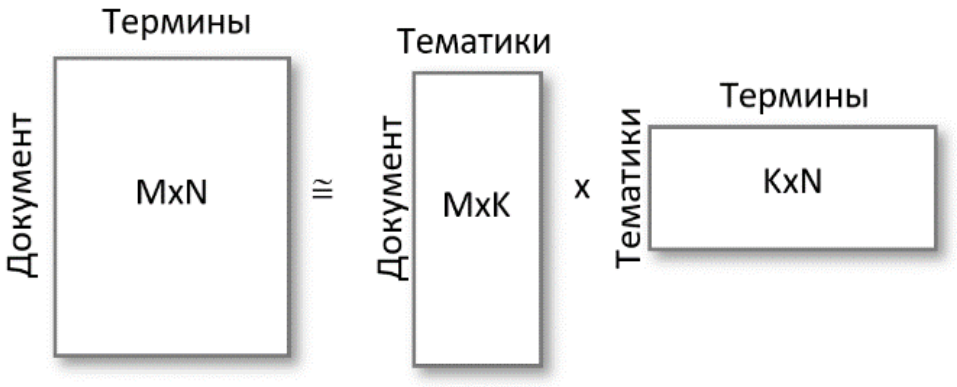


Рисунок 1.2 – Метод NMF

Основным достоинством векторной модели является ее простота и тот факт, что векторное представление текстов делает возможным использование линейно-алгебраических операций для определения сходства между текстами и ранжирования текстов по соответствию запросу. Для этих целей используется косинусная мера релевантности. Другим очевидным достоинством векторной модели является простота ее построения по заданному корпусу текстов.

Однако, за внешней простотой векторной модели кроются некоторые существенные недостатки. Прежде всего, главная предпосылка векторный модели, которая с статистической точки зрения означает гипотезу о независимости слов, что в корне неверно с точки зрения лингвистики и анализа естественного языка. Использование же нормированного скалярного произведения в качестве меры сходства приводит к тому, что более длинные тексты всегда имеют низкую степень сходства с остальными текстами из-за нормировки длиной текста.

Языковая модель (language model) позволяет оценить вероятность появления последовательности слов в тексте. В этой модели текст представляется с помощью цепей Маркова, где каждому узлу соответствует одно слово, а на ребрах – вероятности того, что одно слово встретится после другого. Наиболее востребованы два вида языковых моделей: модель униграмм (одиночных слов) и модель биграмм (последовательных пар слов). При использовании языковых моделей нет необходимости в нормальном преставлении всего текста – говорят о вероятности текста, или о вероятности появления его фрагмента. Так же, как и

векторная модель, модель униграмм основана на предположении о независимости появления слова в тексте от предыдущего слова.

Языковые модели используются в тех случаях, когда важно сохранить короткие семантические связи: в задачах машинного перевода, распознавания речи, исправлении опечаток [71, 72, 73]. Данная работа посвящена задачам другого рода, поэтому мы не будем в дальнейшем заострять внимание на языковых моделях.

В простейшей формулировке теоретико-множественная модель представления текстов предполагает следующее: каждый текст представляется неупорядоченным набором термов. Естественным применением такой теоретико-множественной модели можно считать вычисления сходства двух текстов. Пусть дано два текста и каждый текст представлен множеством термов. Тогда сходство между двумя текстами можно оценить с использованием любой теоретико-множественной меры близости. Как правило, любой коэффициент тем или иным образом учитывает количество совпадающих термов (мощность пересечения двух множеств термов), так же, как и мощности каждого множества по отдельности или мощность объединения множеств [74].

Главным недостатком всех моделей на базе BOW является огромная разреженность данных и отсутствие семантики. Для устранения этих недостатков векторная модель BOW часто комбинируется с другими моделями.

При разработке модели представления текстов будем в качестве базовой модели использовать векторную модель Bag-of concepts, семантически обогатив ее за счет использования онтологии.

1.4.3 Модели информационного поиска

Существует несколько видов информационного поиска текстовой информации.

1. Информационный поиск с помощью векторно-пространственного представления, которое строится для поискового запроса и для каждого документа.

Поисковая система отбирает документы, пространственные векторы которых подобны пространственному вектора поискового запроса, вычисляя угол между этими векторами. Наиболее подходящими являются документы, пространственно-векторное представление которых направлено туда же, куда и в представления поискового запроса [75].

2. Вероятностный поиск – коэффициент соответствия документа поисковому запросу определяется на основе вероятности того, что документ соответствует поисковому запросу. Предполагая, что условия поиска в поисковом запросе независимые, можно вычислять такую вероятность для каждого поискового термина с поискового запроса. Общая вероятность соответствия документа вычисляется как произведение вероятностей соответствия каждого термина [76].

3. Обычный булевый поиск – документы, которые удовлетворяют логическому запросу попадают в список выдачи по очереди. Идея расширенного булевого поиска заключается в создании возможностей для определения степени соответствия документов поисковому запросу за счет присвоения веса поисковым терминам. Вес терминов учитывается при построении списка соответствия документов к информационному запросу [77].

4. Поиск по скрытому семантическому индексированию основан на обнаружении скрытых семантических связей. Появление терминов в документе представляется с помощью матрицы «термин-документ». В результате удается отделить «шум», так, что два семантически близких документа становятся расположенными рядом в многомерном пространстве [78].

5. Поиск можно строить с использованием нейро-сетей. Узлы нейронной сети «активируются» поисковым запросом. Сила каждой связи нейронной сети используется для вычисления коэффициента соответствия документа к поисковому запросу [79].

6. Поиск возможно выполнять с использованием эволюционных алгоритмов. Путем эволюции можно изменить начальный поисковый запрос. Первоначальный запрос используется с равноправными терминами, или со

терминами, имеющих разный вес. Сгенерированный поисковый запрос остается, если он охватывает известные соответствующие исходному запросу документы, если нет – отвергается [80].

7. Поиск с использованием нечетких множеств построен на нечеткой логике. Документ превращается в нечеткое множество (это множество, содержащее не только сам элемент, но и число, показывающее степень принадлежности элемента множеству).

8. Семантический поиск – вид информационного поиска, в котором релевантность документа запросу определяется с помощью семантики. Задача семантического поиска заключается в повышении точности поисковой выдачи, благодаря пониманию намерений пользователей через контекст.

Каждый из представленных видов поиска имеет свои достоинства и недостатки. Среди перечисленных видов есть и весьма ресурсоемкие, например, использующие нейросетевые и эволюционные алгоритмы, а также неэффективные и устаревшие виды поиска, такие, как булевый.

Так как основные модели и алгоритмы обработки информации научно-образовательных учреждений базируются на семантическом подходе и использовании онтологии, то целесообразно в качестве базовой модели информационного поиска использовать семантический поиск и комбинировать его с поиском, основанном на векторно-пространственном представлении с целью улучшения результатов поиска в случаях, когда онтология недостаточно качественно разработана либо находится на начальной стадии наполнения терминами.

1.4.4 Меры оценки семантической близости

Ключевым моментом при решении задач поиска и классификации данных является разработка алгоритмов расчета количественных оценок семантической близости. Существует также задача разрешения лексической многозначности – выбор правильного значения многозначного термина в зависимости от контекста,

которая в данной системе решается путем расчета семантической близости (СБ) контекста, полученного из онтологии.

Одной из первых моделей оценки СБ является геометрическая модель, при которой каждая ось представляет собой некоторое свойство, а близость объектов интерпретируется как расстояние между объектами. Недостатком этой модели является то, что геометрическая модель никак не учитывает добавление общих свойств к объектам, что по идее должно увеличивать их близость.

Другим устоявшимся подходом является теоретико-множественный подход Тверски, определяющий близость двух объектов через сопоставление одинаковых и различных свойств.

В развитие модели Тверски была предложена нормализованная модель (ratio model), которая используется в большинстве мер при вычислении СБ. Если мера близости S (a, b) между объектами a и b является функцией трех аргументов (A∩B, A-B, B-A), где A и B – множества свойств этих объектов, то формула расчета СБ – следующая:

$$S(a,b) = \frac{f(A \cap B)}{f(A \cap B) + \alpha f(A - B) + \beta f(B - A)} \quad (1.2)$$

Для получения данных большей точности и непротиворечивости используются гибридные модели, которые являются свертками различных мер оценки критериев подобия концептов интегрируемых онтологий и используются при автоматической обработке результатов отображения.

Чаще всего в гибридных мерах используется аддитивная свертка:

$$S(c_1, c_2) = \sum_{i=1}^{n} w_i S^i(c_1, c_2), \quad (1.3)$$

где S (c1, c2) – мера близости между концептами c1и c2; S^i – мера близости по определенному критерию между концептами c1и c2; wi – вес, который определяет относительную важность критерия; n – число критериев.

Еще одной распространенной модификацией аддитивной свертки является использование сигмоидальной функции $Sig(x)$:

$$S(c_1, c_2) = \sum_{i=1}^{n} w_i Sig(S^i(c_1, c_2)), \quad (1.4)$$

$$Sig(x) = \frac{1}{(1+e^{-ax})}, \text{ где } a>0. \quad (1.5)$$

Сигмоидальная функция позволяет повысить веса мер с большими значениями и практически пренебречь мерами с малыми весами.

Веса рассчитываются вручную экспертами или автоматически с помощью обучаемой нейронной сети или генетического алгоритма.

Мера семантической близости между концептами – это числовая оценка их смысловой связанности. По теме оценки СБ терминов и процессам интеграции онтологий написано немало работ. В работе [81] приведен обширный обзор различных методов вычислений мер семантической близости термов внутри онтологий.

Выделяют следующие типы мер семантической близости:

- таксономические – на основе иерархических (родовидовых, таксономических) связей;
- реляционные – на основе неиерархических (ассоциативных, проблемно специфических, «горизонтальных») связей между терминами онтологии;
- атрибутивные.

Также существуют и меры, основанные на обработке лексико-семантических ресурсов и основанные на обработке неразмеченных документов.

В свою очередь таксономические меры подразделяются на:

- основанные на определении кратчайшего пути (количество ребер или вершин) между вершинами;
- основанные на определении глубины таксономического дерева;
- учитывающие глубину наименьшей общей родовой вершины – ближайшего общего родителя (Least Common Subsumer – LCS);
- основанные на понятии общей специфичности двух вершин.

Например, длина кратчайшего пути – простейшая мера СБ, использует для оценки таксономическую иерархию, основанную на отношениях has, part of, is-a.

$$S(C_1, C_2) = \log(\frac{2*N}{d(C_1,C_2)}), \quad (1.6)$$

где N – глубина дерева, d (C_1, C_2) – длина кратчайшего пути.

Недостатками большинства мер, основанных на иерархических структурах, является симметричность. Для расчета меры семантической близости на основе неиерархических (ассоциативных связей) используют сравнение с третьим понятием плюс рекурсивное уточнение.

Среди мер, учитывающих значения атрибутов, известна атрибутивная мера близости, основанная на близости значений общих атрибутов понятий. Пусть A – множество атрибутов, A_i – множество артибутов i-го экземпляра, A_{co} – множество общих атрибутов экземпляров. Тогда атрибутивная мера близости экземпляров S (i_1, i_2) с учетом всех атрибутов из $A_{co}=A_1 \cap A_2$ вычисляется по формуле:

$$S(i_1, i_2) = \frac{1}{A_{co}} \sum_{a \in A_{co}} S(i_1, i_2, a). \quad (1.7)$$

Также существуют и меры, основанные на обработке лексико-семантических ресурсов и основанные на обработке неразмеченных документов. Традиционно в информационно-поисковых системах близость текстовых документов вычисляется как угол между векторами документов, образуемыми весами ключевых слов A_i и B_i – компоненты векторов документов по схеме TF [82]:

$$TF = \frac{\text{количество употреблений слова в документе}}{\text{общее количество слов в документе}}. \quad (1.8)$$

$$cos(\theta) = \frac{A * B}{\|A\| * \|B\|} = \frac{\sum_{i=1}^{n} A_i * B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} * \sqrt{\sum_{i=1}^{n} (B_i)^2}}. \quad (1.9)$$

Многие подходы, основанные на обработке текстовых документов, используют вычисление сходства между словами. Последнее семейство мер, Дайса и Жаккара, коэффициент Танимото – простые, нерекурсивные меры близости, которые традиционно применялись в области информационного поиска [83]. Эти меры определяют схожесть двух множеств на основе общих символов и удобно выражаются в множественно-теоретической форме. Несомненным их плюсом является низкая вычислительная сложность.

Коэффициент Танимото:

$$T = \frac{N_c}{N_a + N_b - N_c}, \quad (1.10)$$

где T – сам коэффициент, который принимает значения от 0 до 1 (чем ближе к 1, тем больше сходство между множествами); N_a – количество элементов в первом множестве; N_b – количество элементов во втором множестве; N_c – количество общих элементов в обоих множествах.

Мера Дайса:

$$D = \frac{|N_a \cap N_b|}{|N_a \cup N_b|}, \quad (1.11)$$

где D – коэффициент Дайса; N_a – количество элементов в первом множестве; N_b – количество элементов во втором множестве.

Коэффициент Жаккара:

$$G = \frac{|N_a \cap N_b|}{|N_a \cup N_b| - |N_a \cap N_b|}, \quad (1.12)$$

где G – коэффициент Жаккара; N_a – количество элементов в первом множестве; N_b – количество элементов во втором множестве.

Среди примеров простейшей гибридной меры можно привести формулу «мягкого косинуса», который кроме векторных характеристик текста учитывает еще и семантику [84]:

$$softcos(\theta) = \frac{\sum_{i,j=1}^{n} S_{ij} * A_i * B_i}{\sqrt{\sum_{i,j=1}^{n} S_{ij} * A_i * A_j} * \sqrt{\sum_{i,j=1}^{n} S_{ij} * B_i * B_j}}, \quad (1.13)$$

где A_i и B_i – компоненты векторов; S_{ij} – матрица семантических связей.

В предложенном подходе [85] мера близости содержит оценку критериев подобия понятий онтологии, состоящую из трех частей: атрибутивная мера (сопоставление атрибутов концептов и значений атрибутов), таксономическая мера (определение степени подобия концептов онтологии на основании их взаимного расположения, рассчитывается длина кратчайшего пути как число концептов в иерархии между двумя рассматриваемыми концептами в онтологии, чем меньше длина пути, тем они ближе) и реляционная мера (учитывает отношения с другими концептами).

$$S(c_1, c_2) = tS^t(c_1, c_1) + pS^p(c_1, c_2) + aS^a(c_1, c_2), \quad (1.14)$$

где S^t, S^p, S^a – таксономическая, реляционная и атрибутивная составляющие коэффициента близост и, соответственно, t, p и a.

Для решения задачи нахождения весовых коэффициентов предложено использование генетического алгоритма, который наиболее эффективно обеспечивает поиск решения для функций, имеющих несколько экстремумов. В качестве общей структуры алгоритма использовался модифицированный генетический алгоритм. В работе [86] предложен методика оценки СБ терминов, отличительной особенностью которой является автоматическое определение весовых коэффициентов с использованием роя частиц.

Очевидно, что чем полнее учитываются характеристики двух сущностей, тем качественнее является мера близости, и, следовательно, гибридные меры близости, сочетающие несколько подходов, являются наиболее перспективными.

1.4.5. Параметры оценки качества поиска и классификации

По результатам поиска или классификации определяем следующие параметры (Таблица 1.1):

Таблица 1.1 – Параметры оценки качества результатов классификации или поисковой системы [87]

параметр	при поиске	при классификации
TP true positive	количество полученных релевантных документов	количество случаев, когда классификатор верно утверждает, что объект принадлежит классу
TN true negative	количество полученных нерелевантных документов	количество случаев, когда классификатор верно утверждает, что объект не принадлежит классу
FP false positive	количество релевантных документов, не выданных по запросу	количество случаев, когда классификатор неверно утверждает, что объект принадлежит классу
FN false negative	количество нерелевантных документов, не выданных по запросу	количество случаев, когда классификатор неверно утверждает, что объект не принадлежит к классу

При оценке качества поискового запроса и при оценке качества классификации используются метрики, которые сведены в таблицу 1.2.

Таблица 1.2 – Метрики оценки качества поисковой системы или результатов классификации [87]

метрика	при поиске	при классификации	формула
Accuracy общая точность	доля релевантных документов в выборке, по отношению ко всем релевантным документам коллекции	показывает долю правильных классификаций	Acc=(TP+TN)/ TP+TN+FP+FN
Recall полнота	доля релевантных документов в выборке по отношению ко всем документам в выборке	показывает долю найденных объектов класса к общему числу объектов класса	Recall= TP/ (TP+FN)
Precision точность	усредненная величина, среднее гармоническое между полнотой и точностью	показывает долю объектов класса среди объектов, выделенных классификатором	Prec=TP/ (TP+FP)
Specificity TPR (true positive rate)	вероятность нахождения релевантного ресурса	показывает долю верных срабатываний классификатора к общему числу объектов за пределами класса.	TPR = TN/ (FP+TN)
Fall-out FPR (false positive rate)	вероятность нахождения нерелевантного ресурса	показывает долю неверных срабатываний классификатора к общему числу объектов за пределами класса.	FPR = FP/ (FP+TN)
F-мера	гармоническое среднее между точностью и полнотой.	гармоническое среднее между точностью и полнотой.	F-Score = 2*Prec *Recall/ (Prec+Recall)

Для многоклассовой классификации матрица несоответствий результатов классификации строится по следующему образцу, представленному в таблице 1.3.

Таблица 1.3 – Матрица несоответствий результатов классификации [87]

Предсказанный класс	**Класс 1 (C_1)**	**Класс 2 (C_2)**	**Класс 3 (C_3)**
1 (P_1)	T_1	F_{12}	F_{13}
2 (P_2)	F_{21}	T_2	F_{23}
3 (P_3)	F_{31}	F_{32}	T_3

В этом случае показатели TP, TN, FP и FN рассчитываются относительно некоторого класса i следующим образом:

$TP_i = T_i$; $FP_i = \sum F_{i,c}$; $FN_i = \sum F_{c,i}$; $TN_i = All - TP_i - FP_i - FN_i$.

1.5 Постановка задачи исследования

Как следует из раздела 1.3, в последние годы активно ведутся исследования по разработке систем управления научными и проектными знаниями различных учреждений. Кроме возможностей СУЗ по решению задач классификации знаний различного вида, системы также могут извлекать и даже формировать новые знания. Такие СУЗ актуальны не только в учебной работе школьных учителей и преподавателей вузов, но и в их методической работе, в научных исследованиях работников научно-исследовательских учреждений, а также при разработке различных ИТ проектов фирмами-разработчиками. Крайне важно, что подобные системы накапливают корпоративные знания, составляя некоторый интеллектуальный капитал, с которым могут работать другие сотрудники учреждений.

Перспективность данного направления подтверждается результатами исследований зарубежных и российских учёных.

Таким образом, можно констатировать, что важность проблематики управления знаниями осознается большинством специалистов, занимающихся корпоративным управлением и информационными технологиями для целей

управления. На данный момент отсутствует устоявшийся и обоснованный набор алгоритмов, методов и моделей, позволяющих эффективно создавать и поддерживать работу с ИР вузов и других научных и научно-образовательных организаций с учетом семантики. Поэтому можно считать, что совершенствование моделей и алгоритмов обработки информации для реализации в системах организационного сопровождения деятельности научно-образовательных учреждений является актуальной и недостаточно исследованной научно-технической задачей, имеющей отраслевое значение.

Целью диссертационной работы является повышение эффективности системы управления информационными ресурсами научно-образовательных учреждений за счет применения онтологического подхода, разработки новых и усовершенствования существующих моделей и алгоритмов поиска, хранения и классификации данных.

Для достижения цели поставлены и решены следующие задачи.

1. Провести системный анализ процессов с целью формализации исследуемого объекта и обосновать возможность использования онтологического подхода к построению системы управления ИР учреждении.

2. Разработать онтологическую модель объектов знаний – аппарат для описания семантики области профессиональной деятельности сотрудников научно-образовательных учреждений.

3. Усовершенствовать гибридную меру оценки семантической близости.

4. Модифицировать векторную комбинированную модель представления текстов на базе известных подходов bag-of-words и bag-of-concepts, улучшив ее за счет использования онтологической модели предметной области.

5. Усовершенствовать модели и алгоритмы поиска, хранения и классификации данных на основе разработанной онтологической модели.

6. Разработать прототипы программных модулей, реализующие предложенные модели и алгоритмы в виде фреймворка, а также выполнить на базе фреймворка реализацию и тестирование программного модуля учета научной деятельности сотрудников учреждения.

1.6 Выводы

1. Показана необходимость совершенствования существующих и разработка новых подходов к сбору, хранению, обработке информации в системах организационного сопровождения деятельности научно-образовательных учреждений.

2. Приведено описание особенностей объекта исследований – ИР как одной из форм представления знаний.

3. На основе обзора существующих разработок и исследований сделан вывод об актуальности выбранной тематики, о перспективности использования онтологий в качестве модели представления знаний и о целесообразности онтологического подхода к разработке системы.

4. Выполнен анализ существующих моделей, методов и алгоритмов решения задач обработки информации: моделей представления знаний, моделей представления текста, видов информационного поиска, метрик оценки качества поиска и классификации, мер оценки семантической близости.

5. На основе проведенного анализа поставлена цель диссертационной работы, сформулированы основные задачи для решения и выбраны основные направления для достижения поставленных целей.

РАЗДЕЛ 2
ОНТОЛОГИЧЕСКИЙ ПОДХОД В СИСТЕМАХ ИНФОРМАЦИОННОЙ ПОДДЕРЖКИ ДЕЯТЕЛЬНОСТИ НАУЧНО-ОБРАЗОВАТЕЛЬНЫХ УЧРЕЖДЕНИЙ

2.1 Формальное описание онтологии

Существует множество определений онтологий, большинство которых сводится к тому, что онтология некоторым образом описывает понятия предметной области при помощи концептуальных схем [88]. Для устранения исторически сформировавшегося семантического разрыва в терминологии области инженерии знаний и во избежание разночтений, на рисунке 2.1 приведем семантическую сеть, описывающую существующие определения и связи между ними.

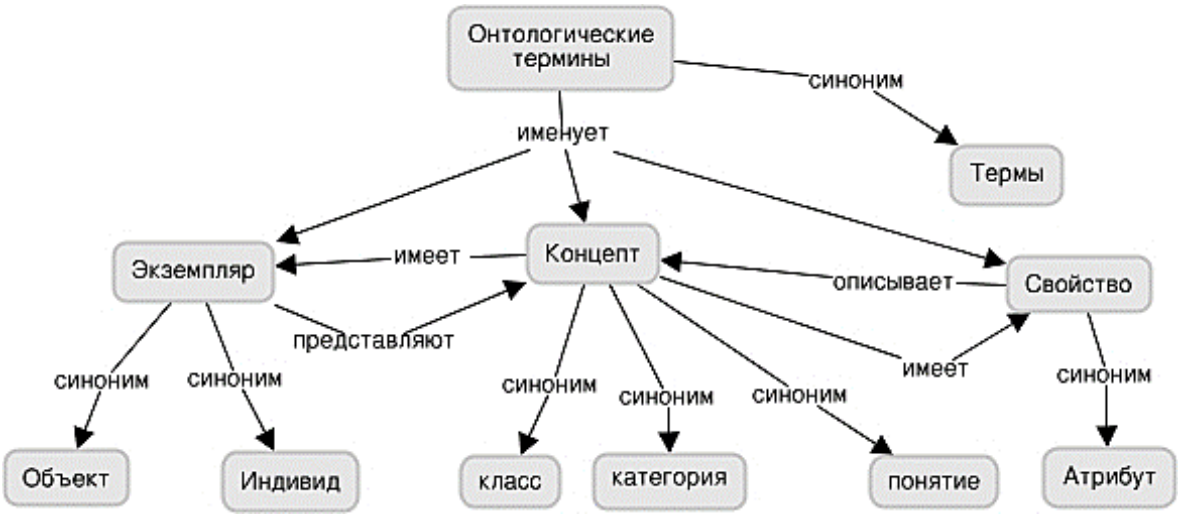


Рисунок 2.1 – Семантическая сеть терминологии онтологий

Концепты, экземпляры, атрибуты и отношения будем называть онтологическими термами. Объекты (индивиды, экземпляры) – слова или словосочетания, являющиеся представлениями концептов.

Все онтологии делятся на следующие виды:

– «весомые» онтологии (Heavy-weighted), содержащие C – множество концептов, R – множество отношений и F – множеств аксиом:

$$O = \{C, R, F\}, \quad (2.1)$$

– «легкие» (Light-weighted), не содержащие аксиом:

$$O = \{C, R, \{\}\}, \quad (2.2)$$

– «словари». Пусть R = Ø и F = Ø. Тогда онтология O трансформируется в простой словарь:

$$O = V = <C, \{\}, \{\}>, \quad (2.3)$$

– «таксономии» – представляющие собой иерархическую систему понятий, связанных между собой отношением is_a («быть элементом класса»). Пусть R = {is_a} и F = Ø. Тогда:

$$O=Tax=<C, \{is\_a\}, \{\} >. \quad (2.4)$$

В силу того, что около 80 % ранее разработанных онтологий относятся к «легким» и они более удобны в реализации, при разработке онтологической модели были использованы «легкие» онтологии.

Формально разрабатываемая онтологическая модель представлена в виде следующей знаковой системы:

$$O = \{C, I, R, A, P, T\} \quad (2.5)$$

где O – онтология; $C = \{c_1, ..., c_n\}$ – конечное множество концептов; n =1...N – количество понятий, присутствующих в онтологии; $I = \{i_1, \ldots, i_u\}$ – конечное множество экземпляров концептов; u=1...U – количество экземпляров, присутствующих в онтологии; $R = \{r_1, ..., r_m\}$ – конечное множество отношений между понятиями $R_i (c_x, c_y)$; m = 1...M – количество отношений между понятиями; $A = \{a_1, ..., a_w\}$ – конечное множество атрибутов, т.е. свойств данных; w =1...W – количество атрибутов; $P = \{p_1, ..., p_l\}$ – свойства объекта (Object Properties), которые представляют собой отношения между двумя экземплярами; l = 1...L – количество объектных свойств; $T = \{t_1, ..., t_k\}$ – конечное множество типов отношений; k =1...K – количество различных типов отношений.

Исходя из требований, предъявляемых к онтологии, следует, что общее количество понятий, используемых в онтологии, должно стремиться к максимальному числу понятий, используемых в данной предметной области при последовательном расширении онтологии [89]:

$$n \rightarrow Nmax. \quad (2.6)$$

2.2. Разработка онтологической модели

Прикладная онтологическая модель предназначена для представления понятий, необходимых для описания процессов научной и методической деятельности организации, а также описания тематик конкретных научных исследований, дисциплин и непосредственно самих информационных ресурсов.

При решении задач системы присутствует некоторая специфика разрабатываемой онтологии. Во-первых, вопрос ручного заполнения разрабатываемой онтологии представляется крайне сложным и длительным. Во-вторых, для обеспечения языковой компетентности, достаточной для построения онтологии предметной области на базе автоматического разбора текстовых корпусов, онтология должна обладать знаниями общими (языковыми) и специальными, относящимися к данной предметной области. Такая онтология должна, по сути, объединять в себе несколько: онтологию русского и английского языков и базовую онтологию предметной области. Например, для онтологии, разрабатываем для описания знаний студентов компьютерных специальностей, необходима терминологическая информация и на русском, и на английском языках, а также направленность на предметную область современных ИТ и компьютерной техники, и программирования, а не только на основные языковые словари.

После изучения предметной области в экспертном режиме, чтобы повысить непротиворечивость создания модели, был разработан каркас прикладной онтологической модели, ее условно статическая часть. Названа такая онтологическая модель «базовой». Затем «базовая» модель была откорректирована, улучшена и расширена за счет процедур автоматического и полуавтоматического пополнения знаний.

Таким образом, процесс построения онтологий – итерационный и состоит из определенных этапов, каждый из которых позволяет извлекать из текста знания (Рисунок 2.2).

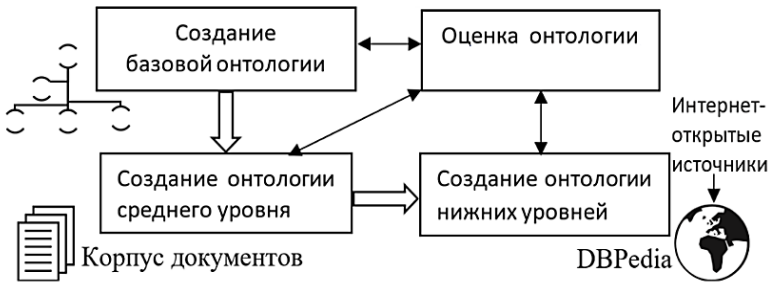


Рисунок 2.2 – Итерационный процесс создания онтологии

Одним из важных этапов является проверка экспертом построенной онтологии. При разработке онтологической модели использовался комбинированный способ формирования онтологии.

В общем разработанный подход к созданию онтологической модели следующий [90]:

1. Подбор и использование онтологий верхнего уровня для формирования «базовой» онтологии.

2. «Экспертное» создание таксономий, мета-тэгов и онтологий для создания каркаса «базовой» онтологии.

3. При создании онтологии среднего уровня использование для расширения онтологии знаний, полученных в результате автоматической обработки существующих корпусов документов.

4. Подбор и использование существующих глоссариев, тезаурусов для выбора имен понятий и населения экземплярами при автоматическом и полуавтоматическом формировании онтологий нижних уровней.

5. Получение знаний при автоматическом формировании онтологий нижних уровней из известных междоменных и тематических онтологий, а также из открытых источников Интернет.

6. Использование разработанной онтологии верхнего уровня для полуавтоматического или автоматического поиска и простой категоризации знаний нижних уровней.

2.2.1 Средства создания онтологии

Первым этапом в процессе проектировании онтологии был выбор соответствующего языка спецификации онтологий. В этом моменте мы руководствовались правилом, что предпочтительнее использовать языки, стандартизированные организацией W3C, например RDF – универсальный язык описания семантики ресурсов и взаимосвязей между ними. RDF использует базовую модель данных «объект-атрибут-значение» и представляет ресурсы в виде ориентированного размеченного графа. Все словари RDF используют базовую структуру, описывающую классы ресурсов и типы связей между ними. Это позволяет использовать разнородные децентрализованные словари, созданные для машинной обработки по разным принципам и методам. Модель схемы RDF включает наследование классов и свойств.

OWL – язык представления онтологий, расширяющий возможности XML, RDF, RDF Schema и DAML+OIL. Онтологии OWL – это последовательности аксиом и фактов, а также ссылок на другие онтологии. Они содержат компоненту для записи авторства и другой подробной информации, являются документами Web, на них можно ссылаться через URI.

Для разработки прикладной онтологии был сделан выбор в пользу OWL.

На следующем этапе было выбрано инструментальное средство для создания онтологических моделей – онтологический редактор. Редактор онтологий должен поддерживать работу с OWL-моделями и реализовывать наилучшим образом импорт-экспорт различных форматов.

Для создания прикладной онтологической модели рассматриваемой предметной области был выбран редактор Protege, так как это свободно распространяемое программное средство, позволяющее создавать и редактировать онтологии, а также экспортировать их во множество форматов таких, как RDF, OWL, XML и другие. Редактор имеет графический интерфейс, который позволяет разработчикам онтологий концентрироваться на концептуальных терминах, не думая о синтаксисе языка вывода. Protege обладает гибкой моделью знаний и

расширяемой архитектурой плагинов. Также Protege дает дополнительные возможности по анализу онтологии, предоставляя совместимость с машинами логического вывода, так называемыми резонерами от различных производителей (Pellet, RacerPro, Fact++, Hermit).

2.2.2 Использование онтологий верхнего уровня при разработке онтологической модели

Разработка онтологий для конкретной предметной области является сложным и трудоемким процессом. Поэтому целесообразно будет использовать существующие онтологии верхнего уровня для подобной или близкой предметной области.

На сегодняшний день разработано несколько онтологий для описания научных публикаций и процесса исследовательской деятельности: BIBO, комплекс онтологий SPAR, CERIF, SWRC, EXPO, FRBR, SKOS, Dublin Core, ЕНИП и др.

В приложении А приведено описание популярных существующих онтологий верхнего уровня и широких междоменных онтологий (Таблица А.1).

В разработанной онтологической модели также были использованы следующие онтологии: Dublin Core [91], FOAF [92], VIVO [93, 94], BIBO [95], DBpedia ontology [96], TEACH [97], VCard [98].

В первую очередь, как наиболее подходящая к предметной области, была выбрана онтология верхнего уровня «Дублинское ядро» (Dublin Core). Расширенный набор метаданных Dublin Core содержит 33 поля, соответствует ISO 158362003 и ГОСТ Р 7.0.10-2010. Рассмотрим некоторые классы:

- BibliographicResource: книга, статья или другой документальный ресурс;
- FileFormat: формат файла цифровых ресурсов;
- MediaType: тип файла или физического носителя;
- MediaTypeOrExtent: тип или экстент носителя;
- PhysicalMedium: физический материал или носитель (бумага, холст или DVD);

– PhysicalResource: материальная вещь;
– SizeOrDuration: измерение или экстент, или время, необходимое для воспроизведения или исполнения (количество страниц, указание длины, ширины или период в часах, минутах и секундах).

Граф выбранных классов приведен на рисунке 2.3, их объектные свойства на рисунке 2.4.

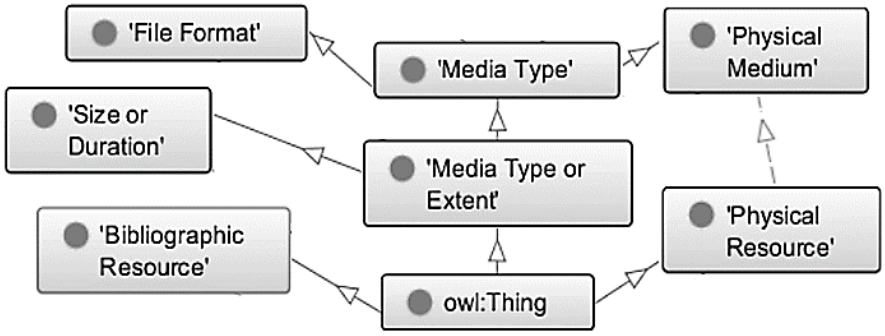


Рисунок 2.3 – Граф Dublin Core для интеграции с моделью системы

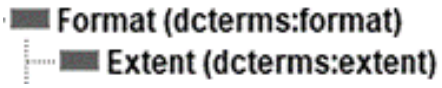


Рисунок 2.4 – Объектные свойства для выбранных классов Dublin Core

Рассмотрим еще одну «top-level» онтологию FOAF:

– Agent: класс агентов; некая абстракция – хорошо известный подкласс, представляющий людей, организацию или группу;
– Group: представляет собой набор отдельных агентов (сам может также играть роль агента);
– Organization: представляет собой своего рода агента, соответствующего социальным институтам, таким как компании, общества и т. д.;
– Person: представляет людей. Класс Person является подклассом класса Agent, так как все люди считаются «агентами» в FOAF;
– OnlineAccount: класс, описывающий предоставление какой-либо формы онлайн-услуги какой-либо стороной для какого-либо Агента.

Граф выбранных классов из онтологии FOAF представлен на рисунке 2.5, а выбранные свойства классов онтологии FOAF на рисунке 2.6.

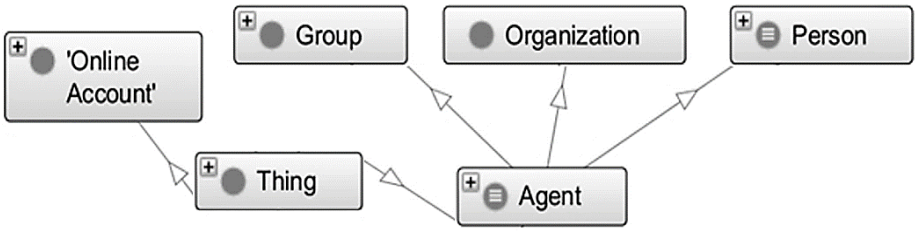


Рисунок 2.5 – Граф FOAF для интеграции с моделью системы

Рисунок 2.6 – Объектные свойства для выбранных классов FOAF

Таким образом, онтология верхнего уровня Dublin Core подходит при разработке формата метаданных, определяющих состав и семантику элементов данных для использования в системах обработки ИР научно-образовательного учреждения, часть информации о персоне и деятельности будем использовать из онтологии FOAF.

Эти онтологии были адаптированы к предметной области профессиональной деятельности сотрудников научно-образовательного учреждения: удалены не используемые классы и свойства и подкорректированы описания классов для дальнейшей интеграции. Классы других онтологий описаны непосредственно при создании онтологической модели.

2.2.3 Разработка «базовой» онтологической модели

Модульный принцип построения значительно облегчает создание онтологий, имеющих пересекающиеся множества понятий и отношений из их предметной области, таким образом, было решено разработать следующие модули онтологии: обобщенная модель «Онтомодель», модель «Научные мероприятия», модель

«Академические звания и должности», модель «Персоналии», модель «Научно-образовательное учреждение», модель «Информационный ресурс», модель «Документ», модель «Образовательный ресурс».

При построении каждой модели необходимо выполнить следующие этапы: определить классы онтологии, расположить классы в таксономическую иерархию, определить объектные свойства (отношения), а также свойства данных (атрибуты), заполнить значения экземпляров [99].

На рисунке 2.7 приведена обобщенная «Онтомодель», которая позволяет описать основные классы предметной области (Таблица 2.1).

$$Ontomodel = <Addr, Cond, Org, Eve, Profile, Themes>. \quad (2.7)$$

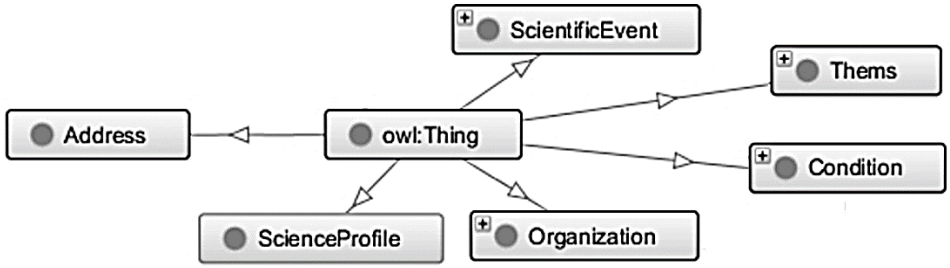


Рисунок 2.7 – Обобщенная «Онтомодель», разработанная в Protege

Таблица 2.1 Классы обобщенной онтомодели

Параметр	Имя класса	Класс в онтологии	Описание класса
Addr	Address	vCard: Address	Адрес
Cond	Condition		Академические звания и должности
Org	Organization	foaf: Organization	Научно - образовательное учреждение
Eve	ScientificEvent		Научные мероприятия
Profile	ScienceProfile		Пользователи системы
Thems	Thems	Vivo: ReseachArea	Тематика ИР

На рисунке 2.8 приведена модель первого уровня иерархии «Научные мероприятия», которая позволяет описать научные мероприятия и события в деятельности организаций (Таблица 2.2).

$$Eve = <Dipp, Pres, Olym, Conf>. \quad (2.8)$$

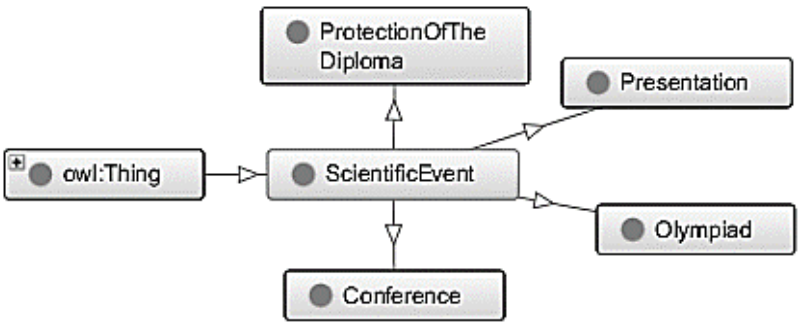


Рисунок 2.8 – Модель «Научные мероприятия», разработанная в Protege

Таблица 2.2 – Классы модели «Научные мероприятия»

Параметр	Имя класса	Класс в онтологии	Описание класса
Dipp	ProtectionOfTheDiploma		Защита диплома
Pres	Presentation		Презентация
Olym	Olympiad		Олимпиада
Conf	Conference	bibo: Conference	Конференция

На рисунке 2.9 приведена модель первого уровня иерархии «Академические звания и должности», которая позволяет описать академические звания и должности сотрудников организаций (Таблица 2.3).

$$Cond = <State, Degree, Rank, NonAcad, Pos>. \quad (2.9)$$

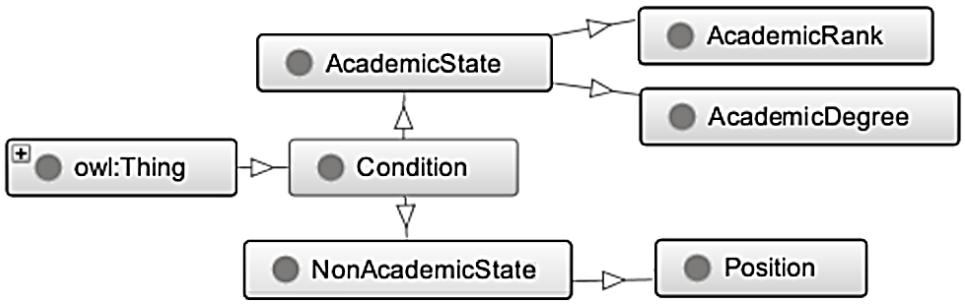


Рисунок 2.9 – Модель «Академические звания и должности», разработанная в Protege

Таблица 2.3 – Классы модели «Академические звания и должности»

Параметр	Имя класса	Класс в онтологии	Описание класса
State	AcademicState	dbo: AcademicState	Академический статус
Degree	AcademicDegree	dbo: AcademicDegree	Ученая степень
Rank	AcademicRank	dbo: AcademicRank	Академическое звание
NonAcad	NonAcademicState		Неакадемический статус
Pos	Position		Должность

На рисунке 2.10 приведена модель первого уровня иерархии «Научно-образовательное учреждение», которая позволяет описать учреждения и организации, занимающиеся научно-образовательной деятельностью или связанными с ними информационными процессами (Таблица 2.4).

Org = <Comp, HEdu, MEdu, Univer, Lab, Mus, Lib, Inst, (2.10)
Fac, Dep, Person, Gr, DepDocs>.

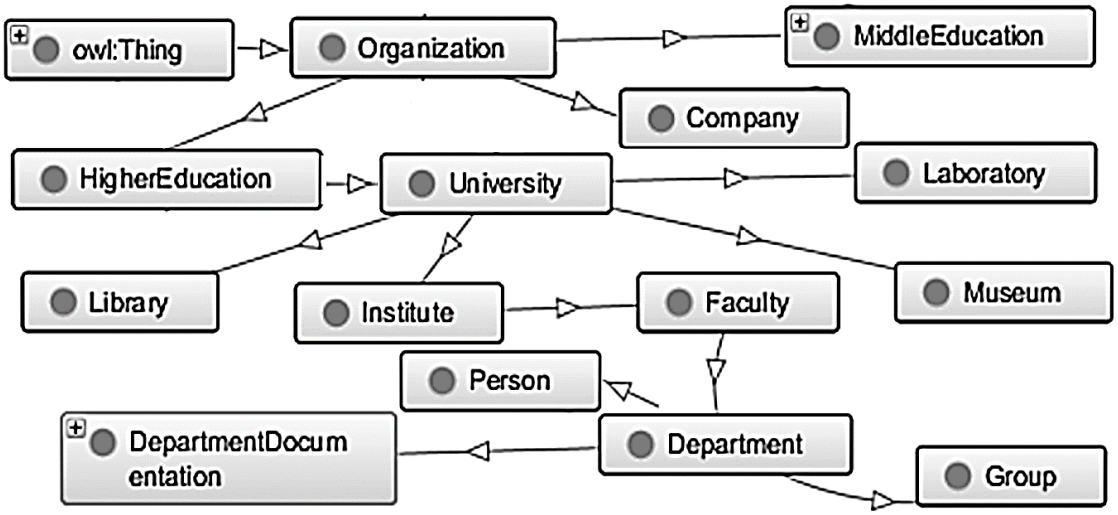


Рисунок 2.10 – Модель «Научно-образовательное учреждение», разработанная в Protégé

Таблица 2.4 – Классы модели «Научно-образовательное учреждение»

Параметр	Имя класса	Класс в онтологии	Описание класса
Comp	Company		Предприятие
HEdu	HigherEducation		Высшее учебное заведение
Inst	Institute		Институт
Univer	University		Университет
Lab	Laboratory		Лаборатория
Lib	Library	vivo: Library	Библиотека
MEdu	MiddleEducation		Среднее учебное заведение
DepDoc	DepartmentDocumentation		Документация
Mus	Museum	vivo: Museum	Музей
Dep	Department	vivo: Department	Кафедра
Fac	Faculty		Факультет
Gr	Group		Группа студенческая
Person	Person	foaf: Person	Сотрудники

На рисунке 2.11 приведена модель нижнего уровня иерархии «Персоналии», которая позволяет описать иерархию сотрудников, пользователей и персонала, участвующих в деятельности организаций (Таблица 2.5).

Person = <St, Emp, Teach, UnSt, GrSt, PersDoc, IndPl, PerProt, LoadLog>. (2.11)

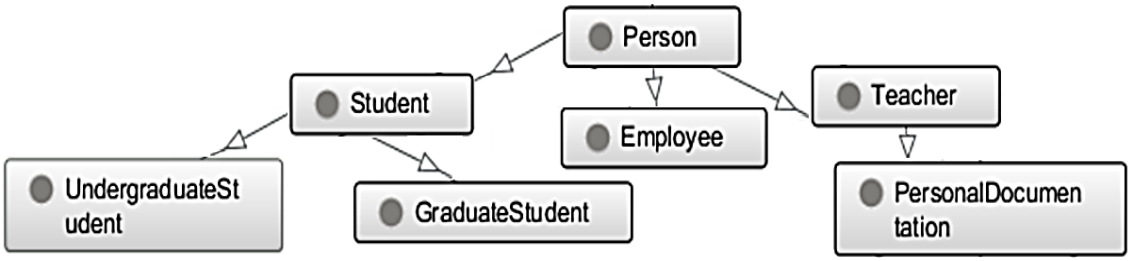


Рисунок 2.11 – Модель «Персоналии», разработанная в Protege

Таблица 2.5 – Классы модели «Персоналии»

Параметр	Имя класса	Класс в онтологии	Описание класса
Teach	Teacher		Преподаватель
Emp	Employee		Сотрудник
St	Student	vivo: Student	Студент
GrSt	Graduate Student	vivo: Graduate Student	Бакалавр (bachelor's degree), работает над степенью магистра
UnSt	UndergraduateStu dent	vivo: Undergraduate Student	Учащийся, еще не получивший диплома бакалавра
PersDoc	PersonalDoc		Персональная документация

Важную часть контента составляют описания релевантных областей знаний системы информационных ресурсов, представленных в Интернет, и описание документов, хранящихся в локальной (текстовой) базе данных системы.

На рисунке 2.12 приведена модель нижнего уровня иерархии «Информационный ресурс», которая позволяет описать различные информационные ресурсы в деятельности организаций:

IR = < ER, AU, DSet, Art, AcArt, ConPap, EduArt, RewArt, Doc, (2.12)
Im, Soft, URL, V, W, MR, EduR>.

При управлении информационными ресурсами контент представляется в виде множества связанных информационных ресурсов. Формально, каждый ИР соответствует некоторому понятию онтологии (является экземпляром какого-либо класса онтологии) и имеет заданную структуру (Таблица 2.6).

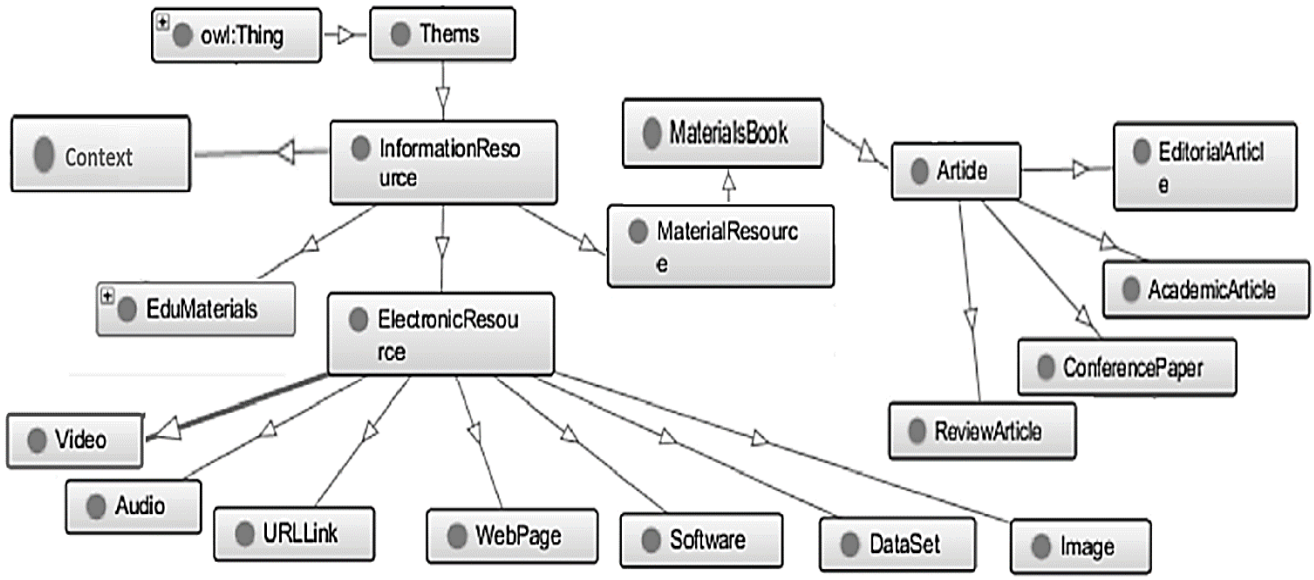


Рисунок 2.12 – Модель «Информационный ресурс», разработанная в Protege

Таблица 2.6 – Классы модели «Информационный ресурс»

Параметр	Имя класса	Класс в онтологии	Описание класса
Context	Context		Контекстное представл
ER	ElectronicResource		Электронный ресурс
IR	Information Resource	vivo: Information Resource	Абстракция ИР
AU	Audio	bibo: AudioDocument	Аудиофайл
DSet	DataSet	vivo: Dataset	Данные
Im	Image	vivo: Image	Изображение
Soft	Software	vivo: Software	ПО
URL	URLLink	vivo: URLlink	Ссылка URL
V	Video		Видеофайл
W	WebPage	bibo: WebPage	Интернет страница
MR	MaterialResource		Материальный ресурс
EduR	EduMaterial		Образоват-й ресурс
Art	Article	bibo: Article	Статья
AcArt	AcademicArticle	bibo: AcademicArticle	Академическая статья
ConPap	ConferencePaper	vivo: ConferencePaper	Статья на конференции
EduArt	EditorialArticle	vivo: EditorialArticle	Статья научная
RewArt	ReviewArticle	vivo: ReviewArticle	Обзорная статья
MatB	MaterialsBook	bibo: Collection	Сборник

Между конкретными информационными ресурсами могут существовать связи (элементы класса), синтаксис и семантика которых определяется отношениями, заданными между соответствующими понятиями онтологии.

Подобную структуру имеет описание любого ИР. Содержательно такие описания включают, с одной стороны, общую информацию (метаинформацию) о

ресурсе или документе, а с другой – представляют содержание (контент) документа в терминах онтологии области знаний ИС.

На рисунке 2.13 приведена модель нижнего уровня иерархии «Документ», которая позволяет описать различные виды документов в деятельности организаций (Таблица 2.7).

Doc= < Ab, PartDoc, UDC, Cont, Ref, Annotation, Author, KW, Link, Par, Th, Title, MatB>. (2.13)

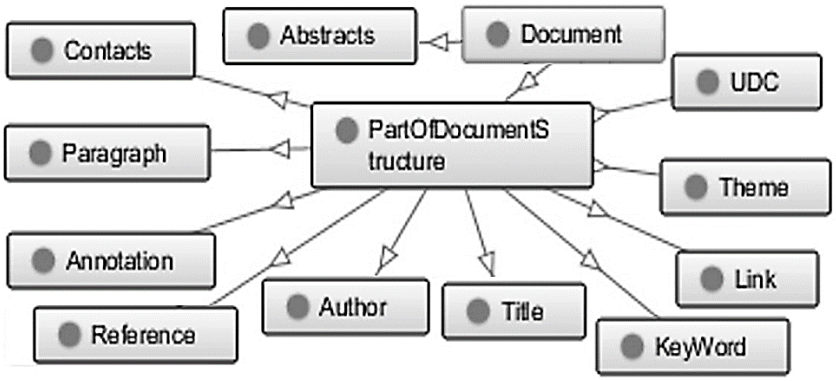


Рисунок 2.13 – Модель «Документ», разработанная в Protege

Таблица 2.7 – Классы модели «Документ»

Параметр	Имя класса	Класс в онтологии	Описание класса
Ab	Abstracts	bibo: Thesis	Тезисы
PartDoc	PartOfDocumentStructute	bibo: DocumentPart	Структура документа по частям
UDC	UDC		УДК
Cont	Contacts		Контакты
Ref	Reference	bibo: ReferenceSource	Единица списка
Annotation	Annotation	dbo: Abstract	Аннотация
Author	Author		Автор
KW	KeyWord		Ключевое слово
Link	Link		Ссылка
Par	Paragraph		Параграф
Th	Theme		Тема документа
Title	Title		Заголовок

На рисунке 2.14 приведена модель нижнего уровня иерархии «Образовательный ресурс», которая позволяет описать ИР и события образовательной дятельности организаций (Таблица 2.8).

EduR = < Boo, Lec, Man, Per, Mag, NewP, Rep, GrRep, LabRep, Doc,
PrRep, StResRep, TestRep, Slide, ManLR, ManKR, ManSR, Disc >. (2.14)

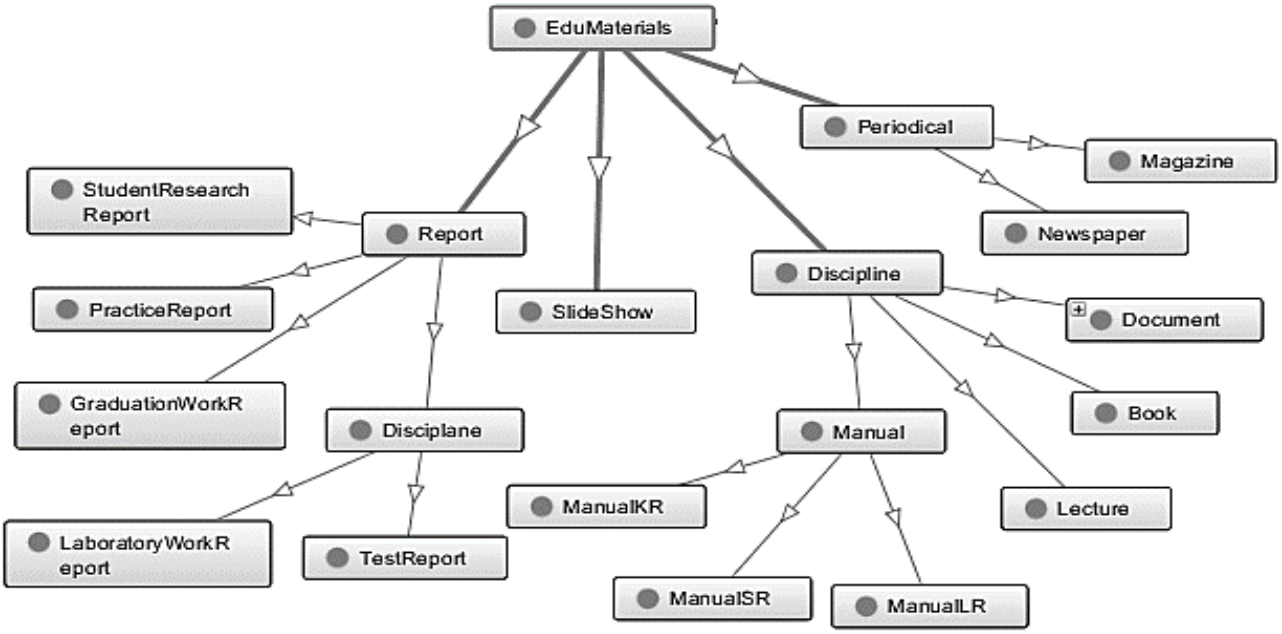


Рисунок 2.14 – Модель «Образовательный ресурс», разработанная в Protege

Таблица 2.8 – Классы модели «Образовательный ресурс»

Параметр	Имя класса	Класс в онтологии	Описание класса
Boo	Book	bibo: Book	Книга
Lec	Lecture		Лекция
Man	Manual	bibo: Manual	Методичка
Per	Periodical	bibo: periodicl	Периодическое издание
Mag	Magazine	bibo: magazine	Журнал
NewP	Newspaper	bibo: journal	Газета
Rep	Report	bibo: Report	Отчет
GrRep	GraduationWorkReport		Отчет по ВКР
LabRep	LaboratoryWorkReport		Отчет по ЛР
PrRep	PracticeReport		Отчет по практике
StResRep	StudentResearchReport		Отчет по НИРС
TestRep	TestReport		Отчет
Slide	SlideShow		Презентация
ManLR	ManualLR	bibo: Manual	Методичка по л.р
ManKR	ManualKR	bibo: Manual	Методичка по курсовой
ManSR	ManualSR	bibo: Manual	Методичка по с.р
Disc	Discipline		Дисциплина
Doc	Document	bibo: Document	Абстракция Документ

Определили основные классы и построили «каркас» онтологической модели, включающий только классы иерархии (Рисунок 2.15). Типы отношений между объектами приведены на рисунке 2.16. Описание классов, атрибутов представлено в приложении Б.

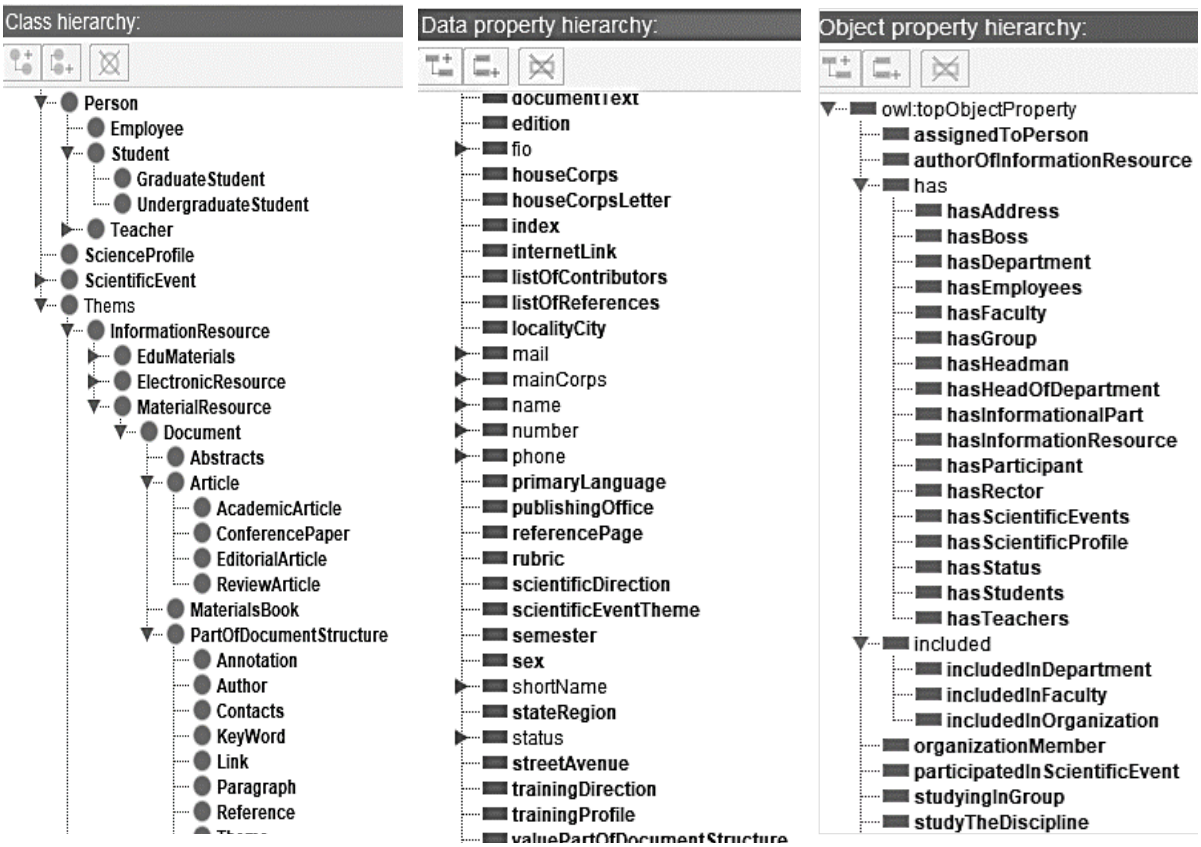


Рисунок 2.15 – Иерархия классов, объектные свойства, свойства данных

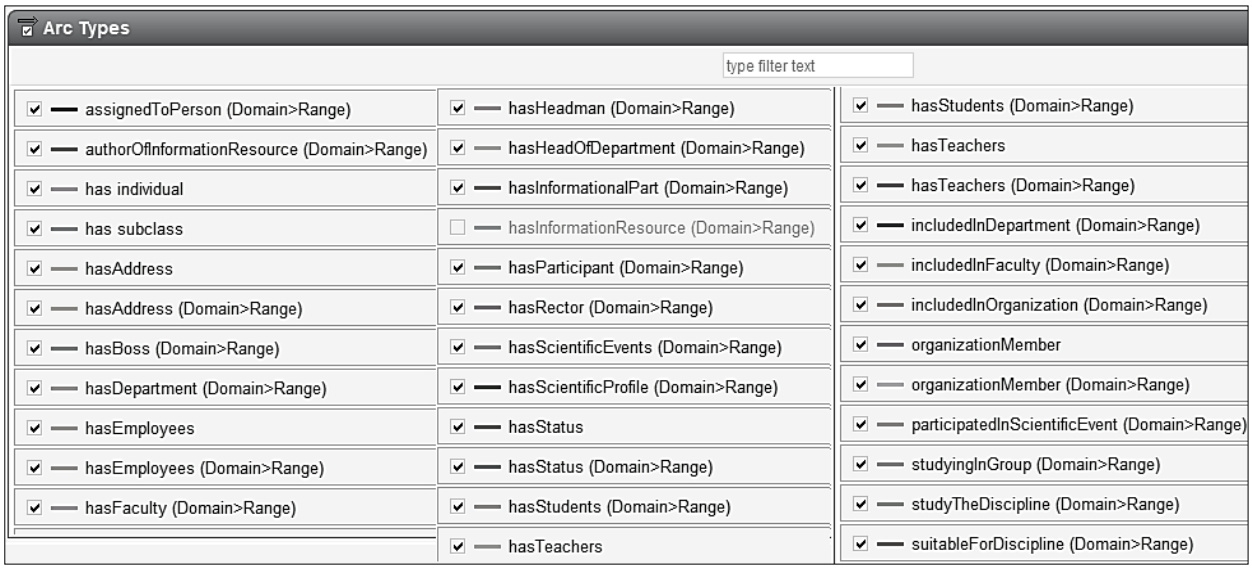


Рисунок 2.16 –Типы отношений онтологической модели

2.2.4 Проверка онтологии на согласованность

Для проверки корректной работы полученной онтологии сначала выполнили несколько запросов SPARQL, результаты работы которых представлены на рисунках 2.17–2.19.

1. Вывести все подклассы и их родительские классы.

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?subclass ?superclass WHERE { ?subclass rdfs:subClassOf ?superclass }

Результат:

subclass	superclass
IndividualPlan	PersonalDocumentation
Institute	HigherEducation
FacultyReport	DepartmentDocumentation
Contacts	PartOfDocumentStructure
Employee	Person
AdvertisingBooklets	DepartmentDocumentation
MiddleEducation	Organization
Teacher	Person

Рисунок 2.17 – Список всех подклассов и их родительских классов

2. Список информационных ресурсов, которые были задействованы в event2.

PREFIX usedInEvent: <http://www.semanticweb.org/valer/ontologies/2020/6/untitled-ontology-18#usedInEvent>

SELECT ?infRes WHERE { ?infRes usedInEvent: <http://www.semanticweb.org/valer/ontologies/2020/6/untitled-ontology-18#event2> }

Результат:

infRes
article2

Рисунок 2.18 – Список ИР, которые были задействованы в event2

3. Список информационных ресурсов, которые будут полезны при изучении информатики.

PREFIX suitable: <http://www.semanticweb.org/valer/ontologies/2020/6/untitled-ontology-18#suitableForDiscipline>

SELECT ?infRes WHERE { ?infRes suitable: <http://www.semanticweb.org/valer/ontologies/2020/6/untitled-ontology-18#informatics>}

Результат:

infRes
article3
article2

Рисунок 2.19 – Список ИР по информатике

При проектировании любой онтологии важную роль играет вопрос непротиворечивости вводимых объектов и индивидов. OWL-онтология была проверена на согласованность с помощью логической машина вывода (reasoner).

После того как, классы, индивиды и свойства определены, получили автоматически выводимую иерархию. Результаты отработки резонера Hermit 1.4.3.456 представлены на рисунке 2.20.

```
INFO  12:49:37  ------------------------------ Running Reasoner ------------------------------
INFO  12:49:38  Pre-computing inferences:
INFO  12:49:38      - class hierarchy
INFO  12:49:38      - object property hierarchy
INFO  12:49:38      - data property hierarchy
INFO  12:49:38      - class assertions
INFO  12:49:38      - object property assertions
INFO  12:49:38      - same individuals
INFO  12:49:38  Ontologies processed in 1386 ms by HermiT
```

Рисунок 2.20 – Результаты работы резонера

Отметим, что различия в иерархии вручную построенных классов и выводимой отсутствуют, ошибки не определяются, следовательно, классы в онтологии корректны, онтологическая модель представлена в согласованном виде.

2.3 Разработка методов наполнения онтологической модели

Методы и средства наполнения онтологий были и есть самым слабым этапом в процессе проектирования систем управления знаниями. Поскольку создание онтологий вручную чрезвычайно трудоемко и время затратно, автоматизация этого процесса имеет большие практические перспективы. Без хорошо налаженной технологии наполнения онтологий невозможно решить проблему актуальности данных. Пополнение (создание) онтологии, или Ontology Learning – процесс автоматического или полуавтоматического создания онтологий, включающий извлечение терминов и отношений между понятиями, которые эти термины представляют, из корпуса текста на естественном языке.

При наполнении онтологической модели экземплярами использован комбинированный подход:

- пополнение прикладной онтологии на базе уже существующей онтологии, совпадающей по предметной области;
- пополнение онтологии с помощью шаблонизированных и семантически размеченных документов;
- автоматическое пополнение на базе словарных определений готовых словарей;
- подход, основанный на обработке произвольных источников с помощью NLP технологий с целью обнаружения контекстно-зависимых слов, относящихся к одной предметной области.

2.3.1 Способ пополнения онтологии с помощью шаблонизированных документов

Большая часть методов, предназначенных для извлечения заданного количества терминов любой длины из коллекции документов и не различающих вхождения одного термина, укладывается в общую схему. Согласно этой схеме, метод извлечения терминов состоит из трех перечисленных ниже этапов [100]:

1. Предобработка текстового документа и сбор кандидатов: фильтрация слов и словосочетаний, извлеченных из коллекции документов, по статистическим и лингвистическим критериям.

2. Подсчет признаков: перевод каждого кандидата в вектор признакового пространства (в простейшем случае – простая скалярная оценка). После этого может проводиться проверка – сравнение слов, которые могли бы быть емкими лексическими единицами, с эталонным списком ключевых слов.

3. Вывод на основе признаков: оценка вероятности быть термином для каждого кандидата на основе значений признаков, последующая сортировка всех кандидатов, по этой оценке, и взятие заранее определенного числа кандидатов.

Первый этап включает обработку, форматирование текстового документа и приведение его к формату, пригодному для дальнейшей обработки и распознавания. Например, производится лексический анализ и морфологический анализ. Лексический анализ разбивает текстовый массив на отдельные лексемы: основные и служебные, производится удаление стоп-слов. Стоп-слова – это слова, которые не несут никакой смысловой нагрузки (артикли, предлоги, союзы, частицы, местоимения, вводные слова, междометия и т. п.). В результате морфологического анализа могут быть найдены лексемы, представляющие собой разные формы одного и того же слова. Например, существительные, которые различаются только падежными окончаниями [101].

Для реализации процесса сбора кандидатов необходимо выполнить нескольких шагов. На первом шаге применяются лингвистические фильтры, цель которых – оставить только существительные и именные группы, то есть словосочетания с существительным в роли главного слова, которые чаще всего и используются для наименования понятий. Для этого применяется либо поверхностный синтаксический разбор (который может вносить дополнительный шум), либо не использующая синтаксической информации фильтрация цепочек слов по частям речи по предопределенным правилам. На последующих шагах сбора кандидатов с целью снижения шума производится дополнительная фильтрация:

- по частоте: как правило, исключаются из рассмотрения кандидаты с числом вхождений меньше 2 или 3, так как при этом становятся неприменимы многие статистические признаки;
- по содержанию в составе цепочки-кандидата так называемых стоп-слов из заранее составленного списка (многие слова, такие как «хороший» или «интересный», очень редко входят в состав терминов, при этом могут достаточно часто образовывать именные группы – например, «хороший метод»);
- по длине слов в составе кандидата или содержанию в них особых символов: часто исключаются из рассмотрения неалфавитные символы и слова из одной-двух букв.

Распознавание новых концептов – следующий этап после нахождения подходящих терминов в тексте предметной области. Обычно найденные термины делят на два класса – те, которые выражают существующие концепты, и относящиеся к новым концептам. В случае небольших баз знаний фиксированных предметных областей, обычно считают, что все специфичные термины несут в себе новые концепты.

В полном объеме этот этап формирования онтологии является довольно сложной и длительной задачей и требует значительных затрат по реализации. Для получения практических результатов в приемлемые сроки задача была сужена. Была организована работа в полуавтоматическом режиме, а не только в автоматическом, с возможностью редактирования онтологии. Найденные термины считались однозначно определенными концептами и не рассматривались вопросы интеграции онтологий. Все спорные вопросы решались с помощью применения экспертного режима.

Модуль автоматического разбора типовых документов обрабатывал такие типовые документы, как ВКР, научные статьи факультетских сборников и журналов, отчеты по курсовым работам, рефераты и т.п.

В первую очередь была решена задача извлечения текстовой информации из файлов различных форматов, таких как doc, pdf, docx, rtf. Далее произведен парсинг

текста с помощью регулярных выражений PCRE — библиотеки PHP, реализующий работу регулярных выражений в стиле Perl, синтаксис которых значительно более мощный и гибкий, чем регулярных выражений POSIX.

Реализованы основные поисковые выражения:

1. Регулярное выражение для поиска ФИО:

@([А-Я][а-яё]+\s[А-Я]\.[\s]?[А-Я]?[\,.\s]?|[А-Я]\.[А-Я]\.[\s]?[А-Я][а-яёя]+)@u.

2. Регулярное выражение для поиска дат:

@[“”\s]\d{1,2}[”””\s]\s?[а-я]+\s\d{4}@u.

3. Регулярное выражение для поиска ключевых слов:

@[Кк]лючевые слова[:][\s]?[а-яА-яёЁ,;\-\s]+@u.

4. Регулярное выражение для поиска интернет-ресурсов:

@(\n.+?URL.+?pdf)|(.+?Режим доступа:\s?.*?www.URL.+?\.(com|ru))|(.*?Интернет-ресурс.*?(com|ru))@u.

При извлечении из текста неструктурированных информации использована библиотека морфологического анализа PHPMorphy.

На рисунке 2.21 изображен фрагмент программного окна, демонстрирующий результат извлечения ключевых слов из текста статьи, а также результат применения регулярного выражения к поиску ФИО. В случае ошибочного извлечения возможно вручную удалить слово из набора.

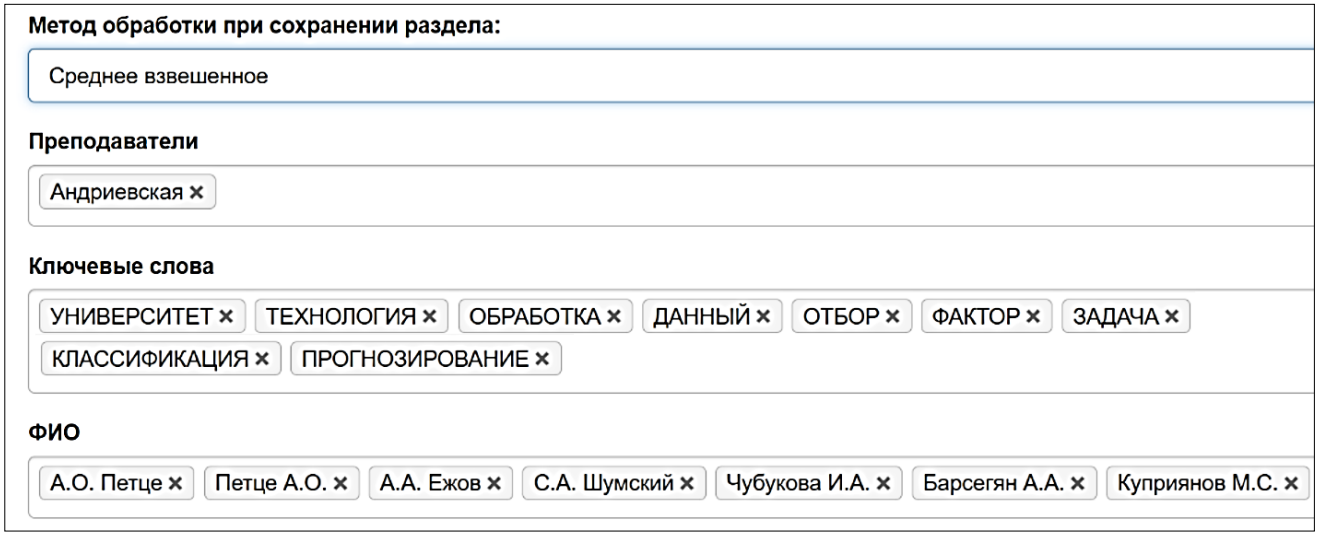


Рисунок 2.21 – Результаты работы регулярного выражения для поиска ФИО

2.3.2 Способ полуавтоматического пополнения на базе словарных определений готовых словарей

Важным принципом успешного функционирования систем, построенных с помощью онтологического подхода, является наполненность онтологической модели. Автоматические и полуавтоматические способы пополнения онтологий значительно повышают скорость создания онтологий.

По тематике автоматического и полуавтоматического пополнения онтологий SOAT (Semi-automatic domain Ontology Acquisition Tool) выполнено значительное количество исследований и написано немало работ. Задача автоматического и полуавтоматического пополнения онтологий на основе уже существующих знаний находится сегодня в фокусе внимания, что говорит одновременно и о ее актуальности, и о ее нерешенности в целом.

Одним из самых распространенных способов создания онтологий и ее пополнения является способ, базирующийся на использовании словарных концептов из электронных словарей, тезаурусов.

Выполнялся поиск слов по различным ключам в тестовых фрагментах текстов, наполненных множеством слов по тематике информационных технологий, программирования и вычислительной техники.

Целесообразность использования словаря оценивалась следующим образом: выполнен поиск слов по ключу в текстах; рассчитаны показатели эффективности поиска.

При проверке эффективности работы словаря оценивались следующие показатели:

- количество найденных слов;
- релевантность найденных слов;
- скорость поиска;
- кривая «полнота-точность»;
- наличие API.

Так как значения коэффициентов полноты и точности определись однозначно для каждого из запросов пользователей, то это позволило вычислить средние значения для фиксированных интервалов и построить кривую «полнота-точность», которая использовалась для оценки качества алгоритма поиска.

Результаты исследований по определению эффективности и целесообразности использования для пополнения онтологии приведены для наиболее популярных словарей, таких как, WordNet, Wikipedia, Wiktionary [102].

Наиболее часто в работах исследователей по данному вопросу встречались рекомендации по использованию словаря WordNet, разработанного в Princeton University. Словарь содержит 4 семантические сети для описания существительных, глаголов, прилагательных и наречий. Единицей хранения в тезаурусе является синсет – набор синонимов, связанных между собой различными семантическими отношениями. Имеет возможность задавать предметную область. WordNet имеет API, но он способен работать лишь с английским языком. Одним из больших преимуществ является возможность быстрого поиска семантических отношений (прямые гипонимы, гиперонимы, синонимы и др.) и лексических отношений [103].

Фрагмент текста с выделением найденных слов по ключу IT, полученный в результате работы программного модуля со словарем WordNet, приведен на рисунке 2.22. Найдено на фрагменте 15 слов, не найдено – 6 слов (MySQL, PostgreSQL, SQLite, MacOS, iOS, Objective-C). Среднее арифметическое время выполнения скрипта: 3,497 секунд.

Let us say you want to save precious bandwidth and develop locally. In this case, you will want to install a web server, such as **Apache**, and of course **PHP**. If you need you can also use **Nginx**. You will most likely want to install a database as well, such as MySQL, PostgreSQL or SQLite after all all these ones use **SQL**. It is easy to setup a web server with PHP support on any operating system, including MacOS, Linux and **Windows**. For developing on Android platform from **Google**, firstly, you need knowledge of **Java** or **Kotlin**. And for iOS platform from Apple you probably need to know **Swift** or older Objective-C. Of course, in addition to the above, you should know the principles of **SOLID** and some patterns. Knowledge of internet protocols such as **DHCP**, **IP**, **TCP**, **HTTP**, **UDP** e.t.c will not be superfluous.

Рисунок 2.22 – Фрагмент с результатами работы модуля для WordNet

Результаты поисковых запросов для словаря WordNet сведены в таблицу 2.9, а результаты расчетов параметров качества поиска в таблицу 2.10. Кривая «полнота-точность» приведена на рисунке 2.23. Анализ качества поиска показал, что в целом по коллекции качество поиска, выраженное F-мерой, удовлетворительное (0.69), полнота поиска достигает приемлемого значения (0.86), но точность поиска при этом невелика (0.58).

Таблица 2.9 – Результаты поисковых запросов для WordNet

№	Ключ	WordNet	Всего слов в коллекции
1	operating system	Windows	Linux, Windows, Android, MacOS, iOS
2	computer languages, programming language	Java, Kotlin, Swift	PHP, SQL, Java, Kotlin, Swift, Objective-C
3	server	Apache, Nginx, HTTP	Apache, Nginx
4	protocol	DHCP, IP, TCP, SOLID, UDP	DHCP, IP, TCP, HTTP, UDP
..	..	..	..
20	database	SQL	SQL, MySQL, PostgreSQL, SQLite

Таблица 2.10 – Расчет параметров качества поиска для WordNet

№	TP	TN	FP	FN	Recall (полнота)%	Precision (точность) %	F-мера
1	1	0	4	0	1,00	0,20	0,33
2	3	0	3	0	1,00	0,50	0,67
3	3	1	0	0	1,00	1,00	1,00
4	4	1	1	1	0,80	0,80	0,80
..	..	..	..	..	..	..	..
20	1	0	3	0	1,00	0,25	0,40
В среднем по коллекции					0,86	0,58	0,69

Для пополнения онтологии русскоязычной терминологией был использован RussNet – аналог WordNet. Основной единицей также является синсет, но существует его отличие от английского словаря, которое состоит в наличии дополнительных семантических связей. Словарь RussNet является оригинальным ресурсом в том смысле, что он не переводится с Принстонского WordNet, а создается как отдельный ресурс [104]. Поскольку русскоязычных ресурсов мало, то данный словарь использовался без подробного исследования его работы.

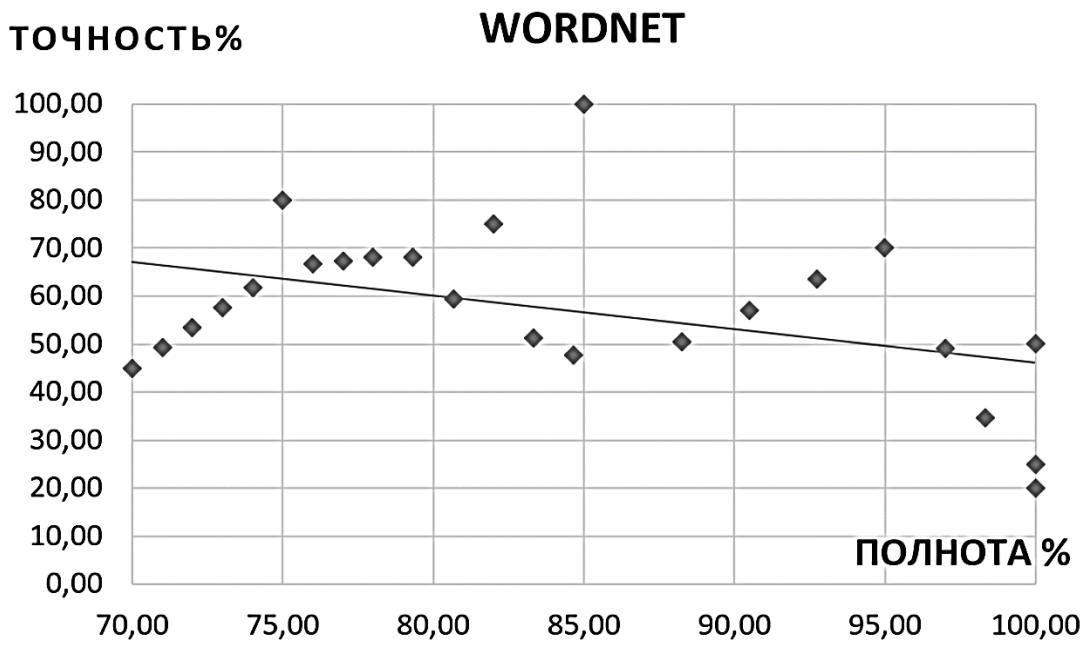


Рисунок 2.23 – Кривая «полнота-точность» для WordNet

Одним из источников онтологического описания предметных областей является семантическая Википедия. Таким образом, набирает популярность способ пополнения онтологии с использованием Википедии в качестве источника данных для извлечения знаний из систем.

Технология Wiki – современная интеллектуальная технология, позволяющая использовать модели знаний и выдавать поисковые запросы пользователям. MediaWiki является свободной программой и распространяется на условиях общественной лицензии GNU [105]. В Semantic MediaWiki [106] существует свой собственный встроенный инструмент для экспорта в RDF, позволяющий извлекать размеченную семантическую информацию из списка страниц [107].

При разработке программного модуля, использующего ресурс WikiPedia для пополнения предметной онтологии системы, были использованы следующие программные компоненты (библиотеки): библиотека Python WikipediaBot Framework [108], которая использует MediaWiki API (специальный интерфейс прикладного программирования) для взаимодействия с MediaWiki-системой для авторизации, получения данных и внесения изменений, и библиотека RDFLib [109], которая сохраняет в файл на языке OWL.

Частично результаты поиска помещены в таблицу 2.11, а результаты расчетов параметров качества поиска для словаря WikiPedia приведены в таблице 2.12.

Таблица 2.11 – Фрагменты запросов с использованием WikiPedia

Ключ	WikiPedia	Всего слов в коллекции
MySQL	MySQL, MySQL Cluster, MySQL AB, SQL injection, MySQLi, MySQL Archive, Michael Widenius	MySQL, MySQL AB, MySQLi, MySQL Archive, Michael Widenius, MySQL Workbench, List of Apache–MySQL–PHP packages, MySQL Cluster LAMP (software bundle)
сеть	Magnit, Heather Marsh, net, се́тка, Moscow City Telephone Network, Mado (food company), Kuzina (confectionery chain), Papa John's Pizza, mesh, grid, се́тка, сетево́й, KFC	net, network, се́тка, сетево́й, gigabit
SQL	SQL, Microsoft SQL Server, SQL injection, NoSQL, MySQL, History of Microsoft SQL Server, PostgreSQL, PL/SQL, Join (SQL), Transact-SQL	SQL, Microsoft SQL Server, SQL injection, NoSQL, MySQL, History of Microsoft SQL Server, PostgreSQL, PL/SQL, Join (SQL), Transact-SQL, Structured Query Language, ESQL, PL / SQL, Transact-SQL, T-SQL, SQLite

Таблица 2.12 – Расчет параметров для WikiPedia

№	TP	TN	FP	FN	Recall (полнота)%	Precision (точность) %	F-мера
1	6	1	3	0	1,00	0,67	0,80
2	13	9	1	4	0,76	0,93	0,84
3	10	6	0	0	1,00	1,00	1,00
4	10	0	1	1	0,91	0,91	0,91
..	..	..	..	..			
20	1	2	1	6	0,14	0,50	0,22
В среднем по коллекции					0,59	0,81	0,68

Кривая «полнота-точность» изображена на рисунке 2.24.

Анализ результатов тестирования WikiPedia показал, что при примерно одинаковых значениях F-меры по сравнению с тестами словаря WordNet, показатели полноты и точности отличаются. Таким образом, из WikiPedia при меньшем значении полноты происходит более точное извлечение знаний.

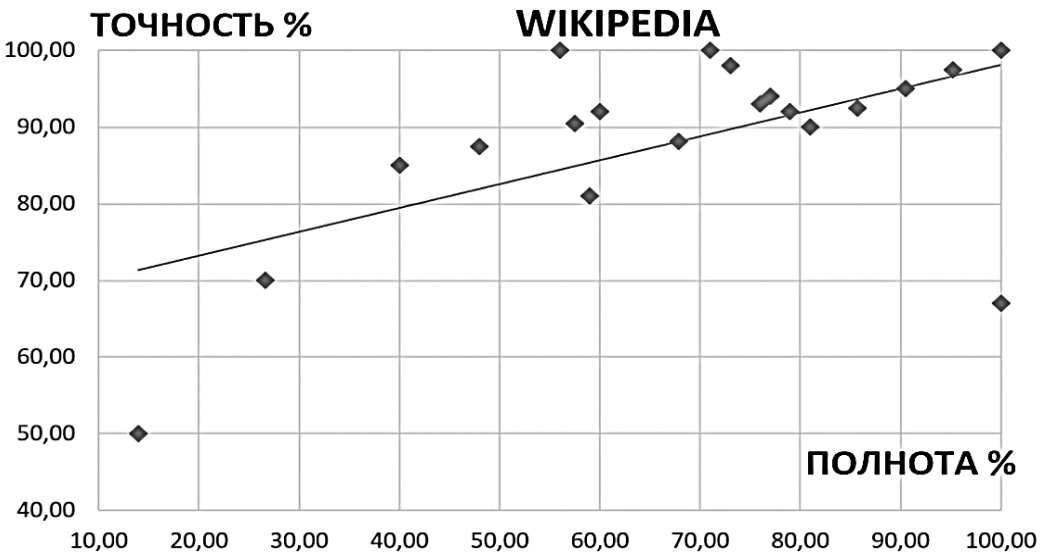


Рисунок 2.24 – Кривая «полнота-точность» для WikiPedia

Следует отметить, что провал в значениях полноты поиска наблюдается в основном для русскоязычных терминов. Для улучшения качества поиска был использован поиск по англоязычным терминам с последующим автоматическим переводом с помощью модуля Яндекс.Переводчик API [110]. Примеры показали, что собственные названия, например Siri, MacOS, как нам и требуется, не переводятся, следовательно можно применять поиск по англоязычным терминам с последующим автоматическим переводом для пополнения онтологий.

Полученные результаты приведены на рисунке 2.25.

```
Web  Перевод: Web
Web (programming system)  Перевод: Веб (система
Dark web  Перевод: Темная паутина
Web browser  Перевод: веб-браузер
Web mining  Перевод: Веб майнинг
Web page  Перевод: страница в Интернете
Web colors  Перевод: Веб-цвета
Web application  Перевод: веб приложение
Web server  Перевод: веб сервер
Website  Перевод: Веб-сайт
```

Рисунок 2.25 – Фрагмент результатов работы модуля перевода

Викисловарь Wiktionary [111] – это свободно пополняемый многофункциональный многоязычный онлайн словарь и тезаурус. Wiktionary, как и все проекты на движке MediaWiki имеет API. Среди особых отличий стоит выделить многоязычность платформы и единый API для всех Wiki-проектов.

Кроме этого, фонд Викимедиа регулярно публикует дампы каждого из своих проектов бесплатно. Эти дампы доступны в виде больших XML-файлов, которые можно импортировать в базу данных SQL с помощью специального программного обеспечения «MWDumper», что позволит локально использовать ресурс и значительно сократить время поиска по сравнению с Web версией.

Кроме словарных входов с толкованиями, семантическими отношениями и переводами, Wiktionary содержит словарные пометы. По сравнению с WordNet, наиболее близким аналогом среди электронных словарей, включающих словарные пометы и находящихся в открытом доступе, Wiktionary содержит около 370 помет против 170 WordNet [112]. Wiktionary превосходит WordNet по следующим параметрам: больший объем и более быстрое обновление материала (последнее особенно актуально); большее количество редакторов (сотни редакторов в Английском Викисловаре, десятки – в Русском Викисловаре).

Редакторами Викисловарей разработана система категорий словарных помет, призванная упорядочить и систематизировать словарные пометы. Наиболее детально на данный момент в Английском Викисловаре проработаны категории помет предметных областей, например topical приведен на рисунке 2.26.

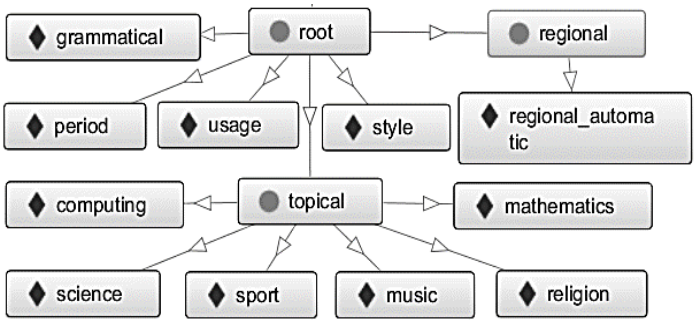


Рисунок 2.26 – Категории помет предметных областей для словаря Wiktionary

Результаты исследования возможности использования словаря Wiktionary для поиска терминов приведены в таблице 2.13, а результаты расчетов параметров качества поиска для словаря Wiktionary приведены в таблице 2.14. Кривая «полнота-точность» изображена на рисунке 2.27. Среднее арифметическое время выполнения скрипта в среде Интернет: 23,458 секунд. Среднее арифметическое

время выполнения скрипта при локально установленном словаре значительно уменьшается и составляет всего 2,8 секунды.

Таблица 2.13 – Результаты поисковых запросов с Wiktionary

№	Ключ	Wiktionary	Всего слов в коллекции
1	operating system	Linux, Windows, Android	Linux, Windows, Android, MacOS, iOS
2	computer languages	PHP, SQL, Java, Kotlin, Swift	PHP, SQL, Java, Kotlin, Swift, Objective-C
3	server	Apache, Nginx, HTTP	Apache, Nginx
4	protocol	DHCP, IP, TCP, HTTP, UDP	DHCP, IP, TCP, HTTP, UDP
..	..		..
20	database	MySQL, Postgre SQL, SQLite	MySQL, PostgreSQL, SQLite, SQL Linux

Таблица 2.14 – Расчет параметров для словаря Wiktionary

№	TP	TN	FP	FN	Recall (полнота)%	Precision (точность) %	F-мера
1	3	0	1	0	1,00	0,75	0,86
2	5	0	1	0	1,00	0,83	0,91
3	2	1	0	0	1,00	1,00	1,00
4	5	0	0	0	1,00	1,00	1,00
5	6	1	1	1	0,86	0,86	0,86
..	..	..	..	..	..	..	..
20	3	0	1	0	1,00	0,75	0,86
В среднем по коллекции					0,81	0,79	0,8

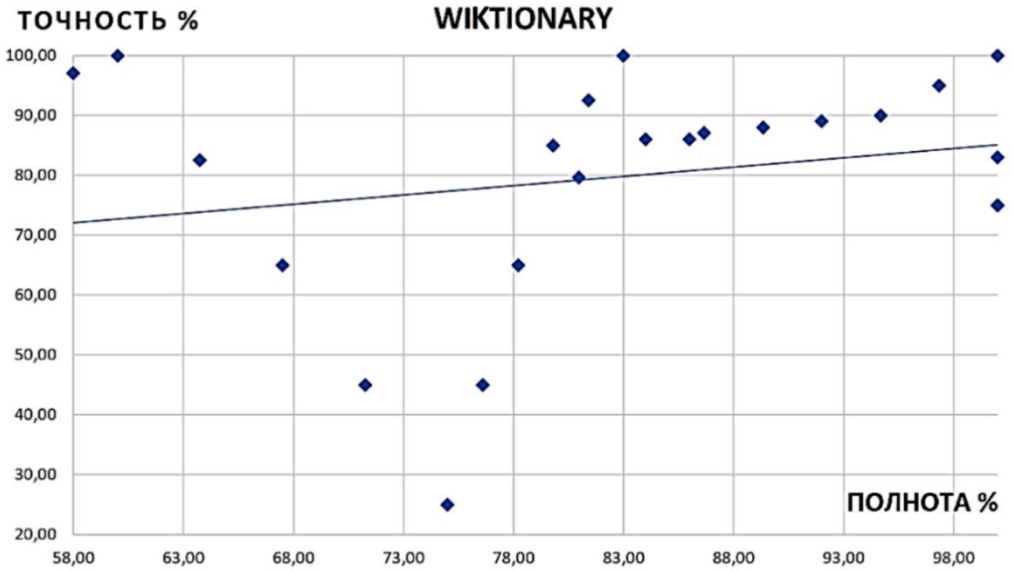


Рисунок 2.27 – Кривая «полнота-точность» для Wiktionary

Судя по параметрам, таким как, F-мера (0.8), полнота поиска (0.81) и форма кривой, результаты, возвращенные этим запросом, лучшего качества, чем предыдущие. Но тем не менее, есть провалы по определенным терминам, что уменьшает точность поиска (0.79).

Для запросов с уточнением предметной области поиска результаты поиска приведены в таблице 2.15. Результаты расчетов параметров качества поиска для случая с использованием словаря Wiktionary с уточнением предметной области поиска сведены в таблицу 2.16.

Кривая «полнота-точность» приведена на рисунке 2.28.

Таблица 2.15 – Результат поискового запроса для Wiktionary с уточнением предметной области поиска

К	Wiktionary	Всего слов в коллекции
SQL	SQL, Microsoft SQL Server, SQL injection, NoSQL, MySQL, History of Microsoft SQL Server, PostgreSQL, PL/SQL, Join (SQL), Transact-SQL	SQL, Microsoft SQL Server, SQL injection, MySQL, History of Microsoft SQL Server, PostgreSQL, PL/SQL, Join (SQL), Transact-SQL, Structured Query Language, ESQL, T-SQL, QL, HSQLDB
MySQL	MySQL, MySQL Workbench, MySQL Cluster, MySQL AB, LAMP (software bundle), SQL injection, MySQLi, MySQL Archive, List of Apache–MySQL–PHP packages, Michael Widenius	MySQL, MySQL Workbench, MySQL Cluster, MySQL AB, MySQLi, MySQL Archive, Michael Widenius, MariaDB
	-	-
сеть	net, mesh, grid, network, system, пу́ты, се́тка, сетевуха, се́тка, сетево́й	net, network, се́тка, сетевуха, се́тка, сетево́й

Таблица 2.16 – Расчет параметров для для Wiktionary с уточнением предметной области поиска

№	TP	TN	FP	FN	Recall (полнота)	Precision (точность)	F-мера
1	9	1	5	0	1,00	0,64	0,78
2	7	2	1	0	1,00	0,88	0,93
3	8	3	0	2	0,80	1,00	0,89
4	10	1	0	0	1,00	1,00	1,00
5	7	1	0	3	0,70	1,00	0,82
..	..	..	..	..			
20	6	4	0	2	0,75	1,00	0,86
В среднем по коллекции					0,82	0,92	0,87

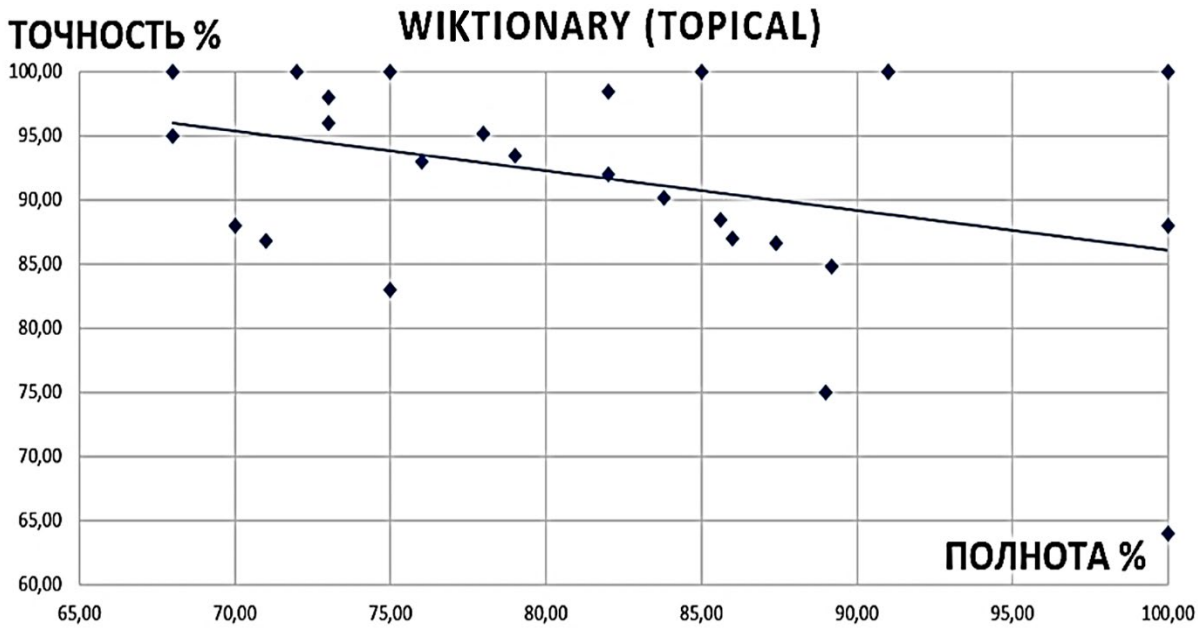


Рисунок 2.28 – Кривая «полнота-точность» для Wiktionary с уточнением предметной области поиска

Реализованный программный модуль использует следующие библиотеки [102]:

– cURL – библиотека функций, которая позволяет взаимодействовать с множеством различных серверов по различным протоколам (http, https, ftp, gopher, telnet и д.р.);
– PHPQuery – библиотека для парсинга, аналог jQuery на PHP;
– MediaWikiApi – веб-служба, обеспечивающая доступ к таким функциям Wiki, как аутентификация, операции над страницами и поиск по Wiki;
– PHP Text Analysis – это библиотека PHP для выполнения задач поиска информации (IR) и обработки естественного языка (NLP);
– phpMorphy – библиотека для морфологического анализа, реализованная на платформе PHP.

Фрагмент программного модуля, содержащего описание некоторых рабочих классов, приведен на рисунке 2.29.

```
class WikiSemantic implements ISemanticParsable
{   const TERMS_TAG = 'div.mw-parser-output ol li';
    private $parser;    private $wikiApi;
    public function __construct()
        {  $this->parser  = new PHPQueryParser();
           $this->wikiApi = new WikipediaApi();   }
    public function getTByW(string $word): string
{  return $this->parser->ParseText($this->wikiApi->
   GetWikiPage($word),self::TERMS_TAG); } }
```

```
class WordSemantic implements ISemanticParsable
{ private $wordnet_api;
  public function __construct()   {
        $this->wordnet_api = new WordNetApi();   }
  public function getTByW(string $word): string   {
         return $this->wordnet_api-
     >getSynsetsGloss ($word);   }
                                   }
```

Рисунок 2.29 – Фрагмент описания основных классов программного модуля

Исследования показали, что электронные словари-тезаурусы, такие, как WordNet и его русскоязычный аналог RussNet можно рассматривать как элементы для построения требуемой лингвистической онтологии.

Что касается использования словарей для пополнения предметной онтологии, то здесь преимущества имеют Wiki – ресурсы. Авторы проектов Wiki уделяют большое внимание доступности проекта на многих языках мира, в отличие от WordNet. Но если для реализации конкретной задачи необходимо проводить поиск и анализ специфичных тематических определений, то, как показали исследования, словарь Wiktionary лучше способен справиться с этой задачей. При сравнении кривых «полнота-точность» видим, что в случае использования словаря Wiktionary форма кривой «полнота-точность» более сглаженная, имеет необходимую форму и близка к идеальной.

Как показали эксперименты, словарь Wiktionary, для которого показатель качества поиска, выраженного F-мерой, достиг самого высокого среди тестируемых значения 0,87, обладает наибольшей терминологической полнотой для заданной предметной области и в среднем возвращает большее количество релевантных результатов.

2.3.3 Способ пополнения онтологии с использованием онтологии DBpedia

Большинство работ по тематике SOAT сосредоточены исключительно на захвате терминов и их отношений в онтологии. Они совершенно не фокусируются на населяющих их индивидах в пределах онтологии. Однако продукция знаний возможна только в том случае, если онтология содержит значительное число индивидов вместе с их отношениями.

Опираясь на подход, описанный в работе [113], будем использовать онтологию DBpedia, которая в последнее десятилетие развивалась быстрыми темпами благодаря усилиям открытого сообщества. DBpedia – это универсальная междоменная онтология, основанная на наиболее часто используемых информационных блоках в статьях Википедии, содержит более 685 классов, 2795 различных свойств и более 4,2 миллиона экземпляров [114].

На рисунке 2.30 представлены фрагмент онтологии DBpedia, содержащий необходимые понятия и их взаимосвязи для извлечения данных по предметной области, связанной с программным обеспечением.

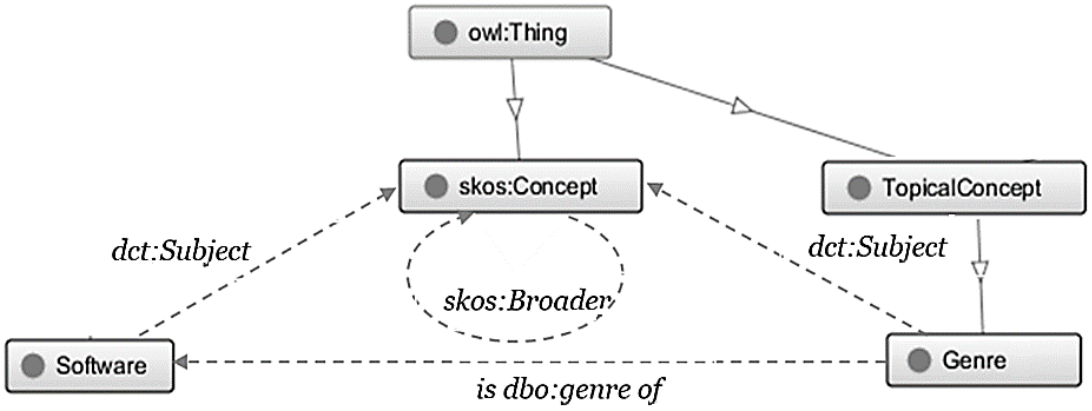


Рисунок 2.30 – Подмножество свойств и связей онтологии DBpedia

Для извлечения концептов были использованы следующие свойства: dbo: abstract, dct:subject, owl:sameAs, rdfs:comment. Эти свойства в онтологии DBpedia (dbo) являются производными от верхних онтологий, включая Dublin core (dct), Web Ontology Language (owl), Resource Description Framework Schema (rdfs) и Simple Knowledge Organization System (skos), на что указывают их префиксы.

Skos: Concept – фундаментальный элемент онтологии SKOS, позволяющий утверждать, что ресурс является концептом. Отношение (свойство объекта) R rdf: type C, указывает, что ресурс R является типом понятия C.

TopicalConcept – базовый класс онтологии DBpedia для тематических понятий.

Genre является подклассом тематического концепта и позволяет охватить жанр конкретного объекта. Экземпляры genre включают такие жанры, как, “relational database”, “graph database и др. В свою очередь genre имеет обратную связь с понятием “Software” через отношение "is dbo:genre of". Кроме того, genre может быть связан отношением с более высоким уровнем концепции dct:subject. Например, такие жанры, как “graph database" and “column-oriented database” связаны с понятием “database models”.

Software имеет два свойства объекта, а именно: dbo:genre и dct:subject. Например, программное обеспечение "Neo4j" относится к жанру” graph database". Аналогично, домен отношения dct: subject – это программное обеспечение, а его диапазон значений – skos:Concept. Например, "Neo4J" имеет отношение типа dct:subject с концептом "NoSQL".

Для извлечения самих данных из онтологии использовался SPARQL, который является рекомендованной практикой при публикации данных консорциума W3C и одной из технологий семантической паутины [115, 116].

Автоматическая обработка текста проводится двумя модулями: первый модуль выполняет автоматический разбор текста и его аннотирование (Рисунок 2.31), второй модуль извлекает подробную информацию по выбранному термину (Рисунок 2.32).

Для аннотирования текста с понятиями в онтологии DBpedia используем плагин с именем DBpedia Spotlight [117]. DBpedia Spotlight выполняет две основные задачи: распознавание фраз и устранение неоднозначности (Шаг 3 на рисунке 2.32).

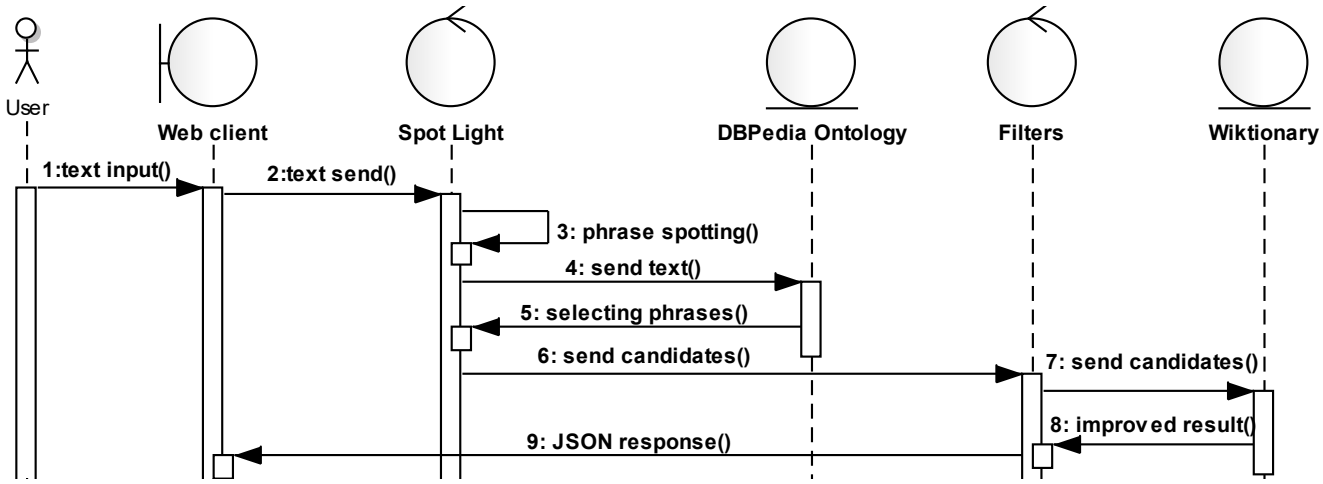


Рисунок 2.31 – Диаграмма последовательности процесса аннотирования документа

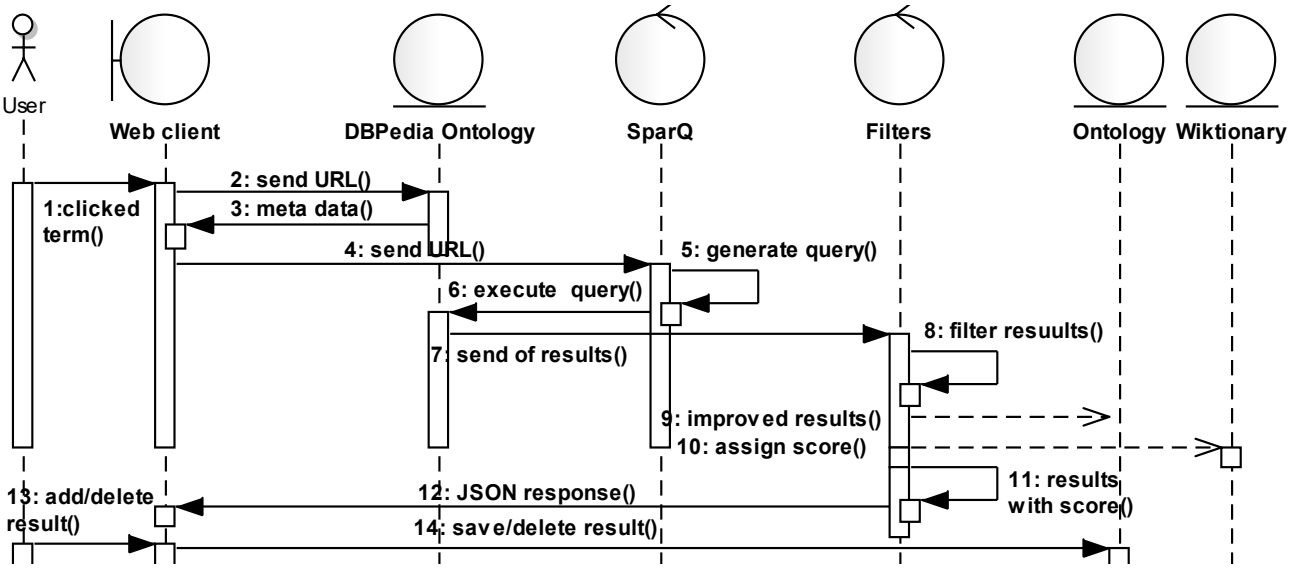


Рисунок 2.32 – Диаграмма последовательности процесса извлечения данных по запросу пользователя

Сначала алгоритм phrasespotting определяет фразы, которые должны быть связаны с сущностями DBpedia в данном тексте. Затем идентифицированные фразы оцениваются с использованием алгоритмов весов TF-IDF и меры косинусного сходства. Все фразы с оценкой сходства выше настраиваемого порогового значения включаются в результат. Для повышения точности и фильтрации ненужных терминов используется словарь Wiktionary и ручная корректировка терминов пользователем. Это показано на шагах 4–8 нарисунке 2.31. Если пользователь заинтересован в определенном термине, то новый запрос вместе с URI элемента отправляется на сервер приложений. Компонент поиска на сервере использует URI

для извлечения свойств элемента из онтологии DBpedia. На рисунке 2.32 шаги 1–4 отражают описанный выше в рамках онтологии процесс, которые необходимы для формулировки запросов на извлечение данных из DBpedia. Свойства dbo:abstract или dct:subject запрашиваемого элемента компилируются в объект JSON и отправляются обратно клиенту. Пользователь может выполнить окончательную корректировку и сохранить найденные концепты в онтологии.

В рамках тестового примера было выбрано десять объёмных англоязычных статей, подготовленных в рамках научных конференций, связанных с информационными технологиями. Для точного фиксирования результатов оценки качества поисковой системы были рассчитаны такие показатели, как полнота, точность и F-мера.

Сначала вручную были проанализированы понятия и отношения между понятиями в онтологии DBpedia для каждого поискового запроса, сформулированных с помощью SPARQL языка запросов и выполненных компонентом SPARQ.

Чтобы получить количество автоматически полученных результатов, документы были впоследствии загружены в веб-клиент OKMS, и входной текст был аннотирован компонентом DBpedia SpotLight. В таблице 2.17 показано сравнение результатов ручной разметки и автоматической обработки документов.

Таблица 2.17 – Результаты оценки работы аннотатора концептов

Text	Expert	True	False	Точность	Полнота	F-мера
Text 1	12	9	4	0,69	0,42	0,52
Text 2	34	32	2	0,94	0,88	0,91
Text 3	44	41	4	0,91	0,84	0,87
Text 4	27	23	2	0,92	0,78	0,84
Text 5	18	16	2	0,89	0,78	0,83
Text 6	29	26	3	0,90	0,79	0,84
Text 7	30	28	5	0,85	0,77	0,81
Text 8	15	16	1	0,94	1,00	0,97
Text 9	21	17	3	0,85	0,67	0,75
Text10	35	33	5	0,87	0,80	0,83
Average				0,87	0,78	0,82

В рамках тестового примера 266 понятий были вручную размечены экспертом. Программный модуль с точностью 0.87, полнотой 0.78 и F-мерой 0.82 автоматически аннотировал 241 элемент. Результаты тестирования показывают, что концепты могут быть извлечены из документов с использованием общедоступной онтологии DBpedia.

Результаты документа Text1 (Таблица 2.17) являются наименее точными (0.69), поскольку в этом документе содержатся термины, связанные с названиями организаций и сокращениями. Поэтому Recall также невелика (0.52).

Тем не менее, приведенные выше результаты тестирования показывают, что в общем концепты могут быть извлечены из документов с использованием общедоступной онтологии DBpedia.

Приведем результаты тестирования второго модуля, который извлекает семантически подобную информацию по выбранному термину. Чтобы получить эти данные из онтологии DBpedia, выполняются несколько запросов SPARQL, подобных приведенному на рисунке 2.33.

```
var query = ["PREFIX dbpedia2:
<http://dbpedia.org/resource/>",
"PREFIX Abs: <http://dbpedia.org/ontology/>",
"SELECT ?terms  WHERE {"*s ns:type dbo:Genre . *s dct:subject ?concept",
"?terms dct:subject ?concept . ?terms dct:subject ?concept ",
". ?terms ns:type dbo:Genre",
"}»]. join (" ");
var queryUrl = url+"?query="+ encodeURIComponent(query) +"&format=json";
```

Рисунок 2.33 – Пример запроса

Переменная *s заменяется значением. Запрос сначала проверяет, имеет ли ресурс тип жанр, а затем извлекает все связанные с ним понятия. Далее, используя обратную зависимость, он извлекает все остальные жанры из выявленных концептов.

Анализ результатов тестирования второго модуля приведен в таблице 2.18.

Точность для 1 запроса составляет 0.78, для второго 0.82, а для третьего 0.81. Эти результаты подтверждает гипотезу о том, что полезные данные могут быть извлечены из онтологии DBpedia.

Таблица 2.18 – Анализ результатов тестирования второго модуля

Ключ поиска	Релевантные результаты	Всего	Общая точность
Архитектуры	112	143	0,78
Программное обеспечение	154	186	0,82
Языки программирования	142	175	0,81

После анализа выяснено, что результаты по тематике, связанной с программированием и программными системами, являются более эффективными по сравнению с остальными, потому что программные технологии значительно лучше поддерживаются в онтологии DBpedia.

2.4 Онтологический подход к разработке системы

Основным преимуществом использования онтологий в системах обработки данных научных и образовательных организаций является эффективное представление и обработка знаний. Функции системы для работы со знаниями представлены следующим образом:

- работа с онтологиями, мета-описаниями и классификаторами;
- работа с семантическими сетями;
- экспертное наполнение хранилища знаний;
- приобретение знаний;
- поиск знаний в Интернете и по хранилищу;
- интеграция знаний и моделей;
- импорт как структурированной, так и не структурированной информации в хранилище;
- проверка данных на корректность;
- проверка документов на уникальность;
- извлечение знаний;
- формирование новых знаний;
- работа с пользователями системы (регистрация и аутентификация).

На рисунке 2.34 представлена абстрактная модульная структура системы, разработанная на базе онтологического подхода.

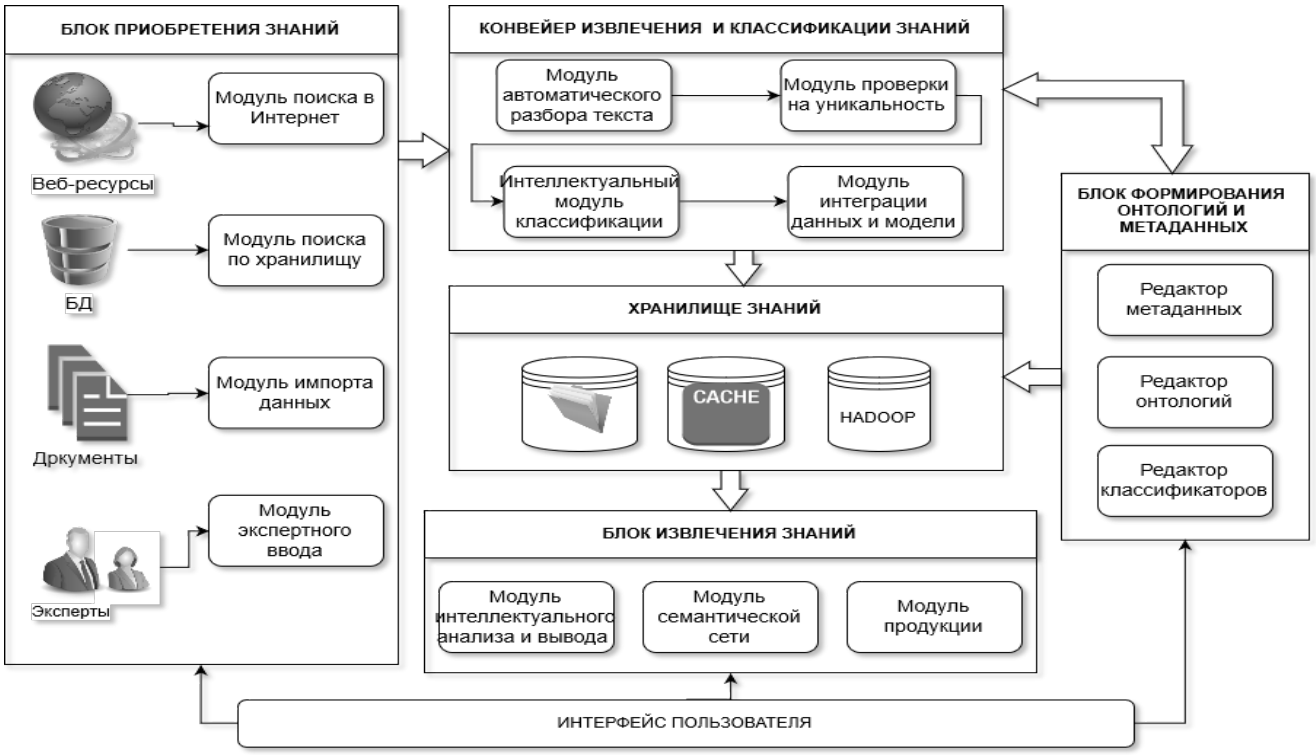


Рисунок 2.34 – Абстрактная модульная структура системы, разработанная на базе онтологического подхода

Основными укрупненными блоками системы являются: блоки приобретения данных, конвейер для обработки и классификации данных, блок для работы с онтологиями, а также блок выдачи и продукции знаний.

Основными модулями являются [118]:

- модуль поиска информации в сети Интернет;
- модуль поиска информации по хранилищу данных;
- модуль экспертного ввода данных;
- модуль импорта данных из семантически размеченных и шаблонизированных документов;
- модуль интеграции данных и модели;
- интеллектуальный модуль тематической классификации.
- модуль автоматического разбора текста.

- модуль проверки на уникальность;
- редактор онтологий;
- редактор классификаторов;
- редактор метаданных;
- модуль интеллектуального анализа данных и вывода знаний;
- модуль семантической сети;
- модуль продукции новых и комбинированных знаний;
- интерфейс пользователя.

Использование онтологий при решении основных задач конкретных модулей системы управления информационными ресурсами научно-образовательных организаций показано на рисунке 2.35.

Внутреннее хранилище знаний, а также навигация по нему, организованы на основе классов и отношений, определенных ранее в онтологии. Настройка и наполнение хранилища выполняется с помощью набора специализированных редакторов онтологий, классификаторов, метаданных. Основными функциональными возможностями редактора онтологий являются создание, модификация и удаление отдельных элементов онтологии, а также работа с иерархией классов.

Поиск в системе также осуществляется в соответствии с содержанием онтологии. При поиске информации пользователю предоставляется возможность сформулировать запрос в терминах предметной области и из области метаданных.

Навигация по хранилищу также осуществляется в соответствии с содержанием онтологии, позволяя переходить от понятий онтологии к ее экземплярам (информационным объектам), а затем осуществлять переход по онтологическим связям от конкретного экземпляра к спискам связанных с ним экземпляров.

Таким образом, основные факторы, определяющие целесообразность применения онтологического подхода следующие:

- онтология является, по сути дела, каркасом, представлением и моделью предметной области;

– одна и та же онтология широко используется при решении различных задач поиска и обработки информационных ресурсов;

– наполнение онтологии индивидами позволит динамически сформировать хранилище данных системы.

Рисунок 2.35 – Использование онтологий при решении основных задач модулей системы

2.5 Выводы

1. Разработан гибридный подход к созданию онтологии, заключающийся в том, что на различных этапах создания онтологии использованы различные способы ее создания. На начальном этапе использовались существующие онтологии «верхнего уровня», а также метод «экспертного создания». При формировании онтологий нижних уровней использовались методы автоматической обработки корпуса документов, словарные способы пополнения, а также ранее разработанные универсальные онтологии.

2. Разработан способ полуавтоматического пополнения онтологии на базе словарных определений готовых словарей, таких, как WordNet, RussNet, Wiktionary. Исследования показали, что электронные словари-тезаурусы WordNet и его русскоязычный аналог RussNet можно рассматривать как элементы для построения требуемой лингвистической онтологии. Для пополнения предметной онтологии нижних уровней предпочтительнее использовать Wiki – ресурсы, в частности словарь Wiktionary, для которого показатель качества поиска, выраженного F-мерой, достиг самого высокого среди тестируемых значения 0.87.

3. Разработан способ пополнения онтологии с использованием междоменной онтологии DBpedia. Исследования показали, что концепты могут быть извлечены из документов с точностью 0.87 (доля автоматически извлекаемых элементов, которые являются релевантными), полнотой 0.78 (доля релевантных элементов, которые были успешно извлечены) и F-мерой, равной 0.82.

4. Разработан способ пополнения онтологии с помощью шаблонизированных и семантически размеченных документов.

5. В рамках онтологического подхода выделены основные функции, при которых использование онтологий в системах обработки данных научных и образовательных организаций является необходимым. На базе онтологического подхода спроектирована абстрактная модульная структура СУИР научно - образовательных организаций.

РАДЕЛ 3
РАЗРАБОТКА МОДЕЛЕЙ, МЕТОДОВ И АЛГОРИТМОВ ВЫЯВЛЕНИЯ, ПРИОБРЕТЕНИЯ И КЛАССИФИКАЦИИ ЗНАНИЙ

3.1 Разработка обобщенной метамодели

Для учета связанности знаний и обеспечения однородности представления данных в рамках единой тематики проектируемой системы была предложена единая концептуальная схема – обобщенная метамодель (Таблица 3.1).

Таблица 3.1 – Обобщенная метамодель

УРОВЕНЬ ПРИОБРЕТЕ-НИЯ ЗНАНИЙ	УРОВЕНЬ ВЫЯВЛЕНИЯ ЗНАНИЙ	УРОВЕНЬ ПРЕДСТАВЛЕН ИЯ ЗНАНИЙ	УРОВЕНЬ ИНТЕГРАЦИ И ДАННЫХ	УРОВЕНЬ ХРАНЕНИЯ ДАННЫХ	УРОВЕНЬ ИЗВЛЕЧЕНИЯ ЗНАНИЙ
					
БАЗОВЫЕ МОДЕЛИ, МЕТОДЫ И АЛГОРИТМЫ					
Методы извлечения на основе метаданных Методы разбора шаблонных документов Экспертная модель	Методы TextMining DataMining Модель семантического анализа Методы частотной обработки текстов	Модели семантических сетей Продукционная модель Фреймовая модель Модель нечеткой логики	Модель классификации	Файловая модель Иерархическая модель Объектная модель Реляционная модель	Формально-логическая модель Модель семантической сети
РАЗРАБОТАННЫЕ И ИЗМЕНЕННЫЕ МОДЕЛИ, МЕТОДЫ И АЛГОРИТМЫ					
Алгоритмы разбора документов на базе регулярных выражений Методы извлечения на основе метаданных	Интеллектуальная гибридная мера определения СБ Модель информационного поиска Модель формирования поисковых запросов Векторная модифицированная модель представления текстов	**ОНТО-МОДЕЛЬ** Модель N-мерного представле-ния знаний RDF-графа	Алгоритм классификации на базе онтологического подхода	Алгоритм динамического формирования хранилища	Алгоритм извлечения релевантных данных по запросу

Метамодель объединяет все модели предметной и проблемной областей, на основе которой затем строятся внутренние хранилища знаний системы, а также выполняется поиск и извлечение знаний. Созданная метамодель базируется на основополагающих принципах построения СУЗ, описанных в трудах таких российских авторов, как Б.З. Мильнера, Т.А. Гавриловой, В.Ш. Рубашкина и др.

Разработанная укрупненная метамодель системы управления знаниями научно-образовательных организаций представлена в шести уровнях:

1. Уровень выявления знаний – поиск данных из различных источников, таких, как документы и файлы различных форматов, веб-ресурсы, датасеты, БД и др. Информацию, представленную в структурированном виде (метаописания, онтологии, шаблонные факты, семантически размеченные файлы и т.п.) обрабатывать проще, но в таком виде представлено относительно мало информации, гораздо больший объём информации содержится в неструктурированных данных.

2. Уровень приобретения знаний – получение знаний, извлечение неформализованных знаний из разнородных источников информации с помощью методов статистической обработки, семантического анализа, технологий TextMining и DataMining, а также экспертных моделей.

3. Уровень представления знаний – формализация знаний на основе создания онтологии, семантических и других моделей. Сначала выполняется идентификация знаний – первый этап определения системы междисциплинарных связей между элементами знаний используемых предметных областей. Так можно выявить те знания, которые будут рассматриваться как уже приобретенные. Улучшение знаний происходит в процессе семантического поиска в неоднородных распределенных источниках знаний и энциклопедических справочных системах на основе онтологических моделей успешных прецедентов поисковых запросов. Улучшение знаний можно рассматривать как подзадачу создания новых знаний и как задачу автоматического пополнения базы знаний. Осуществляется также классификация и систематизация знаний, без чего немыслимо эффективное хранение знаний с целью обеспечения эффективного поиска.

Перед помещением знаний в хранилище осуществляется их аннотирование – процесс создания метаописаний. Аннотирование может происходить как с участием человека, так и без него, с помощью специальных алгоритмов. Результатом аннотирования является набор метаописаний, который хранится в классах онтологии. В метаописаниях можно выделить три типа метаданных: системные (служебные); структурные и семантические. Системные предназначены для функционирования информационных систем и систем управления знаниями. Они включают имена файлов и баз, даты их создания, тип и формат, размер файла и вид носителя и т.п. Структурные содержат, как правило, справочную информацию об объектах. Семантические – особый вид метаописаний, включающий концептуальное (аннотированное) изложение содержания и смысла информации об объекте.

4. Уровень интеграции данных. На этом этапе выполняется занесение собранных структурированных материалов, онтологий и извлеченных знаний из данных в общее интегрированное хранилище, при необходимости выполняя корректировку данных в экспертном режиме.

5. Уровень хранения знаний. Так как знания явные и неявные отличаются между собой, различаются и способы их хранения. Явные знания (текстовые документы различных форматов, электронные таблицы, базы данных, тестовые наборы данных, Web-страницы, чертежи, схемы, почтовые сообщения и т.п.) хранятся в специально создаваемых для этой цели хранилищах знаний. Для хранения используются такие модели данных, как иерархическая, объектная, реляционная и файловая.

6. Уровень извлечения и производства знаний. В результате поиска мы ищем не сам документ, а сведения об информационном объекте, реальном или виртуальном, т.е. некоторые знания. Таким образом ядром, базовым компонентом разработанной метамодели системы СУИР является его онтология – модель знаний, которая используется для описания семантики предметной области.

3.2 Разработка моделей, методов и алгоритмов уровня приобретения знаний

На уровне приобретения знаний решаются две основные задачи. Задача получения знаний (knowledge elicitation) – это процесс приобретения знаний экспертом, была решена за счет разработки редактора онтологий. Задача извлечения информации (knowledge acquisition) – это процесс автоматического (полуавтоматического) извлечения структурированных данных из неструктурированных или слабоструктурированных машиночитаемых документов. Среди основных методов ее решения представлены методы, построенные на использовании регулярных выражений и метаданных.

СУИР научно-образовательных учреждений позволяет в полуавтоматическом и автоматическом режиме обрабатывать документы различных форматов. Изначально документы могут подаваться системе в форматах HTML, TXT, DOCX, PDF. В процессе предобработки документы приводятся к единому виду. В случае, если не поддерживается тип файла системой, необходимо предварительно вручную его преобразовать с помощью внешнего конвертера или в крайнем случае не выполнять автоматический разбор текста, а переходить к экспертному его описанию.

Для успешной автоматической обработки файлы должны загружаться из проверенных источников, должны иметь поддерживаемый тип файла и иметь стандартную структуру, утвержденную ГОСТ либо стандартами организации.

Разработанный укрупненный алгоритм полуавтоматической программной обработки текста в СУИР.

1. Выбирается пользователем документ для обработки.
2. Выбирается пользователем тип документа (необязательно).
3. Определяется формат загруженного документа.
4. Извлекается текст в соответствии с форматом документа.
5. При определении типа документа выбирается нужный набор парсеров, в обратном случае применяется набор по умолчанию.

6. Выбранными парсерами обрабатывается документ и извлекается полезная информация.

7. Пользователю предоставляется возможность откорректировать извлеченную информацию.

8. Извлечённая информация сохраняется в базу данных и передается в другие модули для дальнейшего анализа документа.

Разработанный укрупненный алгоритм извлечения терминов.

1. Убираются все ненужные для анализа символы (пунктуационные знаки, цифры, т.д.).

2. Выполняется токенизация (текст разбивается по пробелам для получения слов по отдельности).

3. Убираются стоп-слова (междометия, союзы, вводные слова и т.д.).

4. Проверяется орфография (если возможно, исправляется в случае ошибки).

5. Определяется часть речи слова (как правило, интересуют только существительные).

6. Проверяется наличие дефиса в слове. Если дефис имеется:
   a. Слово разбивается по дефису.
   b. Слова приводятся к единой форме (лемме).
   c. Слова соединяются.
   d. Полученное слово добавляется в массив вместо изначального.

7. Все остальные слова приводятся к единой форме (лемме).

8. Находятся слова с максимальным количеством вхождений.

На основе алгоритмов извлечения текстовой информации были разработаны и сами программные модули-парсеры для обработки наиболее часто встречающихся типов файлов: txt, html, pdf, rtf, docx.

Для обработки структурированных и слабоструктурированных документов были разработаны алгоритмы обработки для типовых документов (статья, отчет, ВКР и т.п).

Метаданные, т.е. данные о «данных», используются уже достаточно давно в

различных системах и сервисах (электронных библиотеках, Web-сайтах, хранилищах ИР и пр.), являясь базовым компонентом таких систем.

Семантические метаданные ИР представляют собой описание самого ИР: название ИР, его тип, назначение, объем, предметное содержание, технические особенности, сведения об авторах и разработчиках и другую информацию, которая может быть полезна при выборе ресурса (Рисунок 3.1).

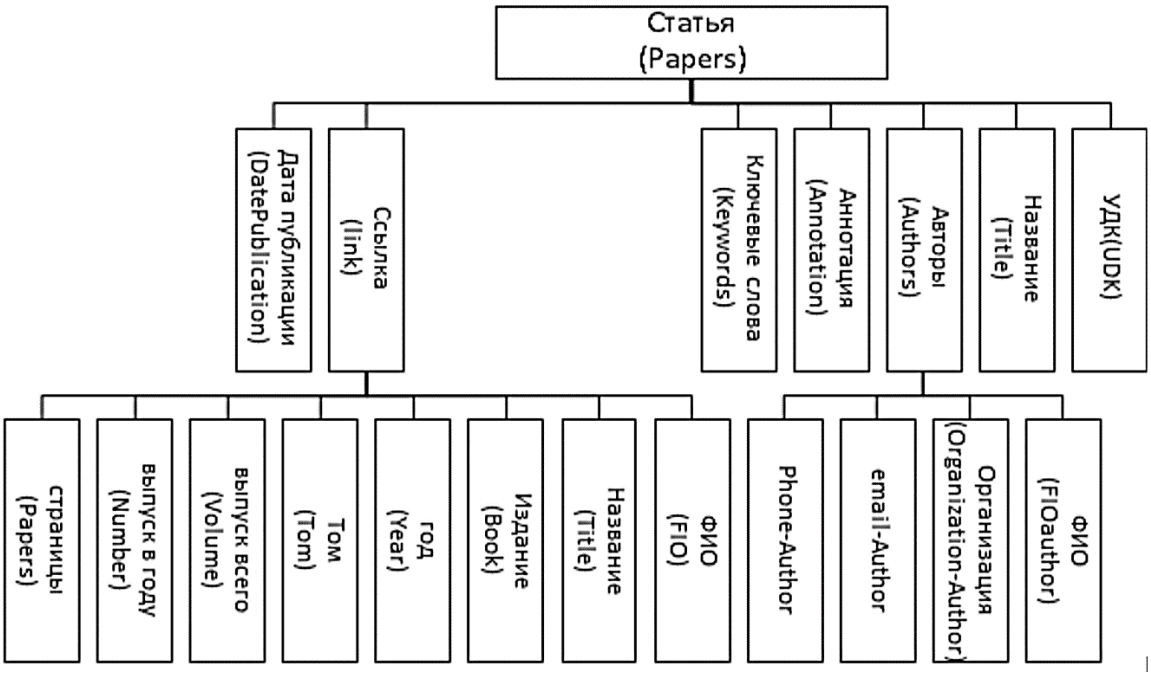


Рисунок 3.1 – Пример описания метаданных на концепте «Статья»

К числу основных требований к системе метаданных относятся: универсальность в рамках установленного понимания ИР как объекта систематизации; структурированность и формализованность метаданных, необходимых для их автоматической обработки; достаточная выразительность для обеспечения решения задач, требующих наличия метаданных; совместимость с международными стандартами и протоколами; возможность хранения метаданных как совместно с ИР, так и отдельно; возможность представления в метаданных сведений о создателях и владельцах ИР.

Метаданные были выделены и созданы – они неявно вошли в состав классов онтологии. Это позволило обеспечить все предъявляемые требования к метаданным в полном объёме и привело к упрощению структуры системы в целом.

3.3 Разработка интеллектуальной гибридной меры определения семантической близости

Ключевым моментом при построении любой СУЗ является разработка алгоритмов расчета количественных оценок семантической близости онтологических термов.

Функция F (X, Y), ставящая в соответствие каждой паре термов X и Y некоторый вещественный коэффициент, называется функцией, определяющей семантическую близость между двумя термами.

Для F (X, Y) действительны следующие свойства:

- $0 \leq F(X, Y) \leq 1$;
- $F(X, Y) = 1 \Leftrightarrow X = Y$ (объекты X, Y идентичны);
- $F(X, Y) = 0 \Leftrightarrow X \neq Y$ (объекты X, Y совершенно различны и не имеют схожих характеристик);
- F (X, Y) = F (Y, X), свойство симметричности функции подобия.

В свою очередь, каждый терм представляет собой некоторое размытое множество, куда попадают и другие подобные термы со значением семантической близости выше заданного порога.

Принадлежность к нечеткому множеству задается с помощью значения семантической близости.

Подобие сущностей X и Y означает, что F (X, Y)> = t, где t - уровень подобия.

Отсечение сущностей X и Y означает, что F (X, Y) <= t1, где t1-уровень отсечения.

Согласно анализу методов, приведенных в разделе 1.4.4, гибридные меры семантической близости, сочетающие несколько подходов и мер к определению числовой оценки СБ, являются наиболее перспективными для дальнейшего использования в алгоритмах поиска и классификации при решении этих задач в рамках онтологического подхода.

3.3.1 Разработка модели N-мерного представления знаний RDF-графа

Современным подходом при построении баз знаний является использование стека семантических технологий. Стандартизованный консорциумом W3C RDF специфицирует архитектуру, синтаксис и семантику, а также базовый словарь RDF Schema (RDFS) для построения моделей предметных областей [119]. В концепции Semantic Web модель данных в виде RDF-графа представляется следующим образом: Subject-Predicate-Object. Каждая сущность в свою очередь имеет свой универсальный и уникальный идентификатор ресурса – URI (Uniform Resource Identifier).

Для моделирования бинарных отношений на RDF – графе удобно использовать трёхмерный тензор [120]. Наилучшим определением тензора будет цитата Tamara G. Kolda: “A tensor is a multidimensional array” [121]. У тензора имеются верхние, нижние и смешанные индексы. Верхние меняют значение при переходе от одной строки к другой, а нижние при переходе от одного столбца к другому.

В работе [122] авторами было предложено представление графа знаний в виде тензорного разложения RESCAL. Для представления многомерных семантических данных авторы работы использовали формализм RDF семантической сети, где отношения представляются как тройки (субъект, предикат, объект). Предикат моделирует либо отношения между двумя сущностями, либо отношения между сущностью и значением атрибута. Тензорная запись $S_{ijk} \neq 1$ обозначает тот факт, что существует RDF отношение (i-я сущность, j-й субъект, k-й предикат). В противном случае для несуществующих и неизвестных отношений запись устанавливается равной нулю.

В данной работе для улучшения семантики предлагается значение трехмерного тензора семантических связей определять в диапазоне [0; 1], кроме иерархических отношений, по-прежнему принимающих значений «0» или «1» (Рисунок 3.2).

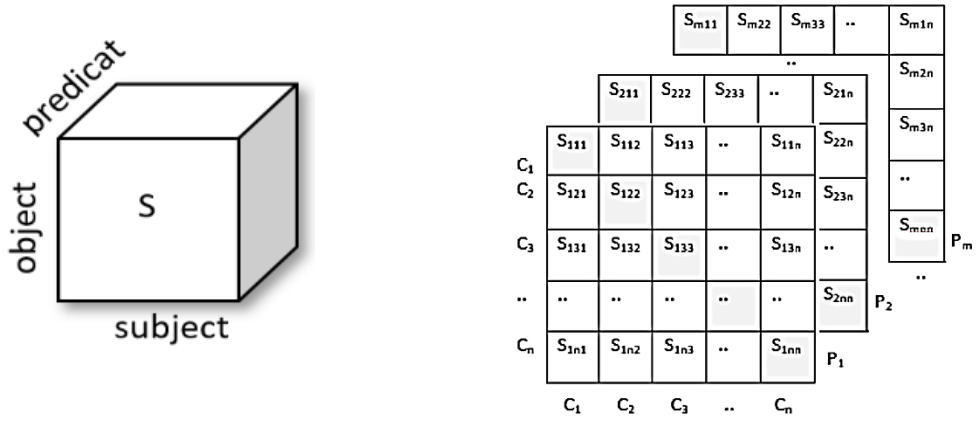


Рисунок 3.2 – Трехмерный тензор семантических связей

Опишем отношения, которые были использованы в предлагаемой гибридной интеллектуальной модели и способы определения их СБ в таблице 3.2 [123].

Таблица 3.2 – Виды оценок СБ, используемых в гибридной модели

Отношения	Вид оценки	Способ определения
P1 - таксономические P2 - партономические P3 - родовидовые	по иерархии онтологии	Длина кратчайшего пути (см. формулу 1.6)
P4 - синонимы, P5 – экземпляры (individual)	по свойствам концептов	Длина кратчайшего пути (см. формулу 1.6)
P6 - ассоциации	по горизонтальным отношениям	Заполняется экспертом значениями в диапазоне [0;1]
P7 - семантические	по общим атрибутам концептов	Атрибутивная мера (см. формулу 1.7)
P8 - семантические	на векторном представлении текста	Косинусная мера (см. формулу 1.9) Мягкая косинусная мера (см. формулу 1.13)

3.3.2 Оценка семантической близости по онтологии

СБ двух термов может быть оценена по положению вершин в иерархической структуре данных – онтологии. Для получения иерархии классов предметной области в виде графа был выделен «скелет» («каркас») онтологии. При этом ребра дерева отражают иерархические отношения трех типов:

– P1 – таксономические (IS-A, KIND-OF, has, имеет);

– P2 – партономические (PART-WHOLE, part-of, часть-целое);

– P3 – родовидные (PARENT-CHIEILD, TOPIC-SUBTOPIC).

Зафиксировав индекс k=1 в трехмерном тензоре семантических связей, получили след тензора – матрицу семантических связей S_{ij1} для первого отношения P1 (Рисунок 3.3).

	C_2	C_3	..	C_n
C_1	S^{11}	S^{12}	S^{13}	..	S^{1n}
C_2	S^{21}	S^{22}	S^{23}	..	S^{2n}
C_3	S^{31}	S^{32}	S^{33}	..	S^{3n}
..	..	..	..	..	..
C_n	S^{n1}	S^{n2}	S^{n3}	..	S^{nn}

Рисунок 3.3 – Матрица семантических связей S_{ij1}

Элемент матрицы смежности S_{ij1} равен числу 1, если существует ребро между вершинами C_i и C_j. Элемент S_{ij1} равен нулю, если рёбер между вершинами C_i и C_j не существует. Для отношений P2 и P3 матрица строится подобным образом.

За расстояние между концептами была принята длина минимального пути между ребрами, так как требуется построить в максимальной степени сбалансированное дерево. При расчете использовался путь от ближайшего общего предка обоих термов и лишь в случае его отсутствия расчет шел от корня, затем значения нормировались.

3.3.3 Оценка семантической близости отношений, представленных OWL свойствами концептов

Важным свойством графов знаний является возможность возникновения различных корреляций между множеством взаимосвязанных узлов. Подобные корреляции могут быть вычислены за счёт включения обработки атрибутов, связей

и классов связанных сущностей. Мера СБ показывает высокие значения для концептов, которые находятся в семантических отношениях (синоним, гипоним, ассоциативность) и нулевые значения для всех остальных пар.

Свойства типа Data Property определяют связь между объектами и данными. Выделим отдельно отношение синонимизации ввиду его семантической важности. Зафиксировав индекс k=4, мы получим матрицу семантических связей для отношения P4 – синонимизации (подобия). Если концепты связаны отношением эквивалентности, то СБ=1, иначе 0.

Свойства типа Object Property определяют отношения между индивидуальными объектами. Зафиксировав индекс k=5, мы получим матрицу семантических связей для отношения P5 – свойства Individual. Если два концепта связаны отношением Object Property, то их мера связанности также равна 1.

Между классами и экземплярами различных классов наряду с иерархическими связями могут быть и другие «горизонтальные» семантические связи, представляемые объектными бинарными отношениями, характерными для описываемой предметной области.

Будем считать, что свойству $p \in P$ может быть задан весовой коэффициент (семантический вес) $pv \in [0; 1]$, задающий смысловую близость между субъектом и объектом утверждения (значение коэффициента 1 означает, субъект и объект считаются полностью сходными по семантике, значение коэффициента 0 – совсем не похожими). Задание значений коэффициентов pv для предикатов выполняется специалистами-разработчиками в соответствии с их пониманием онтологии и потребностями решаемых задач. Для индекса k=6 мы определили матрицу семантических связей для отношения P6 – свойства ассоциации (семантических отношений). Поскольку данная матрица не является матрицей смежности, то ее значения всех связанных семантически ячеек рассчитываются с помощью эксперта, остальные обнуляются.

Поскольку в OWL стандарте предусмотрено хранение свойств-атрибутов, то целесообразно будет и вычисление атрибутивной меры близости концептов для отношения P7 – атрибутивной меры сходства, значения которой находятся в [0; 1].

3.3.4 Оценка семантической близости, основанная на векторном представлении текста

Следующей группой мер являются меры, построенные на предположении, что семантически близкие концепты встречаются в тексте в одинаковых контекстах, т.е. обладают похожим набором ключевых (контекстных) слов, которое называется контекстным множеством. Таким образом мера сходства определяется по векторному представлению документа. Для определения меры сходства между двумя документами наиболее часто используются меры на базе косинусного сходства, которые особенно эффективны в качестве оценочной меры для разреженных векторов, когда учитываются только ненулевые измерения.

Таким образом, мы добавили в тензор под индексом k=8 матрицу семантических связей P8, рассчитанную по косинусной векторной мере сходства контекстных множеств документов.

3.3.5 Генетический алгоритм определения коэффициентов весовой функции интеллектуальной гибридной меры

Тензорное представление графа семантических отношений позволяет эффективным образом вычислить оценки семантической близости между двумя концептами i и j через факторизацию срезов тензора. Выполнив аддитивную свертку тензора S^{ij}_k с вектором коэффициентов значимости для каждого типа отношения W^k получаем:

$$R^{ij} = \sum_{k=1}^{p} W^k S_k^{ij}, \qquad (3.1)$$

где W^k – вес, который определяет относительную важность каждого типа отношения; p – число отношений; R – матрица семантических связей.

Для решения задачи нахождения весовых коэффициентов предлагается использование генетического алгоритма, который наиболее эффективно

обеспечивает поиск решения для функций, имеющих несколько экстремумов. В качестве общей структуры алгоритма использовался модифицированный генетический алгоритм.

ГА характеризуется следующими основными параметрами:

$$ГА = (P^0, \boldsymbol{\lambda}, L, S, P, f, t),$$

где $P^0 = (w^0_i \ ... \ w^0_{\boldsymbol{\lambda}})$ – исходная популяция (одно поколение решений, представленное хромосомами); w^0_i – решение задачи в виде одной хромосомы; $\boldsymbol{\lambda}$ – размер популяции; L – длина каждой хромосомы популяции; S – оператор отбора (репродукции); P – отображение, определяющее оператор рекомбинации; f – целевая функция; t – критерий остановки алгоритма.

Каждое решение задачи (особь) представлено в виде десятичной (вещественной) хромосомы. Ниже приведен вид популяции, представленный в виде таблицы 3.3.

Таблица 3.3 – Популяция

Особь 1	W^1	W^2	W^3	W^4	W^5	W^6	W^7	W^8
Особь 2	W^1	W^2	W^3	W^4	W^5	W^6	W^7	W^8
Особь ...	..	..	..	..	..	..	..	..
Особь N	W^1	W^2	W^3	W^4	W^5	W^6	W^7	W^8

На множестве решений определяется целевая функция (ЦФ), которая позволяет оценить качество каждого из полученных решений. Целевую функцию построили с использованием Евклидового расстояния:

$$\sqrt{\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{p} W^k(S_k^{ij} - 1)\right)^2} => min,$$

где W^k – весовые коэффициенты; p – число отношений; n – количество концептов; S – матрица семантических связей концептов.

Целевая функция имеет следующие ограничения: $W^k \in [0; 1]$, $\sum_{k=1}^{p} W^k = 1$.

Для нахождения первоначальных весов допустим, что все W^k равноценны, тогда $W^k = 1/p$.

Блок-схема работы генетического алгоритма представлена на рисунке 3.4.

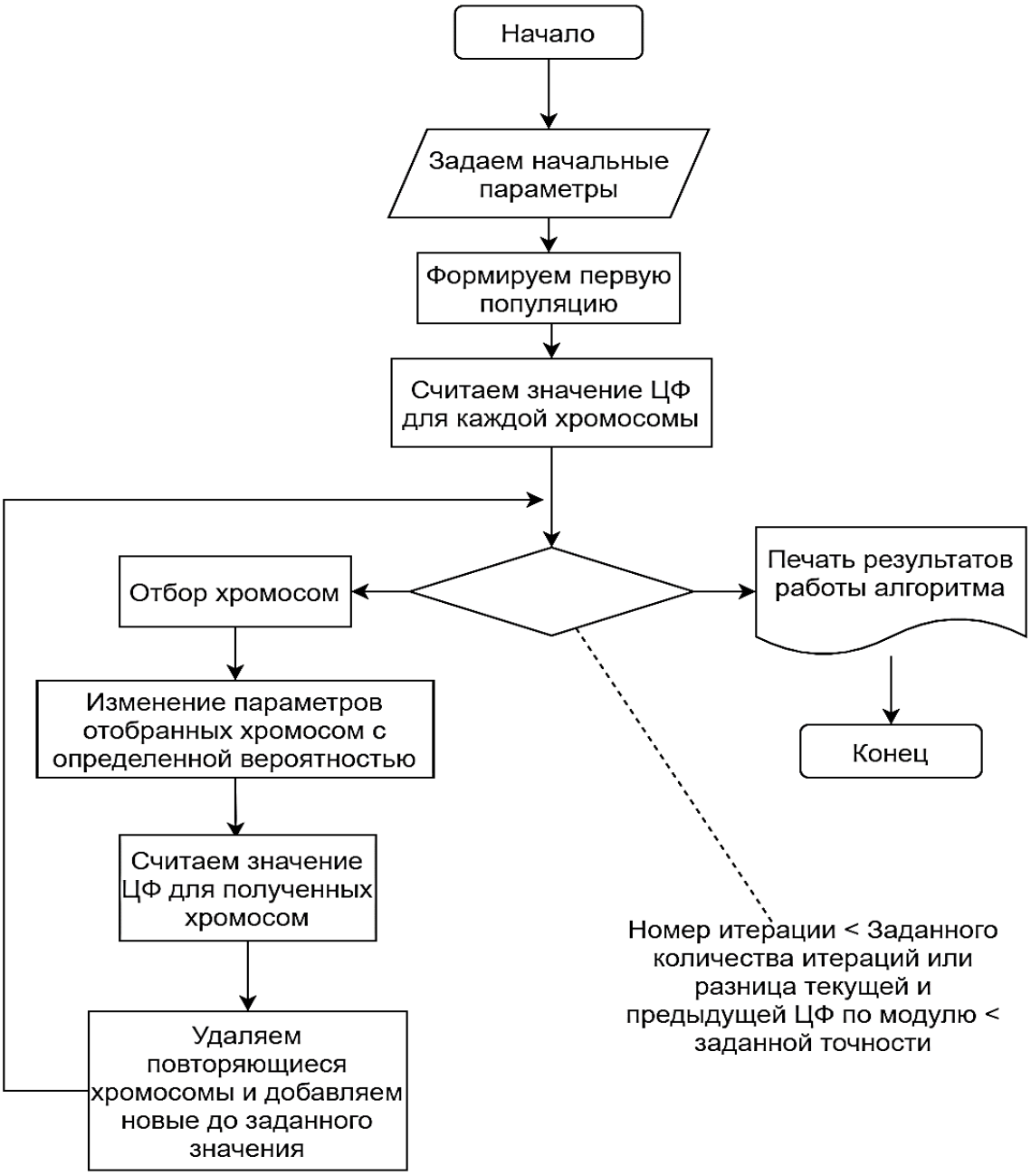


Рисунок 3.4 – Генетический алгоритм

После формирования исходной популяции действует оператор отбора. Принцип его работы состоит в следующем: выбираются две особи. Если вероятность больше 0.02, то в родительский пул попадает та особь, которая лучше (меньше) по значению ЦФ, иначе та, которая была отобрана случайным образом. Алгоритм работы оператора отбора отображен на рисунке 3.5.

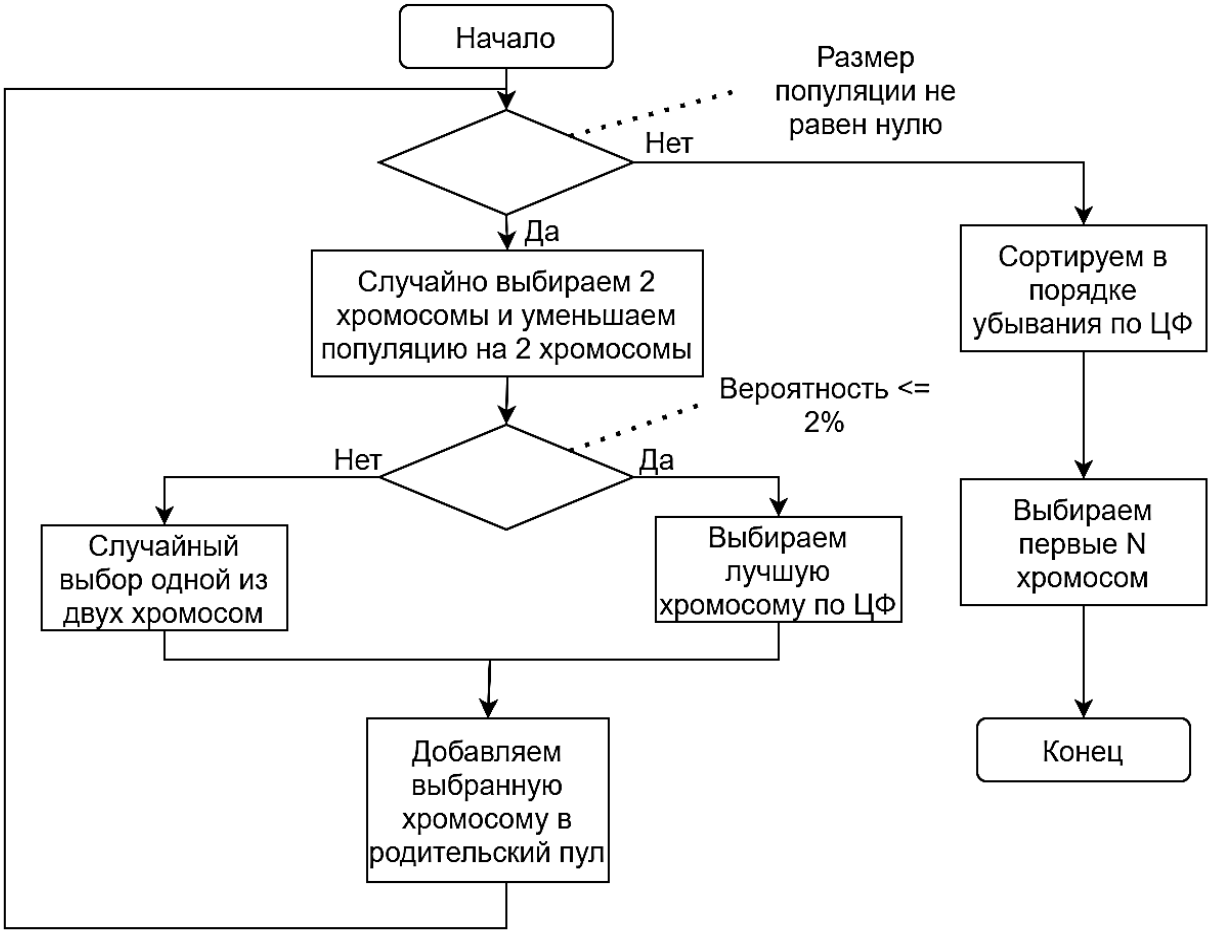


Рисунок 3.5 – Алгоритм работы оператора отбора

После формирования родительского пула сортируем особей в порядке возрастания целевой функции и забираем первые N особей (N задается перед началом работы программы).

На рисунке 3.6 приведем разработанный алгоритм изменения параметров особи. Программа в цикле извлекает каждый элемент из родительского пула и увеличивает параметр, выбранный случайным образом, на 1–3%, но так как должно соблюдаться выражение $\sum_{k=1}^{\mathrm{p}} W^k = 1$, то необходимо пересчитать все остальные параметры. Затем проводится проверка массива на наличие дубликатов и в случае их обнаружения – удаление.

Далее выполнили пересчет значений ЦФ для каждой особи с учетом новых параметров. И уже из полученного массива особей выбрали лучшую по значению ЦФ и сравнили с лучшей особью предыдущей популяции.

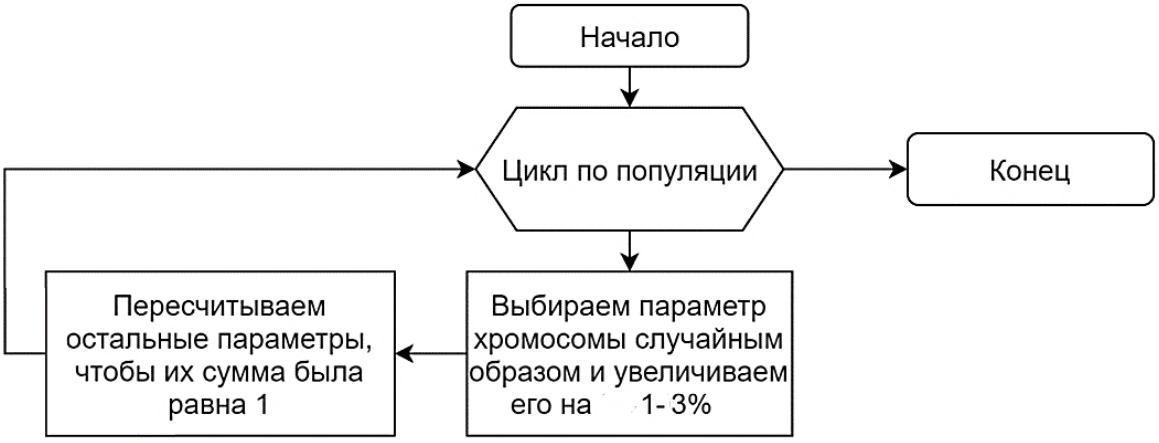


Рисунок 3.6 – Алгоритм изменения параметров особи

В алгоритме предусмотрено два критерия останова: достижение предельного числа поколений и схождение популяции. Основным критерием является последний, если разница по модулю ЦФ двух особей будет меньше заданной погрешности – работа алгоритма прекращается и выводится массив с лучшей популяцией.

На рисунке 3.7 приведены результаты работы ГА с некоторым субоптимальным решением, для которого значение ЦФ≈0.27 и точностью вычисления целевой функции равна 0.00028.

```
Population:
 w1          w2          w3          w4          w5          w6          w7          w8          ЦФ
0.130234 | 0.185133 | 0.180771 | 0.036964 | 0.097532 | 0.116861 | 0.047196 | 0.205308 | 0.276812 |
0.101862 | 0.068426 | 0.194581 | 0.181352 | 0.116653 | 0.055203 | 0.151173 | 0.130752 | 0.277092 |
0.055759 | 0.122883 | 0.153942 | 0.026728 | 0.156899 | 0.165881 | 0.154323 | 0.163586 | 0.277158 |
0.198237 | 0.194902 | 0.082862 | 0.105857 | 0.080561 | 0.061234 | 0.078408 | 0.197939 | 0.277284 |
0.056046 | 0.013472 | 0.150313 | 0.094314 | 0.200469 | 0.165354 | 0.136235 | 0.183796 | 0.277369 |
0.220158 | 0.051768 | 0.006839 | 0.223706 | 0.028054 | 0.216086 | 0.080989 | 0.172399 | 0.278051 |
0.092462 | 0.050950 | 0.104905 | 0.264480 | 0.047488 | 0.077022 | 0.150700 | 0.211993 | 0.277667 |
ЦФ1=0.277092  ЦФ2=0.276812  EPS=0.00028
================================================================================
```

Рисунок 3.7 – Результаты работы ГА

Получены следующие весовые коэффициенты: W^1=0.13; W^2=0.18; W^3=0.18; W^4=0.04; W^5=0.1; W^6=0.12; W^7=0.05; W^8=0.2, которые установлены в качестве параметров модели в настройках системы.

3.4 Разработка моделей, методов и алгоритмов уровней выявления и извлечения знаний

Уровень приобретения знаний (knowledge discovery) – получение знаний, извлечение неформализованных знаний из разнородных источников информации с помощью методов статистической обработки, семантического анализа, технологий TextMining и DataMining экспертных моделей, а также формирование знаний путем обучения (machine learning).

3.4.1 Разработка модели информационного поиска знаний

Основной задачей системы управления информационными ресурсами научно-образовательных учреждений, является задача информационного поиска (Information retrieval), которая занимается поиском релевантной поисковому запросу неструктурированной документальной информации [123] и помогает уменьшить проблему «информационной перегрузки».

Большинство ИР представляет собой тексты на естественном языке, преимущественно на русском и английском языках. В связи с накопившимся большим объёмом ресурсов возникает проблема поиска релевантных документов, несущих необходимые целевые знания. Проблема поиска связана с отсутствием инструментов, помогающих ориентироваться в этих накопленных информационных массивах данных, среди которых большая часть остается невостребованной и неиспользуемой, а со временем устаревает и теряет актуальность. Сам же поиск представляет собой неэффективную и трудозатратную задачу, которая часто по-прежнему решается примитивным способом.

Для разработки эффективных моделей поиска и использования ранее накопленных материалов необходимо перейти к новому уровню обработки информации, семантическому. В области семантического веба лучшим средством представления семантики является онтология.

Извлечение знаний из документов и представление их в удобном для использования виде представляет собой задачу семантического анализа, который в нашем случае базируется на использовании онтологий. При этом в полуавтоматическом режиме можно получить множество ключевых слов, важность которых определяется с помощью весов (задача индексации).

В ходе изучения проблемы было обнаружено, что поисковые запросы возвращают недостаточно релевантные результаты и эта проблема решается в задаче расширения и семантического обогащения поискового запроса. Для устранения избыточности поисковых запросов выполняется ранжирование результатов поиска.

Для отнесения ИР к тематическому разделу необходимо решить задачу классификации, также являющейся одной из задач информационного поиска. Для задач классификации требуется предварительное определение числа и наименований рубрик, которые уже определены и хранятся в онтологии. Точность классификации зависит от ряда факторов: содержания текстового документа, качества предварительной подготовки документа к классификации, качества построенной онтологии. Анализ разработок в данной области показал, что достаточной теоретической базы по применению онтологического подхода к классификации документов научно-образовательных организаций не разработано. Поэтому для решения поставленной задачи требуется разработать алгоритм классификации документов, учитывающий специфику рассматриваемой области научно-образовательной деятельности учреждений.

Формально постановку задачи информационного поиска можно описать следующим образом.

Пусть имеем формализованную модель онтологии $O = (C, R, A, P, T)$ предметной области, описанную по формуле (2.1).

На основе онтологии для каждого документа из корпуса $D = \{d_1, d_2, ..., d_n\}$ имеется контекстный вектор $V = \{v_1, v_2, ..., v_k\}$.

Вектор запроса записан в виде: $Q = \{q_1, q_2, ..., q_o\}$.

Определена также функция F (X, Y), ставящая в соответствие каждой паре термов X и Y некоторый вещественный коэффициент, определяющий семантическую близость между двумя термами.

Тогда решаемая задача информационного поиска заключается в том, что для каждого запроса Q требуется определить подмножество R из множества информационных ресурсов D, которое состоит только из максимальным образом релевантных документов при активном использовании онтологических ресурсов и экономии вычислительных. При этом документ считается релевантным, если числовая оценка близости находится больше некоторой пороговой величины t (уровень подобия).

Процесс информационного поиска включает ряд последовательных операций, направленных на сбор, обработку и предоставление пользователю знаний. Укрупненно процесс информационного поиска приведен на рисунке 3.8.

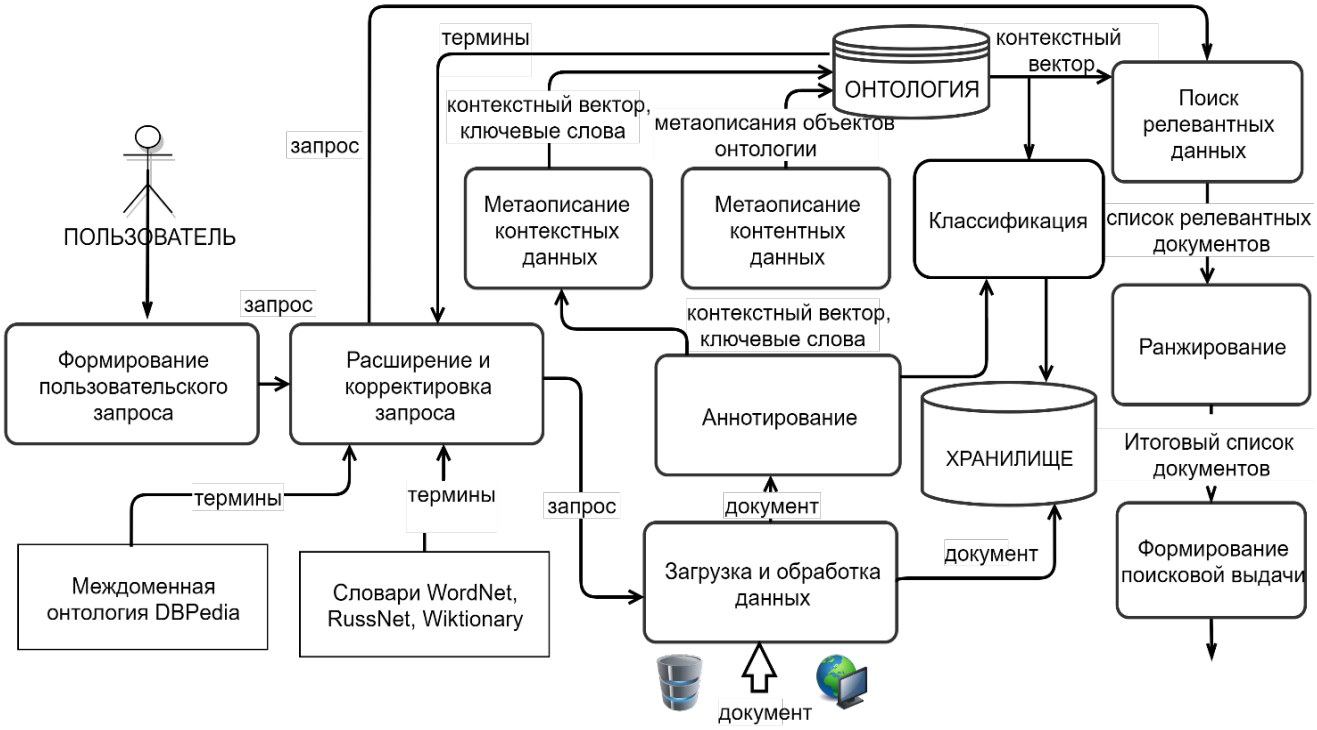


Рисунок 3.8 – Процесс информационного поиска

В ходе изучения проблемы [124] было обнаружено, что поисковые запросы возвращают недостаточно релевантные результаты по ряду причин: резкий рост

объемов информационного наполнения, порожденный популярностью и дешевизной Web-технологий; формат данных не ориентирован на автоматическую обработку; неудовлетворительная работа поисковых машин из-за огромного количества ресурсов Web-пространства; неправильные формулировки текстов запросов пользователями; отсутствие семантики в запросах и др. Некоторые из этих проблем были учтены в процессе разработки поискового модуля.

Разработка математической модели информационного поиска состоит из разработки нескольких моделей:

- модели представления текстов (документов);
- модели формирования поисковых запросов;
- модели получения релевантных данных по запросу.

3.4.2 Разработка моделей представления текстов

Для повышения эффективности обработки данных и преодоления ограничений модель bag-of-words была модифицирована за счет комбинации модели bag-of-words, bag-of-concepts, модели контекстных векторов, а также применения предметной онтологии [125].

Суть метода отображения текста в вектор заключается в том, что каждому слову соответствует определенная координата в пространстве признаков или вес в соответствии с выбранной весовой функцией.

Для полного определения векторной модели текста необходимо выбрать весовую функцию. Существует несколько базовых локальных схем взвешивания, например, булево (бинарное), по частоте слова TF, нормализованное, логарифмическое и несколько глобальных, таких, как IDF – инвертированная документная частота, вероятностная инверсия, GFIDF и др.

Использование глобальных схем взвешивания, теоретически, должно компенсировать недостатки локальных схем. При этом использование глобальных схем взвешивания предполагает многократное использование всего корпуса документов, что приводит к высокой нагрузке на производительность системы и

повышенным требованиям к ресурсам ПК, особенно к объёму оперативной памяти. Поэтому принято решение отказаться от использования глобальных схем по корпусу документов в пользу одной из локальных схем, а конкретно TF. Проблема же часто встречаемых, не несущих смысловой нагрузки слов, решена в процессе предварительной обработки текста за счет фильтрации стоп-слов, что в некотором смысле снизило необходимость расчета обратной документной частоты IDF при взвешивании терминов. Таким образом, в документе вес каждого термина рассчитывался на основе статистических методов по частоте появления термина в тексте по формуле (1.8).

Представлением документа и поисковым образом ИР является контекстный вектор, элементы которого формируются на этапе автоматической обработки текста и хранятся в онтологии одновременно со списком ключевых слов, что позволяет многократно их использовать. В случае необходимости пользователь может изменить термины и пересчитать вектора в модуле редактора онтологий.

Укрупненный алгоритм формирования векторного представления текста в виде контекстного вектора V(k) следующий:

1. Из модуля извлечения информации передаются все необходимые данные после автоматического разбора текста.
2. Формируется вектор из ключевых слов, полученный с помощью частотного анализа.
3. Вектор дополняется термами из онтологии.
4. Поисковый запрос расширяется терминами со сторонних ресурсов (WordNet, Wiktionary).
5. В случае необходимости корректируется экспертом или автором ресурса полученный вектор.
6. Для каждого термина вычисляется TF по формуле (1.8).
7. Элементы вектора сортируются в зависимости от значения TF.
8. Вектор ограничивается по количеству элементов k (параметр k можно изменять).

9. Полученный контекстный вектор сохраняется в онтологии в дальнейшем является векторным представлением документа.

10. Попутно наполняется онтология новыми терминами из массива ключевых слов.

Каждый документ соотносится при помещении в онтологию с определенной темой, что позволяет сформировать из контекстных векторов документов контекстные векторы тематических пространств и использовать их при поиске и классификации документов.

Для представления онтологии была использована разработанная в разделе 3.3.1 модель N-мерного пространства представления семантико-синтаксических отношений RDF-графа, где отношения моделируются как тройки (субъект, предикат, объект) и где предикат либо определяет отношения между двумя сущностями, либо отношения между сущностью и значением атрибута. При формировании тензора семантической близости для разных типов предикатов меры семантической близости были рассчитаны по-разному. Среди мер были как меры, вычисляемые по онтологии (таксономические и атрибутные), так и меры, вычисляемые по векторной модели представления текста.

Работа с тензорами не вызывает особых затруднений. Тензорная алгебра представлена большим количеством различных операций для работы с тензорами, среди которых выделим сложение, вычитание, свертка, тензорное произведение и тензорное разложение различных видов [126]. Существует и достаточное количество инструментальных средств и библиотек, предназначенных для работы с тензорной алгеброй, в том числе и для Python [127].

Поскольку вместо матрицы семантических связей «термина-на-термины» размерности $n_x n$, состоящей из контекстных векторов n-мерного пространства мы использовали трехмерное представление в виде семантического тензора «концепт-концепт-предикат» размерности $[n_x n_x p]$, то необходимо модифицировать модель bag-of-concepts, обобщив ее для использования в пространствах третьего порядка.

В качестве оператора перехода из одного пространства дескрипторов в другое использовался 3-х мерный тензор семантических связей S_p^{nn}. Выполнив

аддитивную свертку тензора T_p^{nn} по формуле (3.1) получили значение обобщенной семантической матрицы R^{nn}.

Пусть текст представлен контекстным множеством в виде ковариантного тензора первого ранга V_n. Полученный семантический тензор второго порядка R^{nn} позволяет отобразить V_n – исходное представление текста в новое представление контравариантного тензора первого ранга U^n по формуле (1.1) – уже отражающее семантические связи между словами:

$$U^n = V_n * R^{nn} \quad (3.2)$$

Но проблему большой вычислительной сложности это не устранило, так как в больших онтологиях количество концептов велико и загрузить в память исходный тензор для выполнения дальнейших операций проблематично.

Поскольку многие алгоритмы поиска и классификации информации чрезвычайно чувствительны ко времени вычисления, то проблеме уменьшения размерности пространства уделялось внимание еще на этапе извлечения знаний с помощью модуля автоматического разбора текста. Но эти меры не привели к значительному снижению размерности. Чтобы справиться с большим количеством RDF-троек, а также с разреженностью данных, была использована техника редукции признакового пространства, т.е. переход к пространствам более низких порядков.

Данные в системе управления информационными ресурсами на примере кафедры вуза имеют в основном вполне допустимую размерность (1000-2000 текстовых документов). Несмотря на незначительное количество документов, размерность матрицы, полученной после векторизации текстов, может быть довольно значительной для хранения её в памяти ЭВМ (7000-50000 индексированных термов). Следует предусмотреть и увеличение размерности задачи хотя бы до 5000 документов.

Снижение размерности следует проводить до размерности, позволяющей осуществлять обработку без существенного ухудшения качества поиска (до 10%).

Стандартные подходы снижения размерности исходного признакового пространства могут быть разделены на два больших класса: с трансформацией

признакового пространства (PCA, SVD, NMF); без трансформации исходного пространства с исключением неинформативных признаков.

PCA (анализ главных компонент) – один из часто используемых методов сокращения размерности, однако признаки PCA трудно интерпретировать и метод PCA непредсказуем в выборе координат объектов в новом пространстве.

Еще одним часто используемым методом, применяемым для снижения размерности, является SVD разложение на 3 компонентные матрицы.

Неотрицательное матричное разложение NMF предусматривает разложение в произведение двух матриц меньшего размера, при этом элементы этих матриц неотрицательны. Результат – признаки, которые лучше интерпретируются и могут иметь больше смысла по сравнению с разложением SVD. Проблема самих методов SVD и NMF заключается в том, что это очень медленные и ресурсоемкие алгоритмы.

Среди подходов с уменьшением признаков без трансформации пространства весьма перспективным представляется использование тематических разделов и контекстных векторов онтологии для решения задачи снижения размерности.

Часть проблем, связанных с производительностью и ресурсоемкостью системы, решилась с помощью эффективно полученных представлений. В этом случае в процессе обработки мы работаем не с самими, иногда очень объёмными ИР, а с их структурированными представлениями, сохраненными в онтологии.

В процессе разработки далее были разработаны две редуцированные модели представления текстовых информационных ресурсов. Первая модель без трансформации признакового пространства получила название ONTO, вторая с трансформацией признакового пространства получила название TEN.

Для модели ONTO семантический образ каждого тематического раздела формируется на базе семантических образов отдельных концептов, принадлежащих данному разделу, и, соответственно, представляет собой частоты слов TF в ИР.

Пусть пользователь сформулировал некоторый запрос Q_k.

Тензор 2-го ранга Z_t^k задает соответствие между концептами, являющимися ключевыми словами и темами.

Редукция была выполнена по тематике из тензора Z_t^k, наиболее соответствующей пользовательскому запросу Q_k. Для определения подходящей тематики сначала выполнили аддитивную свертку тензора Z_t^k с тензором первого ранга поискового запроса Q_k. Тема с максимальным значением вектора Y_t была выбрана, чтобы участвовать в редукции:

$$Y_t \cong Z_t^k{}_* Q_k. \quad (3.3)$$

Сформированный путем отбора тематически редуцированный тензор 3-го ранга G_p^{kk} содержал только семантически близкие термины (Рисунок 3.9).

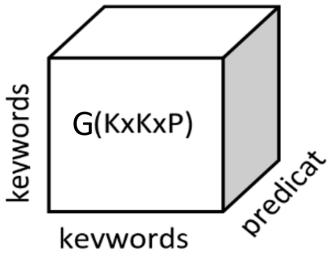


Рисунок 3.9 – Тематический редуцированный тензор

Для модели TEN редукцию выполнили с помощью подхода, основанного на тензорной факторизации. Информационное пространство наполнено работами по технологии факторизации, включая работы по латентному семантическому анализу (Deerwester et al., 1990), по вероятностным вариантам LSA (Hofmann, 1999), по анализу основных компонентов и вероятностным версиям PCA (Buntine & Perttu, 2003; Tipping & Bishop, 1999) и по неотрицательной матричной факторизации (NMF) (Paatero & Tapper, 1994; Lee & Seung, 1999) [128].

Факторизация N-мерного тензора генерирует N матриц, состоящих из к-векторов для каждого измерения скрытого семантического пространства, и служит уникальным средством моделирования и выявления взаимосвязей и совместного поведения n переменных в массиве n-мерных данных [129]. Факторизация данных в более низкое размерное пространство вводит компактный базис, который при

соответствующей настройке может описать исходные данные в сжатой форме, ввести некоторую невосприимчивость к шуму.

В некотором смысле метод неотрицательной факторизации тензоров можно назвать n-мерным обобщением SVD, так как SVD матричное разложение для тензора второго порядка является частным случаем тензорного разложения [130]. В работе [131] также отмечается, что единого способа обобщения SVD на тензоры 3-го и более порядка не существует.

Существует большое количество типов тензорных разложений – CANDECOMP/PARAFAC, TUCKER, INDSCAL, PARAFAC2, CANDELINC, PARATUCK2, а также их вариаций, например, TUCKER2, графическая интерпретация которого приведена на рисунке 3.10.

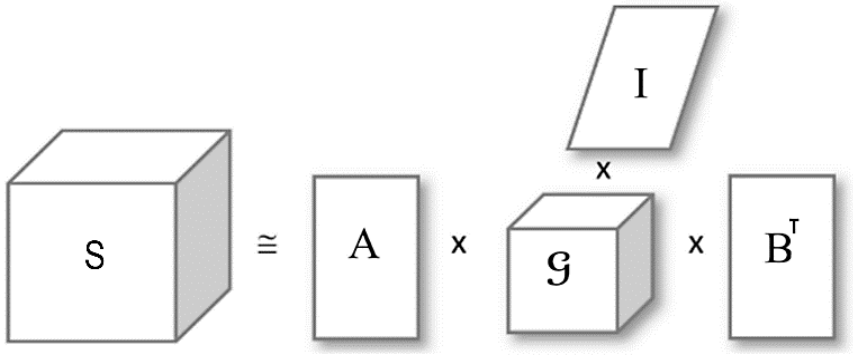


Рисунок 3.10 – Тензорное разложение Tucker2

Воспользовавшись вариацией разложения Tucker2 RESCAL, для которого одна из матриц разложения единичная (I), а две другие матрицы (A, B) равны между собой, получаем:

$$S \cong A \times_1 G \times_2 A^T = [G; A, A^T, I] \quad (3.4)$$

В индексном виде, исходя из параметров тензоров S и G:

$$S_{oop} \cong \sum_{k=1}^{K} \sum_{k=1}^{K} \sum_{p=1}^{P} A_{ok} G_{kkp} A_{ok}^{T}, \quad (3.5)$$

где: S(OxOxP) – исходный тензор 3-го ранга; O – количество терминов; P – количество отношений; G(KxKxP) – ядро, тензор 3-го ранга, A – тензор 2-го ранга; K – количество ключевых терминов, K <<O.

Поскольку K намного меньше по значению, чем O, то тензор G можно рассматривать как сжатую версию S, что существенно уменьшает размерность пространства.

3.4.3 Тестирование разработанных моделей представления текстов

Перед реализацией программных модулей выполнен ряд экспериментов на прототипах для тестирования разработанных алгоритмов и моделей. Прототипы разработаны в программной среде Python 3.8 на базе нескольких стандартных библиотек.

Для проведения экспериментов использованы собственные ИР – материалы конференций факультета за последние 10 лет в объеме более 1000 документов. С помощью модуля автоматической обработки тексты документов были распарсены, получены контекстные множества и векторные представления документов, которые сохранены в онтологии.

Сравнивались алгоритмы, построенные на базе двух моделей: ONTO – тематически редуцированной модели и TEN – обобщенной на трехмерный случай модели, построенной на базе тензорного разложения Такера.

В процессе экспериментов изменялись такие параметры, как количество концептов модели, количество семантических мер модели, размер контекстного вектора (от 0 до 100) с целью оценить, насколько эти параметры влияют на скорость выполнения базовых операций: ввода и обработки данных.

Основные параметры модели опишем в таблице 3.4.

Таблица 3.4 – Параметры и переменные модели

№	Параметр	Описание параметра
1	o	Количество объектов, извлеченных из RDF- хранилища
3	p	Количество типов предикатов, извлеченных из RDF- хранилища
4	k	Длина контекстного вектора
5	w	Контекстный вектор

Результаты экспериментов, показывающие, как влияют на скорость выполнения базовых алгоритмов операции с многомерным тензором, сведены в таблицу 3.5.

Таблица 3.5 – Результаты экспериментов

p	Время выполнения алгоритмов для o=5000 к=30**, *s***	
	Ввод данных TEN	**Обработка данных TEN**
1	0,1456	0,076
2	8,456	0,876
3	10,8002	1,652
4	10,8972	1,887
5	20,8992	2,09
6	20,0997	2,431
7	20,4991	2,786
8	30,399	3,876

Можно сделать вывод, что при количестве концептов равном 5000 и длине контекстного вектора равном 30 изменение параметра p от 1 до 8 не влияет критично на производительность системы.

Результаты исследований, показывающие, как влияет количество входных термов, а, следовательно, и размеры исходного тензора, на скорость выполнения базовых операций (ввода данных и обработки данных) сведены в таблицу 3.6.

Таблица 3.6 - Результаты экспериментов

p=8 k=30 o	Время выполнения функциональных блоков алгоритмов			
	Процедура ввода данных, s		**Процедура обработки данных, s**	
	ONTO	**TEN**	**ONTO**	**TEN**
10	0,0005	0,002	0,000001	0,09495
30	0,00059	0,006	0,000001	0,089
50	0,0006	0,05	0,000001	0,09695
100	0,0042	0,09	0,000001	0,0993
750	0,00899	0,1679	0,000002	0,01099
1500	0,01299	0,61916	0,000099	0,006
2250	0,02399	1,56565	0,0001	0,008
3000	0,02699	6,27798	0,001004	0,08495
3750	0,03998	8,40323	0,001004	0,09495
4500	0,03598	13,1394	0,000999	0,64206
5250	0,04	11,9704	0,001000	0,49979
6750	0,06	Ошибка!	0,002099	---------
7500	0,08		0,0023	
8250	0,0899		0,003004	
9000	0,08		0,001904	

Как показал анализ процедур ввода, алгоритм на базе модели ONTO работает значительно быстрее, чем алгоритм TEN. Кроме того, для алгоритма TEN при выполнении операций загрузки с объёмом тензора 6000x6000x8 при тестировании произошло аварийное завершение работы, связанное с переполнением памяти (Рисунок 3.11).

Рисунок 3.11 – Время выполнения процедур ввода при увеличении количества терминов

Выполнено сравнение алгоритмов обработки данных с использованием моделей TEN и ONTO по скорости выполнения. Среднее время обработки по алгоритму с использованием модели TEN на несколько порядков больше, чем при использовании модели ONTO, для которой тестирование выполнялось практически мгновенно (Рисунок 3.12). Таким образом исследования показали, что для дальнейшего использования целесообразней выбрать модель без трансформации признакового пространства ONTO.

Существенное снижение размерности достигнуто за счет использования обобщенного модифицированного векторного представления документа на базе модели bag-of-concepts, онтологического подхода, а также за счет тематической редукции семантического тензора знаний. При снижении размерности от 2250 терминов до 30 процедура ввода выполняется в среднем в 40 раз быстрее, а процедура обработки в 100 раз быстрее.

Рисунок 3.12 – Время выполнения процедур обработки при увеличении количества терминов

Дальнейшие исследования проведены для выбранной модели ONTO. На рисунке 3.13 приведены две зависимости от длины контекстного вектора: общего количества найденных слов в процессе выполнения поиска и количества слов, которые были выбраны программным модулем в качестве ключевых терминов.

Рисунок 3.13 – Зависимости общего количества найденных слов и количества ключевых слов от длины контекстного вектора

На рисунке 3.14 приведена зависимость времени решения задачи поиска от длины контекстного вектора.

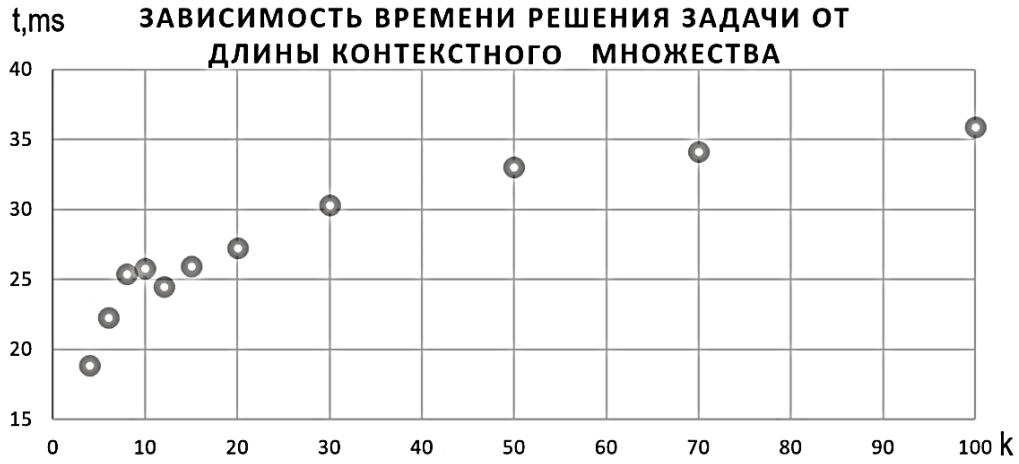


Рисунок 3.14 – Зависимость времени решения задачи поиска от длины контекстного вектора

На рисунке 3.15 приведена зависимость полученных значений меры гибридной семантической близости от длины контекстного вектора.

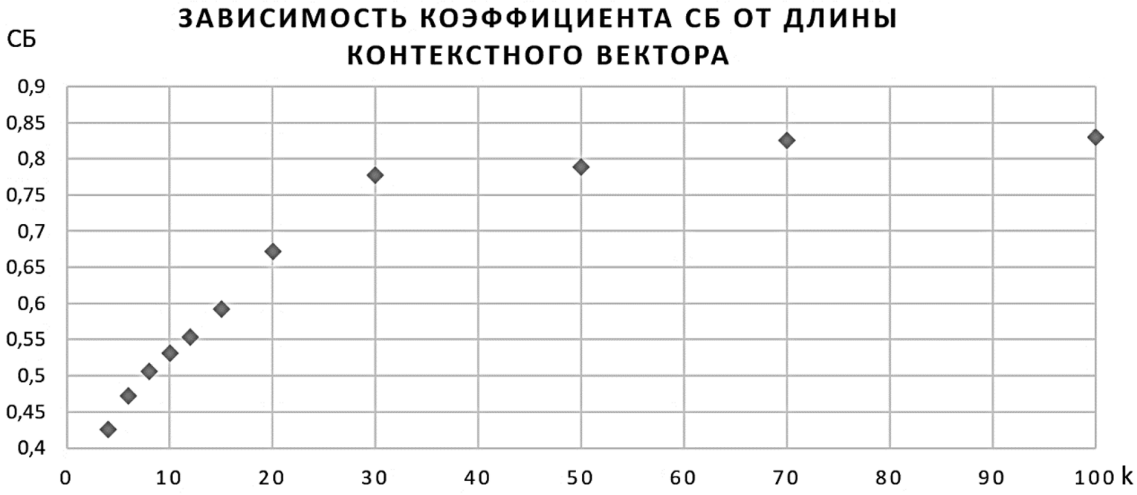


Рисунок 3.15 – Зависимость гибридной меры СБ от длины контекстного вектора

Как видно из графиков 3.13 и 3.14, оптимальным значением k является значение, равное 20. Но, как следует из графика 3.15, при значении k=20 потери значения коэффициента СБ составляют 19.2%, при значениях k=30 уже 6.2%, а при k=50 всего 5%. Таким образом, исследования показали, что целесообразно использовать контекстное множество длиной k=30.

3.4.4 Разработка модели формирования поисковых запросов

Пользователь часто формулирует запрос, который не полностью, в силу различных причин, отражает его потребности. Во-первых, пользователь может не знать специфических терминов по теме запроса, не всегда точно может сформулировать сам поисковый запрос, иногда просто допускает ошибки в тексте запроса. Во-вторых, не менее существенную роль играют в формировании правильного запроса расширение области поиска за счет использования синонимов, а также слов, которые часто встречаются в контексте поисковой фразы.

В разделе 2.3.2 проведены исследования по подбору готовых онтологий для возможности их использования в задаче пополнения онтологии экземплярами. Базируясь на этих исследованиях, для расширения запроса и устранения ошибок в тексте запроса были использованы те же словари WordNet, Russnet, Wiktionary. С точки зрения удобства использования API и скорости извлечения данных преимущество закрепилось за словарем Wiktionary, особенно за его локальной версией.

Для семантического соответствия необходимо, чтобы семантические значения запроса и документа были известны. Семантика документа формально и явно определяется в онтологии [132].

Наличие собственной разработанной предметной онтологии позволяет в полной мере воспользоваться обработкой уже структурированных документов и использовать концепты онтологии для расширения поискового запроса. Термином будем называть обозначение слова на естественном языке, а концептами будем называть термины, хранящиеся в онтологии.

Чаще всего используется вся онтология полностью, но возможно и использование определенной тематики для уменьшения размерности. Контекстный вектор тематики образуется суммой всех терминов тематики.

Если СБ между поисковым запросом и контекстным представлением документа в онтологии имеет значение больше порогового значения, то добавляем ключевые термины семантически близкого документа к запросу. Также добавляем

слова из связанных концептов отношениями «is-a», «part-of» и, конечно, «synonym», формируя из полученных терминов множество слов поискового запроса.

Пусть задан запрос $Q=\{q_1, q_2, .. q_o\}$, в котором имеется некоторый термин q_i. Модель формирования семантически обогащенного поискового запроса с получением наиболее близких по смыслу слов к термину q_i:

1. Поисковый запрос расширяется терминами сторонних ресурсов (WordNet, Wiktionary), параллельно наполняя онтологию новыми концептами с помощью специальной процедуры.
2. Поисковый запрос наполняется множеством понятий из онтологии.
3. Выполняется расчет весов терминов по мере TF и их ранжирование.
4. Формируется вектор запроса на базе контекстных векторов терминов.

Таким образом, использование онтологического подхода обеспечило переход к новому уровню в решении задач поиска информации и при формировании поисковых запросов позволило:

- устранить опечатки и ошибки;
- расширить область поиска за счет подбора и ранжирования терминов с помощью готовых словарей и веб-ресурсов;
- расширить область поиска за счет поиска терминов из собственной прикладной онтологии системы, что особенно важно для документов, имеющих достаточно стандартную структуру, и улучшить качество поиска;
- оперировать не словами (plain words), а смыслами (senses of words), т.е. перейти к новому семантическому уровню представления данных.

3.4.5 Разработка модели получения релевантных данных по запросу

Извлечение релевантных данных по запросу выполняется в два этапа: поиск релевантных семантически близких данных и ранжирование их относительно запроса. Модель получения релевантных данных по запросу построена на базе

алгоритма определения СБ и функции ранжирования, которая упорядочивает документы относительно запроса.

Пусть задан вектор запроса Q. Для создания эффективного алгоритма извлечения релевантных данных по запросу необходимо искать только те документы, которые лежат в той же области пространства понятий, что и запрос. Сначала определяем близость поискового запроса к тематическим векторам, затем процесс повторяем, сравнивая контекстный вектор запроса и документов. Тематические разделы описаны в онтологии и имеют свои контекстные множества и вектора длиной k.

Разработанный алгоритм извлечения релевантных данных по запросу:

1. Определяем тематический раздел по формуле (3.3), получаем тематически редуцированный тензор.

2. Получаем значение обобщенной семантической матрицы R^{kk} путем аддитивной свертки семантического тензора по третьему измерению (p) по формуле (3.1).

3. Получаем семантически обогащенный поисковый образ документа:

$$C^k = Q_k R^{kk} \tag{3.6}$$

где: R – матрица семантических связей терминов; k – число ключевых терминов; Q – поисковый запрос.

4. Получаем СБ запроса к другим документам тематического раздела:

$$X^l = \mathrm{C}_k B^{lk} \tag{3.7}$$

где: B(lxk) – тензор 2-го ранга, хранящий контекстные вектора терминов тематического раздела; k – число ключевых терминов; l – общее число терминов тематического раздела, при чем l <=o; если o – общее количество терминов онтологии; C – поисковый образ документа.

5. Определяем по функции F (X, Y) релевантные запросу документы (раздел 3.3). Чем больше значение СБ между поисковым образом документа и представлением документа, тем выше позиция документа в финальной поисковой выдаче.

Для идеальной поисковой системы список отобранных документов и соответствующих документов должны совпадать. В реальных поисковых системах в списках отобранных документов находятся и несоответствующие запросу документы, и отсутствуют некоторые соответствующие.

Эффективность поиска определяется, по крайней мере, двумя основными показателями – точностью и полнотой, и четырьмя дополнительными – специфичностью, избирательностью, коэффициентом потери информации и коэффициентом поискового шума [133].

В реальных информационных системах полнота поиска составляет 60…70%, а точность – 40 … 50% [134]. По другим данным, полнота поиска составляет 70…90%, а коэффициент точности обычно находится в пределах 10…100% [135].

3.4.6 Тестирование модели получения релевантных данных по запросу

Релевантность результата поиска в первую очередь зависит от используемой меры определения СБ. Для оценки этого параметра была проанализирована поисковая выдача по различным поисковым словам, относящимся к разным темам.

Поисковое соответствие определяет долю соответствующих документов среди отобранных на запрос. Поисковое качество определяет долю полученных системой соответствующих запросу документов среди общего числа соответствующих запросу документов в коллекции. Общее число соответствующих запросу документов всегда неизвестно и может быть установлено только экспертом.

Для тестирования модели была проанализирована размеченная экспертом коллекция документов, представляющих собой материалы конференций факультета по трем разделам конференции с приблизительно одинаковым количеством работ в каждом.

Результаты представлены в таблице 3.7 в виде матрицы несоответствий (Таблица 1.3 раздела 1.4.5).

Таблица 3.7 – Матрица несоответствий (confusion matrix)

Темы для поиска	КМ	ИИ	ВЕБ
Компьютерное моделирование (КМ)	55	14	2
Искусственный интеллект (ИИ)	6	47	4
ВЕБ	3	0	39

На основе данных таблицы 3.7 были выполнены расчеты различных показателей качества поиска, таких, как точность (Accuracy), полнота (Recall), точность (Precision), вероятность нахождения релевантного ресурса (TPR), вероятность нахождения нерелевантного ресурса (FPR), F-мера (F) и оформлены в виде таблицы 3.8.

Таблица 3.8 – Показатели качества поиска при гибридной модели СБ

Темы	TP	TN	FP	FN	Acc	Recall	Prec	F	TPR	FPR
КМ	55	90	16	9	0,85	0,86	0,77	0,81	0,85	0,15
ИИ	47	99	10	14	0,86	0,77	0,82	0,80	0,91	0,09
ВЕБ	39	122	3	6	0,95	0,87	0,93	0,90	0,98	0,02
Среднее					0,89	0,83	0,84	0,83	0,91	0,09

Для сравнения была проанализирована базовая модель поиска, построенная на векторной модели представления текста bag-of-words и косинусной мере определения семантической близости. Результаты эксперимента представлены в таблице 3.9.

Таблица 3.9 – Показатели качества поиска при косинусной мере СБ

Темы	TP	TN	FP	FN	Acc	Recall	Prec	F	TPR	FPR
КМ	45	97	16	12	0,84	0,79	0,74	0,76	0,86	0,14
ИИ	30	120	15	5	0,88	0,86	0,67	0,75	0,89	0,11
ВЕБ	39	102	26	3	0,83	0,93	0,60	0,73	0,80	0,20
Среднее					0,85	0,86	0,67	0,75	0,85	0,15

В результате исследований получили, что качество поиска разработанной модели получения релевантных данных запросу, выраженное значением F – меры, в среднем составляет 0.83, что на 10.7% больше, чем для базовой модели (0.75).

3.5 Разработка моделей, методов и алгоритмов уровней интеграции и хранения данных

3.5.1 Разработка модели классификации

Без классификации и систематизации знаний немыслимо эффективное хранение знаний с целью обеспечения эффективного поиска.

Существует два основных подхода к решению задачи классификации. Первый подход базируется на методах машинного обучения, а второй на методах инженерии знаний, которая представляет собой раздел искусственного интеллекта. Каждый подход имеет свои преимущества и недостатки, области применения, особенности и ограничения. Эффективность же отдельных методов зависит от конкретной решаемой задачи.

Процедура классификации заключается в том, чтобы установить соответствие между ИР и тематическим разделом, при этом один ИР может принадлежать нескольким разделам, т.е. классификация мультиклассовая. Правила соответствия ИР и тематического раздела задаются на основе классификатора. Классификатор представляет собой особый документ, вектор которого формируется на этапе обучения и состоит из усреднённых значений весов терминов, входящих в документы обучающей выборки.

Если для классификации используется онтология предметной области, то вектор документа можно сравнивать с представлениями документов, хранящихся в самой онтологии. Отсюда следует два важных отличия от классических методов машинного обучения.

Первое заключается в том, что применение онтологий позволяет отказаться от этапа обучения классификатора. Описание предметной области в виде онтологии само является классификатором, таким образом, не тратится время и вычислительные ресурсы на построение среднего документа из обучающей выборки.

Второе отличие заключается в том, что при таком подходе в вектор документа включаются только те термины, которые включены в рассматриваемую онтологию. Это значит, что те понятия, которые не входят в набор концептов онтологии, уходят из процесса вычисления весов терминов. Также имеется отличие классификатора в виде онтологии от классификатора в виде усреднённого документа. В обоих случаях классификатор является моделью «эталонного» документа, соответствующего предметной области. Но если он является «усреднённым» документом, то в его состав могли войти термины, употреблявшиеся в документах, но не имеющие отношения к описываемому разделу. В случае онтологии, напротив, классификатор является описанием предметной области без каких-либо лишних для неё понятий. В таком случае этот классификатор является более универсальным с точки зрения использования его в составе различных систем и для различных задач.

В рамках единого подхода, поскольку мы уже разработали онтологическую модель, то для классификации использовали методы, основанные на знаниях.

Пусть имеется множество объектов $D = \{d_1, d_2, \ldots, d_n |D|\}$, представляющих собой электронные версии текстовых ИР. Предполагается, что количество документов в системе относительно мало (около 2000 в среднем) и не превышает 10 000 документов.

Также существует конечное множество классов объектов C = $\{c_1, c_2, \ldots, c_m |C|\}$, в нашем случае тематических разделов. Тематические разделы описаны в онтологии, имеют свои тематические контекстные множества и вектора, а также другие свойства.

Также задана Ф: $D \times C \rightarrow \{0, 1\}$ – некоторая пороговая целевая функция, вычисление которой по паре $<d_i, c_j>$ позволяет определить, принадлежит ли документ d_i категории c_j или нет.

Задача классификации состоит в построении классификатора Ф′: $D \times C \rightarrow \{0, 1\}$, максимально близкого к Ф.

Для решения задачи классификации и соотнесения ИР к тематическому разделу (рубрики для классификации определены и хранятся в онтологии)

разработан алгоритм классификации, который выполняется в два этапа: определение максимально подходящей темы и определение максимально близкого документа в пределах темы для сохранения документа в нужное место:

1. Определение тематического раздела для классификации производим по формуле (3.3) и в результате получаем тему, максимально подходящую документу.
2. Рассчитываем обобщенную матрицу R^{kk} семантической близости через факторизацию срезов тензора по формуле (3.1).
3. Получаем семантически обогащенное представление документа C^k по формуле (3.6).
4. Получаем СБ документа к другим документам X^l по формуле (3.7).
5. Находим документ с максимальным весом X^l, и соответственно, определяем место для нового документа.
6. Пополняем онтологию метаданными и помещаем новый документ в хранилище.

3.5.2 Оценка качества классификации

Задача классификации заключалась в распределении статей факультетских конференций по тематическим разделам.

Чтобы доказать эффективность разработанного алгоритма классификации для сравнения были протестированы еще три алгоритма, построенные на различных способах определения степени соответствия документа тематическому разделу:

- алгоритм определения СБ по косинусному сходству;
- алгоритм определения СБ по мягкому косинусному свойству, использующему матрицу семантической близости на базе онтологии;
- алгоритм определения СБ по контекстному множеству;
- алгоритм определения СБ по интеллектуальному гибридному способу оценки СБ, описанному в разделе 3.3.

Результаты тестирования первого алгоритма представлены в таблице 3.10 в виде Confusion matrix (матрица несоответствий) для случая многоклассовой классификации. Показатели качества классификации первого алгоритма приведены в таблице 3.11.

Таблица 3.10 – Confusion matrix (матрица несоответствий) для первого алгоритма определения СБ по косинусному сходству

Cosine	ИУС	КМ	ИИ	ВЕБ	КС+АСУ
Информационные управляющие системы (ИУС)	276	9	6	5	13
Компьютерное моделирование (КМ)	9	75	2	2	6
Искусственный интеллект (ИИ)	2	5	28	6	7
ВЕБ	15	5	2	90	3
Системы управления (КС+АСУ)	13	20	7	10	184

Таблица 3.11 – Результаты классификации первого алгоритма определения СБ по косинусному сходству

Разделы	TP	TN	FP	FN	Acc	Recall	Prec	F	TPR	FPR
ИУС	276	452	39	33	0,91	0,89	0,88	0,88	0,92	0,08
КМ	75	667	19	39	0,93	0,66	0,80	0,72	0,97	0,03
ИИ	28	735	20	17	0,95	0,62	0,58	0,60	0,97	0,03
ВЕБ	90	662	25	23	0,94	0,80	0,78	0,79	0,96	0,04
КС+АСУ	184	537	50	29	0,90	0,86	0,79	0,82	0,91	0,09
Среднее по коллекции					0,93	0,77	0,77	0,76	0,95	0,05

У разделов КМ и ИИ показатели качества несколько хуже, чем у остальных. Возможно, это связано с различным количеством данных в коллекциях разделов, либо с тем фактом, что разделы очень близки по смыслу.

Тем не менее, для системы, которая поддерживает экспертный режим и не предъявляет чрезвычайно высоких требований к точности, результаты вполне удовлетворительные.

Результаты тестирования второго алгоритма определения СБ по мягкому косинусному свойству представлены в таблице 3.12. Показатели качества классификации второго алгоритма приведены в таблице 3.13.

Таблица 3.12 – Confusion matrix (матрица несоответствий) для второго алгоритма определения СБ по мягкому косинусному свойству

Soft cosine	ИУС	КМ	ИИ	ВЕБ	КС+АСУ
Информационные управляющие системы	272	11	5	18	12
Компьютерное моделирование (КМ)	4	54	3	4	8
Искусственный интеллект (ИИ)	12	8	32	6	8
ВЕБ	3	3	2	91	3
Системы управления (КС+АСУ)	17	21	7	12	184

Таблица 3.13 – Результаты классификации второго алгоритма определения СБ по мягкому косинусному свойству

Разделы	TP	TN	FP	FN	Acc	Recall	Prec	F	TPR	FPR
ИУС	272	446	36	46	0,90	0,86	0,88	0,87	0,93	0,07
КМ	54	684	19	43	0,92	0,56	0,74	0,64	0,97	0,03
ИИ	32	717	34	17	0,94	0,65	0,48	0,56	0,95	0,05
ВЕБ	91	658	11	40	0,94	0,69	0,89	0,78	0,98	0,02
КС+АСУ	184	528	57	31	0,89	0,86	0,76	0,81	0,90	0,10
Среднее по коллекции					0,92	0,72	0,75	0,73	0,95	0,05

Для второго тестирования результаты в целом аналогичны. По-прежнему, у двух разделов КМ и ИИ показания качества хуже, чем у остальных.

Результаты тестирования третьего алгоритма определения СБ по контекстному множеству представлены в таблице 3.14.

Результаты классификации и итоговые показатели качества классификации третьего алгоритма определения СБ по контекстному множеству приведены в таблице 3.15.

Тестирование третьего алгоритма на основе контекстного множества показало самые плохие результаты, хуже, чем у двух предыдущих и со значением показателей качества на грани допустимого. Вероятно, семантическая близость всех документов набора между собой повлияла на результаты или плохая степень наполнения онтологии экземплярами повлияла на качество классификации. Данный алгоритм нельзя рекомендовать для классификации без усовершенствования.

Таблица 3.14 – Confusion matrix (матрица несоответствий) для третьего алгоритма определения СБ по контекстному множеству

Context	ИУС	КМ	ИИ	ВЕБ	КС+АСУ
Информационные управляющие системы	148	17	23	21	85
Компьютерное моделирование (КМ)	13	55	14	2	7
Искусственный интеллект (ИИ)	15	3	39	0	12
ВЕБ	8	6	4	47	37
Системы управления (КС+АСУ)	26	19	22	7	170

Таблица 3.15 – Результаты классификации третьего алгоритма определения СБ по контекстному множеству

Разделы	TP	TN	FP	FN	Acc	Rec	Prec	F	TPR	FPR
ИУС	148	444	62	146	0,74	0,50	0,70	0,59	0,88	0,12
КМ	55	664	36	45	0,90	0,55	0,60	0,58	0,95	0,05
ИИ	39	668	30	63	0,88	0,38	0,57	0,46	0,96	0,04
ВЕБ	47	668	55	30	0,89	0,61	0,46	0,53	0,92	0,08
КС+АСУ	170	415	74	141	0,73	0,55	0,70	0,61	0,85	0,15
Среднее по коллекции					0,83	0,52	0,61	0,55	0,91	0,09

Результаты тестирования последнего алгоритма определения СБ по интеллектуальному гибридному способу представлены в таблице 3.16.

Таблица 3.16 – Confusion matrix (матрица несоответствий) для алгоритма определения СБ по интеллектуальному гибридному способу

Hybrid	ИУС	КМ	ИИ	ВЕБ	КС+АСУ
Информационные управляющие системы	286	1	2	6	7
Компьютерное моделирование (КМ)	9	58	8	0	2
Искусственный интеллект (ИИ)	10	1	60	0	3
ВЕБ	7	4	2	60	28
Системы управления (КС+АСУ)	22	8	9	2	205

Результаты классификации и итоговые показатели качества классификации алгоритма определения СБ по интеллектуальному гибридному способу приведены в таблице 3.17.

Таблица 3.17 – Результаты классификации последнего алгоритма определения СБ по интеллектуальному гибридному способу

Разделы	TP	TN	FP	FN	Acc	Rec	Prec	F	TPR	FPR
ИУС	286	450	48	16	0,92	0,95	0,86	0,90	0,90	0,10
КМ	58	709	19	14	0,96	0,81	0,75	0,78	0,97	0,03
ИИ	60	705	14	21	0,96	0,74	0,81	0,77	0,98	0,02
ВЕБ	60	691	41	8	0,94	0,88	0,59	0,71	0,94	0,06
КС+АСУ	205	514	41	40	0,90	0,84	0,83	0,84	0,93	0,07
Среднее по коллекции					0,93	0,84	0,77	0,80	0,95	0,05

Если рассмотреть протоколы исследований, то показательными являются случаи, когда за счет взвешивания различных мер по гибридному способу получаем правильный результат классификации даже в тех случаях, когда все остальные алгоритмы показали неверный результат (Таблица 3.18).

Таблица 3.18 – Фрагменты протокола исследования

Файл	Expert	Cosine	Soft cosine	Context	Hybrid
1_Двойкин	ИУС	ИУС	ИУС	ИУС	ИУС
1_Демина	ИУС	ИУС	ИУС	ИУС	ИУС
1_Квитко	ИУС	ИУС	ИУС	ИУС	ИУС
1_Керенцева	ИУС	ИУС	ИУС	КС+АСУ	КС+АСУ
1_Киричек	ИУС	ИУС	ИУС	ИУС	ИУС
1_Коваленко	ИУС	ИУС	ИУС	ВЕБ	ИУС
1_Матях	**ИУС**	**ИУС**	**КМ**	**ВЕБ**	**ИУС**
1_Осипова	**ИУС**	**ИУС**	**ВЕБ**	**КС+АСУ**	**ИУС**
1_Струков	**ИУС**	**ИУС**	**КМ**	**КС+АСУ**	**ИУС**
1_Бакшевник	**ИУС**	**КМ**	**КМ**	**ИИ**	**ИУС**
1_Даниев	**ИУС**	**КМ**	**КМ**	**КС+АСУ**	**ИУС**
1_Карпук	**ИУС**	**ВЕБ**	**ВЕБ**	**КС+АСУ**	**ИУС**
1_Карпунов	**ИУС**	**ВЕБ**	**ВЕБ**	**КС+АСУ**	**ИУС**
..		..	..	..	..

Для алгоритма с использованием гибридной меры оценки СБ качество классификации, оцененное F-мерой, в среднем достигло 0.8, что для столь близко расположенных между собой тематических разделов является вполне приемлемым результатом.

С одной стороны, применение онтологии позволило не проводить весьма затратное обучение классификатора документов на обучающей выборке. С другой стороны, результаты показали, как сильно качество классификации зависит от качества и полноты составленной онтологии. Тем не менее, благодаря использованию интеллектуальной гибридной меры, в комплексе учитывающей различные меры оценки СБ, недостатки, связанные с качеством онтологии, были нивелированы за счет использования других мер.

3.6 Выводы

1. Разработана концептуальная метамодель для учета связанности знаний и обеспечения однородности представления данных в рамках единой тематики проектируемой системы, ядром которой является онтология.
2. Модифицирована модель N-мерного представления знаний на базе RDF-графа в виде трехмерного тензора семантических связей, значения которого определяются различным образом для различных отношений RDF-графа знаний и содержат значения в диапазоне [0; 1], что позволило для отдельных видов отношений учитывать не только наличие связей, но и их численное значение.
3. Усовершенствована гибридная мера оценки семантической близости на базе модифицированной модели N-мерного представления RDF-графа знаний, что дало возможность определять сходство с учетом семантики, частотных характеристик текста, контекста и структуры онтологии и улучшить качество поиска разработанной модели получения релевантных данных по запросу, выраженное значением F-меры, на 10.7% по сравнению с базовой моделью. Для определения весовых коэффициентов интеллектуальной гибридной меры оценки семантической близости использовался генетический алгоритм.
4. Разработаны две редуцированные модели представления текстовых информационных ресурсов: без трансформации признакового пространства и с трансформацией признакового пространства. Экспериментальные исследования показали целесообразность использования модели без трансформации

признакового пространства, которая использует редукцию по тематическим разделам онтологии и снижает размерность задачи до размера контекстного вектора тематического раздела, что позволило для количества концептов онтологии, равном 2250 объектов, и размеру контекстного вектора, равному 30 элементов, уменьшить вычислительную сложность более чем в 40 раз практически без потерь качества поиска (не более 6.2%) по сравнению с базовой моделью bag-of-concepts.

5. Усовершенствованы алгоритмы поиска и классификации данных с использованием онтологии, гибридной меры оценки СБ и векторной модифицированной модели представления текстов, что привело к повышению качества модели классификации, выраженного F-мерой (0.8), примерно на 45.4% по сравнению с базовой моделью bag-of-concepts (0.55), на 5.3% по сравнению с алгоритмом, использующим меру «косинусного сходства» (0.76) и на 9.5% по сравнению с алгоритмом на базе «мягкой косинусной меры» (0.73).

РАЗДЕЛ 4
РАЗРАБОТКА СИСТЕМЫ УЧЕТА ИНФОРМАЦИОННЫХ РЕСУРСОВ КАФЕДРЫ

4.1 Разработка структурной архитектурной модели фреймворка

Процесс разработки систем начинается с этапа построения архитектурных моделей. При выполнении архитектурного проектирования определяется непосредственно архитектура информационной системы – концепция, задающая структуру, выполняемые функции и взаимосвязь компонентов ИС.

При разработке архитектурных моделей был взят за основу архитектурный стиль (Architecture-driven), потому что этот стиль лишен недостатков, присущих стилю, основанному на управлении требованиями. Использование архитектурного стиля позволяет реализовать инкрементное и итеративное проектирование, т. е. оперативно изменять существующую и добавлять новую функциональность.

Для подробного описания архитектуры инструментальных комплексов и различных систем воспользуемся аппаратом формализации. Создание формальных моделей – сложный и трудоемкий процесс, тем более архитектурных моделей. Одним из известных и широко распространенных является подход к описанию архитектуры с использованием видов (представлений).

Филипп Крухтен описал модель представления архитектуры программного обеспечения «4+1» для описания архитектуры сложных программных систем в 1995 году [136]. Дальнейшее развитие эта модель получила в разработках модели Rational Unified Process, а также в стандарте IEEE 1471-2000/ISO 42010.

Взяв за основу известную архитектурную модель «4+1», модифицируем ее, уделив в первую очередь внимание структурным и статическим аспектам построения ИС. Диаграммы модели представляют собой средство для визуализации.

Описание полученной архитектурной модели сведем в таблицу 4.1.

Таблица 4.1 – Описание архитектурной модели

Виды архитектурной модели	Диаграммы модели
Component view – это вид, описывающий структуру системы на уровне компонентов, в качестве которых могут выступать пакеты, файлы, библиотечные и пакетные файлы, а также разработанные программные модули и функциональные модули.	**1. Component diagram** **2. Structural model** Обобщенная структурно-функциональная модель
Use Case View – описывает поведенческие механизмы системы в терминах вариантов использования с точки зрения внешних по отношению к системе действующих лиц и функциональность системы в целом.	Диаграмма вариантов использования **use case diagram**
Deployment view – вид с точки зрения развертывания, показывает связь технических вычислительных средств и размещенных на них программных компонентов.	Диаграмма развертывания **deployment diagram**
Model Management view – дополнительное представление, отражает внутреннюю организацию модели, описывая ее разбиение на пакеты и указывая отношения между ними.	Диаграмма пакетов **package diagram**

Таким образом, архитектурная модель СУИР состоит из ряда формальных моделей, представленных в виде графических диаграмм нотации UML, а также дополнительной Structural diagram, в комплексе описывающих структуру и функциональность разрабатываемой ИС [137].

4.1.1 Разработка функциональной модели фреймворка

Для описания функциональности ИС, взаимодействия пользователя с внешними информационными системами на практике обычно используются USE-CASE диаграммы в нотации UML [138]. Обобщенная функциональная модель включает в себя 16 основных функций и представлена на рисунке 4.1.

С функциями системы работают несколько пользователей: эксперт, администратор знаний, преподаватель и студент. Самым востребованным специалистом на этапе наполнения базы знаний является эксперт. Помощь ему оказывает инженер по управлению знаниями (администратор знаний).

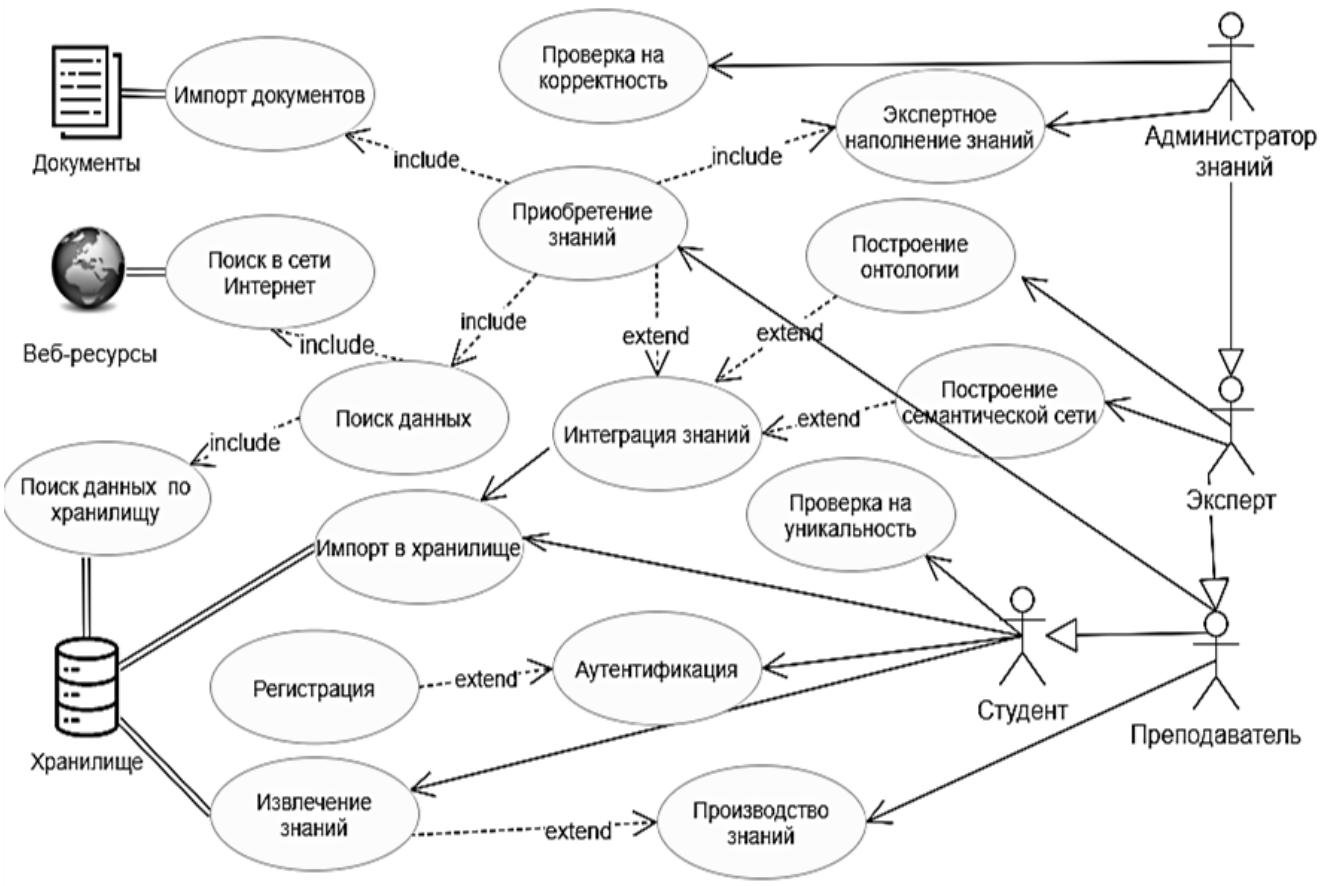


Рисунок 4.1 – Диаграмма вариантов использования (use-case diagram)

Но без участия всего коллектива преподавателей трудно создать качественную базу знаний. Между ролями установлена иерархия пользователей. Это означает, что пользователь с более высоким уровнем прав наследует всю функциональность менее привилегированного пользователя.

4.1.2 Разработка модульной структурной модели фреймворка

Диаграмма пакетов используется, чтобы изобразить зависимости между пакетами, которые составляют модель [139]. Основная цель – показать взаимосвязь между различными крупными компонентами, которые образуют сложную систему. Пакет (package) объединяет некоторые элементы диаграммы UML в единую структурную единицу. В нашем случае пакеты представляют собой отдельные независимые подсистемы, реализующие основные функции системы управления ИР.

Структура системы представляет собой типичную структуру системы управления знаниями (СУЗ), построенную на онтологическом подходе. Основными компонентами предлагаемой технологии создания системы являются: блоки приобретения данных, конвейер для обработки и классификации данных, а также блок выдачи и продукции знаний. Поскольку ядром, базовым компонентом метамодели системы является его онтология, то центральным блоком системы является онтологический редактор, который предназначен для реализации основных операций по работе с онтологиями, в том числе и процедур автоматического и полуавтоматического пополнения знаний.

Модульная структурная модель системы (package diagram) представлена на рисунке 4.2, а назначение и описание основных пакетов сведено в таблицу 4.2.

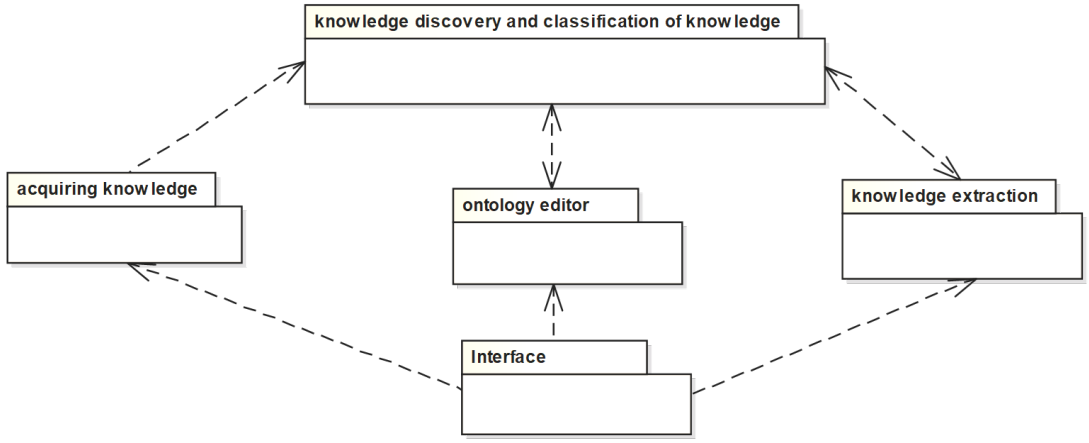


Рисунок 4.2 – Модульная структурная модель системы (package diagram)

Таблица 4.2 – Описание основных пакетов системы

Имя	Пакет	Описание
P_1	acquiring knowledge	Подсистема поиска и выявления знаний предназначена для поиска данных из различных источников, как структурированных данных, так и неструктурированных данных.
P_2	knowledge discovery and classification of knowledge	Подсистема приобретения знаний предназначена для получения знаний, извлечения неструктурированных знаний из разнородных источников информации с помощью методов статистической обработки, семантического анализа, технологий Text Mining и Data Mining, а также экспертных моделей.

Таблица 4.2 (продолжение)

Имя	Пакет	Описание
P_3	knowledge extraction	Подсистема интеграции, хранения и извлечения данных предназначена для организации эффективной работы с хранилищем данных. Основные функции: занесение собранных структурированных материалов, онтологий и извлеченных знаний из данных в интегрированное хранилище, интеллектуальный поиск данных по хранилищу, возвращающий сведения об информационном объекте, т.е. некоторые знания.
P_4	ontology editor	Редактор онтологий – ориентирован на поддержку основных операций для работы с онтологией, решение вопросов импорта и экспорта в различные форматы, а также вопросов синхронизации онтологической модели и структуры хранилища данных.
P_5	Interface	Связующий структурный блок, реализующий интерфейс системы и обеспечивающий взаимное функционирование всех остальных подсистем.

4.1.3 Разработка обобщенной компонентной модели системы

Диаграмма компонентов (Component diagram) – статическая структурная диаграмма, показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами [140].

Component diagram предназначена для моделирования иерархии компонентов, а также модулей, пакетов и подсистем, что позволяет определить структуру разрабатываемой системы (Рисунок 4.3).

В качестве физических компонентов выступают файлы, библиотеки, модули, исполняемые файлы, пакеты. В нашем случае component diagram описывает основные пакеты, описание которых приведено в таблице 4.2, и программные модули, описание которых сведено в таблицу 4.3.

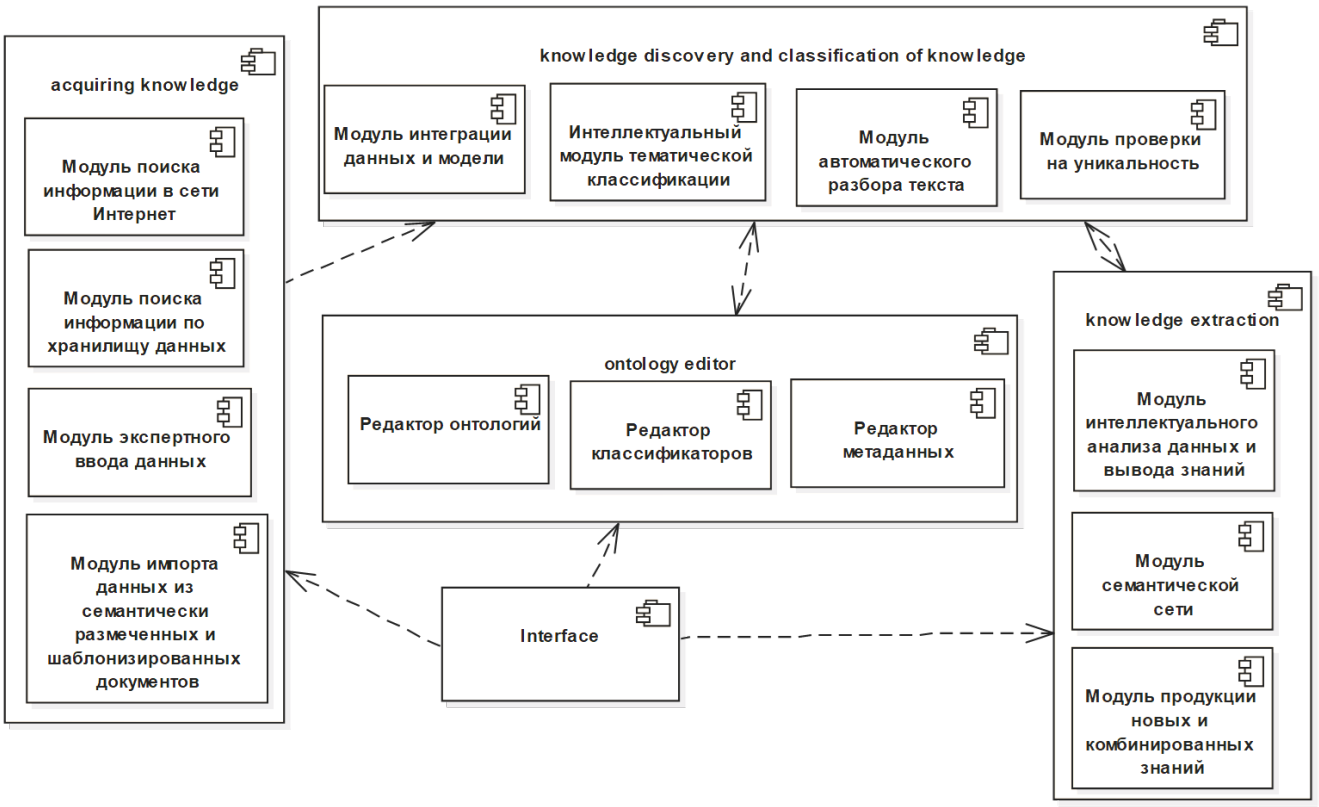


Рисунок 4.3 – Компонентная модель системы (component diagram)

Таблица 4.3 – Список программных модулей системы

№	Пакет	Имя	Описание
1.	P_1	M_1	Модуль поиска информации в сети Интернет
2.	P_1	M_2	Модуль поиска информации по хранилищу данных
3.	P_1	M_3	Модуль экспертного ввода данных
4.	P_1	M_4	Модуль импорта данных
5.	P_2	M_5	Модуль интеграции данных и модели
6.	P_2	M_6	Интеллектуальный модуль VSM и тематической классификации
7.	P_2	M_7	Модуль автоматического разбора текста
8.	P_2	M_8	Модуль проверки на уникальность
9.	P_3	M_9	Редактор онтологий
10.	P_3	M_{10}	Редактор классификаторов
11.	P_3	M_{11}	Редактор метаданных
12.	P_4	M_{12}	Модуль интеллектуального анализа данных и вывода знаний
13.	P_4	M_{13}	Модуль семантической сети
14.	P_4	M_{14}	Модуль продукции новых и комбинированных знаний
15.	P_5	M_{15}	Интерфейс пользователя

4.1.4 Разработка обобщенной структурно-функциональной модели фреймворка

Формализуем обобщенную компонентную модель системы, наполнив ее функциональным содержимым и назовем обобщенной структурно–функциональной моделью системы (**structural model**).

Эта модель представляет собой комбинацию из некоторого конечного множества пакетов P_i, каждый из которых включает в себя некоторый набор программных модулей M_j. В свою очередь каждый j-й модуль реализует некоторое количество функций $F(P_i)$.

Таким образом, **F(MODEL) –** обобщённая функция системы, определяется следующим образом:

$$\mathbf{MODEL= \{P_1, P_2, \dots P_i, \dots P_n\}} \quad (4.1)$$

где i=1...n; P_i – некоторый программный пакет;

В свою очередь, каждый программный пакет состоит из набора некоторых программных модулей:

$$\mathbf{P_i = \{M_1, M_2, \dots M_j, \dots M_m\}} \quad (4.2)$$

где j=1...m; M_j – некоторый программный модуль;

Функциональность каждого программного пакета представляет собой общую функциональность некоторого набора программных модулей:

$$\mathbf{F(P_i)= UF(M_k)}, \quad (4.3)$$

где k=1...l; $F(M_k)$ – функции k-го программного модуля;

Аналогично, функциональность всей системы представляет собой общую функциональность множества некоторых программных пакетов:

$$\mathbf{F(MODEL)= UF(P_i)}, \quad (4.4)$$

где i=1...n; $F(P_i)$ – функции i-го программного пакета;

Сведем описание всех функций ИС в таблицу 4.4.

Таблица 4.4 – Основные функции системы

№	Имя	№	Имя
F_1	поиск по ключевым словам	F_{33}	подсчет кол-во вхождений каждого токена
F_2	поиск с использованием WordNet	F_{34}	формирование топ-списка концептов
F_3	поиск с использованием MediaWiki	F_{35}	атрибутный метод
F_4	поиск с использованием онтологии	F_{36}	структурный метод
F_5	расчет семантической близости концептов	F_{37}	формирование результата
F_6	формирование поисковой выдачи	F_{38}	импорт RDF
F_7	поиск в Elibrary	F_{39}	импорт онтологий OWL
F_8	поиск GoogleAcademy	F_{40}	ввод онтологий вручную
F_9	поиск статей на сайте «Информатика и кибернетика»	F_{41}	конвертация модели
F_{10}	поиск статей на сайте конференции «ИУСКМ»	F_{42}	определение свойств элементов модели
F_{11}	ввод поискового запроса	F_{43}	визуализация структуры
F_{12}	формирование информации	F_{44}	сохранение модели
F_{13}	формирование отчетов	F_{45}	ввод метаданных
F_{14}	ручной ввод новых данных	F_{46}	корректировка метаданных
F_{15}	импорт данных	F_{47}	удаление метаданных
F_{16}	редактирование данных	F_{48}	ввод классификаторов
F_{17}	удаление данных	F_{49}	корректировка классификаторов
F_{18}	сохранение данных	F_{50}	удаление классификаторов
F_{19}	извлечение текстовой информации из docx-файла	F_{51}	предобработка данных
F_{20}	извлечение текстовой информации из pdf-файла	F_{52}	классификация с использованием косинусной меры
F_{21}	извлечение текстовой информации из rtf-файла	F_{53}	классификация с использованием мягкой косинусной меры
F_{22}	извлечение текстовой информации из doc-файла	F_{54}	классификация с использованием онтологий
F_{23}	сохранение данных в хранилище	F_{55}	классификация с использованием гибридной меры
F_{24}	модуль расчета показателей качества модели	F_{56}	подбор материалов по параметрам
F_{25}	интеграция данных	F_{57}	подбор материалов по близости
F_{26}	парсинг текста	F_{58}	отображение текущих данных онтологии
F_{27}	удаление информации, не подлежащей разбору (пунктуационные знаки)	F_{59}	модуль продукции знаний
F_{28}	токенизация (разбитие текста на леммы - слова)	F_{60}	модуль получения интегрированных знаний
F_{29}	расчет векторов (метод «мешок слов»)	F_{61}	обработка меню системы
F_{30}	вывод результатов	F_{62}	работа с личным кабинетом пользователя
F_{31}	сохранение результатов	F_{63}	аутентификация
F_{32}	парсинг текста с помощью регулярных выражений	F_{64}	регистрация

Опишем функциональную модель каждого программного модуля (Таблица 4.5).

Таблица 4.5 – Описание функций программных модулей

№	Описание	Формализация
1	M_1 – множество функций модуля поиска информации в сети Интернет	$F(M_1) = \{F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{10}\}$
2	M_2 – множество функций модуля поиска информации по хранилищу данных	$F(M_2) = \{F_1, F_4, F_5, F_{11}, F_{12}, F_{13}\}$
3	M_3 – множество функций модуля экспертного ввода данных	$F(M_3) = \{F_{14}, F_{15}, F_{16}, F_{17}, F_{18}\}$
4	M_4 – множество функций модуля импорта данных из семантически размеченных и шаблонизированных документов	$F(M_4) = \{F_5, F_6, F_{19}, F_{20}, F_{21}, F_{22}, F_{23}\}$
5	M_5 – множество функций модуля интеграции данных и модели	$F(M_5) = \{F_5, F_{24}, F_{25}\}$
6	M_6 – множество функций модуля тематической классификации	$F(M_6) = \{F_{26}, F_{27}, F_{28}, F_{29}, F_{30}, F_{31}\}$
7	M_7 – множество функций модуля автоматического разбора текста	$F(M_7) = \{F_{27}, F_{28}, F_{30}, F_{31}, F_{32}, F_{33}, F_{34}\}$
8	M_8 – множество функций модуля проверки на уникальность	$F(M_8) = \{F_{27}, F_{28}, F_{32}, F_{35}, F_{36}, F_{37}\}$
9	M_9 – множество функций модуля редактор онтологий	$F(M_9) = \{F_{38}, F_{39}, F_{40}, F_{41}, F_{42}, F_{43}, F_{44}\}$
10	M_{10} – множество функций модуля редактор метаданных	$F(M_{10}) = \{F_{45}, F_{46}, F_{47}\}$
11	M_{11} – множество функций модуля редактор классификаторов	$F(M_{11}) = \{F_{48}, F_{49}, F_{50}\}$
12	M_{12} – множество функций модуля интеллектуального анализа данных и вывода знаний	$F(M_{12}) = \{F_{51}, F_{52}, F_{53}, F_{54}, F_{55}\}$
13	M_{13} – множество функций модуля семантической сети	$F(M_{13}) = \{F_{43}, F_{56}, F_{57}\}$
14	M_{14} – множество функций модуля продукции новых и комбинированных знаний	$F(M_{14}) = \{F_{58}, F_{59}\}$
15	M_{15} - множество функций модуля интерфейса пользователя	$F(M_{15}) = \{F_{60}, F_{61}, F_{62}, F_{63}\}$

Исходя из выше приведенных формул (4.1)– (4.4), получили:

MODEL= $\{P_1, P_2, P_3, P_4, P_5\}$.

$P_1 = \{M_1, M_2, M_3, M_4\}$; $P_2 = \{M_5, M_6, M_7, M_8\}$; $P_3 = \{M_9, M_{10}, M_{11}\}$;

$P_4 = \{M_{12}, M_{13}, M_{14}\}$; $P_5 = \{M_{15}\}$.

$F(P_1) = F(M_1) \cup F(M_2) \cup F(M_3) \cup F(M_4)$; $F(P_2) = F(M_5) \cup F(M_6) \cup F(M_7) \cup F(M_8)$;

$F(P_3) = F(M_9) \cup F(M_{10}) \cup F(M_{11})$; $F(P_4) = F(M_{12}) \cup F(M_{13}) \cup F(M_{14})$; $F(P_5) = F(M_{15})$.

$F(MODEL) = F(P_1) \cup F(P_2) \cup F(P_3) \cup F(P_4) \cup F(P_5)$.

4.1.5 Разработка модели размещения компонентов фреймворка

Диаграммы deployment diagrams (размещения/развертывания) используются для моделирования физической архитектуры системы [141]. Диаграмма отображает аппаратные узлы, программные модули и связи между ними. Узел (node) представляет собой некоторый физически существующий элемент системы, обладающий некоторым вычислительным ресурсом. Графически на диаграмме развертывания узел изображается в форме трехмерного куба.

На данной диаграмме хорошо просматривается трехуровневая клиент-серверная архитектура приложения. Особенностью является физическое отделение данных от программ (сервер приложения), обрабатывающих эти данные. Такое разделение программных компонент позволяет оптимизировать нагрузки как на сетевое, так и на вычислительное оборудование комплекса. Компоненты трехуровневой архитектуры фреймворка, с точки зрения программного обеспечения, реализуются сервером БД, Web-сервером и браузером в качестве Web-клиента (Рисунок 4.4).

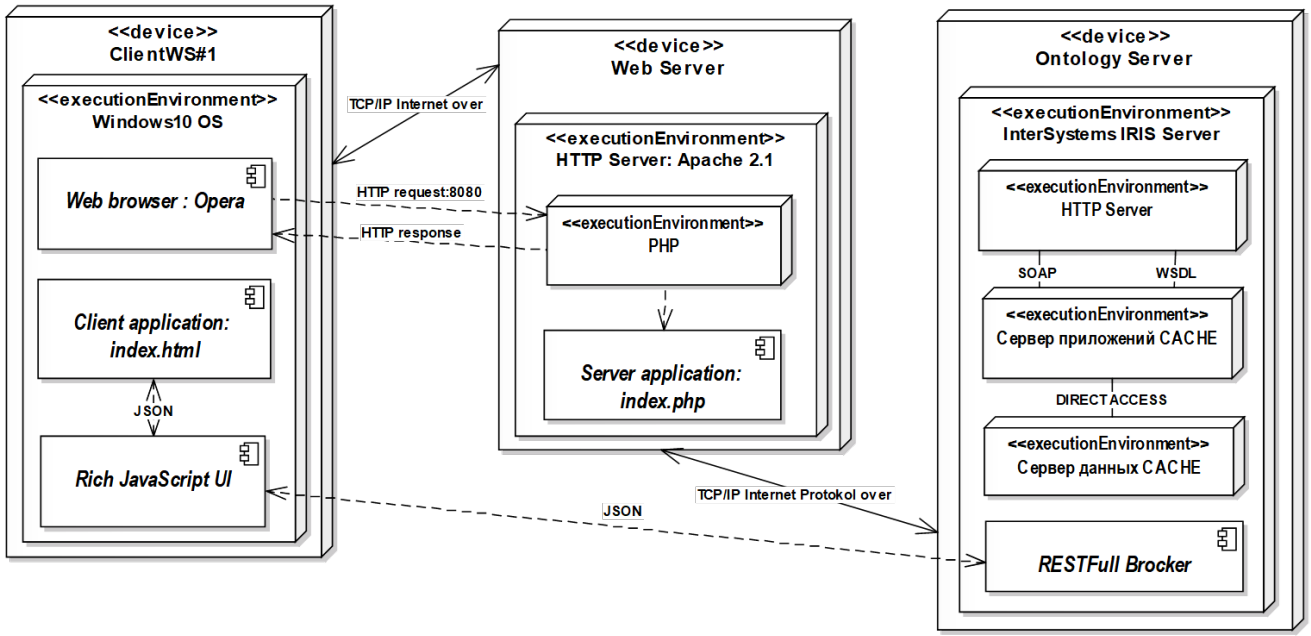


Рисунок 4.4 – Модель размещения компонентов фреймворка

Опишем взаимодействие компонентов трехуровневой архитектуры клиент-серверного приложения. Сервер БД представлен Cache-сервером Ontology Server,

роль сервера приложений играет «Web Server», роль клиента «ClientWS#1» выполняет компьютер с клиентским ПО.

Никаких ограничений на количество клиентских станций не накладывается. На клиентском ПК может быть установлен любой браузер, в нашем случае Opera, и локально установленный компонент Rich Java Script UI, задействованный в обмене данных с сервером данных.

Сервер приложений представляет веб-сервер Apache с установленным серверным ПО. Разработка серверного ПО проводилась с использованием языка PHP 7.0 и различных фреймворков и библиотек, таких как phpmorphy, phpquery-single, adam-lutka/php-wndb, php-text-analysis, pdfparser, wikibase-api, ext-json, ext-curl, ext-zip, PHPCache и др.

Узел «OntologyServer» представляет собой сервер данных CACHE InterSystems IRIS, который представляет собой современную платформу для работы с многомерными данными, ориентированную на работу в среде Веб-пространства и использующей в работе собственный встроенный HTTP Server.

Взаимодействие между узлами «ClientWS#1» и «Web Server», а также узлами «Web Server» и «OntologyServer» осуществляется стандартным образом с помощью HTTP-запросов поверх сетевого протокола TCP/IP. Передача данных происходит в JSON формате через взаимодействие компонета Server RESTFull Brocker, входящего в состав CACHE InterSystems IRIS и компонента Rich Java Script UI, установленного на клиенте.

4.2 Разработка на базе фреймворка системы учета информационных ресурсов научно-образовательной деятельности сотрудников вуза

В результате своей профессиональной деятельности, преподаватели кафедры выполняют поиск материалов для подготовки лекционных курсов и для самостоятельного изучения студентами, подбирают источники для формирования списка литературы, обрабатывают данные для своей научной и методической

деятельности, а также ведут учет и архивирование документации по результатам учебного процесса и организационной деятельности.

Сотрудники, методисты кафедры работают с информационными ресурсами в виде разнообразных документов, обеспечивающих качественную организацию учебного процесса и научно-методической деятельности. Студенты кафедры используют различные материалы для учебной и научной деятельности.

Ежегодно проводятся научно-технические мероприятия, на которых преподаватели представляют свои труды в различных областях науки. Результатами данных мероприятий являются статьи, сборники материалов конференций, отчеты и т.д. Данные материалы могут публиковаться в сети Интернет или храниться в электронном виде локально. Формированием годового отчета, подсчетом рейтинга и обработкой необходимой информации занимается секретарь кафедры. Этот процесс выполняется вручную и занимает продолжительное время, что негативно сказывается на оперативности и качестве получения результатов.

Система управления научно-образовательной деятельности сотрудников кафедры вуза на основе фреймворка СУИР имеет следующую структуру:

1. Информационная подсистема ведения электронного кафедрального архива (модуль «АРХИВ»):
    a. Учет работ студентов обязательного хранения (ВКР, КР и КП);
    b. Учет данных по антиплагиату;
    c. Импорт данных в БД;
    d. Поиск данных по архиву;
    e. Удаление устаревшей информации;
    f. Настройка параметров системы.
2. Информационная подсистема ведения кафедральной документации (модуль «АРМ секретаря»):
    a. Учет документации кафедры;
    b. Учет учебных планов;
    c. Учет сотрудников;

d. Учет студентов;
e. Подбор документации для формирования отчетных документов;
f. Поиск данных по различным критериям;
g. Настройка параметров системы.

3. Подсистема учета научной-исследовательской деятельности преподавателей (модуль «Наука»):

a. Учет научных статей;
b. Поиск статей в сети Интернет, находящихся в открытом доступе;
c. Учет наукометрических показателей преподавателей;
d. Подбор материалов для изучения дисциплин;
e. Ведение личного архива материалов;
f. Подбор данных для формирования итогового отчета по публикациям;
g. Подбор раздела для публикации статьи на факультетскую конференцию;
h. Подбор преподавателя для выполнения исследований;
i. Настройка параметров системы.

4. Подсистема учета образовательной деятельности преподавателей (модуль «Преподаватель»):

a. Учет образовательных ресурсов;
b. Поиск материалов для подготовки к занятиям;
c. Учет документации по курсу;
d. Учет студенческих работ.
e. Подсистема администратора:
f. Онтологический редактор;
g. Учет пользователей;
h. Регистрация и аутентификация пользователей;
i. Настройка параметров системы.

С различными задачами работают различные пользователи системы: администратор, секретарь, преподаватель, студент.

4.3 Разработка программного модуля учета научно-исследовательской деятельности сотрудников вуза «Наука»

Разработанные в диссертационной работе модели и алгоритмы были использованы при реализации программного модуля учета научно - исследовательской деятельности преподавателей кафедры технического вуза «Наука». На рисунке 4.5 приведена функциональная структура модуля «Наука» на основе фреймворка СУИР.

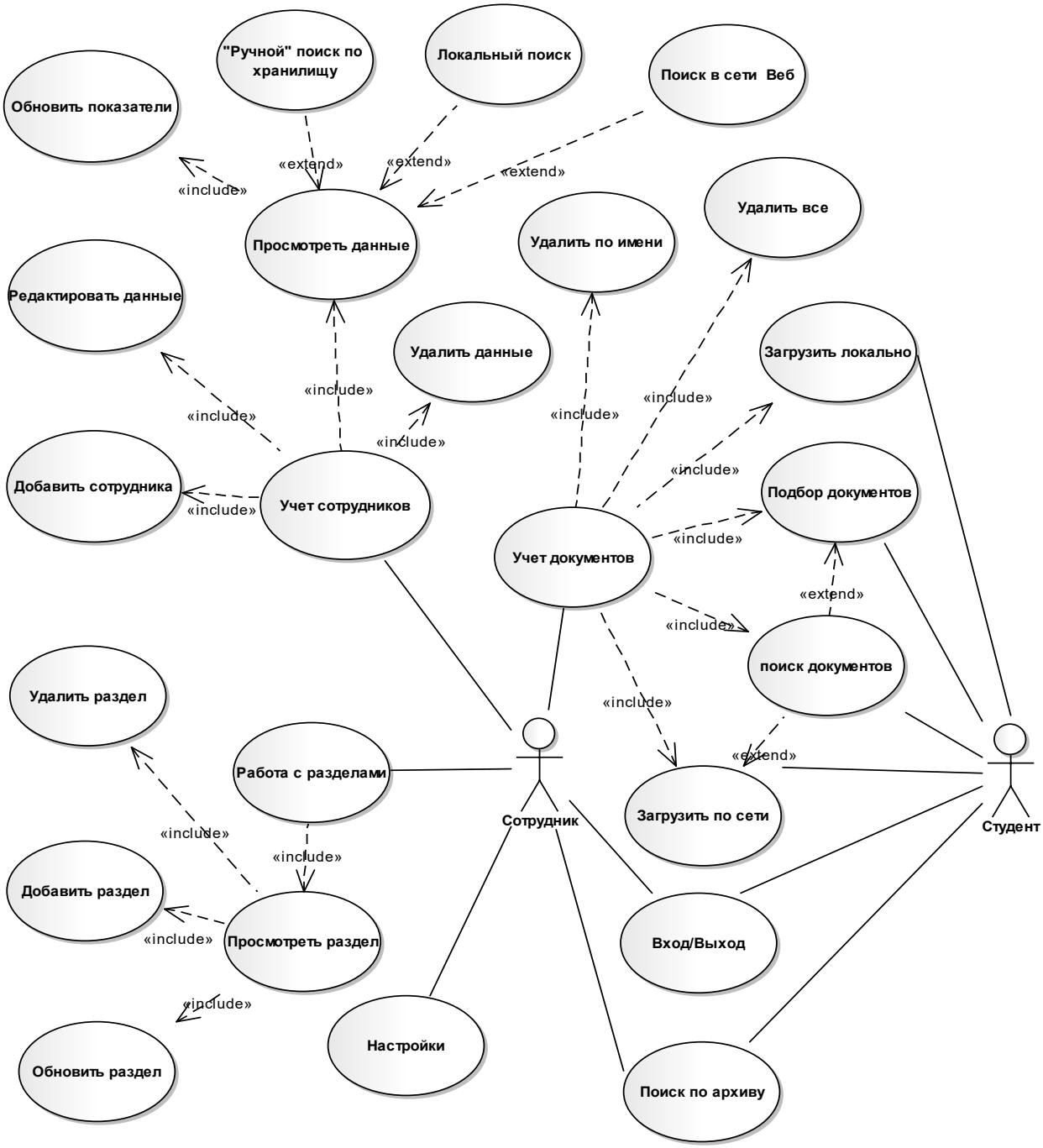


Рисунок 4.5 – Функциональная структура модуля «Наука»

Основными пользователями модуля являются студент и преподаватель.

Список программных модулей системы базового фреймворка СУИР, использованных в реализации программного модуля «Наука» приведен в таблице 4.6, а список дополнительно разработанных программных модулей и функций, использованных в реализации программного модуля «Наука», приведен в таблице 4.7.

Таблица 4.6 – Список программных модулей системы фреймворка СУИР, использованных при реализации модуля «Наука»

Пакет	Имя	Описание
P_1	M_1	Модуль поиска информации в сети Интернет
P_1	M_2	Модуль поиска информации по хранилищу данных
P_1	M_4	Модуль импорта данных
P_2	M_5	Модуль интеграции данных и модели
P_2	M_6	Интеллектуальный модуль тематической классификации
P_2	M_7	Модуль автоматического разбора текста
P_4	M_{12}	Модуль интеллектуального анализа данных и вывода знаний
P_5	M_{15}	Интерфейс пользователя

Таблица 4.7 – Список дополнительно разработанных программных функций и модулей при реализации модуля «Наука»

Пакет	Имя	Описание
P_5	M_{17}	Модуль получения наукометрических показателей
P_5	M_{16}	Модуль учета сотрудников
P_3	F_{66}	Функция подбора документов по запросу
P_5	F_{67}	Функция настройки параметров системы

Программный код программного модуля «Наука» системы составляет около 7 тысяч строк программного кода без учета сторонних библиотек и фреймворков. Фрагмент программы представлен в приложении В. Программный модуль реализован с применением следующих технологий (Рисунок 4.6).

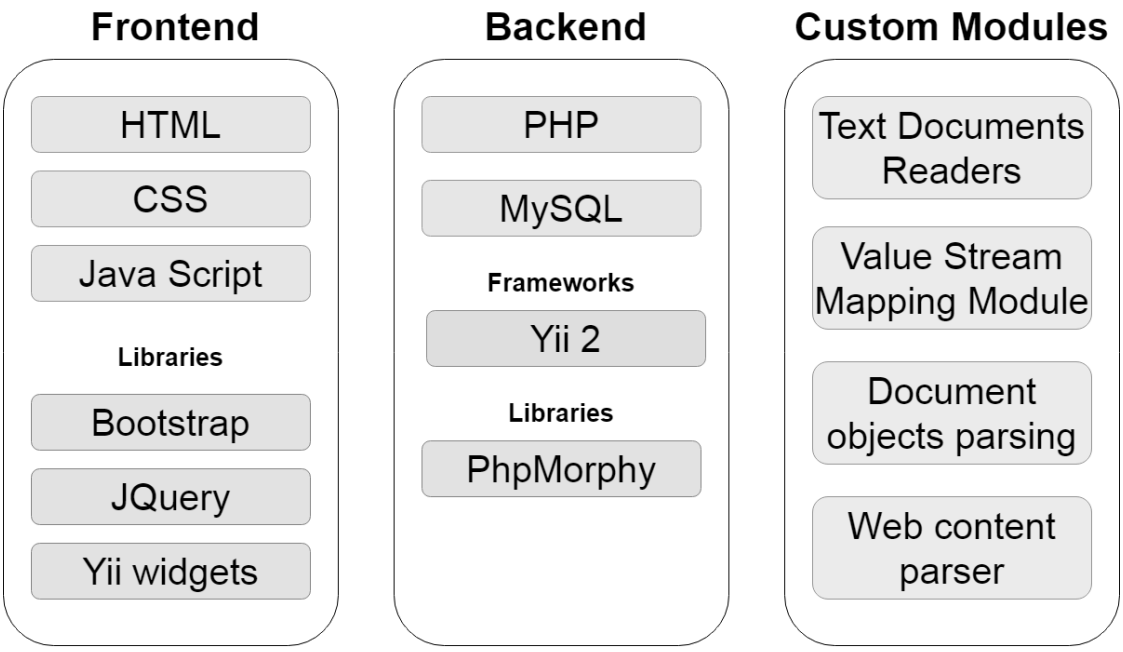


Рисунок 4.6 – Стек программных технологий модуля «Наука»

Основной код написан на языке PHP версии 7.4 с использованием фреймворка YII2. Фреймворк YII распространяется в открытых исходных кодах [138]. В качестве архитектурной схемы проектирования программного комплекса в YII2 используется широко распространенная схема «модель-представление-поведение» (Model-View-Controller, MVC). В качестве базы данных используется MySQL версии 8.0, а в качестве программного интерфейса для доступа и манипулирования данными – Active Record YII2.

При частотном анализе используется библиотека морфологического анализа phpMorphy, распространяемая с открытым исходным кодом [139].

В качестве языка сценариев, исполняемых на стороне клиента, в системе используется язык Javascript. В системе применяется широко распространенный Javascript-фреймворк jQuery.

Структура специального программного обеспечения модуля «Наука» в виде диаграммы компонентов приведена на рисунке 4.7.

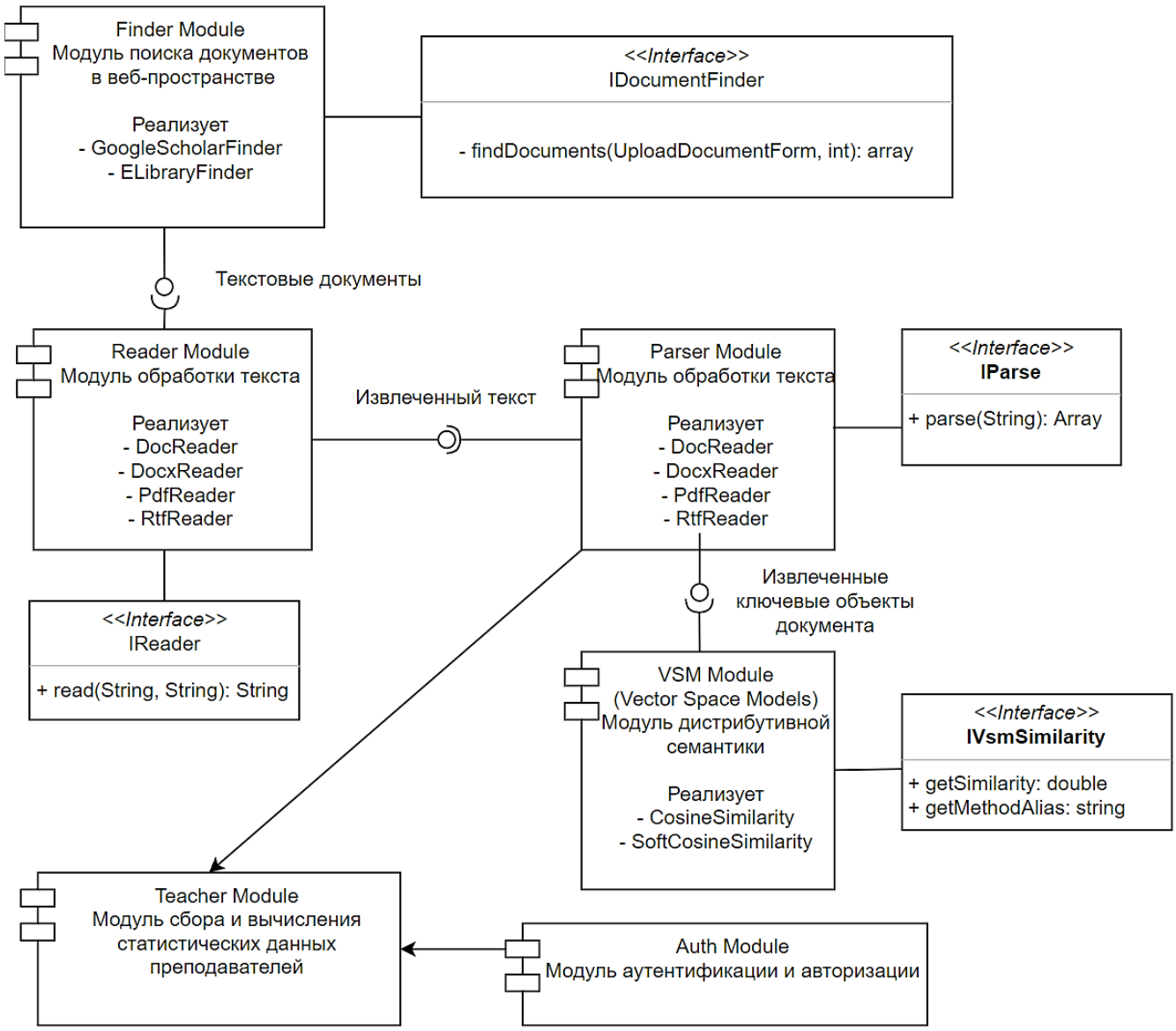


Рисунок 4.7 – Диаграмма компонентов программного модуля «Наука»

Диаграмма развертывания программного модуля «Наука» приведена на рисунке 4.8. На диаграмме развертывания видно, что на веб сервере разделены следующие слои:

– Controller layer – принимает запросы с клиентской части и делегирует бизнес-логику сервисам;
– Service layer – реализует бизнес-логику и взаимодействует с моделями;
– Model layer – реализует доменную часть, описывая сущности системы и взаимодействует с БД посредством Active Record, который обеспечивает

объектно-ориентированный интерфейс для доступа и манипулирования данными, хранящимися в базах данных;

– хранилище файлов.

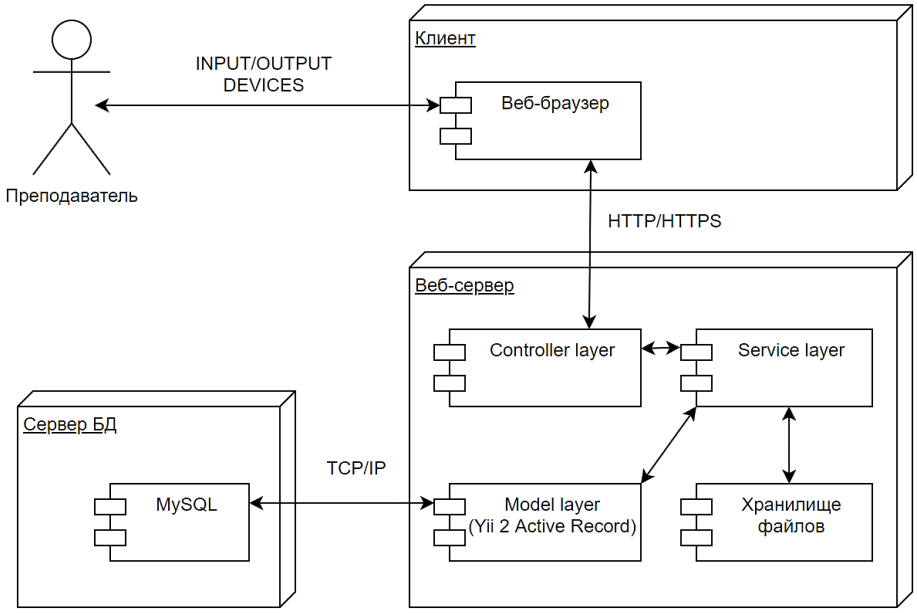


Рисунок 4.8 – Диаграмма развертывания программного модуля «Наука»

4.3.1 Разработка эффективного RDF-хранилища

В связи с резким ростом объёмов хранимой информации, в том числе и онтологий, при разработке современных ИС их хранение в плоских файлах оказывается непродуктивным. Одним из способов решения данной проблемы является организация RDF-хранилища – эффективного хранилища для онтологических баз знаний.

Эффективное RDF-хранилище должно удовлетворять следующим требованиям: высокая производительность; минимальные затраты памяти (дискового пространства) для хранения онтологий; возможность хранения онтологий любой структуры.

Для решения поставленной задачи необходимо предварительно решить ряд вопросов и провести ряд исследований: подобрать наиболее подходящую модель данных для реализации собственного хранилища; изучить существующие RDF-

хранилища и способы их организации; выбрать программное средство, обладающее необходимым набором возможностей для реализации хранилища.

Онтологический подход помогает динамически сформировать саму структуру хранилища данных. Сравним, как представлены данные в онтологии (Рисунок 4.9) и как по этим же данным построена динамическая структура (Рисунок 4.10).

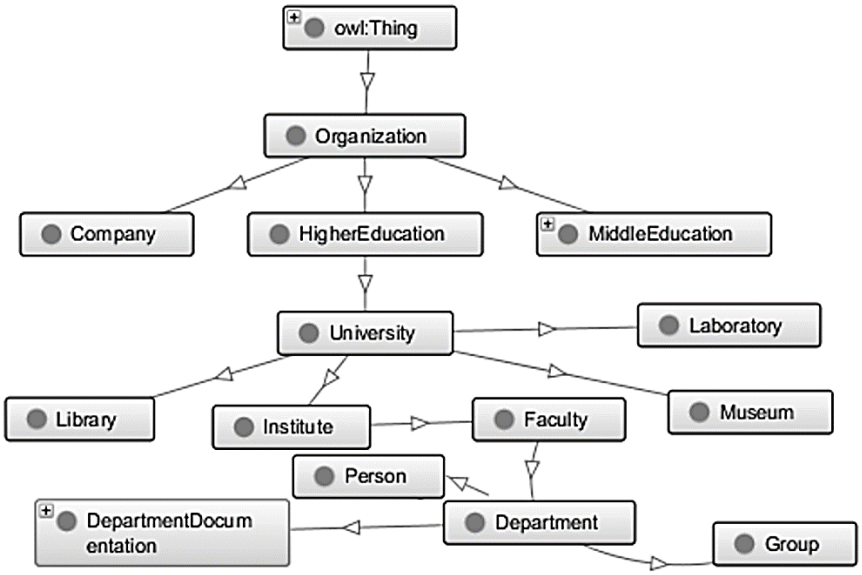


Рисунок 4.9 – Фрагмент онтологии научно-образовательных организаций

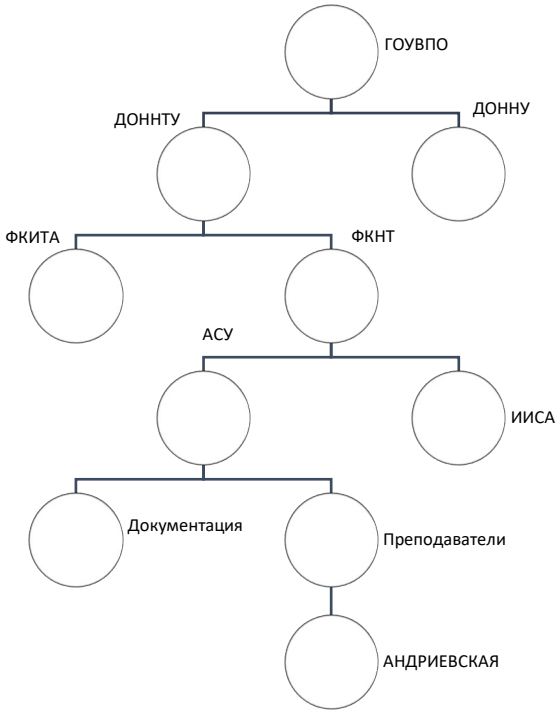


Рисунок 4.10 – Хранилище данных

Алгоритм динамического формирования структуры хранилища:

1. Получили из тензора след по отношению P5 «individual» $P^{oo}=S^{oo}{}_5$ - матрицу смежности. Вычислили матрицу расстояний R (Рисунок 4.11).

ГОУВПО	1	2	3	4	5	6	7	8	9
ДОННУ 1	1	1							
ДОННТУ 2	1	1	1	1					
ФКНТ 3		1	1		1	1			
ФКИТА 4		1		1					
АСУ 5			1		1		1	1	
ИИСА 6			1			1			
Преподаватели 7					1		1		1
Документация 8					1			1	
Андриевская 9							1		1

	1	2	3	4	5	6	7	8	9	R
1	0	1	2	2	3	3	4	4	5	5
2	1	0	1	1	2	2	3	3	4	4
3	2	1	0	0	1	1	2	2	3	3
4	2	1	0	0	0	0	0	0	0	2
5	3	2	1	0	0	0	1	1	2	3
6	3	2	1	0	0	0	0	0	0	3
7	4	3	2	0	1	0	0	0	1	4
8	4	3	2	0	1	0	0	0	0	4
9	5	4	3	0	2	0	1	0	0	5

Рисунок 4.11 – Матрицы смежности и расстояний

Определили радиус R=2 и диаметр d=5. Концевые вершины первая и девятая. Зная эти параметры, построили само дерево. В СУБД сформировалась соответствующая динамическая структура, показанная на рисунке 4.12.

В веб приложении по данной структуре сформировалась структура хранилища (Рисунок 4.13).

Просмотр глобала в области ONTO:

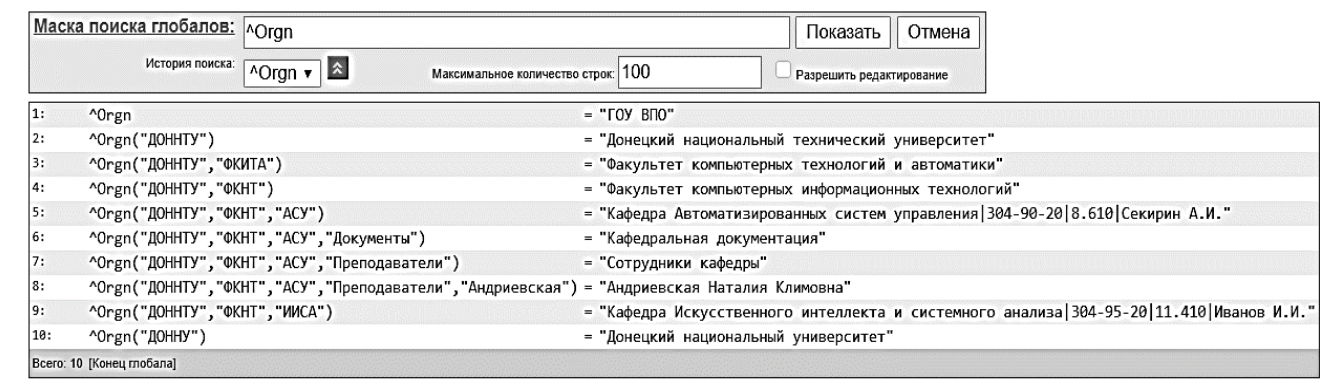


Рисунок 4.12 – Просмотр структуры данных в интерфейсе СУБД

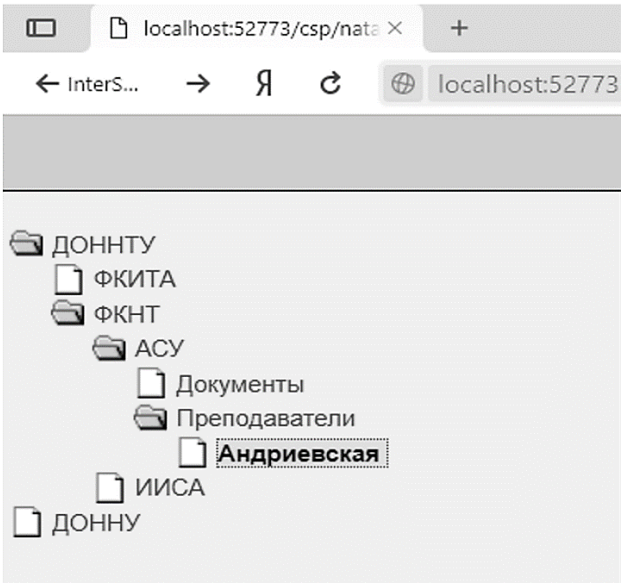


Рисунок 4.13 – Структура хранилища в окне веб-приложения

При разработке модуля автоматического разбора текста и для решения задач регистрации и авторизации была разработана база данных в формате MySQL, которая содержит различные таблицы для служебного использования и промежуточные таблицы для хранения данных перед их обработкой и помещением в RDF-хранилище (Рисунок 4.14).

Навигация по контенту организуется на основе таксономических отношений, задающих иерархию понятий, и ассоциативных отношений, связывающих между собой информационные ресурсы. При этом обеспечивается возможность выбора информационных ресурсов определенного класса, фильтрации списков ресурсов, а также просмотра и редактирования информационных ресурсов.

Навигация начинается с выбора определенного класса в дереве понятий онтологии, построенном на основе отношения таксономии. Формирование списка объектов выполняется с учетом транзитивного замыкания по отношению таксономии. Вследствие этого результирующий список будет включать как объекты искомого класса, так и объекты его классов наследников.

Содержательный доступ к знаниям и данным ИС осуществляется через пользовательский интерфейс, вся работа которого базируется на онтологии. Пользователь может использовать ее в качестве «проводника» для навигации по контенту ИС, а также базиса для формулирования запросов к ИС.

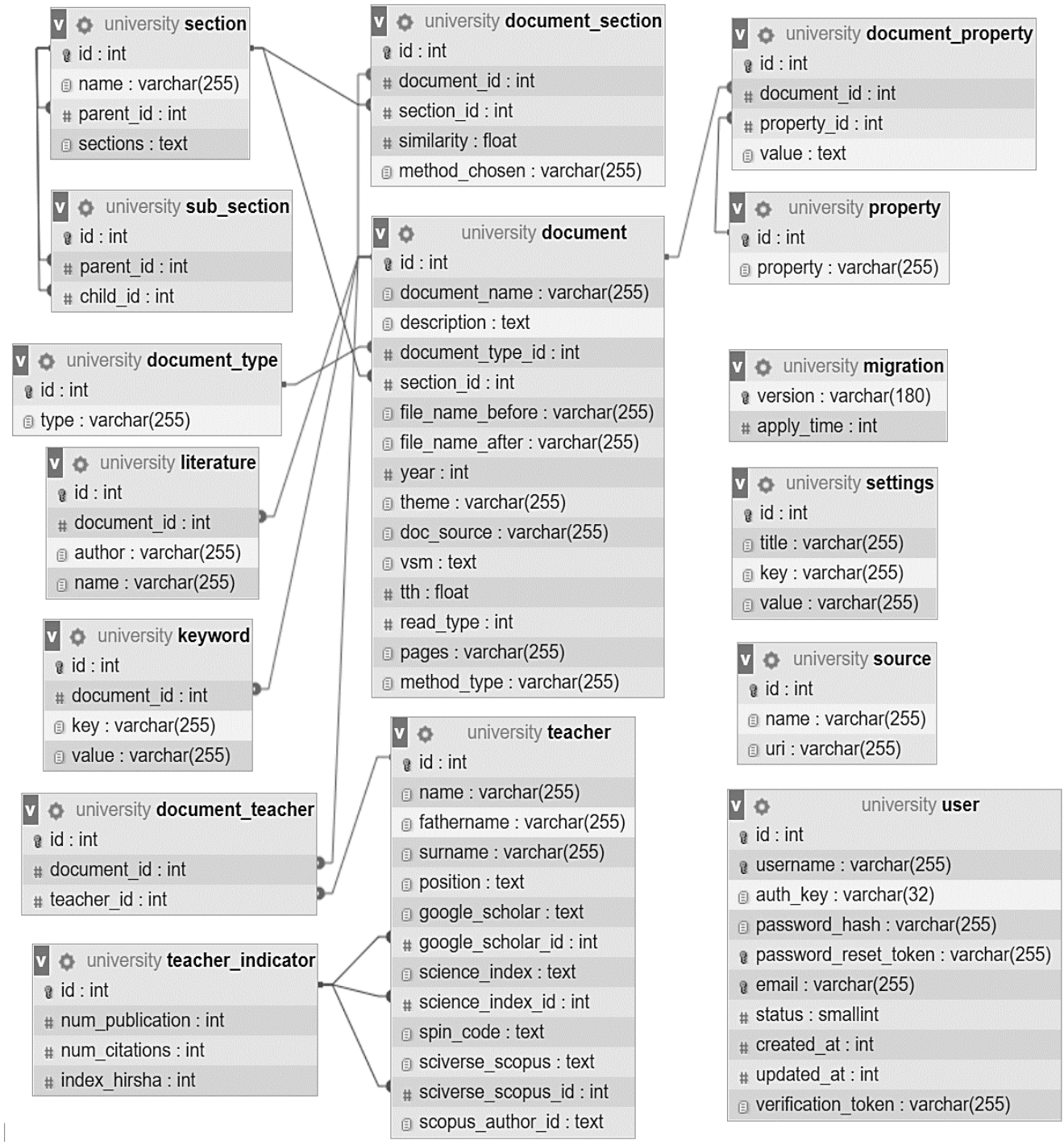


Рисунок 4.14 – База данных MySQL модуля «Наука»

Поисковые запросы задаются через специальный графический интерфейс, управляемый онтологией ИС. Для хранения каркаса онтологии была использована простейшая форма хранения онтологий в виде OWL-файла. Использование URI для задания субъектов и свойств позволяло связывать отдельные утверждения (RDF-триплеты) в сложные семантические сети.

4.3.2 Построение модели системы информационной безопасности

Одной из ключевых задач создания эффективной информационной среды является разработка системы управления доступом к ИР. Управление доступом реализует собой модель, определяющую правила доступа к ИС, к ее информационным ресурсам и правила извлечения знаний из защищаемых ресурсов. В структуре разработанной системы задачи управления доступом реализуются в модуле интерфейса. Организация доступа включает в себя поддержку процедуры авторизации и аутентификации пользователей, а также поддержку механизма разграничения доступа к информационным ресурсам.

Авторизация пользователей выполняется на основе учетных записей. Все разрешения пользователей управляются с помощью ролей.

В основе подсистемы лежит штатный механизм СУБД MySQL. Каждому пользователю или группе пользователей назначается роль, обеспечивающая доступ к объектам системы.

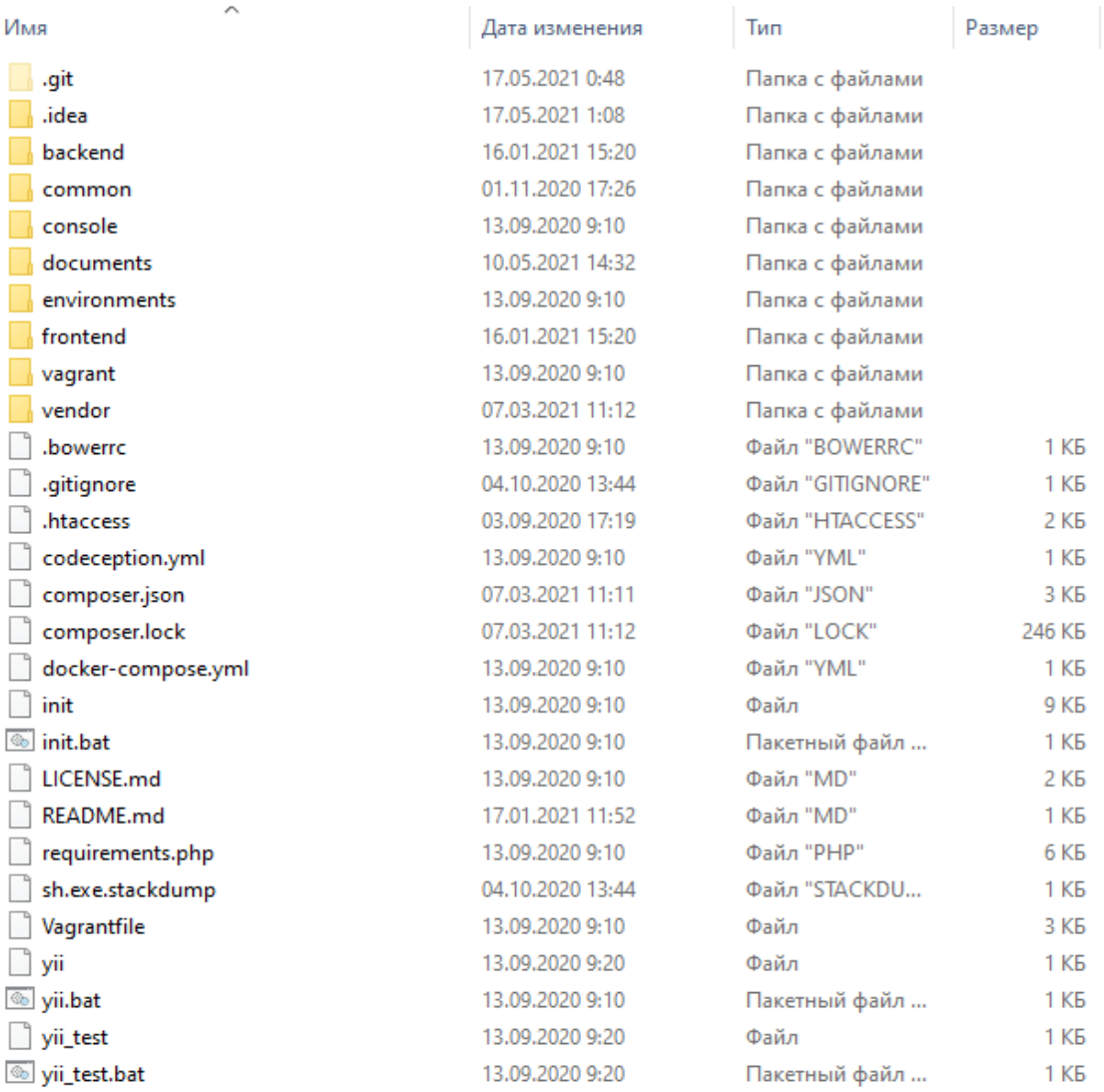
Группы пользователей и соответствующие им роли и права представлены в таблице 4.8.

Таблица 4.8 – Пользователи системы

Группа	Роль	Права	Описание
Администраторы	«Администратор ИР»	Все действия	Осуществляет назначение экспертов и других ролей. Управляет учётными записями пользователей. Выполняет обслуживание системы.
Администраторы	«Эксперт по знаниям»	Редактирование онтологий	Предоставляет знания для системы. Производит оценку онтологии, отображенной средствами системы по критериям эргономичности, полноты и непротиворечивости.
Сотрудники	«Сотрудник»	Добавление, изменение и удаление ИР	Использует знания в учебных, исследовательских целях. Может просмотреть, удалить, добавить любые свои данные в БД.
Сотрудники	«Преподаватель»	Добавление, изменение и удаление ИР	Использует знания в учебных, исследовательских целях. Может просмотреть, удалить, добавить любые свои данные в БД.
Студенты	«Пользователь»	Просмотр ИР	Взаимодействие в познавательных целях, имеет право на просмотр данных.

4.3.3 Тестирование программного модуля «Наука»

Программный модуль «Наука» был разработан на языке PHP с использованием фреймворка YII2 и других библиотек. Структура программного проекта приведена на рисунке 4.15.



Имя	Дата изменения	Тип	Размер
.git	17.05.2021 0:48	Папка с файлами	
.idea	17.05.2021 1:08	Папка с файлами	
backend	16.01.2021 15:20	Папка с файлами	
common	01.11.2020 17:26	Папка с файлами	
console	13.09.2020 9:10	Папка с файлами	
documents	10.05.2021 14:32	Папка с файлами	
environments	13.09.2020 9:10	Папка с файлами	
frontend	16.01.2021 15:20	Папка с файлами	
vagrant	13.09.2020 9:10	Папка с файлами	
vendor	07.03.2021 11:12	Папка с файлами	
.bowerrc	13.09.2020 9:10	Файл "BOWERRC"	1 КБ
.gitignore	04.10.2020 13:44	Файл "GITIGNORE"	1 КБ
.htaccess	03.09.2020 17:19	Файл "HTACCESS"	2 КБ
codeception.yml	13.09.2020 9:10	Файл "YML"	1 КБ
composer.json	07.03.2021 11:11	Файл "JSON"	3 КБ
composer.lock	07.03.2021 11:12	Файл "LOCK"	246 КБ
docker-compose.yml	13.09.2020 9:10	Файл "YML"	1 КБ
init	13.09.2020 9:10	Файл	9 КБ
init.bat	13.09.2020 9:10	Пакетный файл ...	1 КБ
LICENSE.md	13.09.2020 9:10	Файл "MD"	2 КБ
README.md	17.01.2021 11:52	Файл "MD"	1 КБ
requirements.php	13.09.2020 9:10	Файл "PHP"	6 КБ
sh.exe.stackdump	04.10.2020 13:44	Файл "STACKDU...	1 КБ
Vagrantfile	13.09.2020 9:10	Файл	3 КБ
yii	13.09.2020 9:20	Файл	1 КБ
yii.bat	13.09.2020 9:10	Пакетный файл ...	1 КБ
yii_test	13.09.2020 9:20	Файл	1 КБ
yii_test.bat	13.09.2020 9:20	Пакетный файл ...	1 КБ

Рисунок 4.15 – Структура проекта

Перед работой необходимо зарегистрироваться и выполнить вход с помощью формы, представленной на рисунке 4.16.

После успешной аутентификации появляется главная форма приложения, приведенная на рисунке 4.17 и предназначенная для различных операций с документами.

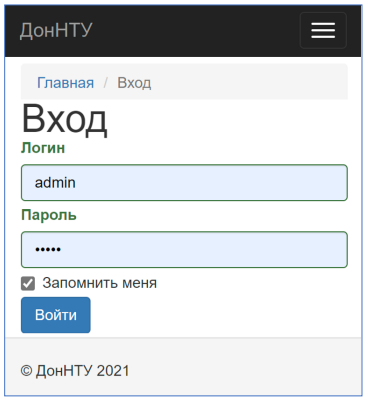


Рисунок 4.16 – Форма входа в систему

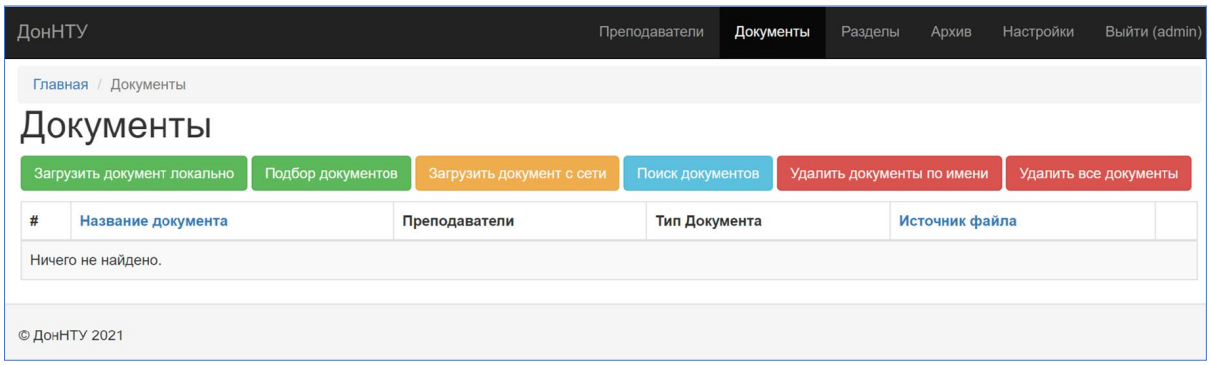


Рисунок 4.17 – Главная форма приложения «Документы»

Выполним тестирование работы функций, реализованных при вызове пункта меню «Документы». При выборе первой операции происходит просмотр локально загруженных файлов. Форма для организации процесса загрузки приведена на рисунке 4.18, а форма просмотра результатов загрузки, т.е. списка загруженных документов, на рисунке 4.19.

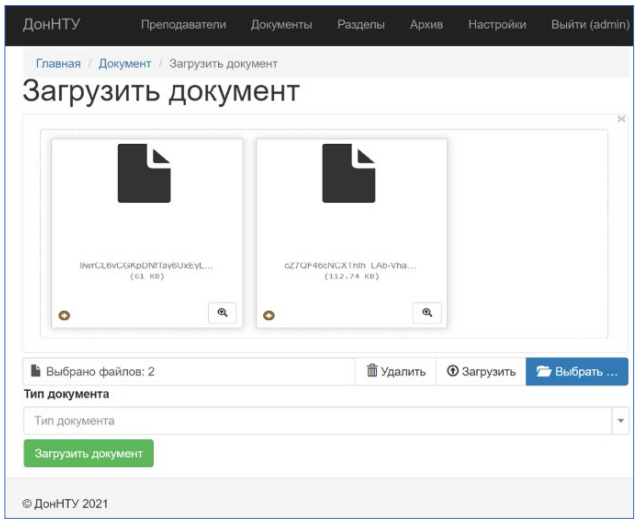


Рисунок 4.18 – Форма загрузки файлов

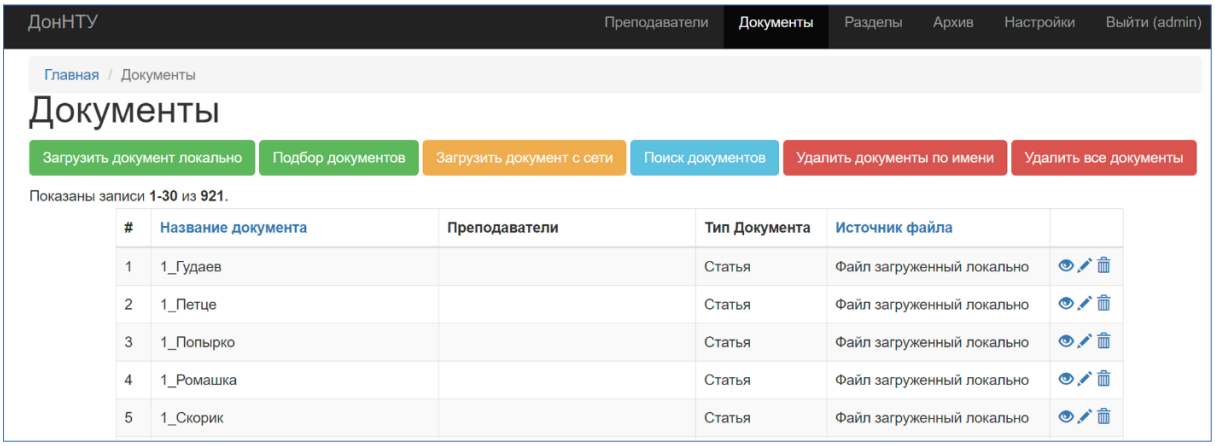


Рисунок 4.19 – Список загруженных документов

Система позволяет в автоматизированном режиме осуществлять поиск необходимой информации в сети Интернет. Использование данных, полученных в результате сбора, не противоречит законодательству Российской Федерации [4], так как данные изначально находятся в открытом доступе.

Форма поискового запроса представлена на рисунке 4.20. Алгоритм интеллектуального поиска позволяют отсеивать заведомо некорректную и нерелевантную информацию, и ранжировать оставшиеся результаты, обеспечивая, таким образом, достоверность данных (Рисунок 4.21). Найденные документы, доступные для скачивания имеют ссылки для закачки, по которым можно свободно перейти к источнику информации.

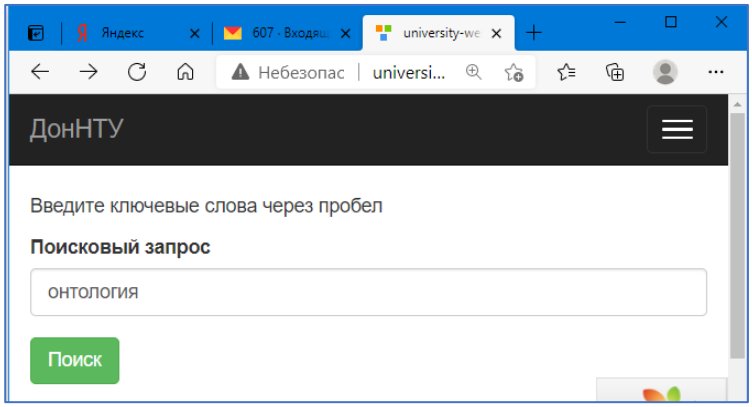


Рисунок 4.20 – Форма поискового запроса

Форма просмотра результатов автоматического разбора файла приведена на рисунке 4.22. Эти данные сохраняются в онтологии.

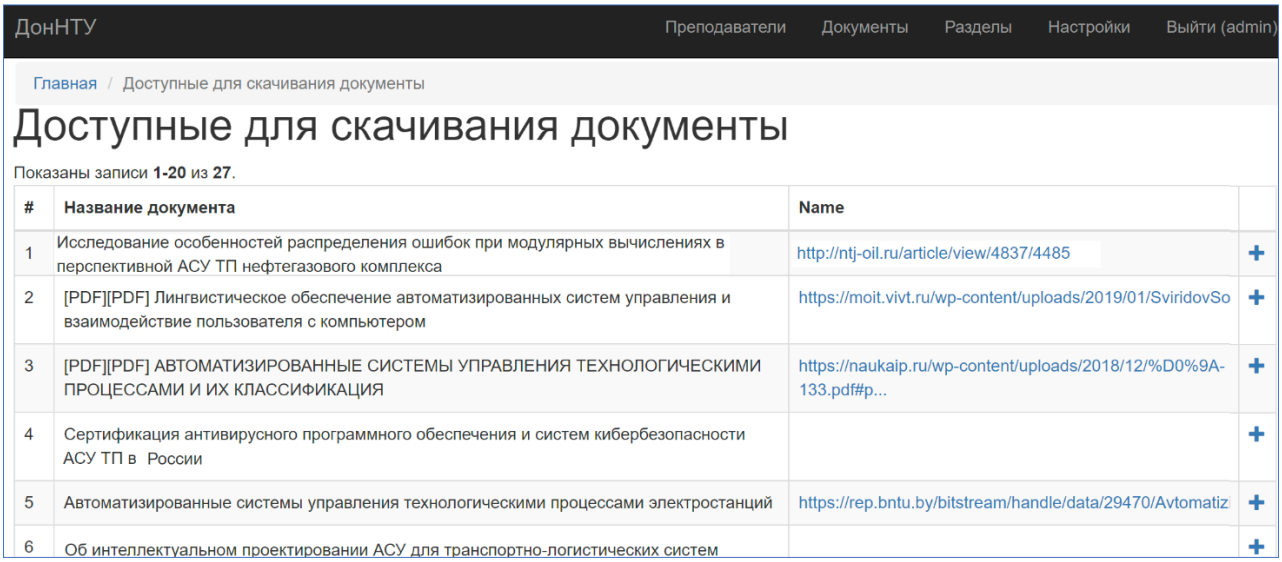


Рисунок 4.21 – Результаты Интернет-поиска

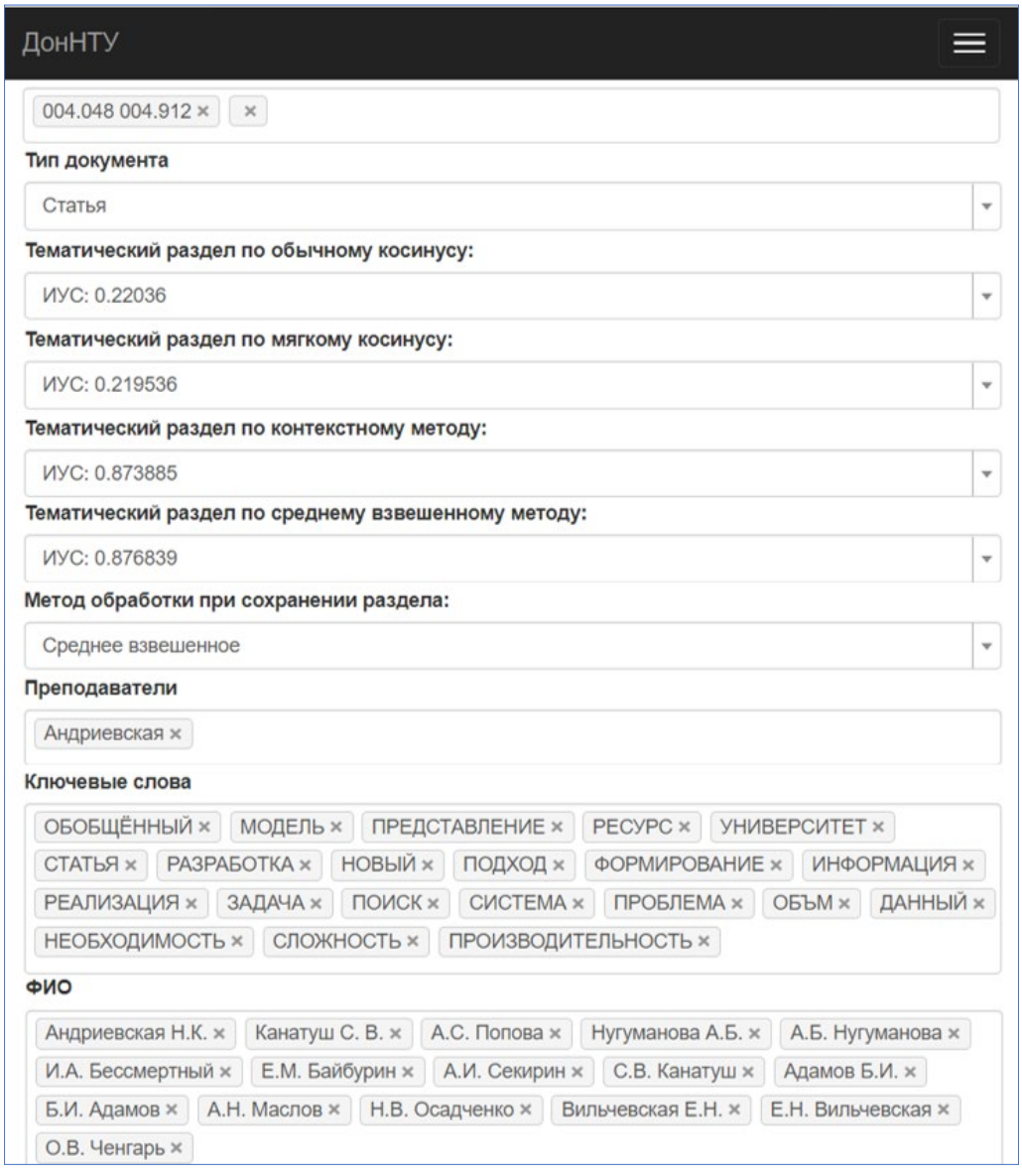


Рисунок 4.22 – Фрагмент формы с результатами разбора документа

При поисковом запросе по внутреннему хранилищу результаты возвращаются в виде списка документов с рассчитанными оценками семантической близости каждого документа к поисковому запросу (Рисунок 4.23). Список ограничен значением семантической близости, задаваемым в настройках системы.

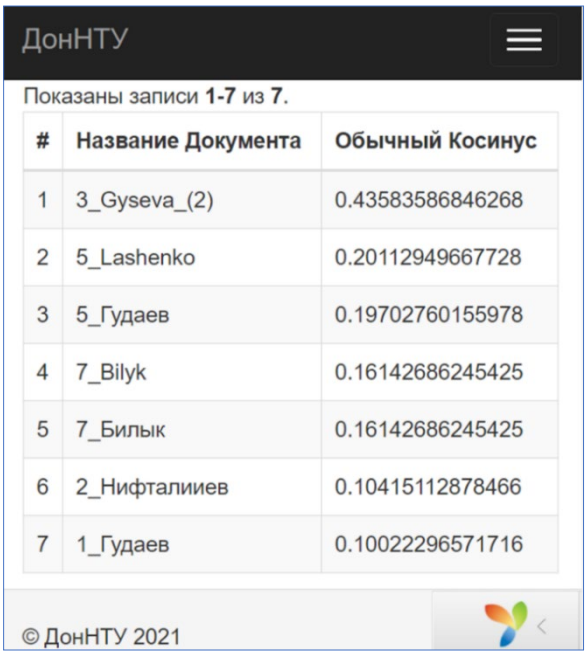


Рисунок 4.23 – Результаты поискового запроса по локальному хранилищу данных

Выполним тестирование пункта меню «Преподаватели». В первую очередь появляется форма, представленная на рисунке 4.24, с помощью которой ведется кафедральный учет сотрудников, занимающихся научными исследованиями.

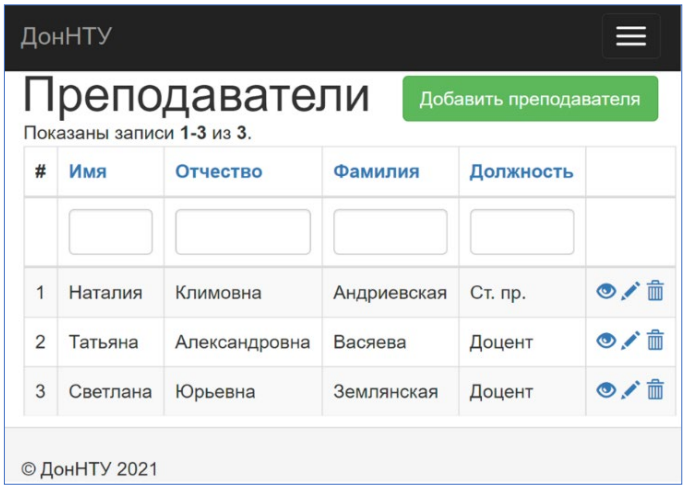


Рисунок 4.24 – Форма учета сотрудников

При просмотре данных о сотрудниках появляется форма с наукометрическими показателями, полученными из профилей преподавателей (Рисунок 4.25). Предусмотрена возможность обновления показателей.

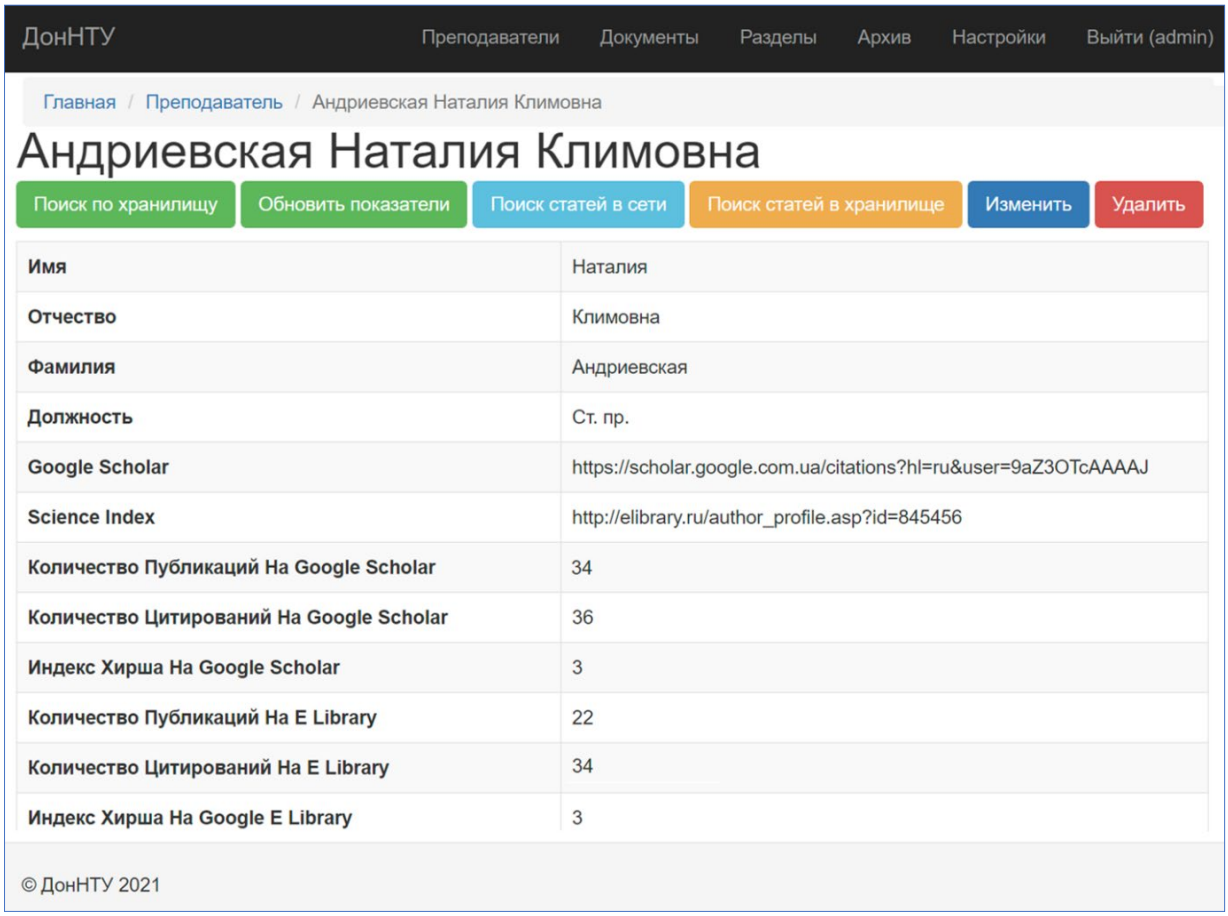
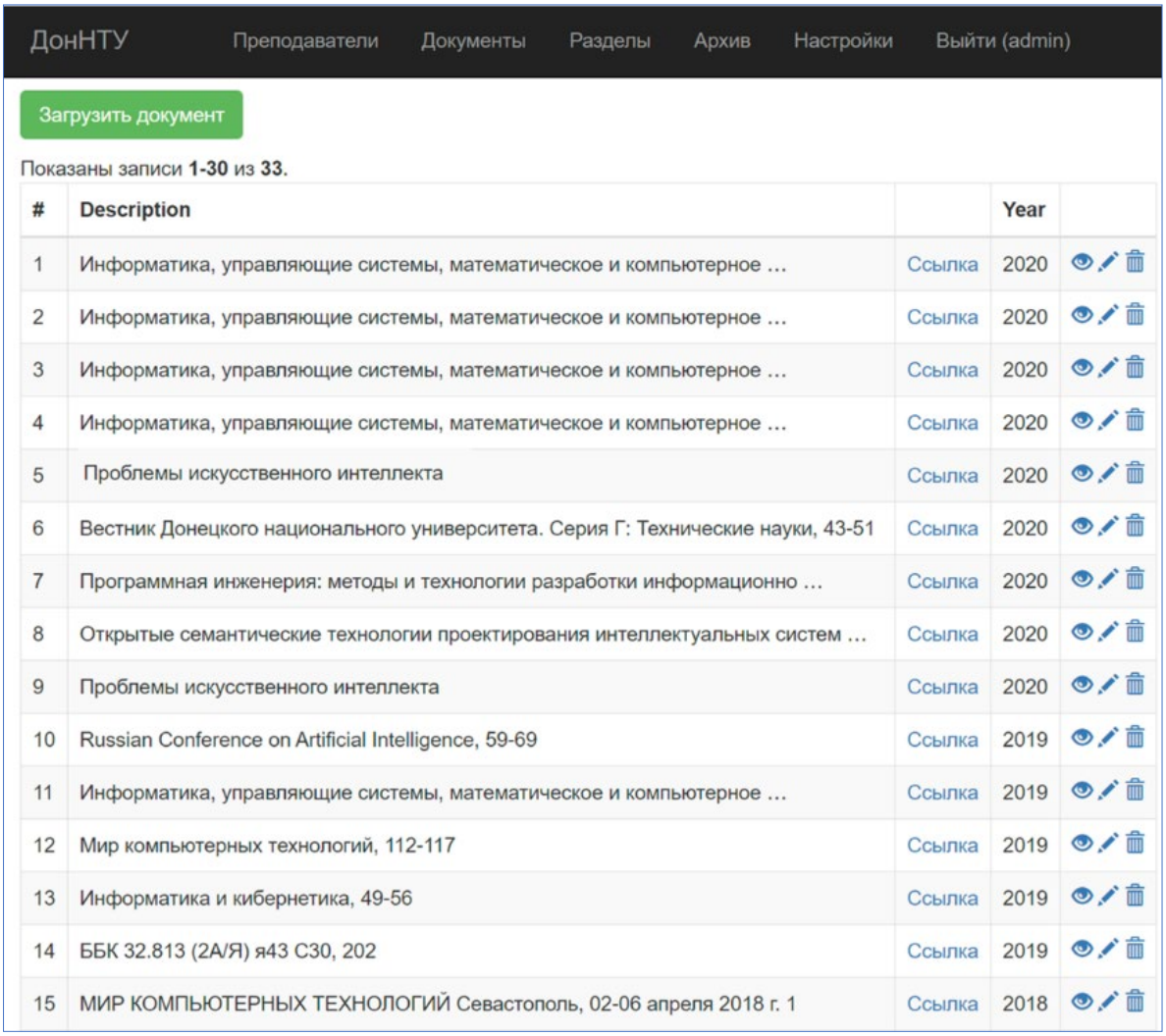


Рисунок 4.25 – Наукометрический профиль сотрудника

Целью поисковых запросов в сети модуля «Наука» является поиск и исследование интернет-пространства на наличие информации публикаций научных сотрудников, не занесенной в хранилище. Полученные данные о научных трудах публикуются в открытом доступе, имеют ссылки на скачивание. Здесь же имеется возможность просмотреть и откорректировать информацию о документе (Рисунок 4.26).

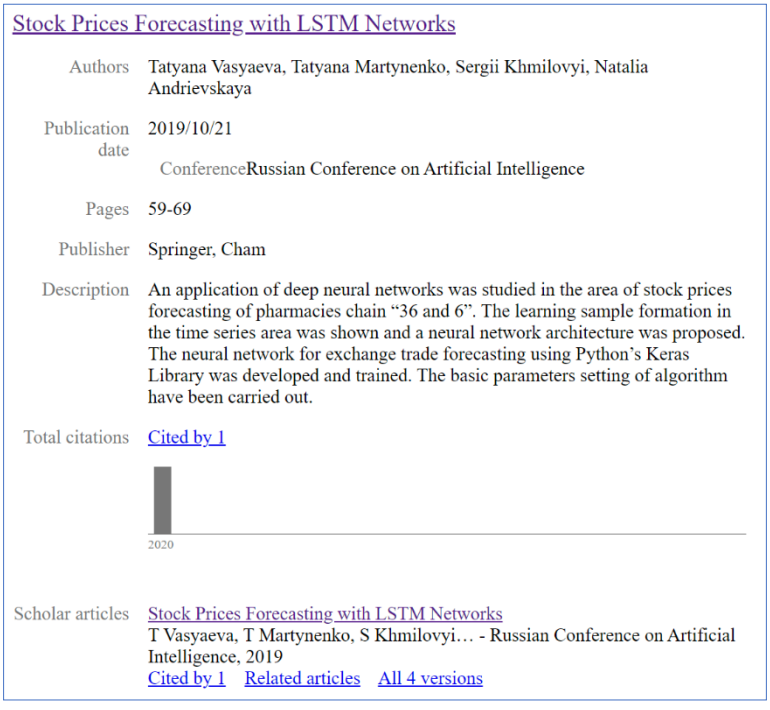
После выбора ссылки идет перенаправление к источнику опубликования информации (Рисунок 4.27).

Существует возможность поиска по иерархической структуре архива, динамически сформированного хранилища программным путем (Рисунок 4.28).



#	Description		Year	
1	Информатика, управляющие системы, математическое и компьютерное ...	Ссылка	2020	
2	Информатика, управляющие системы, математическое и компьютерное ...	Ссылка	2020	
3	Информатика, управляющие системы, математическое и компьютерное ...	Ссылка	2020	
4	Информатика, управляющие системы, математическое и компьютерное ...	Ссылка	2020	
5	Проблемы искусственного интеллекта	Ссылка	2020	
6	Вестник Донецкого национального университета. Серия Г: Технические науки, 43-51	Ссылка	2020	
7	Программная инженерия: методы и технологии разработки информационно ...	Ссылка	2020	
8	Открытые семантические технологии проектирования интеллектуальных систем ...	Ссылка	2020	
9	Проблемы искусственного интеллекта	Ссылка	2020	
10	Russian Conference on Artificial Intelligence, 59-69	Ссылка	2019	
11	Информатика, управляющие системы, математическое и компьютерное ...	Ссылка	2019	
12	Мир компьютерных технологий, 112-117	Ссылка	2019	
13	Информатика и кибернетика, 49-56	Ссылка	2019	
14	ББК 32.813 (2А/Я) я43 С30, 202	Ссылка	2019	
15	МИР КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ Севастополь, 02-06 апреля 2018 г. 1	Ссылка	2018	

Рисунок 4.26 – Список с результатами поискового запроса



Stock Prices Forecasting with LSTM Networks

Authors: Tatyana Vasyaeva, Tatyana Martynenko, Sergii Khmilovyi, Natalia Andrievskaya

Publication date: 2019/10/21

Conference: Russian Conference on Artificial Intelligence

Pages: 59-69

Publisher: Springer, Cham

Description: An application of deep neural networks was studied in the area of stock prices forecasting of pharmacies chain “36 and 6”. The learning sample formation in the time series area was shown and a neural network architecture was proposed. The neural network for exchange trade forecasting using Python’s Keras Library was developed and trained. The basic parameters setting of algorithm have been carried out.

Total citations: Cited by 1

2020

Scholar articles: Stock Prices Forecasting with LSTM Networks
T Vasyaeva, T Martynenko, S Khmilovyi… - Russian Conference on Artificial Intelligence, 2019
Cited by 1 Related articles All 4 versions

Рисунок 4.27 – Ссылка на источник публикации

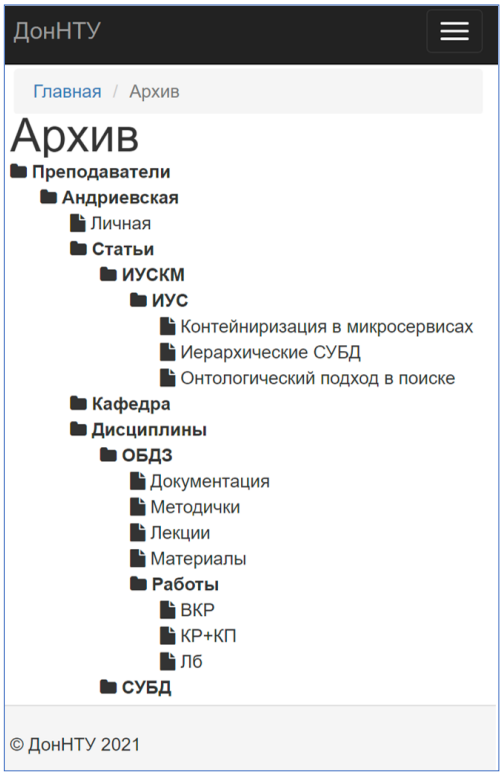


Рисунок 4.28 – Просмотр структуры хранилища

Эксперту по знаниям в данном интерфейсе предназначены функции ведения тематических разделов для экспертной поддержки онтологий. Он определяет и корректирует наборы понятий (объектов) тематических разделов (Рисунок 4.29).

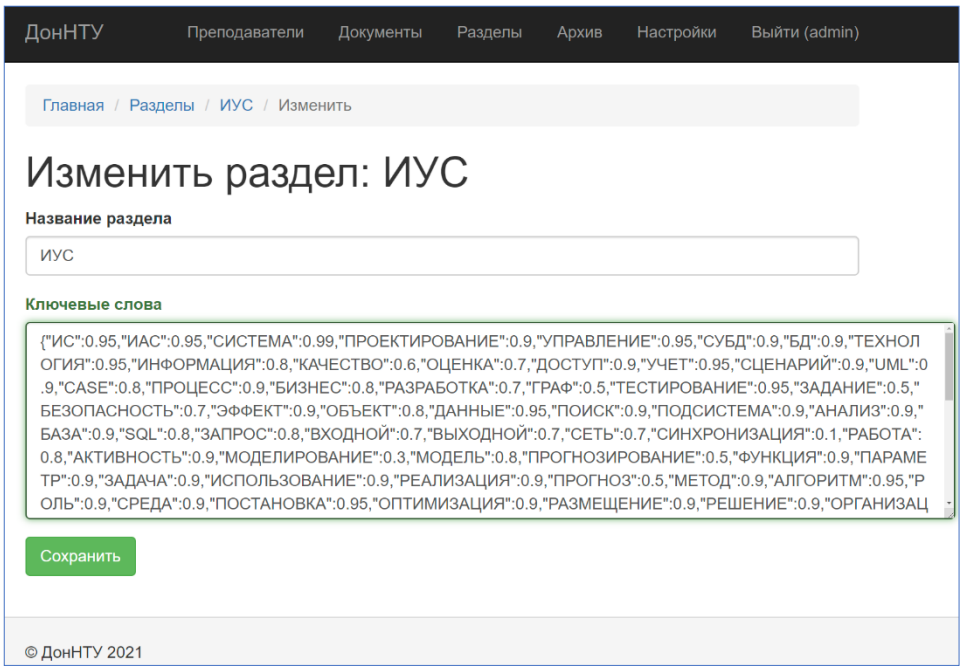


Рисунок 4.29 – Форма ведения тематических разделов

В системе предусмотрено определение различных параметров и переменных, обеспечивающих работу алгоритмов (Рисунок 4.30).

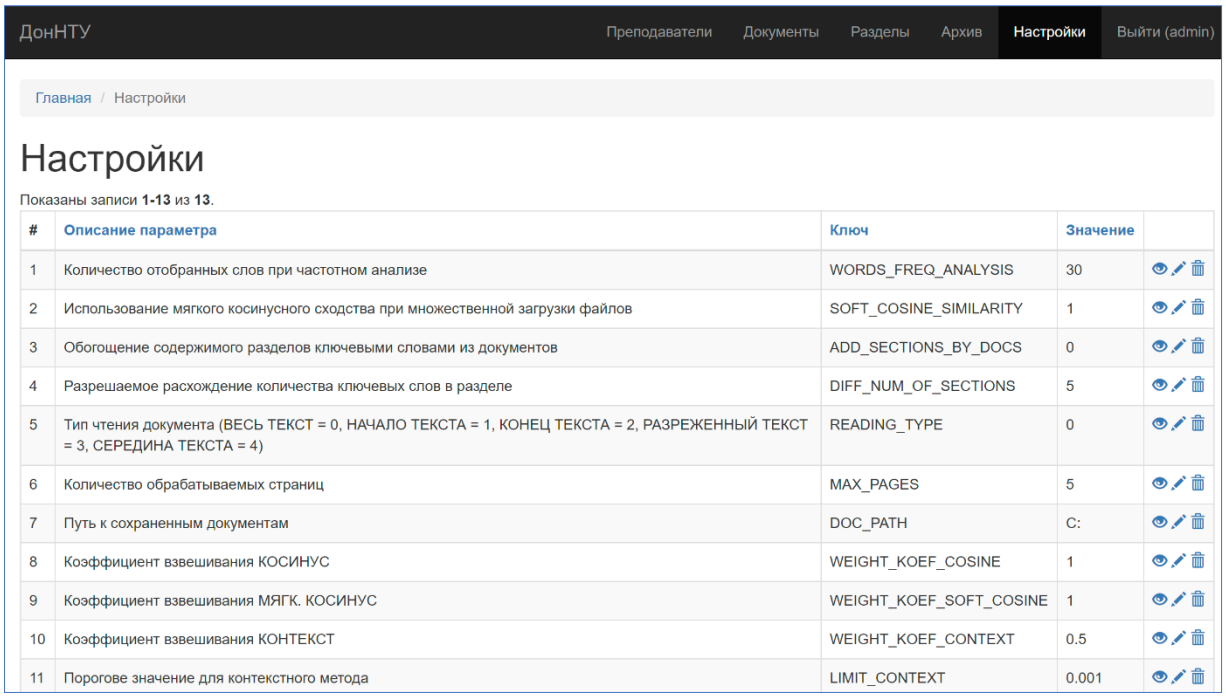


Рисунок 4.30 – Форма с настройками системы

Таким образом, тестирование системы показало результативность и корректность выполнения различных алгоритмов обработки информации.

4.4 Выводы

1. Разработана структурная архитектурная модель фреймворка системы управления информационными ресурсами научно-образовательных организаций, реализующая разработанные модели и алгоритмы.
2. На базе фреймворка разработано программное приложение для учета научной и исследовательской деятельности преподавателей.
3. Выполнено тестирование программных модулей, которое показало работоспособность, корректность полученных результатов, приемлемое время решения поисковых задач и задачи классификации.

ЗАКЛЮЧЕНИЕ

Диссертационная работа является законченной научно-исследовательской работой, в которой дано новое решение актуальной научно-технической задачи повышения эффективности системы управления информационными ресурсами научно-образовательных учреждений за счет применения онтологического подхода, разработки новых и усовершенствования существующих моделей и алгоритмов поиска, хранения и классификации данных.

Основные результаты и выводы, полученные при выполнении работы состоят в следующем.

1. На основе анализа существующих разработок и исследований сделан вывод об актуальности выбранной тематики, о перспективности использования онтологий в качестве модели представления знаний и о целесообразности онтологического подхода к разработке системы управления ИР.

2. Разработана гибридная процедура формирования онтологии, когда на начальном этапе создания «базовой» онтологии использованы различные онтологии «верхнего уровня» и метод «экспертного создания». Для построения и пополнения нижних уровней онтологии предметных областей обоснован выбор словаря Wiktionary, для которого показатель качества поиска, выраженного F-мерой, достиг наибольшего среди тестируемых значения 0.87. Разработан способ пополнения онтологии с использованием DBpedia, при этом концепты могут быть извлечены из документов с точностью 87% и полнотой 78%.

3. Разработана концептуальная метамодель для учета связанности знаний и обеспечения однородности представления данных в рамках единой тематики проектируемой системы, ядром которой является онтология.

4. Модифицирована модель N-мерного представления знаний на базе RDF-графа в виде трехмерного тензора семантических связей, значения которого определяются различным образом для различных отношений RDF-графа знаний и содержат значения в диапазоне [0; 1], что позволило для отдельных видов отношений учитывать не только наличие связей, но и их силу.

5. Усовершенствована гибридная мера оценки семантической близости на базе модели N-мерного представления RDF-графа знаний, что дало возможность определять сходство с учетом семантики, частотных характеристик текста, контекста и структуры онтологии и улучшить качество поиска. Для определения весовых коэффициентов интеллектуальной гибридной меры оценки СБ использован генетический алгоритм.

6. Разработаны две редуцированные модели представления текстовых информационных ресурсов: без трансформации признакового пространства и с трансформацией признакового пространства. Экспериментальные исследования показали целесообразность использования модели без трансформации признакового пространства, которая использует редукцию по тематическим разделам онтологии и снижает размерность задачи до размера контекстного вектора тематического раздела, что позволило для количества концептов онтологии, равном 2250 объектов, и размеру контекстного вектора, равному 30 элементов повысить скорость выполнения базовых алгоритмов более чем в 40 раз практически без потерь качества поиска (на 6.2%).

7. Усовершенствован алгоритм поиска данных с использованием онтологии, гибридной меры оценки СБ и векторной модифицированной модели представления текстов, что привело к повышению качества модели информационного поиска, выраженного F-мерой, на 10.7% по сравнению с алгоритмом, использующим меру «косинусного сходства».

8. Усовершенствован алгоритм классификации данных с использованием гибридной меры оценки СБ, что привело к повышению качества модели классификации, выраженного F-мерой, на 45.4% по сравнению с алгоритмом, для которого СБ вычислялась только по онтологии, на 5.3% по сравнению с алгоритмом, использующим меру «косинусного сходства» и на 9.5% по сравнению с алгоритмом на базе «мягкой косинусной меры».

9. Разработана структурная архитектурная модель фреймворка системы управления информационными ресурсами научно-образовательных учреждений, реализующая предложенные модели и алгоритмы. Тестирование созданного на

базе фреймворка программного приложения для учета научной деятельности сотрудников «Наука» показало корректность полученных результатов и приемлемое для пользователя время решения поисковых классификационных задач.

СПИСОК ЛИТЕРАТУРЫ

1. Андриевская, Н.К. Основные принципы и подходы при разработке системы управления профессиональными знаниями ВУЗа / Н.К. Андриевская // Научный журнал «Информатика и кибернетика». – 2019. – № 4 (18). – С. 49–56.

2. Babar, M.A. Software architecture knowledge management / M.A. Babar, T. Dingsøyr, P. Lago, H. van Vliet. – Springer, 2009.

3. De Boer, R.C. Experiences with semantic wikis for architectural knowledge management / R.C. De Boer and H. Van Vliet // WICSA. – IEEE. – 2011. – P. 32– 41.

4. Ameller, D. Ontology-based architectural knowledge representation: structural elements module / D. Ameller, X. Franch. // Advanced Inform. Syst. Eng. Workshops. – Springer. – 2011. – P. 296 – 301.

5. Bhat, M. Meta-model-based framework for architectural knowledge management / M. Bhat, K. Shumaiev, A. Biesdorf, U. Hohenstein, M. Hassel, F. Matthes // Proceedings of the 10th ECSA Workshops. – ACM. – 2016. – P. 12.

6. Kruchten, P. An ontology of architectural design decisions in software intensive systems // 2nd Groningen workshop on software variability. – Citeseer. – 2004. – P. 54 – 61.

7. Kruchten, P. The decision view's role in software architecture practice / P. Kruchten, R. Capilla, J. C. Duen˜as // IEEE software. – 2009. – Vol. 26(2). – P. 36–42.

8. Lytra, I. Supporting consistency between architectural design decisions and component models through reusable architectural knowledge transformations / I. Lytra, H. Tran, U. Zdun // ECSA. – Springer. – 2013. – P. 224–239.

9. Тузовский, А.Ф. Системы управления знаниями (методы и технологии) / А.Ф. Тузовский, С.В. Чириков, В.З. Ямпольский // Под общ. ред. В.З. Ямпольского. – Томск: НТЛ. – 2005. – 260 с.

10. Лапшин, В.А. Онтологии в компьютерных системах / В.А. Лапшин. – М.: Научный мир. – 2010. – 222 с.

11. Олейник, А.Г. Разработка онтологий интегрированного пространства знаний / А.Г. Олейник, П.А. Ломов // Онтология проектирования. – 2016. –Т.6. –№ 4 (22). – С. 465–474.

12. Гаврилова, Т.А. Инженерия знаний. Модели и методы / Т.А. Гаврилова, Д.В. Кудрявцев, Д.И. Муромцев. – СПб.: Лань. – 2016. – 324 с.

13. Закон РФ «Об информации, информационных технологиях и о защите информации» от 27.07.2006 № 149-ФЗ.

14. ГОСТ 34.321-96 Информационные технологии. Система стандартов по базам данных. Эталонная модель управления данными, п. 27.

15. Маличенко, И.П. Управление знаниями как эффективный механизм формирования непрерывной системы обучения и развития персонала в организации / И.П. Маличенко // Вестник НГУЭУ. – 2016. – № 1. – С. 174–188.

16. Абдикеев, Н.М. Управление знаниями корпорации и реинжиниринг бизнеса: Учебник / Н.М. Абдикеев, А.Д. Киселев; под науч. ред. Н.М. Абдикеева. – Москва: НИЦ ИНФРА-М. – 2015. – 382 с.

17. Гейтс, Билл. Бизнес со скоростью мысли. Изд. 2-е, исправленное: Эксмо. – Москва. – 2003. – 118 с.

18. Вольфсон, Ю.Р. Проблема классификации теорий информационного общества / Ю.Р. Вольфсон, А.Е. Вольчина // Современные исследования социальных проблем. – 2017. –Том 8. – № 3. – С. 80–109.

19. Козачков, Л.С. Некоторые методологические проблемы теории инфомационного поиска / Л.С. Козачков // Научно-Техническая Информация. – 1969. – Серия 2. – No. 12.

20. Козачков, Л.С. Категориальные Тезаурусы в Базах Данных / Л.С. Козачков // Научно-Техническая Информация. – 1985. – Серия 2. – No. 5. – С.11–19.

21. Wiig, K.M. Introducing knowledge management into the enterprise // Knowledge management handbook / Ed. by J. Liebowitz. – NY: CRCPress. – 1999. – P. 3.1 – 3.41.

22. Davenport, T. Working Knowledge: how organizations manage what they know. / T. Davenport, L. Prusak. – Boston. – Harvard Business School Press. – 1998.

23. Нонака, И. Компания – создатель знания. Зарождение и развитие инноваций в японских фирмах (The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation) / И. Нонака, Х. Такеучи. – М.: Олимп-Бизнес. – 2003. – 384 с.

24. Ромер, Д. Высшая макроэкономика [Текст]: учебник / Дэвид Ромер; пер. с англ. под науч. ред. В. М. Полтеровича; Нац. исслед. ун-т «Высшая школа экономики». – М.: Изд. дом Высшей школы экономики. –2014.– 855 с.

25. Новая постиндустриальная волна на Западе. Антология / Под редакцией В.Л. Иноземцева. – М.: Academia. – 1999. – 640 с.

26. Мильнер, Б.З. Управление знаниями. Эволюция и революция в организации. – М.: ИНФРА-М. – 2003.

27. Макаров, В.Л. Микроэкономика знаний / В.Л. Макаров, Г.Б. Клейпер. – 2007.

28. Gruber, T.R. Toward Principles for the Design of Ontologies Used for Knowledge Sharing // International Journal of Human-Computer Studies. – Vol. 43. – P. 907–928.

29. Gruber, T.R. The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases // Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference. – Cambridge, MA, Morgan Kaufmann. – 1991. – P. 601–602.

30. Guarino, N. Ontologies and Knowledge Bases. Towards a Terminological Clarification / N. Guarino, P. Giaretta // Towards Very Large Knowledge Bases. – IOS Press, Amsterdam. – 1995.

31. Guarino, N. Formalizing Ontological Commitment. / N. Guarino, M. Carrara, P. Giaretta // Proceedings of AAAI-94. – Vol. 1. – 1994. – P. 560–567.

32. Guarino, N. Handbook on ontologies: What Is an Ontology? / N. Guarino, D. Oberle, S. Staab, R. Studer. – Springer. – 2009. – P. 2–3.

33. Kifer, M. A Realistic Architecture for the Semantic Web / Michael Kifer, Jos de Bruijn, Harold Boley, and Dieter Fensel. – 2005.

34. Гаврилова, Т.А. Управление знаниями: ЧТО ДЕЛАТЬ? Сб. документов седьмой научно-практической конференции “Реинжиниринг бизнес-процессов на основе современных информационных технологий. Системы управления знаниями” (РБП-СУЗ-2004). – М.: Московский государственный университет экономики, статистики и информатики – С. 61–67.

35. Гаврилова, Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, Ф.В. Хорошевский – СПб.: Питер. – 2001. – 384с.

36. Смирнов, А.В. Онтологии в системах искусственного интеллекта: возможности построения и организации / А.В. Смирнов, М.П. Пашкин, Н.Г. Шилов // Новости искусственного интеллекта. – 2002. – №1.

37. Осипов, Г.С. Построение модели предметных областей. Неоднородные семантические сети // Известия АН СССР. Техническая кибернетика. – 1990. – №5. – С. 32–45.

38. Рубашкин, В.Ш. Онтологическая семантика. Знания. Онтологии. Онтологически ориентированные методы информационного анализа текстов. – М.: ФИЗМАТЛИТ. – 2012. –348 с.

39. Гладун, А.Я. Онтологии в корпоративных системах. Раздел: Информационные технологии / А.Я. Гладун, Ю.В. Рогушина // Корпоративные системы – 2006. –№1. – С. 41–47.

40. Berners-Lee, T. The Semantic Web. A New Form of Web Content That Is Meaningful to Computers Will Unleash a Revolution of New Possibilities / T. Berners-Lee, J. Hendler, O. Lassila. – Scientific American. – 2001. – P.1–5.

41. Staab, S. Handbook on ontologies / S. Staab, R. Studer. – Springer Science & Business Media. –2010.

42. Legat, C. Model-based Knowledge Extraction for Automated Monitoring and Control / C. Legat, J. Neidig, M. Roshchin // Preprints of the 18th IFAC World Congress. – 2011.

43. Beimel, D. Using OWL and SWRL to represent and reason with situation-based access control policies / D. Beimel, M. Peleg // Data & Knowledge Engineering. – 2011. – №70.

44. Martinez-Romero, M. The iOSC3 System: Using Ontologies and SWRL Rules for Intelligent Supervision and Care of Patients with Acute Cardiac Disorders / M. Martinez-Romero, J. Vazquez-Naya, J. Pereira, M. Pereira, A. Pazos, G. Bafios. // Computational and Mathematical Methods in Medicine. – 2013.

45. Paraiso, E. An Ontology-Based Utterance Interpretation in the Context of Intelligent Assistance / E. Paraiso, J. Barthés // Programa de Pés-Graduacao em Informatica. – 2007.

46. Martin, A. A Business Intelligence Model to Predict Bankruptcy using Financial Domain Ontology with Association Rule Mining Algorithm / A. Martin, M. Manjula, Dr. V. P. Venkatesan // IJCSI. – 2011.

47. Latfi, F. Ontology-Based Management of the Telehealth Smart Home, Dedicated to Elderly in Loss of Cognitive Autonomy/ F. Latfi, B. Lefebvre, C. Descheneaux // Proceeding of the OWLED. – 2007.

48. Nirenburg, S. Ontological Semantics / S. Nirenburg, V. Raskin – MIT Press, 2004. – 339 p.

49. Akerman, A. Using ontology to support development of software architectures / A. Akerman, J. Tyree // IBM Systems Journal (45:4). – IBM. – 2006. – P. 813–825.

50. Ameller, D. Ontology-based architectural knowledge representation: structural elements module / D. Ameller, X. Franch, // in International Conference on Advanced Information Systems Engineering. – 2011. – P. 296–301.

51. Kruchten, P. An ontology of architectural design decisions in software intensive systems // in 2nd Groningen workshop on software variability. – 2004. – P. 54 – 61.

52. Soliman, M. Improving the Search for Architecture Knowledge in Online Developer Communities / M. Soliman, A. Rekaby Salama, M. Galster, O. Zimmermann and M. Riebisch // IEEE International Conference on Software Architecture (ICSA). – 2018. – P. 186–189.

53. Segura, Y.C. Description and analysis of design decisions: An ontological approach / Y.C. Segura, N.S. Mart // in CITI. – Springer. – 2018. –P. 174–185.

54. Astudillo, H. Automated recovery of design decisions and structure in ANDESCON / H. Astudillo, G. Vald // IEEE Computer Society. – 2012. – P. 105–108.

55. Гелеверя, Т.Е. Porto: Средство визуального проектирования web-порталов. / Т.Е. Гелеверя, В.А. Горовой, Д.О. Горовая. // Труды 5-й всероссийской объединенной конференции «Технологии информационного общества - Интернет и современное общество». – СПбГУ. – 2002. – С.82–84.

56. Gavrilova, T. Ontological Engineering for Corporate Knowledge Portal Design / T. Gavrilova, V. Gorovoy // In "Processes and Foundations for Virtual Organisations", Eds. L. Camarinha-Matos and H. Afsarmanesh. – Kluwer Academic Publishers. – 2003. – P.289–296.

57. Безгинова, Ю.А. Практики управления знаниями в нефтяных компаниях. / Ю.А. Безгинова, Т.А. Гаранина, Д.В. Кудрявцев, А. Ю. Плешкова // Открытое образование. – Российский экономический университет им. Г.В. Плеханова (Москва). – 2018. – Том 22. – № 6. – С. 27–38.

58. Фокина, С.Н. Опыт формирования системы управления знаниями в акционерном обществе «Институт реакторных материалов» / С.Н. Фокина, О.А. Сиденко, К.И. Ильин, Р.Г. Бильданов // Интеллектуальная собственность и инновации: материалы X международной научно-практической конференции. Екатеринбург, 26 апреля 2018 г. – Екатеринбург: УрФУ. – 2018. – С. 239–251.

59. Kontopoulos, E. An Ontology-based Planning System for e-Course Generation / E. Kontopoulos, D. Vrakas, F. Kokkoras, N. Bassiliades, I. Viahavas // Expert Systems with Applications. – 2008. –№35. – P.1–2.

60. Садовничий, В.А. Интеллектуальная система тематического исследования наукометрических данных: предпосылки создания и методология разработки. Часть 1 / В.А. Садовничий, В.А. Васенин // Программная инженерия. – 2018. –Том 9. – № 2. – С. 51–58.

61. Журавлев С.В., Юдина Т.Н., Информационная система РОССИЯ // НТИ. Сер.2. — 1995. — № 3. — С.18-20.

62. Wasson M. Classification Technology at LexisNexis. // SIGIR 2001. – Workshop on Operational Text Classification

63. Баланова, Л.А. Модели представления знаний: виды, применение, достоинства и недостатки / Л.А. Баланова , Е.В. Ющенко // Материалы XII Международной студенческой научной конференции «Студенческий научный форум» URL: https://scienceforum.ru/2020/article/2018018540 (дата обращения: 15.05.2021).

64. Адамова, Л.Е. Миварное понимание текста: разработка методики обучения виртуальной личности предметным знаниям на основе создания сетей концептов / Л.Е. Адамова, Е.А. Скакунова, О.О. Варламов, А.О. Петерсон, Д.А. Протопопова // Автоматизация и управление в технических системах. – 2014. – № 3 – Режим доступа: https://auts.esrae.ru/11-210 (дата обращения 15 декабря 2020).

65. Melucci, Massimo. Vector-Space Model. Encyclopedia on Database Systems. – 2009.

66. TF-IDF [Электронный ресурс]: Википедия. Свободная энциклопедия. – Режим доступа: https://ru.wikipedia.org/wiki/TF-IDF (дата обращения 28 декабря 2020).

67. Word2vec [Электронный ресурс]: Википедия. Свободная энциклопедия. – Режим доступа: https://ru.wikipedia.org/wiki/Word2vec (дата обращения 28 декабря 2020).

68. Нугуманова, А.Б. Обогащение модели Bag of words семантическими связями для повышения качества классификации текстов предметной области / А.Б. Нугуманова, И.А. Бессмертный, П. Пецина, Е.М. Байбурин // Программные продукты и системы. – 2016. – № 2. – С. 89–99.

69. Адамов, Б.И. Применение основных матричных разложений в задачах механики и робототехники / Б.И. Адамов, А.Н. Маслов, Н.В. Осадченко. – М.: Издательство МЭИ. – 2019. – 84 с.

70. Silva C. Knowledge Extraction with Non-Negative Matrix Factorization for Text Classification / C. Silva, B. Ribeiro // In: Corchado E., Yin H. (eds) Intelligent Data Engineering and Automated Learning. IDEAL 2009. Lecture Notes in Computer Science.

– Springer, Berlin, Heidelberg. – Vol. 5788. https://doi.org/10.1007/978-3-642-04394-9_37.

71. Ponte J.M. A Language modeling Approach to InformationRetrieval / J.M. Ponte, W.B. Croft // Proc. Conference on Research and Development in Information Retrieval. – ACM. – 1998. – C. 275–281.

72. Manning, C.D. Foundations of Statistical Natural Language Processing / C.D. Manning, H. Sch¨utze. – Second printing with corrections. – The MIT Press. – 1999. – 680 p. — ISBN: 0262133601.

73. Martin, D. Speech and Language Processing. / D. Martin, D. Jurafsky // An introductionto natural language processing, computational linguistics, and speechrecognition. – 2000.

74. Gomaa, W.H. A Survey of Text Similarity Approaches / W.H. Gomaa, A.A. Fahmy //International Journal of Computer Applications. – 2013. –Т. 68. – № 13.

75. Salton, G. A vector space model for automatic indexing / G. Salton, A. Wong, C.S. Yang // Communications of the ACM. –1975 – Vol.18. – P. 613-620.

76. Maron, M.E. On relevance, probabilistic indexing and information retrieval / M.E. Maron, J.L. Kuhns // Journal of the ACM. – 1960. – Vol. 7(3). – P. 216–244.

77. Fox, E. A. Extended Boolean information retrieval / Edward A. Fox, G. Salton, H. Wu // Communications of the ACM. – Vol. 26(11). – 1983. – P.1022–1036.

78. Deerwester, S. Indexing by latent semantic analysis. / Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Richard Harshman // Journal of the American Society for Information Sciencею. – 1990.

79. Kwok, K.L. A neural network for probabilistic information retrieval // ACM SIGIR Forum. – 1989. – Vol. 23.

80. Hsinchun, C. Machine learning for information retrieval: Neural networks, symbolic learning, and genetic algorithms // Journal of the American Society for Information Science. – Vol. 46(3). – P.194–216.

81. Крюков, К.В. Меры семантической близости в онтологии / К.В. Крюков, Л.А. Панкова, В.А. Пронина, В.С. Суховеров, Л.Б. Шипилина // Проблемы управления. – 2010. – № 5. – С. 2–14.

82. Косинусная мера [Электронный ресурс]: Википедия. Свободная энциклопедия. – Режим доступа: https://ru.wikipedia.org/wiki/ Векторная модель # Косинусная мера (дата обращения 28 декабря 2020).

83. Similarity_of_asymmetric_binary_attributes [Электронный ресурс]: site:wikichi.ru – Режим доступа: https://wikichi.ru/wiki/ Jaccard_index# Similarity_of_asymmetric_binary_attributes (дата обращения 28 декабря 2020).

84. «Мягкая» косинусная мера [Электронный ресурс]: Википедия. Свободная энциклопедия. – Режим доступа: https://ru.wikipedia.org/wiki/ Векторная модель #«Мягкая» косинусная мера (дата обращения 28 декабря 2020).

85. Бова, В.В. Эволюционный подход к решению задачи интеграции онтологий / В.В. Бова, Д.В. Заруба, В.В. Курейчик // Известия ЮФУ. Технические науки. – №6(167). – 2015. – С. 41–56.

86. Семенова, А.В. Оптимизация отображения онтологий методом роя частиц / А.В. Семенова, В.М. Курейчик // Онтология проектирования. – 2018. – Т.8. – №2(28). – С. 285–295.

87. Оценка качества классификации [Электронный ресурс]: – Режим доступа: https: // neerc.ifmo.ru/wiki/index.php?title=Оценка качества классификации (дата обращения 28 декабря 2020).

88. Чайка, В.А. Обзор средств разработки онтологических моделей / В.А. Чайка, С.Ю. Землянская, Н.К. Андриевская // В сборнике: Информатика, управляющие системы, математическое и компьютерное моделирование (ИУСМКМ-2020). Сборник материалов XI Международной научно-технической конференции в рамках VI Международного Научного форума Донецкой Народной Республики. Редколлегия: Ю.К. Орлов [и др.]. – 2020. – С. 233–237.

89. Бажанова, А.И. Исследование применения онтологических моделей для семантического поиска / Бажанова А.И., Мартыненко Т.В. // Информационные управляющие системы и компьютерный мониторинг (ИУСиКМ-2011) / Материалы II научно-технической конференции студентов, аспирантов и молодых ученых. – Донецк. – ДонНТУ. –2011. – С. 244–248.

90. Андриевская, Н.К. Разработка прикладной онтологии в системах обработки данных научных и научно - образовательных организаций / Андриевская Н.К. // Вестник ДонНУ. Сер. Г: Технические науки. – 2020. – № 3. – С. 43–51.

91. Jena documentation overview. URL: https://jena.apache.org/ documentation. Data access: 16.03.2020.

92. FOAF Vocabulary Specification 0.99. URL: http://xmlns.com/ foaf/spec/20140114.html. Data access: 16.03.2020.

93. VIVO Ontology for Researcher Discovery. URL: https://bioportal.bioontology.org/ontologies/VIVO/?p=properties Data access: 16.03.2020.

94. VIVO Ontology Domain Definition. URL: https://wiki.lyrasis.org/ display/ VIVODOC110x/VIVO+Classes?Preview=/96996014/96996018/ VIVO%20 Classes.png. Data access: 16.03.2020.

95. Bibliographic Ontology (BIBO) in RDF URL: https://www.dublincore.org/specifications/bibo/bibo/bibo.rdf.xml Data access: 16.03.2020.

96. DBpedia Ontology - DBpedia Association. URL: https://www. dbpedia. org / resources/ontology/. Data access: 16.03.2020.

97. Teaching Core Vocabulary Specification URL: http://linkedscience.org/ teach/ns/#. Data access: 16.03.2020.

98. vCard Ontology - for describing People and Organizations. URL: https://www.w3.org/TR/vcard-rdf/#classes. Data access: 16.03.2020.

99. Фролова, Н.Б. Разработка OWL-онтологии образовательных ресурсов СГТУ / Н.Б. Фролова // Вестник ВГУ, Серия: Системный анализ и информационные технологии. – 2016. – № 3. – С.149–158.

100. Бажанова, А.И. Разработка морфологического анализатора для построения понятийного аппарата электронной библиотеки кафедры АСУ / А.И. Бажанова, Т.В. Мартынеко, Н.К. Андриевская // Информатика и компьютерные технологии / Сборник трудов VII международной научно-технической

конференции студентов, аспирантов и молодых ученых – 22-23 ноября 2011 г., Донецк. – ДонНТУ. – 2011. – В 2-х томах. – Т.1 – 417 с. – С.326–330.

101. Астраханцев, Н. Автоматическое извлечение терминов из коллекции текстов предметной области с помощью Википедии / Н. Астраханцев // Труды Института системного программирования РАН. – 2014. – Т. 26. – № 4. – С. 7–20.

102. Андриевская, Н.К. Анализ возможностей использования существующих словарей для пополнения онтологии / Н.К. Андриевская, А.И. Секирин, С.В. Канатуш // Научный журнал «Информатика и кибернетика». – 2020. – № 2 (20). – С. 13–21.

103. WordNet. A Lexical Database for English. – URL: https://wordnet.princeton.edu. Data access: 20.05.2020.

104. Проект RussNet. – URL: http://project. phil.spbu.ru/RussNet/ index_ru.shtml. Data access: 20.05.2020.

105. Портал ресурса MediaWiki – URL: https://www.mediawiki.org/ wiki/MediaWiki. Data access: 20.05.2020.

106. Портал ресурса Semantic_MediaWiki – URL: https://www.semantic-mediawiki.org/wiki/ Semantic_MediaWiki. Data access: 20.05.2020.

107. Help: Ontology import – URL: http://semantic-mediawiki.org/ wiki/Help:Ontology_ import. Data access: 20.05.2020.

108. Python Wikipedia Bot Framework. Manual: Pywikibot / Overview – URL: https://www.mediawiki.org/wiki/Manual:Pywikibot/Overview. Data access: 20.05.2020.

109. Project description – URL: https://pypi. org/project/rdflib/. Data access: 20.05.2020.

110. Translate API – URL: https://tech.yandex.com/translate/ Data access: 20.05.2020.

111. Русский Викисловарь – URL: http://ru.wiktionary.org/. Data access: 20.05.2020.

112. Крижановский, А.А. Автоматическое извлечение словарных помет из Русского Викисловаря / А.А. Крижановский, А.В. Смирнов, В.М. Круглов, Н. Б.

Крижановская, И.С. Кипяткова // Труды СПИИРАН. – 2014. – Вып. 33. – С. 164 – 185.

113. Bhat, M. An Ontology-based Approach for Software Architecture Recommendations / Manoj Bhat, Klym Shumaiev, Andreas Biesdorf, Uwe Hohenstein, Michael Hassel, Florian Matthes // 23rd Americas Conference on Information Systems (AMCIS). – At: Boston, MA, USA. – 2017.

114. Querying the DBpedia Open Knowledge Graph with Standard SQL. URL: https: //dzone.com/articles/timbr-dbpedia-or-how-to-query-the-dbpedia-open-kno. Data access: 10.10.2019.

115. SPARQL. URL: https://ru.bmstu.wiki/SPARQL. Data access: 10.10.2019.

116. Prud'Hommeaux, E. et al. 2008. "SPARQL query language for RDF," W3C Recommendation (15). Data access:12.12.2017.

117. Unstructured information management applications. Apache, UIMA. –2016. URL: http://uima.apache.org. Data access: 10.10.2019.

118. Андриевская, Н.К. Онтологический подход в системах обработки данных научных и научно-образовательных организаций / Н.К. Андриевская // Международный научно-теоретический журнал «Проблемы искусственного интеллекта». – 2020. – № 1 (16). – С. 23–36.

119. RDF - Semantic Web Standards [Электронный ресурс]: w3.org. – Режим доступа: https://www.w3.org/RDF / (дата обращения 28 декабря 2020).

120. Nickel M. A Three-Way Model for Collective Learning on Multi-Relational Data / M. Nickel, V. Tresp, H. Kriegel // ICML. – 2011. – Vol. 11.

121. Kolda, Tamara G. Tensor Decompositions and Applications / Tamara G. Kolda, Brett W. Bader // SIAM Rev. – 2009. –Vol. 51(3) – P. 455–500.

122. Nickel, M. A review of relational machine learning for knowledge graphs / M. Nickel et al. // Proceedings of the IEEE. – 2015. – Vol.104. –№1. – P.11–33.

123. Андриевская, Н.К. Гибридный интеллектуальный способ оценки семантической близости / Н.К. Андриевская // Международный научно-теоретический журнал «Проблемы искусственного интеллекта». – 2021. – № 1 (20). – С. 4–17.

124. Канатуш, С. В. Онтологический подход к веб-поиску / С.В. Канатуш, Н.К. Андриевская // Современные проблемы радиоэлектроники и телекоммуникаций: сб. науч. тр. под ред. Ю. Б. Гимпилевича. – Москва-Севастополь: Изд-во: РНТОРЭС им. А.С. Попова. – 2020. –№ 3. – 247 с. – С. 205.

125. Андриевская, Н.К. Обобщенная модифицированная модель представления текстовых информационных ресурсов. / Н.К. Андриевская // Научный журнал «Информатика и кибернетика». – 2020. –№ 4 (22). – С. 38–47.

126. Вильчевская, Е.Н. Тензорная алгебра и тезорный анализ: учеб. пособие / Е.Н. Вильчевская. — СПб.: Изд-во Политехн. ун-та. – 2012. – С. 4.

127. Kossaifi, J. TensorLy: Tensor Learning in Python / Jean Kossaifi, Yannis Panagakis, Anima Anandkumar, Maja Pantic // JMLR. – 2019. – Vol. 20(26). – P.1–6.

128. Lathauwer, L. Multilinear Singular Value Tensor Decompositions. / L Lathauwer, Bart DeMoor, Joos Vandewalle // SIAM Journal on Matrix Analysis and Applications (SIAM J MATRIX ANAL A). – 2000. – Apl. 24.

129. Amnon, Sh. Non-negative tensor factorization with applications to statistics and computer vision. / Shashua Amnon, Tamir Hazan // In Proceedings of the 22nd international conference on Machine learning (ICML '05). – Association for Computing Machinery, New York, NY, USA. – 2005. – P. 792–799.

130. Michael, P. Computing nonnegative tensor factorizations / P. Michael, Kathrin Hatz. Friedlander. – Department of Computer Science Technical Report TR-2006-21. – October 2006. – University of British Columbia.

131. Ma, S. Tensor models: Solution methods and applications / S. Ma, B. Jiang, X. Huang, S. Zhang, S. Cui, A. Hero, Z. Luo, J. Moura (Eds.) // In Big Data over Networks Cambridge: Cambridge University Press. –2016. – P. 3–36.

132. Basipov, A.A. Semantic search: issues and technologies / A. A. Basipov, O. V. Demich // Вестник АГТУ. Серия: «Управление, вычислительная техника и информатика». – 2012. – №1. – С. 104–111.

133. Тявкин, И.В. Математическая модель информационного поиска и оценка эффективности поисковой системы / И.В. Тявкин, В.М. Тютюнник // Вестник ТГТУ. – 2008. –Том 14. – № 3.

134. Дубинский, А.Г. Характеристики эффективности информационного поиска в сети Интернет // Научный сервис в сети Интернет: тез. докл. Всерос. науч. конф. – М.: Изд-во МГУ. – 2001. – С. 136–138.

135. Мальковский, М.Г. Прикладное программное обеспечение: системы автоматической обработки текстов / М.Г. Мальковский, Т.Ю. Грацианова, И.Н. Полякова. – М.: МАКС "Пресс". – 2000. – С. 64.

136. Kruchten, P. B. The 4+1 View Model of architecture // in IEEE Software. – Nov 1995. – Vol. 12. – № 6. – P. 42–50.

137. Андриевская, Н.К. Разработка архитектурной модели системы управления информационными ресурсами организаций / Н.К. Андриевская, А.И. Секирин, О.В. Ченгарь // В сборнике: Программная инженерия: методы и технологии разработки информационно-вычислительных систем (ПИИВС-2020). сборник научных трудов III Международной научно-практической конференции. – Донецк, 2020. – С. 46–54.

138. Светличная, В.А. Разработка функциональной структуры логистической системы формирования заказов для интернета-магазина / В.А. Светличная, Н.К. Андриевская, К.Ю. Чаленко // Научный журнал «Информатика и кибернетика». – Д.: ДонНТУ, 2017. – № 3(9). – С. 111–118.

139. Жан-Луи Марешо (Jean-Louis Maréchaux). Определение архитектуры приложений с помощью Rational Software Architect. [Электронный ресурс] – URL: https://www.ibm.com/ developerworks/ru/library/r-define-application-architecture-rational-software-architect1/index.html (дата обращения: 16.10.2020).

140. Простое руководство по UML-диаграммам и моделированию баз данных. [Электронный ресурс] – URL: https://www.microsoft.com/ru-ru/microsoft-365/business-insights-ideas/resources/guide-to-umldiagramming-and-database-modeling (дата обращения: 16.10.2020).

141. Леоненков, А. Самоучитель UML 2 / Александр Леоненков. – СПб.: БХВ-Петербург. – 2007. – 576 с.

ПРИЛОЖЕНИЕ А

ОПИСАНИЕ ПОПУЛЯРНЫХ СУЩЕСТВУЮЩИХ ОНТОЛОГИЙ ВЕРХНЕГО УРОВНЯ И МЕЖДОМЕННЫХ ОНТОЛОГИЙ

Таблица А.1 – Существующие онтологии верхнего уровня и междоменные онтологии

№	Описание
1	Top-level Cyc – содержит 2,2 миллиона утверждений (фактов и правил), описывающих более 250 тысяч термов, включая почти 15000 предикатов
2	Top-level Sowa's top-level ontology
3	Top-level SUMO
4	Marc – представляющая структуру формата MARC21 и Dublin Core на языке OWL
5	SKOS – Семейство формальных языков для описания классификационных схем, тезаурусов, авторитетных файлов. В настоящее время разработки ведутся в рамках W3C
6	Top-level BFO Basic Formal Ontology – 36 классов, в основном в медицине, ориентирована на создание онтологий в рамках научных исследований, не содержит свойств
7	Top-level GFO General Formal Ontology – 79 классов, в основном в медицине
8	Top-level DOLCE – около 1000 классов, в системах на базе веб-технологий
9	DBpedia междоменная онтология, основанная на наиболее часто используемых терминах в статьях Википедии. Он содержит более 685 классов, 2795 различных свойств и более 4,2 миллиона экземпляров
10	CERIF 2008 Основные объекты – это Person, Organisation Unit и Project
11	TOVE (Toronto Virtual Enterprise) – построение интегрированной модели, состоящей из следующих онтологий: операций, состояний и времени, организации, ресурсов, продуктов, производства, цены, количества
12	Top-level DUBLIN CORE. Описание библиографических ссылок – содержит выходные данные о публикации: о дате выхода, издании, серии, страницах, ISBN, ISSN, краткое содержание, комментарии и пр.
13	FOAF — это проект, посвященный связыванию людей и информации с помощью Интернета. Распространенная модель, которая может использоваться во многих ПрО. Онтология для описания домашних страниц, людей и социальных сетей. Пространство имен: http://xmlns.com/foaf/0.1/

Таблица А.1 (продолжение)

№	Описание
14	KACTUS – построение методологии многократного применения знаний о технических системах во время их жизненного цикла
15	BIBO – включает в себя основные понятия и свойства для описания библиографических ссылок на Semantic Web в RDF Пространство имен: http://purl.org/ontology/bibo/
16	FaBiO (FRBR-aligned Bibliographic Ontology) – онтология, позволяющая описывать библиографические объекты, которые содержат библиографические ссылки
17	CiTO (Citation Typing Ontology) – онтология, предназначенная для описания природы цитат в научных публикациях (факт или утверждение)
18	BiRO (Bibliographic Reference Ontology) – онтология, предназначенная для описания библиографических записей и ссылок, и их компиляцию в библиографические сборники и библиографические списки
19	C4O (Citation Counting and Context Characterisation Ontology) – онтология, которая позволяет оценивать цитаты из цитируемых источников по их числу и расположению в контакте
20	DoCO (Document Components Ontology) – онтология, которая содержит структурированный словарь компонентов документа, включает структурные блоки (например, параграф, раздел, глава) и функциональные блоки (например, введение, обсуждение, благодарность, список литературы, рисунок, приложение)
21	PSO (Publishing Status Ontology) – онтология, которая предназначена для описания состояния публикации на каждом этапе издательского процесса
22	PRO (Publishing Roles Ontology) – онтология, характеризующая роли агентов – людей, юридических лиц и вычислительных средств в процессе публикации
23	PWO (Publishing Workflow Ontology) – онтология для описания шагов в рабочих процессах, связанных с публикацией документа
24	EXPO (EXPeriment) – онтология для описания научных экспериментов, включающая около 200 концептов
25	FRBR (Functional Requirements for Bibliographic Records) – онтология, позволяющая описывать библиографические записи
26	Онтология VIVO [9] Пространства имен: http://vivoplus.aksw.org/ontology# и http://vivoweb.org/ontology/core#
27	Онтология TEACH [10], ориентирована на обучение и охватывает организационные аспекты (аудитория, корпус, преподаватель, студент). Пространство имен: http://linkedscience.org/teach/ns#

ПРИЛОЖЕНИЕ Б

ЭЛЕМЕНТЫ ПРИКЛАДНОЙ ОНТОЛОГИИ

Таблица Б.1 Свойства классов

Описание	Свойство	Тип
Address		
Страна	county	Str
Индекс	index	Num
Штат/Область	stateRegion	Str
Населенный пункт/Город	localityCity	Str
Улица/Проспект	streetAvenue	Str
Дом/Корпус	houseCorps	Num
БуквaДома/Корпуса	houseCorpsLetter	Str
Квартира	apartment	Num
Position		
Статус	positionStatus	Num
Сокращ. название	positionShortName	Str
Discipline		
Статус	disciplineStatus	Num
Сокращ. название	disciplineShortName	Str
Faculty		
Полное название	facultyName	Str
Основной корпус	facultyMainCorps	Num
Аудитория деканата	facultyAuditorium	Str
Телефон деканата	facultyPhone	Str
Почта	facultyMail	Str
Сайт	facultyWebsite	Str
ФИО декана	deanFio	Str
Department		
Полное название	departmentName	Str
Основной корпус	departmentMainCorps	Num
Аудитория кафедры	departmentAuditorium	Str
Телефон кафедры	departmentPhone	Str
Почта	departmentMail	Str
Сайт	departmentWebsite	Str
ФИО зав. кафедры	headDepartmentFio	Str
InformationResource		
Наименование	informationResourceName	Str
Описание	description	Str
ОсновнойЯзык	primaryLanguage	Str
ДатаСоздания/Публикации	informationResourceDate	Date
Год издания	informationResourceYear	Num
Рубрика	rubric	Str

Таблица Б.1 (продолжение)

Описание	Свойство	Тип
ФИО автора	authorInformationResourceFio	Str
Номер статьи	articleNumber	Str
Страницы статьи	articlePages	Str
Нгомера страниц статьи	numberArticlePages	Num
Список литературы	listOfReferences	Str
Список соавторов	listOfContributors	Str
Индекс цитируемости	citationIndex	Float
Издательство	publishingOffice	Str
ElectronicResource		
Размер	electronicResourceSize	Float
Ссылка в Интернете	internetLink	Str
Audio		
Продолжительность	audioDuration	Num
Document		
Количество страниц	documentNumberPages	Num
Издание	edition	Str
Год	documentYear	Num
Текст	documentText	Str
Учебный год	academicYear	Num
Семестр	semester	Num
Количество ставок	quantityRate	Float
Аудит часы всего	classroomTimeTotal	Num
Номер протокола утверждения осень	protocolNumberApprovedAutumn	Num
Дата протокола утверждения осень	protocolDateApprovedAutumn	Date
Номер протокола отчета осень	protocolNumberDoneAutumn	Num
Дата протокола отчета осень	protocolDateDoneAutumn	Date
Номер протокола утверждения весна	protocolNumberApprovedSpring	Num
Дата протокола утверждения весна	protocolDateApprovedSpring	Date
Номер протокола отчета весна	protocolNumberDoneSpring	Num
Дата протокола отчета весна	protocolDateDoneSpring	Date
Video		

Таблица Б.1 (продолжение)

Описание	Свойство	Тип
Продолжительность	videoDuration	Num
Book		
Номер тма	tomNumberBook	Num
Выпусков всего	issuesTotalBook	Num
MaterialsBook		
Тип сборника	collectionType	Str
Номер тома	tomNumberMaterialsBook	Num
Выпусков всего	issuesTotalMaterialsBook	Num
Patent		
НомерПатента	patentNumber	Num
Periodical		
Номер выпуска	isssueNumberPeriodical	Num
Дата выпуска	issueDatePeriodical	Date
SlideShow		
Количество слайдов	numberOfSlides	Num
Organization		
Наименование	organizationName	Str
OrganizationPart		
Научное направление	scientificDirection	Str
Направление подготовки	trainingDirection	Str
Профиль подготовки	trainingProfile	Str
Group		
Количество студентов	numberOfStudents	Num
Статус	groupStatus	Num
PartOfDocumentStructure		
Значение	valuePartOfDocumentStructure	Str
Contacts		
Email	contactsEmail	Str
Телефон	contactsPhone	Str
Reference		
Страница	referencePage	Num
Название	referenceName	Str
Год	referenceYear	Num
Ссылка в Интернет	internetLink	Str
ФИО автора	authorReference	Str
Person		
Фамилия	personSurname	Str
Имя	personName	Str
Отчество	personPatronymic	Str
Дата рождения	personBirthday	Date

Таблица Б.1 (продолжение)

Описание	Свойство	Тип
Пол	sex	Str
Email	personEmail	Str
Телефон	personPhone	Str
Статус	personStatus	Str
ScienceProfile		
Всего статей Google	totalGoogleArticles	Num
Цитируемость Google	citationGoogle	Float
Индекс Хирша Google	googleHirschIndex	Float
Индекс Хирша Яндекс	yandexHirschIndex	Float
Всего статей Scopus	totalScopusArticles	Num
Всего статей Яндекс	totalYandexArticles	Num
Цитируемость Яндекс	citationYandex	Float
Индекс Хирша Scopus	scopusHirschIndex	Float
Цитируемость Scopus	citationScopus	Float
Индекс Хирша Wos	wosHirschIndex	Float
Всего статей Wos	totalWosArticles	Num
Цитируемость Wos	citationWos	Float
Профиль Google	profileGoogle	Str
Профиль Яндекс	profileYandex	Str
Профиль Scopus	profileScopus	Str
Профиль Wos	profileWos	Str
Дата данных профиля	profileDataDate	Date
Student		
Номер зачетки	studentGradeNumber	Str
ScientificEvent		
Наименование	scientificEventName	Str
Дата	scientificEventDate	Date
Контактный телефон	scientificEventPhone	Str
Тематика	scientificEventTheme	Str
Наименование организации	scientificEventOrganizationName	Str
PersonalProtocol		
Номер	personalProtocolNumber	Num
Дата	personalProtocolDate	Date
Protocol		
Номер	protocolNumber	Num
Дата	protocolDate	Date
Curriculum		
Номер протокола	protocolNumberCurriculum	Num
Дата протокола	protocolDateCurriculum	Date
Номер плана	planNumberCurriculum	Num

Таблица Б.2. Объектные свойства

Domain	Отношение	Range	Описание
InformationResource	hasInformationalPart	PartOfDocumentStructure	ИР имеет информационную часть: автор, тема, ключевые слова, аннотация, ссылка и т.д.
Organization	InformationResource hasInformationResource	InformationResource	Организация имеет какой-либо ИР (например, патент на инновационный двигатель принадлежит ДонНТУ)
InformationResource	usedInEvent	ScientificEvent	ИР “участвовал” в мероприятии
InformationResource	suitableForDiscipline	Discipline	Инф. ресурс может быть использован в данной дисциплине
Discipline	InformationResource hasInformationResource	InformationResource	У дисциплины существует множество информационных ресурсов
Organization	hasAddress	Address	Организация имеет адрес
Organization	hasEmployees	Person	У организации имеются сотрудники
Organization	hasScientificEvents	ScientificEvent	Организация проводит под своей эгидой научные мероприятия
Organization	hasBoss	Person	У организации есть главный руководящий сотрудник
Organization	hasScientificProfile	ScienceProfile	У организации есть научный профиль
HigherEducation	hasRector	Person	У ВУЗа есть ректор
Group	hasHeadman	Person	В группе имеется староста
Group	hasStudents	Student	В группе есть студенты
Student	studyingInGroup	Group	Студент учится в группе
Group	вincludedInDepartment	Department	Группа является частью определенной кафедры
Department	hasGroup	Group	Кафедра имеет группы
Department	hasHeadOfDepartment	Person	У кафедры есть заведующий кафедрой

Таблица Б.2 (продолжение)

Domain	Отношение	Range	Описание
Department	hasTeachers	Teacher	Преподаватель числится за определенной кафедрой
Faculty	hasDepartment	Department	Факультет имеет кафедры
Department	includedInFaculty	Faculty	Кафедра является частью определенного факультета
Faculty	includedInOrganization	Organization	Факультет принадлежит организации
Organization	hasFaculty	Faculty	Организация имеет факультеты
Group	studyTheDiscipline	Discipline	Группа изучает/ла данную дисциплину
Person	hasScientificProfile	ScienceProfile	У человека есть научный профиль
ScienceProfile	assignedToPerson	Person	Научный профиль связан с человеком
Person	participatedInScientificEvent	ScientificEvent	Человек участовал в научном мероприятии
ScientificEvent	hasParticipant	Person	Мероприятие имеет участников
Person	organizationMember	Organization	Человек состоит в организации
Person	authorOfInformationResource	InformationResource	Человек является автором информационного ресурса
Person	hasStatus	Condition	Человек имеет научную степень и/или научное звание или должность
ScientificEvent	hasAddress	Address	Научное мероприятие будет проходить по определенному адресу

ПРИЛОЖЕНИЕ В

ФРАГМЕНТЫ ПРОГРАММНЫХ МОДУЛЕЙ

```
<?php
namespace backend\modules\document\services\vsm;
use backend\modules\document\services\parser\ParserFrequency;
use backend\modules\settings\models\Settings;
class Vsm
{
    private $vsm;
    private $limit = 5;
    public function __construct()
    {
        $settedLimit = (int)Settings::getSettings('WORDS_FREQ_ANALYSIS');
        if ($settedLimit !== null) {
            $this->limit = $settedLimit;
        }
    }
    /* Формируем контекстный вектор */
    public function formVectorSpaceModel(ParserFrequency $freqData)
    {
        $this->vsm = [];
        foreach ($freqData->getKeyFreqWords() as $word => $freq) {
            $this->vsm[$word] = $this->calcTermFrequency($freq, $freqData->getCount());
        }
        $this->sortByWeight();
        $this->limit();
        return json_encode($this->vsm);
    }
    /* Считаем TF */
    private function calcTermFrequency($freq, $count)
    {
        return $freq / $count;
    }
    /* Сортируем по возрастанию */
    private function sortByWeight()
    {
        arsort($this->vsm);
    }
    /* Ограничиваем массив */
    private function limit()
    {
        $count = count($this->vsm);
        if ($count <= $this->limit) {
            return $this->vsm;
        } else {
            $this->vsm = array_slice($this->vsm, 0, $this->limit);
        }
    }
}
```

```
<?php
namespace backend\modules\document\services\parser;
use backend\modules\settings\models\Settings;
use phpMorphy;
use phpMorphy_Exception;
final class ParserFrequency extends ParserBase
{    protected $text;
    private $key_words;
    private $key_freq_words;
    private $count;
    private $morphy;
    private $num_max = 5;
    public function __construct(&$text)
    {
        parent::__construct($text);
        $this->key_words = array();
        $this->num_max = Settings::getSettings('WORDS_FREQ_ANALYSIS');
    }

    /**
     * @return mixed
     */
    public function getKeyFreqWords()
    {        return $this->key_freq_words;
    }
    /**
     * @return mixed
     */
    public function getCount()
    {        return $this->count;
    }
    public function parse()
    {
        try {
            $dir = $_SERVER['DOCUMENT_ROOT'] . "/vendor/cijic/phpmorphy/libs/phpmorphy/dicts";
            $lang = 'ru_RU';
            $this->morphy = new phpMorphy($dir, $lang);
        } catch (phpMorphy_Exception $e) {
            die('Error occured while creating phpMorphy instance: ' . $e->getMessage());
        }
        $arr_words = $this->tokenize($this->text);
        $this->count = $count = count($arr_words);
        // получаем части речи, необходимые для парсинга
        $need_words = require __DIR__ . "/config/need_part_speech.php";
        $array_defis = array();
        for ($i = 0; $i < $count; $i++) {
            if (mb_strlen($arr_words[$i]) <= 2) {
                unset($arr_words[$i]);
                continue;
            }
            $arr_words[$i] = mb_strtoupper($arr_words[$i]);
```

```
if ($part = $this->morphy->getPartOfSpeech($arr_words[$i])) {
    // проверка на часть речи
    $isNeedPart = false;
    foreach ($need_words as $nw) {
        if (in_array($nw, $part)) {
            $isNeedPart = true;
        }
    }
    if (!$isNeedPart) {
        unset($arr_words[$i]);
        continue;
    }
    $result = $this->morphy->findWord($arr_words[$i], phpMorphy::IGNORE_PREDICT);
    if ($result === false) {
        // проверка на -
        if (mb_strpos($arr_words[$i], "-")) {
            $tmp = preg_split("@-@", $arr_words[$i]);
            $temp_array = array();
            foreach ($tmp as $s) {
                // добавляем слова по отдельности
                if (mb_strlen($s) >= 2) {
                    if (mb_substr_count($s, 'E') > 0) {
                        $true_change_word = $this->ChangeLetter($s, 'E', 'Ё');
                        if ($true_change_word != NULL) {
                            $true_change_word = $this->morphy->lemmatize($true_change_word)[0];
                            array_push($arr_words, $true_change_word);
                            $temp_array[] = $true_change_word;
                            $count++;
                            continue;
                        }
                    }
                    $s = $this->morphy->lemmatize($s)[0];
                    $count++;
                    array_push($arr_words, $s);
                    $temp_array[] = $s;
                }
            }
            if (!$this->isUnique($array_defis, $temp_array)) {
                $array_defis[] = $temp_array;
            }
            // проверка на идентичность слов
            unset($arr_words[$i]);
            continue;
            // проверка на Ё
        } else if (mb_substr_count($arr_words[$i], 'E') > 0) {
            $true_change_word = $this->ChangeLetter($arr_words[$i], 'E', 'Ё');
            if ($true_change_word != NULL) {
                $arr_words[$i] = $true_change_word;
                continue;
            } else {
                unset($arr_words[$i]);
            }
```

```
                } else {
                    unset($arr_words[$i]);
                }
            }
        }
    }

    $arr_freq = array_count_values($arr_words);
    $maxes = $this->getMaxes($arr_freq, $this->num_max);
    $temp_array = array();
    foreach ($arr_freq as $key => $value) {
        for ($i = 0; $i < $this->num_max; $i++) {
            if ($value == $maxes[$i]) {
                $key_word = $this->morphy->lemmatize($key)[0] ?? $key;
                $this->key_freq_words[$key] = $value;
                $this->key_words[] = $key_word;
            }
        }
    }
    foreach ($array_defis as $arr_d) {
        $str_defis = "";
        foreach ($arr_d as $str) {
            $str_defis .= ($str . "-");
        }
        $str_defis = trim($str_defis, "-");
        if (!empty($str_defis)) {
            $this->key_words[] = $str_defis;
        }
    }
    $stop_words = require(__DIR__ . '/config/stop_words.php');
    foreach ($stop_words as $word) {
        $this->key_words = array_diff($this->key_words, array($word));
    }
    $this->key_words = $this->ArrayUnique($this->key_words);
    $dict_parse_text = "Частотный анализ текста: ";
    for ($i = 0; $i < count($this->key_words); $i++) {
        if ($i != count($this->key_words) - 1) $dict_parse_text .= $this->key_words[$i] . ", ";
        else $dict_parse_text .= $this->key_words[$i] . ".";

//      var_dump($this->count, $this->key_words, $this->key_freq_words);die;
    return array_slice($this->key_words, 0, $this->num_max);
}
private function tokenize(string &$text): array
{
    // убираем все, кроме букв
    $str_freq = preg_replace('@([^А-Яа-яA-Za-z\s\-])@u', '', $text);
    // убираем все лишние пробелы
    $str_freq = preg_replace('@\s{2,}@u', ' ', $str_freq);
    // разбиваем строку по пробелам
    return preg_split("@ @u", $str_freq);
}
```

ПРИЛОЖЕНИЕ Г

ДОКУМЕНТЫ, ПОДТВЕРЖДАЮЩИЕ ВНЕДРЕНИЕ РЕЗУЛЬТАТОВ ДИССЕРТАЦИОННОЙ РАБОТЫ

ДОНЕЦКАЯ НАРОДНАЯ РЕСПУБЛИКА
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
"ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"
283001, г. Донецк, ул. Артема, 58 тел.: (062) 337-17-33, 335-75-62, факс: (062) 304-12-78
эл. почта: donntu.info@mail.ru

28.09.2021 № 06/4-328

На № _____

СПРАВКА

Выдана Андриевской Наталье Климовне в том, что она действительно работала в составе исполнителей следующих кафедральных научно-исследовательских работ:

- Н 17-12 «Разработка научных основ, методов и способов проектирования информационных управляющих систем» (Приказ № 74-15 от 08.02.2012 г.);
- Н 8-18 «Развитие научных основ, методов и средств проектирования информационных систем и технологий» (Приказ № 17-15 от 31.01.2018 г.);
- Н 2020-16 «Методы и средства построения информационных систем с использованием технологий интеллектуального анализа данных» (Приказ № 10-15 от 31.01.2020 г.).

Проректор ГОУ ВПО «ДОННТУ» С.В. Борщевский

Начальник научно-исследовательской части ГОУ ВПО «ДОННТУ» Ю.Н. Лабинский

Соответствует оригиналу
Ученый секретарь Д 01.024.04
В.В. Завадская

ГКНТ ДНР
ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ «НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ И ПРОЕКТНО-КОНСТРУКТОРСКИЙ ИНСТИТУТ ПО АВТОМАТИЗАЦИИ ГОРНЫХ МАШИН «АВТОМАТГОРМАШ ИМЕНИ В.А. АНТИПОВА»
(ГУ «АВТОМАТГОРМАШ ИМ. В.А. АНТИПОВА»)
пр. Ильича, 93, г. Донецк, 283003, тел. (071) 331-09-14
e-mail: avtomatgormash@mail.ru идентификационный код 30556266

02.06.2021 № 42-319

Диссертационный совет Д01.024.04
на базе Донецкого национального технического университета

СПРАВКА

о внедрении результатов исследований диссертационной работы Андриевской Н.К. на тему «Разработка эффективных моделей и алгоритмов обработки информации научно-образовательных организаций», представленную на соискание ученой степени кандидата технических наук по специальности 05.13.01 – Системный анализ, управление и обработка информации (технические науки)

В диссертационной работе Андриевской Н.К. были получены научные результаты, которые имеют практическое значение: предложенные в диссертационной работе модели и алгоритмы могут быть использованы для решения задачи сведения информационных ресурсов, относящихся к одной области знаний в единое информационное пространство, обеспечения возможности открытого и удобного доступа к ним, поддержки их целостности. На основе разработанных в работе моделей, методов и алгоритмов был разработан фреймворк СУИР - система управления информационными ресурсами научно-образовательными организациями.

На базе разработанного фреймворка был создан программный модуль учета результатов научной и исследовательской деятельности, а также учета наукометрических показателей сотрудников под названием «Наука».

Программный модуль «Наука» установлен и успешно прошел тестирование в ГУ «Автоматгормаш им. В.А. Антипова» в условиях отдела систем управления. Тестирование показало, что модуль «Наука» предоставляет следующие возможности:

а) возможность сотрудникам структурных подразделений организации перманентно вести учет результатов своей научной деятельности и в автоматизированном режиме формировать годовые научные отчеты;

Соответствует оригиналу
Ученый секретарь Д. Ф.

б) возможность предоставить руководителям отдельных структурных подразделений и организации в целом автоматизированного средства проведения количественного и тематического анализа научной деятельности каждого из сотрудников, отдельных подразделений и учреждения в целом;

в) возможность обеспечить в удобном для конечного пользователя режиме ввод данных о публикациях путем автоматизированного разбора информации из BibTeX-записей и импорта из порталов eLibrary, google academy. На основе введенных в хранилище системы данных о публикациях для каждого сотрудника автоматически создается отдельная «домашняя» страница, содержащая список результатов его публикационной деятельности;

г) возможность выполнять интеллектуальный подбор материалов и литературы для научной деятельности.

Программный модуль показал работоспособность разработанных в ходе исследований моделей и алгоритмов при решении задач информационного поиска и тематической классификации информационных ресурсов.

Вместе с тем, математическое, алгоритмическое и программное обеспечение тестируемого программного модуля может найти эффективное применение при построении других информационно - аналитических систем, в том числе – систем подготовки принятия решений и систем управления знаниями в организациях научно-технического профиля и в высших учебных заведениях.

Первый заместитель директора
по научной работе,
док. тех. наук, проф. В.Г. Курносов

Соответствует оригиналу
Ученый секретарь Д.
Т.В.

ДОНЕЦКАЯ НАРОДНАЯ РЕСПУБЛИКА
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

283001, г. Донецк, ул. Артема, 58 тел.: (062) 337-17-33, 335-75-62, факс: (062) 304-12-78
эл. почта: donntu.info@mail.ru

10.12.21 № 30-12/714
На №_____

Диссертационный совет Д 01.024.04
при ГОУВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

СПРАВКА

о внедрении результатов исследований диссертационной работы Андриевской Наталии Климовны на тему «Разработка эффективных моделей и алгоритмов обработки информации научно-образовательных организаций», представленную на соискание ученой степени кандидата технических наук по специальности 05.13.01 – Системный анализ, управление и обработка информации (по отраслям) (технические науки) в учебный процесс ГОУВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Результаты диссертационных исследований Андриевской Н.К., а именно: разработанные в ходе выполнения диссертационной работы алгоритмы, модели и методы внедрены в учебный процесс и используются кафедрой «Автоматизированных систем управления» Донецкого национального технического университета в следующих направлениях:

a) при чтении курсов лекций и выполнении лабораторных работ по дисциплинам «Разработка веб-приложений», «Web-технологии и Web-программирование», «Методы и системы искусственного интеллекта» для студентов профиля подготовки 090302 «Информационные системы и технологии в технике и бизнесе», что отражено в учебных программах вышеуказанных дисциплин;
b) при чтении курсов лекций и выполнении лабораторных работ по дисциплинам «Современные распределённые и объектно-ориентированные базы данных», «Методы и технологии проектирования информационных систем и баз данных» для студентов профиля подготовки 090402 «Информационные системы и технологии в технике и бизнесе», что отражено в учебных программах вышеуказанных дисциплин;
c) при чтении курсов лекций и выполнении лабораторных работ по дисциплинам «Web-технологии», «Web-базированные системы» для студентов профиля подготовки 090301 «Автоматизированные системы управления», что отражено в учебных программах вышеуказанных дисциплин;

Соответствует оригиналу
Ученый секретарь Д 01.024.04
Н.В. Завадская

d) при чтении курсов лекций и выполнении лабораторных работ по дисциплине «Интеллектуальный анализ данных» для студентов профиля подготовки 090401 «Автоматизированные системы управления», что отражено в учебных программах вышеуказанных дисциплин;

e) в курсовом проектировании, НИРС, при выполнении бакалаврских и магистерских работ студентами профилей подготовки «Информационные системы и технологии в технике и бизнесе» и «Автоматизированные системы управления» в тематиках, связанных с проектированием систем управления знаниями, использованием онтологий и семантических данных, в задачах информационного поиска, в задачах построения динамических хранилищ данных, основанных на современных СУБД, в том числе и NoSQL.

Проректор по научно-педагогической работе
ГОУВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ», профессор А.Б. Бирюков

Начальник учебного отдела
ГОУВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ», профессор Б.В. Гавриленко

Заведующий кафедрой «Автоматизированные системы управления»
ГОУВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ», профессор А.И. Секирин

Соответствует оригиналу
Ученый секретарь
Т.В. Зава