

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

На правах рукописи



**Кравченко Даниил Юрьевич**

**МОДЕЛИ И АЛГОРИТМЫ ПОИСКА, ПРИОБРЕТЕНИЯ И  
ИСПОЛЬЗОВАНИЯ ЗНАНИЙ В СИСТЕМАХ  
ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ПРИ ОБРАБОТКЕ И  
АНАЛИЗЕ ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ**

Специальность – 1.2.1. Искусственный интеллект и машинное обучение  
(технические науки)

Диссертация на соискание ученой степени  
кандидата технических наук

Научный руководитель –  
Курейчик Владимир Викторович  
доктор технических наук, профессор

Таганрог – 2024

## ОГЛАВЛЕНИЕ

<b>ВВЕДЕНИЕ .....</b>	<b>4</b>
<b>1. ПРОБЛЕМЫ ПОИСКА, ПРИОБРЕТЕНИЯ И ИСПОЛЬЗОВАНИЯ ЗНАНИЙ ПРИ ОБРАБОТКЕ И АНАЛИЗЕ ТЕКСТОВ .....</b>	<b>18</b>
1.1. Аналитический обзор особенностей создания систем искусственного интеллекта и машинного обучения для обработки и анализа текстов.....	18
1.2. Задачи поиска, приобретения и использования знаний при обработке и анализе текстов.....	27
1.3. Поиск знаний на основе применения алгоритмов и инструментов текстового парсинга .....	33
1.4. Приобретение знаний на основе применения больших языковых моделей .....	42
1.5. Алгоритмы и механизмы использования знаний при обработке и анализе текстов.....	47
1.6. Постановка основных задач исследования.....	52
1.7. Выводы по разделу.....	58
<b>2. ПОСТРОЕНИЕ МОДЕЛЕЙ ОНТОЛОГИИ ЗНАНИЙ .....</b>	<b>61</b>
2.1. Верхнеуровневая модель онтологии знаний .....	61
2.2. Нижнеуровневая модель онтологии знаний .....	68
2.3. Алгоритмы группировки предложений при обработке и анализе текстов .....	75
2.4. Выводы по разделу.....	80
<b>3. АЛГОРИТМЫ ПОИСКА, ПРИОБРЕТЕНИЯ И ИСПОЛЬЗОВАНИЯ ЗНАНИЙ ПРИ ОБРАБОТКЕ И АНАЛИЗЕ ТЕКСТОВ .....</b>	<b>82</b>

<b>3.1. Разработка алгоритма поиска знаний в текстах на естественном языке с применением графовых моделей .....</b>	<b>82</b>
<b>3.2. Разработка алгоритма приобретения знаний в текстах на естественном языке с применением множества низкоуровневых правил .....</b>	<b>99</b>
<b>3.3. Разработка модифицированного биоинспирированного алгоритма использования приобретенных знаний в задачах генеративного искусственного интеллекта.....</b>	<b>107</b>
<b>3.4. Выводы по разделу.....</b>	<b>120</b>
<b>4. РАЗРАБОТКА ПРОГРАММНОГО ПРИЛОЖЕНИЯ И ПРОВЕДЕНИЕ ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА .....</b>	<b>123</b>
<b>4.1. Разработка компонентной архитектуры программного приложения</b>	<b>123</b>
<b>4.2. Построение базы данных для хранения онтологических моделей .....</b>	<b>128</b>
<b>4.3. Проведение и результаты вычислительного эксперимента .....</b>	<b>131</b>
<b>4.4. Выводы по разделу.....</b>	<b>144</b>
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>146</b>
<b>СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ ....</b>	<b>150</b>
<b>СПИСОК ЛИТЕРАТУРЫ .....</b>	<b>151</b>
<b>ПРИЛОЖЕНИЕ № 1 .....</b>	<b>165</b>
<b>ПРИЛОЖЕНИЕ № 2 .....</b>	<b>168</b>

## ВВЕДЕНИЕ

**Актуальность темы диссертационного исследования.** Диссертационная работа посвящена *важной научной проблеме искусственного интеллекта* (ИИ, направление в информатике, задачей которого является воссоздание с помощью вычислительных систем и иных искусственных устройств разумных рассуждений и действий), которая заключается в необходимости повышения эффективности процессов *поиска, приобретения и использования знаний в системах искусственного интеллекта при обработке и анализе текстов на естественном языке* [1-3]. Основной научной идеей данного исследования является переход от доминирования систем генеративного искусственного интеллекта, построенных на основе инструментов, представляющих собой «черный ящик», к более надежным интеллектуальным системам, созданным на основе *детерминированных методов и алгоритмов поиска, приобретения и использования знаний, позволяющих минимизировать время отклика системы на пользовательский запрос при условии обеспечения «прозрачности» (англ. transparency) процессов обработки входных данных.*

Проблема «информационного взрыва», причиной возникновения которой стал экспоненциальный рост объемов цифровой информации, привела к ситуации, когда до 95% информационного потока [1] содержит неструктурированные данные. В подобных условиях, *крайне актуальной становится задача создания эффективных интеллектуальных систем поиска и приобретения знаний, в том числе систем искусственного интеллекта для обработки и анализа текстов на естественном языке*. Научным направлением решения этой частной задачи является *Text Mining (TM)* – раскопка знаний в текстовой информации [1-5].

Повышенный интерес исследователей к проблеме поиска и приобретения знаний при обработке и анализе текстовой информации привёл к появлению значительного числа определений основных терминов в данной предметной области. Следствием такой *терминологической рассогласованности* стала

*проблема неопределенности и нечеткости при описании базовых понятий: информация, данные и знания* [6]. В контексте ТМ под *априорной информацией* будем понимать входные текстовые документы, включающие в себя наборы неструктурированных или слабо структурированных данных. *Текстовый документ* является основной лингвистической единицей естественного языка. Построение *модели представления текста* позволяет *закодировать семантические характеристики информации* в виде *вектора*, который в дальнейшем применяется для решения различных прикладных задач [7, 8].

Задача ТМ – поиск и приобретение, а затем использование знаний. *Знания* (англ. *knowledge*) – это сложная иерархия элементов ценной информации с выявленными зависимостями и закономерностями между фактами, событиями, явлениями и процессами [9]. *Ценность информации* определяется на основе расчета вероятностных оценок достижения цели решаемой прикладной задачи до и после получения определенной информации [10]. *Поиск знаний* (англ. *knowledge retrieval*) – процесс возврата информации к структурированной форме. Под *приобретением знаний* (англ. *knowledge acquisition*) для структурированной текстовой информации понимается систематизация полученных знаний [7-10] через построение детализирующих семантику текста *гранул смысла* – триплетов, состоящих из элементов «субъект» - «предикат» - «объект».

Результат последовательных процессов поиска и приобретения знаний определяется достоверностью, релевантностью и интерпретируемостью нового знания. Достижимость данных характеристик знаний напрямую связана с проблемой снижения уровня *информационной неопределенности* – нехватки информации для решения поставленных прикладных задач [9].

В качестве примера прикладной задачи *использования приобретенных знаний*, в данном исследовании, рассматривается значимая проблема информационной поддержки процессов предупреждения и/или ликвидации последствий чрезвычайных ситуаций. В данной задаче исходными данными являются потоки текстовых сообщений (новостной информации, отчетов о

техническом состоянии техногенных объектов, информации о природных явлениях и т.п.), поступающих в центры принятия решений, а на выходе формируются прогностические оценки и/или конкретные инструкции относительно оценки ситуации и предпринимаемых действий определенными специалистами.

Одной из причин, сдерживающих развитие систем искусственного интеллекта для решения задач поиска, приобретения и использования знаний при обработке и анализе текстов, является недостаточно высокий уровень эффективности моделей и алгоритмов, обеспечивающих комплексное решение перечисленных выше задач с учетом особенностей семантики и контекста.

*Таким образом, разработка моделей и алгоритмов поиска, приобретения и использования знаний при обработке и анализе текстов на естественном языке с применением систем искусственного интеллекта и машинного обучения является актуальной проблемой и имеет существенное научное и хозяйственное значение.*

Проведенные в диссертационной работе исследования находятся в русле важнейших научноемких технологий РФ в разделе сквозных технологий (согласно Указу Президента РФ от 18 июня 2024 г. № 529) – пункт 25. «Технологии искусственного интеллекта в отраслях экономики, социальной сферы (включая сферу общественной безопасности) и в органах публичной власти».

**Степень разработанности темы диссертационного исследования.** Проблема поиска, приобретения и использования знаний в системах искусственного интеллекта при обработке и анализе текстов на естественном языке является междисциплинарной и имеет сложную структуру этапов решения. Способы ее решения опираются на теоретические и методологические основы искусственного интеллекта, методов оптимизации, биоинспирированного поиска и принятия решений, семантики, агентного, онтологического и имитационного моделирования.

Задачи поиска и приобретения знаний при обработке и анализе текстовой информации востребованы во многих типах прикладных информационных систем, таких как: *информационно-поисковые; вопросно-ответные; рекомендательные;*

автоматического построения и пополнения баз знаний (БЗ); программы автоматического реферирования и аналитической обработки коллекции документов и т.п. Традиционные методы поиска и приобретения знаний основаны на правилах, тезаурусах, машинном обучении с учителем и требуют наличия достаточных априорных знаний об исследуемой предметной области в виде лингвистических ресурсов, обработанных корпусов текста, словарей и грамматик.

Для создания правил и грамматик отдельных предметных областей разработано достаточное количество инструментов, которые постоянно применяются в информационных системах: CPSL [11]; Jape [12]; LSPL [13]; UIMA Ruta [14]; ABBYY Compreno [15] и др. *Основным недостатком перечисленных инструментов являются значительные трудозатраты на разработку правил и грамматик. Востребованность снижается так же из-за отсутствия гибкости данных правил и грамматик при переходе в другую предметную область.*

Для снижения трудозатрат на создание интеллектуальных систем обработки и анализа текста применяют *модели машинного обучения с учителем* [16-18]. Для обучения подобных моделей применяются размеченные корпуса текстовой информации, использование которых не требует привлечения экспертных оценок, но по-прежнему остается проблема адаптации интеллектуальных систем к новым предметным областям. Также необходимо отметить развитие технологий «открытого извлечения информации» (open information extraction) [19], которые объединяют в себе ряд методов, позволяющих в рамках процессов поиска и приобретения знаний извлекать разнородную информацию из неструктурированных корпусов текстов без учета специфики предметной области, что повышает их универсальность. В большинстве случаев такие методы основаны на методах машинного обучения с частичным привлечением учителя или без учителя. Методы открытого извлечения информации позволяют строить триплеты, благодаря идентификации сущностей и поверхностных связей между ними. Основным недостатком данных методов является высокая вероятность построения

неинформативных триплетов, которые не позволяют получить адекватную семантическую интерпретацию смысла текстовой информации [19].

Перспективным решением данной проблемы стало *развитие и применение методов построения онтологий*, в том числе, для создания модели Мира. *Онтологией* является точная спецификация концептуализации [9]. *Модель мира* – иерархическая структура знаний, которая позволяет построить описание реальности для более эффективной поддержки принятия решений [10]. Работы А.Е. Ермакова, И.А. Минакова, Е.А. Рабчевского, В.М. Курейчика, Т. Gruber, S.Lynn и D.W. Embley, J.Völker, D.Vrandečić, M. Sabou и Y.Sure и других исследователей в области решения задач автоматического построения онтологий вносят значимый вклад в решение обозначенной проблемы [9].

Наиболее известными работами в области решения задач поиска и приобретения знаний являются труды следующих ученых: Э. Тоффлера; Д. Белла; М. Маклюэна; Ё. Масуды; Ю.А. Шрейдера; Р.С. Гиляревского; К. Виига; П. Сенге; И. Нонака; Х. Такеучи; Т. Давенпорта; Л. Прусака; К. Ньюэлла; Д. Смита; В.Л. Иноземцева; Б.З. Мильнера; А.Л. Гапоненко [9].

Необходимо отметить еще несколько базовых работ, непосредственно связанных с проблемами ТМ в области поиска, приобретения и использования знаний в интеллектуальных системах обработки и анализа текстов на естественном языке. В статье [20] описаны технические решения задач поиска и приобретения знаний на основе применения методов лексического анализа. Авторы предлагают алгоритм построения лексических шаблонов, который опирается на анализ частотных характеристик и алгоритм кластеризации c-means. В работе [21] исследованы вопросы классификации текстов и извлечения информации с применением методов Data Mining. В данной статье так же используется частотный метод с оценкой вероятности присутствия словоформ в доменных словарях.

В статье [22] авторы при решении задач поиска, приобретения и использования знаний опираются на лексический и синтаксический анализ и используют системы, основанные на правилах. Работа [23] описывает процессы

обнаружения знаний в БД с применением технологий Data Mining. В работе [24] решается задача интеграции базы знаний с модулем интеллектуального анализа текста. Во многих работах [16-18] задачи обработки и анализа текстов на естественном языке (Natural Language Processing, NLP) решаются на основе применения искусственных нейронных сетей.

Несмотря на всю важность и значимость проведенных исследований, на текущий момент *не существует общепринятого подхода к поиску и приобретению знаний из текстовой информации*, который бы подходил для любой предметной области. Такой *подход* должен обеспечивать точное и полное извлечение терминов и связей между ними, а также должен позволять объединять формализованные структуры из разных информационных источников для более эффективного использования приобретенных знаний.

В результате, *современные методы не справляются с обработкой больших объемов текстовой информации достаточно эффективно*. Получается парадоксальная ситуация, когда пользователи имеют полный доступ к значительному объему ценной информации, но не имеют возможности быстро и качественно извлечь из нее знания.

В данной работе применен комплексный подход к решению проблемы поиска, приобретения и использования знаний в системах искусственного интеллекта при обработке и анализе текстов на естественном языке. *Развитие моделей и алгоритмов решения задач поиска, приобретения и использования знаний, извлеченных из текстовой информации, на основе применения методов искусственного интеллекта и машинного обучения является отличительной особенностью и преимуществом представленного исследования*.

**Объект исследований** – тексты на естественном языке.

**Предмет исследований** – модели и алгоритмы поиска, приобретения и использования знаний в системах искусственного интеллекта при обработке и анализе текстов на естественном языке.

**Целью** диссертационного исследования является повышение эффективности

моделей и алгоритмов поиска, приобретения и использования знаний в системах искусственного интеллекта при обработке и анализе текстов. Под эффективностью понимается минимизация времени отклика системы на запрос пользователя при условии обеспечения «прозрачности» процессов обработки входной текстовой информации.

Для достижения указанной цели в работе поставлены и решены следующие **основные задачи**:

1. Построена верхнеуровневая модель онтологии знаний, созданная на основе оригинальной компонентной архитектуры и применяемая при обработке и анализе текстов на естественном языке, которая позволяет обеспечить необходимую степень детализации анализируемой текстовой информации (пункт 4 паспорта специальности 1.2.1);

2. Построена нижнеуровневая модель онтологии знаний, созданная на основе оригинальной структуры отношений между понятиями и применяемая при обработке и анализе текстов на естественном языке, которая позволяет получить набор смысловых паттернов и проводить оценку их семантической близости (пункт 4 паспорта специальности 1.2.1);

3. Разработан алгоритм поиска знаний в текстах на естественном языке с применением графовых моделей при фильтрации информации на выходе парсера, извлекающий смысловую часть предложения для использования в процессах приобретения знаний (пункт 5 паспорта специальности 1.2.1);

4. Разработан алгоритм приобретения знаний в текстах на естественном языке с применением множества низкоуровневых правил семантического анализа полученных смысловых паттернов, позволяющий определить основные гранулы смысла для процессов использования знаний (пункт 5 паспорта специальности 1.2.1);

5. Разработан модифицированный биоинспирированный алгоритм использования приобретенных знаний в задачах генеративного искусственного интеллекта, основанный на улучшенных механизмах интенсификации поиска

решений и процедурах выхода из локальных оптимумов для уменьшения времени отклика системы искусственного интеллекта и машинного обучения на пользовательский запрос при обработке и анализе текстов на естественном языке (пункт 5 паспорта специальности 1.2.1).

#### **Методология и методы диссертационного исследования.**

Методологической и теоретической основой проведенных исследований послужили положения теории искусственного интеллекта, биоинспирированной оптимизации, алгоритмов, графов, а также методы поиска, приобретения и использования знаний, имитационного, семантического и онтологического моделирования.

Выбор методов исследования предопределен спецификой решаемых задач поиска, приобретения и использования знаний, отличительной особенностью которых является присутствие информационной неопределенности и большой размерности, что исключает возможность применения переборных методов.

**Тематика работы соответствует** п. 4 «Разработка методов, алгоритмов и создание систем искусственного интеллекта и машинного обучения для обработки и анализа текстов на естественном языке, для изображений, речи, биомедицины и других специальных видов данных», п. 5 «Методы и технологии поиска, приобретения и использования знаний и закономерностей, в том числе – эмпирических, в системах искусственного интеллекта. Исследования в области совместного применения методов машинного обучения и классического математического моделирования. Методы и средства использования экспертных знаний» паспорта специальности 1.2.1. Искусственный интеллект и машинное обучение (технические науки).

**Научная новизна.** Научной новизной проведенного исследования являются модели и алгоритмы поиска, приобретения и использования знаний в системах искусственного интеллекта при обработке и анализе текстов на естественном языке, которые направлены на решение научной задачи повышения эффективности и обеспечения «прозрачности» функционирования средств и инструментов

генеративного искусственного интеллекта, что имеет важное значение для развития информатики, а именно:

1. Построена верхнеуровневая модель онтологии знаний, применяемая при обработке и анализе текстов на естественном языке, которая отличается включением в состав ее компонентов множеств понятий с различным уровнем нормализации, что позволяет обеспечить необходимую степень детализации анализируемой текстовой информации (пункт 4 паспорта специальности 1.2.1; страницы 61-68 диссертации);

2. Построена нижнеуровневая модель онтологии знаний, применяемая при обработке и анализе текстов на естественном языке, которая отличается использованием структуры отношений между понятиями, детализирующими семантику текстовой информации, что позволяет получить набор смысловых паттернов, а также проводить оценку их семантической близости (пункт 4 паспорта специальности 1.2.1; страницы 68-75 диссертации);

3. Разработан алгоритм поиска знаний в текстах на естественном языке, отличающийся созданием дополнительного фильтра на выходе парсера с применением графовых моделей, что позволяет извлечь смысловую часть предложения из полученной синтаксической схемы текстовой информации для использования в процессах приобретения знаний (пункт 5 паспорта специальности 1.2.1; страницы 82-99 диссертации);

4. Разработан алгоритм приобретения знаний в текстах на естественном языке, отличающийся применением множества низкоуровневых правил семантического анализа полученных смысловых паттернов, позволяющий определить основные гранулы смысла для процессов использования знаний (пункт 5 паспорта специальности 1.2.1; страницы 99-107 диссертации);

5. Разработан модифицированный биоинспирированный алгоритм использования приобретенных знаний в задачах генеративного искусственного интеллекта, отличающийся улучшенными механизмами интенсификации поиска решений и процедурами выхода из локальных оптимумов, что позволило

уменьшить время отклика системы искусственного интеллекта и машинного обучения на пользовательский запрос при обработке и анализе текстов на естественном языке (пункт 5 паспорта специальности 1.2.1; страницы 107-120 диссертации).

**Теоретическая значимость работы.** Полученная научная новизна развивает аппарат искусственного интеллекта и машинного обучения в области решения важной научной задачи повышения эффективности при условии обеспечения «прозрачности» процессов *поиска, приобретения и использования знаний в системах искусственного интеллекта при обработке и анализе текстов на естественном языке*. Теоретическая значимость диссертации заключается в создании новых моделей онтологий, позволяющих обеспечить необходимую степень детализации спецификаций текстовой информации для повышения эффективности оценки семантической близости элементов знаний при обработке и анализе текстов, а также новых алгоритмов поиска, приобретения и использования знаний, позволяющих извлекать смысловую часть предложения, определять основные гранулы смысла, снизить частоту появления ошибок и минимизировать время отклика систем искусственного интеллекта и машинного обучения на пользовательский запрос.

**Практическая значимость работы.** Практическая ценность работы заключается в создании программного приложения, позволяющего использовать разработанные модели и алгоритмы поиска, приобретения и использования знаний в системах искусственного интеллекта при обработке и анализе текстов на естественном языке для снижения времени отклика системы на запрос пользователя при условии обеспечения «прозрачности» процессов обработки входной текстовой информации. Данное программное приложение позволяет автоматизировать процесс обработки и анализа текстов на естественном языке в условиях большой размерности и информационной неопределенности, а также увеличить объем структурированного знания.

**Положения, выносимые на защиту:**

1. Верхнеуровневая модель онтологии знаний, представленная в виде оригинальной компонентной архитектуры, позволяет обеспечить необходимую степень детализации анализируемой текстовой информации;
2. Нижнеуровневая модель онтологии знаний, представленная в виде оригинальной структуры отношений между понятиями, детализирующими семантику текстовой информации, позволяет получить набор смысловых паттернов и проводить оценку их семантической близости;
3. Алгоритм поиска знаний в текстах на естественном языке с применением графовых моделей для создания дополнительного фильтра на выходе парсера позволяет извлечь смысловую часть предложения из полученной синтаксической схемы текстовой информации для использования в процессах приобретения знаний;
4. Алгоритм приобретения знаний в текстах на естественном языке с применением множества низкоуровневых правил семантического анализа полученных смысловых паттернов позволяет определить основные гранулы смысла для процессов использования знаний;
5. Модифицированный биоинспирированный алгоритм использования приобретенных знаний в задачах генеративного искусственного интеллекта с применением улучшенных механизмов интенсификации поиска решений и процедур выхода из локальных оптимумов позволяет уменьшить время отклика системы искусственного интеллекта и машинного обучения на пользовательский запрос при обработке и анализе текстов на естественном языке.

**Степень достоверности результатов.** Достоверность научных результатов работы подтверждается непротиворечивостью и согласованностью с известными фактами и исследованиями в рассматриваемой области, высокой степенью сходимости теоретических результатов с данными экспериментов и определяется применением теоретических и методологических основ разработок ведущих ученых в области создания теоретических основ поиска, приобретения и

использования знаний в системах искусственного интеллекта при обработке и анализе текстов на естественном языке, а также корректным и обоснованным использованием математического аппарата, экспериментальными исследованиями разработанных моделей и алгоритмов.

**Личный вклад автора.** Все основные результаты диссертации получены непосредственно лично автором.

**Реализация и внедрение результатов работы.** Внедрение теоретических и практических результатов работы проводилось в сотрудничестве с проектной организацией ООО «Газэксперт плюс» (г. Краснодар). Полученные в работе научные результаты позволили повысить качество процедур поиска и интеграции прототипов проектных решений в газотранспортной отрасли. Теоретические и практические результаты, полученные в диссертации, внедрены в учебный процесс Южного федерального университета.

**Апробация результатов.** Основные теоретические положения и практические результаты диссертационной работы докладывались и обсуждались в рамках ряда научных мероприятий, основными из которых являются следующие: 7<sup>th</sup> Computational Methods in Systems and Software (Чехия, октябрь 2023); «5th International Scientific Convention UCIENCIA» (Куба, сентябрь 2023); Международный конгресс по интеллектуальным системам и информационным технологиям (IS&IT) (п. Дивноморское, Краснодарский край, Россия, сентябрь 2022-2023); XII Международная научно-техническая конференция «Технологии разработки информационных систем» (ТРИС) (г. Феодосия, Республика Крым, Россия, сентябрь 2022); 7-я Международная конференция по информационным технологиям в инженерном образовании (г. Москва, Россия, апрель 2024).

Также теоретические и практические результаты исследований вошли в материалы отчетов по грантам РНФ № 22-71-10121 (2022-2024) и № 23-21-00089 (2023-2024), а также РФФИ № 20-01-00148 (2020-2022).

**Публикации.** По теме диссертации опубликовано 19 научных работ, из которых: 5 статей опубликованы в научных изданиях, индексируемых

международными базами данных, перечень которых определен в соответствии с рекомендациями ВАК; **4** статьи – в издании из перечня, утвержденного ВАК, рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук, в т.ч. **1** научная работа, принадлежащая лично автору. Имеется **2** свидетельства о государственной регистрации программ для ЭВМ. В трудах всероссийских и международных конгрессов и конференций опубликовано **7** работ.

**Структура и объем диссертации.** Диссертация состоит из введения, **4** разделов, заключения, списка сокращений и условных обозначений, списка литературы, содержащего **122** наименования, **2** приложений. Основная часть работы содержит **164** страницы, включая **39** рисунков и **10** таблиц.

**Во введении** приведены цель работы, обоснование актуальности темы диссертационной работы, основные научные положения, выносимые на защиту, данные о научной новизне и практической ценности, апробации результатов диссертационной работы, реализации и внедрении, а также приведено содержание разделов диссертации.

**Первый раздел** диссертационной работы посвящен анализу исследуемой предметной области. Проведен аналитический обзор особенностей создания систем искусственного интеллекта и машинного обучения для обработки и анализа текстов на естественном языке. Особое внимание уделено построению компонентной архитектуры подобных систем. Исследованы основные современные модели, алгоритмы, механизмы и инструменты поиска, приобретения и использования знаний в системах искусственного интеллекта при обработке и анализе текстов на естественном языке. Даны постановки основных задач диссертационного исследования.

**Второй раздел** диссертационной работы посвящен описанию построения верхнеуровневой и нижнеуровневой моделей онтологии знаний, применяемых при обработке и анализе текстов на естественном языке, отличающихся

использованием оригинальной компонентной архитектуры и структуры отношений между понятиями, которые позволяют обеспечить необходимую степень детализации анализируемой текстовой информации, а также создание множества смысловых паттернов с возможностью проведения оценки их семантической близости. Помимо этого, в данном разделе представлено описание разработки эвристических алгоритмов предварительной группировки предложений, имеющих схожие смысловые характеристики, а также определения последовательности обработки построенных групп предложений для упрощения последующих процедур анализа текстовой информации.

**Третий раздел** диссертации посвящен описанию разработки алгоритмов поиска и приобретения знаний в текстах на естественном языке, а также использования приобретенных знаний, что позволяет извлекать смысловую часть предложения из полученной синтаксической схемы текстовой информации и строить гранулы смысла при интенсификации и диверсификации поисковых процедур для уменьшения времени отклика системы искусственного интеллекта и машинного обучения на пользовательский запрос при обработке и анализе текстов на естественном языке.

**Четвертый раздел** диссертации посвящен описанию разработки программного приложения и проведения вычислительного эксперимента для сравнения эффективности разработанных моделей и алгоритмов. Приведенные автором результаты показали непротиворечивость разработанных моделей и алгоритмов поиска, приобретения и использования знаний в текстах на естественном языке. Временная сложность предложенных алгоритмов в худшем случае составляет  $O(n^2)$ . В среднем при значительном размере пространства поиска решений (более 1 миллиона вершин) предложенный автором модифицированный алгоритм бактериальной оптимизации снижает время отклика системы искусственного интеллекта на 5 – 7 %.

**В заключении** изложены основные выводы и результаты диссертационной работы, даны рекомендации по перспективам применения результатов.

**В приложениях** приведены свидетельства об официальной регистрации программ для ЭВМ и копии актов о внедрении результатов работы.

# **1. ПРОБЛЕМЫ ПОИСКА, ПРИОБРЕТЕНИЯ И ИСПОЛЬЗОВАНИЯ ЗНАНИЙ ПРИ ОБРАБОТКЕ И АНАЛИЗЕ ТЕКСТОВ**

Первый раздел посвящен анализу исследуемой предметной области. Проведен аналитический обзор особенностей создания систем искусственного интеллекта и машинного обучения для обработки и анализа текстов на естественном языке. Особое внимание уделено построению компонентной архитектуры подобных систем. Исследованы основные современные модели, алгоритмы, механизмы и инструменты поиска, приобретения и использования знаний в системах искусственного интеллекта при обработке и анализе текстов на естественном языке. Даны постановки основных задач диссертационного исследования.

## **1.1. Аналитический обзор особенностей создания систем искусственного интеллекта и машинного обучения для обработки и анализа текстов**

«Прозрачность» (*transparency*) функционирования систем искусственного интеллекта означает, что все действия и решения системы можно проанализировать и понять. Это включает в себя доступность сведений о том, как система обрабатывает информацию, какие алгоритмы используются и какие параметры влияют на принимаемые решения. Команда Стэнфордских ученых предложила «Индекс прозрачности основной модели» (*Foundational Model Transparency Index, FMTI*) и проверила с его помощью 10 крупнейших моделей искусственного интеллекта (ИИ). Оказалось, что даже самые высокие оценки «прозрачности» наиболее популярных систем ИИ не превысили значения равного 54 баллам по стобалльной шкале [25].

Технологии обработки и анализа текстов на естественном языке (Natural Language Processing, NLP) предъявляют ряд требований к разработке интеллектуальных систем, обеспечивая тем самым конфигурируемость данных инструментов. Особенности каждой конфигурации системы искусственного интеллекта и машинного обучения для решения задач поиска, приобретения и использования знаний при обработке и анализе текстов зависят от имеющихся пересечений определенных разделов лингвистики [26], а именно: технологии построения и анализа слов (морфология); технологии построения и анализа предложений (синтаксис); технологии содержательного анализа и смысловой оценки (семантика).

Сложность *морфологического анализа* во многом зависит от используемого языка. Например, в русскоязычных предложениях сложная информация о времени передается на основе аффикса глагола. Подобным образом, в русском языке морфологический анализ будет усложняться необходимостью учета влияния на контекст префиксов и постфиксов. В английском языке подобная проблема отсутствует, например, в описанном ранее случае анализа информации о времени достаточно будет учесть вспомогательный глагол. Задача морфологического анализа при обработке и анализе текстов успешно решается на основе процедуры POS-тегирования, при которой любое слово получает определенные атрибуты, включающие информацию о части речи и наборе морфологических признаков.

Более трудоемкой задачей является проведение *синтаксического анализа* текста, позволяющего построить модель синтаксических отношений между словами в предложении. Основной сложностью при решении данной задачи является неоднозначность естественного языка, которая приводит к ситуации, когда на выходе парсера строятся несколько синтаксических схем [26]. В этом случае, для последующего анализа семантики необходимо выбрать ту синтаксическую схему предложения, которая наиболее корректно отражает смысл контекста исследуемого текста. В противном случае, результаты последующего семантического анализа будут искажены, что отрицательно скажется на качестве

извлекаемых из полученной информации знаний в рамках решения задач их поиска и приобретения. Традиционно эта проблема неоднозначности синтаксических схем предложений решается экспертным путем, благодаря составлению «банков деревьев» (treebanks) с множеством размеченных лингвистами предложений. В данной работе автором разработан *алгоритм поиска знаний* в текстах на естественном языке с применением графовых моделей для извлечения смысловой части предложения из полученной синтаксической схемы текстовой информации на основе создания дополнительного фильтра на выходе парсера.

Успешность процессов решения задач поиска и приобретения знаний напрямую зависит от способности системы искусственного интеллекта и машинного обучения обработки и анализа текстов понимать смысл высказывания, в этом заключается проблема *семантического анализа*. Это крайне сложный этап семантической интерпретации смысла в контексте исследуемой информации. Данный этап плохо формализуется, а его результаты сильно зависят от качества выполнения предыдущих этапов морфологического и синтаксического анализа. В данной диссертации автором разработан *алгоритм приобретения знаний* в текстах на естественном языке, отличающийся применением оригинального множества низкоуровневых правил семантического анализа полученных смысловых паттернов, позволяющий построить основные гранулы смысла для процессов использования знаний.

Отметим, что семантический анализ определяет смысл высказываний на уровне отдельных предложений, в то время, как поиск смысла с учетом контекста происходит на основе интерпретации результатов семантической обработки при реализации этапа *прагматического анализа*. Таким образом, прагматический анализ устанавливает меру ценности информации в тексте.

*Ценность информации* – одно из важнейших ее свойств. Применение ценной информации способствует повышению эффективности процедур поиска и приобретения знаний. В работе [10] дано следующее определение понятию ценности информации: «...в основе определения *ценности* информации лежат

такие ее свойства, как действенность и полипотентность». *Полипотентность информации* – «одна и та же информация может быть использована для решения самых разных задач» [10]. При обработке и анализе текстов полипотентность информации выражается в наличии множества синтаксических схем. *Действенность информации* – «будучи включена в свою информационную систему, информация, соответственно ее семантике, может быть использована для построения того или иного оператора, который, в свою очередь, будучи помещен в подходящее пространство режимов, может совершать те или иные целенаправленные действия», то есть «действенность информации может выявляться лишь в адекватной ей информационной системе» [10]. Таким образом, в данном исследовании действенность информации обеспечивается релевантностью смысла, определенного в предложении на основе проведения семантического анализа.

В [10] в качестве меры ценности информации используется следующая вероятностная оценка:

$$C = \frac{P-p}{1-p}, \quad (1.1)$$

где  $p$  – вероятность достижения цели до получения информации, а  $P$  – после её получения. В данном исследовании целью является повышение эффективности моделей и алгоритмов поиска, приобретения и использования знаний в интеллектуальных системах обработки и анализа текстов, что напрямую связано с повышением качества процедур структурирования, систематизации (классификации) и интеграции ценной информации для последующего построения онтологий знаний.

Достижение данной цели требует применения *моделей машинного вывода*, работа которых предполагает учет множества признаков приобретаемых знаний. В задачах обработки и анализа текста на естественном языке извлечение признаков состоит в [27] переводе предварительно обработанного текста в векторное пространство. Построение семантического вектора в качестве вектора признаков

является необходимым условием для формализации семантики текстовой информации и обеспечения эффективности процессов поиска и приобретения знаний.

Семантический вектор включает в себя набор весовых коэффициентов понятий (концептов). В модели векторного пространства такие весовые коэффициенты  $\omega_{i,j}$  расчитываются на основе метода  $tf.idf$ , в котором они определяются по следующей формуле [28]:

$$\omega_{i,j} = pf_{i,j} \cdot idf_i = pf_{i,j} \cdot \log \frac{N}{n_i}, \quad (1.2)$$

где  $N$  – количество ресурсов;  $n_i$  – количество ресурсов, содержащих понятие  $p_i$ ;  $pf_{i,j}$  – частота появления понятия  $p_i$  в ресурсе  $d_j$ ;  $idf_i$  – обратное значение нормализованной частоты ресурсов [9].

Весовые коэффициенты применяются так же при построении векторных моделей источников текстовой информации. Допустим, что  $d$  – это первый источник текстовой информации, а  $q$  – второй, данные источники включают в себя множества последовательностей понятий (концептов). В модели векторного пространства источники текстовой информации  $d$  и  $q$  представлены в виде векторов весовых коэффициентов, заданных в следующем виде [28]:

$$\vec{d} = (\omega_{1,d}, \omega_{2,d}, \dots, \omega_{n,d}) \text{ и } \vec{q} = (\omega_{1,q}, \omega_{2,q}, \dots, \omega_{n,q}). \quad (1.3)$$

Семантическая близость между источниками  $d$  и  $q$  расчитывается на основе косинусной меры сходства между представленными в (1.3) векторами [9,28]:

$$sim(q, d) = \cos(\vec{q}, \vec{d}) = \frac{\vec{d} \cdot \vec{q}}{|\vec{d}| \cdot |\vec{q}|} = \frac{\sum_{i=1}^n \omega_{i,d} \cdot \omega_{i,q}}{\sqrt{\sum_{i=1}^n \omega_{i,d}^2} \cdot \sqrt{\sum_{i=1}^n \omega_{i,q}^2}}, \quad (1.4)$$

$$sim(q, d) \in [0,1] \forall \omega_{i,d}, \omega_{i,q} \geq 0.$$

Значимый недостаток подобных векторных моделей – отсутствие учёта таких проблем текста на естественном языке, как синонимия, омонимия и полисемия. Также доказано, что проведение прямого исследования особенностей определения семантической близости невозможно [29]. Это привело к созданию значительного числа методов расчета отдельных мер сходства понятий: методы на основе оценки

длины пути между понятиями; методы на основе оценки глубины вершин в таксономии; методы на основе оценки информационного содержания; методы на основе анализа множества родительских понятий [9]. В работе [30] описан подход к применению гибридных мер оценки семантической близости, необходимость появления которых была обусловлена низким качеством решений, полученных при использовании методов расчета отдельных мер сходства понятий (концептов) [9].

Гибридные меры семантической близости строятся на основе процедур свертки отдельных мер сходства понятий. Например, следующая аддитивная свертка применяется для получения гибридных мер семантической близости [30]:

$$S(p_1, p_2) = \sum_{i=1}^n \omega_i \cdot sim^i(p_1, p_2), \quad (1.5)$$

где  $sim^i$  –  $i$ -я мера близости;  $\omega_i$  – вес, определяющий важность данной меры близости (сумма весов равна единице);  $n$  – количество мер близости.

Следующая сигмоидальная функция используется в качестве модификации аддитивной свертки и предназначена для увеличения веса меры сходства и исключения мер семантической близости с малыми весовыми значениями [9, 30]:

$$sig(x) = \frac{1}{1+e^{-\alpha x}}, \quad (1.6)$$

где  $\alpha > 0$ ;

$$S(p_1, p_2) = \sum_{i=1}^n \omega_i sig(sim^i(p_1, p_2)). \quad (1.7)$$

Таким образом, каждому источнику текстовой информации необходимо поставить в соответствие семантический вектор. Расстояние между семантическими векторами источников, характеризующее их близость, определяется через косинусную меру сходства. Расстояние между семантическими векторами источников текстовой информации позволяет провести их классификацию для повышения эффективности последующих процессов поиска, приобретения и использования знаний.

Одним из упрощенных представлений текста на естественном языке является модель «мешок слов» (bag-of-words). Данная модель содержит в своём составе неупорядоченное множество наиболее часто встречающихся в предобработанном

тексте слов. Чаще всего «мешок слов» применяется для решения задачи классификации текстовой информации на основе обучения классификатора по признаку частотности появления слова в тексте. Расчет частотных характеристик появления отдельных слов позволяет получить простой и весьма эффективный алгоритм обработки и анализа текстов, но не дает возможности учитывать синтаксические и контекстные особенности документа [1]. Частично этот недостаток устраняется путем применения  $N$ -грамм (последовательностей  $N$  соседних слов в предложении). Однако, увеличение значения  $N$  до показателей, превышающих  $N = 3$ , приводит к значительным вычислительным ограничениям, поэтому на практике обычно ограничиваются парами или тройками (биграммами или триграммами) соседних слов.

Описанный выше метод «мешок слов» (bag-of-words) имеет значимый недостаток, заключающийся в том, что нельзя полностью полагаться на частоту появления распространенных слов при оценке их ценности, потому что данные слова могут быть, например, местоимениями, или могут быть инвариантны к смыслу контекста решаемой задачи. Указанный недостаток устранен в методе Tf-Idf (term frequency – inverted document frequency), который позволяет нормализовать частоту появления некоторого слова с учетом частоты его применения в других документах исследуемого корпуса текстовой информации. В этом случае, ценность слова определяется более объективно.

Не менее важным этапом обработки и анализа текстовой информации является решение задач категоризации текстов, что позволяет раскрыть семантический контекст документа. В зависимости от полноты априорной информации, характеризующей особенности определения числа и границ классов в исследуемом тексте, для категоризации используются либо задача классификации, либо – кластеризации.

Как важнейший фактор когнитивного восприятия реальности задача классификации состоит в сортировке текстовой информации на основе процедуры распределения документов по семантическим классам. При решении задачи

*классификации* применяются априори построенные паттерны входных и выходных данных для построения дискриминационных моделей. Данный подход опирается на методы машинного обучения с учителем (supervised technique). При этом применяются технологии, построенные на основе деревьев решений, ассоциативной классификации, графовых моделей терминов, байесовских классификаторов, k-ближайших соседей, опорных векторов (SVM), генетических алгоритмов, нечетких корреляций и др. [1].

При решении задачи *кластеризации*, напротив, отсутствует априорная информация о классах. Данный подход опирается на методы машинного обучения без учителя (unsupervised technique). При этом определение числа и границ кластеров происходит одновременно с сортировкой по ним текстовой информации. В процессе кластеризации каждому тексту ставится в соответствие семантический вектор тем, который определяет весовую меру соответствия каждому кластеру. Кластеризация текстов осуществляется на основе технологий, основанных на иерархическом методе, методе секционирования, методе k-средних, кластеризации на основе относительности слов, техники внутриклассового подобия (intra-cluster similarity, IST), алгоритма на основе плотности распределения и др. [1].

Подводя итог представленному в данном пункте аналитическому обзору особенностей создания интеллектуальных систем обработки и анализа текстов, построим компонентную архитектуру части подобной системы искусственного интеллекта и машинного обучения для реализации этапа предобработки (рис. 1.1).

Опишем процессы данного этапа предобработки. На вход системы искусственного интеллекта и машинного обучения обработки и анализа текстов на естественном языке поступает корпус текстовой информации, которую, используя стандартные механизмы предпроцессинга, имеющийся текст процессор по прямым и косвенным признакам разбивает сначала на документы, потом на абзацы, а затем – на предложения (рис. 1.1). Именно предложение является основной единицей обработки текста на естественном языке.

Любое из поступивших на вход фильтр процессора предложение проверяется на предмет соответствия конкретному запросу пользователю. Данная оценка производится, в том числе, по признаку наличия некоторого ключевого слова в отбираемых для последующего анализа предложениях (рис. 1.1).

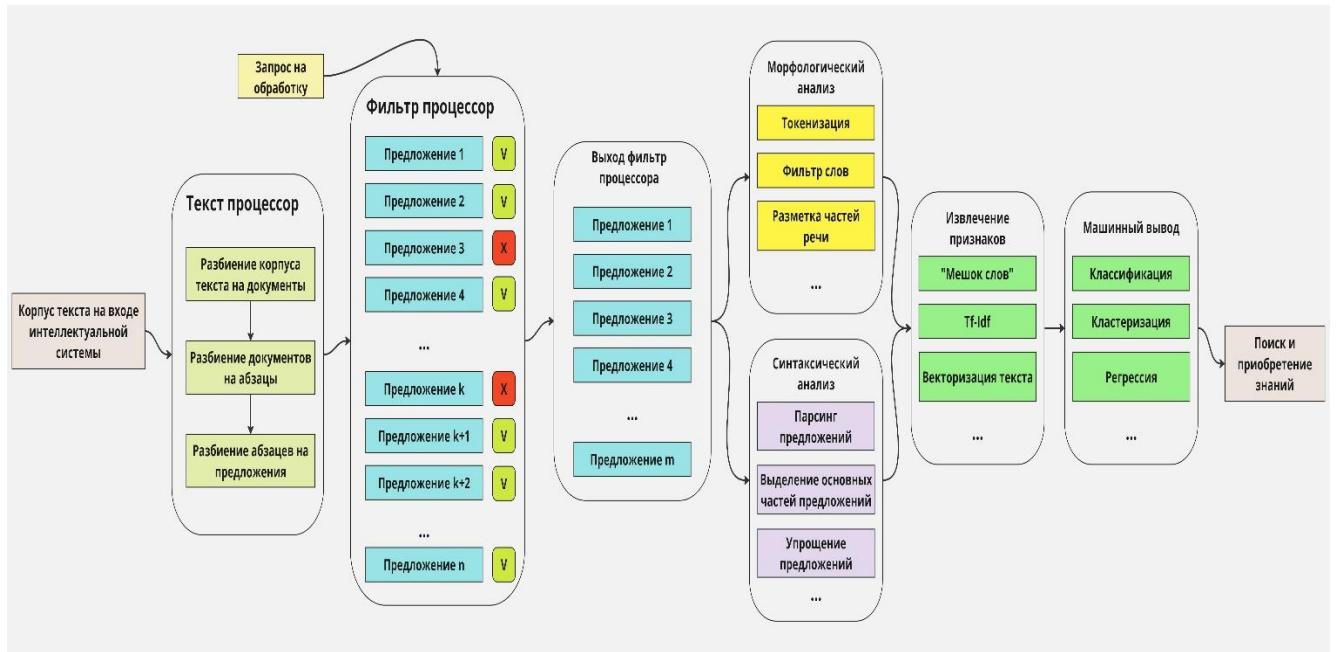


Рисунок 1.1 – Компонентная архитектура части системы искусственного интеллекта и машинного обучения обработки и анализа текстов для этапа предобработки

После проведения морфологического и синтаксического анализов данного предложения, исследователь на выходе парсера получает синтаксическую схему, перегруженную избыточной информацией, инвариантной к смыслу обрабатываемого предложения. Одна из основных задач данного исследования – исключение избыточной информации из синтаксической схемы предложения, получаемой на выходе парсера. В некоторой степени решение данной задачи упрощает применение модулей «извлечения признаков» и «машинного вывода» (рис. 1.1), что позволяет классифицировать или кластеризовать элементы текстовой информации. Однако, в полной мере очистить текст от шума эти методы не позволяют, так как имеют достаточно низкую точность при весьма ограниченных

возможностях для фильтрации избыточной и инвариантной к смыслу информации в предложениях.

**Утверждение 1.** Эффективное решение задачи исключения избыточной информации из текста требует применения низкоуровневых алгоритмов и правил, в том числе на основе использования графовых моделей для создания дополнительного фильтра на выходе парсера, позволяющего извлечь смысловую часть предложения из полученной синтаксической схемы.

Таким образом, в данном пункте первой главы диссертации проведен аналитический обзор особенностей создания систем искусственного интеллекта и машинного обучения для решения задач поиска, приобретения и использования знаний при обработке и анализе текстов на естественном языке. Особое внимание уделено построению компонентной архитектуры подобных систем.

## 1.2. Задачи поиска, приобретения и использования знаний при обработке и анализе текстов

Для развития полученного в предыдущем пункте анализа проблем обработки текстовой информации представим указанные в названии пункта задачи поиска, приобретения и использования знаний в виде процесса извлечения знаний (*knowledge extraction*). Данный процесс в самой общей форме представляет собой последовательность следующих этапов, показанных на рисунке 1.2: поступление корпуса текста на вход системы искусственного интеллекта и машинного обучения; решение задач предобработки текста; применение оператора преобразования полученной после предобработки входных данных текстовой информации.

Формально данное отображение запишем в следующем виде:

$$F_{\text{extraction}}: [Doc_x, x = 1 \dots T] \rightarrow \text{knowledge}, \quad (1.8)$$

где  $F_{\text{extraction}}$  – оператор преобразования полученной после предобработки входных данных текстовой информации;  $Doc_x$  – множество текстовых документов;

*knowledge* – знания, необходимые для решения последующих, иерархически вышестоящих задач, например, для решения задачи информационной поддержки процессов предупреждения и/или ликвидации последствий чрезвычайных ситуаций.

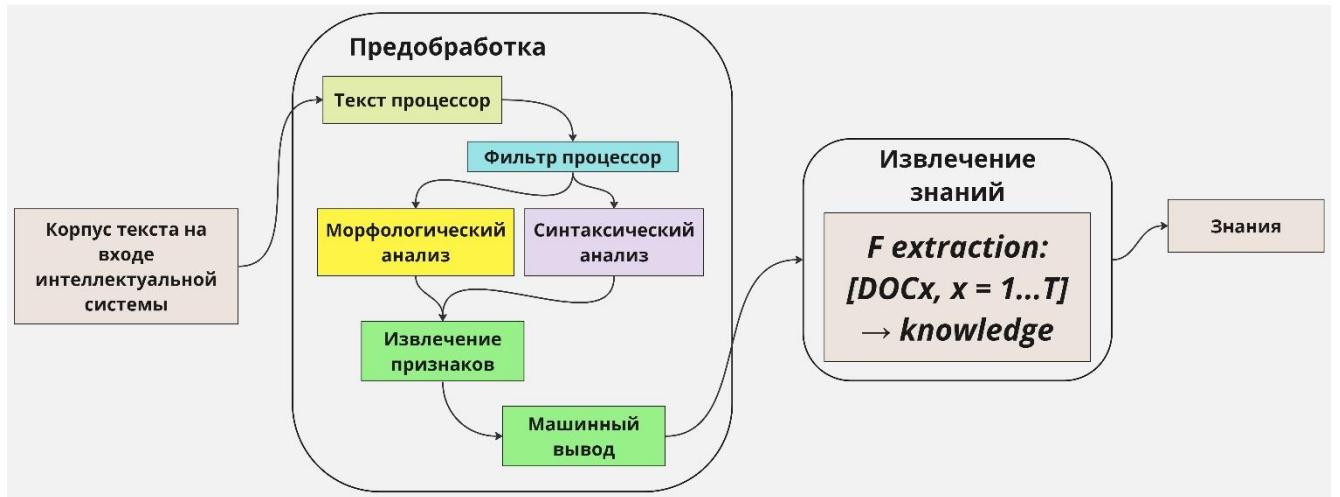


Рисунок 1.2 – Процесс извлечения знаний в общем виде

Одним из способов представления полученных знаний является построение модели, описывающей предметную область в виде множества концептов (понятий) ( $K_i, i = 1...N$ , где  $N$  – количество концептов) с заданной на нём системой связей ( $R_q, q = 1...M$ , где  $M$  – количество связей), являющихся по своей сути несимметричными семантическими отношениями. Таким образом, элементом подобной модели является кортеж длины три:  $\langle k_i, r_q, k_j \rangle$ ,  $i \neq j$ . Система понятий, подходящая для реализации машинного вывода в контексте решения прикладных задач использования приобретенных знаний, подразумевает применение таких категорий семантических отношений, как, например, «эквивалентность», «класс – подкласс», «часть – целое» и другие [2, 3].

Отметим, что тексты на естественном языке в явном или неявном виде содержат в себе компоненты смысла, имеющие в своей основе структуру описанных выше кортежей, имитирующих знания человека и позволяющих строить информационные модели с практически неограниченными возможностями масштабирования. Именно процессы построения и обработки подобных

информационных моделей (онтологий, графов знаний, моделей мира) лежат в основе методов поиска, приобретения и использования знаний при обработке и анализе текстов.

Описанный выше способ представления данных делит процессы поиска и приобретения знаний на этапы *извлечения (понятий) концептов* (concept extraction) и *извлечения отношений* между ними (relation extraction). В интеллектуальных информационных системах происходит комбинирование данных этапов. Способ комбинирования определяет тип конкретной интеллектуальной информационной системы (ИИС). Так, например, *открытые* ИИС извлекают все возможные отношения между концептами (понятиями) в корпусе текста, а *закрытые* – только отношения из ранее заданной выборки концептов.

Для реализации этапа *извлечения концептов (понятий)* из текста на естественном языке применяются методы на основе: правил, статистики, внешних источников, машинного обучения, а также гибридные методы (рис. 1.3)

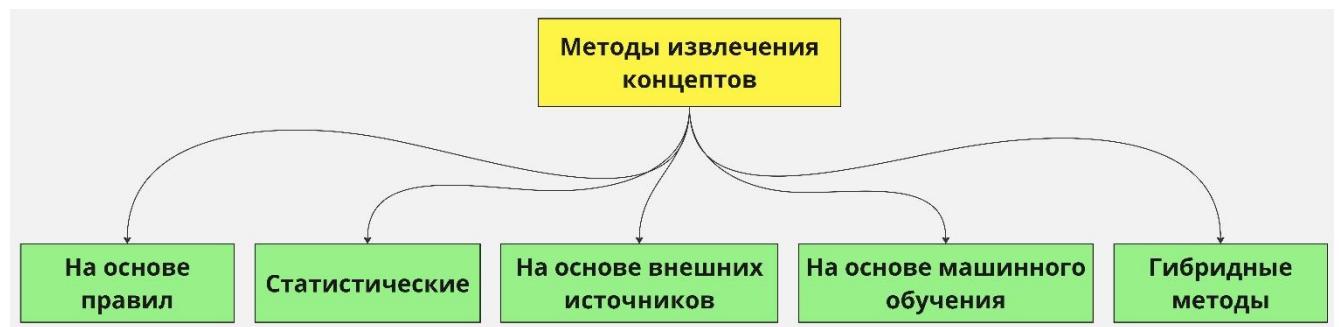


Рисунок 1.3 – Классификация методов извлечения концептов (понятий)

Приведенные в данной классификации (рис. 1.3) порядок методов соответствует хронологии развития этих инструментов с постепенным переходом от методов, построенных на основе низкоуровневых правил, к методам, основанным на процедурах машинного обучения (machine learning), которые применяются в условиях обучения языковых моделей на больших данных. По мнению автора, перспективы использования методов извлечения правил, построенных на основе правил, пока не исчерпаны.

**Утверждение 2.** Комбинированное использование методов извлечения концептов (понятий), основанных на низкоуровневых правилах, совместно с методами машинного обучения, повышает качество решения задач поиска, приобретения и использования знаний при обработке и анализе текстов на естественном языке.

Одной из основных проблем, препятствующей созданию эффективных методов извлечения концептов, является проблема формализации понятия «концепт». Сложность данной проблемы обусловлена тем, что одни и те же элементы информации, извлечённые из текста, в одних условиях – могут считаться «концептами», а в других – нет. Например, понятие – «событие», если оно представляет интерес для пользователя, может быть трактовано как «концепт». Но, при этом, с событием более логично связывать временной интервал, который будет являться уточнением иного «концепта».

Таким образом, следует понимать, что «концепт» может быть не только словом, но и множеством слов. В этом случае для «концепта» должен быть задан некоторый шаблон (паттерн), с которым будут сравниваться результаты обработки текста на естественном языке. Структура данного шаблона должна быть прописана в контексте исследуемой текстовой информации и учтена в методах извлечения концептов. Тогда концепт будет задаваться шаблоном, а наличие концепта в обрабатываемой информации будет определяться как соответствие текста данному шаблону [31].

В общем случае в состав *концепта* может входить либо одно слово, либо – отношение, построенное на множестве слов. Такое общее определение термина «концепт» позволяет объединить в себе все возможные структуры (ключевые слова, словосочетания, лексико-грамматические и лексико-семантические шаблоны), но возникает проблема с применением такого обобщения на практике, связанная с необходимостью формализации множества типов таких отношений между словами в структуре концепта (понятия). Среди наиболее подходящих концептуальных моделей для такой формализации автор выделяет *графы* и

*решётки понятий*. Корректность выбора этих моделей подтверждается данным выше общим определением концепта (понятия) как отношения на множестве слов.

Технологии Text Mining развиваются в двух следующих направлениях:

1) применение методов, использующих статистику частоты встречаемости слов в тексте;

2) применение методов, основанных на построении семантических моделей.

Исследования, представленные в данной диссертации, в большей мере опираются на применение второго – семантического направления.

Преимуществом концептуальных моделей является возможность построения как бинарных, так и  $n$ -арных отношений на множестве понятий (концептов). Представим *графовую модель* с вершинами понятиями и ребрами – связями между понятиями. С точки зрения анализа формальных понятий (АФП) концептуальной моделью такого графа станет *решётка понятий* [32, 33].

Допустим, что с множеством концептов  $K$  связано множество их атрибутов  $H$ . Оба эти множества частично упорядочены системами отношений  $Z$  и  $W$  соответственно, тогда:  $K = (K, Z)$ ;  $H = (H, W)$ . Данные множества необходимы для определения формального контекста  $U = (K, H, V)$ , где  $V \subseteq K \times H$  – отношение между концептами и их атрибутами, представленное рядом кортежей  $\langle k, h \rangle \in V$ .

Отношения между объектами и связанными с ними атрибутами формально задаются следующими отображениями:  $A': A \rightarrow B; B': B \rightarrow A$ . Для данных отображений характерны свойства полноты:  $A' := \{y \in Y | \forall x \in A \langle x, y \rangle \in V\}$ ;  $B' := \{x \in X | \forall y \in B \langle x, y \rangle \in V\}$ . При этом *формальным понятием* контекста  $U$  является пара подмножеств  $(A, B)$ , таких, что  $A' = B, B' = A$  [32]. Условия полноты требуют, чтобы понятия  $(A, B)$  в матрице контекста были заданы максимальными по вложению подматрицами со всеми ненулевыми элементами. В силу композиции отображений  $A'' = A, B'' = B$  множества  $A$  и  $B$  замкнуты. Множество  $A$  образует *объем* формального понятия  $(A, B)$ , а множество  $B$  – его *содержание*. Отношения частичного порядка  $Z, W$  на множествах  $K$  и  $H$  индуцирует отношение частичного порядка  $\leq$  на множестве понятий [32].

Если для понятий  $(A_1, B_1)$  и  $(A_2, B_2)$  выполняется, что  $A_1 \subseteq A_2$  и  $B_2 \subseteq B_1$ , тогда  $(A_1, B_1) \leq (A_2, B_2)$ . Таким образом, понятие  $(A_1, B_1)$  менее общее, чем понятие  $(A_2, B_2)$  [32]. Для представления формального контекста необходимо построить матрицу инцидентности отношения  $V$ . В данной матрице ненулевые элементы обозначают факт принадлежности некоторого атрибута  $h \in H$  концепту  $k \in K$ . Частично упорядоченное по вложению объемов множество формальных понятий контекста  $U$ , согласно основной теореме анализа формальных понятий, образует *решетку понятий* [32, 33], узлами которой являются концепты.

Решетка понятий для определенного контекста является иерархической моделью представления и извлечения знаний (понятий) из текста на естественном языке. Граф решетки понятий не является деревом, что позволяет представлять знания (понятия), имеющие меньшую и большую общность, а также характеризующиеся меньшими и большими объемом и содержанием. В качестве модели формального контекста применяют двудольные графы [32], моделирующие семантические отношения каждого предложения в виде концептов и концептуальных отношений.

Для описания задачи извлечения отношений между концептами рассмотрим множество анализируемых документов (текстов)  $D = \{Doc_x\}_1^T$ , которое  $\forall Doc_x \in D$  представляет собой последовательность слов  $\{word_y\}_1^{L_x}$ , где  $L_x$  – длина документа  $x$ . Необходимо  $\forall Doc_x \in D$  найти множество  $\langle k_i, r_q, k_j \rangle$ ,  $i \neq j$ , называемых отношениями, где  $r_q \in R$ ,  $R$  – множество отношений.

Для оценки качества работы алгоритмов извлечения отношений применяют показатели точности (precision), полноты (recall) и *F1*-меры [34]:

$$Precision = \frac{|R \cap R'|}{|R|},$$

$$Recall = \frac{|R \cap R'|}{|R'|},$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall},$$

где  $R$  – множество извлеченных из документа отношений между концептами,  $R'$  – множество существующих отношений между концептами.

Отсюда следует, что процессы извлечения знаний из текста могут быть сведены к решению задач *извлечения концептов* (concept extraction) и *извлечения отношений* между ними (relation extraction). Существует множество методов решения данных задач. Все эти методы построены на разных принципах (методы на основе правил, статистические методы, методы на основе внешних источников, методы на основе машинного обучения, гибридные методы). Присутствие информационной неопределенности и зашумленности входных данных усложняет выбор методов решения задач извлечения знаний.

Проведем аналитический обзор основных современных моделей, алгоритмов, механизмов и инструментов поиска, приобретения и использования знаний в системах искусственного интеллекта при обработке и анализе текстов на естественном языке.

### **1.3. Поиск знаний на основе применения алгоритмов и инструментов текстового парсинга**

В данном пункте проанализирована проблема отсутствия «прозрачности» в процессах поиска знаний при обработке и анализе текстов на естественном языке, связанная с использованием нейросетей для анализа зашумленных структур текстовой информации, полученных после работы текстового парсера [35].

В динамичном мире обработки естественного языка (Natural Language Processing, NLP) синтаксический анализ (парсинг) играет ключевую роль в раскрытии сложностей естественного языка. Как основа к пониманию структуры и смысла предложений, парсеры служат незаменимыми инструментами в различных задачах NLP, позволяя машинам воспринимать и обрабатывать естественный язык с более высокой точностью и эффективностью. От анализа настроений до машинного перевода, а также для систем вопросов и ответов, парсеры играют

ключевую роль в преобразовании предложений в синтаксические структуры, что в свою очередь облегчает более точную и контекстно значимую обработку языка. Разбивая предложения на понятные единицы, парсеры создают фундамент для машинного понимания семантики и взаимосвязей между словами, делая возможным достижение более сложных и тонких результатов в различных приложениях [36], поэтому *создание эффективных текстовых парсеров является актуальной научной проблемой*.

Основные подходы к синтаксическому парсингу включают синтаксический анализ составляющих (constituency parsing) и синтаксический анализ зависимостей (dependency parsing). Анализ составляющих и зависимостей – это взаимодополняющие подходы, которые направлены на анализ синтаксической структуры предложений. Эти методы анализа предоставляют ценные сведения о грамматической структуре и семантических отношениях в предложении [35, 36].

Анализ составляющих сосредоточен на определении конституентов, которые являются группами слов, выполняющими единую функцию в предложении. Эти конституенты могут быть фразами, такими как именные фразы (NP) или глагольные фразы (VP), или даже более крупными единицами, такими как предложения. Анализ составляющих представляет иерархическую схему предложения с использованием древовидной структуры, называемой деревом разбора или синтаксическим деревом. С другой стороны, анализ зависимостей сосредоточен на отношениях между отдельными словами в предложении. Он представляет эти отношения в виде направленных связей или зависимостей, где каждое слово связано со своим синтаксическим корневым или управляющим словом. Анализ зависимостей обеспечивает более линейное представление структуры предложения, акцентируя внимание на зависимостях между словами, а не на иерархической организации конституентов [37].

Одним из известных алгоритмов в конституентном анализе являются алгоритм CYK (Cocke-Younger-Kasami). Этот классический алгоритм разбора, основанный на динамическом программировании, эффективно строит дерево

разбора, разбивая предложения на более мелкие конституенты с использованием контекстно-свободной грамматики. Так же известен алгоритм Эрли, который способен обрабатывать неоднозначные грамматики и разбирать предложения с использованием предсказывающего сверху-вниз и снизу-вверх подхода, что приводит к более надежному процессу разбора [35].

С другой стороны, популярным алгоритмом анализа зависимостей является алгоритм Arc-Eager. Это алгоритм разбора на основе переходов (Transition-based), который предсказывает последовательность действий для построения дерева зависимостей, эффективно отображая отношения между корневыми и зависимыми словами. Другим подходом на основе переходов является алгоритм Arc-Standard, который строит деревья зависимостей, сводя предложение к однокоренному дереву с помощью серии действий [38].

Исследования в области обработки естественного языка открыли потенциал глубокого обучения для повышения эффективности синтаксического анализа зависимостей [38, 39]. Используя архитектуры нейронных сетей и обширные объемы размеченных данных, парсеры на основе глубокого обучения достигли значительного улучшения точности и эффективности. Существуют техники, включая парсинг на основе переходов с использованием нейронных сетей [38], которые улучшают традиционные парсеры путем интеграции нейронных сетей для более точного улавливания контекстуальных особенностей и зависимостей.

В [39] используют графовые нейронные сети (graph-based) для выполнения синтаксического анализа зависимостей, что позволяет более эффективно обрабатывать неявные зависимости и синтаксические структуры.

Было установлено, что глубоко контекстуализированные представления слов оказывают еще больший положительный эффект на transition-based парсинг, чем graph-based парсинг [40]. Информация о синтаксической структуре, содержащаяся в глубоко контекстуализированных представлениях слов, помогает смягчить главный недостаток transition-based алгоритмов в виде ошибок при обработке длинных предложений. Модели глубоко контекстуализированных представлений

слов BERT [41] и ELMo [42] позволили значительно улучшить результаты для обоих алгоритмов парсинга, причем для transition-based парсинга улучшение оказалось более значимым.

Transition-based и graph-based подходы обладают взаимодополняющими преимуществами и недостатками. Несмотря на то, что transition-based и graph-based парсеры показывают примерно равную точность, они совершают ошибки разного рода. Transition-based парсеры чаще ошибаются в длинных предложениях, зависимостях около корня дерева, зависимостях с глаголом и союзами, а также в определении корневого слова. Это связано с жадным алгоритмом, где ошибка в одной зависимости может привести к каскадным ошибкам в других зависимостях. С другой стороны, graph-based парсеры чаще допускают ошибки в коротких предложениях, зависимостях с существительными и местоимениями, а также в зависимостях вблизи листьев дерева. Это связано с ограниченным набором признаков. Таким образом, оба подхода имеют свои преимущества и недостатки, которые дополняют друг друга.

С целью выяснения различий между этими двумя типами лингвистических анализаторов и изучения их влияния на производительность и скорость задач анализа текста сравним известные лингвистические анализаторы, каждый из которых принадлежит к различным подходам: SpaCy [43], основанный на переходах (анализ зависимостей), и Stanza [44] (анализ зависимостей) и AllenNLP [45] (анализ составляющих), основанные на графах. Оценим скорость каждого парсера при анализе длинных текстов и коротких предложений, а также влияние этих парсеров на задачу извлечения ключевых фраз.

Основными алгоритмами парсинга зависимостей являются Transition-Based Dependency Parsing и Graph-Based Dependency Parsing. Transition-Based Dependency Parsing основан на механизме shift-reduce. Алгоритм был предложен Ямада и Матсумото [46] и Нивре [47] на основе history-based parsing [48] и data-driven shift-reduce parsing [49]. Идея алгоритма заключается в сведении задачи парсинга к пошаговому прогнозированию наличия или отсутствия зависимости между двумя

словами в предложении и направления выявленной зависимости. Парсер состоит из буфера входных токенов (слов), стека, предиктора и набора определенных зависимостей. В первоначальной конфигурации буфер входных токенов состоит из слов предложения в порядке, в котором они расположены в предложении, набор определенных зависимостей пуст, стек состоит из одного служебного элемента ROOT. Парсер обрабатывает предложение слева направо, последовательно сдвигая элементы из буфера в стек. На каждом шаге предиктор отправляет один токен из буфера в стек, анализирует два верхних элемента в стеке и принимает одно из следующих решений:

- назначить первое слово в стеке главным по отношению ко второму (левая дуга) и удалить второе слово из стека;
- если для верхнего слова в стеке уже назначены все зависимые слова, тогда назначить второе слово в стеке главным по отношению к первому слову (правая дуга) и удалить первое слово из стека;
- отложить обработку текущего слова, сдвинув его вниз по стеку.

Дополнительное условие для второго оператора (правая дуга) необходимо для того, чтобы слово не было извлечено из стека до того, как ему будут присвоены все его зависимые элементы.

Окончательное синтаксическое дерево будет составлено, когда буфер окажется пуст, а в стеке останется только служебный символ ROOT. Преимуществом алгоритма в сравнении с динамическими алгоритмами парсинга является его линейная сложность в зависимости от длины предложения. Данный алгоритм представляет собой жадный алгоритм, так как предиктор делает один выбор на каждом шаге, данный выбор считается квазиоптимальным, повторно элементы не обрабатываются, и другие варианты построения зависимостей не рассматриваются. Некорректный выбор на одном шаге ведет к построению ошибочного дерева, без возможности вернуться назад и исправить ошибку. Кроме того, алгоритм возвращает только один вариант синтаксического дерева, в то время

как, ввиду проблемы двусмысленности, возможно наличие более одного варианта корректных синтаксических деревьев.

Предиктор парсера может быть основан на классификаторе на основе признаков (classic feature-based algorithm) или на нейронном классификаторе. Алгоритм на основе признаков полагается на такие признаки, как форма слова, лемма, часть речи главного и зависимого слова; форма слова, лемма, часть речи для слов перед или между главным и зависимым словом; также учитывается состояние буфера входных токенов, стека и набора определенных зависимостей.

Признаки определяются вручную или с помощью обучения классификатора. Выбор признаков вручную несет в себе несколько проблем. Во-первых, слишком большое количество признаков может привести к переобучению и замедлению модели. Во-вторых, для корректного выбора признаков необходимы глубокие знания в области лингвистики [50].

Что касается классификатора, в последние годы произошел переход к нейронным классификаторам, который привел к значительному повышению точности предиктора [40]. Стандартный алгоритм состоит в следующем: предложение проходит через энкодер, затем векторные представления двух первых слов из стека и первого слова из буфера конкатенируются и подаются в нейронную сеть прямого распространения. Кроме того, был разработан нетерпеливый (arc eager) transition-based алгоритм, который использует парсер SpaCy [38]. Главным отличием arc eager алгоритма от стандартного заключается в применении операторов к первому слову в стеке и первому слову в буфере. Это позволяет избавиться от условия предварительного назначения всех зависимых слов. Такое изменение дает возможность быстрее назначать зависимости слева направо (правая дуга) и ускоряет работу парсера. Для корректной работы парсера добавлен оператор «сокращение/reduce», необходимый для завершения процедуры парсинга в случае, если входной буфер оказался пуст.

Для повышения точности transition-based парсинга он может быть дополнен алгоритмом лучевого поиска (beam-search algorithm) [42]. Вместо того, чтобы

выбирать единственный оператор на каждом шаге, выбираются все операторы на каждом шаге, а затем все полученные частичные деревья оцениваются классификатором. Оценка каждого следующего дерева в одной последовательности рассчитывается как сумма оценки предшествующего дерева и оценки примененного к нему оператора:

$$Tscore(i) = TScore(i-1) + OpScore(i-1),$$

где  $TScore$  – оценка дерева,  $OpScore$  – оценка оператора,  $i$  – порядковый номер дерева.

Количество деревьев ограничивается предустановленным лимитом – шириной луча. Когда ширина луча достигает лимита, новые деревья добавляются вместо худших, если оценка нового дерева выше оценки худшего дерева в луче. Процесс парсинга завершается, когда луч содержит только полные деревья входного предложения. Синтаксический анализатор выбирает из луча дерево с наивысшей оценкой и возвращает его в качестве окончательного вывода. Таким образом, парсеру не приходится принимать окончательные решения слишком рано, есть возможность вернуться на начальные этапы построения дерева и исправить ошибку, что значительно повышает точность парсера.

*Алгоритм парсинга зависимостей на основе графа (graph-based parsing)* разработан Макдональдом [51] на основе работы Эйснера [52]. В отличие от transition-based парсинга, полагающегося на жадные локальные решения, graph-based парсинг основан на оценке полного синтаксического дерева. Идея graph-based алгоритма заключается в представлении пространства возможных синтаксических деревьев в виде ориентированного графа (вершинами которого являются слова, а направленными ребрами – зависимости) и поиске в этом графе дерева с наилучшей оценкой. Общая оценка каждого дерева вычисляется как сумма весов отдельных зависимостей, из которых оно состоит. В результате, graph-based алгоритм рассматривает и оценивает все возможные зависимости в предложении, что

является предпосылкой для более высокой точности по сравнению с transition-based парсингом.

Таким образом, нахождение наилучшего дерева зависимостей сводится к нахождению максимального оставного дерева, которое представляет собой подграф с максимальной суммой весов ребер, содержащий все вершины исходного графа и корневую вершину ROOT.

Так же, как и при transition-based парсинге, оценка зависимостей и полного дерева, как суммы оценок зависимостей, может производиться классификатором на основе признаков (classic feature-based algorithm) или на нейронном классификаторе. В feature-based алгоритме оценка зависимости (ребра графа) вычисляется как взвешенная сумма признаков:

$$Score(S, e) = \sum_{i=1}^N f_i(S, e), \quad (1.9)$$

где  $S$  – предложение,  $e$  – зависимость (ребро),  $f$  – признак,  $N$  – количество признаков.

Главной задачей является выявление релевантных признаков и их комбинаций. Могут использоваться следующие признаки: форма слова, лемма, часть речи главного и зависимого слова; расстояние между главным и зависимым словом, направление связи (слева направо или справа налево), векторные представления слов и т.д. По сравнению с transition-based парсингом, для graph-based парсинга возможен ограниченный набор признаков, так как алгоритм рассматривает признаки только самой пары слов (рассматриваемых как потенциальная зависимость) и игнорирует признаки других слов в предложении, упуская тем самым глобальный контекст.

Нейронные классификаторы показывают более высокую точность. Предложение подается в энкодер, где для каждого токена строится глубокое контекстуализированное векторное представление. Ряд исследователей установили, что такие представления содержат информацию о синтаксической структуре предложения [53, 54]. Затем полученные представления передаются в нейронную сеть, которая присваивает оценки каждой зависимости.

Дозат и Мэннинг предложили архитектуру нейронной сети, в которой использовали биафинное внимание (biaffine attention) вместо стандартного билинейного внимания [55]. В такой сети на вход подаются последовательность токенов, конкатенированных с тегами их частей речи. В этом случае операция конкатенации представляется следующим выражением:

$$x_i = v_i(\text{word}) \oplus v_i(\text{tag}), \quad (1.10)$$

где  $x_i$  – конкатенированный вектор,  $v_i(\text{word})$  – представление токена,  $v_i(\text{tag})$  – представление тега части речи токена.

Затем они обрабатываются энкодером в виде многослойной двунаправленной сети долгой краткосрочной памяти (LSTM):

$$r_i = BiLSTM(r_0, (x_1, \dots, x_n))_i, \quad (1.11)$$

где  $r_i$  – конечное состояние,  $r_0$  – первоначальное состояние,  $(x_1, \dots, x_n)$  – конкатенированные вектора. Такой энкодер отражает глобальный контекст в локальных представлениях слов, что расширяет набор признаков за пределы непосредственно главного и зависимого слова, смягчая тем самым главный недостаток graph-based парсинга.

Проведенный в данном пункте анализ особенностей применения текстовых парсеров обосновывает представленное ранее *Утверждение 1* и указывает на необходимость разработки алгоритма поиска знаний в текстах на естественном языке, отличающегося применением графовых моделей для создания дополнительного фильтра на выходе парсера, что позволит извлечь смысловую часть предложения из полученной синтаксической схемы текстовой информации для использования в процессах приобретения знаний. Анализ проблем приобретения знаний на основе применения больших языковых моделей проведен в следующем пункте данного раздела. Основной акцент сделан на оценку точности обработки текстовой информации в системах генеративного искусственного интеллекта, а также соответствия используемых моделей информационного пространства сложности решаемых задач приобретения знаний.

## 1.4. Приобретение знаний на основе применения больших языковых моделей

Анализируя проблемы приобретения знаний при обработке текстов, необходимо рассмотреть последние достижения в области применения больших языковых моделей (Large Language Models, LLM), которые сейчас успешно используются в системах генеративного искусственного интеллекта, таких как, например, ChatGPT. Эти модели представляют собой тщательно сконструированные комбинации из исключительно простых алгоритмов, огромных объемов данных и грандиозных вычислительных мощностей. LLM учится, играя сами с собой в игру «угадай следующее слово». В каждом раунде такой игры модель смотрит на часть предложения и пытается угадать, или предсказать, следующее слово. Если слово угадано – модель обновляет параметры для того, чтобы подкрепить свою уверенность; в противном случае, модель учится на своей ошибке для того, чтобы в следующий раз её догадка была бы точнее.

Проблема заключается в том, что модель, обученная по принципу «угадай следующее слово», не может усвоить «смысл» языка. Впечатляющие результаты таких моделей – это всего лишь результат заучивания «поверхностной статистики», то есть – длинного списка корреляций, который не отражает причинно-следственную модель процесса, генерирующего некую последовательность данных. Без сведений о том, так ли это на самом деле, сложно подстраивать модель под человеческие ценности и убирать из неё освоенные ей ложные корреляции. Эта проблема представляет практический интерес, поскольку, если полагаться на ложные корреляции, можно столкнуться с проблемами при работе с данными, отличающимися от тех, на которых обучалась модель.

Сегодня ChatGPT захватил умы многих специалистов, работающих в области анализа знаний. Генеративный искусственный интеллект показывает высокие результаты применения во многих областях техники и экономики. Многие профессионалы считают подобные инструменты способными решить проблемы бизнес аналитики в условиях больших данных и информационной

неопределенности. Насколько достоверным является их мнение? Что необходимо для понимания контента?

Разработчики ChatGPT заявляют, что предложенный ими в версии 4.0 «механизм внимания» позволил системе достигнуть еще более высокого уровня понимания контента. Чтобы подтвердить или усомниться в этом необходимо дать определение термину «понимание». На рисунке 1.4 представлена модель иерархии общеизвестных информационных категорий «данные», «информация» и «знания». Данная модель была построена одним из классиков исследования операций – Расселом Акоффом [9, 56], который предложил расширить ее категориями «понимание» и «мудрость».

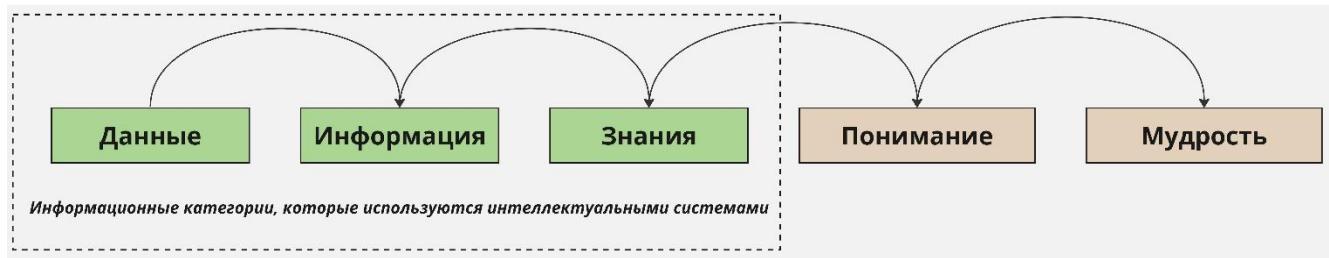


Рисунок 1.4 – Модель иерархии информационных категорий [9]

Объясняя различия между представленными информационными категориями, Акофф считал [56], что данные представляются в виде символов. Например, адрес здания описывает его положение в городе с помощью букв и цифр. Информация состоит из данных, которые подверглись обработке, повысившей их ценность. Информацией являются описания, ответы на вопросы, начинающиеся словами «кто», «что», «когда», «сколько». Знания – это сведения об отношениях между компонентами информации, отвечающие на вопрос «как». Одно дело – знать, в каком городе находится объект (информация), и совсем другое – знать, как туда доехать (знание). Ещё более важно знать, почему человек хочет туда ехать. Объяснение содержится в ответе на вопрос, начинающийся словом «почему», оно-то и даёт понимание. Данные, информация, знание и понимание дают нам возможность эффективно расходовать ресурсы для получения промежуточных результатов в продвижении к цели. Мудрость связана с

эффективностью достижения самой конечной цели, то есть с тем, стремились ли мы сделать правильный выбор из имеющихся альтернатив. Мудрость выражается в оценивании полученного конечного результата.

В целом, представленная выше модель понимается как описание эволюционного процесса представления знаний, обеспечивающего увеличение ценности информации на каждом более высоком уровне иерархии.

ChatGPT реализует последовательную схему рассуждений при построении цепочки токенов, такая стратегия позволяет эффективно обрабатывать знания, как сложную сетевую иерархию элементов информации с выявленными зависимостями и/или существенными связями между фактами, событиями, явлениями и процессами, но в данном случае понимание сложного контента с определением и осознанием закономерностей отношений между распределенными и неоднородными объектами знаний является недостоверным. Продвижение по уровням данной иерархии информационных категорий – не механическое суммирование, а получение на каждом шаге новой основы для более высокого качества знаний. В таких условиях проблема обработки длинных связей между понятиями станет для ChatGPT непреодолимым препятствием. Успешное решение данной задачи смогут обеспечить механизмы диверсификации пространства поиска. Отметим, что децентрализованное управление в иерархических интеллектуальных системах является предпочтительнее централизованного способа, так как это обеспечивает экономию памяти и увеличение скорости диспетчирования.

Еще более проблематичной задачей является построение адекватного признакового пространства для эмбеддинга. Скалярное произведение семантических векторов различных ресурсов позволяет оценить релевантность исследуемых знаний. Для построения семантического вектора информационного ресурса необходимо описать достаточное количество устойчивых кластеров признаков, отражающих контекст и тип конкретной модели представления знаний. Феноменология оценки семантической близости не доступна для прямого

исследования или измерения, что представляет собой значимое препятствие для успешной разработки подходов, решающих описанные проблемы. Сложность определения системно значимых формальных признаков в условиях несимметричности семантической близости и создания категориального аппарата семантического поиска является следствием этого.

Развитие подходов к построению многоуровневых информационных моделей, в том числе онтологических, привело к представлению знаний в виде следующей многоуровневой структуры, показанной на рисунке 1.5.

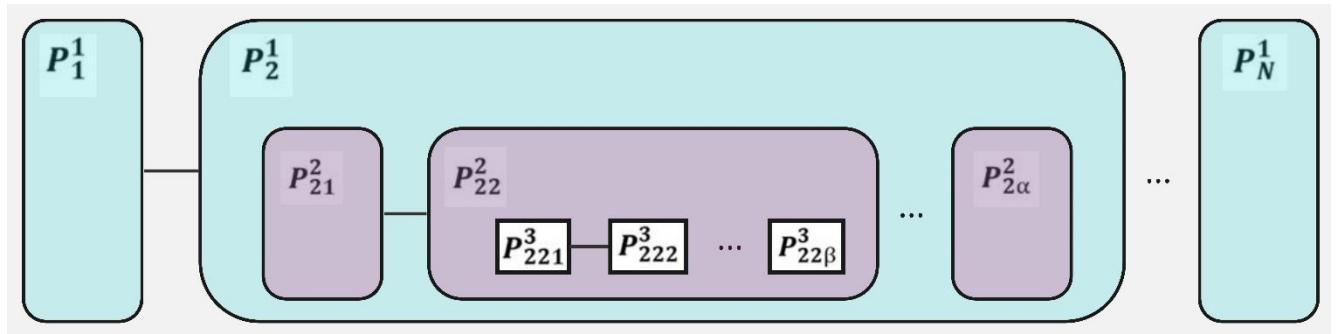


Рисунок 1.5 – Многоуровневая структура объектов знаний

Между понятиями  $P$  разных уровней, представленных на рисунке 1.5, устанавливаются отношения различного типа. Каждый такой объект знаний имеет разную глубину вложения уровней декомпозиции. Знания обладают внутренней интерпретируемостью и структурированностью. На множестве сложноструктурированных знаний устанавливается также внешняя структура отношений различного типа [57], задающих иерархию объектов знаний.

Фрагмент, состоящий из двух соседних уровней описанной выше структуры знаний, представим в виде следующих выражений [9]:

$$P_i^\gamma = \{P_{i1}^{\gamma+1}, P_{i2}^{\gamma+1}, \dots, P_{i\alpha}^{\gamma+1}\}; \quad (1.12)$$

$$P_{ij}^{\gamma+1} = \{P_{ij1}^{\gamma+2}, P_{ij2}^{\gamma+2}, \dots, P_{ij\beta}^{\gamma+2}\}, \quad (1.13)$$

где  $P_i^\gamma$  – понятие информационной модели знаний под номером  $i$  на уровне иерархии  $\gamma$ ;  $i$  – номер понятия на уровне  $\gamma$  ( $i = 1, \dots, N$ );  $N$  – количество понятий на уровне  $\gamma$ ;  $\gamma$  – номер уровня иерархии структуры знаний ( $\gamma = 1, \dots, L-2$ );  $L$  –

количество уровней иерархии представленной структуры знаний;  $j$  – номер понятия на уровне  $\gamma+1$  ( $j = 1, \dots, a$ );  $\alpha$  – количество понятий на уровне  $\gamma+1$ ;  $\beta$  – количество понятий на уровне  $\gamma+2$ .

Построение групп объектов одного уровня описанной выше структуры знаний на основе использования признакового описания в виде векторного представления проводится на основе классификации. Определение значимых отношений на множестве объектов знаний одной предметной области или на междисциплинарном уровне является основой процессов приобретения знаний.

Обработка такой многоуровневой системы опосредованных отношений между элементами знаний с помощью ChatGPT возможна только при условии грамотно построенной последовательности запросов (промптов). Получается, что сам по себе ChatGPT не может «понимать» смысл обрабатываемого контента, это «понимание» ему дает специалист (промптер), который через логически построенную схему запроса определяет для интеллектуальной генеративной системы процесс принятия решений.

Как видно из проведенного в данном пункте анализа, системы генеративного интеллекта для достижения возможности понимания контекста знаний должны преодолеть ряд следующих ограничений:

- 1) понимание сложного содержания с определением и осознанием закономерностей отношений между распределенными и разнородными объектами знания нереализуемо при последовательной схеме рассуждения;
- 2) сложность определения системно значимых формальных признаков в условиях несимметричности семантической близости и создания категориального аппарата семантического поиска;
- 3) ChatGPT не может понять смысл обрабатываемого контента; это понимание ему дает специалист (промптер), который через логически построенную схему запроса определяет траекторию принятия решения интеллектуальной генеративной системой.

Определим основные направления развития алгоритмов и механизмов использования приобретенных системами генеративного искусственного интеллекта знаний, которые позволяют выйти на уровень понимания смысла обрабатываемой текстовой информации.

### 1.5. Алгоритмы и механизмы использования знаний при обработке и анализе текстов

В условиях стохастичности, частичной наблюдаемости и динамичности информационных моделей представления знаний (онтологий), усугубляемых проблемой несимметричности семантической близости понятий (концептов), очевидным является тот факт, что для понимания смысла интеллектуальной системе недостаточно иметь только знания о контексте и семантической близости (semantic similarity) понятий, как это показано на рисунке 1.6.

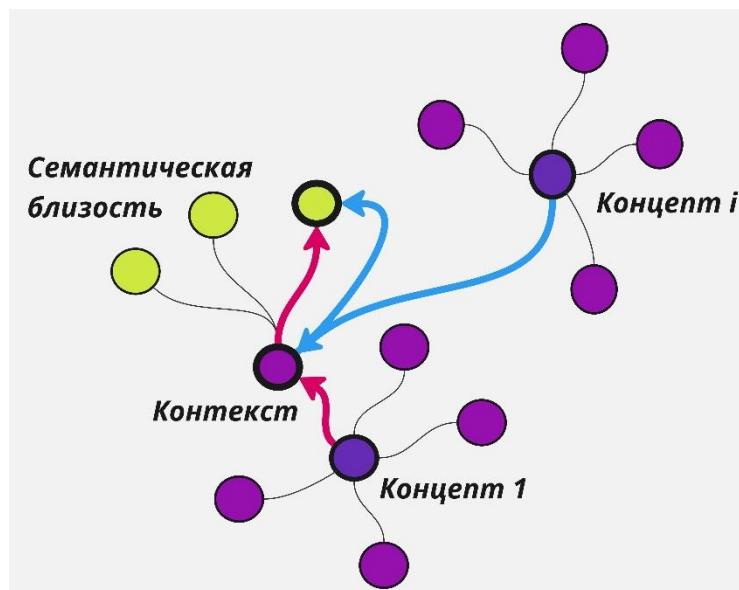


Рисунок 1.6 – Построение отношений между концептами

Подобная ситуация приводит к «галлюцинациям» ChatGPT (выдача неконтролируемых и недостоверных результатов) на основе неправильной интерпретации информации или при отсутствии достаточного описания контекста. «Галлюцинации» ChatGPT могут иметь серьезное влияние на создаваемый контент,

особенно если он используется в критических областях, таких как, например, поддержка принятия решений при предотвращении или ликвидации последствий чрезвычайных ситуаций. Кроме того, «галлюцинации» могут привести к созданию фейковых новостей и манипуляции общественным мнением. Если система создает текст, который содержит неправильную информацию о политических событиях или общественных проблемах, это может привести к негативным последствиям для общества в целом.

Для исключения возможности появления «галлюцинаций» и расширения пространства признаков необходимо обеспечить обучение системы на достаточно большом и разнообразном объеме данных. Одной из важных характеристик этого является гранулярность (granularity) информации (рис. 1.7). Гранулярность – это уровень детализации или точности в данных. Введение уровня детализации и уточнения информации позволит системе генеративного искусственного интеллекта правильно обрабатывать иерархические отношения.

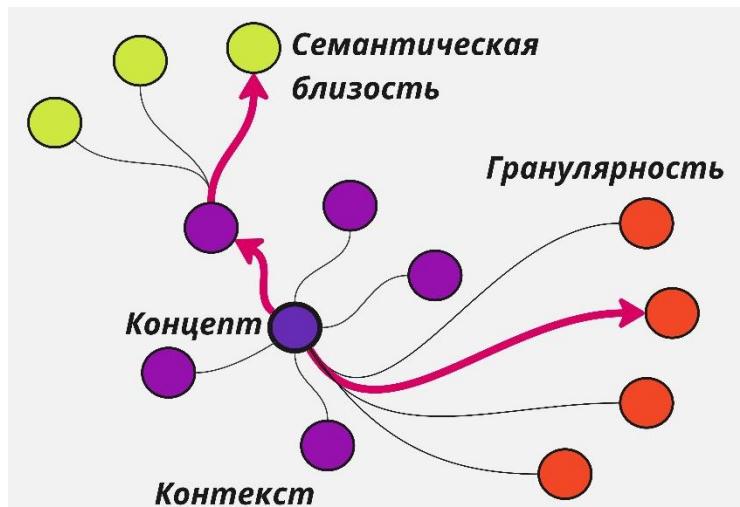


Рисунок 1.7 – Гранулярность как основа обработки иерархических отношений

Информационное пространство обладает свойствами динамичности, стохастичности и частичной наблюдаемости. Модель Мира требует постоянного обновления и сопровождения. В подобных условиях принятие решений интеллектуальной системой должно осуществляться с учетом версионности (version) обрабатываемых знаний (рис. 1.8).

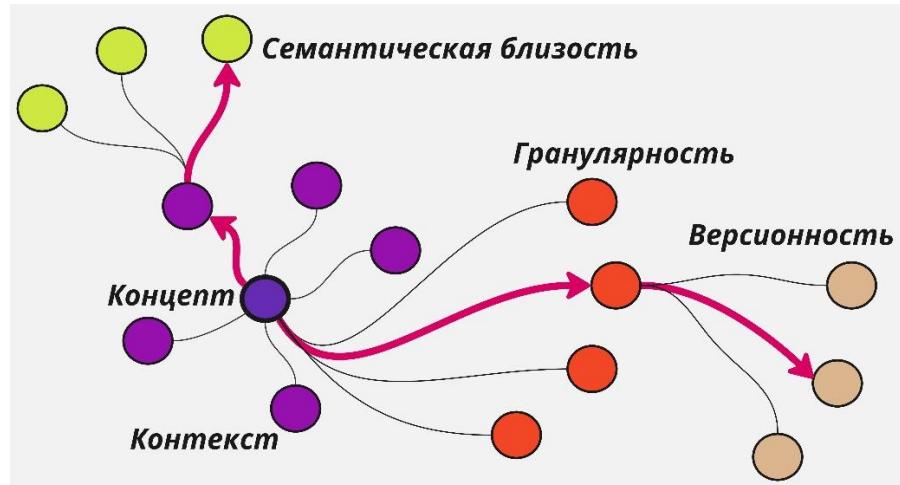


Рисунок 1.8 – Обновление на основе версионности информации

Важно иметь механизмы управления версиями и обновлениями модели мира. Это позволяет отслеживать изменения, поддерживать совместимость с предыдущими версиями и управлять процессом обновления модели.

Специалисты из разных сфер науки и экономики могут иметь разный взгляд (view) на понятия одной предметной области (рис. 1.9). Онтология может содержать правила и механизмы для разрешения такой неоднозначности (ambiguity), связанной с семантикой и интерпретацией текста на естественном языке. Необходимо разработать стратегии и алгоритмы для дисамбигуации неоднозначностей в тексте. Это может включать определение контекстуального значения слова, выявление семантических связей между фразами и разрешение противоречий в информации.

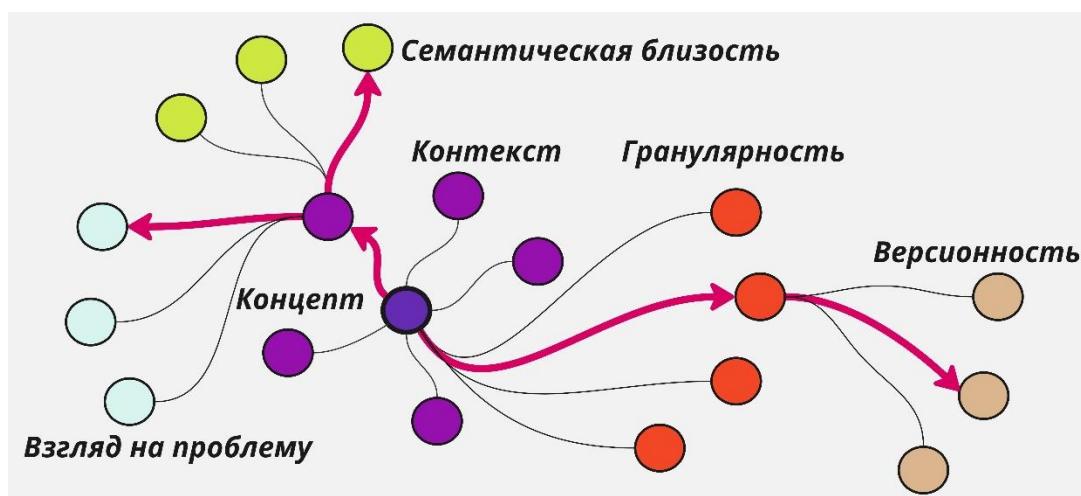


Рисунок 1.9 – Неоднозначность взглядов

Таким образом, извлечение смысла из источников знаний является гораздо более сложным процессом, требующим учета множества параметров, характеристик и особенностей (рис. 1.10) [18, 25, 26, 58].

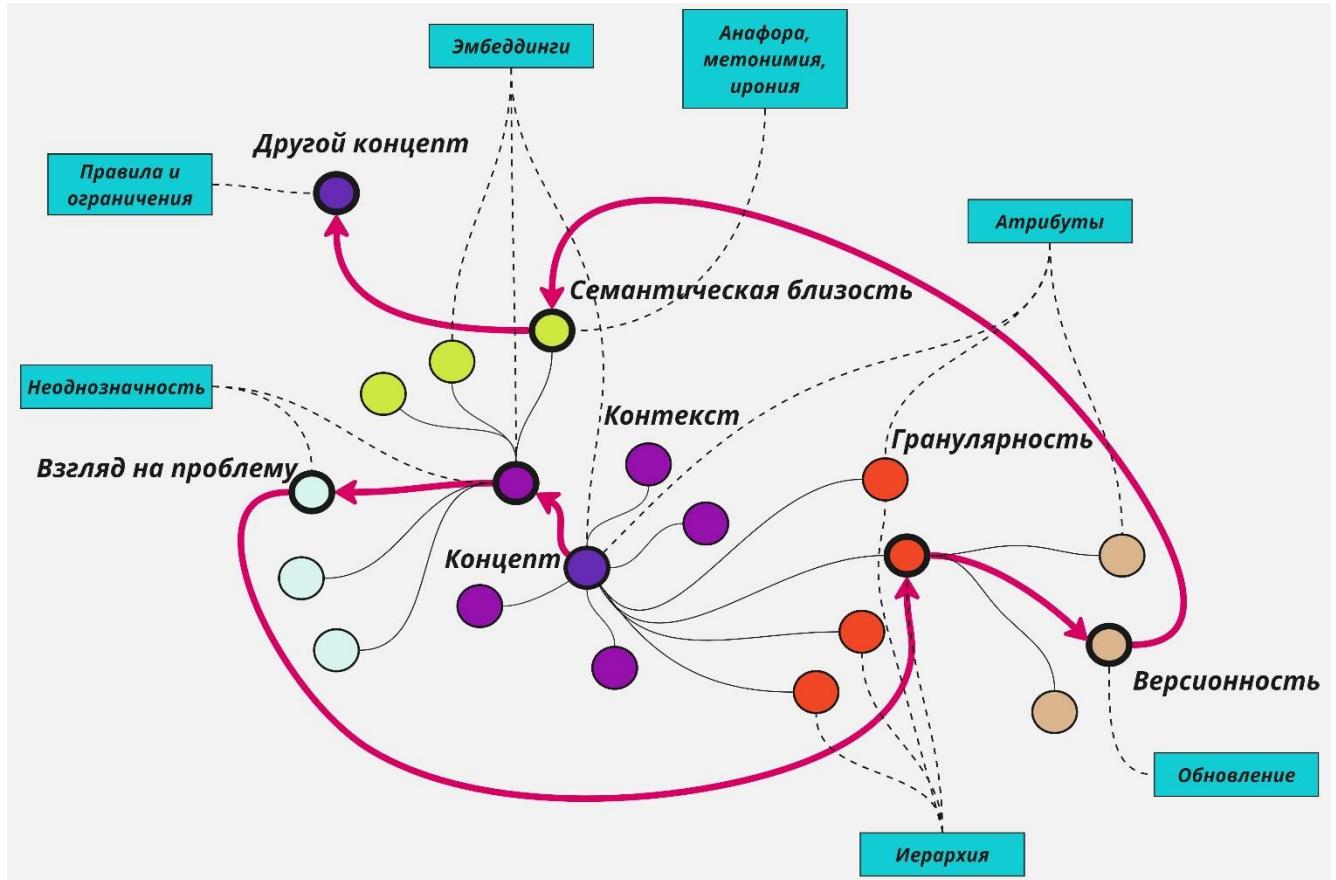


Рисунок 1.10 – Процесс извлечения смысла при обработке текста

Для реализации анализа семантического контекста в модели мира недостаточно построить только контекстные эмбеддинги. Контекстные эмбеддинги (embeddings) – это векторные представления слов или фраз, учитывающие контекст, в котором они появляются. Они позволяют улавливать семантические сходства и различия в разных контекстах.

Отметим, что смысл слов и выражений может меняться в зависимости от контекста, в котором они используются. Необходимо разработать механизмы для анализа и понимания контекстуальных отношений, таких как анафора, метонимия или ирония.

Анафора (anaphora) – это явление, когда наличие или отсутствие определенных выражений в тексте зависит от предыдущего контекста. Например,

в предложении: "Она взяла книгу и начала читать. Он сидел рядом и смотрел на нее", слова "она", "он" и "ее" связаны анафорически и указывают на одну и ту же персону. Модель Мира должна быть способна распознавать и устанавливать связи между такими анафорическими выражениями.

Метонимия (metonymy) – это замена одного выражения другим на основе контекстуальной связи. Например, в фразе "Прочитай Гоголя" слово "Гоголь" используется вместо "произведений Гоголя". Модель Мира должна иметь знания о таких связях и быть способной распознавать, когда метонимические выражения используются в тексте.

Ирония (irony) – это использование выражений, противоположных их буквальному значению, с целью передачи скрытого смысла. Ирония может быть сложной для автоматического понимания, поскольку она требует учета контекста и намерений говорящего. Модель Мира должна позволять распознавать и идентифицировать иронические выражения на основе контекста и других сигналов, таких как сарказм или противоречие с ожидаемым смыслом [58].

Таким образом, для эффективного решения задач использования знаний при обработке текстовой информации системы генеративного искусственного интеллекта должны преодолеть следующие ограничения:

- 1) «галлюцинации» ChatGPT могут оказать серьезное влияние на создаваемый контент, особенно если он используется в критических областях;
- 2) введение уровня детализации и уточнения информации позволит системе генеративного искусственного интеллекта правильно обрабатывать иерархические связи;
- 3) онтология должна содержать правила и механизмы исключения неоднозначности, связанной с семантикой и интерпретацией смысла текстовой информации;
- 4) недостаточно построить только контекстные вектора (embedding), чтобы реализовать семантический контекстный анализ в модели Мира.

Проведенный автором анализ проблем повышения эффективности при условии обеспечения «прозрачности» процессов поиска, приобретения и использования знаний в системах искусственного интеллекта при обработке и анализе текстов на естественном языке дает понимание особенностей исследуемой предметной области. Представим постановки основных задач исследования.

## 1.6. Постановка основных задач исследования

Задача *поиска знаний* в системах искусственного интеллекта при обработке и анализе текстов на естественном языке в большей степени связана с качественным построением синтаксической схемы текста на основе применения парсера. С этой точки зрения наиболее важными являются процедуры структурирования и фильтрации текстовой информации, которые позволяют определить основные смысловые элементы текста, такие как, например, ключевые слова.

Извлечение ключевых слов является фундаментальной задачей в обработке естественного языка (Natural Language Processing, NLP) и включает в себя выявление и извлечение наиболее релевантных и значимых слов или фраз из заданного текста. Парсеры играют ключевую роль в этом процессе, анализируя синтаксическую структуру текста и помогая выявлять ключевые компоненты, представляющие важные понятия или темы.

1. *Синтаксический анализ структуры для извлечения ключевых слов.* Для выполнения извлечения ключевых слов с использованием синтаксического анализа структуры, можно сосредоточиться на конкретных синтаксических единицах, таких как именные фразы (NPs) [59] и глагольные фразы (VPs). Эти фразы часто являются хорошими кандидатами на роль ключевых слов, так как обычно содержат важную информацию о субъекте, объекте или действии в предложении. Например, рассмотрим предложение: "The swift fox jumps over the lazy dog." С помощью синтаксического анализа структуры будут выделены следующие фразы:

- именные фразы (NPs): "The swift fox", "the lazy dog";

- глагольная фраза (VP): "jumps over".

Из извлеченных фраз можно выделить следующие ключевые слова: "swift fox" (быстрая лиса); "lazy dog" (ленивая собака); "jumps over" (перепрыгивает через), как существенные компоненты данного предложения.

## 2. Синтаксический анализ зависимостей для извлечения ключевых слов.

Синтаксический анализ зависимостей помогает определить отношения между субъектом, глаголом и объектом, а также другие существенные синтаксические зависимости, которые вносят вклад в общий смысл предложения. Ключевые слова могут быть извлечены из этих зависимостей, причем предпочтение отдается словам, несущим значительный семантический вес и играющим важные роли в структуре предложения [35].

В контексте извлечения ключевых слов текст преобразуется в граф, где вершины представляют собой возможные ключевые слова, а ребра – их отношения. Взаимосвязь между ключевыми фразами-кандидатами может быть определена по тому, как часто они встречаются вместе или насколько семантически близки.

Предположим, что строится ориентированный граф  $G = (V, E)$ , где  $V$  – множество вершин, а  $E$  – множество ребер. Оценка или важность вершины определяется как [35, 59]:

$$S(V_i) = (1 - d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j), \quad (1.14)$$

где  $In(V_i)$  – это набор вершин, которые указывают на  $V_i$ , а  $Out(V_i)$  – это набор вершин, на которые указывает  $V_i$ . При этом  $d$  – это коэффициент затухания, который устанавливается в диапазоне от 0 до 1.

В рамках решения задачи *поиска знаний* в системах искусственного интеллекта при обработке и анализе текстов на естественном языке данное исследование предполагает создание дополнительного фильтра на выходе парсера, что позволит извлечь смысловую часть предложения из полученной

синтаксической схемы текстовой информации для дальнейшего использования в процессах приобретения знаний.

При решении задач *приобретения знаний* необходимо сформулировать постановку и условия задачи, равно как и само понятие решения. Пусть существует определенный подход  $S$ , позволяющий на множестве  $K$  анализируемых объектов знаний из множества  $P$  разных предметных областей атрибутам (характеристикам)  $X$  ставить в соответствие значения (величины)  $Y$ , которые приводят к оптимальной обработке разнородных распределенных информационных ресурсов  $R$  на основе применения методов  $M$ . При этом обработка информационных ресурсов должна проводиться с учетом контекста  $H$ , что является необходимым условием повышения эффективности информационного процесса приобретения знаний (функции)  $F$ . Таким образом,  $Y$  является достоверным выводом, а  $S$  – решающим подходом, построенным на основе комплекса методов и правдоподобных рассуждений. Очевидно, что в данном случае эффективность вывода напрямую связана с интеллектуальностью метода обработки и анализа имеющихся информационных ресурсов. Полученный вывод представляет собой систему проанализированных отношений и соответствий, приводящих к обнаружению неявных зависимостей и закономерностей между атрибутами  $X$  на множестве объектов знаний  $K$ . Идентифицированные новые зависимости и закономерности между объектами знаний являются основой для приобретения нового знания [9].

Реализация моделей и методов приобретения знаний на основе применения систем искусственного интеллекта существенно изменило аппарат формальных рассуждений. Основной особенностью задач построения рассуждений в контексте данного исследования является наличие информационной неопределенности и большой размерности, что требует получения и анализа значительного числа альтернативных вариантов структуры системно значимых отношений между объектами знаний исследуемых предметных областей.

Задача *приобретения знаний* сводится к оценке семантической близости понятий (концептов), позволяющей распределить их по классам для дальнейшего построения онтологии предметной области.

Представим постановку задачи *классификации знаний* – процесса группировки объектов знаний в соответствии с их системно значимыми признаками. Дадим постановку этой задачи на основе использования признакового описания исследуемых объектов знаний в виде векторного представления.

Предположим, что имеется  $N$  объектов знаний, каждый из которых описывается  $M$  признаками. Пронумеруем признаки – индексом  $m$  ( $m = 1, \dots, M$ ), а объекты индексом  $n$  ( $n = 1, \dots, N$ ). Предположим так же, что имеется  $K$  наименований классов, к которым необходимо отнести имеющиеся объекты знаний. Каждый класс, так же как и объект, описывается  $M$  признаками. Пронумеруем признаки классов индексом  $m$  ( $m = 1, \dots, M$ ) соответственно, а классы – индексом  $k$  ( $k = 1, \dots, K$ ). Для объекта знания с номером  $n$  обозначим через  $x_{m,n}$  – значение признака  $m$ , а через  $y_{m,k}$  – значение целевого признака  $m$  для класса  $k$ . Тогда по аналогии с выражениями (1.34) и (1.35) сформулируем постановку задачи классификации знаний. Рассмотрим объект знаний  $\vec{X}_n$  и класс  $\vec{Y}_k$  [9]:

$$\vec{X}_n = (x_{1,n}, x_{2,n}, \dots, x_{M,n}); \quad (1.15)$$

$$\vec{Y}_k = (y_{1,k}, y_{2,k}, \dots, y_{M,k}). \quad (1.16)$$

Известно, что скалярное произведение векторов и косинус угла  $\theta$  между ними связаны следующим соотношением:

$$(X_n, Y_k) = \|X_n\| \cdot \|Y_k\| \cdot \cos(\vec{X}_n, \vec{Y}_k); \quad (1.17)$$

$$\|X_n\| = \sqrt{x_{1,n}^2 + x_{2,n}^2 + \dots + x_{M,n}^2}; \quad (1.18)$$

$$\|Y_k\| = \sqrt{y_{1,k}^2 + y_{2,k}^2 + \dots + y_{M,k}^2}. \quad (1.19)$$

Определим расстояние между рассматриваемым объектом и классом:

$$\|X_n - Y_k\| = \sqrt{(x_{1,n} - y_{1,k})^2 + (x_{2,n} - y_{2,k})^2 + \dots + (x_{M,n} - y_{M,k})^2}. \quad (1.20)$$

Введём пороговое значение  $\Delta$  максимального расстояния, превышение которого исключает возможность отнесения объекта  $\vec{X}_n$  к классу  $\vec{Y}_k$ . В этом случае оптимизационная задача классификации примет следующий вид [9]:

$$Y^* = \arg \min_k \|X_n - Y_k\|; \quad (1.21)$$

$$\begin{cases} \|X_n - Y^*\| \leq \Delta, Y^* \text{ -- определен для } X_n \\ \text{otherwise, } X_n \text{ -- не классифицирован} \end{cases}. \quad (1.22)$$

В случае, когда признаки принимают числовые значения, но не являются координатами, рассмотрено описание объектов знаний и классов в виде упорядоченных множеств (кортежей) системно значимых признаков. Определим объект знаний  $X_n$  и класс  $Y_k$ :

$$X_n = \langle x_{1,n}, x_{2,n}, \dots, x_{M,n} \rangle; \quad (1.23)$$

$$Y_k = \langle y_{1,k}, y_{2,k}, \dots, y_{M,k} \rangle. \quad (1.24)$$

Наилучшим решением элементарной процедуры распределения объектов знаний по классам и группам в задачах классификации или структурирования на данных упорядоченных множествах является следующий вариант равенства представленных кортежей [9]:

$$\forall i = 1, \dots, M; x_{i,n} = y_{i,k}. \quad (1.25)$$

Очевидно, что в условиях информационной неопределенности данный вариант реализуется на практике крайне редко. Поэтому предусмотрено введение допустимого порога на несоблюдение данного равенства кортежей значений признаков  $X_n$  и  $Y_k$ . Данный допустимый порог  $\Delta > 0$  вводится предварительно и имеет положительное целочисленное значение, которое устанавливает максимальное число признаков  $X_n$ , отличие значений которых от соответствующих значений признаков  $Y_k$  считается незначительным и позволяет отнести  $X_n$  к классу

(группе)  $Y_k$ . Тогда алгоритм решения оптимизационной задачи определения соответствия объекта знаний  $X_n$  классу или группе  $Y_k$  примет следующий вид [9]:

<i>Алгоритм определения соответствия объекта знаний <math>X_n</math> классу или группе <math>Y_k</math></i>	
Ввод	Кортежи значений признаков $X_n$ и $Y_k$ ; значение $\Delta$ допустимого порога неравенства значений признаков $X_n$ и $Y_k$
Вывод	Результат классификации (структурирования) $X_n$ в $Y_k$
1:	$i = 1$
2:	<i>While</i> ( $i \leq M$ ) <i>do</i>
3:	<i>if</i> ( $x_{i,n} \neq y_{i,k}$ ) <i>then</i>
4:	$\Delta = \Delta - 1$
5:	$i++$
6:	<i>else</i>
	$i++$
	<i>end</i>
	<i>end</i>
7:	<i>if</i> ( $\Delta \geq 0$ ) <i>then</i>
8:	$  Y_k$ – определен для $X_n$
	<i>else</i>
9:	$  X_n$ – не классифицирован
	<i>end</i>
10:	Сохранение результата

Эффективное *использование знаний* при обработке и анализе текстов на естественном языке предполагает построение системы опосредованных отношений между гетерогенными элементами знаний, поэтому актуальной задачей является *интеграция* онтологий различных предметных областей. Дадим постановку задачи *интеграции знаний* множества онтологий [9]:

$$O^U = \bigcup_i O_i, i = \overline{1, N}, \quad (1.26)$$

где  $O_i$  – онтограф  $i$ -той предметной области;  $i$  – номер рассматриваемой предметной области;  $N$  – общее число предметных областей.

Преимуществом эффективного решения задачи интеграции знаний является увеличение вероятности возникновения новых научных направлений, так, например, появились биофизика и геохимия. Междисциплинарный подход

решения данной задачи позволяет заимствовать для научных исследований общие принципы, приемы, способы и методы из других наук.

В данном пункте даны постановки основных задач исследования.

Задача *поиска знаний* в системах искусственного интеллекта при обработке и анализе текстов на естественном языке представлена с точки зрения повышения эффективности процедур структурирования и фильтрации текстовой информации. В рамках решения задачи поиска знаний данное исследование предполагает создание дополнительного фильтра на выходе парсера, что позволит извлечь смысловую часть предложения из полученной синтаксической схемы текстовой информации для дальнейшего использования в процессах приобретения знаний.

Задача *приобретения знаний* сводится к оценке семантической близости понятий (концептов), позволяющей распределить их по классам для дальнейшего построения онтологии предметной области.

Эффективное *использование знаний* при обработке и анализе текстов на естественном языке предполагает построение системы опосредованных отношений между гетерогенными элементами знаний, поэтому актуальной задачей является *интеграция онтологий* различных предметных областей.

Выполнение поставленных в данном пункте задач обеспечивает построение упорядоченной структуры извлекаемых знаний в информационном пространстве поиска решений на основе детерминированных («прозрачных») моделей и алгоритмов, что позволяет в итоге *минимизировать время отклика системы искусственного интеллекта на запрос пользователя* ( $T_{response} \rightarrow \min$ ).

## 1.7. Выводы по разделу

Первый раздел посвящен анализу исследуемой предметной области.

Проведен аналитический обзор особенностей создания систем искусственного интеллекта и машинного обучения для поиска, приобретения и использования знаний при обработке и анализе текстов на естественном языке.

Особое внимание уделено построению компонентной архитектуры подобных систем. Исследованы основные современные модели, алгоритмы, механизмы и инструменты решения задач поиска, приобретения и использования знаний при обработке и анализе текстов. Проанализированы понятия ценности и действенности информации. Исследованы особенности векторизации и оценки семантической близости текстовой информации.

По мнению автора, эффективное решение задачи исключения избыточной информации из текста требует применения низкоуровневых алгоритмов и правил, в том числе на основе использования графовых моделей для создания дополнительного фильтра на выходе парсера, позволяющего извлечь смысловую часть предложения из полученной синтаксической схемы. Комбинированное использование методов извлечения концептов, основанных на низкоуровневых правилах, совместно с методами машинного обучения, повышает качество решения задач поиска, приобретения и использования знаний при обработке и анализе текстов на естественном языке.

Среди наиболее подходящих концептуальных моделей для формализации представления концептов выделены *графы* и *решётки понятий*. Преимуществом концептуальных моделей является возможность построения как бинарных, так и *n*-арных отношений на множестве понятий (концептов).

Проанализирована проблема отсутствия «прозрачности» в процессах *поиска знаний* при обработке текстов на естественном языке, связанная с использованием нейросетей при анализе зашумленных структур текстовой информации, полученных после работы текстового парсера.

Проведен анализ проблем *приобретения знаний*. Основной акцент сделан на оценку точности обработки текстовой информации в системах генеративного искусственного интеллекта, а также соответствия используемых моделей информационного пространства сложности решаемых задач приобретения знаний.

Представлен анализ проблем *использования знаний* при решении задач обработки текстовой информации. Описаны изменения в моделях обработки

приобретенных системами генеративного искусственного интеллекта знаний, необходимые для выхода на уровень понимания смысла используемой текстовой информации. Выделены основные ограничения систем генеративного искусственного интеллекта.

Представлены постановки основных задач исследования. Задача *поиска знаний* в интеллектуальных системах обработки и анализа текстов на естественном языке в большей степени связана с качественным построением синтаксической схемы текста на основе применения парсера. Данное исследование предполагает создание дополнительного фильтра на выходе парсера, что позволит извлечь смысловую часть предложения из полученной синтаксической схемы текстовой информации для дальнейшего использования в процессах приобретения знаний.

Задача *приобретения знаний* сводится к оценке семантической близости понятий (концептов), позволяющей распределить их по классам для дальнейшего построения онтологии предметной области. Данная формальная постановка задачи *классификации знаний* – процесса группировки объектов знаний в соответствии с их системно значимыми признаками.

Эффективное *использование знаний* при обработке и анализе текстов на естественном языке предполагает построение системы опосредованных отношений между гетерогенными элементами знаний, поэтому актуальной задачей является *интеграция* онтологий различных предметных областей. Успешное решение поставленных задач обеспечивает *снижение времени отклика системы на запрос пользователя*.

Следующий раздел диссертации посвящен построению верхнеуровневой и нижнеуровневой моделей онтологии знаний, применяемых при обработке и анализе текстов, отличающихся использованием оригинальных компонентной архитектуры и структуры отношений между понятиями, которые позволяют обеспечить необходимую степень детализации анализируемой текстовой информации, а также создание набора смысловых паттернов с возможностью проведения оценки их семантической близости.

## 2. ПОСТРОЕНИЕ МОДЕЛЕЙ ОНТОЛОГИИ ЗНАНИЙ

Данный раздел посвящен построению верхнеуровневой и нижнеуровневой моделей онтологии знаний для обработки и анализа текстов на естественном языке. Верхнеуровневая модель онтологии знаний отличается включением в состав ее компонентов множеств понятий с различным уровнем нормализации, что позволяет обеспечить необходимую степень детализации анализируемой текстовой информации. Нижнеуровневая модель онтологии знаний отличается использованием структуры отношений между понятиями, детализирующими семантику текстовой информации, что позволяет получить набор смысловых паттернов, а также проводить оценку их семантической близости. Разработаны эвристические алгоритмы группировки предложений, имеющих схожие смысловые характеристики, а также процедура обработки построенных групп предложений для упрощения последующего анализа текстовой информации.

### 2.1. Верхнеуровневая модель онтологии знаний

Построение верхнеуровневой модели онтологии знаний, применяемой при обработке и анализе текстов на естественном языке, проведено с учетом результатов представленного в предыдущем разделе анализа проблем поиска, приобретения и использования знаний в системах искусственного интеллекта и машинного обучения.

Обработка и анализ текстов на основе извлечения ключевых слов дает значительный эффект при определении смысла предложения. Корпус текста после предварительной обработки приобретает вид входного набора данных, содержащего множество предложений. Для повышения эффективности процесса извлечения ключевых слов онтология имеет список множества их шаблонов (*Key Words, KWs*), что позволяет в дальнейшем на основе процедур нормализации и определения их синонимии находить прецеденты, ускоряющие работу системы

искусственного интеллекта и машинного обучения обработки и анализа текстов (рис. 2.1).

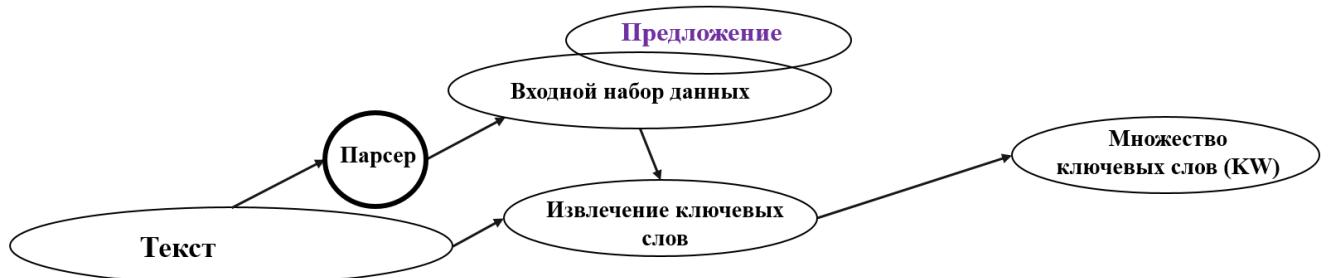


Рисунок 2.1 – Компонент онтологии для извлечения и обработки ключевых слов

Таким образом, представленный на рисунке 2.1 компонент верхнеуровневого описания онтологии знаний включает в себя множество предложений, поступивших на вход системы искусственного интеллекта и машинного обучения (2.1), и множество ключевых слов [60-64], извлеченных и сохраненных в онтологии (2.2):

$$Sent = \{sent_1, \dots, sent_n\}, \quad (2.1)$$

$$KWs = \{kw_1, \dots, kw_m\}. \quad (2.2)$$

На данных множествах построено бинарное отношение инцидентности (2.3) между *Sent* и *KWs*:

$$Rel_{SentKWs} \subseteq Sent \times KWs. \quad (2.3)$$

В онтологии обязательно должно храниться предложение, из которого было выбрано ключевое слово (*KW*). Также необходимо хранить информацию о том, является *KW* объектом или субъектом, так как это сильно влияет на векторную презентацию *KW*. Вектор *KW* также хранится в онтологии. Каждое вновь извлеченное ключевое слово попадает в онтологию либо новым элементом множества *KWs*, либо как синоним уже существующего *KW*. При этом, необходимо учитывать лингвистические особенности нового элемента онтологической модели (часть речи, род, число, лицо и т.п.), так как эта информация значительно влияет на смысл предложения (рис. 2.2) [65-68]. Например, в следующих предложениях: «Я

системный администратор»; «Я выполняю функции системного администратора»; «Я отвечаю за системное администрирование»; «Он хотел бы стать системным администратором», – при одинаковых (или почти одинаковых) ключевых словах определяется разный смысл. Нужен каталог *KWs* для каждой доменной (предметной) области, так как *KWs* нужны до начала разбора предложения. Все слова в онтологии надо делить на категории «значимые» (*Meaningful Words, MWs*) и «уточняющие» (*Clarifying Words, CWs*). Для «значимых» слов должны быть сформированы списки синонимов, а также учтены лингвистические особенности.

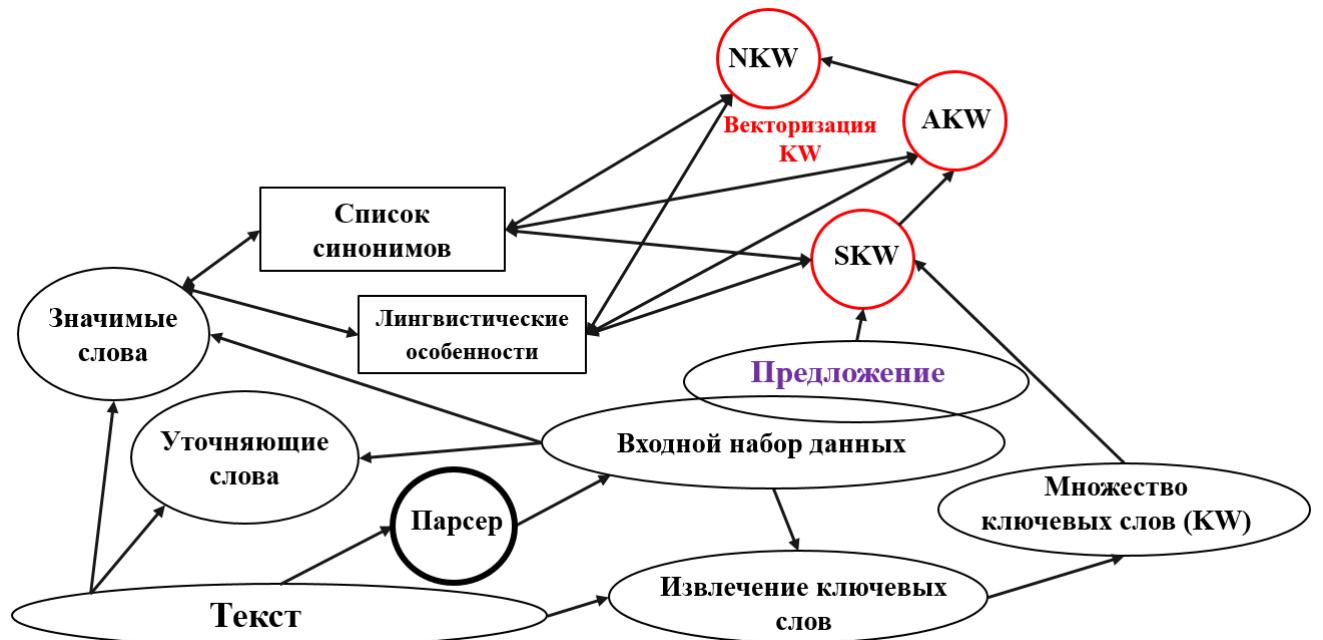


Рисунок 2.2 – Расширенный компонент онтологии для извлечения и обработки ключевых слов

Применение онтологии повышает эффективность управления ключевыми словами, что позволяет организовать поиск синонимов (их «склеивание») и, при необходимости, проводить модификацию онтологии (изменение системы внутренних связей между понятиями (концептами)) [67, 68]. Для этого построена иерархия ключевых слов, в которой на верхнем уровне находится поисковое ключевое слово (*Search Key Word, SKW*), включенное в запрос пользователя на входе системы искусственного интеллекта и машинного обучения, например, «транспортное предприятие». Затем, после обработки предложений, на следующем

уровне иерархии появляется набор актуальных ключевых слов (*Actual Key Word, AKW*), например, более широкое понятие – «транспортно-логистический кластер» или «транспортный холдинг» и т.п. Данная группа синонимичных AKWs на следующем уровне иерархии объединяется в единый класс, характерным понятием которого является нормализованное ключевое слово (*Normalized Key Word, NKW*) – часто встречающееся ключевое слово в данном контексте, с наибольшим значением семантической близости, полученной на основе оценки косинусной меры сходства с вектором признаков данного класса (рис. 2.2).

Таким образом, представленный на рисунке 2.2 расширенный компонент верхнеуровневого описания онтологии знаний включает в себя множество «значимых» слов с точки зрения извлекаемого смысла *MWs* (2.4), множество «уточняющих» слов *CWs* (2.5), множество поисковых ключевых слов *SKWs* (включает в себя извлекаемое из предложений множество *KWs*, а также ключевые слова, поступающие в интеллектуальную систему через запросы пользователей) (2.6), множество актуальных ключевых слов *AKWs* (2.7), множество нормализованных ключевых слов *NKWs* (2.8):

$$MWs = \{mw_1, \dots, mw_e\}, \quad (2.4)$$

$$CWs = \{cw_1, \dots, cw_j\}, \quad (2.5)$$

$$SKWs = \{skw_1, \dots, skw_x\}, \quad (2.6)$$

$$AKWs = \{akw_1, \dots, akw_y\}, \quad (2.7)$$

$$NWKs = \{nkw_1, \dots, nkw_z\}. \quad (2.8)$$

На данных множествах построены следующие бинарные отношения инцидентности (2.9), (2.10):

$$Rel_{SentMWs} \subseteq Sent \times MWs, \quad (2.9)$$

$$Rel_{SentCWS} \subseteq Sent \times CWs, \quad (2.10)$$

а также следующие бинарные отношения инцидентности (2.11) - (2.15) между синонимами:

$$Sym_{SKWSAKWS} \subseteq SKWs \times AKWs, \quad (2.11)$$

$$Syn_{SKWsNKWs} \subseteq SKWs \times NKWs, \quad (2.12)$$

$$Syn_{AKWsNKWs} \subseteq AKWs \times NKWs, \quad (2.13)$$

$$Syn_{AKWSAKWs} \subseteq AKWs \times AKWs, \quad (2.14)$$

$$Syn_{MWsMWs} \subseteq MKWs \times MKWs. \quad (2.15)$$

Глаголы (*Verbs*) тоже хранятся в онтологии, а также имеют нормализованный вид (*NVerbs*) и взаимосвязи с определенными *KWs* на основе выявленных закономерностей и зависимостей между ними (иногда один глагол может быть связан с несколькими объектами). В онтологии описаны критерии принадлежности к определенной группе контекстов для кластеризации нормализованных смыслов. Предложения с определенными в них *KWs* распределены по контекстам (предметным областям). Предусмотрена фильтрация слов (наиболее часто встречающихся), определяющих конкретный контекст. Сам контекст определяется на основе «реального смысла» *Real Meaning* (триплета «субъект – предикат (глагол) – объект» (*Sbj – Verb – Obj*)), построенного на основе слов взятых непосредственно из предложения) (рис. 2.3). Наиболее проблемной задачей является построение кластеров определенных контекстов.

Таким образом, полное верхнеуровневое описание онтологии знаний для интеллектуальных систем обработки и анализа текстов (рис. 2.3) включает в себя множества глаголов (*Verbs*) (2.16) и нормализованных глаголов (*NVerbs*) (2.17):

$$Verbs = \{verb_1, \dots, verb_s\}, \quad (2.16)$$

$$NVerbs = \{nverb_1, \dots, nverb_h\}. \quad (2.17)$$

Отношения синонимии между элементами данных множеств представлено следующим бинарным отношением инцидентности (2.18):

$$Syn_{VerbsNVerbs} \subseteq Verbs \times NVerbs. \quad (2.18)$$

В построенной онтологии определены ограничения и правила. Например, можно установить правило, что каждое предложение должно содержать как минимум одно существительное и один глагол. Это поможет проверять полноту и состоятельность онтологической структуры при решении задач извлечения знаний.

Необходимо предусмотреть возможность масштабируемости онтологии, что позволит добавлять и изменять ее компоненты.

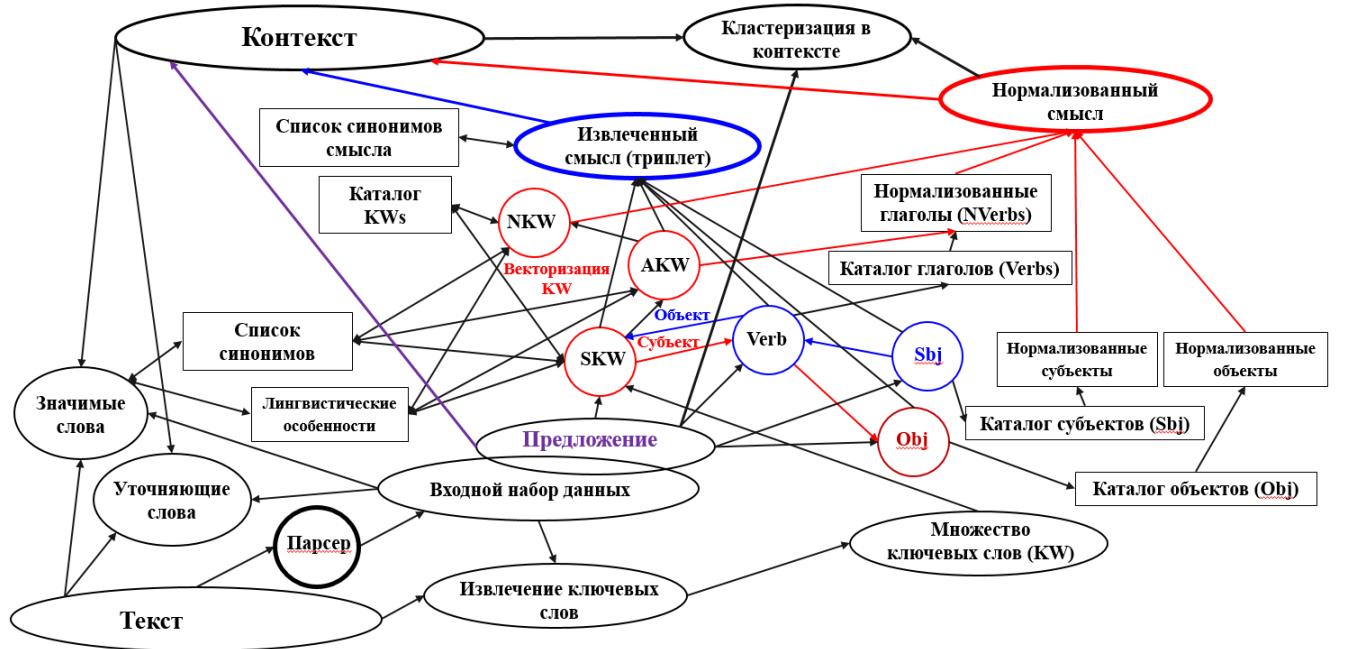


Рисунок 2.3 – Верхнеуровневое описание модели онтологии знаний для интеллектуальных систем обработки и анализа текстов

Имеющиеся в онтологии субъекты ( $Sbj$ ) и объекты ( $Obj$ ) составляют множество концептов (понятий) ( $K_i$ ,  $i = 1...N$ , где  $N$  – количество концептов) с заданной на нём системой связей (бинарных отношений) ( $R_q$ ,  $q = 1...M$ , где  $M$  – количество связей), являющихся по своей сути несимметричными семантическими отношениями, как это было показано в главе 1.

Конечное множество лексических меток (словарь онтологии) представлено множеством  $L$  (2.19):

$$L = \{L_1, \dots, L_u\}. \quad (2.19)$$

Выражением (2.20) представлено антисимметричное, транзитивное, нерефлексивное бинарное отношение, являющееся отношением частичного порядка на множестве понятий (концептов)  $K$ :

$$Rel_K \subseteq K \times K, Rel_K \in K. \quad (2.20)$$

Бинарное отношение инцидентности между множествами  $L$  и  $K$  задано следующим выражением (2.21):

$$Rel_{LK} \subseteq L \times K. \quad (2.21)$$

Бинарное отношение инцидентности между множествами  $L$  и  $R$  задано следующим выражением (2.22):

$$Rel_{LK} \subseteq L \times R. \quad (2.22)$$

Формализованной моделью онтологии верхнего уровня описания знаний является следующий кортеж (2.23):

$$O = < K, R, L, I, Sent, KW, Word, Verb, Rel, Syn, A >, \quad (2.23)$$

где  $I$  – множество экземпляров понятий;  $KW = \{KWs, SKWs, AKWs, NKWs\}$  – интегрированное множество ключевых слов;  $Word = \{MWs, CWs\}$  – интегрированное множество слов в онтологии;  $Verb = \{Verbs, NVerbs\}$  – интегрированное множество глаголов в онтологии;  $Rel = \{Rel_k, Rel_{LK}, Rel_{LR}, Rel_{SentKW}, Rel_{SentMW}, Rel_{SentCW}\}$  – интегрированное множество бинарных отношений между элементами онтологии;  $Syn = \{Syn_{SKWSAKWs}, Syn_{SKWSNKWs}, Syn_{AKWSNKWs}, Syn_{AKWSAKWs}, Syn_{MWSMW}, Syn_{VerbsNVerbs}\}$  – интегрированное множество бинарных отношений синонимии между элементами онтологии;  $A$  – аксиомы онтологии [64].

Таким образом, в данном подразделе построена верхнеуровневая модель онтологии знаний, которая отличается от известных аналогов включением в состав ее компонентов множеств понятий с различным уровнем нормализации, что позволяет обеспечить необходимую степень детализации анализируемой текстовой информации. Представленная модель даёт понимание термина «онтология» в зависимости от контекста текстовой информации, а также целей ее обработки и анализа, что обеспечивает гибкость системы искусственного интеллекта и машинного обучения и повышает эффективность процессов извлечения знаний.

В следующем подразделе в целях детализации семантики отношений между понятиями предложенной онтологической структуры представлено описание построения нижнеуровневой модели онтологии знаний.

## 2.2. Нижнеуровневая модель онтологии знаний

Нижнеуровневая модель онтологии знаний отличается применением структуры отношений между понятиями, детализирующими семантику текстовой информации, что позволяет получить набор смысловых паттернов, а также проводить оценку их семантической близости.

В большинстве случаев *оперативная информация о чрезвычайных ситуациях (ЧС) поступает в центры мониторинга в виде текстовых корпусов данных*. Для построения упорядоченной *системы нечетких правил* при обработке данных корпусов текста создана информационная модель, позволяющая дифференцировать информацию о субъектах и их деятельности в сложившихся условиях, а также об объектах, которые являются либо результатом действия, либо его состоянием, либо инструментом. Все информационные элементы такой модели хранятся в своих доменах (предметных областях) и имеют заданную систему связей между собой как внутри, так и между доменами. Данные связи отражают семантические отношения между информационными элементами и являются основой построения нижнеуровневой модели онтологии знаний для интеллектуальных систем обработки и анализа текстов при решении задач предупреждения и ликвидации последствий ЧС [69, 70].

Представим концептуальное описание указанной информационной модели. Очевидно, что в моделируемых процессах ликвидации последствий ЧС принимают участие действующие субъекты (акторы) различных типов. Например, с одной стороны, это сотрудники МЧС России, которые непосредственно осуществляют действия по предотвращению и/или устраниению последствия ЧС, а с другой стороны – потенциальные потерпевшие, действия которых либо отличаются от действий первой группы субъектов, или являются взаимосвязанными с ними. В моделируемых процессах также выделены другие типы субъектов [69].

*Каждый тип субъектов выделен в отдельную группу данных.* В этой группе обозначен основной элемент (субъект), например, «сотрудник МЧС». Все

остальные элементы уточняют активность основного субъекта, по сути, являясь его синонимом. Объединенные в подгруппы субъекты одного типа, выполняющие определенные коллективные функции, прописаны в информационной модели как элемент, имеющий множественное число (plural) [69]. Таким образом, концептуальная информационная модель для одного типа субъектов примет вид, представленный на рисунке 2.4.

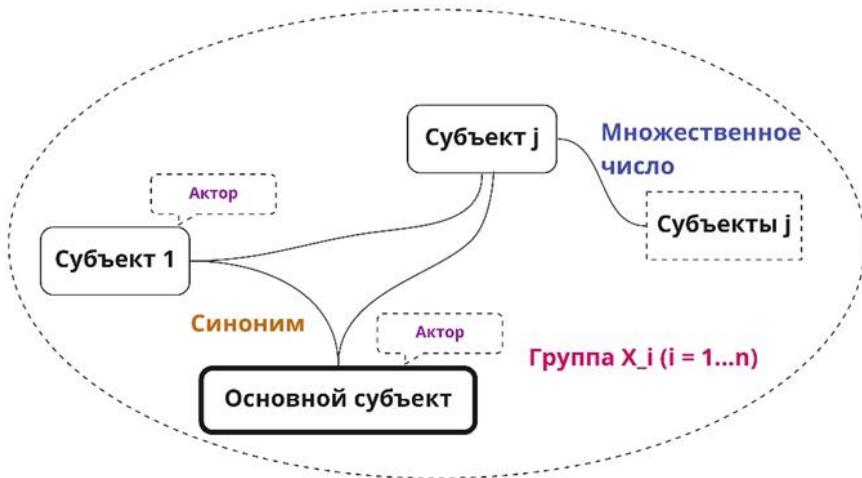


Рисунок 2.4 – Концептуальная информационная модель для одного типа субъектов

Каждый субъект в модели имеет свой вид деятельности, который должен быть задан активным глаголом (предикатом). Необходимо отметить, что существует достаточное количество открытых программных приложений (parser), которые позволяют автоматически декомпозировать корпус текста, выделив в нем всю необходимую информацию о частях речи, находящихся в тексте, и типах синтаксических отношений между ними [69].

*Все виды активности субъектов в данной информационной модели разбиты на определенные смысловые группы. В каждой такой группе, по аналогии с субъектами, выделен базовый вид деятельности, все остальные ему синонимичны* [69]. Концептуальная информационная модель для одной группы видов деятельности при ликвидации последствий ЧС представлена на рисунке 2.5.

Иначе ситуация обстоит с построением концептуальной информационной модели для объектов исследуемых процессов предупреждения возникновений и

ликвидации последствий чрезвычайных ситуаций. В этом случае, *в исследуемом пространстве зачастую будут формироваться устойчивые выражения (compound noun)*, моделирование которых требует проведения процедуры интеграции определенных информационных элементов [69]. Например, в случае, если некоторая природная или искусственно созданная деятельность привела к возникновению результата «обрушение здания» (collapse building), тогда для исследователя это словосочетание является устойчивым выражением, полученным на основе интеграции информационных элементов «обрушение» и «здание».

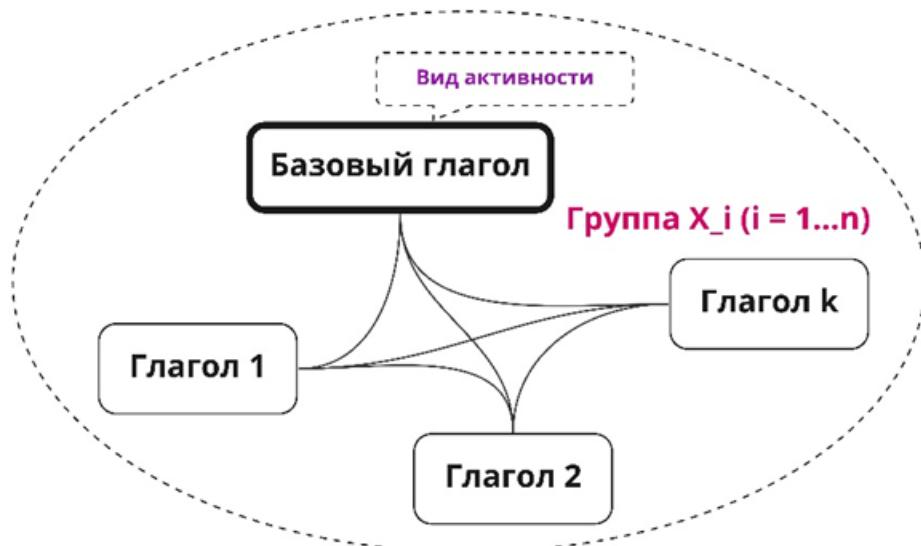


Рисунок 2.5 – Концептуальная информационная модель данных для одной группы видов деятельности

При этом, по умолчанию, это устойчивое выражение приобретает в модели значение константы, так как указанные выше возможные виды деятельности имеют также другую связь, например, с устойчивым выражением «повреждение здания» (damage building), которое тоже присутствует в информационной модели в виде постоянной вершины, но получено будет в результате интеграции других информационных элементов [69]. Для исключения конфликтных ситуаций при создании описаний объектов данный процесс организован в двух уровнях, как это проиллюстрировано на рисунке 2.6.

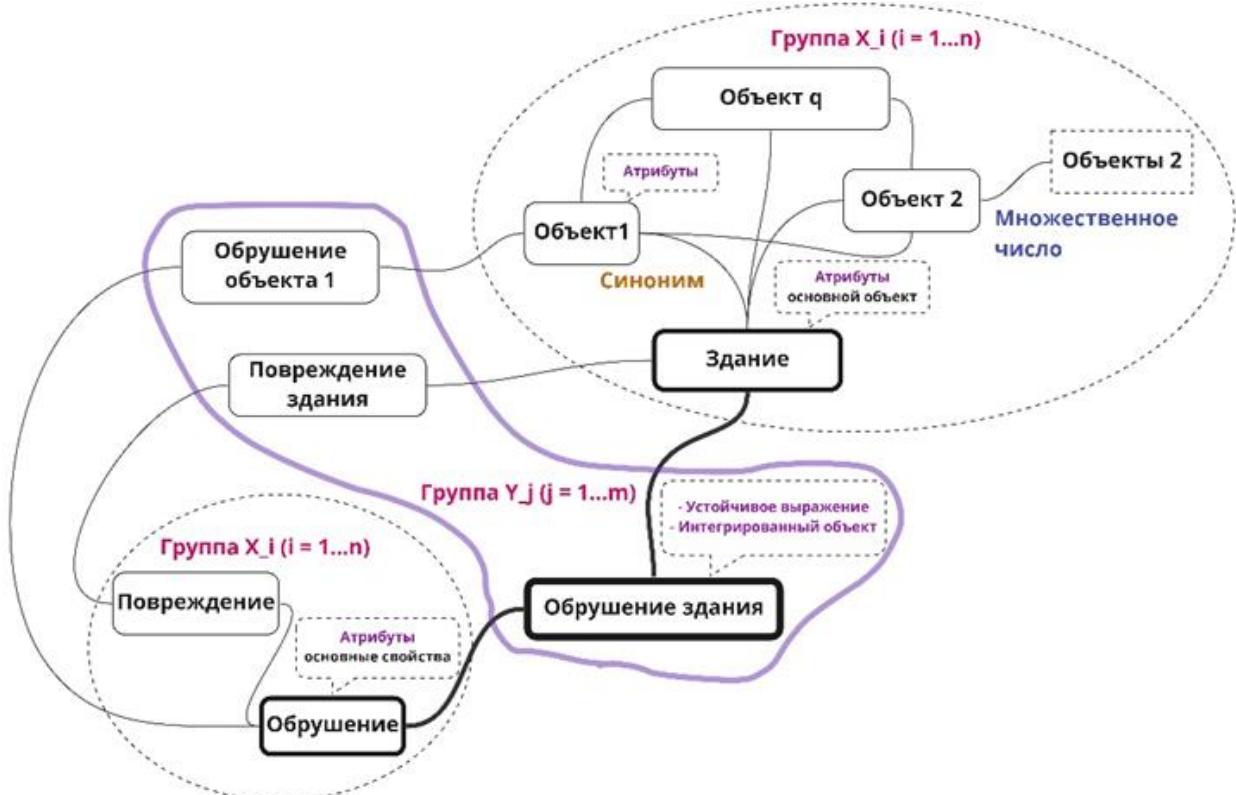


Рисунок 2.6 – Концептуальная информационная модель объектов исследуемых процессов

Очевидно, что все связи в представленных на рисунках 2.4-2.6 информационных моделях имеют *весовые коэффициенты*, заданные в виде функций принадлежности, для каждого из альтернативных вариантов построения триад «субъект – предикат – объект» [69]. Заданные функции принадлежности позволяют *перейти к модели векторного пространства*, в котором на основе, например, методов биоинспирированного поиска оценивается семантическая близость информационных элементов для построения альтернативных траекторий поддержки принятия решений.

Объединим описанные на рисунках 2.4-2.6 концептуальные информационные модели в нижнеуровневую онтологию знаний смыслового паттерна части информационного пространства для мониторинга чрезвычайных ситуаций, используемую при решении задач поддержки принятия решений. Проиллюстрируем полученную модель онтологии на рисунке 2.7.

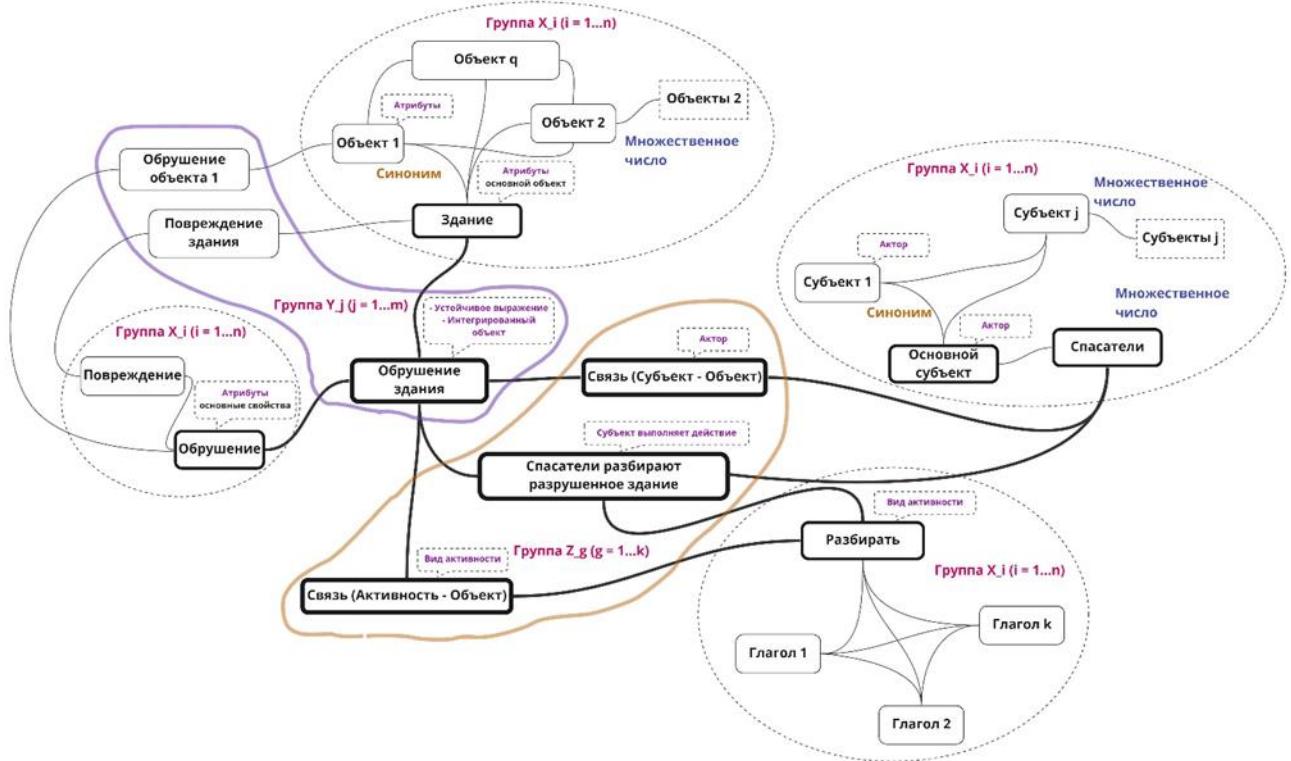


Рисунок 2.7 – Нижнеуровневая модель онтологии знаний смыслового паттерна информационного пространства для мониторинга чрезвычайных ситуаций

Представленный на рисунке 2.7 частный случай описывает версию того, как формируется паттерн решения «субъект – выполняет – действие» для класса чрезвычайной ситуации «обрушение здания» (collapse building), где результатом поддержки принятия решений является альтернатива – «спасатели разбирают разрушенное здание» (Rescuers remove collapsed building).

Одним из вариантов формализации подобной модели онтологии является переход к векторному представлению информационного пространства [69]. Это позволит решать на множестве элементов информации задачу их классификации для распределения по классам чрезвычайных ситуаций.

Как было показано в подразделе 2.1 данного раздела, имеющиеся в онтологии субъекты ( $Sbj$ ) и объекты ( $Obj$ ) наполняют множество концептов (понятий) ( $K_i$ ,  $i = 1\dots N$ , где  $N$  – количество концептов) с заданной на нём системой связей (бинарных отношений) ( $R_q$ ,  $q = 1\dots M$ , где  $M$  – количество связей), являющихся по своей сути несимметричными семантическими отношениями.

Элементам множества понятий (концептов)  $K$  ставится в соответствие набор векторов, значения компонентов которых определяют их атрибуты [9]:

$\beta_1 = \{\beta_{1i}\}, i = 1, \dots, N$  – вектор идентификаторов понятий;  $\beta_{1i}$  – идентификатор  $i$ -го понятия  $k_i$ ;

$\beta_2 = \{\beta_{2i}\}, i = 1, \dots, N$  – вектор названий понятий;  $\beta_{2i}$  – название  $i$ -го понятия  $k_i$ ;

$\beta_3 = \{\beta_{3i}\}, i = 1, \dots, N$  – вектор описания смысла понятий; где  $\beta_{3i}$  – описание  $i$ -го понятия  $k_i$ ;

$\beta_4 = \{\beta_{4i}\}, i = 1, \dots, N$  – вектор весов понятий; где  $\beta_{4i}$  – вес  $i$ -го понятия  $k_i$  в интервале  $(0,1]$ .

Веса понятий определяются как на основе экспертных оценок, так и частотных характеристик появления в информационных ресурсах исследуемых предметных областей, а также контекста исследуемого информационного процесса [9].

Определим виды отношений, объединяющих понятия (концепты) онтологии:

$$A = \{\alpha_l\}, l = 1, \dots, T_A. \quad (2.24)$$

Элементам множества  $A$  ставится в соответствие набор векторов, значения компонент которых определяют их атрибуты:

$\alpha_1 = \{\alpha_{1l}\}, l = 1, \dots, T_A$  – вектор идентификаторов типов отношений между понятиями;  $\alpha_{1l}$  – идентификатор  $l$ -го типа отношения  $\alpha_l$ ;

$\alpha_2 = \{\alpha_{2l}\}, l = 1, \dots, T_A$  – вектор описаний типов отношений между понятиями;  $\alpha_{2l}$  – описание  $l$ -го типа отношения  $\alpha_l$ ;

Тогда элементам множества  $R$  ставится в соответствие набор векторов, значения компонент которых определяют их атрибуты:

$\gamma_1 = \{\gamma_{1q}\}, q = 1, \dots, M$  – вектор идентификаторов связей между двумя связываемыми понятиями из множества  $P$ ;  $\gamma_{1q}$  – идентификатор  $q$ -ого отношения;

$\gamma_2 = \{\gamma_{2q}\}, q = 1, \dots, M$  – вектор идентификаторов типов отношений между понятиями  $K_{q1}$  и  $K_{q2}$  ( $K_{q1} \in K, K_{q2} \in K$ );  $\gamma_{2q}$  – идентификатор типа  $q$ -ого отношения, значение  $\gamma_{2q}$  совпадает с одним из значений компонентов вектора  $\alpha_1$ ;

$\gamma_3 = \{\gamma_{3q}\}, q = 1, \dots, M$  – вектор описаний связей между понятиями  $K_{q1}$  и  $K_{q2}$ ;  $\gamma_{3q}$  – описание  $q$ -ого отношения (дополняющее соответствующее описание  $\alpha_{2l}$ );

$\gamma_4 = \{\gamma_{4q}\}, q = 1, \dots, M, \gamma_5 = \{\gamma_{5q}\}, q = 1, \dots, M$  – векторы, компоненты которых задают идентификаторы, соответственно, первого и второго связываемых понятий  $K_{q1}$  и  $K_{q2}$ ; если отношение является направленным, то оно направлено от первого понятия ко второму;

$\gamma_6 = \{\gamma_{6q}\}, q = 1, \dots, M$  – вектор, компоненты которого задают направленную ( $\gamma_{6q} = 1$ ) или ненаправленную ( $\gamma_{6q} = 0$ ) связь между понятиями (связь направлена от понятия  $K_{q1}$  к понятию  $K_{q2}$ );

$\gamma_7 = \{\gamma_{7q}\}, q = 1, \dots, M$  – вектор весов отношений;  $\gamma_{7q}$  – вес  $q$ -ого отношения, в интервале  $(0,1]$ .

Веса отношений характеризуют их важность для определения тематики информационного процесса.

*Критерием оценки принадлежности* определенному классу является аргумент минимизации расстояния между элементами информации в векторном пространстве. Переход к векторному представлению осуществляется на основе применения открытых инструментов, например, *Word2Vec* [69].

Таким образом, в данном подразделе представлено нижнеуровневое описание модели онтологии знаний, применяемой при обработке и анализе текстов, отличающейся использованием структуры отношений между понятиями, детализирующими семантику текстовой информации, что позволяет получить набор смысловых паттернов, а также проводить оценку их семантической близости.

В следующем подразделе в целях дополнительной фильтрации и устранения шума в поступающей на вход системы искусственного интеллекта и машинного

обучения текстовой информации представлено описание разработки эвристических алгоритмов предварительной группировки предложений, имеющих схожие смысловые характеристики, а также определения последовательности обработки построенных групп предложений для упрощения последующих процедур обработки и анализа текстов на естественном языке.

### **2.3. Алгоритмы группировки предложений при обработке и анализе текстов**

Группировка предложений при обработке и анализе текстов необходима для повышения точности семантического поиска релевантных знаний. Это позволяет исключить один из основных недостатков систем генеративного искусственного интеллекта, который ограничивает одним единственным вариантом ответа на запрос (prompt) пользователя. В данном случае группировка предложений позволяет построить дерево решений, включающее в себя весь спектр смыслов, отфильтрованный по введенному ключевому слову и его синонимам. Такой грубый фильтр на основе группировки предложений создает подмножество смысла в виде отдельных ветвей дерева решений, что дает пользователю возможность выбрать конкретное направление семантического поиска знаний в соответствии с его собственными предпочтениями.

Представим описание эвристического алгоритма группировки предложений, позволяющего построить дерево решений при уточнении искомого смысла извлекаемых из текстовой информации знаний. Опишем уровни данного дерева. На самом верхнем (нулевом) уровне все предложения, включающие в себя поисковое ключевое слово (*SKW*), находятся в одной большой корневой вершине (группе) дерева. На первом уровне декомпозиции корневая вершина делится на ветви, формирующие группы по типам предложений, например, простое, сложносочиненное, сложноподчиненное, в которых обнаружено поисковое ключевое слово (*SKW*).

На втором уровне детализации, каждая верхняя группа разбивается на подгруппы по признаку того, в субъект, в действие, или в объект попало ключевое слово. На следующем (третьем) уровне, каждая такая подгруппа разбивается на несколько по нормализованным видам деятельности (по предикату) субъекта, которые являются либо результатом действия (делает что-то), либо его состоянием (является чем-то), либо инструментом (использует для чего-то) (рис. 2.8).

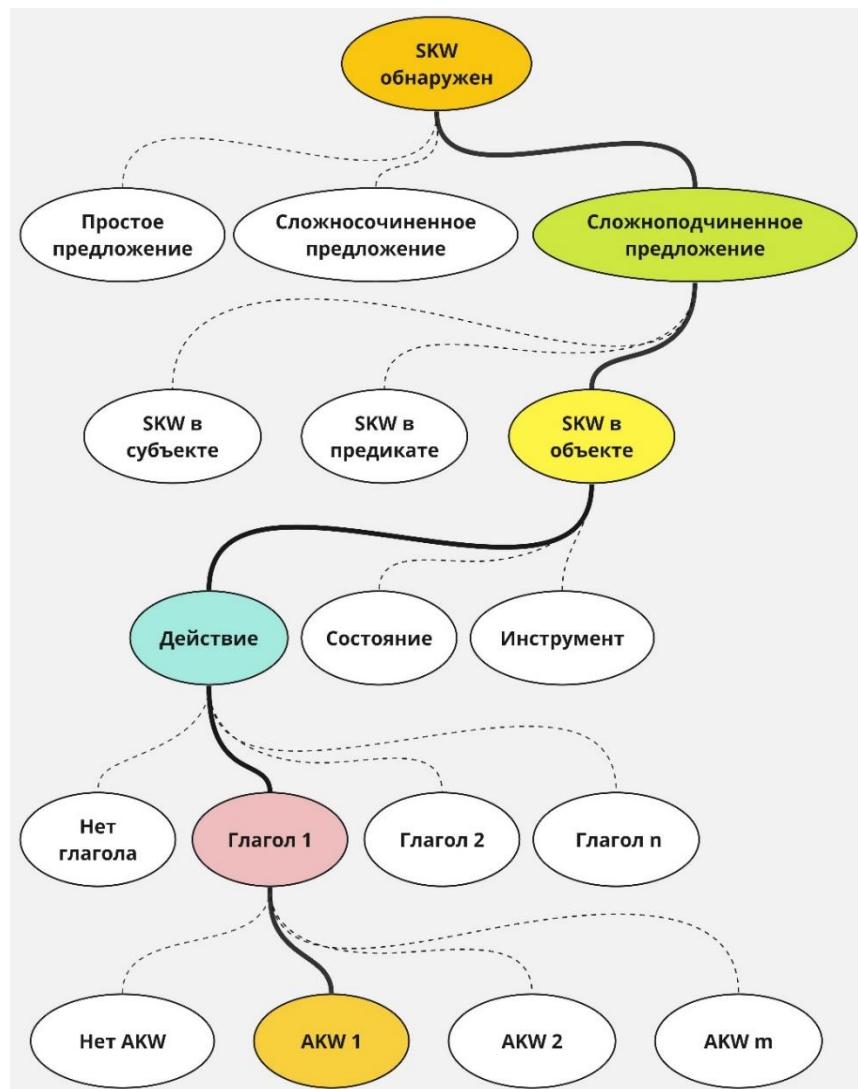


Рисунок 2.8 – Дерево решений при уточнении искового смысла извлекаемых из текстовой информации знаний

Оставшиеся четвертый и пятый уровни иерархии дерева решений группируют предложения по реальным имеющимся глаголам (verbs) и актуальным ключевым словам (AKW) соответственно. На представленном рисунке 2.8 в

абстрактом примере дерева выделена одна из множества возможных траекторий уточнения искомого смысла извлекаемых из текстовой информации знаний.

На первом шаге работы эвристического алгоритма группировки предложений происходит выбор уровня дерева решений  $X$ . После чего поступившие на вход предложения  $S_j$  проверяются на соответствие критериям отбора в группы  $G_i$  выбранного уровня (рис. 2.9).

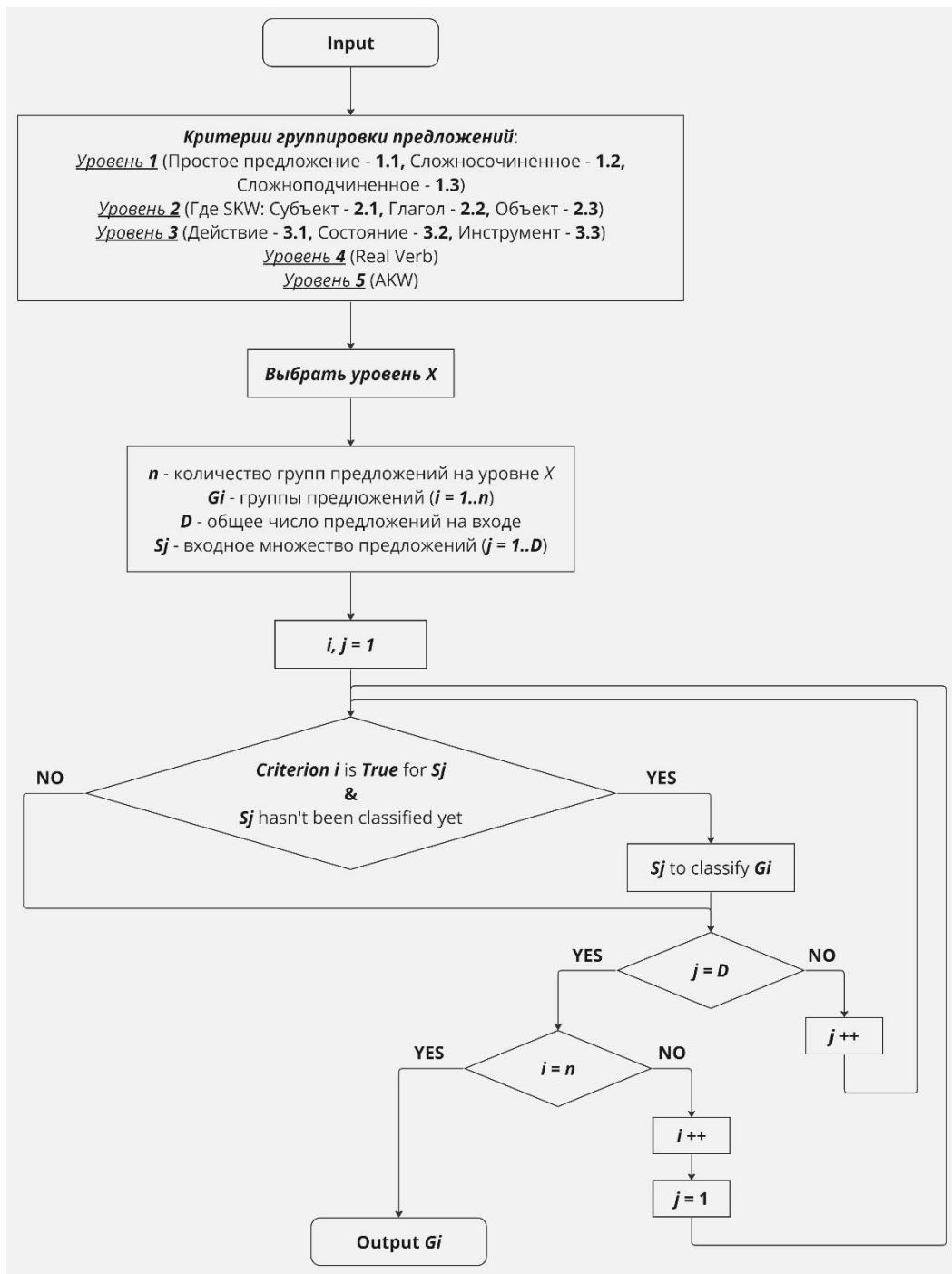


Рисунок 2.9 – Эвристический алгоритм группировки предложений

В случае, если анализируемое предложение еще не было классифицировано и соответствует критерию классификации в рассматриваемую группу, данное предложение включается в состав этой группы. Далее проверяются условия останова алгоритм. Если они не достигнуты, тогда происходит увеличение значений счетчиков предложений и групп, после чего процесс классификации повторяется (рис. 2.9). После достижения условия останова алгоритма для текущего уровня  $X$ , происходит переход на следующий уровень дерева решений, и весь процесс повторяется снова в следующей итерации.

Для упорядочивания полученных групп предложений написан эвристический алгоритм, позволяющий выстроить последовательность групп в порядке убывания в них общего числа предложений (рис. 2.10).

В начале работы алгоритма значения всех счётчиков  $i, j$ , у приравниваются к единице, а значению максимального числа предложений в группе  $Max$  присваивается значение первого по порядку элемента из массива  $X[i]$ , являющееся числом предложений  $Q_i$  в группе  $G_i$  (рис. 2.10). Затем проверяется выполнение условия  $X[i+1] > Max$ , если «да», тогда значение  $X[i+1]$  присваивается переменной  $Max = X[i+1]$ . После чего происходит увеличение значения счётчика  $i$  на единицу  $i = i + 1$  и процедура поиска элемента массива  $X[i]$ , имеющего максимальное значение, повторяется до тех пор, пока не пройдут сравнение все элементы этого массива. Данный цикл заканчивается при достижении счётчиком  $i$  максимального значения числа групп предложений  $i = m$  (рис. 2.10).

По окончании цикла порядковый номер группы  $G_i$ , имеющей максимальное число предложений  $Max = Q_i$ , присваивается переменной  $y$ , которая запоминается в упорядоченном массиве  $Z[j] = y$ . Группа  $G_y$  исключается из дальнейшего рассмотрения ( $X[y] = 0$ ). После чего алгоритм через обратную связь переходит на новую итерацию ( $j = j + 1; i = 1$ ), и все описанные выше действия повторяются снова. Алгоритм продолжает свою работу пока не будут упорядочены по убыванию числа предложений номера всех групп в массиве  $Z[j]$ , то есть алгоритм остановится при  $j = m$  (рис. 2.10). Результатом работы алгоритма является массив  $Z[j]$ ,

содержащий последовательность номеров групп, упорядоченную по убыванию числа предложений.

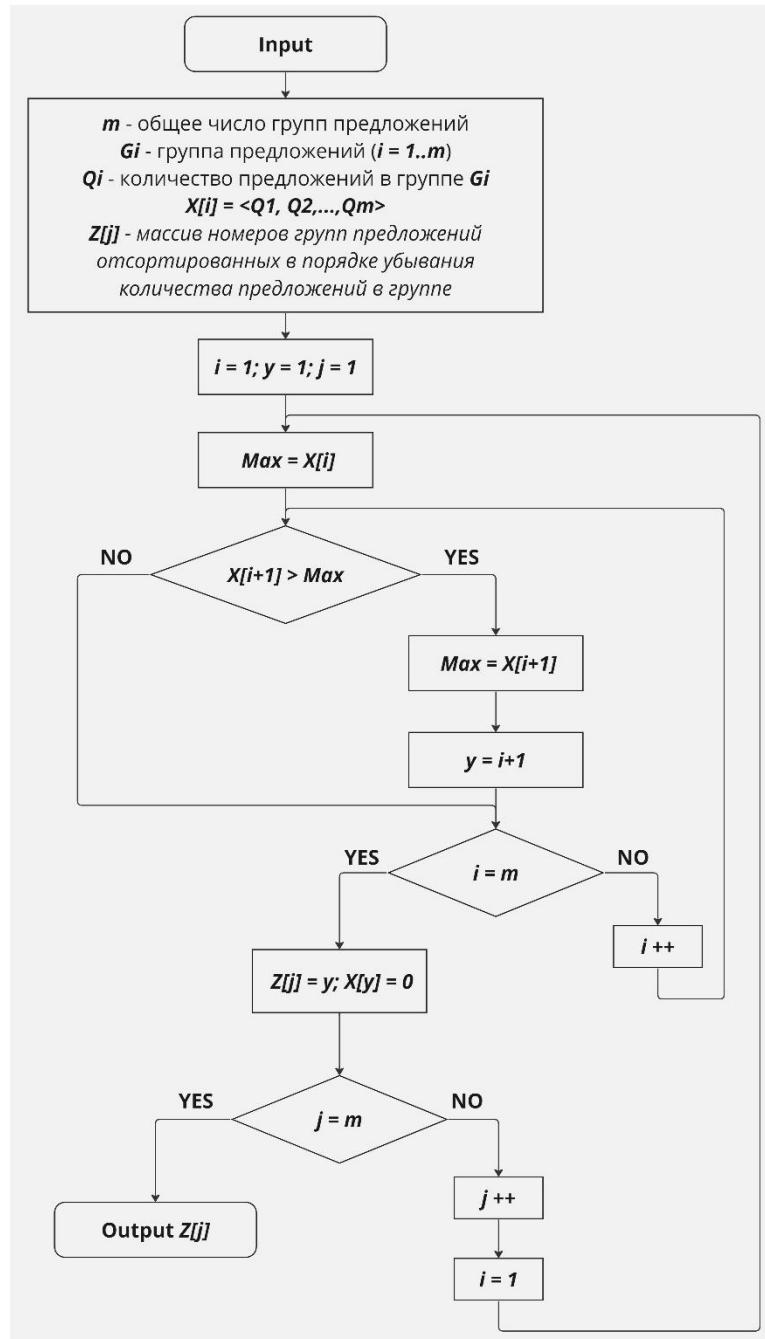


Рисунок 2.10 – Эвристический алгоритм упорядочивания групп предложений

Предложенный эвристический алгоритм дает возможность в процессе семантического поиска выбирать на обработку сначала только самые крупные группы предложений, что позволяет сократить время на получение результата решения данной задачи.

В данном подразделе в целях дополнительной фильтрации и устранения шума в поступающей на вход системы искусственного интеллекта и машинного обучения текстовой информации представлено описание разработки эвристических алгоритмов предварительной группировки предложений, имеющих схожие смысловые характеристики, а также определения последовательности обработки построенных групп предложений для упрощения и ускорения последующих процедур обработки и анализа текстов на естественном языке.

#### **2.4. Выводы по разделу**

Второй раздел посвящен созданию верхнеуровневого и нижнеуровневого описаний моделей онтологии знаний, применяемых при обработке и анализе текстов на естественном языке. Помимо этого, в данном разделе представлено описание разработки эвристических алгоритмов предварительной группировки предложений, имеющих схожие смысловые характеристики, а также определения последовательности обработки построенных групп предложений для упрощения последующих процедур анализа текстовой информации.

Построена верхнеуровневая модель онтологии знаний, которая отличается включением в состав ее компонентов множеств понятий с различным уровнем нормализации, что позволяет обеспечить необходимую степень детализации анализируемой текстовой информации.

Построена нижнеуровневая модель онтологии знаний, которая отличается использованием структуры отношений между понятиями, детализирующими семантику текстовой информации, что позволяет получить набор смысловых паттернов, а также проводить оценку их семантической близости.

В целях дополнительной фильтрации и устранения шума в поступающей на вход системы искусственного интеллекта и машинного обучения текстовой информации представлено описание разработки эвристических алгоритмов предварительной группировки предложений, имеющих схожие смысловые

характеристики, а также определения последовательности обработки построенных групп предложений для упрощения и ускорения последующих процедур обработки и анализа текстов на естественном языке.

Третий раздел диссертации посвящен развитию полученных результатов и содержит в себе описание разработки алгоритмов поиска и приобретения знаний в текстах, а также использования приобретенных знаний, что позволяет извлекать смысловую часть предложения из полученной синтаксической схемы текстовой информации и определять гранулы смысла, а также проводить интенсификацию и диверсификацию поисковых процедур для уменьшения времени отклика системы искусственного интеллекта и машинного обучения на пользовательский запрос при обработке и анализе текстов на естественном языке.

### **3. АЛГОРИТМЫ ПОИСКА, ПРИОБРЕТЕНИЯ И ИСПОЛЬЗОВАНИЯ ЗНАНИЙ ПРИ ОБРАБОТКЕ И АНАЛИЗЕ ТЕКСТОВ**

Данный раздел посвящен разработке алгоритма поиска знаний в текстах на естественном языке для создания дополнительного фильтра на выходе парсера с применением графовых моделей, который позволяет извлечь смысловую часть предложения из полученной синтаксической схемы текстовой информации для использования в процессах приобретения знаний. Разработан алгоритм приобретения знаний в текстах на естественном языке с применением оригинального множества низкоуровневых правил семантического анализа полученных смысловых паттернов, позволяющий находить основные гранулы смысла для процессов использования знаний. Разработан модифицированный биоинспирированный алгоритм использования приобретенных знаний в задачах генеративного искусственного интеллекта с применением улучшенных механизмов интенсификации поиска решений и процедур выхода из локальных оптимумов, позволяющий уменьшить время отклика на пользовательский запрос системы искусственного интеллекта и машинного обучения при обработке и анализе текстов на естественном языке.

#### **3.1. Разработка алгоритма поиска знаний в текстах на естественном языке с применением графовых моделей**

Поиск знаний в текстах на естественном языке напрямую связан с процессами автоматической обработки и анализа текстовой информации. Это одно из наиболее быстро развивающихся направлений искусственного интеллекта [71, 72]. В системах искусственного интеллекта и машинного обучения обработки и анализа текстов на естественном языке успешно применяются методы теории

графов. Графовые модели позволяют получить структурированную, компактную и формализованную форму описания текстовой информации, которая упрощает процесс поиска знаний. В данном случае, граф – математический объект, который описывает структуру отношений между словами в предложении.

*Основная гипотеза* заключается в том, что графовое представление используется в качестве инструмента для сокращения и упрощения предложения за счет исключения "мусорных" (бессмысленных) слов и нахождения прямых отношений между словами – носителями смысла. Для этого используются процедуры (эвристики) восстановления субъекта, нормализации глагола и объекта, определения ключевых слов с учетом домена, определения кратчайшего пути между вершинами графа, исключения инвариантных к смыслу слов на основе удаления «висячих», «туниковых» и «изолированных» вершин, а также учета мощности окрестности соответствующих вершин графа и списка стоп-слов.

Исследуем работу предлагаемого алгоритма поиска знаний на основе обработки и анализа следующего предложения: «Нашей миссией является задача, помочь вам ориентироваться в постоянно растущем множестве правил безопасности». Для построения графовой модели этого предложения был выбран ациклический ориентированный граф. Данный выбор сделан на основе анализа синтаксической схемы указанного предложения на выходе парсера (рис. 3.1).

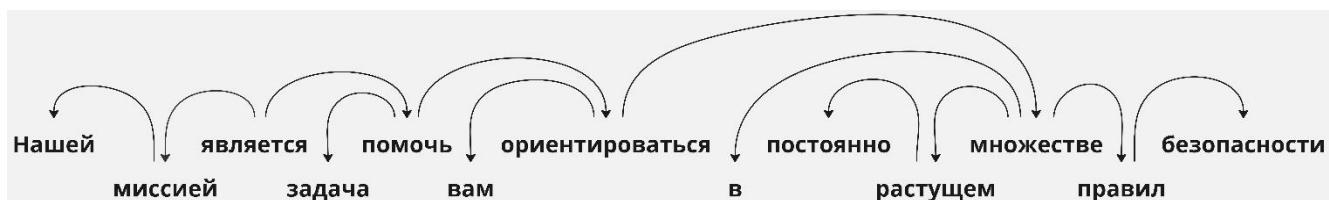


Рисунок 3.1 – Синтаксическая схема исследуемого предложения на выходе парсера

Отметим, что результат работы автоматического парсера при обработке сложных предложений не точен на 100% в большинстве случаев. Преимуществом предложенного автором алгоритма поиска знаний в текстах является то, что данный недостаток парсера с высокой вероятностью не отражается на качестве

последующего семантического анализа. Это происходит потому, что предложенный алгоритм обладает определенной гибкостью и при обработке построенной графовой модели предложения позволяет исключать из рассмотрения синтаксические связи, построенные с ошибками.

Примем порядковые номера слов в анализируемом предложении в качестве номеров соответствующих вершин графа. Графовая модель указанного предложения представлена на рисунке 3.2.

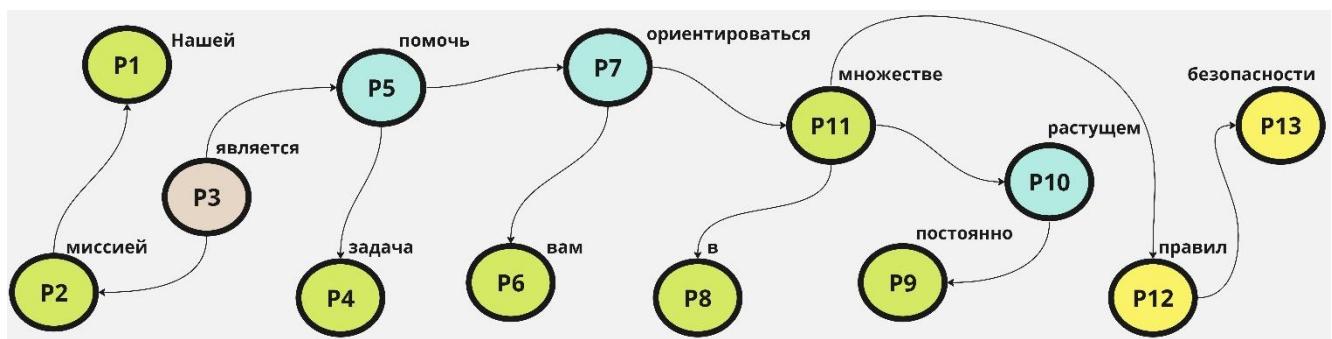


Рисунок 3.2 – Ациклический ориентированный граф, моделирующий систему отношений между словами анализируемого предложения

В построенной графовой модели (рис. 3.2) бежевым цветом фона выделена корневая (root) глагольная вершина  $P_3$  «является», которая определена как источник всех связей в графе, так как не имеет входных ребер. Голубым цветом фона выделены остальные глагольные вершины  $P_5$ ,  $P_7$  и  $P_{10}$ , включающие в себя глаголы: «помочь» и «ориентироваться», а также особую форму глагола – причастие «растущем».

Также внимание уделено определению (заданию) ключевой фразы (объекта), которая в соответствии с тематикой домена (предметной области) заключена в словосочетании «правила безопасности». Данная ключевая фраза является объектом смысла, заложенного в предложении, и находится в вершинах  $P_{12}$  и  $P_{13}$ , выделенных на рисунке 3.2 желтым цветом фона. Определение ключевой фразы или ключевого слова позволяет извлечь один из основных элементов смысла – объект (patient) действия, описанного в предложении [73, 74]. Другими элементами

смысла являются субъект (agent), осуществляющий действие, а также само действие – глагол (predicate).

Анализируемое предложение является сложным для автоматического поиска смысла, так как имеет в своем составе несколько действий, что требует проведения процедуры нормализации глаголов, несущих основную смысловую нагрузку. Помимо этого, данное предложение не имеет явно выраженного субъекта. Синтаксически выделенный субъект «наша миссия» является косвенным и не может быть интерпретирован в качестве элемента смысла (знания). Проведение процедур *нормализации глаголов и восстановления субъекта*, в данном случае, является первостепенной задачей, решаемой алгоритмом поиска знаний.

Для решения данных задач необходимо провести анализ всего текста (документа), в котором находится данное предложение, на предмет определения сегментов контекста [71, 75-77]. Автор предлагает представлять структуру (внутреннюю организацию) текста в виде информационного графа, содержащего верхнеуровневое описание контекста.

Известно, что, например, для восстановления субъекта (нахождения его именованной сущности) необходимо проанализировать первые предложения в документе, так как именно в них с наибольшей вероятностью будет найдено упоминание названия субъекта, а дальше по тексту в роли субъекта чаще используются общие слова, например, «мы», «они», «наши специалисты», «наша команда», «сотрудники компании» и т.п. Подобным образом проходит и восстановление основного объекта в анализируемом предложении, а общее описание целей и задач в информационном графе позволяет определить нормализованное значение глагола (предиката), которое будет отражать особенности имеющегося контекста.

Построим пример информационного графа для текста в предметной области «*предупреждение и/или ликвидация последствий чрезвычайных ситуаций (ЧС)*» (рис. 3.3). Очевидно, что структура текста по данной тематике на верхнем уровне в качестве корневой вершины имеет описание цели, связанной с деятельностью

Министерства Российской Федерации по делам гражданской обороны, чрезвычайным ситуациям и ликвидации последствий стихийных бедствий (МЧС России). Например, «*МЧС России повышает эффективность своей работы*» (вершина  $X_1$ ). Таким образом, получен *субъект* всех основных действий – «*МЧС России*» [78].

Вершинами, раскрывающими смысл основных задач, решение которых необходимо для достижения субъектом описанной цели, являются вершины:  $X_2$  – «*Миссия – передать населению знания о правилах безопасности*» (описание данной задачи позволяет на основе экспертной оценки выделить нормализованный глагол осуществляемого действия – «*узнать*» (передать знания), а также определить прямой объект действия – «*население*»);  $X_3$  – «*Повышать профессионализм работы подразделений*»;  $X_4$  – «*Решать задачи по ликвидации последствий ЧС*».

Вершинами сегментов основной содержательной части текста в информационном графе являются:  $X_5$  – «*Описание действий по снижению рисков возникновения ЧС*»;  $X_6$  – «*Описание действий по уменьшению ущерба от последствий ЧС*». Финальной вершиной графа обозначим  $X_7$  – «*Достижение безопасности жизнедеятельности*» (заключение) (рис. 3.3).

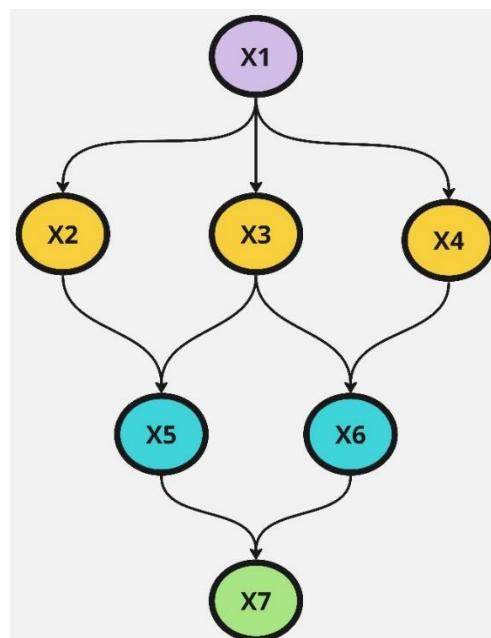


Рисунок 3.3 – Информационный граф структуры текстового документа, содержащий верхнеуровневое описание контекста

Вершины представленного на рисунке 3.3 информационного графа разбиты на разные категории с помощью отношений порядка, которые задают процесс движения информации, формируя последовательность тактов. Порядок вершины  $X_i$  – значение максимальной длины пути в данную вершину из начальной вершины  $X_1$ . В представленном на рисунке 3.3 примере информационного графа порядок вершин  $X_2, X_3, X_4$  равен 1 (первый такт), порядок  $X_5, X_6$  – 2 (второй такт),  $X_7$  – 3 (третий такт). Максимальное число тактов, необходимое для полной обработки информации – порядок информационного графа (данный пример информационного графа соответственно имеет порядок 3).

Для получения информации о структуре анализируемого текста построена матрица смежности данного информационного графа и найдены 2-я, 3-я и 4-я ее степени:

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad A^2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad A^3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$A^4 = 0.$$

Из построенной матрицы смежности и ее степеней извлекаются сведения о структуре текста. Например, вершина  $X_j$  является начальной в информационном графе, если  $j$ -ый столбец содержит одни нули. Подобным образом,  $X_i$  является заключительной вершиной в информационном графе, если  $i$ -ая строка содержит одни нули.

Наивысший порядок для всех вершин принадлежит вершине  $X_j$  и всегда имеет значение меньше чем порядок матрицы. Любой сегмент текста перестает участвовать в обработке после такта, который определяется порядком соответствующей вершины графа. Номер такта, после которого сегмент  $X_i$  уже не

учитывается при анализе текста, равен максимальному из порядков вершин, отвечающих отличным от нуля элементам  $i$ -ой строки матрицы смежности  $A$  [71].

Отметим, что в приведенном примере порядок графа равен 3, так как  $A \neq 0$ ;  $A^2 \neq 0$ ;  $A^3 \neq 0$ , но уже  $A^4 = 0$ . Данный вывод согласуется с ранее полученными результатами. Порядки вершин  $X_5$  и  $X_6$  равны 2, потому что данные вершины соответствуют отличным от нуля элементам столбцов матрицы  $A^2$ . Очевидно, что порядок вершины  $X_7$  равен 3, т.к. в матрице  $A^3$  в 7-м столбце имеется отличный от 0 элемент.

Построим для последующего анализа матрицу  $B = A + A^2 + \dots + A^n$ , где  $n$  – порядок матрицы  $A$ :

$$B = \begin{pmatrix} 0 & 1 & 1 & 1 & 2 & 2 & 4 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Значения элементов построенной матрицы  $B$  позволяют определить число существующих путей между вершинами информационного графа, представленного на рисунке 3.3. Так, например, элемент  $b_{17} = 4$  дает информацию о том, что в данном графе есть 4 варианта путей, которые ведут из  $X_1$  в  $X_7$ . В то же время, из  $X_3$  в  $X_7$  есть только два пути, так как  $b_{37} = 2$ . Элемент  $b_{26} = 0$  указывает на то, что из  $X_2$  в  $X_6$  вообще нет путей и т.д. Подобным образом извлекается полная информация о путях в исследуемом графе.

Элементы  $j$ -го столбца матрицы  $B$ , имеющие отличные от нуля значения, указывают на вершины, формирующие результат  $X_j$ . Таким образом, порядковые номера элементов  $j$ -го столбца, имеющие отличные от нуля значения, равны номерам вершин, формирующих следующий результат  $X_j$ . Тогда в исследуемом графе получаем, что, например, в формировании  $X_2$ , а также  $X_3$  и  $X_4$  принимает участие только  $X_1$ . В формировании  $X_5$  участвуют  $X_1$ ,  $X_2$  и  $X_3$ , а в формировании  $X_6$  участвуют  $X_1$ ,  $X_3$  и  $X_4$ . В формировании результата  $X_7$  принимают участие все

остальные вершины графа, а именно  $X_1, X_2, X_3, X_4, X_5$  и  $X_6$ . При решении прикладных задач на информационных графах, если нарушена логическая связь при построении какого-либо сегмента, ошибка обнаруживается с помощью применения данной методики.

Элементы  $i$ -ой строки матрицы  $B$ , имеющие отличные от нуля значения, номерами столбцов, в которых данные элементы находятся, указывают на результаты, сформированные при участии  $X_i$ , т.е. эти номера отличных от нуля элементов  $i$ -ой строки равны номерам вершин, в формировании которых была задействована вершина  $X_i$ .

Так, например, первая строка матрицы  $B$  имеет вид, указывающий на участие вершины  $X_1$  в последующем формировании всех остальных вершин  $X_2, X_3, X_4, X_5, X_6$  и  $X_7$ .

	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$	$X_7$
$X_1$	0	1	1	1	2	2	4

Вторая строка указывает на то, что вершина  $X_2$  была задействована в формировании вершин  $X_5$  и  $X_7$ .

	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$	$X_7$
$X_2$	0	0	0	0	1	0	1

Третья строка указывает на то, что вершина  $X_3$  была задействована в формировании вершин  $X_5, X_6$  и  $X_7$ . И так далее...

	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$	$X_7$
$X_3$	0	0	0	0	1	1	2

Подобным образом анализируются все строки матрицы  $B$ . Проведенный анализ матричного представления информационного графа позволяет определить и распознать структуру текста в процессе поиска знаний.

Проведем анализ и обработку ациклического ориентированного графа (рис. 3.2) на основе экспертной оценки с целью упрощения анализируемого предложения, что позволяет построить последовательность действий (алгоритм)

для автоматического преобразования подобных графовых моделей при решении задачи поиска знаний в текстах на естественном языке.

На начальном этапе обработки графовой модели предложения задается ключевое слово или фраза, отражающие основной смысл этой текстовой информации в контексте всего документа. В каждой предметной области (домене) формируется свой список ключевых слов (фраз). По сути, ключевая фраза является запросом пользователя. Очевидно, что в рассматриваемом примере ключевой является устоявшаяся фраза *«правила безопасности»* (вершины  $P_{12}$  и  $P_{13}$  объединяются в вершину  $P_{kw}$ ).

Используя представленный на рисунке 3.3 информационный граф структуры текстового документа, содержащий верхнеуровневое описание контекста, восстанавливаются субъект и объект – *«МЧС России»* и *«население»* соответственно. Таким образом, вершины синтаксически определенного субъекта *«наша миссия»*  $P_1$  и  $P_2$  заменяются вершиной  $P_{sbj}$  *«МЧС России»*, а вершина объекта *«вам»*  $P_6$  – вершиной  $P_{obj}$  *«население»*. Значение предиката для ключевой фразы в вершине  $P_7$  *«ориентироваться»* изменяется на определенное выше нормализованное значение глагола *«узнать»* (вершина получает название  $P_{NVerb}$ ).

Все «тупиковые» вершины ( $P_4$ ,  $P_8$ ,  $P_9$ ), не замененные ранее и/или не получившие собственных имен, удаляются из графа. Косвенным признаком других инвариантных к смыслу вершин, подлежащих удалению, служит низкое значение мощности окрестности (локальной степени) данной вершины. Учет мощности окрестности вершины является свободным параметром алгоритма, использование которого требуется при обработке сложных предложений, т.е тогда, когда исключение только висячих вершин должным образом не упрощает структуру текста.

«Висячая» корневая вершина  $P_3$ , имеющая только выходные ребра, является последовательным отношением между субъектом *«МЧС России»* и предикатом *«помогает»*. При упрощении структуры предложения подобная вершина заменяется ребром.

После всех описанных преобразований на выходе алгоритма граф анализируемого предложения приобретает вид, представленный на рисунке 3.4.

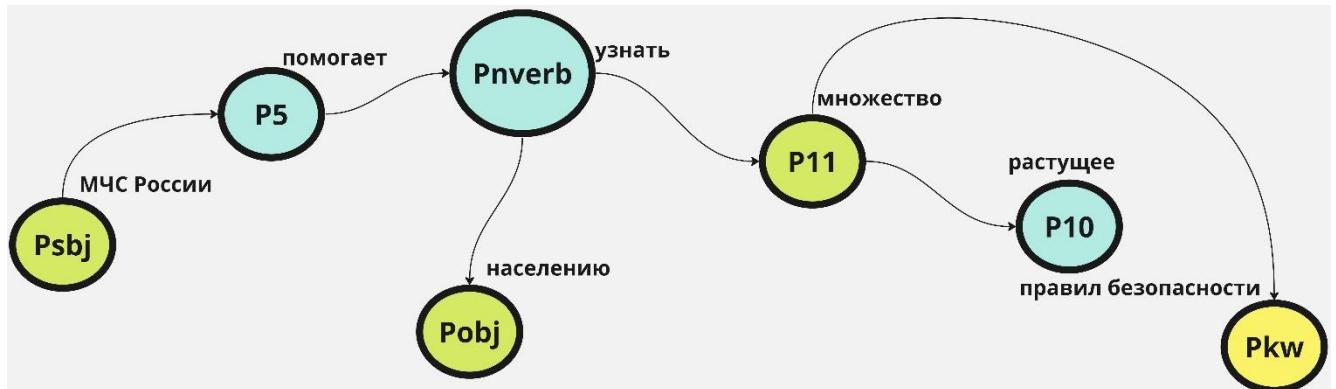


Рисунок 3.4 – Упрощенный ациклический ориентированный граф, моделирующий систему отношений между элементами и компонентами смысла

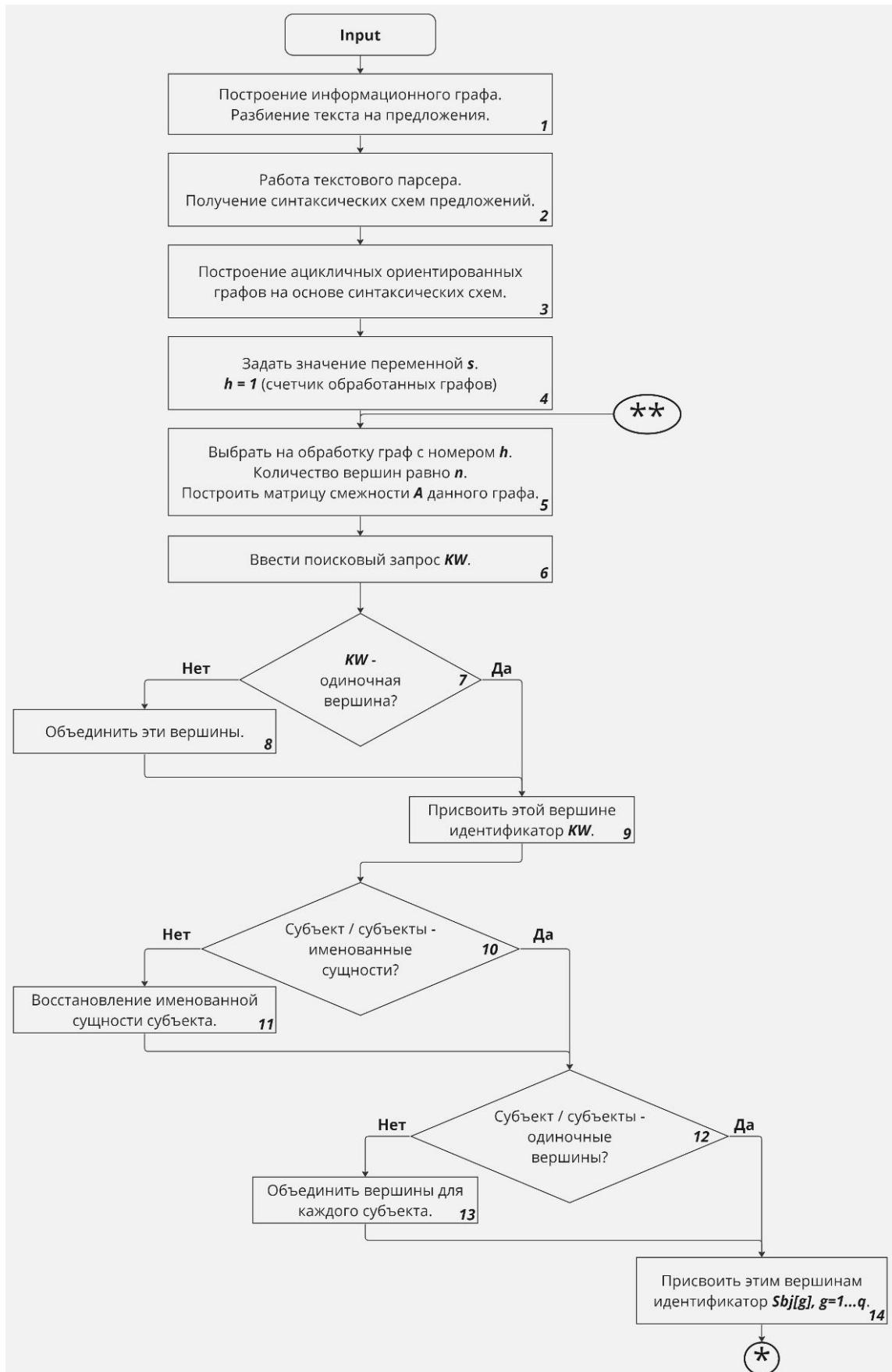
Таким образом, предложенный алгоритм поиска знаний в текстах на естественном языке с применением графовых моделей представлен следующей последовательностью действий:

1. *Начало алгоритма.* Построить информационный граф структуры текстового документа, содержащий верхнеуровневое описание контекста. Разбить исследуемый текст (документ) на предложения (стандартная процедура, реализуемая текстом процессором);
2. Обработать полученные предложения с помощью текстового парсера, получить на выходе парсера синтаксические схемы предложений;
3. Построить ациклические ориентированные графы, моделирующие системы отношений между словами внутри каждого предложения (граф строится на основе синтаксической схемы, где каждое слово становится вершиной, ребрами становятся связи между словами в синтаксической схеме предложения);
4. Задать значение переменной  $s$  – общее число предложений в исследуемом тексте (документе). Присвоить счетчику обработанных предложений  $h$  значение 1 ( $h = 1$ );
5. Приступить к обработке ациклического ориентированного графа (далее *граф*) с порядковым номером  $h$ . Построить матрицу смежности графа  $A = \|a_{ij}\|$ .

6. Ввести поисковый запрос в виде ключевого слова / фразы (*Key Word, KW*).
7. Если *KW* является одиночной вершиной графа, тогда перейти к п. 9;
8. Объединить все вершины графа, включающие в себя ключевую фразу, в одну вершину.
9. Присвоить вершине, включающей в себя ключевое слово, идентификатор *KW*;
10. Субъект / субъекты предложения, определенные синтаксической схемой на выходе парсера, являются именованной сущностью? Если «да», тогда перейти к п. 12;
11. Провести процедуру восстановления именованной сущности субъекта / субъектов предложения на основе сведений из информационного графа структуры текстового документа;
12. Если субъект / субъекты предложения являются одиночными вершинами графа, тогда перейти к п. 14;
13. Объединить все вершины графа, включающие в себя конкретный субъект / субъекты, в одну вершину для каждого субъекта предложения.
14. Присвоить этим вершинам идентификатор  $Sbj_g$ ,  $g = \overline{1, q}$ , где  $q$  – порядковый номер субъекта в предложении. Провести нормализацию значений субъекта / субъектов;
15. Если объект / объекты предложения, без учета *KW*, являются одиночными вершинами графа, тогда перейти к п. 17;
16. Объединить все вершины графа, включающие в себя конкретный объект / объекты, в одну вершину для каждого объекта предложения.
17. Присвоить этим вершинам идентификатор  $Obj_o$ ,  $o = \overline{1, d}$ , где  $d$  – порядковый номер объекта в предложении. Провести нормализацию (восстановление) значений объекта / объектов;
18. Провести нормализацию значений глагола / глаголов. Присвоить всем вершинам включающим в себя глаголы (предикаты) идентификатор  $NVerb_v$ ,  $v = \overline{1, w}$ , где  $w$  – порядковый номер глагола в предложении;

19. Задать значение порядкового номера вершины  $k = 1$ ;
20. Если  $k > n$ , тогда перейти к п. 24;
21. Если  $a^k = 0$ , тогда перейти к п. 23;
22.  $k = k + 1$ . Возврат к п. 20;
23. Исключить из рассмотрения и заменить ребром в графе «висячую» вершину с номером  $k$ , в том случае, если она не имеет идентификатора. Возврат к п. 22;
24. Задать значение порядкового номера вершины  $k = 1$ ;
25. Если  $k > n$ , тогда перейти к п. 29;
26. Если  $a_k = 0$ , тогда перейти к п. 28;
27.  $k = k + 1$ . Возврат к п. 25;
28. Исключить в графе из рассмотрения вместе с входящим ребром «тупиковую» вершину с номером  $k$ , в том случае, если она на имеет идентификатора. Возврат к п. 27;
29. На выходе алгоритма поиска знаний получить упрощенный ациклический ориентированный граф, моделирующий систему отношений между элементами и компонентами смысла в исследуемом предложении. Построить матрицу смежности полученного упрощенного графа  $A' = \|a'_{ij}\|$ .
30.  $h = h + 1$ ;
31. Если  $h \leq s$ , Тогда вернутся к п. 5;
32. *Окончание алгоритма.* Вывод результатов поиска знаний при обработке и анализе текста (документа).

Представим укрупненную схему разработанного алгоритма поиска знаний в текстах на естественном языке с применением графовых моделей на рисунке 3.5.



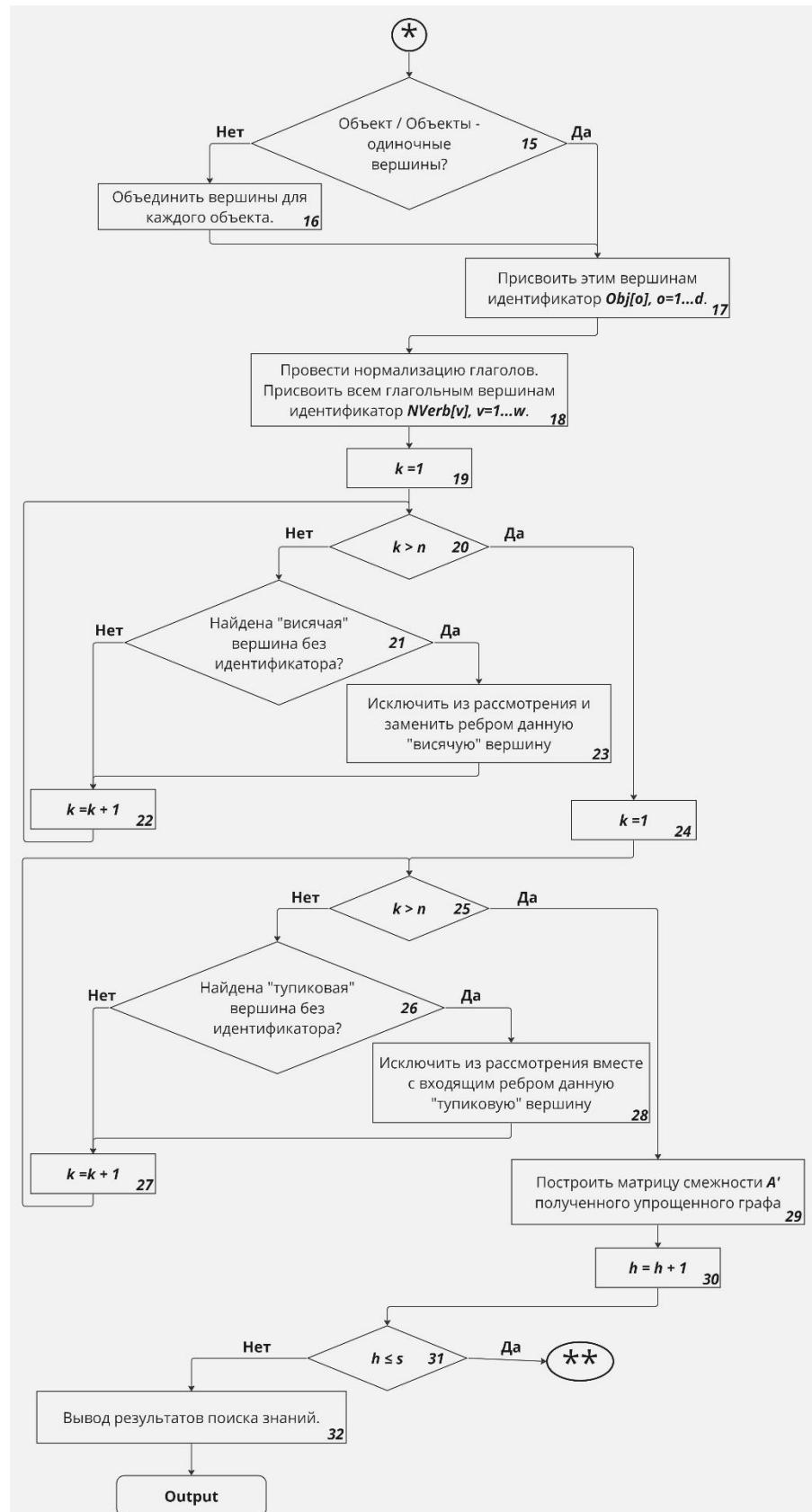


Рисунок 3.5 – Укрупненная схема алгоритма поиска знаний в текстах на естественном языке с применением графовых моделей

Отметим, что содержательная и структурная сущность текста зависит от таких его конструктивных признаков как целостность и связность, которые так же определяются на основе анализа матрицы смежности графа, описывающего структуру отношений между словами в предложении. Задачи выявления и исключения из анализа «изолированных», «висячих» и «тупиковых» вершин имеют особое важное значение при обработке текстов на естественном языке. «Изолированные» вершины не инцидентны ни одному из ребер графа, что указывает на то, что данная вершина не имеет связей с другими вершинами в графе. «Висячие» вершины инцидентны только выходящим из них ребрам (дугам). «Тупиковые» вершины, наоборот, инцидентны только входящим в них ребрам [71].

При анализе особенностей работы текстового парсера установлено, что в графе, задающем структуру отношений между словами в предложении (рис. 3.2), будет только одна «висячая» начальная вершина  $P_3$ , являющаяся корневой (обычно это глагол в основной части предложения), а также целый ряд «тупиковых» вершин ( $P_1, P_4, P_6, P_8, P_9, P_{13}$ ), не имеющих продолжения пути. Автор исследовал графовые модели текстов без «изолированных» вершин, так как их наличие указывает на отсутствие связности и целостности текста, что является частным случаем и представляет ограниченный интерес при решении задач поиска знаний на основе применения интеллектуальных информационных систем. Исследуемые графовые модели текстовой информации не являются избыточными по количеству связей между вершинами, так как любые две вершины имеют только одно общее ребро.

Поиск «изолированных», «висячих» и «тупиковых» вершин происходит на основе анализа матрицы смежности графа  $A = \|a_{ij}\|$ , где для всех вершин  $k$  ( $k = \overline{1, n}$ ,  $n$  – общее число вершин в графе) находится вектор  $a(k) = (a_k, a^k)$  со следующими компонентами:

$$a_k = \sum_{j=1}^n a_{kj}, a^k = \sum_{i=1}^n a_{ik}, \quad (3.1)$$

где  $a_k$  – сумма элементов  $k$ -ой строки матрицы смежности,  $a^k$  – сумма элементов  $k$ -го столбца матрицы смежности. При этом, выходящие из вершины  $k$  дуги,

определяются значением  $a_k$ , а входящие в нее дуги – значением  $a^k$ . При  $a_k = a^k = 0$ , вершина  $k$  является «изолированной», при  $a_k = 0$  – «тупиковой», а при  $a^k = 0$  – «висячей» [71, 77], как это показано в матрице смежности (табл. 3.1) ациклического ориентированного графа, моделирующего систему отношений между словами анализируемого предложения (рис. 3.2).

Таблица 3.1 – Матрица смежности анализируемого ациклического ориентированного графа

	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$	$P_9$	$P_{10}$	$P_{11}$	$P_{12}$	$P_{13}$
$P_1$	0	0	0	0	0	0	0	0	0	0	0	0	0
$P_2$	1	0	0	0	0	0	0	0	0	0	0	0	0
$P_3$	0	1	0	0	1	0	0	0	0	0	0	0	0
$P_4$	0	0	0	0	0	0	0	0	0	0	0	0	0
$P_5$	0	0	0	1	0	0	1	0	0	0	0	0	0
$P_6$	0	0	0	0	0	0	0	0	0	0	0	0	0
$P_7$	0	0	0	0	0	1	0	0	0	0	1	0	0
$P_8$	0	0	0	0	0	0	0	0	0	0	0	0	0
$P_9$	0	0	0	0	0	0	0	0	0	0	0	0	0
$P_{10}$	0	0	0	0	0	0	0	0	1	0	0	0	0
$P_{11}$	0	0	0	0	0	0	0	1	0	1	0	1	0
$P_{12}$	0	0	0	0	0	0	0	0	0	0	0	0	1
$P_{13}$	0	0	0	0	0	0	0	0	0	0	0	0	0

В матрице смежности (табл. 3.1) нулевые строки указывают на порядковые номера «тупиковых» вершин ( $P_1, P_4, P_6, P_8, P_9, P_{13}$ ), а нулевой столбец указывает на порядковый номер «висячей» корневой вершины  $P_3$ .

На рисунке 3.6 проиллюстрирован результат работы предложенного алгоритма поиска знаний в текстах на естественном языке с применением графовых моделей на примере рассматриваемого в данном подразделе предложения: «*Нашей миссией является задача, помочь вам ориентироваться в постоянно растущем*

множестве правил безопасности». Как уже было показано выше, на выходе алгоритма данное предложение представлено в следующем упрощенном виде с сохранением и уточнением основного смыслового содержания: «МЧС России помогает населению, узнать расшущее множество правил безопасности».

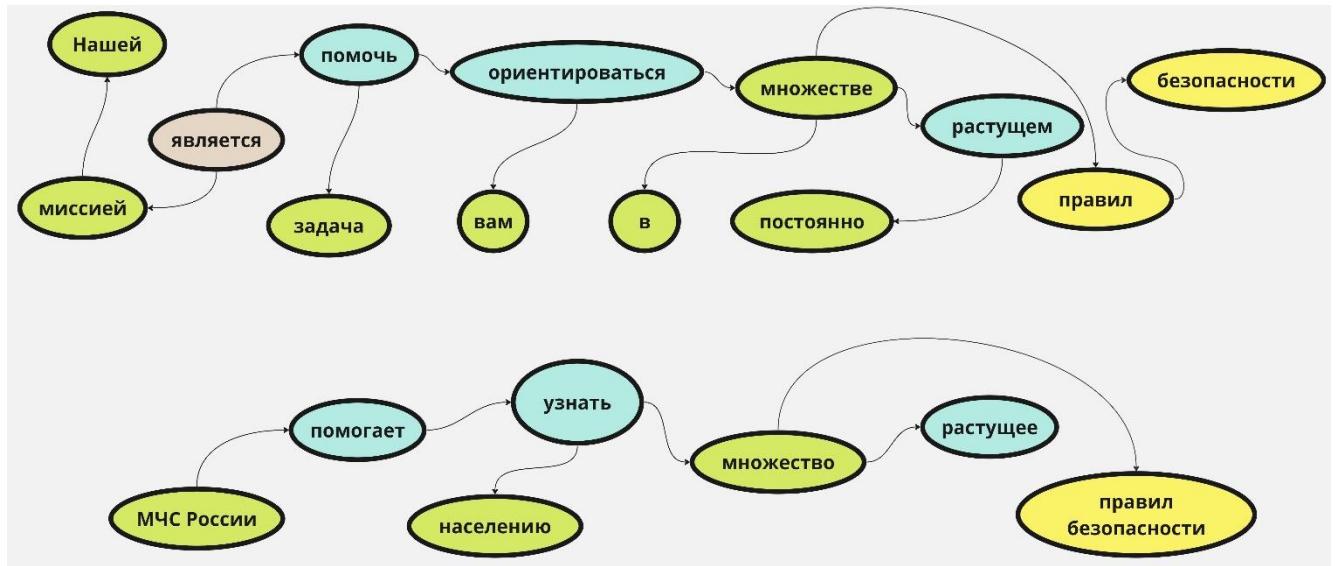


Рисунок 3.6 – Результат работы предложенного алгоритма поиска знаний в текстах на естественном языке с применением графовых моделей

Отметим, что описанный алгоритм полностью не упростил структуру в рассмотренном примере анализируемого предложения до уровня двух имеющихся компонентов смысла: «МЧС России помогает населению» и «Население узнает правила безопасности», но в целом проведенная обработка позволила отфильтровать основные избыточные вершины и восстановить все элементы смысла: «МЧС России»; «помогает»; «населению»; «узнать»; «правила безопасности». Оставшиеся инвариантные к смыслу вершины  $P_{10}$  «растущее» и  $P_{11}$  «множество» исключаются из рассмотрения алгоритмом приобретения знаний в текстах на естественном языке с применением множества низкоуровневых правил семантического анализа уже полученных смысловых паттернов, позволяющим построить основные гранулы смысла для процессов использования знаний. Разработка данного алгоритма описана в следующем подразделе диссертации.

### 3.2. Разработка алгоритма приобретения знаний в текстах на естественном языке с применением множества низкоуровневых правил

Для развития полученных ранее научных результатов автором разработан алгоритм приобретения знаний в текстах на естественном языке с применением множества низкоуровневых правил семантического анализа. Особенность данного алгоритма заключается в том, что он показывает высокую эффективность даже при отдельном использовании без применения описанного в предыдущем разделе алгоритма поиска знаний. Недостатком такого отдельного использования является низкая скорость работы алгоритма, так как его последовательность действий при обработке и анализе предложений близка к полному перебору. Поэтому наибольшую эффективность с точки зрения достижения баланса между качеством получаемых решений и скоростью работы предлагаемый алгоритм приобретения знаний показывает при обработке уже полученных на выходе алгоритма поиска знаний смысловых паттернов [68, 79-81], которые представлены в виде упрощенного ациклического ориентированного графа, моделирующего систему отношений между элементами и компонентами смысла. Пример такого графа представлен на рисунке 3.4.

Опишем разработку алгоритма приобретения знаний в текстах на естественном языке с применением множества низкоуровневых правил [82-84] на примере дальнейшей обработки и анализа результата работы предложенного в предыдущем подразделе алгоритма поиска знаний (рис. 3.5) в текстах на естественном языке с применением графовых моделей (рис. 3.6). Зададим последовательность шагов данного алгоритма:

*Начало алгоритма.*

1. Получить на вход смысловые паттерны (упрощенные предложения), построенные в результате работы предложенного в предыдущем подразделе алгоритма поиска знаний в текстах на естественном языке с применением графовых моделей.

Элементом начала поиска является вершина ключевой фразы  $P_i = P_{kw}$ . Сохраним порядковый номер  $k$  ключевой фразы в предложении  $k = i$ . Анализ синтаксической схемы предложений даёт основание полагать, что предикат (глагол), имеющий отношение к ключевой фразе чаще всего находится слева от нее, но только в том случае, если ключевая фраза определена как объект (patient) и имеет в своем составе смысловое существительное. Если ключевая фраза определена как субъект, тогда в большинстве случаев предикат (глагол) находится справа от нее. Также, если ключевая фраза попала в субъект, тогда в большинстве случаев движение влево вообще будет невозможным, так как основной субъект чаще всего находится в начале предложения, а вторичный субъект одновременно является объектом для предыдущей части предложения, что не противоречит поиску предиката (глагола) слева от него. На данных наблюдениях основана логика действий (передвижений) алгоритма при поиске гранул смысла в предложении. Частный случай (выброс), связанный с попаданием ключевой фразы в предикат, в данной работе не рассматривается из-за малой вероятности получения такой синтаксической схемы предложения в исследуемом домене (предметной области).

Таким образом, для сложноподчиненных предложений (части сложносочиненных предложений анализируются по отдельности, так как не имеют причинно-следственных связей) имеется несколько типовых структур (рис. 3.7), которые необходимо обработать в процессе приобретения знаний. Любая из этих структур предполагает первоначальную проверку условия того, что ключевая фраза не находится в начале предложения, указывая на основной субъект, в любом другом случае на этом шаге алгоритм реализует движение влево от ключевой фразы (рис. 3.8) до тех пор, пока не найдёт предикат (глагол). Вариант с отсутствием глагола исключается, так как предыдущий алгоритм поиска знаний на этапе нормализации восстанавливает предикат даже при отсутствии глагола в оригинальном предложении, поступившем на вход системы.

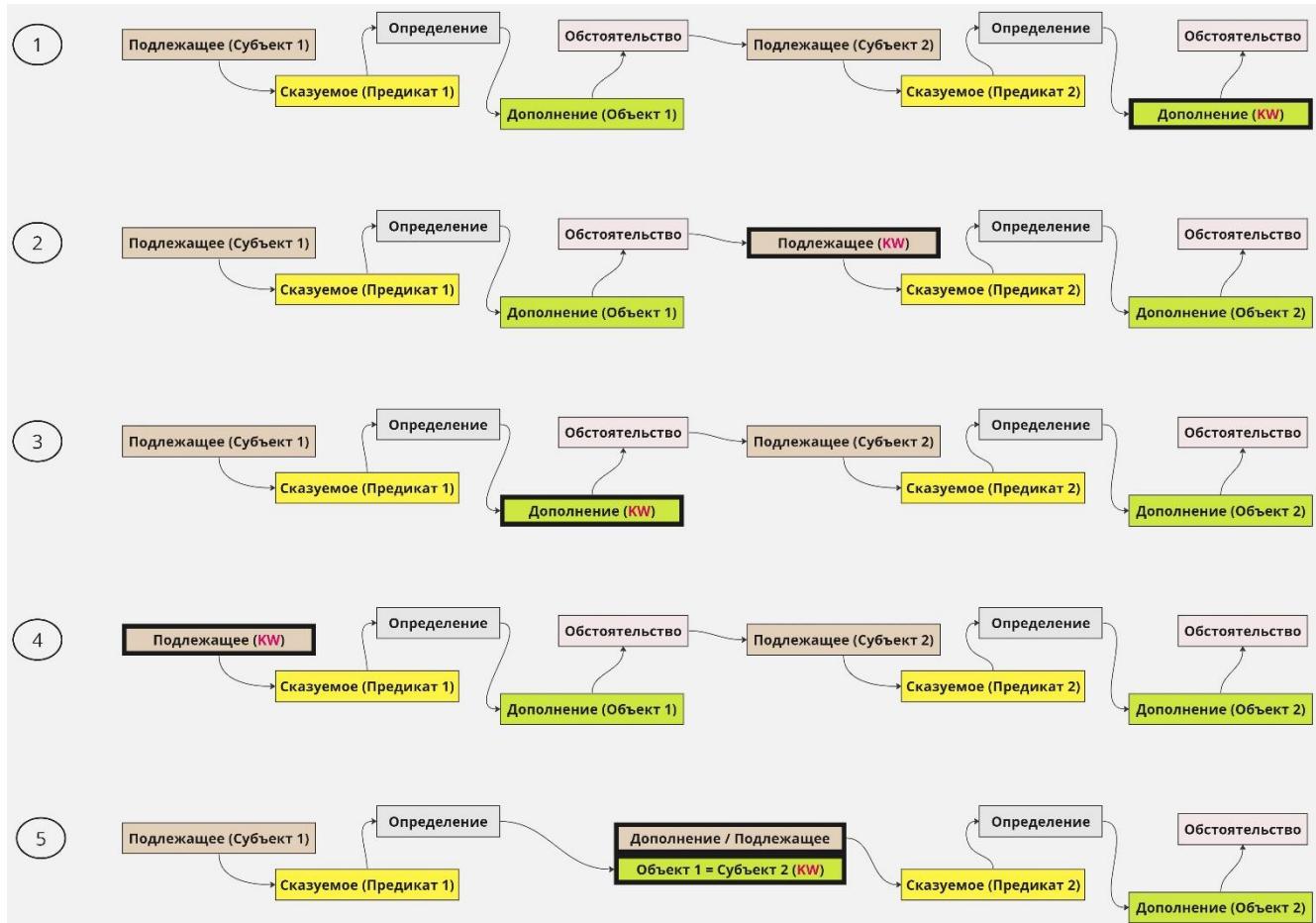


Рисунок 3.7 – Типовые структуры сложноподчиненных предложений с различной локацией ключевой фразы

2. Если ключевая фраза (KW) или любой другой объект занимают первое место в структуре предложения, тогда перейти к п. 17, в противном случае продолжить (п. 3) (*if i = 1, then go to 17, else go to 3*)  $i = 1 \dots n$ , где  $n$  – общее число слов в предложении, поступившем на вход данного алгоритма.
3. Сохранить найденный объект (*Save  $P_i = "object"$* ).
4. Движение влево от ключевой фразы (объекта) (*"move left"*,  $i = i - 1$ ).
5. Если слово слева от ключевой фразы является глаголом (сказуемым), тогда перейти к п. 7. В противном случае продолжить (п. 6) (*if  $P_i = "verb"$ , then go to 7, else go to 6*).
6. Если алгоритм достиг начала предложения, тогда перейти к п. 17. В противном случае вернуться к п. 4 (*if i = 1, then go to 17, else go to 4*).
7. Сохранить найденный предикат (глагол) (*Save  $P_i = "verb"$* ).

8. Если алгоритм достиг начала предложения, тогда перейти к п. 17. В противном случае продолжить (п. 9) (*if i = 1, then go to 17, else go to 9*).
9. Движение влево от найденного предиката (“*move left*”,  $i = i - 1$ ).
10. Если слово слева от глагола является субъектом (подлежащим), тогда перейти к п. 12. В противном случае продолжить (п. 11) (*if  $P_i$  = “subject”, then go to 12, else go to 11*).
11. Если алгоритм достиг начала предложения, тогда перейти к п. 17. В противном случае вернуться к п. 9 (*if i = 1, then go to 17, else go to 9*).
12. Сохранить найденный субъект (подлежащее) (*save  $P_i$  = “subject”*).
13. Если алгоритм достиг начала предложения, тогда перейти к п. 17, в противном случае продолжить п. 14 (*if i = 1, then go to 17, else go to 14*).
14. Если найден объект (в т.ч. для случая, когда в одной вершине находятся субъект и объект), тогда вернуться к п. 2, в противном случае продолжить (п. 15) (*if  $P_i$  = “object”, then go to 2, else go to 16*).
15. Движение влево от найденного субъекта / объекта (в т.ч. для случая, когда в одной вершине находятся субъект и объект) (“*move left*”,  $i = i - 1$ ).
16. Если алгоритм достиг начала предложения, тогда перейти к п. 17. В противном случае вернуться к п. 14 (*if i = 1, then go to 17, else go to 14*).
17.  $i = k$  (переход к ключевой фразе).
18. Если ключевая фраза (объект) одновременно является объектом для одной части предложения и субъектом для следующей, т.е.  $P_i = “object” = “subject”$ , тогда перейти к п. 23, в противном случае продолжить (п. 19) (*if  $P_i$  = “object” = “subject”, then go to 23, else go to 19*).
19. Если алгоритм достиг конца предложения, тогда перейти к п. 35. В противном случае продолжить (п. 20) (*if i = n, then go to 35, else go to 20*).
20. Движение вправо от ключевой фразы (“*move right*”,  $i = i + 1$ ).
21. Если слово справа является субъектом (подлежащим), тогда перейти к п. 23. В противном случае продолжить (п. 22) (*if  $P_i$  = “subject”, then go to 23, else go to 22*).

22. Если алгоритм достиг конца предложения, тогда перейти к п. 35. В противном случае вернутся к п. 20 (*if i = n, then go to 35, else go to 20*).
23. Сохранить найденный субъект (подлежащее) (*save  $P_i$  = "subject"*).
24. Если алгоритм достиг конца предложения, тогда перейти к п. 35, в противном случае продолжить п. 25 (*if i = n, then go to 35, else go to 25*).
25. Движение вправо от найденного субъекта ("move right",  $i = i + 1$ ).
26. Если слово справа от субъекта является глаголом (сказуемым), тогда перейти к п. 28. В противном случае продолжить (п. 27) (*if  $P_i$  = "verb", then go to 28, else go to 27*).
27. Если алгоритм достиг конца предложения, тогда перейти к п. 35. В противном случае вернутся к п. 25 (*if i = n, then go to 35, else go to 25*).
28. Сохранить найденный предикат (глагол) (*save  $P_i$  = "verb"*).
29. Если алгоритм достиг конца предложения, тогда перейти к п. 35. В противном случае продолжить (п. 30) (*if i = n, then go to 35, else go to 30*).
30. Движение вправо от найденного предиката ("move right",  $i = i + 1$ ).
31. Если слово справа от глагола является объектом (дополнением), тогда перейти к п. 33. В противном случае продолжить (п. 32) (*if  $P_i$  = "object", then go to 33, else go to 32*).
32. Если алгоритм достиг конца предложения, тогда перейти к п. 35. В противном случае вернутся к п. 30 (*if i = n, then go to 35, else go to 30*).
33. Сохранить найденный объект (дополнение) (*save  $P_i$  = "object"*).
34. Если алгоритм достиг конца предложения, тогда перейти к п. 35. В противном случае вернутся к п. 18 (*if i = n, then go to 35, else go to 18*).
35. Вывод результатов приобретения знаний при обработке и анализе текста (документа). Вывести по возрастанию порядковых номеров значения всех найденных гранул смысла, представляющих собой триплеты («субъект» - «предикат» - «объект/KW»).

*Окончание алгоритма.*

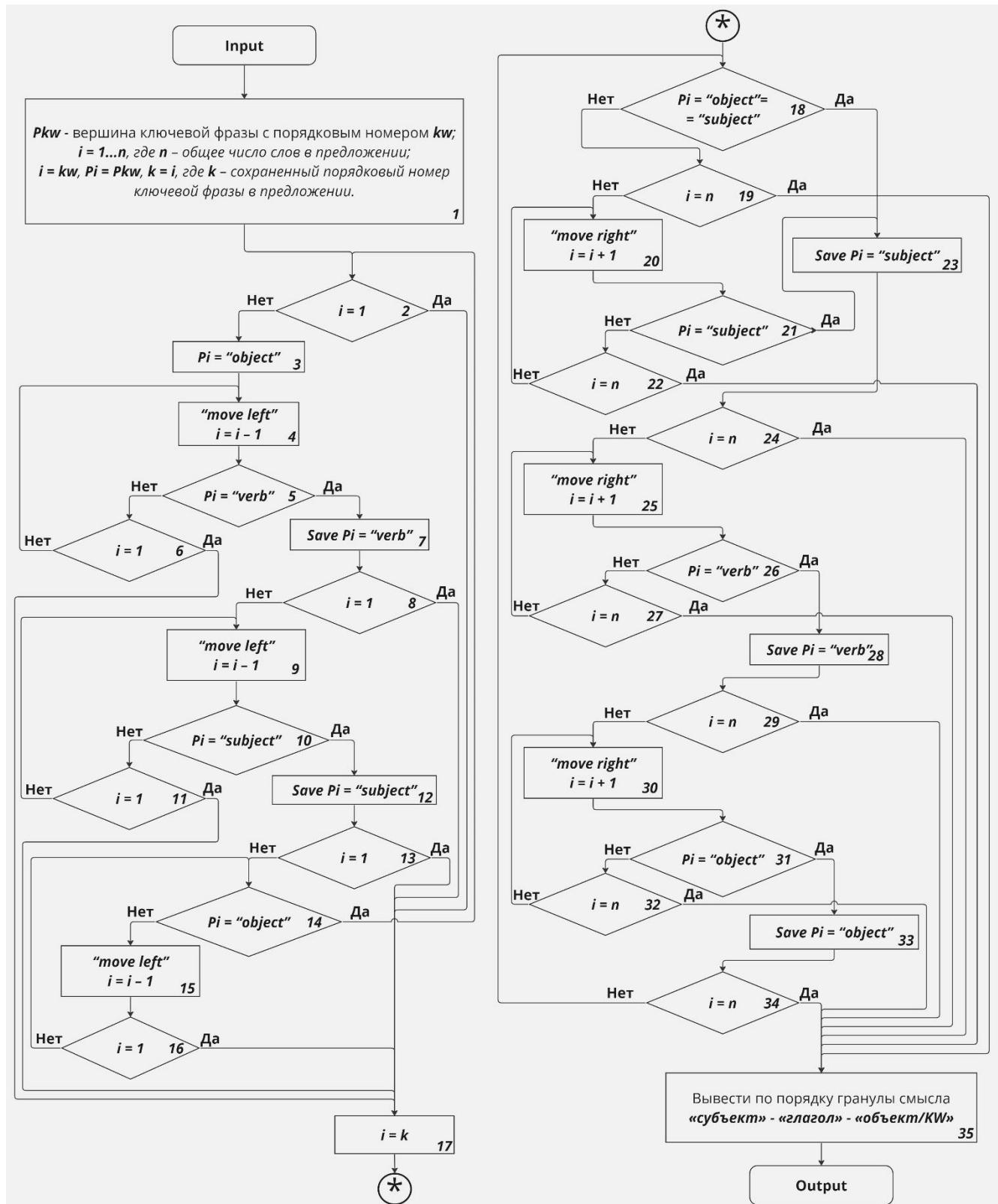


Рисунок 3.8 – Укрупненная схема алгоритма приобретения знаний в текстах на естественном языке с применением множества низкоуровневых правил семантического анализа

На рисунке 3.8 представлена укрупненная схема разработанного алгоритма приобретения знаний в текстах на естественном языке с применением множества низкоуровневых правил семантического анализа. Данный алгоритм позволяет обеспечить высокую эффективность обработки и анализа текстов на естественном языке, благодаря использованию точных низкоуровневых правил поиска гранул смысла.

Проверим работу предложенного алгоритма приобретения знаний на основе обработки и анализа упрощенной версии исследуемого примера предложения: «*МЧС России помогает населению, узнать растущее множество правил безопасности*», – полученного на выходе алгоритма поиска знаний в текстах на естественном языке с применением графовых моделей. В таблице 3.2 представим входные данные для предложенного в данном подразделе алгоритма приобретения знаний.

Таблица 3.2 – Входные данные для алгоритма приобретения знаний

Вершина	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
<b>Значение</b>	<i>МЧС России</i>	<i>помогает</i>	<i>населению</i>	<i>узнать</i>	<i>растущее</i>	<i>множество</i>	<i>правила безопасности</i>
<b>Тип элемента в грануле смысла</b>	<i>Субъект</i>	<i>Глагол (предикат)</i>	<i>Объект - Субъект</i>	<i>Глагол (предикат)</i>	<i>-</i>	<i>-</i>	<i>Объект (ключевая фраза)</i>

Напомним, что в данном примере ключевой является фраза «правила безопасности», соответственно  $P_{kw} = P_7$ , тогда согласно п. 1 алгоритма необходимо сохранить порядковый номер этой вершины  $kw = k = i = 7$ . Таким образом работа алгоритма начнется с 7-ой вершины.

Представим последовательность действий алгоритма приобретения знаний в текстах на естественном языке с применением множества низкоуровневых правил в виде таблицы 3.3.

Таблица 3.3 – Последовательность действий алгоритма приобретения знаний

Номер активного пункта структурной схемы алгоритма	Используемое правило (действие или условие)	Результат
Пункт 1.	Ввод данных: $P_i = P_{kw}$ ; $k = i$ ; $n$ – общее число вершин	$P_7 = P_{kw}$ ; $k = i = 7$ ; $n = 7$ ; $P_7 = \text{«правила безопасности»}$
Пункт 2.	<i>if</i> $i = 1$ , then go to 17, else go to 3	Нет, $i = 7$ , переход к п. 3
Пункт 3.	Save $P_i = \text{“object”}$	$P_7 = \text{«объект»}$
Пункт 4.	“move left”, $i = i - 1$	$i = 6$ ; $P_6 = \text{«множество»}$
Пункт 5.	<i>if</i> $P_i = \text{“verb”}$ , then go to 7, else go to 6	Нет, $P_6 \neq \text{“verb”}$ , переход к п. 6
Пункт 6.	<i>if</i> $i = 1$ , then go to 17, else go to 4	Нет, $i = 6$ , возврат к п. 4
Цикл Пункт 4.	“move left”, $i = i - 1$	$i = 5$ ; $P_5 = \text{«растущее»}$
Пункт 5.	<i>if</i> $P_i = \text{“verb”}$ , then go to 7, else go to 6	Нет, $P_5 \neq \text{“verb”}$ , переход к п. 6
Пункт 6.	<i>if</i> $i = 1$ , then go to 17, else go to 4	Нет, $i = 5$ , возврат к п. 4
Цикл Пункт 4.	“move left”, $i = i - 1$	$i = 4$ ; $P_4 = \text{«узнать»}$
Пункт 5.	<i>if</i> $P_i = \text{“verb”}$ , then go to 7, else go to 6	Да, $P_4 = \text{“verb”}$ , переход к п. 7
Пункт 7.	Save $P_i = \text{“verb”}$	$P_4 = \text{«глагол»}$
Пункт 8.	<i>if</i> $i = 1$ , then go to 17, else go to 9	Нет, $i = 4$ , переход к п. 9
Пункт 9.	“move left”, $i = i - 1$	$i = 3$ ; $P_3 = \text{«население»}$
Пункт 10.	<i>if</i> $P_i = \text{“subject”}$ , then go to 12, else go to 11	Да, $P_3 = \text{“subject”}$ , переход к п. 12
Пункт 12.	save $P_i = \text{“subject”}$	$P_3 = \text{«субъект»}$
Пункт 13.	<i>if</i> $i = 1$ , then go to 17, else go to 14	Нет, $i = 3$ , переход к п. 14
Пункт 14.	<i>if</i> $P_i = \text{“object”}$ , then go to 2, else go to 16	Да, $P_3 = \text{“object”}$ , возврат к п. 2
Цикл Пункт 2.	<i>if</i> $i = 1$ , then go to 17, else go to 3	Нет, $i = 3$ , переход к п. 3
Пункт 3.	Save $P_i = \text{“object”}$	$P_3 = \text{«объект»}$
Пункт 4.	“move left”, $i = i - 1$	$i = 2$ ; $P_2 = \text{«помогать»}$
Пункт 5.	<i>if</i> $P_i = \text{“verb”}$ , then go to 7, else go to 6	Да, $P_2 = \text{“verb”}$ , переход к п. 7
Пункт 7.	Save $P_i = \text{“verb”}$	$P_2 = \text{«глагол»}$
Пункт 8.	<i>if</i> $i = 1$ , then go to 17, else go to 9	Нет, $i = 2$ , переход к п. 9
Пункт 9.	“move left”, $i = i - 1$	$i = 1$ ; $P_1 = \text{«МЧС России»}$
Пункт 10.	<i>if</i> $P_i = \text{“subject”}$ , then go to 12, else go to 11	Да, $P_1 = \text{“subject”}$ , переход к п. 12
Пункт 12.	save $P_i = \text{“subject”}$	$P_1 = \text{«субъект»}$
Пункт 13.	<i>if</i> $i = 1$ , then go to 17, else go to 14	Да, $i = 1$ , (начало предложения) переход к п. 17
Пункт 17.	$i = k$ (переход (возврат) к ключевой фразе)	$i = 7$
Пункт 18.	<i>if</i> $P_i = \text{“object”} = \text{“subject”}$ , then go to 22, else go to 19	Нет, $P_7 = \text{“object”}$ , переход к п. 19
Пункт 19.	<i>if</i> $i = n$ , then go to 35, else go to 20	Да, $i = n = 7$ (конец предложения), переход к п. 35
Пункт 35.	Вывод результатов приобретения знаний при обработке и анализе текста (документа). Вывести по возрастанию порядковых номеров значения всех найденных гранул смысла, представляющих собой триплеты («субъект» - «предикат» - «объект/KW»)	Триплет 1: $P_1 = \text{«МЧС России»}$ ; $P_2 = \text{«помогать»}$ ; $P_3 = \text{«население»}$ Триплет 2: $P_3 = \text{«население»}$ ; $P_4 = \text{«узнать»}$ ; $P_7 = \text{«правила безопасности»}$

Очевидно, что без учета лемматизации, указанные в таблице 3.3 гранулы смысла, определенные в предложении, поступившем на вход алгоритма,

сформируют следующие два смысловых триплета: «МЧС России помогает населению» и «Население узнает правила безопасности». Таким образом, последовательное применение предложенных в 3-ем разделе данной диссертации алгоритмов поиска и приобретения знаний в текстах на естественном языке с применением графовых моделей и множества низкоуровневых правил семантического анализа полученных смысловых паттернов позволило создать дополнительный фильтр на выходе парсера. Данный фильтр необходим для извлечения смысловой части предложения из полученной синтаксической схемы и определения основных гранул смысла при использовании знаний.

Разработанные алгоритмы поиска и приобретения знаний используются для наполнения смысловыми паттернами онтологической структуры, построение которой описано в предыдущем разделе диссертации. Онтологическая структура является базой прецедентов для определенного домена (предметной области) и применяется в качестве информационного пространства для поиска оперативных решений при обработке пользовательского запроса биоинспирированным алгоритмом использования приобретенных знаний, разработка которого описана в следующем подразделе данного диссертационного исследования.

### **3.3. Разработка модифицированного биоинспирированного алгоритма использования приобретенных знаний в задачах генеративного искусственного интеллекта**

В данном подразделе описана разработка модифицированного биоинспирированного алгоритма использования приобретенных знаний в задачах генеративного искусственного интеллекта с применением улучшенных механизмов интенсификации поиска решений и процедур выхода из локальных оптимумов, позволяющий уменьшить время отклика на пользовательский запрос системы искусственного интеллекта и машинного обучения при обработке и анализе текстов на естественном языке.

Первоочередной задачей построения эффективной интеллектуальной информационной системы обработки пользовательских запросов и генерации релевантных ответов для поддержки принятия решений по предупреждению и ликвидации последствий чрезвычайных ситуаций (ЧС) является построение множества моделей прецедентов [70], описывающих возможные развития тех или иных ЧС. Данные модели интегрируют смысловые паттерны информационного пространства, представленные во 2-ом разделе на рисунке 2.7. Накопление системой множества указанных прецедентов, состоящих из гранул смысла, является этапом машинного обучения. После его прохождения система становится способной проводить оценку семантической близости оперативно полученных моделей с имеющимися моделями прецедентов, которые попали в разряд шаблонов. Наиболее удобной формой задания подобных информационных моделей, как уже было описано ранее, являются онтологические структуры.

Процедура оценки эквивалентной семантической близости имеет значительную вычислительную сложность, так как в случае полного перебора необходимо проверить сходство множеств атрибутов всех понятий (концептов) отображаемых онтологий [85]. Определим следующим образом эквивалентную семантическую близость при отображении онтологий:  $mapping(P^1) = P^2, \text{ if } sim(P^1, P^2) \geq b$ , где  $b$  – пороговое значение меры семантической близости  $sim(P^1, P^2)$ , при котором строится отображение понятия  $P^1$  в онтологию  $O_2$ . Совпадение большинства или всех атрибутов  $f(R^1, R^2) \rightarrow \max$  понятия  $P_i^1$  обучающей онтологии прецедентов  $O_1$  со всеми предикатами понятия  $P_j^2$  оперативной (динамической) онтологии  $O_2$  (эквивалентность множеств  $R^1, R^2$ ) указывает на семантическую близость данных понятий [86-89]. Наличие большинства эквивалентных понятий, определяемых заданным порогом, в двух отображаемых онтологиях указывает на сходство между прецедентом и реальной ситуацией развития ЧС.

Заданы  $R^1, R^2$  – множества атрибутов понятий онтологий  $O_1$  и  $O_2$ . Каждое множество атрибутов состоит из упорядоченных подмножеств  $R_i^1$  и  $R_j^2$ , принадлежащих каждому из понятий  $P_i^1$  и  $P_j^2$  указанных онтологий, где  $i = [1, N]$ ,  $N$  – количество понятий в  $O_1$ ;  $j = [1, M]$ ,  $M$  – количество понятий в  $O_2$ . Для оценки эквивалентной семантической близости понятий  $P_i^1$  и  $P_j^2$  используется следующий алгоритм [9, 69]:

---

*Алгоритм определения семантической близости понятий  $P_i^1$  и  $P_j^2$*

---

<i>Ввод</i>	Кортежи значений атрибутов $R_i^1$ и $R_j^2$ понятий $P_i^1$ и $P_j^2$ ; значение $\Delta$ допустимого порога неравенства; $k$ – номер атрибута, $k = 1 \dots W$ , $W$ – число атрибутов; $i, j – const, i = \overline{1, n}, j = \overline{1, m}$
<i>Выход</i>	Результат определения семантической близости между понятиями $P_i^1$ и $P_j^2$
1:	$k = 1$
2:	<i>While</i> ( $k \leq W$ ) <i>do</i>
3:	$  if(R_{ik}^1 \neq R_{jk}^2) then$
4:	$  \Delta = \Delta - 1$
5:	$  k++$
6:	$  else$
7:	$  k++$
8:	$  end$
9:	$  end$
10:	$  if(\Delta \geq 0) then$
11:	$  P_i^1$ – семантически близок $P_j^2$
12:	$  else$
13:	$  Для P_i^1 и P_j^2 семантическая близость отсутствует$
14:	$  end$
15:	Сохранение результата

---

Причем, максимизация показателя  $\Delta \rightarrow max$  в процессе выполнения данного алгоритма указывает на полноту эквивалентной семантической близости между рассматриваемыми концептами (понятиями)  $P_i^1$  и  $P_j^2$ .

Автор применяет для оценки эквивалентной семантической близости биоинспирированный алгоритм с децентрализованным механизмом поиска решений, что обеспечивает высокую параллельность вычислений и позволяет, с одной стороны, интенсифицировать поиск в различных локальных областях информационного пространства, а с другой, проводить диверсификацию пространства поиска решений на основе реализации механизма выхода из локальных оптимумов [86, 87].

Указанным характеристикам в полной мере соответствует алгоритм бактериальной оптимизации (БО), предложенный Пассино (Passino) в 2002 году [9, 69, 90-97]. В соответствии с приведенной выше информационной моделью, пространство для поиска решений при генерации ответа на запрос пользователя представлено двумя онтологическими структурами: первая – обучающая; вторая – оперативная. Обучающая онтология  $O_1$  построена на основе успешных precedентов принятия решений, т.е. является эталоном. Оперативная онтология  $O_2$  отражает систему отношений на множестве элементов информации (понятий) реальной оцениваемой чрезвычайной ситуации.

Для простоты описания представлена система связей, построенная на множестве вершин онтологии, каждая из которых является определенным понятием (концептом). Связи между вершинами отражают множества отношений  $C^1$  и  $C^2$  онтологий  $O_1$  и  $O_2$  соответственно. Параллельность вычислений обеспечивает применение колонии бактерий, размер которой является числом агентов в колонии бактерий и задается значением  $S$ . В предложенной модификации данного алгоритма используются жадные эвристики при репродукции бактерий с высоким значением целевой функции [98-100], что позволило повысить эффективность решения поставленной оптимизационной задачи.

В вершинах онтографов находятся значения атрибутов сравниваемых понятий, т.е. подмножества  $R_1^1; R_2^1; \dots; R_i^1; \dots; R_N^1$  для обучающей онтологии  $O_1$ , и подмножества  $R_1^2; R_2^2; \dots; R_j^2; \dots; R_M^2$  для оперативной онтологии  $O_2$ . Задано значение  $\Delta$  допустимого порога неравенства. Для определения эквивалентной

семантической близости между подмножествами атрибутов, принадлежащих понятиям  $P_i^1$  и  $P_j^2$ , применен описанный выше эвристический алгоритм в сочетании с модифицированным алгоритмом бактериальной оптимизации. Случайным образом на каждой итерации работы алгоритма для колонии бактерий выбирается пара атрибутов с номерами  $a$  и  $b$ , где  $a \in [1, N], b \in [1, M]$ . В итоге формируется выборка данных, где каждая пара атрибутов содержит по одному элементу из обучающей онтологии  $O_1$  и оперативной (динамической) –  $O_2$ . Предложенный автором для разрабатываемого биоинспирированного алгоритма асимметричный механизм поиска реализуется передвижениями агентов-бактерий только в обучающей онтологии  $O_1$ , а находящиеся с ними в паре агенты-бактерии из оперативной онтологии  $O_2$ , применяются как неподвижные константы для сравнения предыдущих и последующих значений целевых функций. Это позволяет сократить процесс использования приобретенных знаний при генерации ответа на запрос пользователя, так как обучающая онтология содержит только основные гранулы смысла, что делает ее более компактной, чем в аналогах данного алгоритма.

В предложенном модифицированном алгоритме бактериальной оптимизации предусмотрена реализация ряда канонических механизмов поиска. Моделирование целенаправленного передвижения агента-бактерии называется «хемотаксисом» [96] и позволяет имитировать поиск пищи бактериями. Механизм «кувырка» – возврат бактерии в предыдущее местоположение и выбор нового направления движения в случае ухудшения значения целевой функции. Интенсификацию поиска в успешных областях информационного пространства моделирует механизм «репродукции». Диверсификация пространства поиска основана на моделировании процессов «ликвидации» наиболее отстающих агентов и «рассеивания» вновь сгенерированных [96].

Текущее положение агента-бактерии  $s_i \in S$  на  $t$ -м шаге хемотаксиса,  $r$ -м шаге репродукции и  $l$ -м шаге ликвидации и рассеивания задано следующим выражением:

$$X_{i,r,l} = X_{i,r,l}(t), \quad (3.2)$$

где  $i \in [1, |S|]$ ,  $t \in [1, T]$ ,  $r \in [1, T_r]$ ,  $l \in [1, T_l]$ , где  $T, T_r, T_l$  – значения числа шагов «хемотаксиса», «репродукции», «ликвидации и рассеивания» соответственно (свободные параметры алгоритма), а  $|S|$  – четное число бактерий в колонии [96].

Для определения вектора передвижения агента-бактерии  $V_i(\varepsilon)$  задана локальная степень (мощность окрестности)  $\varepsilon_m$ , где  $m \in [1, M]$  для вершины онтографа  $O_2$ , для вершины онтологии  $O_1$  мощность окрестности обозначена  $\varepsilon_n$ , где  $n \in [1, N]$ . На каждом новом шаге передвижения агент-бактерия выбирает вершину онтографа с максимальным значением локальной степени (мощности окрестности) [9, 69], это позволяет повысить вероятность улучшения значений целевой функции.

В работах [9] и [69] для интенсификации поиска при бактериальной оптимизации на множествах данных большой размерности авторами вводилась переменная  $\eta \geq 1$ , задающая число взаимосвязанных вершин онтографа, которые бактерия проходила за один шаг «хемотаксиса», что позволяло настроить показатель скорости передвижения агента-бактерии в пространстве поиска решений. Это повышает эффективность биоинспирированного поиска на множестве необработанной предварительно текстовой информации. В предложенном автором диссертации модифицированном алгоритме бактериальной оптимизации нет необходимости в подобной постоянной интенсификации процедур поиска решений, так как в подразделах 3.1 и 3.2 описаны новые алгоритмы поиска и приобретения знаний, позволяющие построить онтологическую структуру, содержащую только отфильтрованные гранулы смысла в паттернах прецедентов. Таким образом, введение любого множителя для определения длины шага бактерии, непременно приведет к пропуску некоторых гранул смысла («субъект» - «предикат» - «объект») в одном паттерне, что в целом значительно снизит качество механизмов поиска, так как нарушит целостность паттернов прецедентов.

Чтобы исключить этот недостаток в данном исследовании при модификации алгоритма бактериальной оптимизации автор применяет для интенсификации поиска решений механизм «локального прыжка», взятый из одной разновидности алгоритма обезьяньего поиска (ОП), предложенной Чжао (Zhao) и Тангом (Tang) в 2007 году [96]. Данный алгоритм использует модель передвижения обезьян по горам. Согласно концепции алгоритма, гора с наивысшей вершиной содержит наибольшее количество пищи. Исследуемые обезьянами склоны гор являются ландшафтом целевой функции. Решением задачи максимизации является пик самой высокой вершины [96, 101-105].

Отличительной особенностью алгоритма является способность обезьян, достигнувших одной из вершин, совершать локальные и глобальные прыжки, с целью обнаружения наиболее высоких пиков. Локальные прыжки необходимы для обследования ближайших к координатам местонахождения обезьяны территорий.

Допустим, что бактерия способна изменить свою локацию в пространстве поиска на заданную глубину  $1 < \lambda_i \leq \lambda_{max}$ , соблюдая заданный вектор перемещения и моделируя тем самым механизм «локального прыжка», что позволяет интенсифицировать поиск в области нахождения бактерии при отрицательной динамике изменения целевой функции. Это является *основным отличием* предложенного модифицированного алгоритма бактериальной оптимизации. При этом, не происходит разрыва связей между гранулами смысла, так как после «локального прыжка» бактерия продолжает последовательный поиск семантически близких концептов текстовой информации в рамках одного паттерна precedента. Момент реализации «локального прыжка» регулируется переменной  $k$ , значение которой является свободным параметром алгоритма и задает число неудачных шагов агента-бактерии подряд, после которых происходит «локальный прыжок». Соответственно, в этом случае, количество «кувыроков» бактерии до «локального прыжка» равно  $k - 1$ .

Если после «локального прыжка» бактерия попадет не в начало паттерна, тогда при дальнейшем последовательном движении по ближайшим вершинам

онтологии она самостоятельно определит все гранулы смысла в этом паттерне, благодаря заданному вектору перемещений и механизму «кувырка» для смены направления данного вектора. Длина «локального прыжка» (watch-jump process) является свободным параметром алгоритма, либо определяется случайным образом, но при соблюдении условия, что длина «локального прыжка» не менее 2 и не более  $\lambda_{max}$ . Максимальная длина прыжка  $\lambda_{max}$  так же является свободным параметром алгоритма и задается пользователем.

Значение целевой функции представлено виде канонического выражения (3.3) [96]:

$$\varphi_{i,r,l} = \varphi_{i,r,l}(T). \quad (3.3)$$

Как уже было сказано выше, локальную оптимизацию в алгоритме осуществляет механизм «хемотаксиса». Новое местоположение  $X'_{i,r,l}$  агента-бактерии  $s_i$  на  $t+1$  шаге хемотаксиса вычисляется на основе выражения (3.4):

$$X'_{i,r,l} = X_{i,r,l} + \eta_i V_i(\varepsilon), \quad (3.4)$$

напомним, что, как уже было описано выше, в данной модификации алгоритма бактериальной оптимизации длина шага  $\eta_i$  всегда будет константой со значением равным 1. Это сделано для построения последовательной цепочки шагов поиска гранул смысла внутри одного паттерна прецедентов.

В случае реализации «локального прыжка» данное выражение примет следующий вид:

$$X'_{i,r,l} = X_{i,r,l} + \lambda_i V_i(\varepsilon). \quad (3.5)$$

Вектор  $V_i(\varepsilon)$  остается неизменным  $V'_i(\varepsilon) = V_i(\varepsilon)$  в том случае, если не происходит уменьшения значений целевой функции  $\varphi'_{i,r,l} \geq \varphi_{i,r,l}$ , то есть агенту-бактерии удается улучшать показатель семантической близости между анализируемыми на данном шаге вершинами обучающей и оперативной онтологий. Проиллюстрируем на рисунках 3.9 – 3.11 логику передвижения агента-бактерии сначала пошагово с «кувырками», а затем в виде «локального прыжка».

Непосредственно на рисунке 3.9 продемонстрирован этап начала поиска, когда агент-бактерия делает первый шаг из начальной вершины  $P_1$  в вершину  $P_2$ , имеющую наибольшее из всех инцидентных  $P_1$  вершин значение мощности окрестности  $\varepsilon$ . В примере на рисунке 3.9 данное значение  $\varepsilon(P_2) = 9$ .

Предположим, что в результате шага бактерии, показанного на рисунке 3.9, значение целевой функции не уменьшилось  $\varphi'_{i,r,l} \geq \varphi_{i,r,l}$ , в этом случае данный шаг считается успешным, и бактерия продолжает поиск вершин с наибольшим значением мощности окрестности  $\varepsilon$  в том же направлении вектора движения  $V'_i(\varepsilon) = V_i(\varepsilon)$ . В рассматриваемом на рисунке 3.9 примере такой вершиной является  $P_3$ , для которой значение  $\varepsilon(P_3) = 6$ .

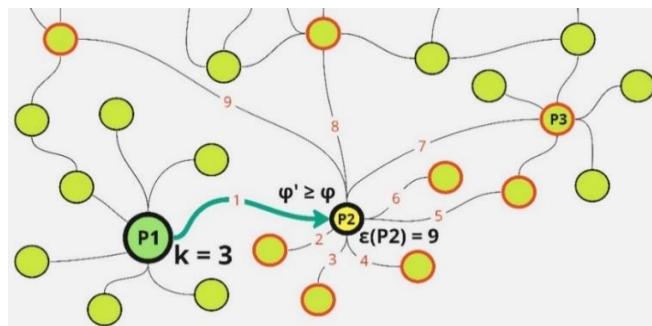


Рисунок 3.9 – Иллюстрация шага бактерии с увеличением значения целевой функции

Если при перемещении в вершину  $P_3$  произойдет уменьшение значения целевой функции  $\varphi'_{i,r,l} < \varphi_{i,r,l}$ , тогда бактерия сделает «кувырок» и вернется в вершину  $P_2$ , чтобы продолжить поиск в другом направлении, как это показано на рисунке 3.10.

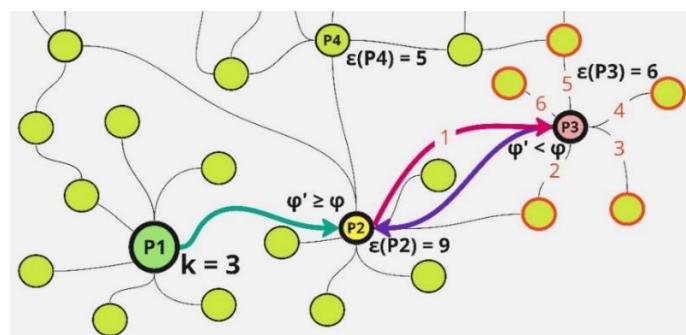


Рисунок 3.10 – Иллюстрация «кувырка» бактерии при реализации шага с уменьшением значения целевой функции

Следующей вершиной поиска станет  $P_4$ , для которой  $\varepsilon(P_4) = 5$ , как это показано на рисунке 3.11. Если ситуация с уменьшением значения целевой функции повторится для вершины  $P_4$ , тогда снова произойдёт «кувырок» бактерии с возвратом в  $P_2$  и последующим продолжением поиска. Напомним, что до начала поиска решений задается свободный параметр  $k$ , ограничивающий число подряд проведенных ошибочных шагов. В рассматриваемом примере значение данного показателя равно 3 (рис. 3.10). Поэтому, если после шага в вершину  $P_4$  агент-бактерия проведет перемещение, например, в вершину  $P_5$ , для которой  $\varepsilon(P_5) = 4$  (рис. 3.11), и этот шаг так же будет ошибочным 3-ий раз подряд ( $\varphi'_{i,r,l} < \varphi_{i,r,l}$ ), тогда после этого (без возврата в вершину  $P_2$ ) будет реализован «локальный прыжок» с установленной длиной  $\lambda_i = 4$  из вершины  $P_5$  напрямую в вершину  $P_9$ , для которой  $\varepsilon(P_9) = 7$ , и которая не имеет общего ребра с  $P_5$ , как это показано на рисунке 3.11.

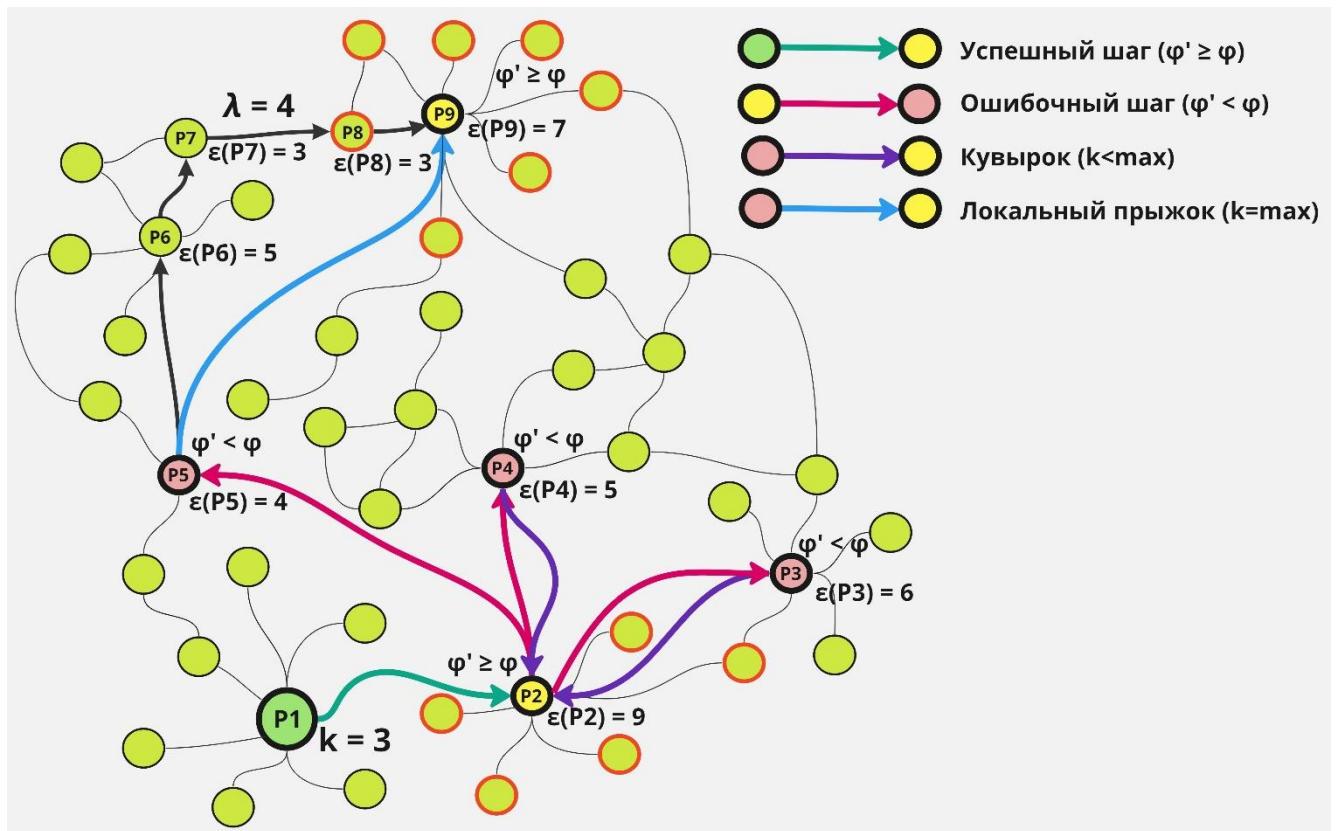


Рисунок 3.11 – Иллюстрация «локального прыжка» бактерии

Проведённый «локальный прыжок», помимо интенсификации поисковых процедур, также позволяет диверсифицировать пространство поиска, исключив траекторию передвижения агента-бактерии по ребрам онтографа через вершины  $P_6$  –  $P_7$  –  $P_8$ , тем самым снижая временные затраты на обработку и анализ используемых в описанной информационной модели знаний. Диверсификация пространства поиска в целом будет значительней при больших значениях  $\lambda_{max}$ .

В случае попадания агента-бактерии в локальный оптимум, что характеризуется локализацией значений уровня текущего здоровья  $h_i$  бактерии  $s_i$  в средней части области допустимых значений. Вычисление текущего уровня здоровья происходит на основе канонического выражения (3.6), в котором данный параметр определяется суммой значений целевой функции, рассчитанной во всех точках пройденной траектории перемещений [9, 69, 96]:

$$h_i = \sum_{\tau=1}^T \varphi_{i,r,l}(\tau), i \in [1, |S|]. \quad (3.6)$$

Для достижения баланса между скоростью сходимости и диверсификацией пространства поиска в предлагаемом модифицированном алгоритме бактериальной оптимизации, с одной стороны, реализован механизм «репродукции», позволяющий интенсифицировать протекание процесса поиска в локальных областях, а с другой, используется механизм «ликвидации и рассеивания», позволяющий агентам-бактериям при необходимости покинуть локальные оптимумы.

Достижение указанного баланса имеет важное значение для обеспечения точности и производительности биоинспирированных алгоритмов при поиске квазиоптимальных решений. На баланс влияют механизм отбора решений и аттрактивность применяемых операторов. На вычислительную трудоемкость влияют сортировка популяции решений, измерения в популяции и сложность целевой функции [9, 69, 96]. Среди важных преимуществ предложенного алгоритма на основе биоинспирированного поиска выделим незначительный требуемый объем памяти и относительно простую настройку параметров в сочетании с невысокой трудоемкостью реализации.

Механизм «репродукции» задействуется после вычисления значений уровня текущего здоровья  $h_i$  всех агентов-бактерий, номера которых записываются в порядке убывания показателей здоровья. Затем, при достижении переменной  $r$  значения  $T_r$  половина наиболее слабых агентов-бактерий исключается из рассмотрения, а каждый выживший (сильный) агент-бактерия дублируется своей копией, с такими же координатами местоположения в пространстве поиска.

Например, если успешный агент-бактерия  $s_j, j \in [1, |S|]$  имеет местоположение  $X_{j,r,l}$ , тогда после «репродукции» появится агент-бактерия  $s_k$ , причем,  $k = \frac{|S|}{2} + j$ ,  $X_{j,r+1,l} = X_{j,r,l}$ ,  $X_{k,r+1,l} = X_{j,r,l}$ . Таким образом, после «репродукции» размер популяции останется неизменным.

Интенсификация поиска решений в области пространства ограниченного размера с течением времени приводит к попаданию агента-бактерии в локальный оптимум, что является недостатком алгоритма бактериальной оптимизации [9, 69, 96]. Для снижения вероятности возникновения подобных ситуаций используется механизм «ликвидации и рассеивания», который позволяет агентам-бактериям покинуть «локальные ямы».

Для определения момента запуска механизма «ликвидации и рассеивания» введена переменная  $\theta_z$ , где  $z \in [1, Z]$ ,  $Z$  – число реализованных «репродукций» до начала «ликвидации и рассеивания» (свободный параметр алгоритма), который задает момент запуска механизма «ликвидации и рассеивания». После выполнения каждой «репродукции» агентов-бактерий начальное значение переменной  $\theta_z = 1$  увеличивается на единицу, а достижение переменной  $\theta_z$  значения  $Z$  активирует механизм «ликвидации и рассеивания», при реализации которого исключаются из рассмотрения случайным образом выбранные  $w < |S|$  агентов-бактерий [9, 69, 96]. Вместо исключенных из рассмотрения генерируется равное число агентов-бактерий с начальными координатами в случайно выбранных парах  $P_i^1$  и  $P_j^2$  онтологий  $O_1$  и  $O_2$ . Для вновь сгенерированных агентов-бактерий выполнение алгоритма начинается заново. Укрупненная схема модифицированного алгоритма бактериальной оптимизации (БО) представлена на рисунке 3.12.

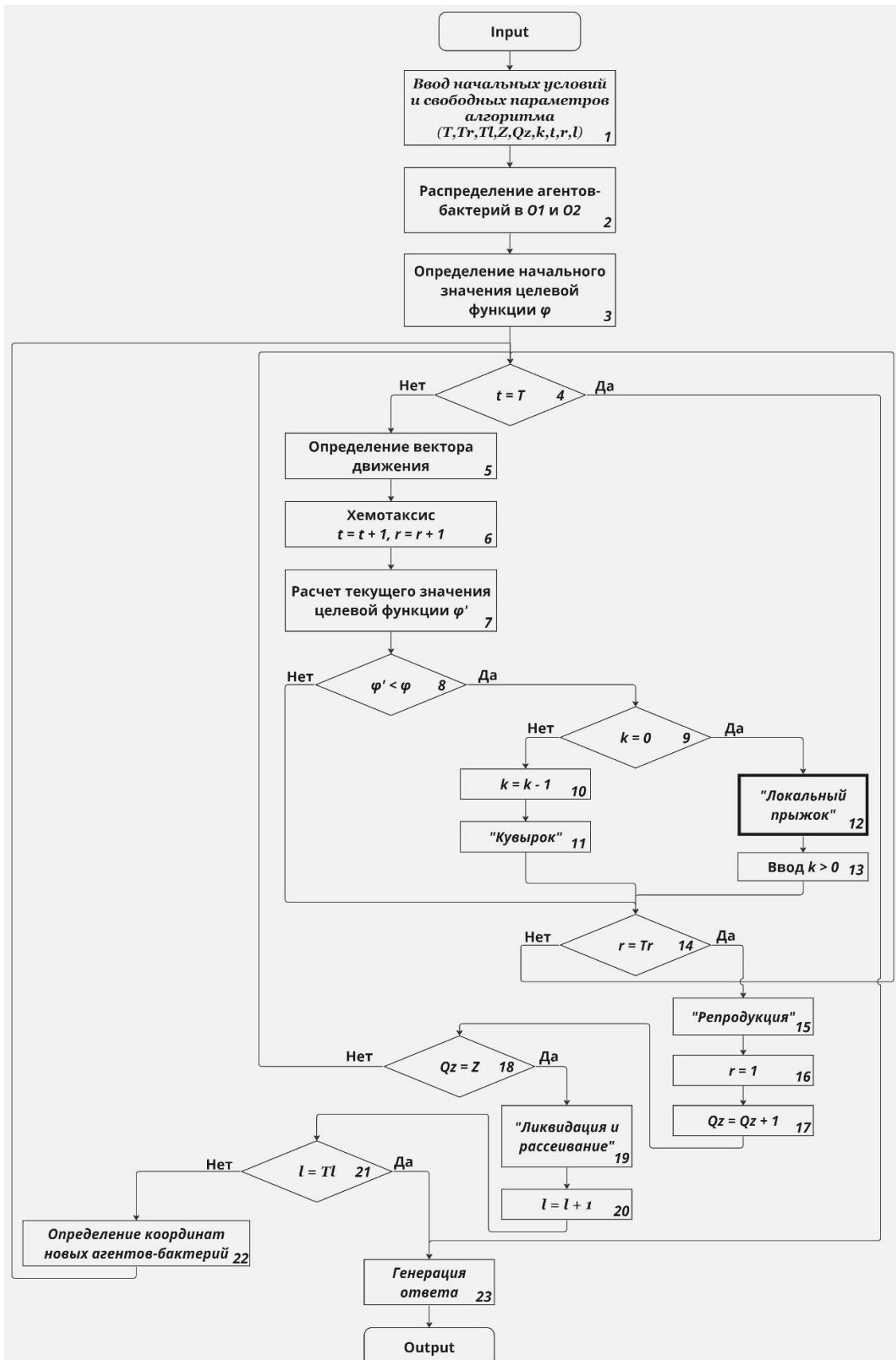


Рисунок 3.12 – Укрупненная схема модифицированного алгоритма бактериальной оптимизации при использовании знаний

Как видно из представленной на рисунке 3.12 схемы, в предложенном алгоритме последовательно реализуются основные механизмы («кувырка», «локального прыжка», «репродукции», «ликвидации и рассеивания») интенсификации поиска решений и диверсификации информационного пространства использования знаний системой искусственного интеллекта и машинного обучения при обработке и анализе текстов на естественном языке. При этом применение более компактной (с точки зрения приобретения знаний, содержащих только основные гранулы смысла) обучающей онтологии позволяет ускорить процесс генерации ответа на запрос пользователя.

Отметим также, что колония агентов-бактерий значительной размерности позволяет построить масштабные пространственно-временные структуры со сложной системой опосредованных отношений, позволяющих повысить эффективность использования приобретенных знаний при поддержке принятия решений по предупреждению и ликвидации последствий ЧС [9, 69, 96].

В данном подразделе диссертационной работы автором предложен модифицированный алгоритм бактериальной оптимизации для использования знаний при обработке и анализе текстов на естественном языке, отличающийся применением новых механизмов асимметричного поиска семантически близких концептов и «локального прыжка» агента-бактерии для интенсификации поиска решений и диверсификации информационного пространства при выходе из «локальных ям».

### **3.4. Выводы по разделу**

В представленном разделе решена задача разработки комплекса алгоритмов поиска и приобретения знаний в текстах, а также использования приобретенных знаний, что позволяет извлекать смысловую часть предложения из полученной синтаксической схемы текстовой информации и определять гранулы смысла, а также проводить интенсификацию и диверсификацию поисковых процедур для

уменьшения времени отклика системы искусственного интеллекта и машинного обучения на пользовательский запрос при обработке и анализе текстов на естественном языке.

Разработан алгоритм поиска знаний в текстах на естественном языке с применением графовых моделей, позволяющих получить структурированную, компактную и формализованную форму описания текстовой информации, которая упрощает процесс поиска знаний. При этом графовое представление используется в качестве инструмента для сокращения и упрощения предложения за счет исключения "мусорных" (бессмысленных) слов и нахождения прямых отношений между словами – носителями смысла.

Разработан алгоритм приобретения знаний в текстах на естественном языке с применением множества низкоуровневых правил семантического анализа уже полученных смысловых паттернов, позволяющий улучшить результат работы алгоритма поиска знаний и построить основные гранулы смысла для процессов использования знаний. Разработанные алгоритмы поиска и приобретения знаний используются для наполнения обучающей онтологии смысловыми паттернами прецедентов.

В целях реализации процессов использования приобретенных знаний разработан модифицированный алгоритм бактериальной оптимизации, который отличается от аналогов улучшенными механизмами интенсификации поиска решений и процедурами выхода из локальных оптимумов, что позволяет проводить оценку семантической близости прецедентов и анализируемых знаний с уменьшением времени генерации ответа системы искусственного интеллекта и машинного обучения на запрос пользователя при обработке и анализе текстов на естественном языке.

В следующем разделе диссертации описана подготовка и проведение вычислительного эксперимента для оценки эффективности разработанных моделей и алгоритмов. Представлено описание технической реализации базы данных для хранения построенных в диссертации онтологических моделей.

Предложена компонентная архитектура программного приложения, реализующего разработанные алгоритмы поиска, приобретения и использования знаний в системах искусственного интеллекта и машинного обучения при обработке и анализе текстов на естественном языке. Описан процесс построения и обработки графовых моделей текстовой информации. Представлены и проанализированы результаты вычислительного эксперимента. Проведена оценка временной сложности предложенных алгоритмов. Приведенные автором результаты показали непротиворечивость разработанных моделей и алгоритмов поиска, приобретения и использования знаний в текстах на естественном языке.

## 4. РАЗРАБОТКА ПРОГРАММНОГО ПРИЛОЖЕНИЯ И ПРОВЕДЕНИЕ ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА

В данном разделе диссертации описана разработка программного приложения, а также подготовка и проведение вычислительного эксперимента для оценки эффективности созданных моделей и алгоритмов. Построена компонентная архитектура программного приложения, реализующего разработанные алгоритмы поиска, приобретения и использования знаний в системах искусственного интеллекта и машинного обучения при обработке и анализе текстов на естественном языке. Представлено описание технической реализации базы данных для хранения построенных в диссертации онтологических моделей. Описан процесс построения и обработки графовых моделей текстовой информации. Представлены и проанализированы результаты вычислительного эксперимента.

### 4.1. Разработка компонентной архитектуры программного приложения

Для проведения вычислительного эксперимента разработано программное приложение, компонентная архитектура которого представлена на рисунке 4.1. Приложение реализует функции первичной обработки текста, сегментации изображений, получения синтаксической схемы текста от парсера, построения и обработки ациклического ориентированного графа, создания онтологий, получения сравнительных оценок и реализации разработанных автором алгоритмов поиска, приобретения и использования знаний, а также генерации ответов на запрос пользователя [106, 107].

Клиентское приложение представляет собой динамически обновляемый *Single Page Application (SPA)* на популярном *Javascript* фреймворке *React*. В данном SPA приложении, когда пользователь открывает страницу, браузер загружает сразу весь код приложения. Но показывает только конкретный модуль – часть сайта, которая нужна пользователю [108, 109]. Когда пользователь переходит в другую

часть приложения, браузер берёт уже загруженные данные и показывает ему. И, если нужно, динамически подгружает с сервера нужный контент без обновления страницы.

С одной стороны, такие приложения работают быстро и меньше нагружают сервер. С другой стороны, они требуют большей загрузки на старте [109].

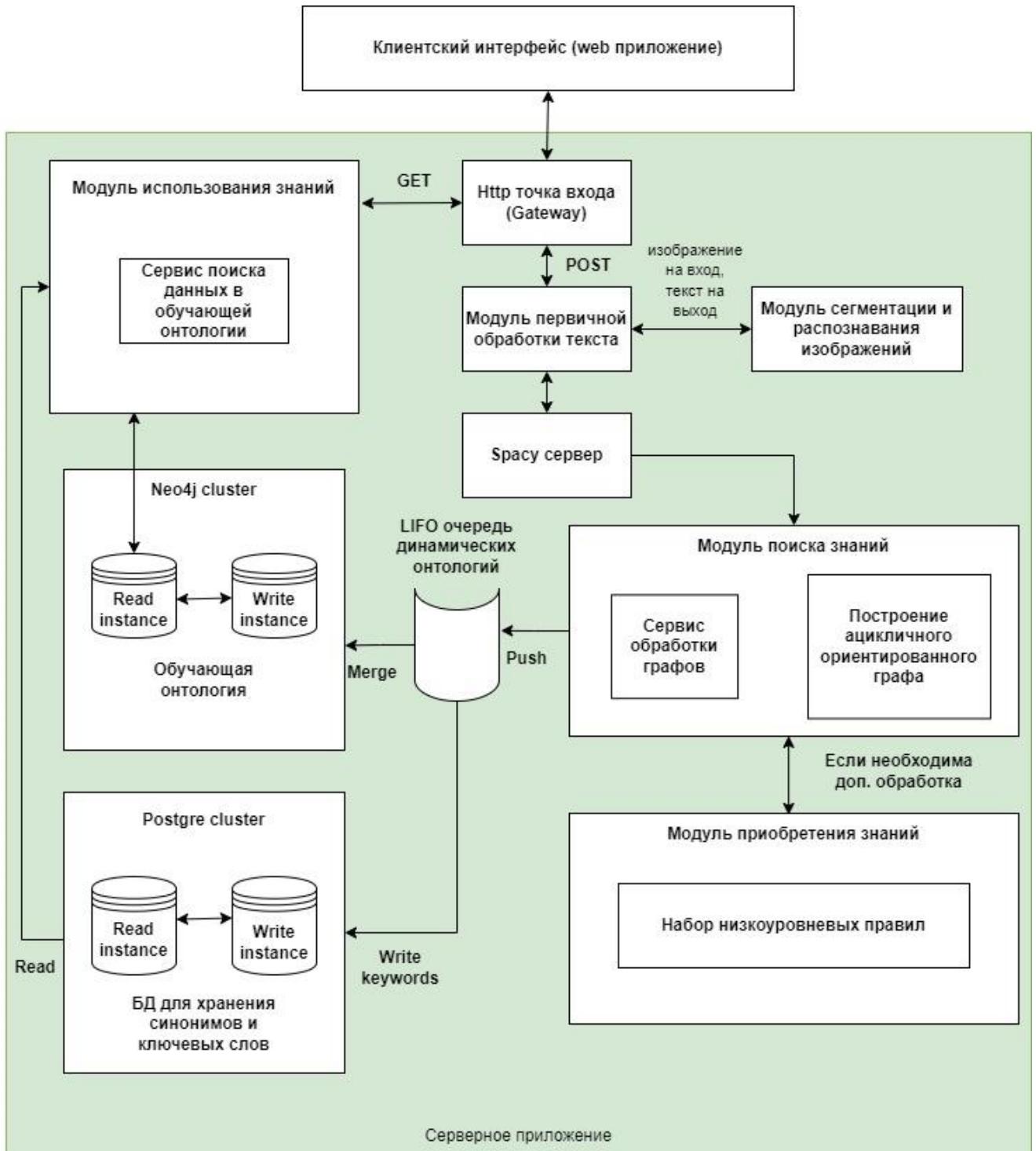


Рисунок 4.1 – Компонентная архитектура программного приложения

Вся компонентная архитектура разворачивается в виде *Open Container Initiative (OCI) Docker*-образа, состоящего из следующих компонентов (рис. 4.1):

- *Nginx* (веб-сервер для балансировки высокой нагрузки и отдачи статического контента, например, html страниц, медиафайлов, документов, архивов, картинок и т.д.);
- библиотека *React* – это *JavaScript* библиотека для создания клиентских приложений. Основана данная библиотека на компонентах и связях между ними, что является одной из главных ее особенностей и преимуществ, все свойства передаются от родительских компонентов к дочерним. Компоненты получают свойства, как множество неизменяемых значений. Также особенностями данной библиотеки является *virtual-dom*, который ускоряет скорость рендеринга приложений, за счет создание кэш-структур, способной вычислять разницу между предыдущим и текущим состоянием интерфейса для оптимального обновления *DOM* браузера;
- сервер парсера *SpaCy*;
- основное приложение с модулями (*NestJS App*), в которых реализованы разработанные алгоритмы поиска, приобретения и использования знаний;
- *Neo4j* – это графовая база данных для хранения построенных онтологий;
- *PostgreSQL* – это объектно-реляционная СУБД (ORDBMS) для хранения таблиц синонимов и ключевых слов.

В процессе работы приложения на вход приходит корпус текста. Предусмотрена возможность обработки текста, представленного в виде изображения. Для этого в состав построенной архитектуры приложения включен компонент (модуль) сегментации и распознавания изображений. Точкой входа в приложение является программа *OPC-HTTP Gateway*, которая реализует простой шлюз *OPC-HTTP* и позволяет страницам веб-сервера взаимодействовать с *OPC*-серверами через защищённые каналы. *OPC* (*Open Platform Communications*) – семейство программных технологий, обеспечивающих единый интерфейс для управления объектами автоматизации и технологическими процессами [110].

Серверное веб-приложение создано с применением фреймворка *NestJS* [111], позволяющего строить эффективные масштабируемые приложения на *Node.js*. После инициализации проекта *Node.js* на следующем шаге произведена установка пакетов *Express*. Созданный затем *Express*-сервер поддерживает взаимодействие приложения с пользователями.

При поступлении текста в любом из описанных выше форматов, контроллер *Gateway* выполняет проверку типа данных, и если на вход пришло изображение, то вызывается модуль сегментации и распознавания изображений. По умолчанию в данном модуле используется сервис *Amazon Web Service (AWS) Rekognition*, потом данные в виде текста поступают в модуль первичной обработки текста, если же на вход пришел корпус текста в файле из текстового редактора, тогда контроллер отправляет входные данные в модуль первичной обработки текста сразу, как это показано на рисунке 4.1.

В модуле первичной обработки текста реализуется функция разбиения всего корпуса на множество предложений, каждое из которых затем обрабатывается на сервере парсера *SpaCy* [112]. Полученные синтаксические схемы поступают на вход разработанного модуля поиска знаний. В данном модуле происходит построение и обработка моделей текста для каждого предложения, представленных в виде ациклических ориентированных графов.

В случае, если после обработки первой версии графовой модели предложения при помощи разработанного алгоритма поиска знаний очистка предложения произведена не в полной мере, тогда текущая версия графа, размещенная в оперативной онтологии, отправляется на дополнительную обработку в модуль приобретения знаний, функционирующий на основе алгоритма низкоуровневых правил. Происходит доочистка графовой модели предложения от вершин (слов) инвариантных к смыслу. В конечном итоге оперативная онтология с полученными смысловыми паттернами записывается в очередь с дисциплиной обслуживания *LIFO*.

Далее происходит отображение (*mapping*) оперативной и обучающей онтологий. При этом, обучающая онтология хранится в *Neo4j* кластере [113], где каждая графовая база данных имеет полную копию для обеспечения отказоустойчивости и высокой доступности. В процессе отображения онтологий параллельно происходит запись ключевых слов и синонимов в дополнительную объектно-реляционную систему управления базами данных (*ORDBMS*) *PostgreSQL* [114], одну из наиболее развитых из открытых СУБД в мире.

Третий модуль использования знаний взаимодействует с кластерами данных для получения необходимой информации. Основная функция поиска в модуле использования знаний имеет вид итерационной процедуры с блоком проверок. Суть каждой итерации заключается в перемещении агентов по онтологии и измерении разности первоначального значения целевой функции со значением полученным на данной итерации. Далее в зависимости от результатов сравнения существует несколько вариантов продолжения работы поиска. Также необходимо учитывать инкрементирующие переменные, которые хранят в себе информацию о пройденных итерациях и являются абстрактными счетчиками, сравниваемыми со свободными параметрами реализуемого алгоритма.

Завершение параллельных процедур поиска популяцией агентов приводит к получению массивов, количество которых равно количеству изначально заданных пар элементов отображаемых онтологий. Данные массивы содержат идентификаторы элементов с наибольшим значением целевой функции и являются основой полученного результата использования знаний.

В данном подразделе описана разработка компонентной архитектуры программного приложения, позволяющего проводить вычислительные эксперименты для оценки качества и времени работы предложенных автором алгоритмов поиска, приобретения и использования знаний в процессе генерации ответа на запрос пользователя системой искусственного интеллекта и машинного обучения при обработке и анализе текстов на естественном языке.

## 4.2. Построение базы данных для хранения онтологических моделей

Представим описание базы данных для хранения предложенных в диссертации структур онтологических моделей (рис. 4.2). Построение базы данных для хранения онтографа при обработке и анализе текстов на естественном языке предполагает выполнение нескольких обязательных этапов [115, 116]:

1. *Определить структуру данных.* Описать основные сущности и отношения между ними, которые будут использоваться для представления онтологии. Например, это могут быть термины, определения, синонимы, антонимические пары, аксиомы и т.д.
2. *Разработать схему базы данных.* На основе выбранной структуры данных создать схему базы данных, которая будет хранить информацию о терминах, отношениях и аксиомах. Определить таблицы, столбцы, типы данных и ограничения для каждой сущности и отношения.
3. *Реализовать базу данных.* Выбрать подходящую СУБД (систему управления базами данных) и реализовать спроектированную схему.
4. *Импортировать данные.* Загрузить текстовые данные в построенную базу данных. Это включает словари, тезаурусы, тексты на разных языках и т.п.
5. *Определить правила и ограничения.* Установить правила и ограничения на отношения между терминами и аксиомами, чтобы обеспечить согласованность и непротиворечивость информации в базе данных.

Правила и ограничения при построении базы данных представляют собой набор условий (рис. 4.3), которые определяют, как данные могут быть связаны друг с другом, какие операции можно выполнять над ними и как они должны быть представлены. Они помогают обеспечить целостность и безопасность данных, а также их корректное использование [115, 116]. Например, ограничение может быть установлено на количество символов в поле, уникальность значений в столбце или отношение между таблицами.

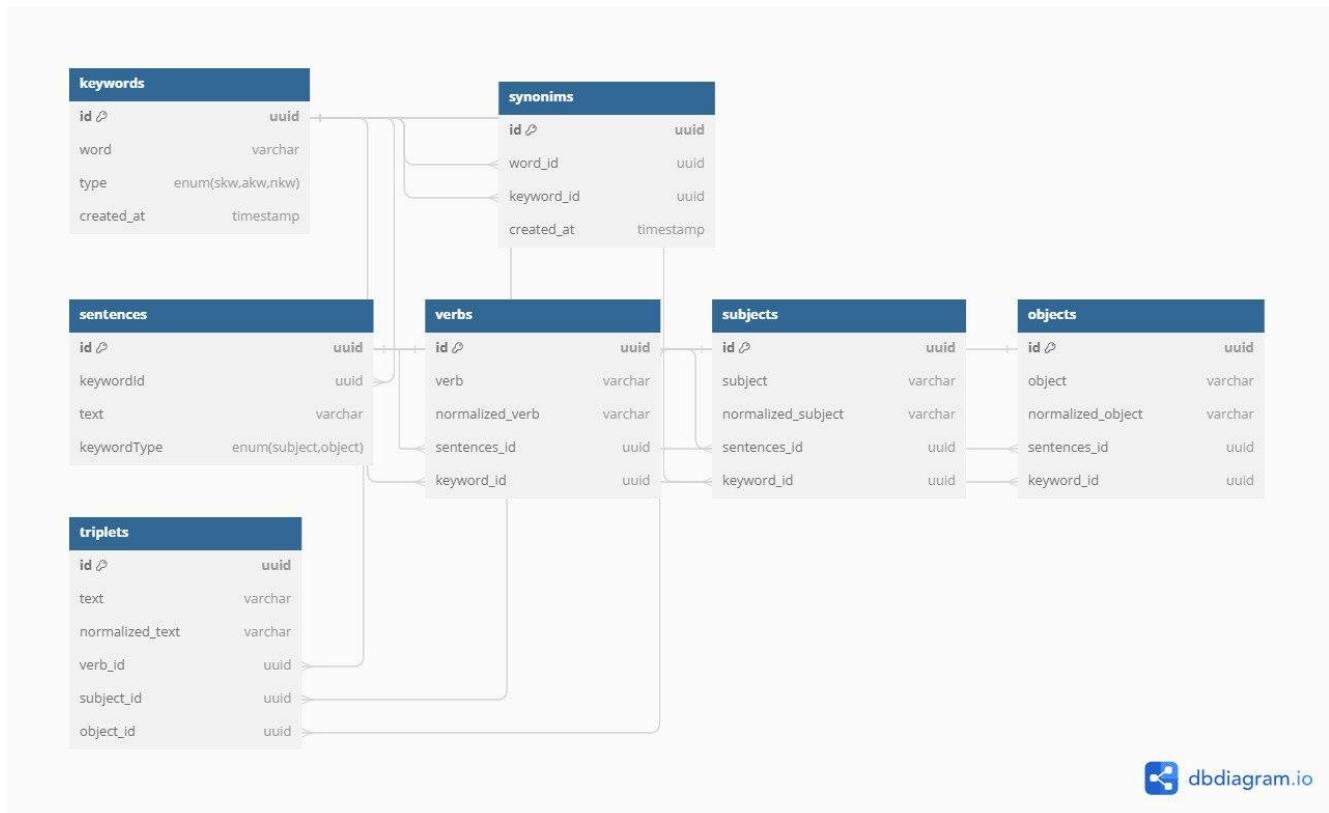


Рисунок 4.2 – Модель базы данных

Правила могут определять, какие действия разрешены или запрещены для пользователей, таких как доступ к определенным данным или изменение определенных параметров. В целом, правила и ограничения помогают поддерживать порядок и структуру в базе данных, предотвращая ошибки и нарушения.

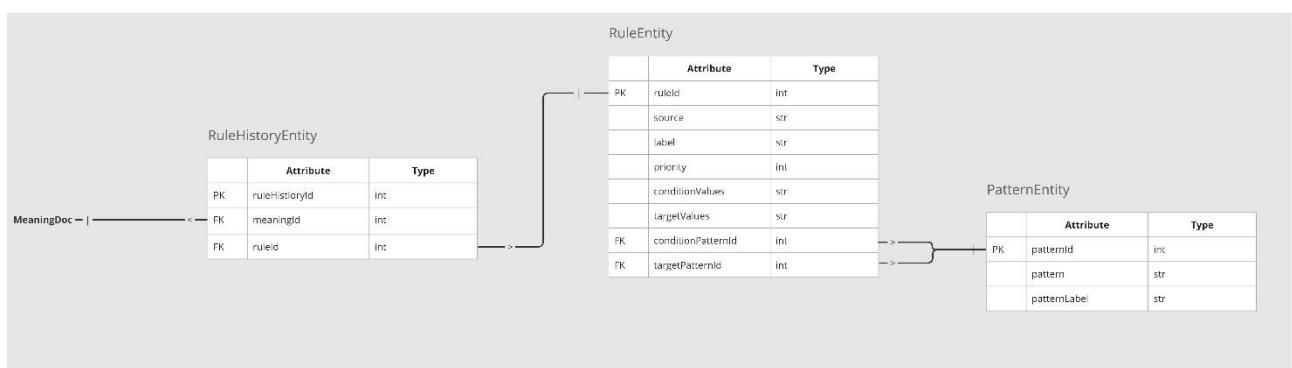


Рисунок 4.3 – Определение правил и ограничений при построении базы данных

Представим более детально описание структур таблиц базы данных для хранения построенных в диссертации онтологических моделей. Для хранения списков ключевых слов и синонимов применяется множество связей между словами. Создана таблица *keywords*, состоящая из следующих полей: *id*; *word*; *created\_at*, где *id* – уникальный идентификатор записи в формате *uuidv4*; *word* – само слово, которое хранится в формате *varchar*; *created\_at* – дата создания записи.

Для хранения связей между словами синонимами была создана таблица *synonyms*, состоящая из следующих полей: *id*; *word\_id*; *keyword\_id*; *created\_at*. Поля *id* и *created\_at* аналогичны одноименным полям таблицы *keywords*, следующие поля: *keyword\_id* – ссылка на слово; *word\_id* – ссылка на синоним. Благодаря такой архитектуре, достигается нормализация реляционных баз данных, а также исключаются возможности дубликатов.

Предложения, объекты, субъекты, предикаты и триплеты смысла хранятся в графовой базе данных *Neo4j* [117], о которой уже упоминалось ранее, в пункте 4.1. Таблица предложений имеет следующие поля: идентификатор; текст предложений; идентификатор ключевого слова, а также тип ключевого слова (объект или субъект). С таблицей предложений соединены три других таблицы: предикаты субъекты и объекты, все три таблицы имеют поле ссылки на запись в таблице предложений, поля слова и его нормализованного значения, а также ссылку на таблицу ключевых слов. Данные три таблицы являются частями таблицы триплетов смысла, которая хранит ссылки на записи в этих таблицах, сам триплет и его нормализованное значение. Реализовано горизонтальное масштабирование с помощью шардирования как в *Neo4j* так и в *PostgreSQL*, например, в *PostgreSQL* шардирование реализовано в таблице *keywords*, по полю *type*. В *Neo4j* шардирование реализовано по таблице предложений в зависимости от типа ключевого слова.

На заключительном этапе формирования базы данных решены следующие задачи:

1. *Протестирована работа предложенных алгоритмов.* Проведена проверка корректности выполнения функций поиска в графе, извлечения информации, построения онтологического каркаса и т.д.
2. *Оптимизирована производительность.* Протестирована производительность базы данных и проведена ее оптимизация для работы с большими объемами данных и сложными запросами.
3. *Обеспечена безопасность.* Приняты меры для обеспечения безопасности доступа к данным, авторизации пользователей и защиты от атак на базу данных и сервер.
4. *Создан регламент поддержки и обновления.* Обеспечена возможность регулярного обновления базы данных, расширена функциональность с помощью имплементации новых моделей и алгоритмов.

В данном подразделе описаны основные этапы построения базы данных, необходимой для создания интеллектуальных систем обработки и анализа текстов на естественном языке с применением методов искусственного интеллекта и машинного обучения. В следующем подразделе описан процесс построения и обработки графовых моделей, которые позволяют получить структурированную, компактную и формализованную форму описания текстовой информации.

### **4.3. Проведение и результаты вычислительного эксперимента**

С применением созданного программного приложения был проведен вычислительный эксперимент, подтверждающий эффективность основных предложенных в работе моделей и алгоритмов поиска, приобретения и использования знаний в процессе генерации ответа на запрос пользователя системой искусственного интеллекта и машинного обучения при обработке и анализе текстов на естественном языке. Отмечено, что все разработанные в диссертации алгоритмы демонстрируют высокие по качеству результаты.

Учитывая специфику решаемой задачи поддержки принятия решений по предупреждению и ликвидации последствий ЧС, для обеспечения необходимой оперативности работы приложения критерием оценки эффективности было выбрано время отклика системы на запрос пользователя с определением соответствующих ограничений по качеству принимаемых решений. Для сравнительного анализа были выбраны известные алгоритмы роя частиц (АРЧ), обезьяньего поиска (АОП) и поиска кукушки (АПК). Все перечисленные алгоритмы, так же как и метод бактериальной оптимизации, имеют схожие с ним принципы действия [95, 96]. Преимуществом предложенного автором алгоритма перед известным алгоритмом бактериальной оптимизации является его модификация, предусматривающая применение более эффективных механизмов интенсификации поисковых процедур и диверсификации пространства поиска решений, что позволяет проводить оценку семантической близости прецедентов и анализируемых знаний с уменьшением времени генерации ответа на запрос пользователя системой искусственного интеллекта и машинного обучения при обработке и анализе текстов на естественном языке.

В процессе оценки временных характеристик работы указанных биоинспирированных алгоритмов с применением разработанного программного приложения на основе текстовой информации из открытых источников (сайтов) различных подразделений МЧС России строилась обучающая онтология.

В базовой терминологии [78, 118] ЧС делят на два основных класса: *конфликтные* и *бесконфликтные* (рис. 4.4 и 4.5).

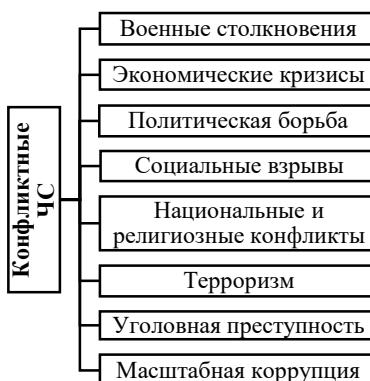


Рисунок 4.4 – Классификация конфликтных ЧС

Данные классы имеют четкую границу разделения признаков, характеристики и значения которых практически не пересекаются. Классификация бесконфликтных ЧС является более широкой и развернутой. Данную классификацию необходимо рассматривать по многим признакам, которые характеризуют явления с разных сторон [78, 118].

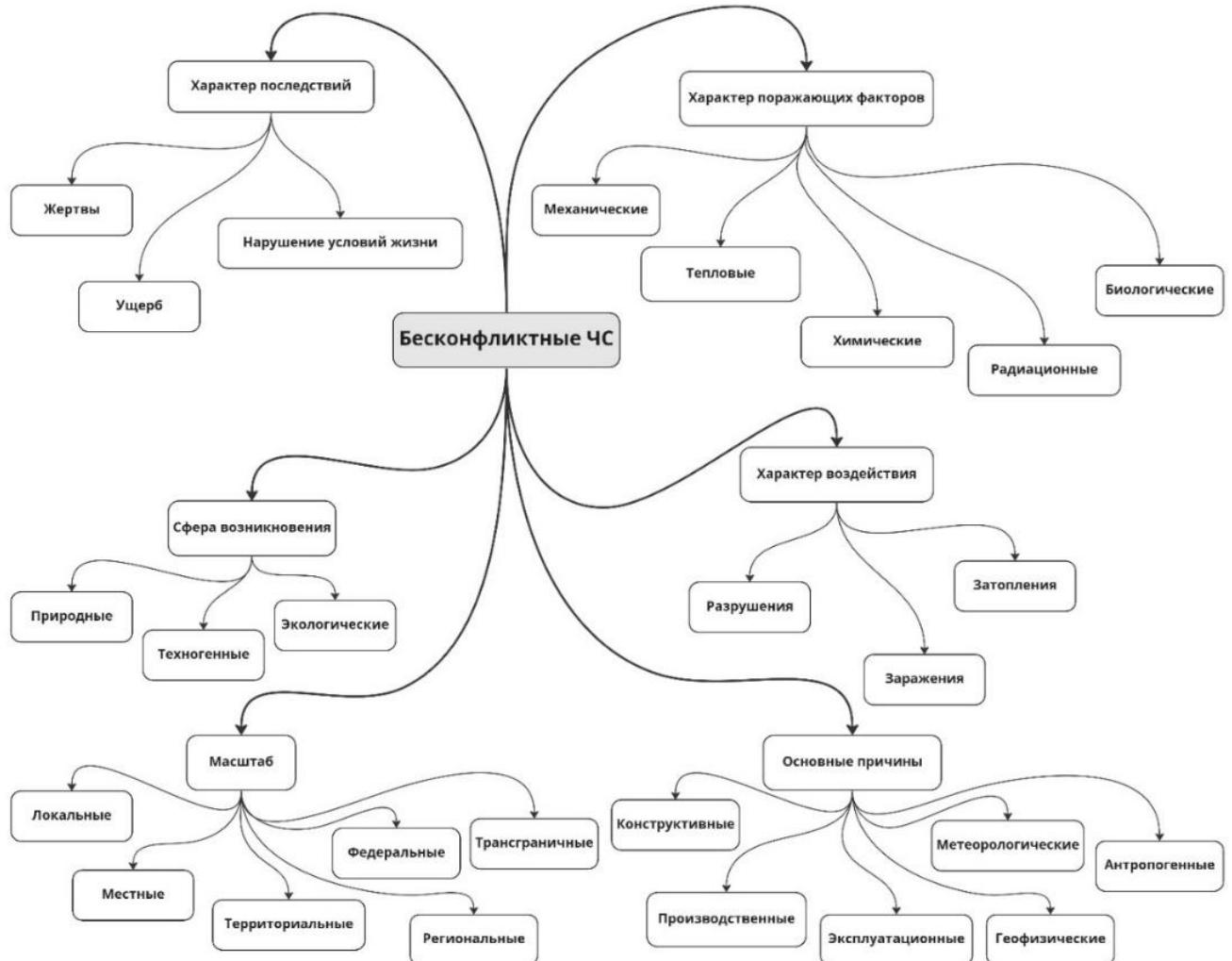


Рисунок 4.5 – Классификация бесконфликтных ЧС

Рассмотрим более подробно надкласс ЧС, связанный с определением сферы их возникновения [78, 118-122]. Раскроем в данном классе состав трёх базовых подклассов чрезвычайных ситуаций: *ЧС природного характера*; *ЧС техногенного характера*; *ЧС экологического характера*. Составы данных подклассов ЧС опишем на примере, проиллюстрированном на рисунке 4.6.

Очевидным является то, что столь значительное число классов и подклассов чрезвычайных ситуаций обеспечивает наличие большого объема инструктирующей информации по правилам безопасности и методам ликвидации последствий, требующей обработки и анализа с последующей генерацией множества различных рекомендаций.

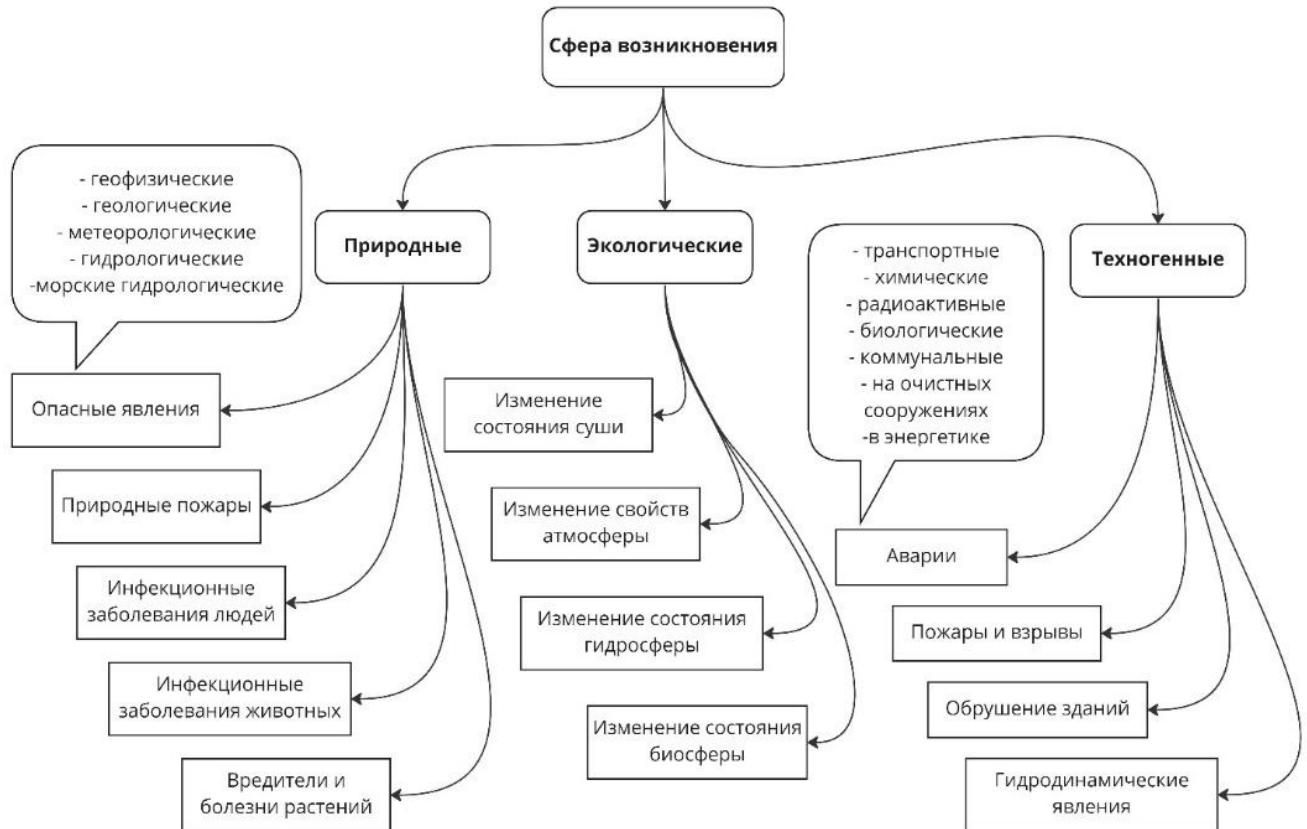


Рисунок 4.6 – Состав базовых подклассов ЧС по классу сфер возникновения

Отметим, что территория нашей страны имеет широкий спектр типов ЧС природного характера, которые могут быть вызваны различными опасными природными явлениями и процессами (землетрясения, ураганы, бури, смерчи, метели, выюги, оползни, сели, обвалы, снежные лавины, пожары, наводнения и т.д.).

Вычислительный эксперимент был реализован для подкласса природных гидрологических ЧС (на примере паводкового наводнения). Для построения обучающей онтологии precedentов использовалась информация с открытых сайтов различных региональных подразделений МЧС России. Построенная

обучающая онтология является масштабируемой и позволяет расширять число учитываемых классов и подклассов ЧС до необходимого уровня. Очевидно, что помимо информации постоянно размещенной в открытых источниках МЧС России, в обучающую онтологию прецедентов входят сведения, получаемые от различных региональных служб, учреждений и подразделений в процессе предотвращения и/или ликвидации последствий ЧС.

Вся полученная текстовая информация обрабатывалась последовательно, сначала предложенным алгоритмом поиска знаний в текстах на естественном языке, извлекающим смысловую часть предложения для использования в процессах приобретения знаний. Затем, алгоритмом приобретения знаний в текстах на естественном языке с применением множества низкоуровневых правил семантического анализа полученных смысловых паттернов, позволяющим определить основные гранулы смысла для дальнейшей реализации процессов использования знаний на основе модифицированного биоинспирированного алгоритма бактериальной оптимизации в задачах генеративного искусственного интеллекта.

В таблице 4.1 приведены несколько примеров, показывающих результаты поиска и приобретения знаний при обработке и анализе текстовых инструкций МЧС России для построения взаимосвязанной последовательности смысловых паттернов (гранул смысла).

Таким образом, полученные результаты при дальнейшем использовании в обучающей онтологии позволяют определить следующие основные гранулы (триплеты) смысла: субъекты (*угроза наводнения, население, вода, эвакуация*); предикаты / глаголы (*присутствует, сберегает, спасает, закрепляет, размещает, затапливает, закрывает, забивает, заготавливает, отключает, гасит, началась, берет, отступила, открывает*); ключевые фразы / объекты (*район проживания, продукты питания, медикаменты, домашние животные, плавучие предметы, окна, двери, трёхдневный запас продуктов, трёхдневный запас питьевой воды, запас медикаментов, запас одежды, электричество, газ, печи, паспорт, деньги*,

*телефон, строение).* Все перечисленные гранулы смысла (знаний) и отношения между ними в обучающей онтологии являются подмножеством модели *Картины Мира* для отдельной предметной области.

Таблица 4.1 – Примеры результатов поиска и приобретения знаний

№	Инструкции МЧС России	Результаты поиска и приобретения знаний
1.	Если в районе проживания присутствует угроза наводнения, заблаговременно перенесите продукты питания, ценные вещи, документы, медикаменты и домашних животных на безопасное место.	Угроза наводнения присутствует в районе проживания. Население сберегает продукты питания. Население сберегает медикаменты. Население спасает домашних животных.
2.	Если в районе проживания присутствует угроза наводнения, закрепите все плавучие предметы внутри зданий или разместите их в подсобных помещениях.	Угроза наводнения присутствует в районе проживания. Население закрепляет плавучие предметы. Население размещает плавучие предметы.
3.	Если вода затапливает строение, закройте окна, двери дома и подвала, забив их досками и укрепив мешками с песком.	Вода затапливает строение. Население закрывает окна, Население закрывает двери. Население забивает окна. Население забивает двери.
4.	Если в районе проживания присутствует угроза наводнения, подготовьте трёхдневный запас продуктов, питьевой воды, медикаментов и одежды.	Угроза наводнения присутствует в районе проживания. Население заготавливает трёхдневный запас продуктов. Население заготавливает трёхдневный запас питьевой воды. Население заготавливает запас медикаментов. Население заготавливает запас одежды.
5.	Если в районе проживания присутствует угроза наводнения, отключите электричество, газ и погасите печи.	Угроза наводнения присутствует в районе проживания. Население отключает электричество. Население отключает газ. Население гасит печи.
6.	Если эвакуация началась в районе проживания, возьмите с собой паспорт, деньги и мобильный телефон.	Эвакуация началась в районе проживания. Население берет паспорт. Население берет деньги. Население берет телефон.
7.	Если вода отступила от строения, откройте окна и двери для проветривания помещений.	Вода отступила от строения. Население открывает окна. Население открывает двери.

Отметим, что приведенные в таблице 4.1 текстовые фрагменты идентифицируются в четырёх основных контекстах: «*Угроза наводнения присутствует в районе проживания*»; «*Вода затапливает строение*»; «*Эвакуация началась в районе проживания*»; «*Вода отступила от строения*». Причем, обратим внимание на тот факт, что в некоторых случаях для разных контекстов в одной предметной области правила поведения имеют диаметрально противоположный смысл (строки 3 и 7 таблицы 4.1), что подтверждает значимость учета контекста при использовании приобретенных знаний.

В оперативную (динамическую) онтологию попадают все тексты, получаемые от населения, спасателей, работающих на местах, а также от любых других источников текущей обстановки из чатов и информационных порталов. Данная информация позволяет уточнить детали и особенности сложившейся

ситуации в разных локациях возникшей ЧС для повышения эффективности применения средств искусственного интеллекта, генерирующих инструкции и правила поведения для потерпевших и спасателей в реальном масштабе времени на основе использования знаний обучающей и оперативной онтологии. В вычислительном эксперименте текстовая информация для оперативной онтологии генерировалась случайным образом для воссоздания модели развития реальной ситуации.

В вычислительном эксперименте тестовые примеры определения семантической близости между оперативной и обучающей онтологиями решались на основе применения предложенного модифицированного алгоритма бактериальной оптимизации и следующих известных алгоритмов: роя частиц (АРЧ); обезьяньего поиска (АОП); поиска кукушки (АПК). Все перечисленные алгоритмы продемонстрировали высокое качество поиска решений, с незначительными количественными значениями отклонений от лучших вариантов. Особое внимание было уделено исследованию временных зависимостей указанных алгоритмов, так как скорость выработки правил поведения и инструкций при возникновении чрезвычайных ситуаций имеет важное значение для обеспечения своевременности принятия решений.

Для сравнения предложенного автором модифицированного алгоритма бактериальной оптимизации (МАБО) проводилось по три группы вычислительных экспериментов оценки временных зависимостей. Непосредственно для оценки МАБО исследованы следующие временные зависимости: зависимость времени работы МАБО от размера оперативной онтологии (число вершин онтографа) представлена в таблице 4.2; зависимость времени работы МАБО от количества шагов хемотаксиса представлена в таблице 4.3; зависимость времени работы МАБО от размера популяции агентов-бактерий представлена в таблице 4.4.

Таблица 4.2 – Зависимость времени работы МАБО от размера оперативной онтологии

№	Размер оперативной онтологии, ед	Длина хемотаксиса, шт	Размер популяции, ед	Время работы, мс
<b>1</b>	<b>10000</b>	<b>10</b>	<b>10</b>	<b>17.813</b>
2	100000	10	10	149.114
3	500000	10	10	1002.377
4	10000	100	100	105.738
5	100000	100	100	166.452
<b>6</b>	<b>500000</b>	<b>100</b>	<b>100</b>	<b>511.122</b>
7	10000	1000	1000	267.981
8	100000	1000	1000	717.554
9	500000	1000	1000	2967.299

Таблица 4.3 – Зависимость времени работы МАБО от длины хемотаксиса

№	Размер оперативной онтологии, ед	Длина хемотаксиса, шт	Размер популяции, ед	Время работы, мс
1	100000	10	10	149.114
2	100000	100	10	198.341
3	100000	1000	10	372.994
4	100000	10000	10	887.936
<b>5</b>	<b>100000</b>	<b>10</b>	<b>100</b>	<b>112.244</b>
6	100000	100	100	166.452
7	100000	1000	100	627.655
8	100000	10000	100	1011.771
9	100000	10	1000	455.178
10	100000	100	1000	983.481
11	100000	1000	1000	717.554
12	100000	10000	1000	1917.729

Таблица 4.4 – Зависимость времени работы МАБО от числа агентов

№	Размер оперативной онтологии, ед	Длина хемотаксиса, шт	Размер популяции агентов, ед	Время работы, мс
1	100000	10	10	149.114
<b>2</b>	<b>100000</b>	<b>10</b>	<b>100</b>	<b>112.244</b>
3	100000	10	1000	455.178
4	100000	10	10000	822.776
5	100000	100	10	198.341
6	100000	100	100	166.452
7	100000	100	1000	983.481
8	100000	100	10000	798.192
9	100000	1000	10	372.994
10	100000	1000	100	627.655
11	100000	1000	1000	717.554
12	100000	1000	10000	2779.256

Таким образом, предложенный МАБО показал наилучшие результаты для обработки и анализа оперативной онтологии размером 10 000 вершин при равной 10 длине хемотаксиса и размере популяции – 10 агентов. Для онтологии размером в 100 000 вершин при равной 10 длине хемотаксиса и размере популяции – 100 агентов. Для онтологии размером в 500 000 вершин при равной 100 длине хемотаксиса и размере популяции – 100 агентов.

При применении алгоритма обезьяньего поиска (АОП) исследованы следующие временные зависимости: зависимость времени работы АОП от размера оперативной онтологии представлена в таблице 4.5; зависимость времени работы АОП от количества локальных прыжков представлена в таблице 4.6; зависимость времени работы АОП от количества глобальных прыжков представлена в таблице 4.7. Число агентов для всех серий вычислительных экспериментов являлось равным 100.

Таблица 4.5 – Зависимость времени работы АОП от размера оперативной онтологии

№	Размер оперативной онтологии, ед	Локальные прыжки, шт	Глобальные прыжки, шт	Время работы, мс
<b>1</b>	<b>10000</b>	<b>10</b>	<b>1</b>	<b>62.898</b>
2	100000	10	1	268.127
3	500000	10	1	2111.975
4	10000	100	10	177.294
<b>5</b>	<b>100000</b>	<b>100</b>	<b>10</b>	<b>166.098</b>
<b>6</b>	<b>500000</b>	<b>100</b>	<b>10</b>	<b>1029.531</b>
7	10000	1000	100	699.024
8	100000	1000	100	2130.941
9	500000	1000	100	3301.717

Таблица 4.6 – Зависимость времени работы АОП от числа локальных прыжков

№	Размер оперативной онтологии, ед	Локальные прыжки, шт	Глобальные прыжки, шт	Время работы, мс
<b>1</b>	<b>100000</b>	<b>100</b>	<b>10</b>	<b>166.098</b>
2	100000	1000	10	775.327
3	100000	10000	10	1158.733
4	100000	100	100	1647.136
5	100000	1000	100	2130.941
6	100000	10000	100	3120.053
7	100000	100	1000	4308.442
8	100000	1000	1000	4901.373
9	100000	10000	1000	8009.599

Таблица 4.7 – Зависимость времени работы АОП от числа глобальных прыжков

№	Размер оперативной онтологии, ед	Локальные прыжки, шт	Глобальные прыжки, шт	Время работы, мс
1	100000	10	1	268.127
2	100000	10	10	477.055
3	100000	10	100	829.506
4	100000	100	1	1018.592
<b>5</b>	<b>100000</b>	<b>100</b>	<b>10</b>	<b>166.098</b>
6	100000	100	100	1647.136
7	100000	1000	1	2067.005
8	100000	1000	10	775.327
9	100000	1000	100	2130.941

Полученные наилучшие значения работы исследованных алгоритмов для каждого размера онтологии выделены в таблицах полужирным шрифтом. Таким образом, исследованный АОП показал собственные наилучшие результаты для обработки и анализа оперативной онтологии размером 10 000 вершин при равном 10 количестве «локальных» прыжков и равном 1 количестве «глобальных». Для оперативных онтологий размером в 100 000 и 500 000 вершин при равном 100 количестве «локальных» прыжков и равном 10 количестве «глобальных».

Необходимо отметить, что алгоритм обезьяньего поиска (АОП) хоть и уступает в скорости предложенному автором модифицированному алгоритму бактериальной оптимизации (МАБО), но при этом достигает близких к МАБО временных показателей работы, поэтому автор проводит в данном подразделе (табл. 4.2 – 4.7) подробное сравнение результатов вычислительного эксперимента по МАБО и АОП. Алгоритмы поиска кукушки (АПК) и роя частиц (АРЧ) значительно уступают МАБО в скорости обработки используемых знаний.

Временная сложность всех рассмотренных биоинспирированных алгоритмов является полиномиальной и в худшем случае составляет  $O(n^2)$ . Диаграмма сравнения временных характеристик представлена на рисунке 4.7.



Рисунок 4.7 – Диаграмма сравнения времени работы исследуемых алгоритмов

Из результатов вычислительного эксперимента видно, что предложенный модифицированный алгоритм бактериальной оптимизации (МАБО) показал наилучший результат по сравнению с рассмотренными известными алгоритмами роя частиц (АРЧ), обезьяньего поиска (АОП) и поиска кукушки (АПК). Высокая скорость работы предложенного алгоритма обусловлена применением более эффективных механизмов интенсификации поисковых процедур и диверсификации пространства поиска решений, что позволяет проводить оценку семантической близости прецедентов и анализируемых знаний с уменьшением времени генерации ответа на запрос пользователя системой искусственного интеллекта и машинного обучения при обработке и анализе текстов на естественном языке.

При размере оперативной онтологии более 1-го миллиона вершин время работы предложенного автором модифицированного алгоритма бактериальной оптимизации не снижается столь значительно, как в показанных выше случаях, но, тем не менее, показывает улучшение данного показателя на 5 – 7 %.

Проигрыш в скорости работы алгоритмами роя частиц (АРЧ) и поиска кукушки (АПК) по сравнению с предложенным модифицированным алгоритмом, по мнению автора, связан с высокой стохастичностью и значительной степенью

зависимости эффективности работы АРЧ и АПК от топологии соседства частиц, тогда как в применяемой архитектуре пространства решений топологические схемы не настраивались в целях экономии временных и вычислительных ресурсов.

Описанный в данном пункте вычислительный эксперимент позволяет сделать вывод о том, что предложенный автором модифицированный алгоритм бактериальной оптимизации находится на лидирующих позициях, что подтверждает необходимость продолжения исследований применимости биоинспирированных алгоритмов при решении задач использования знаний в процессе генерации ответа на запрос пользователя системой искусственного интеллекта и машинного обучения при обработке и анализе текстов на естественном языке.

Полученные в диссертации научные и практические результаты вошли в материалы отчетов по грантам РНФ № 22-71-10121 (2022-2024) и № 23-21-00089 (2023-2024), а также РФФИ № 20-01-00148 (2020-2022). Наиболее значимым по мнению автора являлось внедрение результатов данного исследования в проект РНФ № 22-71-10121 (2022-2024) по теме: «Развитие теоретических основ поддержки принятия решений для задач эвакуации при чрезвычайных ситуациях в нечетких условиях».

Также теоретические и практические результаты, полученные автором в данной диссертации, были внедрены в информационные процессы проектной организации ООО «Газэксперт плюс» (г. Краснодар) для поиска и интеграции прототипов проектных решений в газотранспортной отрасли. Была построена обучающая онтология, что позволило провести классификацию множества версий описаний различных проектных прототипов участков и узлов газовых сетей, отличающихся друг от друга типами и степенью детализации информационных моделей. В дальнейшем технические задания на проектирование новых участков и узлов газовых сетей, поступающие на вход, заносились в оперативную онтологию, а применение предложенного биоинспирированного алгоритма позволило реализовать поиск соответствующих полученным заданиям прототипов в обучающей онтологии.

#### **4.4. Выводы по разделу**

В заключительном разделе диссертации описана подготовка и проведение вычислительного эксперимента для оценки эффективности разработанных моделей и алгоритмов поиска, приобретения и использования знаний в системах искусственного интеллекта при обработке и анализе текстов на естественном языке, позволяющих повысить точность обработки входной текстовой информации, минимизировать время отклика системы на запрос пользователя и обеспечить «прозрачность» процессов получения решений.

Построена компонентная архитектура программного приложения, создаваемого для реализации разработанных в диссертации алгоритмов поиска, приобретения и использования знаний при обработке и анализе предложенных автором онтологических моделей. Приложение включает в себя функции построения онтологий, введения свободных параметров алгоритмов, получения сравнительных оценок решений и реализации механизмов поиска решений и генерации ответов на запрос пользователя.

Представлено описание технической реализации базы данных для хранения построенных в диссертации онтологических моделей. Правила и ограничения при построении базы данных представляют собой набор условий, которые определяют, как данные могут быть связаны друг с другом, какие операции можно выполнять над ними и как они должны быть представлены. Они помогают обеспечить целостность и безопасность данных, а также их корректное использование.

Описан процесс построения и обработки графовых моделей, которые позволяют получить структурированную, компактную и формализованную форму описания текстовой информации. Графовое представление используется в качестве инструмента для сокращения и упрощения предложения за счет исключения "мусорных" (бессмысленных) слов и нахождения прямых отношений между словами – носителями смысла.

Проведен вычислительный эксперимент, представлены основные результаты сравнительной оценки эффективности предложенных в работе моделей и алгоритмов. Основное внимание уделено исследованию модифицированного алгоритма бактериальной оптимизации при решении задач использования знаний.

В разработанных новых механизмах поиска решений в процессе генерации ответа на запрос пользователя системой искусственного интеллекта и машинного обучения при обработке и анализе текстов на естественном языке присутствует надлежащий баланс между скоростью сходимости и диверсификацией пространства поиска решений. Достижение данного баланса имеет важное значение для обеспечения точности и производительности биоинспирированных алгоритмов при поиске квазиоптимальных решений. Среди преимуществ предложенного модифицированного биоинспирированного алгоритма выделяются незначительный требуемый объем памяти и относительно простая настройка параметров в сочетании с невысокой трудностью реализации.

В среднем при значительном размере пространства поиска решений (более 1 миллиона вершин) предложенный автором модифицированный алгоритм бактериальной оптимизации снижает время отклика системы искусственного интеллекта на 5 – 7 %.

Созданные алгоритмы поиска, приобретения и использования знаний имеют полиномиальную временную сложность, что позволяет масштабировать их для достаточно больших размерностей информационного пространства поиска решений.

## ЗАКЛЮЧЕНИЕ

Основной целью выполненной диссертационной работы являлось повышение эффективности моделей и алгоритмов поиска, приобретения и использования знаний в системах искусственного интеллекта при обработке и анализе текстов. Под эффективностью понимается минимизация времени отклика системы на запрос пользователя при условии обеспечения «прозрачности» процессов обработки входной текстовой информации.

Созданные в представленном исследовании модели и алгоритмы поиска, приобретения и использования знаний указывают на перспективность дальнейшей разработки выбранной темы в направлении развития подходов искусственного интеллекта для построения гибких механизмов обработки и анализа текстов на естественном языке с учетом семантики и контекста информационных ресурсов.

Результаты диссертации имеют важное значение для развития технологий искусственного интеллекта, получили высокую оценку научного сообщества при аprobации и положительные рекомендации для внедрения в информационные процессы предприятий, учреждений и организаций различного профиля деятельности.

Основными результатами проведенных исследований являются следующие:

1. Проведен аналитический обзор особенностей создания систем искусственного интеллекта и машинного обучения для обработки и анализа текстов на естественном языке. Сформулированы постановки основных задач исследования;

2. Построена верхнеуровневая модель онтологии знаний, отличающаяся включением в состав ее компонентов множеств понятий с различным уровнем нормализации, что позволяет обеспечить необходимую степень детализации анализируемой текстовой информации;

3. Построена нижнеуровневая модель онтологии знаний, отличающаяся использованием структуры отношений между понятиями, детализирующими

семантику текстовой информации, что позволяет получить набор смысловых паттернов, а также проводить оценку их семантической близости;

4. Разработан алгоритм поиска знаний в текстах на естественном языке, отличающийся созданием дополнительного фильтра на выходе парсера с применением графовых моделей, что позволяет извлечь смысловую часть предложения из полученной синтаксической схемы текстовой информации для использования в процессах приобретения знаний;

5. Разработан алгоритм приобретения знаний в текстах на естественном языке, отличающийся применением множества низкоуровневых правил семантического анализа полученных смысловых паттернов, позволяющий определить основные гранулы смысла для процессов использования знаний;

6. Разработан модифицированный биоинспирированный алгоритм использования приобретенных знаний в задачах генеративного искусственного интеллекта, отличающийся улучшенными механизмами интенсификации поиска решений и процедурами выхода из локальных оптимумов, что позволило уменьшить время отклика системы искусственного интеллекта и машинного обучения на пользовательский запрос при обработке и анализе текстов на естественном языке;

7. Разработано программное приложение, реализующее разработанные алгоритмы поиска, приобретения и использования знаний в системах искусственного интеллекта и машинного обучения при обработке и анализе текстов на естественном языке, позволяющее проводить сравнительный анализ предложенных моделей и алгоритмов с существующими аналогами;

8. Выполнен вычислительный эксперимент, который подтвердил эффективность полученных решений, превосходящих результаты работы известных алгоритмов. В среднем при значительном размере пространства поиска решений (более 1 миллиона вершин) предложенный автором модифицированный алгоритм бактериальной оптимизации снижает время отклика системы искусственного интеллекта на 5 – 7 %.

В целом, предложенные в работе модели и алгоритмы поиска, приобретения и использования знаний находятся на лидирующих позициях и подтверждают высокую значимость разработки детерминированных средств обработки и анализа текстов на естественном языке для систем искусственного интеллекта и машинного обучения.

Достоверность научных результатов работы подтверждается непротиворечивостью и согласованностью с известными фактами и исследованиями в рассматриваемой области, высокой степенью сходимости теоретических результатов с данными экспериментов и определяется применением теоретических и методологических основ разработок ведущих ученых.

Теоретические и практические результаты исследований вошли в материалы отчетов по ряду научно-исследовательских работ (двух грантов РНФ и одного гранта РФФИ). Внедрение теоретических и практических результатов работы проводилось в сотрудничестве с проектной организацией ООО «Газэксперт плюс» (г. Краснодар). Полученные в работе научные результаты позволили повысить качество процедур поиска и интеграции прототипов проектных решений в газотранспортной отрасли.

Во время выполнения работы зарегистрированы результаты интеллектуальной деятельности (два свидетельства о регистрации программ для ЭВМ). Результаты диссертации применяются в процессе подготовки студентов и аспирантов Института компьютерных технологий и информационной безопасности Южного федерального университета, что подтверждено актом об использовании в учебном процессе.

Таким образом, при выполнении диссертационного исследования решены все поставленные задачи, цель выполнения данной работы достигнута. Построены модели и разработаны алгоритмы поиска, приобретения и использования знаний при обработке и анализе текстов, которые направлены на решение актуальной научной задачи, связанной с необходимостью повышения эффективности функционирования систем генеративного искусственного интеллекта, где под

эффективностью понимается минимизация времени отклика системы на запрос пользователя при условии обеспечения «прозрачности» процессов обработки входной текстовой информации, что имеет важное значение для развития информатики.

Дальнейшим направлением исследований является масштабирование предложенных в данной работе онтологических моделей отдельных предметных областей в общую «Модель Мира», что позволит затем перейти к разработке универсальных алгоритмов и методов искусственного интеллекта для поиска, приобретения и использования знаний в глобальном информационном пространстве.

## СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

AKW – Actual Key Word;

CW – Clarifying Word;

CYK – Cocke-Younger-Kasami;

FMTI – Foundational Model Transparency Index;

IST – Intra-Cluster Similarity;

KW – Key Word;

LLM – Large Language Model;

MW – Meaningful Word;

NKW – Normalized Key Word;

NLP – Natural Language Processing;

NP – Noun Phrase;

SKW – Search Key Word;

SVM – Support Vector Machine;

TF-IDF – Term Frequency – Inverted Document Frequency;

TM – Text Mining;

VP – Verb Phrase;

АОП – алгоритм обезьяньего поиска;

АПК – алгоритм поиска кукушки;

АРЧ – алгоритм роя частиц;

АФП – анализ формальных понятий;

БД – база данных;

БО – бактериальная оптимизация;

ИИ – искусственный интеллект;

ИИС – интеллектуальная информационная система;

МАБО – модифицированный алгоритм бактериальной оптимизации;

ОП – обезьяний поиск;

ЧС – чрезвычайная ситуация.

## СПИСОК ЛИТЕРАТУРЫ

1. Мусаев, А.А. Обзор современных технологий извлечения знаний из текстовых сообщений / А.А. Мусаев, Д.А. Григорьев // Компьютерные исследования и моделирование. – 2021. – Т. 13. – № 6. – С. 1291-1315.
2. Наумов, В.Н. Анализ данных и машинное обучение. Методы и инструментальные средства. / В.Н. Наумов. – СПб.: ИПЦ СЗИУ РАНХиГС. – 2020. – 260 с.
3. Пимешков, В.К. Методы извлечения знаний из естественно-языковых текстов / В.К. Пимешков, М.Г. Шишаев // Труды Кольского научного центра РАН. Серия: Технические науки. – 2022. – Т. 13. – № 2. – С. 31-45.
4. Белякова, А.Ю. Обзор задачи автоматической суммаризации текста / А.Ю. Белякова, Ю.Д. Беляков // Инженерный вестник Дона. – 2020. – Т. 10. – С. 142-159.
5. Yang, Y. A Survey of Information Extraction Based on Deep Learning / Y. Yang [et al.] // Applied Sciences. – 2022. – Vol. 12. – № 19. – P. 9691.
6. Барсегян, А. Технологии анализа данных: Data Mining, Text Mining, Visual Mining, OLAP // А. Барсегян, М. Куприяноа, В. Степаненко, И. Холод. – 2-е изд. – БХВ-Петербург. – 2008.
7. Кравченко, Д.Ю. Математическое описание процесса поддержки принятия решений при оценке семантической близости знаний в конкретизированной модели онтологии / Д.Ю. Кравченко, Ю.А. Кравченко, В.В. Курейчик, В.В. Марков // Современные компьютерные технологии: материалы II научно-методической конференции НПР. – Таганрог: Изд-во ЮФУ. – 2021. – С. 25-28.
8. Хорошевский, В.Ф. Семантические технологии: ожидания и тренды / В.Ф. Хорошевский // Открытые семантические технологии проектирования интеллектуальных систем. – 2012. – № 2. – С. 143-158.
9. Курейчик В.В. Интеллектуальные системы: эволюция моделей и методов приобретения, управления и передачи знаний: монография / В.В. Курейчик, Ю.А. Кравченко, С.И. Родзин // Чебоксары: Среда, 2023. – 192 с.

10. Корогодин, В.И. Информация как основа жизни / В.И. Корогодин, В.Л. Корогодина. – Дубна: Издательский центр «Феникс», 2000. – 208 с.
11. Appelt, D.E. The common pattern specification language / D.E. Appelt // Technical report. SRI International, Artificial Intelligence Center. – 1998.
12. Cunningham, H. A framework and graphical development environment for robust NLP tools and applications / H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan // Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics. – 2002. – Р. 168-175.
13. Большая Е.И. Язык лексико-синтаксических шаблонов LSPL: опыт использования и пути развития / Е.И. Большая // Программные системы и инструменты: тематический сборник. – 2014. – 15. – С. 15-26.
14. Kluegl, P. UIMA Ruta: Rapid development of rule-based information extraction applications / P. Kluegl, M. Toepfer, Ph.-D. Beck et al. // Natural Language Engineering. – 2016. – Vol. 22. – Issue 1. – P. 1-40.
15. Starostin, A.S. A production system for information extraction based on complete syntactic semantic analysis / A.S. Starostin, I.M. Smurov, M.E. Stepanova // Papers from the Annual International Conference "Dialogue". – 2014. – P. 659-667.
16. Ebrahimi, J. Chain based RNN for relation classification / J. Ebrahimi, D. Dou // Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. – 2015. – P. 1244-1249.
17. Xu, K. Semantic relation classification via convolutional neural networks with simple negative sampling / K. Xu, Y. Feng, S. Huang, D. Zhao // Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. – 2015. – P. 536-540.
18. Nguyen, T.H. Relation extraction: Perspective from convolutional neural networks / T.H. Nguyen, R. Grishman // Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing. – 2015. – P. 39-48.
19. Banko, M. Open information extraction from the web / M. Banko, M. J. Cafarella, S. Soderland et al. // Proceedings of the 20th International Joint Conference on Artificial Intelligence. – 2007. – P. 2670-2676.

20. Thilagavathi, K. A survey on text mining techniques / K. Thilagavathi, V. Shanmuga // Int. J. Adv. Res. Comput. Sci. Robot. – 2014. – Vol. 2. – Issue 10. – P. 41-50.
21. d'Amato, C. Mining the Semantic Web with Machine Learning: Main Issues that Need to Be Known / C. d'Amato // Reasoning Web. Declarative Artificial Intelligence. – 2022. – Vol. 13100. – P. 76-93.
22. Minaee, S. Deep Learning Based Text Classification: A Comprehensive Review / S. Minaee, N. Kalchbrenner, E. Cambria et al. // ACM Computing Surveys. – 2021. – Vol. 54. – Issue 3. – P. 1-40.
23. Allahyari, M. A brief survey of text mining: Classification, clustering and extraction techniques / M. Allahyari et al. // arXiv:1707.02919. – 2017.
24. Breit, A. Combining Machine Learning and Semantic Web: A Systematic Mapping Study / A. Breit, L. Waltersdorfer, F. Ekaputra et al. // ACM Computing Surveys. – 2023. – Vol. 55. – Issue 14. – P. 1-41.
25. Bommasani, R. The Foundation Model Transparency Index / R. Bommasani, K. Klyman, S. Longpre, S. Kapoor, N. Maslej, B. Xiong, D. Zhang, P. Liang // arXiv:2310.12941. – 2023.
26. Agarwal, M. An Overview of Natural Language Processing / M. Agarwal // International Journal for Research in Applied Science and Engineering Technology. – 2019. – No. 7. – P. 2811-2813.
27. Sesen, M. B. Natural language processing of financial news / M. B. Sesen, Y. Romahi, V. Li // Big Data and Machine Learning in Quantitative Investment. – 2019. – P. 185.
28. Нгуен, Б. Н. Классификация текстов на основе оценки семантической близости терминов / Б.Н. Нгуен, А.Ф. Тузовский // Известия Томского политехнического университета. – 2012. – Т. 320. – № 5. – С. 43-48.
29. Bagheri, E. The State of the Art in Semantic Relatedness: a Framework for Comparison / E. Bagheri, F. Ensan, Y. Feng, J. Jovanovic // The Knowledge Engineering Review. – 2017. – Vol. 32. – P. 1-30.

30. Ефремова, О.А. Онтологическая модель интеграции разнородных по структуре и тематике пространственных баз данных в единую региональную базу данных / О.А. Ефремова, С.В. Павлов // Онтология проектирования. – 2017. – Т. 7. – №3 (25). – С. 323-333.
31. Jing, J. Information Extraction from Text // Mining Text Data. Springer. – 2012. – 524 p.
32. Богатырев, М.Ю. Извлечение фактов из текстов естественного языка с применением концептуальных графовых моделей / М.Ю. Богатырев // Известия ТулГУ. – Технические науки. – 2016. – Вып. 7. – Ч. 1. – С. 198-208.
33. Биркгоф, Г. Теория решеток. / Г. Биркгоф. – М.: Наука. – 1984. – 284 с.
34. Zeng, D. Relation classification via convolutional deep neural network / Daojian Zeng, Kang Liu, Siwei Lai et al. // Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. – 2014. – P. 2335-2344.
35. Kravchenko, D. Algorithm for Optimization of Keyword Extraction Based on the Application of a Linguistic Parser / D. Kravchenko, Yu. Kravchenko, A. Mansour, J. Mohammad, N. Pavlov // Informatics and Automation. – 2024. – Vol. 23. – Issue 2. – P. 467-494.
36. Brown, T. Language models are few-shot learners / T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. // Advances in neural information processing systems. – 2020. – Vol. 33. – P. 1877-1901.
37. Bova, V.V. Simulation of the semantic network of knowledge representation in intelligent assistant systems based on ontological approach / V.V. Bova, Yu.A. Kravchenko, E.V. Kuliev, S.I. Rodzin // Communications in Computer and Information Science this link is disabled. – 2021. – 1396 CCIS. – P. 241–252.
38. Manning, C.D. A fast and accurate dependency parser using neural networks / C.D. Manning // Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). – 2014. – P. 740-750.

39. Goldberg, Y. Simple and accurate dependency parsing using bidirectional LSTM feature representations / Y. Goldberg // *Transactions of the Association for Computational Linguistics*. – 2016. – Vol. 4. – P. 313-327.
40. Kulmizev, A. Deep Contextualized Word Embeddings in Transition-Based and Graph-Based Dependency Parsing – A Tale of Two Parsers Revisited / A. Kulmizev, M. de Lhoneux, J. Gontrum, E. Fano, J. Nivre // arXiv preprint: 07397. – 2019.
41. Brown, T. Language models are few-shot learners / T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. // *Advances in neural information processing systems*. – 2020. – Vol. 33. – P. 1877-1901.
42. Zhang, Y. A tale of two parsers: Investigating and combining graph-based and transition-based dependency parsing / Y. Zhang, S. Clark // *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. – 2008. – P. 562-571.
43. Vasiliev, Y. Natural language processing with Python and SpaCy: A practical introduction. / Y. Vasiliev // No Starch Press. – 2020.
44. Qi, P. Stanza: A Python natural language processing toolkit for many human languages / P. Qi, Y. Zhang, J. Bolton, C.D. Manning // arXiv preprint arXiv: 07082. – 2020.
45. Gardner, M. Allennlp: A deep semantic natural language processing platform / M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. Liu, M. Peters, M. Schmitz, L. Zettlemoyer // arXiv preprint arXiv: 07640. – 2018.
46. Yamada, H. Statistical dependency analysis with support vector machines / H. Yamada, Y. Matsumoto // *Proceedings of the eighth international conference on parsing technologies*. – 2003. – P. 195-206.
47. Peng, Y. Deep Learning-Empowered Semantic Communication Systems with a Shared Knowledge Base / Y. Peng, C. Yang, K. Xin, L. Ying-Chang // *IEEE Transactions on Wireless Communications*. – 2024. – Vol. 23. – Issue 6. – P. 6174-6187.
48. Kim, G. Language models can solve computer tasks / G. Kim, P. Baldi, S. McAleer. – 2023.

49. Liu, B. LLM+P: Empowering large language models with optimal planning proficiency / B. Liu, Y. Jiang, X. Zhang, Q. Liu, S. Zhang, J. Biswas, P. Stone. – 2023.
50. Pei, W. An effective neural network model for graph-based dependency parsing / W. Pei, T. Ge, B. Chang // Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). – 2015. – P. 313-322.
51. McDonald, R. Online large-margin training of dependency parsers / R. McDonald, K. Crammer, F. Pereira // Proceedings of the 43rd annual meeting of the association for computational linguistics (ACL'05). – 2005. – P. 91-98.
52. Eisner, J. Three new probabilistic models for dependency parsing: An exploration / J. Eisner // arXiv preprint cmp-lg/ 9706003. – 1997.
53. Tenney, I. BERT rediscovers the classical NLP pipeline / I. Tenney, D. Das, E. Pavlick // arXiv preprint arXiv: 05950. – 2019.
54. Hewitt, J. A structural probe for finding syntax in word representations / J. Hewitt, C.D. Manning // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). – 2019. – P. 4129-4138.
55. Dozat, T. Deep biaffine attention for neural dependency parsing / T. Dozat, C.D. Manning // arXiv preprint arXiv: 01734. – 2016.
56. Ackoff, Russell L. The Art of Problem Solving / Russell L. Ackoff // University of Pennsylvania, A Wiley-Interscience Publication, John Wiley & Sons. – New York. – 1978.
57. Cheng, P. Attending to entities for better text understanding / P. Cheng, K. Erk // arXiv preprint arXiv: 1911.04361. – 2019.
58. Saju, C.J. A survey on efficient extraction of named entities from new domains using big data analytics / C. J. Saju, A. Shaja // ICRTCCM. – 2017. – P. 170–175.
59. Mao, X. Automatic keywords extraction based on co-occurrence and semantic relationships between words / X. Mao, S. Huang, R. Li, L. Shen // IEEE Access. – 2020. – V. 8. – P. 117528-117538.

60. Kravchenko, D.Y. Architecture and Method of Integrating Information and Knowledge on the Basis of the Ontological Structure / D.Y. Kravchenko, Y.A. Kravchenko, I.O. Kursitys // Advances in Intelligent Systems and Computing. – 2018. – Vol. 658. – P. 93-103.
61. Kravchenko, D.Y. Ontological Approach for Designing a Multi-agent Behavior Model in the Internet Environment / D.Y. Kravchenko, Yu.A. Kravchenko, I.O. Kursitys // Journal of Physics: Conference Series. – 2019. – Vol. 1333. – № paper 032043.
62. Ефремова, О.А. Онтологическая модель интеграции разнородных по структуре и тематике пространственных баз данных в единую региональную базу данных / О.А. Ефремова, С.В. Павлов // Онтология проектирования. – 2017. – Т. 7. – №3 (25). – С. 323-333.
63. Kozierkiewicz-Hetmanska, A. The Knowledge Increase Estimation Framework for Ontology Integration on the Concept Level / A. Kozierkiewicz-Hetmanska, M. Pietranik // Journal of Intelligent and Fuzzy Systems. – 2017. – Vol. 32. – P. 1161-1172.
64. Кравченко, Д.Ю. Модель онтологии знаний для интеллектуальных систем обработки и анализа текстов / Д.Ю. Кравченко // Известия ЮФУ. Технические науки. – 2024. – № 2 (238). – С. 38-50.
65. Brandt, S. Ontology-Based Data Access with a Horn Fragment of Metric Temporal Logic / S. Brandt, R. Kontchakov, V. Ryzhikov et al. // 31st Conference on Artificial Intelligence. – 2017. – Vol. 31. – P. 1-17.
66. Eiter, T. Spatial Ontology Mediated Query Answering over Mobility Streams / T. Eiter, J. X. Parreira, P. Schneider // The Semantic Web - 14th International Conference. – 2017. – Part I. – Vol. 10249. – P. 219-237.
67. Кравченко, Д.Ю. Метод автоматического извлечения ключевых слов / Д.Ю. Кравченко, Ю.А. Кравченко, А.М. Мансур, Ж.Х. Мохаммад // Труды международного научно-технического конгресса «Интеллектуальные системы и информационные технологии – 2022» («ИС & ИТ-2022», «IS&IT'22»). Научное издание. – Таганрог: Изд-во Ступина С.А., Т.1. – 2022. – С. 90-97.

68. Кравченко, Д.Ю. Модифицированный метод построения семантического представления текста на основе методов кластеризации и взвешивания терминов / Д.Ю. Кравченко, Ю.А. Кравченко, А.М. Мансур, Ж.Х. Мохаммад // Труды XII международной научно-технической конференции «Технологии разработки информационных систем (ТРИС-2022)». – Таганрог: 2022. – С. 94-100.
69. Кравченко, Д.Ю. Модифицированный биоинспирированный метод поддержки принятия решений по предупреждению и ликвидации последствий чрезвычайных ситуаций / Д.Ю. Кравченко, Е.М. Герасименко, В.В. Курейчик, Э.В. Кулиев, Ю.А. Кравченко, С.И. Родзин // Информационные технологии. – 2023. – Т. 29. – № 8. – С. 423-436.
70. Варшавский, П.Р. Моделирование временных зависимостей в интеллектуальных системах поддержки принятия решений на основе прецедентов / П.Р. Варшавский, А.П. Еремеев, И.Е. Куриленко // Information technologies and knowledge. – 2012. – Vol. 6. – № 3. – P. 227-239.
71. Ганичева, А.В. Дискурсный метод распознавания структурированности текстов / А.В. Ганичева, А.В. Ганичев // Мир лингвистики и коммуникации: электронный научный журнал. – № 2. – 2016. – С. 31-38.
72. Кравченко, Д.Ю. Семантический поиск с использованием генетических операторов / Д.Ю. Кравченко, В.В. Марков, А.А. Новиков, Ю.С. Старкова // Известия ЮФУ. Технические науки. – 2017. – № 7 (192). – С. 122-133.
73. Chen, M., Joint Learning with Pre-trained Transformer on Named Entity Recognition and Relation Extraction Tasks for Clinical Analytics / M. Chen, G. Lan, F. Du, V. Lobanov // Proceedings of the 3rd Clinical Natural Language Processing Workshop Clinical NLP-EMNLP 2020, Online: Association for Computational Linguistics. – 2020. – P. 234–242.
74. Nasar, Z. Named Entity Recognition and Relation Extraction: State of the Art / Z. Nasar, S.W. Jaffry, M. Malik // ACM Computing Surveys. – 2021. – vol. 54.
75. Kravchenko, D. Computational Model of Swarm Algorithm for Optimizing Process of Keywords Extraction from Text Information Presented as Graph / D.

Kravchenko, S. Rodzin, N. Pavlov, L. Rodzina, E. Kuliev, Y. Kravchenko // 7th International Conference on Information Technologies in Engineering Education, Inforino. – IEEE. – 2024. – P. 1-10.

76. Bender, E.M. Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data / E.M. Bender, A. Koller // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. – 2020. – P. 5185-5198.

77. Kravchenko, D.Y. Harnessing key phrases in constructing a concept-based semantic representation of text using clustering techniques / D.Y. Kravchenko, Ali Mansour, Juman Mohammad, Nemury Silega, Yury Kravchenko // Lecture Notes in Computer Science. – 2023. – Vol. 14335. – P. 190-201.

78. Кравченко, Д.Ю. Поддержка принятия решений по предупреждению и ликвидации последствий чрезвычайных ситуаций на основе нечеткого метода структурирования информации / Д.Ю. Кравченко, Е.М. Герасименко, Ю.А. Кравченко, Э.В. Кулиев // Известия ЮФУ. Технические науки. – Таганрог: Изд-во ЮФУ. – 2023. – № 2 (232). – С. 201-212.

79. Chhillar, N. Parsing: process of analyzing with the rules of a formal grammar / N. Chhillar, N. Yadav, N. Jaiswal // Journal Of Harmonized Research (JOHR). – 2013. – Vol. 1. – Issue. 2. – P. 73-79.

80. Fok, W.W.T. Prediction model for students' future development by deep learning and tensorflow artificial intelligence engine / W.W.T. Fok, Y.S. He, H.H. Au Yeung, K.Y. Law, KH Cheung, YY. Ai, P. Ho // 2018 4th International Conference on Information Management (ICIM). – 2018. – P. 103-106.

81. Qi, P. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages / P. Qi, Y. Zhang, Y. Zhang et al // Proceedings of the System Demonstrations of The 58th Annual Meeting of the Association for Computational Linguistics. – 2020. – P. 101-108.

82. Кравченко, Д.Ю. Алгоритм идентификации предпосылок возникновения чрезвычайных ситуаций на основе правил «если-то» / Д.Ю. Кравченко, Ю.А.

Кравченко, Э.В. Кулиев, С.И. Родзин // Научный журнал «Информатизация и связь», № 2, 2023. – С. 11-17.

83. Кравченко, Д.Ю. Перспективы интеграции технологии коллективного семантического поиска и формата rdf / Д.Ю. Кравченко, А.А. Новиков, В.А. Козачков // 1-я Международная научно-практическая конференция «Программная инженерия: методы и технологии разработки информационно-вычислительных систем» (ПИИВС-2016). Сборник научных трудов. ГОУ ВПО «Донецкий национальный технический университет». – Донецк: 2016. – С. 149-152.

84. Кравченко, Д.Ю. Модель онтологии электронного учебного курса для систем дистанционного обучения / Д.Ю. Кравченко, Д.В. Лещанов, Ю.Ю. Запорожец // Студенческая наука для развития информационного общества: сборник материалов V Всероссийской научно-технической конференции. – Ставрополь: Изд-во СКФУ. – 2016. – С. 352-356.

85. Бова, В.В. Модель семантического поиска в системах управления знаниями на основе генетических процедур / В.В. Бова, В.В. Курейчик, Д.В. Лещанов // Информационные технологии. – 2017. – Т.23. – № 12. – С. 876-883.

86. Кравченко, Д.Ю. Баланс между скоростью сходимости биоэвристики и диверсификацией пространства поиска решений (на примере модели роя саранчи) / Д.Ю. Кравченко, О.Н. Родзина // Труды межд. научно-технического конгресса "Интеллектуальные системы и информационные технологии - 2023" («ИС & ИТ-2023», «IS&IT'23»). – Таганрог: Изд-во Ступина С.А. – 2023. – Т.1. – С. 208-217.

87. Kravchenko, D.Y. The Swarm Bacterial Algorithm Based on New Attractive Operators and Patterns of Agent Behavior / D.Y. Kravchenko, Yu.A. Kravchenko, E.V. Kuliev, S.I. Rodzin, L.S. Rodzina // Lecture Notes in Networks and Systems. – 2024. – Vol. 934. – Paper ID: 202183. – P. 147-168.

88. Кравченко, Д.Ю. Гибридный биоинспирированный алгоритм отображения онтологий в задачах извлечения и управления знаниями / Д.Ю. Кравченко, Ю.А. Кравченко, В.В. Марков // Известия ЮФУ. Технические науки. – Таганрог: Изд-во ЮФУ. – 2020. – № 2 (212). – С. 16-28.

89. Kureychik, V.M. Application of Swarm Intelligence for Domain Ontology Alignment / V.M. Kureychik, A. Semenova // Advances in Intelligent Systems and Computing. – 2016. – Vol. 450. – P. 261-270.
90. Borovets, Ya. Development of a Discrete Optimization Operation Solution Information Technologies Based on Swarm Intelligence / Ya. Borovets, V. Lytvyn, R. Olyvko, D. Uhryna // Technology Audit and Production Reserves. – 2018. – Vol. 6. – № 2 (44). – P. 27-32.
91. Zaporozhets, D. Parametric Optimization Based on Bacterial Foraging Optimization / D. Zaporozhets, D. Zaruba, E. Kuliev // Advances in Intelligent Systems and Computing. – 2017. – Vol. 573. – P. 54-63.
92. Maleszka, M. Particle Swarm of Agents for Heterogenous Knowledge Integration / M. Maleszka // Proc. of ICCCI. – 2017. – P. 54-62.
93. Maleszka, M. Integration Computing and Collective Intelligence / M. Maleszka, N.T. Nguyen // Expert Syst. Appl. – 2015. – Vol. 42. – P. 332-340.
94. Родзин, С.И. Состояние, проблемы и перспективы развития биоэвристик / С.И. Родзин, В.В. Курейчик // Программные системы и вычислительные методы. – 2016. – № 2. – С. 158-172.
95. Родзин, С.И. Теоретические вопросы и современные проблемы развития когнитивных биоинспирированных алгоритмов оптимизации (обзор) / С.И. Родзин, В.В. Курейчик // Кибернетика и программирование. – 2017. – № 3. – С. 51-79.
96. Карпенко, А.П. Современные алгоритмы поисковой оптимизации: учебное пособие / А.П. Карпенко. – М.: Издательство МГТУ им. Н.Э. Баумана, 2014. – 446 с.
97. Нацкевич, А.Н. Модель решения задачи кластеризации данных на основе использования бустинга алгоритмов адаптивного поведения муравьиной колонии и k-средних / А.Н. Нацкевич, Ю.А. Кравченко // Известия ЮФУ. Технические науки. – 2017. – № 7 (192). – С. 90-102.
98. Запорожец, Д.Ю. Метод интеллектуального принятия эффективных решений на основе биоинспирированного подхода / Д.Ю. Запорожец, Ю.А.

Кравченко, Э.В. Кулиев, О.А. Логинов // Известия КБНЦ РАН. – 2017. – № 6 (80). – Ч. 2. – С. 162-169.

99. Каплунов, Т.Г. Адаптивный генетический алгоритм на основе нечетких правил / Т.Г. Каплунов, В.М. Курейчик // Известия ЮФУ. Технические науки. – 2018. – № 5 (199). – С. 26-34.

100. Водолазский, И.А. Роевой интеллект и его наиболее распространённые методы реализации / И.А. Водолазский, А.С. Егоров, А.В. Краснов // Молодой ученый. – 2017. – № 4. – С. 147-153.

101. Кравченко, Д.Ю. Структуризация информации на основе комбинации генетического, роевого и обезьяньего алгоритмов / Д.Ю. Кравченко, Н.В. Кулиева, Ю.С. Новикова, М.И. Анчеков // Российская академия наук. Научный журнал. Известия КБНЦ РАН. – Нальчик: Изд-во КБНЦ РАН. – 2019. – № 5(91). – С.5-13.

102. Kuliev, E. Monkey Search Algorithm for ECE Components Partitioning / E. Kuliev, V. Kureichik, Vl. Kureichik // Journal of Physics: Conference Series. – 2018. – Vol. 1015. – № paper 042026.

103. Кулиев, Э.В. Модель адаптивного поведения обезьян для решения задачи компоновки блоков ЭВА / Э.В. Кулиев, В.В. Курейчик, Вл.Вл. Курейчик // Информатизация и связь. – 2018. – № 4. – С. 31-37.

104. Kwasnicka, H. Nature Inspired Methods and their Industry Applications-Swarm Intelligence Algorithms / H. Kwasnicka, A. Slowik // IEEE Transactions on Industrial Informatics. – 2018. – Vol. 14. – № 3. – P. 1004-1015.

105. Agarwal, M. An Overview of Natural Language Processing / M. Agarwal // International Journal for Research in Applied Science and Engineering Technology. – 2019. – No. 7. – P. 2811-2813.

106. Кравченко, Д.Ю. Программный модуль оптимизации работы классификатора при векторизации текста на основе биоэвристик / Д.Ю. Кравченко, Ю.А. Кравченко, А. Мансур, М. Жуман // Свидетельство регистрации программы для ЭВМ. – 12.12.2023. – № 2023687185.

107. Кравченко, Д.Ю. Программный модуль сегментации изображений на основе модели «хищники – травоядные» / Д.Ю. Кравченко, Д.Ю. Запорожец, В.В. Курейчик, С.И. Родзин // Свидетельство регистрации программы для ЭВМ. – 17.10.2023. – № 2023681642.
108. Миковски, М. Разработка одностраничных веб-приложений = Single Page Web Applications: JavaScript End-to-end / М. Миковски, Д. Пауэлл. – ДМК Пресс. – 2014. – 512 с.
109. Scott, E. Spa Design and Architecture: Understanding Single Page Web Applications / E. Scott. – Manning Publications Company. – 2015.
110. Информационный ресурс «OPC Foundation Launches New Working Group “OPC UA for AI”» [электронный ресурс]: официальный сайт // спецификация сервисов. Режим доступа <https://opcfoundation.org/news/press-releases/opc-foundation-launches-new-working-group-opc-ua-for-ai-revolutionizing-manufacturing-solutions/> (дата обращения 25.04.2024).
111. Bell, J. Nest.js: A Progressive Node.js Framework / J. Bell, G. Magolan, P. Housley et. al. // Bleeding Edge Press. – 2018. – 303 p.
112. Информационный ресурс «The Power of spaCy in Natural Language Processing» [электронный ресурс]: официальный сайт. Режим доступа <https://www.codersarts.com/post/spacy-library> (дата обращения 10.05.2024).
113. Информационный ресурс «The Neo4j Operations Manual» [электронный ресурс]: официальный сайт. Режим доступа <https://neo4j.com/docs/operations-manual/current/clustering/introduction/> (дата обращения 10.05.2024).
114. Информационный ресурс «СУБД PostgreSQL: почему её стоит выбрать для работы с данными и как установить» [электронный ресурс]: официальный сайт. Режим доступа <https://practicum.yandex.ru/blog/chto-takoe-subd-postgresql/> (дата обращения 10.05.2024).
115. Григорьев, Ю.А. Сравнение стратегий выборки данных для приближенной обработки запросов к большой базе данных / Ю.А. Григорьев, А.Д.

Плутенко, А.В. Бурдаков, О.Ю. Ермаков // Информационные технологии. – 2022. – Т. 28. – № 5. – С. 240-249.

116. Бутенко, Ю.И. Использование базы данных моделей структурных переводческих трансформаций для извлечения многокомпонентных терминологических единиц / Ю.И. Бутенко // Системы и средства информатики. – 2023. – Т. 33. – № 1. – С. 35-44.

117. Виноградова, М.В. Создание кластера в графовой базе данных NEO4J / М.В. Виноградова, Е.А. Алексеева, А.Э. Самохвалов // Вестник РГГУ. Серия: Информатика. Информационная безопасность. Математика. – 2022. – № 2. – С. 18-32.

118. Государственный доклад о состоянии защиты населения и территорий Российской Федерации от чрезвычайных ситуаций природного и техногенного характера в 2014 году. – М.: МЧС России, 2015. – 318 с.

119. Современные системы мониторинга и прогнозирования чрезвычайных ситуаций / под общ. ред. В.А. Пучкова // МЧС России. – М.: ФКУ ЦСИ ГЗ МЧС России, 2013. – 352 с.

120. Болов, В.Р. Применение современных технологий, методов мониторинга и прогнозирования в обеспечении системы управления в кризисных ситуациях / В.Р. Болов // Журнал-каталог «Средства спасения. Противопожарная защита. Российские инновационные системы». – 2010. – № 10.

121. Исаев, В.С. Методика оценки эффективности мероприятий по повышению устойчивости функционирования критически важных объектов и объектов жизнеобеспечения в условиях угроз террористического характера / В.С. Исаев, Ю.Д. Макиев, В.П. Малышев, А.А. Таранов, В.Л. Камзолкин // Информационный сборник. – М.: ЦСИ ГЗ МЧС России, 2010. – № 42. – С. 52-68.

122. Горбунов, С.В. Анализ технологий прогнозирования чрезвычайных ситуаций природного и техногенного характера / С.В. Горбунов, Ю.Д. Макиев, В.П. Малышев // Стратегия гражданской безопасности, проблемы и решения: Науч.-аналит. сб. – М.: 2011. – Т.1. – №1 (1) . – С. 43-53.

**ПРИЛОЖЕНИЕ № 1**

**СВИДЕТЕЛЬСТВА О ГОСУДАРСТВЕННОЙ РЕГИСТРАЦИИ  
ПРОГРАММ ДЛЯ ЭВМ**

РОССИЙСКАЯ ФЕДЕРАЦИЯ



# СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

**№ 2023681642**

## Программный модуль сегментации изображений на основе модели «хищники – травоядные»

Правообладатель: *федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южный федеральный университет» (RU)*

Авторы: *Родзин Сергей Иванович (RU), Курейчик Владимир  
Викторович (RU), Запорожец Дмитрий Юрьевич (RU),  
Кравченко Даниил Юрьевич (RU)*

Заявка № **2023680672**

Дата поступления **10 октября 2023 г.**

Дата государственной регистрации

в Реестре программ для ЭВМ **17 октября 2023 г.**

Руководитель Федеральной службы  
по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ  
Сертификат 429fb6a0f3853164ba9f96f83573b4aa7  
Владелец **Зубов Юрий Сергеевич**  
Действителен с 10.05.2023 по 02.08.2024

**Ю.С. Зубов**



РОССИЙСКАЯ ФЕДЕРАЦИЯ



## СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2023687185

Программный модуль оптимизации работы  
классификатора при векторизации текста на основе  
биоэвристик

Правообладатель: *федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южный федеральный университет» (RU)*

Авторы: *Кравченко Даниил Юрьевич (RU), Кравченко Юрий  
Алексеевич (RU), Мансур Али (RU), Мохаммад Жуман  
(RU)*

Заявка № 2023686057

Дата поступления 30 ноября 2023 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 12 декабря 2023 г.

Руководитель Федеральной службы  
по интеллектуальной собственности

документ подписан электронной подписью  
Сертификат 429b6a0f3853164ba9f96183b73b4a7  
Владелец Зубов Юрий Сергеевич  
Действителен с 10.05.2023 по 02.08.2024

Ю.С. Зубов



**ПРИЛОЖЕНИЕ № 2**

**АКТЫ О ВНЕДРЕНИИ РЕЗУЛЬТАТОВ РАБОТЫ**



ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ  
«ГАЗЭКСПЕРТ ПЛЮС»

350051, г. Краснодар, Шоссе Нефтяников, 28, офис 507  
ИНН 2310197671, КПП 230801001, ОГРН 1172375002304

Сайт: [gazekspert.ru](http://gazekspert.ru). E-mail: [gazekspert@bk.ru](mailto:gazekspert@bk.ru)

Тел.: (861) 212-69-57

Регистрационный номер члена  
саморегулируемой организации  
П-209-002310197671-0070

УТВЕРЖДАЮ

Генеральный директор  
ООО «Газэксперт плюс»

— В.И. Рустамов



АКТ

о внедрении результатов диссертационной работы на соискание ученой степени кандидата  
технических наук Кравченко Даниила Юрьевича в обществе с ограниченной ответственностью  
«Газэксперт плюс» (ООО «Газэксперт плюс»)

Комиссия в составе:

председатель комиссии — генеральный директор В.И. Рустамов;

члены комиссии: главный инженер проекта Д.Н. Ходус;

инженер 1 категории Д.В. Скрипник,

составили настоящий акт о том, что научные результаты диссертационной работы аспиранта Южного федерального университета Д.Ю. Кравченко по теме «Модели и алгоритмы поиска, приобретения и использования знаний в системах искусственного интеллекта при обработке и анализе текстов на естественном языке», представленной на соискание ученой степени кандидата технических наук, использованы в ООО «Газэксперт плюс» для решения задач поиска и интеграции прототипов проектных решений на основе классификации множества версий описаний различных участков и узлов газовых сетей, отличающихся друг от друга типами и степенью детализации информационных моделей.

В частности, были использованы следующие конкретные научные результаты кандидатской диссертации Д.Ю. Кравченко:

- низкоуровневая модель онтологии знаний;
- алгоритм поиска знаний в текстах на основе применения графовых моделей;
- алгоритм приобретения знаний в текстах на основе применения множества низкоуровневых правил семантического анализа полученных смысловых паттернов;
- модифицированный биоинспирированный алгоритм использования приобретенных знаний.

Внедренные результаты диссертационного исследования позволили повысить качество процедур поиска и интеграции прототипов проектных решений в газотранспортной отрасли.

Председатель комиссии:  
Генеральный директор

— В.И. Рустамов

Члены комиссии:  
Главный инженер проекта

— Д.Н. Ходус

Инженер 1 категории

— Д.В. Скрипник

Подписи В.И. Рустамова, Д.Н. Ходуса, Д.В. Скрипника заверяю

Начальник отдела кадров — И.А. Руденко





УТВЕРЖДАЮ

Директор ИКТИБ

Г.Е. Веселов

2024 г.

## АКТ

об использовании научных результатов диссертационной работы

Д.Ю. Кравченко «Модели и алгоритмы поиска, приобретения и использования знаний в системах искусственного интеллекта при обработке и анализе текстов на естественном языке» на соискание ученой степени кандидата технических наук

Научные результаты, полученные в диссертационной работе Д.Ю. Кравченко, использовались при выполнении гранта РНФ № 22-71-10121 «Развитие теоретических основ поддержки принятия решений для задач эвакуации при чрезвычайных ситуациях в нечетких условиях».

В частности были использованы следующие результаты кандидатской диссертации Д.Ю. Кравченко:

1. Верхнеуровневая модель онтологии знаний, позволяющая обеспечить необходимый уровень детализации спецификаций анализируемой текстовой информации;

2. Низкоуровневая модель онтологии знаний, позволяющая строить множество смысловых паттернов, а также проводить оценку их семантической близости;

3. Алгоритм поиска знаний в текстах на естественном языке, позволяющий извлечь смысловую часть предложения из полученной синтаксической схемы текстовой информации для использования в процессах приобретения знаний;

4. Алгоритм приобретения знаний в текстах на естественном языке, позволяющий определить основные гранулы смысла для процессов использования знаний;

5. Модифицированный биоинспирированный алгоритм использования приобретенных знаний в задачах генеративного искусственного интеллекта, позволяющий уменьшить время отклика системы искусственного интеллекта и машинного обучения на пользовательский запрос при обработке и анализе текстов на естественном языке.

Использование указанных теоретических результатов, полученных в диссертационной работе Д.Ю. Кравченко, позволило разработать эффективные процедуры поддержки принятия решений для задач эвакуации при чрезвычайных ситуациях в нечетких условиях.

Руководитель проекта РНФ № 22-71-10121,  
к.т.н., доцент

Е.М. Герасименко



УТВЕРЖДАЮ

Директор ИКТИБ

Г.Е. Веселов

июль

2024 г.

## АКТ

об использовании научных результатов диссертационной работы  
 Д.Ю. Кравченко «Модели и алгоритмы поиска, приобретения и использования  
 знаний в системах искусственного интеллекта при обработке и анализе текстов на  
 естественном языке» на соискание ученой степени кандидата технических наук

Научные результаты, полученные в диссертационной работе Д.Ю. Кравченко, использовались при выполнении гранта РНФ № 23-21-00089 «Эффективные биоэвристики, инспирированные животным миром, на основе выявления паттернов поведения для задач оптимизации многомерных функций и сегментации изображений».

В частности были использованы следующие результаты кандидатской диссертации Д.Ю. Кравченко:

1. Алгоритм поиска знаний в текстах на естественном языке, отличающийся применением графовых моделей для создания дополнительного фильтра на выходе парсера, что позволяет извлечь смысловую часть предложения из полученной синтаксической схемы текстовой информации для использования в процессах приобретения знаний;

2. Алгоритм приобретения знаний в текстах на естественном языке, отличающийся применением множества низкоуровневых правил семантического анализа полученных смысловых паттернов, позволяющий определить основные гранулы смысла для процессов использования знаний;

3. Модифицированный биоинспирированный алгоритм использования приобретенных знаний в задачах генеративного искусственного интеллекта, отличающийся улучшенными механизмами интенсификации поисковых процедур и диверсификации пространства поиска решений, что позволило уменьшить время отклика системы искусственного интеллекта и машинного обучения на пользовательский запрос при обработке и анализе текстов на естественном языке.

Использование указанных теоретических результатов, полученных в диссертационной работе Д.Ю. Кравченко, позволило разработать эффективные биоэвристики для решения оптимизационных задач в сфере искусственного интеллекта и машинного обучения.

Руководитель проекта РНФ № 23-21-00089,  
 к.т.н., профессор

С.И. Родзин



УТВЕРЖДАЮ  
Директор ИКТИБ

Г.Е. Веселов

июль 2024 г.

## АКТ

об использовании научных результатов диссертационной работы  
Д.Ю. Кравченко «Модели и алгоритмы поиска, приобретения и  
использования знаний в системах искусственного интеллекта при обработке  
и анализе текстов на естественном языке» на соискание ученой степени  
кандидата технических наук

Научные результаты, полученные в диссертационной работе  
Д.Ю. Кравченко, использовались при выполнении гранта РФФИ № 20-01-  
00148 «Разработка теоретических основ и основных принципов поддержки  
принятия решений при диспетчеризации в Grid- системах на основе  
природных вычислений».

В частности были использованы следующие результаты кандидатской  
диссертации Д.Ю. Кравченко:

1. Алгоритм приобретения знаний в текстах на естественном языке, позволяющий определить основные гранулы смысла для процессов использования знаний;
2. Модифицированный биоинспирированный алгоритм использования приобретенных знаний в задачах генеративного искусственного интеллекта, позволяющий уменьшить время отклика системы искусственного интеллекта и машинного обучения на пользовательский запрос при обработке и анализе текстов на естественном языке.

Использование указанных теоретических результатов, полученных в диссертационной работе Д.Ю. Кравченко, позволило разработать комплекс алгоритмов на основе природных вычислений для решения задач поддержки принятия решений при диспетчеризации в Grid- системах.

Руководитель проекта РФФИ № 20-01-00148,  
д.т.н., доцент

А.Э. Саак



УТВЕРЖДАЮ

Директор ИКТИБ

Г.Е. Веселов

2024 г.

## АКТ

об использовании в учебном процессе

Института компьютерных технологий и информационной безопасности  
(ИКТИБ)

Южного федерального университета (ЮФУ)

результатов кандидатской диссертации Д.Ю. Кравченко «Модели и алгоритмы  
поиска, приобретения и использования знаний в системах искусственного  
интеллекта при обработке и анализе текстов на естественном языке»

Мы, ниже подписавшиеся, руководитель образовательной программы (ОП) «Разработка информационных систем и web-приложений» по направлению магистратуры 09.04.01 Информатика и вычислительная техника, к.т.н., доцент Кулиев Э.В. и ученый секретарь кафедры систем автоматизированного проектирования им. Виктора Михайловича Курейчика, к.т.н., ст. преподаватель Данильченко В.И. составили настоящий акт о том, что в учебном процессе кафедры систем автоматизированного проектирования им. Виктора Михайловича Курейчика Института компьютерных технологий и информационной безопасности используются следующие результаты, полученные в кандидатской диссертации Д.Ю. Кравченко:

- Верхнеуровневая модель онтологии знаний;
- Низкоуровневая модель онтологии знаний;
- Алгоритм поиска знаний в текстах на естественном языке;
- Модифицированный биоинспирированный алгоритм использования приобретенных знаний в задачах генеративного искусственного интеллекта;
- Программное приложение для решения задач поиска, приобретения и использования знаний.

Указанные результаты используются при проведении следующих образовательных курсов в Институте компьютерных технологий и информационной безопасности: «Онтологические модели в информационных системах», «Методы и средства проектирования информационных систем и web-приложений», «Методы машинного обучения при построении информационных систем».

Внедрение в учебный процесс ряда теоретических и практических результатов диссертационной работы Кравченко Д.Ю. позволило повысить качество подготовки магистрантов.

Руководитель ОП «Разработка информационных систем и web-приложений», к.т.н., доцент

Э.В. Кулиев

Уч. секретарь каф. САПР им. В.М. Курейчика,  
к.т.н., ст. преп.

В.И. Данильченко