TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Инженерная школа информационных технологий и робототехники
Направление подготовки – 09.04.04 Программная инженерия
Отделение школы (НОЦ) – Отделение информационных технологий

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
Комбинированный подход к извлечению структурированных данных для языков со свободным порядком слов

УДК 519.1:004.422.63:811.161.1'36

Студент

Группа	ФИО	Подпись	Дата
8ПМ7И	Радишевский Владислав Леонидович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Губин Евгений Иванович	к.ф.-м.н.		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ОСГН ШБИП	Потехина Нина Васильевна			

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД ШБИП	Горбенко Михаил Владимирович	к.т.н.		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Губин Евгений Иванович	к.ф.-м.н.		

Томск – 2019 г.

Запланированные результаты обучения по программе

Код результата	Результат обучения
Общие по направлению подготовки 09.04.04 «Программная инженерия»	
Р1	Способность проводить научные исследования, связанные с объектами профессиональной деятельности
Р2	Способность разрабатывать новые и улучшать существующие методы и алгоритмы обработки данных в информационно-вычислительных системах
Р3	Способность составлять отчеты о проведенной научно-исследовательской работе и публиковать научные результаты
Р4	Способность проектировать системы с параллельной обработкой данных и высокопроизводительные системы
Р5	Способность осуществлять программную реализацию информационно-вычислительных систем, в том числе распределенных
Р6	Способность осуществлять программную реализацию систем с параллельной обработкой данных и высокопроизводительных систем
Р7	Способность организовывать промышленное тестирование создаваемого программного обеспечения
Профиль «Технологии больших данных» / «Big data solutions»	
Р8	Способность исследовать и анализировать большие данные, создавать их модели и интерпретировать структуры данных в таких моделях
Р9	Способность понимать принципы создания, хранения, управления, передачи и анализа больших данных с использованием новейших технологий, инструментов и систем обработки данных в высокопроизводительных сетях
Р10	Способность применять теорию распределенной системы управления базами данных к традиционным распределенным системам реляционных баз данных, облачным базам данных, крупномасштабным системам машинного обучения и хранилищам данных

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Инженерная школа информационных технологий и робототехники
Направление подготовки – 09.04.04 Программная инженерия
Отделение школы (НОЦ) – Отделение информационных технологий

УТВЕРЖДАЮ:
Руководитель ООП
_____ _____ / Губин Е.И.

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Магистерской диссертации

Студенту:

Группа	**ФИО**
8ПМ7И	Радишевскому Владиславу Леонидовичу

Тема работы:

Подготовка исходных данных для построения кредитного скоринга	
Утверждена приказом директора	№1436/с от 25.02.2019

Срок сдачи студентом выполненной работы:	

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	1) Разработка методики для извлечения фактов и событий из текстовых данных 2) Реализация программного инструмента на разработанной методике 3) Проектирование и создание приложения для решения задачи извлечения фактов из резюме

Перечень подлежащих исследованию, проектированию и разработке вопросов *(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).*	1) Изучение предметной области 2) Аналитический обзор существующих методов извлечения и структурирования информации из текста 3) Разработка методики решения задачи 4) Выбор подходящих инструментов для разработки решения 5) Разработка библиотеки на основе предложенной методики 6) Проектирование и создание приложения для решения задачи извлечения фактов из резюме 7) Оценка потенциала разработки с точки зрения особенностей рынка, актуальности решаемой задачи и экономической эффективности 8) Рассмотрение условий труда исполнителей настоящего проекта 9) Обсуждение результатов выполненной работы
Перечень графического материала	
Консультанты по разделам выпускной квалификационной работы	
Раздел	**Консультант**
Социальная ответственность	Горбенко Михаил Владимирович, доцент ООД ШБИП, к.т.н.
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Потехина Нина Васильевна, старший преподаватель ОСГН ШБИП
Обязательное приложение на английском языке	Диденко Анастасия Владимировна, доцент ОИЯ ШБИП, к.ф.н.
Названия разделов, которые должны быть написаны на русском и иностранном языках	
Комбинированный подход к извлечению структурированных данных для языков со свободным порядком слов (*Hybrid approach for parsing languages with a free word order*)	

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент ОИТ ИШИТР	Губин Е.И.	к.ф.-м.н		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ПМ7И	Радишевский Владислав Леонидович		

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Инженерная школа информационных технологий и робототехники
Направление подготовки – 09.04.04 Программная инженерия
Уровень образования магистратура
Отделение школы (НОЦ) – Отделение информационных технологий
Период выполнения: весенний семестр 2018 /2019 учебного года

Форма представления работы:

магистерская диссертация

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
15.02.2019	*Обзор предметной области обработки естественных языков*	*20*
15.03.2019	*Комбинированный подход для извлечения структурированных данных из текста*	*25*
15.04.2019	*Проектирование и разработка приложения по извлечению информации из резюме на основе предлагаемого подхода*	*25*
01.05.2019	*Финансовый менеджмент, ресурсоэффективность и ресурсосбережение*	*10*
16.05.2019	*Социальная ответственность*	*10*
31.05.2019	*Hybrid approach for parsing languages with a free word order*	*10*

СОСТАВИЛ:
Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Губин Евгений Иванович	к.ф.-м.н.		

СОГЛАСОВАНО:

Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Губин Евгений Иванович	к.ф.-м.н.		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8ПМ7И	Радишевскому Владиславу Леонидовичу

Школа	Инженерная школа информационных технологий и робототехники	**Отделение школы (НОЦ)**	Отделение информационных технологий
Уровень образования	Магистратура	**Направление/специальность**	09.04.04 Программная инженерия

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:	
1. *Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих*	*1. Оклад инженера – 21760;* *2. Оклад научного руководителя – 33664;*
2. *Нормы и нормативы расходования ресурсов*	*1. Месячная норма амортизации – 2,8%*
3. *Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования*	*1. Ставки налоговых отчислений во внебюджетные фонды (ст. 426 НК РФ) – 30%* *2. Районный коэффициент по г. Томску (ст. 426 НК РФ, Постановление Правительства РФ от 13.05.92. №309) – 1,3*
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. *Оценка коммерческого и инновационного потенциала НТИ*	*1. Анализ потенциальных потребителей;* *2. Анализ конкурентоспособности;* *3. SWOT-анализ.*
2. *Разработка устава научно-технического проекта*	*Формирование цели, задач и ожидаемых результатов проекта.*
3. *Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок*	*1. Планирование структуры работ проекта;* *2. Определение трудоемкости выполнения работ;* *3. Формирование бюджета;* *4. Разработка риск-стратегии.*
4. *Определение ресурсной, финансовой, экономической эффективности*	*Расчет показателя финансовой эффективности.*
Перечень графического материала *(с точным указанием обязательных чертежей):*	
1. *Сегментация рынка* 2. *Результаты анализа конкурентных систем анализа текста* 3. *Матрица SWOT разработки* 4. *Календарный план-график проекта* 5. *Бюджет затрат* 6. *Реестр рисков*	

Дата выдачи задания для раздела по линейному графику	

Задание выдал консультант:

Должность	ФИО	Подпись	Дата
Старший преподаватель ОСГН ШБИП	Потехина Нина Васильевна		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ПМ7И	Радишевский Владислав Леонидович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8ПМ7И	Радишевскому Владиславу Леонидовичу

Школа	ИШИТР	Отделение (НОЦ)	ОИТ
Уровень образования	Магистратура	**Направление/специальность**	09.04.04 Программная инженерия

Исходные данные к разделу «Социальная ответственность»:	
Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Программная библиотека для извлечения структурированных данных для языков со свободным порядком слов на основе комбинированного подхода. использования контекстно-свободных грамматик и грамматик зависимостей.
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны), правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны.	– специальные правовые нормы трудового законодательства при работе с компьютером и орг. техникой; – требования к организации рабочего места;
2. Производственная безопасность – анализ выявленных вредных и опасных факторов – обоснование мероприятий по снижению воздействия	– вредные и опасные факторы: – отклонение показателей микроклимата; – превышение уровня шума; – отсутствие или недостаток естественного света; – повышенная напряженность магнитного поля; – психофизиологический фактор.
3. Экологическая безопасность	– анализ воздействия объекта на литосферу, гидросферу и атмосферу (отходы, связанные с утилизацией вышедшего из строя ПК, люминесцентных ламп и др.); – разработка решений по обеспечению экологической безопасности.
4. Безопасность в чрезвычайных ситуациях	– возможная ЧС – пожар; – разработка мер по предупреждению пожара; – разработка действий при пожаре.

Дата выдачи задания для раздела по линейному графику	

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Горбенко Михаил Владимирович	к.т.н., доцент		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ПМ7И	Радишевский Владислав Леонидович		

РЕФЕРАТ

Выпускная квалификационная работа содержит 94 страницы, 26 рисунков, 22 таблицы и 55 источников.

Ключевые слова: обработка естественных языков, контекстно-свободные грамматики, грамматики зависимостей, извлечение информации, извлечение фактов.

Объектом исследования является текущая проблематика задачи по извлечению и структуризации информации из текстовых данных.

Целью работы является разработка методики по извлечения фактов и событий из текстов, основанной на комбинировании подходов использования правил на контекстно-свободных грамматиках совместно с анализом синтаксических деревьев грамматик зависимостей.

В процессе исследования проводился анализ текущей проблематики задачи и существующих методов решения, описание предлагаемого подхода и существующих программных инструментов для его реализации.

В результате исследования была реализована программная библиотека, позволяющая разрабатывать шаблоны для извлечения фактов из текстов на русском языке. Разработано веб-приложение для анализа документов резюме соискателей как решение частного случая задачи. Сделан вывод, что разработанная библиотека показала свою успешную применимость для решения данных задач.

В качестве дальнейших шагов планируется расширение функционала для составления шаблонов, использование готовых общедоступных тезаурусов / словарей для возможности использования синонимов, публикация проекта в открытый доступ, а также разработка банка основных шаблонов.

Области применения: Анализ различных документов и отчетов, в т.ч из открытых источников, анализ описаний товаров, различных событий в новостях по тематикам, бренд аналитика.

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

NLP – Natural Language Processing / Обработка естественных языков

IBM – International Business Machines

IE – Information Extraction / Извлечение информации

NER – Named entity recognition / Извлечение именованных сущностей

LSTM – Long short-term memory / Долгая краткосрочная память

Bi-LSTM – Bidirectional Long Short-Term Memory / Двунаправленная сеть с долгой краткосрочной памятью

CRF – Conditional Random Fields / Условные случайные поля

MLAS – Morphology-Aware Labeled Attachment Score

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

На сегодняшний день в эпоху Больших данных объемы производимой человечеством информации больше, чем когда-либо и ее количество растет с каждым днем. Человек уже не в состоянии вручную обрабатывать, анализировать, извлекать знания из неструктурированных данных, преимущественно текстовых, и передавать их по различным каналам. В связи с этим, особую актуальность приобрела задача преобразования текстов, написанных на естественном языке в структурированное представление для применения в прикладных задачах.

Под извлечением информации подразумевается поиск в слабо структурированных документах отдельных интересующих фактов. Сегодня уже существуют решения, позволяющие извлекать из текстов различные именованные сущности с приемлемым качеством, однако задача по извлечению их отношений, фактов и событий является наименее проработанной и наиболее актуальной.

Основной целью настоящей работы является разработка методики по извлечения фактов и событий из текстов, основанной на комбинировании подходов использования правил на контекстно-свободных грамматиках совместно с анализом синтаксических деревьев грамматик зависимостей. Данная методика является попыткой в решении этой задачи для языков со свободным порядком слов, в частности для русского языка.

1. Обзор предметной области обработки естественных языков

Обработка естественных языков (Natural Language Processing – NLP) является общим направлением исследований, связанным с изучением проблем понимания и анализа текстов или речи на естественном языке компьютером, для решения прикладных задач [1 – 3]. Исследователи этой области стремятся собирать и систематизировать знания о том, как люди понимают и используют естественный язык, чтобы создавать и разрабатывать методы, инструменты, системы и программы для решения задач. Область NLP опирается на ряд дисциплин, таких как: компьютерные науки, математика, лингвистика, искусственный интеллект, психология и т.д.

К наиболее распространённым задачам в NLP можно отнести: машинный перевод, автоматическое реферирование текста, распознавание и синтез речи, генерация текста [1 – 2]. Также обработка естественных языков используется для анализа текста, таких как классификация (сентиментальный анализ, фильтрация спама и др.), создание поисковых и вопрос-ответных систем, включая виртуальных собеседников, задачи по извлечению информации.

1.1 История становления обработки естественных языков

Принято считать, что история обработки естественного языка началась с 1950-х годов, после публикации Аланом Тьюрингом статьи «*Computing Machinery and Intelligence*» [4]. В ней он предложил эмпирический тест, в последствие названным Тестом Тьюринга, целью которого было определение способностей у машины совершать действия, которые не отличаются от обдуманных действий человека вместо того, чтобы рассматривать вопрос «*Могут ли машины думать?*». Суть теста заключалась в следующем: человек общается с компьютером в режиме «только текст» на естественном языке, а судья-человек наблюдает за перепиской, не видя самих участников. Задача судьи определить, кто из собеседников человек, а кто робот. Задача

компьютера – общаться с собеседником так, чтобы ввести судью в заблуждение и заставить думать, что он на самом деле человек.

Одновременно с этим, начался поиск решений задач машинного перевода. Отправной точкой стал Джорджтаунский эксперимент, который состоялся 7 января 1954 года в штаб-квартире IBM в Нью-Йорке [5]. В ходе эксперимента был показан полностью автоматический перевод более 60 предложений с русского языка на английский. Система, которая лежала в основе этого перевода была довольно проста: в ее лексиконе было всего 6 грамматических правил и внутренний словарь из 250 лексических единиц.

В эти же годы появляются первые диалоговые программы и системы, которые ведут с человеком диалог на естественном языке. Первыми виртуальными собеседники, получившими широкую огласку были компьютерные программы ELIZA [6] и SHRDLU [7], разработанные в 1966 и 1968 годах, соответственно. ELIZA – это программа–собеседник, которая, пародировала диалог с психотерапевтом. Вся ее работа заключается в перефразировании высказываний собеседника. Она брала значимые слова (глаголы и существительные) из введённого человеком предложения и подставляла их в шаблонную фразу. Например, на высказывание: «У меня болит голова», она отвечала: «Почему вы говорите, что у вас болит голова?» и т. д (рисунок 1.1). На ключевые слова «мать», «отец», «сын» ELIZA могла попросить поподробнее рассказать про семью [6]. В ходе диалога у многих людей создавалась иллюзия общения с человеком, хоть и кратковременная. В отличие от ELIZA, программа SHRDLU в качестве интерфейса принимала от пользователя обычные выражения английского языка. После получения команды она перемещала простые объекты в свой «мир блоков»: кубики, шары, конусы разных цветов и размеров. Примеры команд: «помести на», «сними с» и т. д. SHRDLU обладала внутренней памятью, запоминала историю действий, названия этих объектов и могла отвечать на вопросы в контексте своей памяти [7].

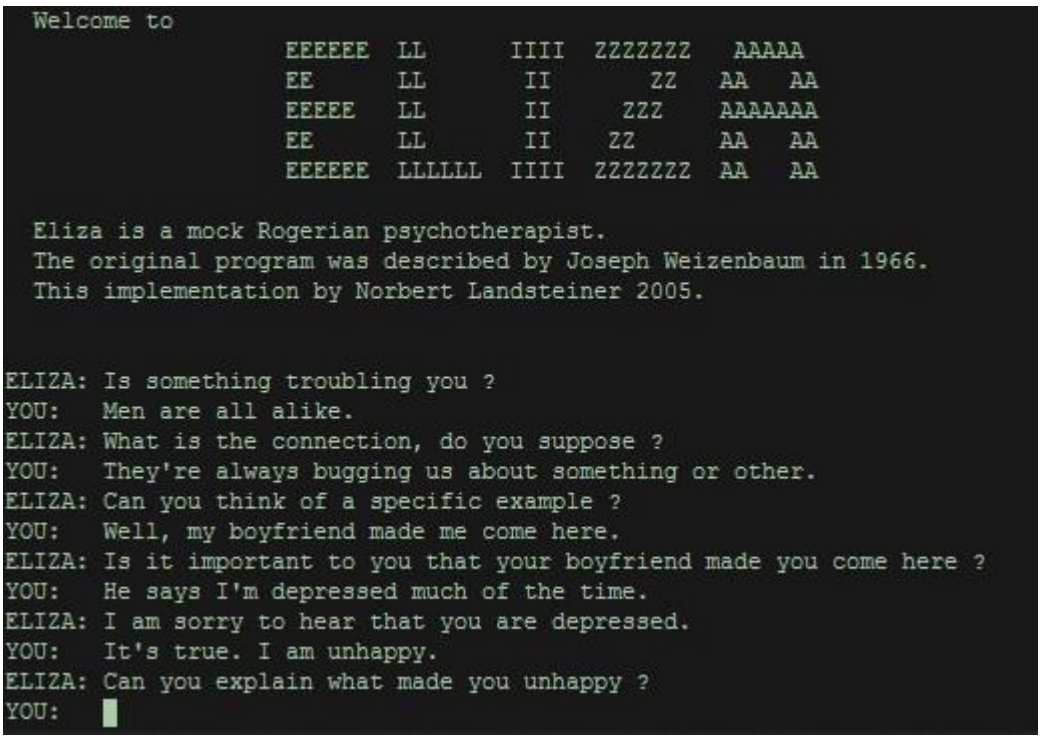

Рисунок 1.1 – Диалог с Eliza.

До 1980-х годов все системы по работе с естественными языками были основаны на сложном наборе составленных вручную правил. Однако с конца 1980-х с введением алгоритмов машинного обучения, они нашли свое применение также в задачах по обработке языков [8]. Кроме того, в то время вычислительная мощность компьютеров непрерывно возрастала по закону Мура [9].

Одной из первых и немаловажных этапов по анализу текста становиться частиречная разметка. Для задачи частиречной (part-of-speech tagging) разметки успешно нашли применения скрытые Марковские модели (CRF) [10].

Благодаря исследователям из IBM, которые разрабатывали и совершенствовали все более сложные статистические модели, системы машинного перевода достигли заметных успехов [11]. Однако модели машинного перевода требуют наличие параллельных корпусов текстов. Примерами существующих параллельных корпусов являются стенограммы заседаний ООН и Европейского Союза, переводы различных фильмов и сериалов и множество других составленных корпусов и переводов. Но все же

из-за ограниченной проработанности самих корпусов систем перевода по тематикам и размерам, актуальным вопросом остается возможность более эффективного обучения на ограниченных наборах данных [12].

Многие недавние исследования все больше касаются темы обучения без учителя [13] или обучения с подкреплением [12]. Такие алгоритмы направлены на обучение по данным, которые не были размечены вручную. В случае с обучением с подкреплением, в основном объеме неразмеченных данных используется небольшое количество размеченных. Было показано, что такой способ обучения может значительно улучшить точность обучения [12].

С 2010 года обучение признакам совместно с методами глубокого обучения получили широкое применение в NLP. Решения на основе глубокого обучения показали наилучшие результаты в таких задачах как языковое моделирование, синтаксический разбор, извлечение именованных сущностей, машинный перевод и т. д. [14 – 17]. Векторное представление слов, например Word2vec, fastText, Bert и др., позволяет сопоставлять каждому слову вектор в многомерном пространстве на основе встречающихся контекстов. Согласно предположению, которое носит название дистрибутивной гипотезы, слова с похожим смыслом будут встречаться в похожих контекстах, т. е. быть семантически близкими [18]. Вектора слов используются при кластеризации слов на текстовых корпусах, оценки семантической близости, анализе тональности и других задач [18].

1.2 Извлечение информации

Извлечение информации (Information Extraction – IE) – это идентификация, последующая или одновременная классификация и структурирование в семантические классы конкретных упоминаний, найденных в неструктурированных источниках данных, таких как текст на естественном языке [1]. Примерами извлекаемой информацией могут служить любые описания происходящих событий.

К неструктурированным источникам данных можно отнести письменный и устный текст, изображения, видео- и аудиозаписи. Такие данные не являются структурно несогласованными, а вероятнее всего, информация в них закодирована таким образом, что компьютеру сложно их интерпретировать. Предполагается, что в процессе извлечения все данные преобразуются в структурированную форму для дальнейшей возможности их обработки и анализа [1].

В процессе извлечения вся информация в текстах идентифицируется, с учетом особенностей языковой организации. Не зависимо от языка, текст состоит из целого набора составляющих его выражений и правил. Это можно рассматривать как следствие принципа композиционности [19], общего понятия лингвистической философии, который лежит в основе многих современных подходов к языку. Согласно этому принципу, значение любого сложного языкового выражения является функцией значений его составных частей. То есть, как правило, типичное предложение содержит ряд его составляющих: субъект и предикат, выражаемые подлежащим и сказуемым. Значение каждого слова в предложении, их последовательность и морфологические признаки позволяют нам понимать смысл данного предложения [19].

Несмотря на то, что семантическая информация в тексте не указывается явно с вычислительной точки зрения, тем не менее ее можно извлечь, принимая во внимание поверхностные закономерности, которые отражают ее непрозрачную внутреннюю организацию [1]. Система извлечения будет использовать набор шаблонов, которые создаются вручную, либо выводятся автоматически, позволяющих извлечь и представить информацию из текста в структурированном виде. Более подробные алгоритмы и методы будут рассмотрены позднее.

Использование термина «извлечение» подразумевает, что необходимая семантическая информация явно присутствует в лингвистической организации текста, т. е. она легко доступна в лексических элементах (словах

и группах слов), грамматических конструкциях (фразах, предложениях, временных выражениях, и т. д.), прагматическом упорядочении и риторической структуре (абзацах, главах и т. д.) исходного текста [1]. Кроме того, обнаружение знаний также возможно с помощью методов сбора статистических данных, которые работают с извлечённой из текста информацией. Во всех операциях извлечение информации часто является обязательным этапом предварительной обработки, как сбор дополнительных признаков. Например, данные, извлеченые из полицейских отчетов, могут использоваться в качестве входных, для применения модели интеллектуального анализа, выявления общих тенденций преступности или для алгоритма рассуждений на основе конкретного случая, который будет пытаться предсказывать местоположение следующего преступления, основанного на схожих случаях.

1.3 Подзадачи извлечения информации

Типичные задачи по извлечению информации включают в себя следующее [1 – 3, 20 – 34]:

- Извлечение именованных сущностей (Named entity recognition – NER). Суть задачи состоит в поиске упоминаний именованных объектов по заранее определенным категориям, таким как имена людей, организации, местоположения, значения дат и времени и т. д.

- Разрешение кореференции (Coreference resolution). Оно заключается в объединении упоминания объектов из текстовых описаний в группы. В качестве члена цепочки может выступать имя, субстантив, местоимение.

- Извлечение отношений (Relationship extraction). Задача, которая связана с поиском совместных упоминаний именованных сущностей, определении и классификации семантических отношений между ними. Например, сущности человек и организации могут быть связаны как: «директор компании», «сотрудник» и т. д.

•Извлечение атрибутов, фактов и событий (Fact / Event Extraction) – поиск и классификация различных фактов: события, мнения, отзывы, контактные данные, новости, объявления и др.

Ниже будут рассмотрены существующие подходы к решению данных задач.

1.3.1 Извлечение именованных сущностей

На сегодняшний день подавляющее число существующих систем можно разделить на те, которые основаны на грамматических правилах и с применением машинного обучения, а также гибридные подходы, совмещающие оба варианта (рисунок 2) [20 – 29]. С одной стороны, созданные вручную системы с использованием грамматик, как правило, обладают высокой точностью. Такие системы основаны на контекстно-свободных грамматиках, введенных Ноамом Хомским, в которых одни синтаксические категории могут быть описаны через другие, или через последовательность конечных терминалов (например слов в предложении) и, как правило, слева направо [4]. Однако, для написания грамматик требуются обширные словари предметной области, например, словари имен и фамилий, процесс составления которых является достаточно трудоемким и может занять более месяца работы.

Рисунок 1.2 – Пример извлекаемых именованных сущностей [5].

При использовании машинного обучения для решения задачи извлечения именованных сущностей достигается наилучший результат, если применять архитектуру нейронной сети, состоящей из рекуррентных слоев, например, слоев Bi-directional Long Short-Term Memory (Bi-LSTM), предназначенных для обработки последовательностей [20]. В отличие от обычной LSTM (Long Short-Term Memory), обработка последовательности в Bi-LSTM происходит как слева-направо, так и в обратном направлении. Это является полезным, так как в этом случае учитывается не только предшествующий контекст, но и будущий (рисунок 1.3). Кроме того, возможно применение слоя CRF, который создает матрицу перехода состояний для прогнозирования текущего тега каждого слова: является ли этот тег сущностью, если да, то какого типа [22].

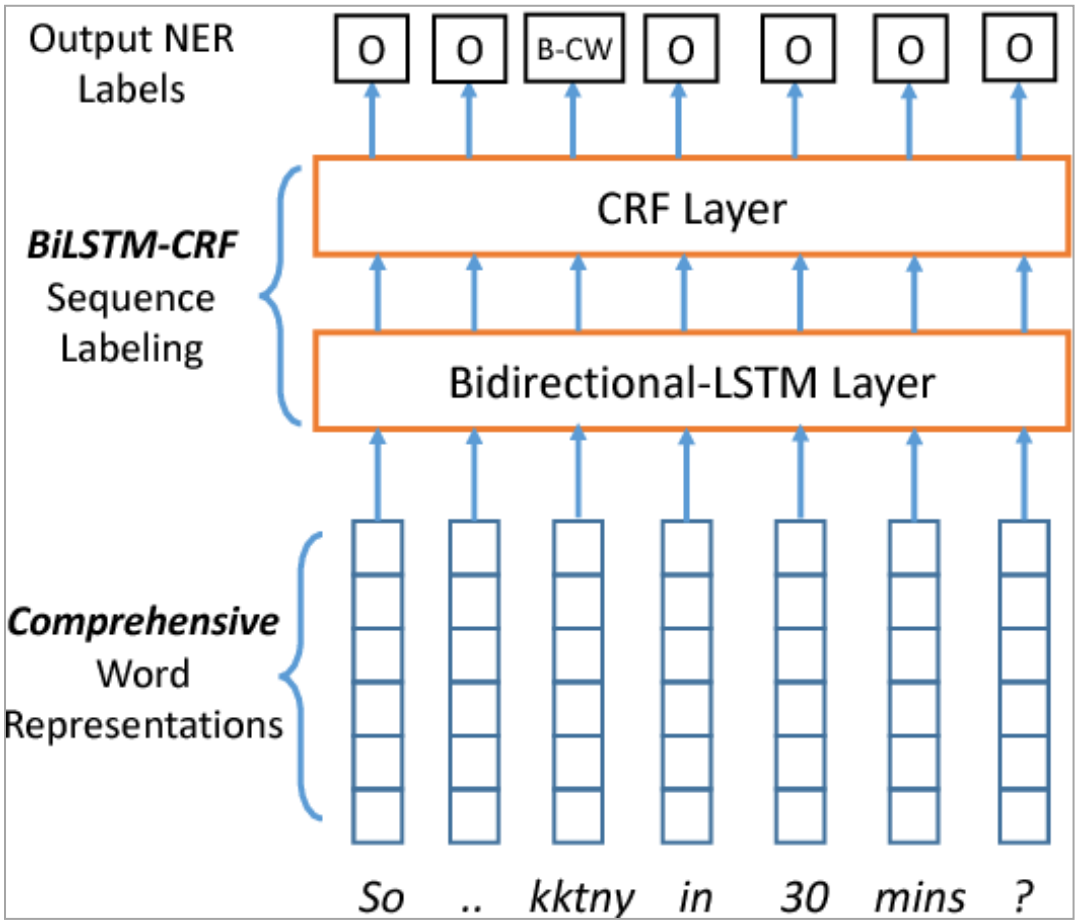

Рисунок 1.3 – Обзор архитектурного подхода для решения задачи NER [22].

На входе нейросети могут подаваться предобученные вектора слов, полученные с помощью таких моделей как Word2vec, Glove, FastText и других. Кроме того, помимо векторов слов можно использовать различные комбинации векторов символов, полученных с помощью тех же самых методов, и теги частей речи, как результат других моделей, и любой другой дополнительной информации [22].

Существуют довольно много готовых решений для NER, в основе которых схожие архитектуры: Stanford NLP [16], spaCy [23], NLTK [24] и другие. Для русского языка можно выделить такие решения как DeepPavlov [25], PullEnti [26], spaCy-ru [27]. Исследования показывают [20], что на сегодняшний момент даже самые современные системы NER разработанные для одной предметной области, плохо работают в любой другой, вне зависимости от принципа их работы. При использовании машинного обучения в локальной предметной области может не оказаться обучающей выборки и/или времени для ее создания, а также необходимого количества документов. В этом случае будет лучше использовать подход, основанный на грамматических правилах [29]. В некоторых случая имеет смысл комбинировать эти подходы [28 – 29]. Так, например, для извлечения имен и организаций использовать готовую предобученную модель, а для извлечения именованных объектов предметной области реализовать эти правила.

1.3.2 Разрешение кореференции

Разрешение кореференции является поисковой задачей, связанной с нахождением всех цепочек упоминаний, которые ссылаются на одну и ту же сущность в тексте. С ее помощью можно решить множество задач в NLP, таких как: автореферирование, вопрос-ответные системы, чатботы и задачи по извлечению информации.

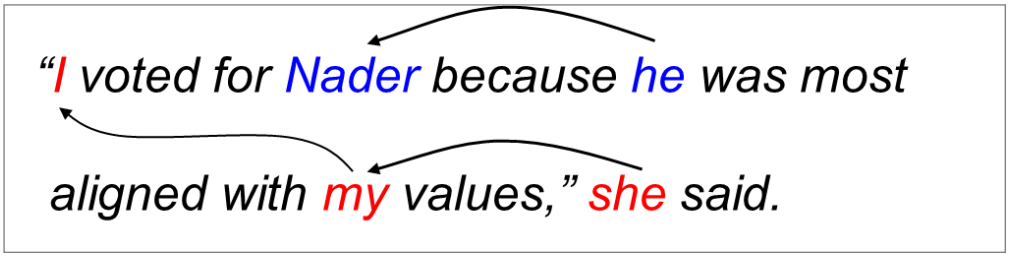

Рисунок 1.4 – Пример цепочек упоминаний сущностей.

Для создания модели, решающей данную задачу, применяются нейросети, обученные на достаточно большой выборке. Можно выделить следующие виды моделей [30 – 31]:

- Mention-Pair. Данная модель работает по принципу бинарной классификации, предсказывающей кореферентность для каждой найденной паре сущностей. Однако их применение на практике сложно осуществить. Во-первых, может нарушаться свойство транзитивности, т. е. встречаются такие упоминания, кореферирующие с двумя другими, но не связанные между собой. Во-вторых, имеет место сильная несбалансированность классов, т. к. большинство сущностей между собой не являются упоминаниями одного и того же.
- Mention-ranking. Эта модель используется для ранжирования наиболее вероятных упоминаний. Для каждого найденного упоминания попарно оценивается вероятность кореферирования с каждой предшествующей сущностью или ключевыми словами. Выбирается одно наиболее вероятное упоминание. Обучение может происходить по различным признакам: расстояние, учет векторов слов, данных морфологии (например, род).
- Entity-based. Модель выделяет кластеры упоминаний различных сущностей, местоимений и субстантивов. Каждая последующая найденная сущность сравнивается с уже предшествующими найденными кластерами. Из набора выявленных кластеров выбирается тот, у которого оценка вероятности его принадлежности будет максимальной. Если же наиболее подходящих кластеров не найдется, тогда из рассматриваемого не

отнесенного элемента создается новый кластер. Такая модель показывает более точный результат разрешения кореференции [6]. Во-первых, ее результаты не нарушают условия транзитивности; во-вторых, они менее противоречивы, так как сравниваются сущность с кластером, включающих информацию от всех элементов кластера (например, род).

Помимо подходов, основанных на машинном обучении, существуют подходы на правилах и эвристиках. Однако их использование является довольно общим решением и не привязывается к конкретным предметным областям.

1.3.3 Извлечение отношений

Задача по извлечению отношений (Relation Extraction) заключается в объединении сущностей определенного типа, например, люди, организации, локации, на ряд заранее определенных семантических категорий: «женат», «живет в», «работает в» и др. [32 – 33]. На рисунке 1.5 показаны примеры таких отношений.

Класс Принадлежность	Примеры	Тип
Персональные	мать, женат на	PERS → PERS
Организационные	директор, оф. представитель	PERS → ORG
Предметные	владеть, производить	(PERS|ORG) → OBJ
Пространственные		
Близость	рядом с	LOC → LOC
Направление	к югу от	LOC → LOC

Рисунок 1.5 – Примеры общих отношений.

Для решения данной задачи может применяться метод обучения с учителем (Supervised learning), представляющий классификатор, который принимает на вход текстовые фрагменты, пару именованных сущностей и их типы, встречающиеся во фрагменте [33]. На выходе определяется их наличие и тип, либо отсутствие отношения. В результате обучающая выборка должна

представлять набор, включающий фрагмент текста, пару сущностей, их типы и тип связи. Примеры, в которых отсутствуют связи, являются отрицательными для обучающей выборки. Помимо этих признаков могут использоваться более нетривиальные: наличие определенных синтаксических конструкций, дистанция между словами, путь в синтаксическом дереве, окно текста до первого сущности, после второй и между ними и т. д.

Кроме того, решить данную задачу можно путем применения подхода с минимальным привлечением учителя (Distant Supervision) [32]. Предполагается, что обучающая выборка в этом случае либо отсутствует, либо требуется ее расширение. Для расширения обучающей выборки используется готовая база знаний (Freebase, DBpedia), содержащая информацию о том, как связаны те или иные сущности. Далее сущности извлекаются из текста, а затем связываются с экземплярами из базы знаний и фильтруются только те предложения или фрагменты, в которых совместно упоминаются эти пары. Так генерируется обучающая выборка. При отсутствии связи в базе, экземпляры помечаются? как отрицательные примеры. Таким образом можно собирать выборку и использовать ее для дальнейшего обучения классификатора.

Метод Lightly Supervised заключается в пополнении обучающей выборки из имеющейся данных за счет выявления синтаксических шаблонов, обхватывающих встречаемые отношения и их дальнейшего обобщения. Например, если пара сущностей совместно упоминается в контексте вида «Глагол > сущность$_1$ > предлог > сущность$_2$», тогда строится предположение, что наличие такого шаблона в другом предложении будет свидетельствовать о связи этого же типа [32].

1.3.4 Извлечение атрибутов, фактов и событий

Задача по извлечению фактов представляют наибольший интерес при анализе текстовых данных и является конечным по отношению к вышеупомянутым [1]. На сегодняшний день не существует универсального подхода к ее решению в целом. Вместо этого рассматриваются частные

задачи, в зависимости от предметной области. Примерами могут быть: анализ спортивных событий из новостей (участвующие в матче команды, конечный счет, победитель); анализ политических и экономических событий: слияния и поглощения (покупатель и покупаемый, сумма сделки), смена должностей (сотрудник, старая должность, новая должность), отставки и назначения, объявления/пресс-релизы людей и компаний; анализ различных документов: судебные решения (истец, ответчик, предмет иска, исковые требования, решение); анализ резюме (желаемая заработная плата, претендуюемая должность, опыт работы, навыки). Кроме того, существует потребность по извлечению различных атрибутов, таких как: свойства товаров из их описания (производитель, модель, различные характеристики), адреса (область, город, район, станция метро, улица, дом, помещение, квартира/офис), контактные данные организаций (юридическое название, телефон, email, адрес).

Решения подобных задач можно условно разделить на 2 способа [34]:

1. Разметка данных и тренировка модели машинного обучения для извлечения сущностей и отношений аналогичным образом, как было рассмотрено в предыдущих 3-х подпараграфах. Такой подход способен дать хороший результат (точность и полноту) только при наличии подходящего качественно размеченного корпуса. Другой особенностью является то, что при возникновении ошибки невозможно вручную исправить модель, а только переобучить ее совместно с поиском, исправлением или пополнением корпуса необходимыми примерами.

2. Подход, с использованием различных эвристик и правилах, основанных на формальных грамматиках, в частности контекстно-свободных. Применяются готовые инструменты, позволяющие составлять подобные правила. В этом случае не требуется разметка корпуса, однако нужно время на составление грамматик и тематических словарей для формирование этих правил. Такой подход дает высокую точность, но

относительно невысокую полноту, которая напрямую зависит от качества проработанности правил и словарей.

1.4 Парсеры на основе контекстно-свободных грамматик

Существует несколько реализаций парсеров контекстно-свободных грамматик для русского языка. Самой популярной библиотекой, работающей с контекстно-свободными грамматиками, является NLTK [24]. Для русского среди наиболее распространенных и некоммерческих парсеров можно выделить Tomita парсер [35] и Yargy [36]. Tomita, которая разрабатывалась в компании Яндекс на протяжении многих лет, состоит из нескольких десятков тысяч строк кода. Библиотека реализована на языке C++. Tomita доступна в виде бинарного файла, однако в открытом доступе отсутствует банк готовых грамматик. В Tomita-парсере используется собственный язык для описания грамматик (рисунок 1.6).

```
#encoding "utf-8"
Born -> Verb<kwtype=born>;
City -> Noun<kwtype=city>;
Person -> AnyWord<gram="имя">;
S -> Person interp(BornFact.Person) Born "в" City interp(BornFact.Place);
```

Рисунок 1.6 – Пример синтаксиса грамматик для Tomita-парсера.

Парсер Yargy, напротив, проект с открытым исходным кодом, опубликованным на Github, и доступен в виде библиотеки Python. В связи с этим, все грамматики и словари должны описываться на языке программирования Python. Yargy использует морфологический анализатор Pymorphy, который также является библиотекой с открытым исходным кодом. Более того, у него есть несколько разработанных готовых наборов правил для извлечения таких сущностей, как имена, даты, деньги, адреса и др., которые полностью доступны в репозитории Natasha на Github [39]. На рисунке 1.7 приведен пример извлечения дат с помощью Yargy.

```
MONTH_NAME = dictionary(MONTHS)
MONTH = and_(
    gte(1),
    lte(12)
)
DAY = and_(
    gte(1),
    lte(31)
)
YEAR = and_(
    gte(1900),
    lte(2100)
)
DATE = or_(
    rule(DAY, MONTH_NAME, YEAR),
    rule(YEAR, '-', MONTH, '-', DAY),
    rule(YEAR, 'г', '.')
).named('DATE')
parser = Parser(DATE)
text = '''2015г.
18 июля 2016
2016-01-02
'''
for line in text.splitlines():
    match = parser.match(line)
    display(match.tree.as_dot)
```

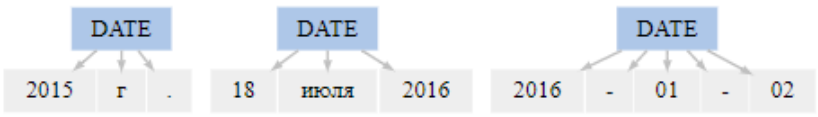

Рисунок 1.7 – Пример извлечения дат с помощью Yargy [36].

Следует отметить, что эти парсеры не показывают структуру данного предложения, а просто извлекают шаблоны, которые составляют искомый факт.

1.5 Синтаксические парсеры

Синтаксическим разбором является задача сопоставления предложения и его синтаксической структуры в виде дерева составляющих или дерева зависимостей (рисунок 1.8). Такие деревья полезны непосредственно в приложениях, таких как проверка грамматики в системах обработки текста. Например, предложение, которое не может быть проанализировано, вероятнее всего может иметь грамматические ошибки, либо является

трудночитаемыми. Как правило, деревья служат важной промежуточной стадией представления для семантического анализа и, таким образом, играют важную роль в таких приложениях, как вопрос-ответных системах и задачах по извлечению информации [7]. Поскольку существует два основных подхода к представлению синтаксической структуры любого предложения, они также используются в различных инструментах его анализа.

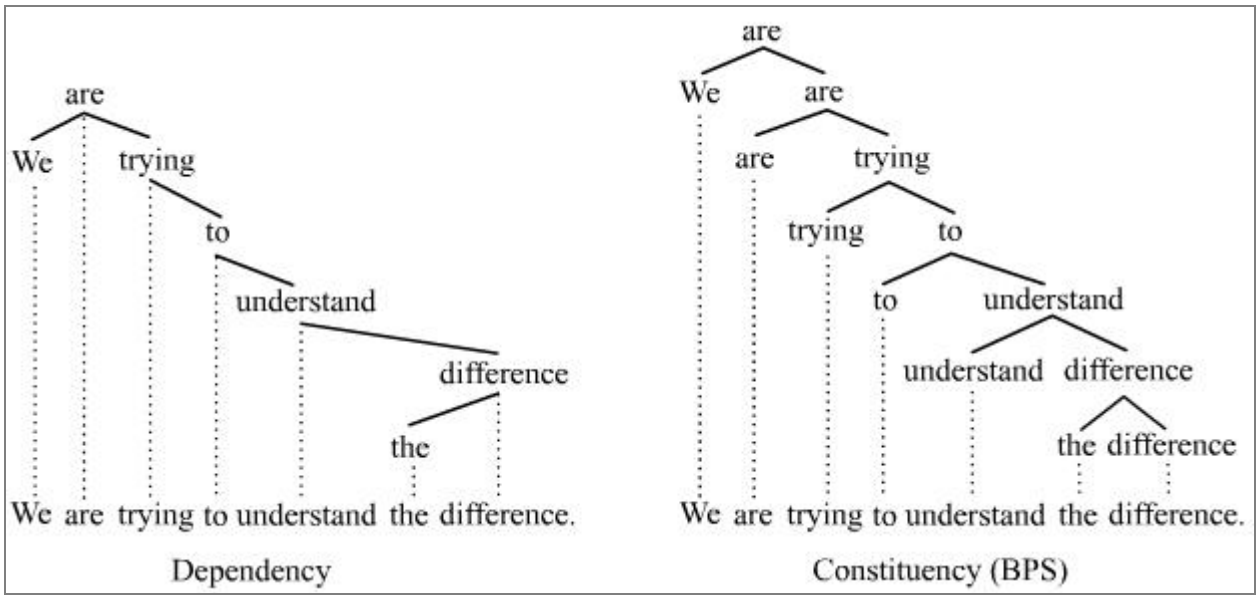

Рисунок 1.8 – Наглядное сравнение дерева зависимостей (слева) и дерева составляющих (справа) [37].

Грамматика зависимостей представляет куда больший интерес для языков со свободным порядком слов, в частности для русского языка, поскольку в рамках этой модели слова представлены в виде иерархии компонентов, между которыми установлены зависимости. Благодаря этим зависимостям есть возможность определять как связаны те или иные сущности и упоминания. Рассмотрим модели построения деревьев зависимостей.

Анализатор грамматик зависимостей выделяет грамматическую структуру предложения, устанавливая связь между «главными» словами и словами, которые являются зависимыми по отношению к ним. Результатом его работы является синтаксическое дерево, пример которого показан на рисунке 1.9.

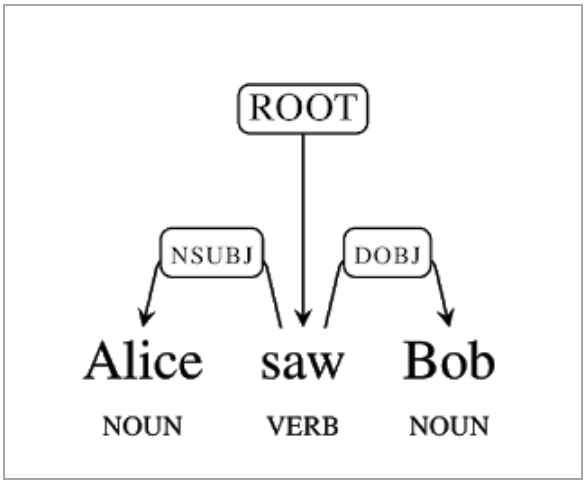

Рисунок 1.9 – Пример дерева зависимости.

В отличие от парсеров на грамматиках составляющих, парсеры деревьев зависимости разрабатываются с использованием методов машинного и глубокого обучения. Среди наиболее популярных инструментов можно выделить SyntaxNet [40], MaltParser [38] и UDPipe [41]. Чтобы обучить такую модель и проверить ее правильность для каждого языка, необходимо иметь большой аннотированный синтаксический корпус.

1.5.1 Синтаксически аннотированные корпуса

Процесс создания синтаксических парсеров начинается с аннотации корпуса с использованием специальных программ (например, Brat annotation tool [42]). В целом любой лингвистический корпус представляет собой совокупность текстов, которые обычно структурированы по темам, датам, авторам и т. д. В области компьютерной лингвистики слово «корпус» обычно подразумевает совокупность текстов, которые имеют морфологическую, синтаксическую, семантическую или любую другую разметку. Например, синтаксическая разметка обычно включает в себя такие признаки, как часть речи, лемма, регистр, пол и теги зависимостей для каждого слова. Альтернативное название для разбираемого текстового корпуса, которое широко используется – это деревья синтаксического разбора (Treebank) [43]. Разработка таких корпусов – очень сложная задача, которая требует

большого количества времени и работы лингвистов. В результате такие корпуса могут быть использованы для множества задач и приложений.

Одним из самых популярных аннотированных корпусов для русского языка является OpenCorpora [44]. По сути, это проект направлен на развитие корпуса усилиями сообщества. Основная цель проекта – создать полностью аннотированный корпус, включающий морфологическую, синтаксическую и семантическую разметку, который был бы общедоступным для скачивания любым лицом и для различных целей, согласно лицензии CC-BY-SA. Другим популярным корпусом для русского языка, имеющего синтаксическую аннотацию, считается SynTagRus. Это набор деревьев синтаксического разбора, созданных на основе Национального корпуса русского языка и содержащих более 52 000 предложений и около 770 000 слов. Корпус снабжен морфологической и синтаксической аннотацией в форме дерева зависимостей для каждого предложения. Кроме того, SynTagRus содержит в себе аннотации и других типов, в первую очередь, лексическую функциональную аннотацию в терминах лексических функций, как это определено в модели «Смысл-текст», которая рассматривает формальную грамматику русского языка. SynTagRus находиться в свободном доступе для исследовательских и образовательных целей [46].

Во время разработки аннотированных корпусов лингвисты обычно используют определенное количество тегов для маркировки части речи, функций и отношений. На текущий момент каждый корпус имеет свои собственные соглашения и указания по разметке, однако было бы удобнее пользоваться едиными и универсальными правилами для этой цели. Одним из таких проектов, который направлен на создание согласованной структуры для аннотации дерева зависимостей на разных языках, считается Universal Dependencies [43]. Этот проект позволил разработать деревья синтаксического разбора значительного размера и разной степенью качества более чем для 30 языков, которые предоставлены в едином согласованном формате. Формат называется CoNLL-U, который исторически получил

название на Конференции по изучению естественного языка (the Conference on Natural Language Learning – CoNLL), которая специализируется на анализе языковых зависимостей [43].

ID	FORM	LEMMA		UPOSTAG	XPOSTAG	FEATS	HEAD	DEPREL	DEPS
1	Trešdienas	trešdiena	'Wednesday'	NOUN	ncfsg4		2	nmod	2:nmod:gen
2	vakarā	vakars	'evening'	NOUN	ncmsl1		7	obl	7:obl:loc
3	79	79	'79'	NUM	xn		4	nummod	4:nummod
4	gadu	gads	'year'	NOUN	ncmpg1		5	nmod	5:nmod:gen
5	vecumā	vecums	'age'	NOUN	ncmsl1		7	obl	7:obl:loc
6	mūžībā	mūžība	'eternity'	NOUN	ncfsl4		7	obl	7:obl:loc
7	aizgājis	aiziet	'leave'	VERB	vmnpdmsnasnpn		0	root	0:root
8	tautā	tauta	'nation'	NOUN	ncfsl4		9	obl	9:obl:loc
9	mīlētais	mīlēt	'love'	VERB	vmnpdmsnpsypn		10	amod	10:amod
10	dzejnieks	dzejnieks	'poet'	NOUN	ncmsn1		11	nmod	11:nmod
11	Imants	Imants	'Imants'	PROPN	npmsn1		7	nsubj	7:nsubj
12	Ziedonis	Ziedonis	'Ziedonis'	PROPN	npmsn2		11	flat:name	11:flat:name
13	.	.	'.'	PUNCT	zs		7	punct	7:punct

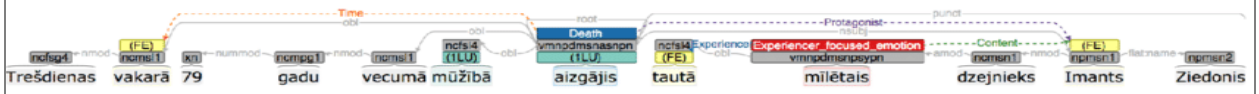

Figure 1: FrameNet annotation in WebAnno on top of a UD tree (Table 1). Only head nodes are selected while annotating frame elements (FE). The FE spans can be acquired automatically by traversing the respective subtrees: [*trešdienas vakarā*]$_{\text{Time}}$, [*tautā*]$_{\text{Experiencer}}$, [*tautā mīlētais dzejnieks Imants Ziedonis*]$_{\text{Protagonist}}$. Multi-word lexical units (LU) are indicated by generic LU tags: *mūžībā aizgājis*$_{\text{DEATH}}$ versus *mīlētais*$_{\text{EXPERIENCER\_FOCUSED\_EMOTION}}$.

Рисунок 1.10 – Пример результата синтаксического анализа, представленного в формате CoNLL-U.

Как видно из рисунка 1.10, существует три поля токена, связанные с тегами части речи: UPOSTAG (универсальный тег для части речи), XPOSTAG (тег части речи, являющийся локальным по отношению к каждому языку и корпусу, конвертируемый в последствии в XPOSTAG), FEATS (список морфологических особенностей, уточняющий универсальный тег части речи). [43] Поля HEAD и DEPREL используются для кодирования дерева зависимостей над словами. Поле HEAD содержит ссылку на ID зависимого по отношению к текущему токену слову. Значение DEPREL отражает тип зависимости или специфичный подтип такого отношения для каждого конкретного языка. Как и в случае с морфологией, синтаксическая аннотация предоставляется только для слов, а токены, не являющиеся словами, имеют подчеркивание в полях HEAD и DEPREL [43].

1.5.2 Модели синтаксического анализа

Размеченные синтаксические корпуса позволяю обучать модели глубокого обучения для их использования в синтаксическом разборе предложений. Например, модель UDPipe (анализатор универсальных зависимостей) является многофункциональной моделью для токенизации, тегирования, лемматизации и анализа зависимостей файлов CoNLL-U. В результате использования этого инструмента мы можем ввести текст и получить вывод готовый CoNLL-U файл [41].

Для обучения нейронных сетей предоставляются аннотированные данные в формате CoNLL-U. Архитектура нейронных сетей (рисунок 1.11), используемая для этой задачи, была подробно описана Ченом и Мэннингом [46]. Входной слой имеет несколько узлов, которые представляют каждое слово в дереве предложения. Как было упомянуто Чжаном и Нивром в [47] и Ченом и Мэннингом в [46], каждый токен может быть представлен в виде вектора, включающего теги части речи, морфологические теги и метки связей. Часть речевых меток и меток дуг инициализируются случайным образом и устанавливаются в процессе обучения. Входной слой имеет функцию активации Softmax и передается в скрытые слои.

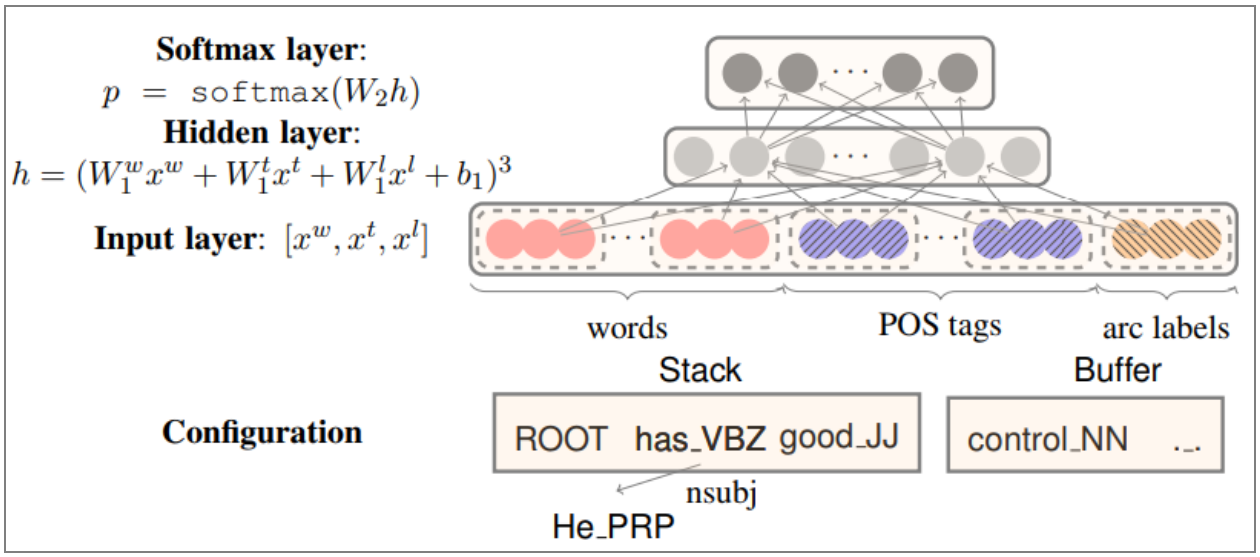

Рисунок 1.11 – Архитектура нейронной сети для синтаксического разбора [46].

Кроме того, в процессе обучения используется векторное представления слов, которые ранее обучались в наборе данных Википедии с помощью модели скип-граммы с отрицательной выборкой. Сеть обучается с использованием стохастического градиентного спуска с размером партии 10. Кросс-энтропия минимизируется с помощью L2-регуляризации.

2. Комбинированный подход для извлечения структурированных данных из текста

В настоящей работе предлагается подход к решению задачи по извлечению фактов для русского языка, а также для языков со свободным порядком слов. На рисунке 2.1 представлена диаграмма с пошаговым описанием разработанной методики по извлечению фактов из текста.

Рисунок 2.1 – Методика предлагаемого подхода.

Идея методики заключается в комбинировании подхода по извлечению информации, основанного на слиянии результатов, полученных с помощью контекстно-свободных грамматик с результатами выделения шаблонов при синтаксическом разборе текста. Далее будут подробно рассмотрены шаги подхода, а также описана программная библиотек, разработанная на его основе.

Основным преимуществом грамматик зависимостей является унифицированность результатов по отношению к языкам со свободным порядком слов, например русский язык. Поэтому было бы очень удобно использовать грамматики зависимостей для извлечения информации из таких языков. Кроме того, невозможно покрыть все синтаксические шаблоны самописными правилами на контекстно-свободных грамматиках. Несмотря на этот недостаток, два самых популярных парсера для русского языка (Tomita и Yargy) используют контекстно-свободные грамматики.

2.1 Выбор парсера контекстно-свободных грамматик

Среди библиотек, рассматриваемых в предыдущей главе, необходимо осуществить выбор, исходя из предъявляемых требований и критериев:

- функциональные возможности;
- наличие подробной документации и примеров готовых правил;
- доступность интерфейса для удобной работы на основном используемом языке программировании – Python;
- производительность.

Наиболее подходящим решением, с учетом основных предъявляемых требований, является программная библиотека Yargy, реализованная на языке Python. По сравнению с Tomita парсером, Yargy обладает схожими функциональными возможностями, однако в качестве морфологического анализатора он использует Pymorphy2 [48]. Данный анализатор на каждую словоформу на входе выдает несколько возможных результатов морфологической информации и лемм, тогда как анализатор Tomita-парсера выбирает единственный вариант на основе контекста. Еще одним отличием является то, что Tomita парсер работает через консольный интерфейс, Yargy – это библиотека на Python. Кроме того, для написания правил в Tomita используется свой язык и Protobuf-файлы, а в Yargy грамматики и словари описываются как переменные в Python. Главным преимуществом в Yargy является наличие банка готовых грамматик в библиотеке Natasha, которая разработана для извлечения имен, организаций, локаций, адресов и т. д. и поддерживает добавление собственных грамматик.

Необходимо отметить, что библиотека Yargy уступает по производительности Tomita парсеру. Это связано с тем, что он реализован на Python, тогда как Tomita – на языке C++ и, вероятнее всего, не так хорошо оптимизирован. С другой стороны, относительно небольшая производительность частично компенсируется использованием интерпретатор PyPy, а также распараллеливанием на отдельные процессы или даже компьютеры с помощью асинхронных очередей для распределения

задач [49], например с использованием Redis. На рисунке 2.2 приведен пример извлечения топонимов, начинающихся прилагательными и заканчивающихся словами «федерация» или «республика».

```
from yargy import Parser, rule, and_
from yargy.predicates import gram, is_capitalized, dictionary


GEO = rule(
    and_(
        gram('ADJF'),  # так помечается прилагательное, остальные пометки о
                       # http://pymorphy2.readthedocs.io/en/latest/user/gra
        is_capitalized()
    ),
    gram('ADJF').optional().repeatable(),
    dictionary({
        'федерация',
        'республика'
    })
)


parser = Parser(GEO)
text = '''
В Чеченской республике на день рождения ...
Донецкая народная республика провозгласила ...
Башня Федерация — одна из самых высоких ...
'''
for match in parser.findall(text):
    print([_.value for _ in match.tokens])
```

```
['Донецкая', 'народная', 'республика']
['Чеченской', 'республике']
```

Рисунок 2.2 – Пример извлечения топонимов, начинающихся прилагательными и заканчивающихся словами «федерация» или «республика».

Правила в Yargy могут состоять из других правил и предикатов. Предикат – это функция, которая принимает на вход токен и возвращает True или False. Правила и предикаты могут логически комбинироваться при помощи функций – логических операторов and_, or_ и not_ (рисунок 2.3).

eq(value)	a == b
caseless(value)	a.lower() == b.lower()
in_(value)	a in b
in_caseless(value)	a.lower() in b
gte(value)	a >= b
lte(value)	a <= b
length_eq(value)	len(a) == b
normalized(value)	Нормальная форма слова == value
dictionary(value)	Нормальная форма слова in value
gram(value)	value есть среди граммем слова
type(value)	Тип токена равен value
tag(value)	Тег токена равен value
custom(function[, types])	function в качестве предиката
true	Всегда возвращает True
is_lower	str.islower
is_upper	str.isupper
is_title	str.istitle
is_capitalized	Слово написано с большой буквы
is_single	Слово в единственном числе

Рисунок 2.3 – Предикаты, доступные в библиотеки Yargy [?].

Таким образом, библиотека Yargy будет использоваться на начальном этапе извлечения информации, в частности для извлечений именованных сущностей, значений, дат, маркировок, навыков, и др, состоящих из нескольких ищущих друг за другом слов. В дальнейшим, результат извлечения сущностей будет комбинироваться с синтаксическими деревьями.

2.2 Выбор модели синтаксического анализа

Среди множества существующих решений по синтаксическому разбору предложений, также есть и с открытым исходным кодом, среди которых можно выделить модели SyntaxNet и UDPipe. Согласно результатам соревнования «Multilingual Parsing from Raw Text to Universal Dependencies»

в рамках конференции Conference on Natural Language Learning 2018 (CoNLL 2018), наилучшие результаты для русского языка по метрике MLAS (Morphology-Aware Labeled Attachment Score) показала модель UDPipe (рисунок 2.4) [50]. Данная модель представляет собой целый Pipeline, включающий в себя последовательность этапов токенизации, лемматизации, морфологическому анализу, и синтаксическому разбору предложения, основанного на грамматиках зависимостей (рисунок2.5).

	Treebank	MLAS	Best system	Avg	StDev
1.	pl_lfg	86.93	UDPipe Future	73.73	± 7.29
2.	ru_syntagrus	86.76	UDPipe Future	71.63	± 9.36
3.	cs_pdt	85.10	UDPipe Future	**73.61**	± 6.32
4.	cs_fictree	84.23	ICS PAS	69.91	± 7.77
5.	ca_ancora	84.07	UDPipe Future	**74.62**	± 7.69
6.	es_ancora	83.93	Stanford	74.61	± 7.43
7.	it_isdt	83.89	Stanford	**77.14**	± 8.89
8.	fi_pud	83.78	Stanford	62.38	±14.83
9.	no_bokmaal	83.68	UDPipe Future	**70.75**	± 8.92
10.	cs_cac	83.42	UDPipe Future	**71.39**	± 6.89
11.	bg_btb	83.12	UDPipe Future	**73.18**	± 7.15
12.	fr_sequoia	82.55	Stanford	70.42	± 9.04
13.	sl_ssj	82.38	Stanford	62.41	± 9.18
14.	no_nynorsk	81.86	UDPipe Future	**68.62**	± 9.45
15.	ko_kaist	81.29	HIT-SCIR	**70.18**	± 9.36

Рисунок 2.4 – Ранжирование лучших результатов синтаксических парсеров (метрика MLAS) [50].

```
# newdoc
# newpar
# sent_id = 1
# text = На предприятиях, занимающихся открытой разработкой полезных ископаемых, в последние годы активно идет процесс внедрени
я карьерных самосвалов большой грузоподъемности, использующих в качестве трансмиссии электрический привод переменно-переменного
тока.
1       На      на      ADP     _       _       2       case    _       _
2       предприятиях    предприятие     NOUN    _       Animacy=Inan|Case=Loc|Gender=Neut|Number=Plur   14      obl     _
SpaceAfter=No
3       ,       ,       PUNCT   _       _       2       punct   _       _
4       занимающихся    заниматься      VERB    _       Aspect=Imp|Case=Gen|Number=Plur|Tense=Pres|VerbForm=Part|Voice=Mid
2       amod    _       _
5       открытой        открытый        ADJ     _       Case=Ins|Degree=Pos|Gender=Fem|Number=Sing      6       amod    _
_
6       разработкой     разработка      NOUN    _       Animacy=Inan|Case=Ins|Gender=Fem|Number=Sing    4       obl     _
_
7       полезных        полезный        ADJ     _       Case=Gen|Degree=Pos|Number=Plur 8       amod    _       _
8       ископаемых      ископаемое      NOUN    _       Animacy=Inan|Case=Gen|Gender=Neut|Number=Plur   6       nmod    _
SpaceAfter=No
9       ,       ,       PUNCT   _       _       8       punct   _       _
10      в       в       ADP     _       _       12      case    _       _
11      последние       последний       ADJ     _       Animacy=Inan|Case=Acc|Degree=Pos|Number=Plur    12      amod    _
_
12      годы    год     NOUN    _       Animacy=Inan|Case=Acc|Gender=Masc|Number=Plur   14      obl     _       _
13      активно активно ADV     _       Degree=Pos      14      advmod  _       _
14      идет    идти    VERB    _       Aspect=Imp|Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin|Voice=Act      0
root    _       _
15      процесс процесс NOUN    _       Animacy=Inan|Case=Nom|Gender=Masc|Number=Sing   14      nsubj   _       _
16      внедрения       внедрение       NOUN    _       Animacy=Inan|Case=Gen|Gender=Neut|Number=Sing   15      nmod    _
_
17      карьерных       карьерный       ADJ     _       Case=Gen|Degree=Pos|Number=Plur 18      amod    _       _
18      самосвалов      самосвал        NOUN    _       Animacy=Inan|Case=Gen|Gender=Masc|Number=Plur   16      nmod    _
```

Рисунок 2.5 – Пример синтаксического разбора предложения.

Для избежания потери качества результатов на этапе синтаксического анализа не следует заменять текущий анализатор на анализатор Pymorphy или любой другой стороний, т. к. текущая модель обучалась именно на нем. Этапы лемматизации и морфологического анализа будут производиться параллельно с теми же этапами синтаксического разбора на контекстно-свободных грамматиках.

2.3 Комбинирование подхода по разбору на основе контекстно-свободных грамматик и грамматик зависимости

На основе преимуществ и недостатков контекстно-свободных грамматик и грамматик зависимости, их результаты возможно комбинировать для решения задач по извлечению информации. На первом шаге правила контекстно-свободных грамматик применяются для извлечения сущностей. Далее с помощью UDPipe строиться синтаксическое дерево предложения. Затем, токены из первого шага сопоставляются с элементами полученного дерева. Информация о полученных на первом шаге сущностях добавляется в элементы синтаксического дерева. В результате в полученном дереве, некоторые элементы которого помечены как сущности, появляется возможность описывать шаблоны для выделения более сложных фактов.

Прежде всего для анализа данных необходимо получить синтаксическое представление каждого рассматриваемого предложения. Так, полученные в формате CoNNL-U данные преобразуются структуру данных «дерево», содержащих информацию в виде словаря у каждого токена-элемента дерева о его морфологических и синтаксических признаках. Например, анализируя следующее предложение: «Дональд Трамп выбрал нового представителя США в ООН» и рассмотрев токен «Трамп», получим словарь, представленный на рисунке2.6. Далее к каждому токену необходимо добавить поле spans, показывающее пару индексов символов начала и окончания вхождения этого токена в тексте. Это необходимо для корректного сопоставления токенов, полученных с помощью UDPipe с

токенами, полученными Yargy, так как на их основе работают разные токенизаторы.

```
OrderedDict([('id', 2),
             ('form', 'Трамп'),
             ('lemma', 'Трамп'),
             ('upostag', 'PROPN'),
             ('xpostag', None),
             ('feats',
              OrderedDict([('Animacy', 'Anim'),
                           ('Case', 'Nom'),
                           ('Gender', 'Masc'),
                           ('Number', 'Sing')])),
             ('head', 1),
             ('deprel', 'appos'),
             ('deps', None),
             ('misc', None)])
```

Рисунок 2.6 – Пример содержащейся информации токена «Трамп».

Следующим шагом является извлечение именованных сущностей с помощью библиотеки Yargy или Natasha. В результате анализа данного предложения анализатор находит именованную сущность (рисунок 2.7).

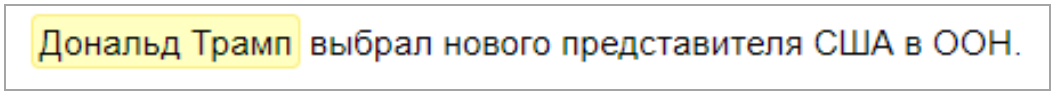

Рисунок 2.7 – Пример извлечения имени с помощью Natasha.

После сопоставления токенов извлеченных сущностей с токенами синтаксического дерева можно писать-правила шаблоны, например для извлечения связей «субъект-предикат», в которых в качестве субъекта выступает именованная сущность. Используя связь типа «nsubj» для связи с токеном предикатом и тип токена «VERB» можно извлекать простейшие фразы. Пример показан на рисунке 2.8.

```
{'subject': 'Дональд Трамп', 'predicate': 'выбрал'}
```

Рисунок 2.8 – Пример извлекаемого отношения «*субъект-предикат*».

Расширим шаблон до «субъект-предикат-объект». Зададим в шаблон описание связи с типом «obj» между токенами предиката и объекта. После этого получим извлеченный факт, который показан на рисунке 2.9.

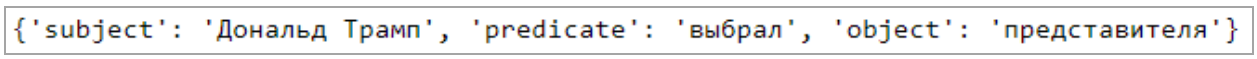

Рисунок 2.9 – Пример извлекаемого отношения «субъект-предикат-объект».

Данный подход не ограничивается извлечение пар и троек вида «субъект-предикат-объект», он может описываться более сложными, разнообразными шаблонами и применяться для случаев извлечения более интересных фактов. Следует отметить, что для этого нужна, *во-первых*, для построения таких шаблонов нужна доступная визуализация таких деревьев для эффективной работы и, *во-вторых*, конструктор для удобного написания, чтения и редактирования таких шаблонов. Для решения этих задач была разработана библиотека.

2.4 Описание разработанной программной библиотеки

Разработанная библиотека предлагает следующий функционал:

- визуализация синтаксических деревьев;
- возможность описания синтаксических шаблонов;
- извлечение фактов по шаблонам.

Метод визуализации деревьев реализован с помощью библиотеки matplotlib. Необходимо отметить, что каждое синтаксическое дерево обладает некоторыми особенностями, такими как:

- граф является однонаправленным;
- у основания находится только один корневой элемент;
- у каждого элемента может быть только один родительский и несколько дочерних элементов.

В связи с этим, для визуализации такого направленного графа требуется вычислять координаты каждого элемента, предварительно вычислив количество элементов, приходящихся на каждый уровень и общую глубину дерева. Пример визуализации представлен на рисунке 2.10.

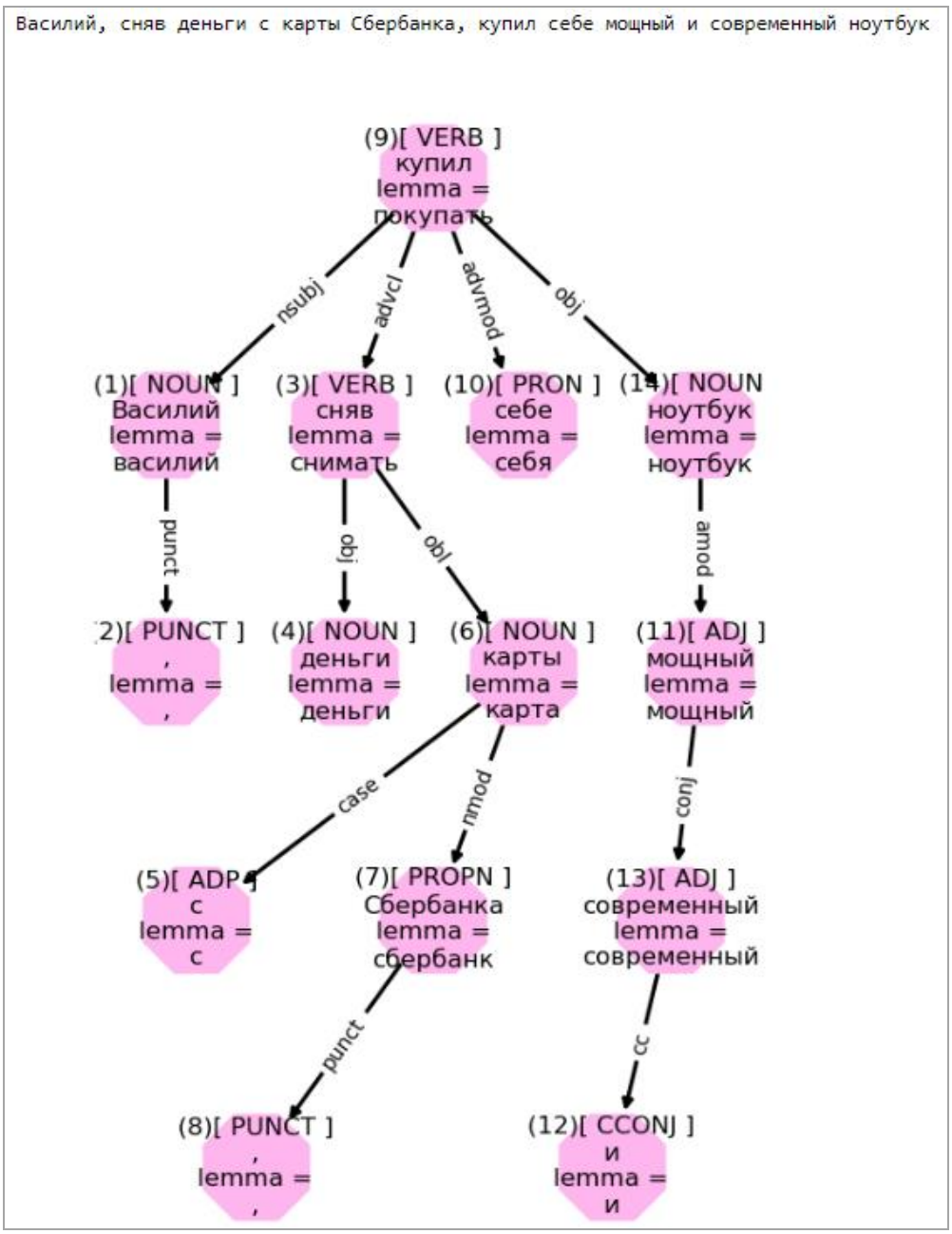

Рисунок 2.10 – Визуализация синтаксического дерева.

Для разработки шаблонов по извлечению фактов были созданы конструкторы на языке Python. Их можно условно разделены на предикаты, т. е. функции, проверяющие каждый токен на соответствие определенному правилу. Например, функция «pos(…)» проверяет часть речи токена на соответствие значению аргумента. «pos_in(…)» в качестве аргумента принимает массив доступных частей речи. Использование функций «lem» и «lem_in» проверяют на соответствие лемму слова; regex – упоминание регулярного выражения в словоформе. Функции «rel» и «rel_in» проверяют

на соответствие типа связи у дочернего элемента. Спецификация доступных на текущий момент предикатов представлена в таблице 2.1:

Таблица 2.1 – Спецификация предикатов для построения шаблонов

Предикаты	
pos(value)	Part of Speech == value
pos_in(value)	Part of Speech in value
lem(value)	Лемма == value
lem_in(…)	Лемма in value
regex(value)	Регулярное выражение value срабатывает в словоформе
rel(value)	Тип связи == value
rel_in(value)	Тип связи in value
ent(value)	Парсер КСГ нашел сущность с типом value

Для логического комбинирования правил разработаны конструкторы (таблица 2.2): child(…) – конструктор для описания дочернего элемента, принимающий в качестве аргумента другие конструкторы или правила. *_and*, *_or*, *_not* – логические операторы для группировки правил и конструкторов.

Таблица 2.2 – Спецификация конструкторов для построения шаблонов

Конструкторы	
child(…)	Указание на дочерний элемент
_and(…)	Логическое «и»
_or(…)	Логическое «и»
_not(…)	Логическое отрицание

На рисунке 2.11 представлен пример использования правила для извлечения фактов по шаблону, содержащих существительное, глагол и число с процентом.

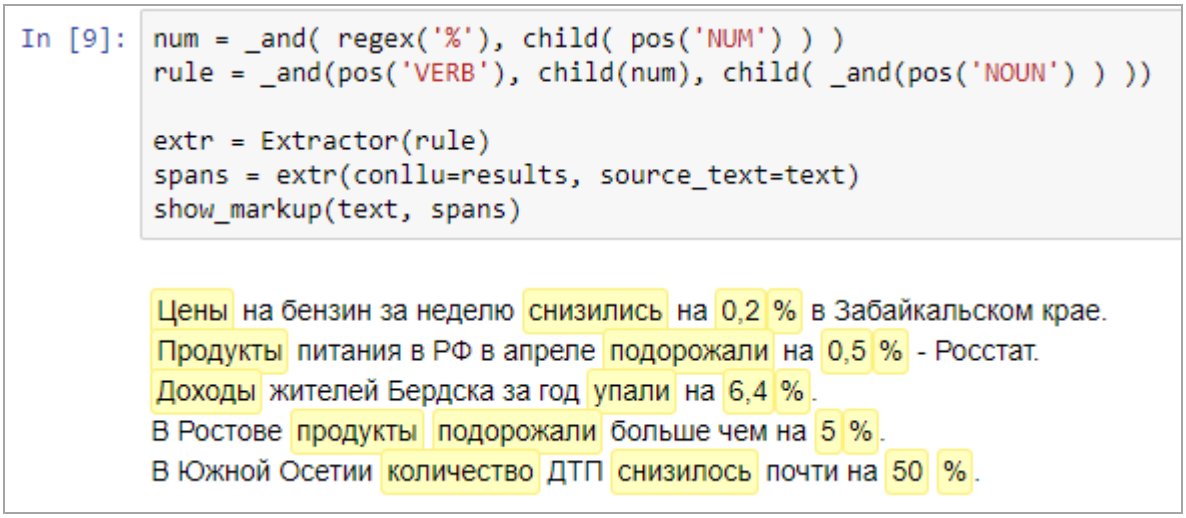

Рисунок 2.11 – Пример написания шаблонов в синтаксической структуре.

Как видно, такой шаблон отлично работает на примерах по изменению цен, количеств ДТП, доходов и т. д.

Выводы

В настоящем разделе была описана методика предлагаемого подхода к решению задачи по извлечению фактов для русского языка, который может быть распространен и для других языков со свободным порядком слов. В качестве парсера на контекстно-свободных грамматиках использовалась бибилотека Yargy совместно с Natasha, а в качестве модели синтаксического разбора – UDPipe. Была описана разработанная программная библиотека для описания шаблонов и приведены примеры ее использования.

3. Проектирование и разработка приложения по извлечению информации из резюме на основе предлагаемого подхода

С целью проверки и демонстрации работоспособности предлагаемого подхода и разработанной на его основе библиотеки, было создано приложение, целью которого является анализ данных из документов резюме и извлечение следующей информации: контактные данные (ФИО, дата рождения, телефон, адрес электронной почты, претендуемая должность, ЗП), описание опыта работы (организация, занимаемая должность, период работы), различные навыки, компетенции и дополнительная информация (разрешение на работу, готовность к командировкам, военный билет и т.д.).

3.1 Проектирование приложения и выбор средств разработки

На рисунке 3.1 представлена диаграмма жизненного цикла по загрузке, анализу и сохранении данных кандидата в базу.

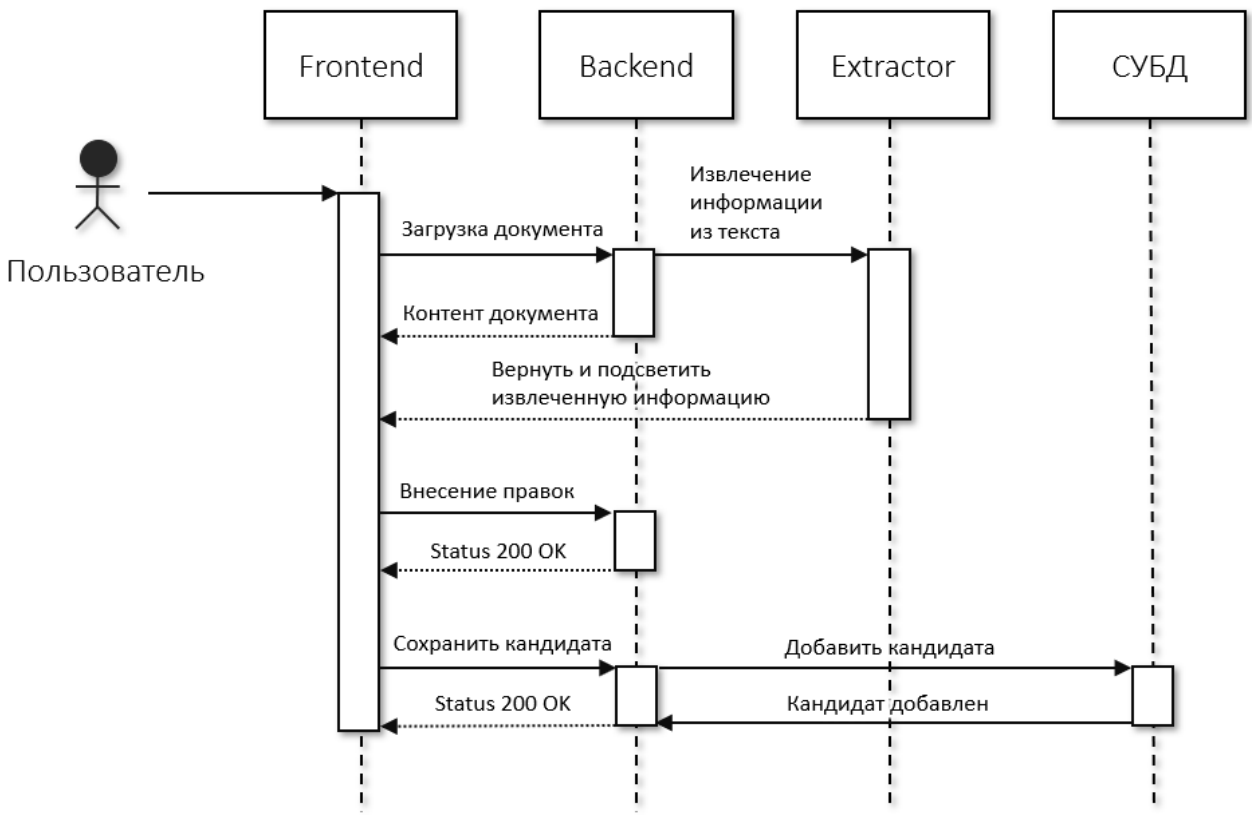

Рисунок 3.1 – Диаграмма последовательности приложения по извлечению информации из резюме.

Как видно из представленной диаграммы, пользователь взаимодействует с приложением через Frontend часть. Frontend часть, в свою очередь, реализует основную логику и операции с СУБД, вызовы компонента Extractor и отдает конечный ответ пользователю средствами REST API. Компонент Extractor является отдельным процессом, выполняющим извлечение информации из текста с помощью разработанных правил-шаблонов, полученных с использованием реализованной библиотеки. Взаимодействие Backend с Extractor осуществляется через асинхронные очереди на основе Redis.

Основываясь на поставленных задачах, а также особенностях работы созданной библиотеки, для разработки текущего приложения был выбрал следующий стек технологий:

- для реализации Frontend части – Vue.js совместно с фреймворком для компонент Vuetify, разработанного в соответствии со спецификацией Material Design;
- разработка Backend осуществлялась с использованием связки языка Python совместно с фреймворком Flask. В качестве главного преимущества Flask можно выделить гибкость и удобство использования в относительно небольших проектах. Для объектно-реляционного отображения применялась библиотека SQLAlchemy;
- в качестве СУБД была выбрана PostgreSQL. PostgreSQL, которая является наиболее развитой объектно-реляционной системой на сегодняшний день и достойной альтернативой коммерческим базам данных.

Таким образом, настоящее приложение является относительно несложным с архитектурной точки зрения и выполняет лишь одну основную функцию: извлечение структурированной информации из резюме для заполнения базы данных.

3.2 Сбор данных для написания грамматик

Для разработки грамматик, обеспечивающих достаточную полноту при извлечении информации, был осуществлен сбор данных резюме соискателей из открытых источников. В качестве источника использовался один из известных в России и СНГ порталов для поиска работы, а также исполнителей.

Извлечение набора резюме проводилось путем разработки парсера данных на основе библиотеки Selenium и ChomeDriver в качестве браузера. На рисунке 3.2 представлена графическая визуализация схемы по сбору данных.

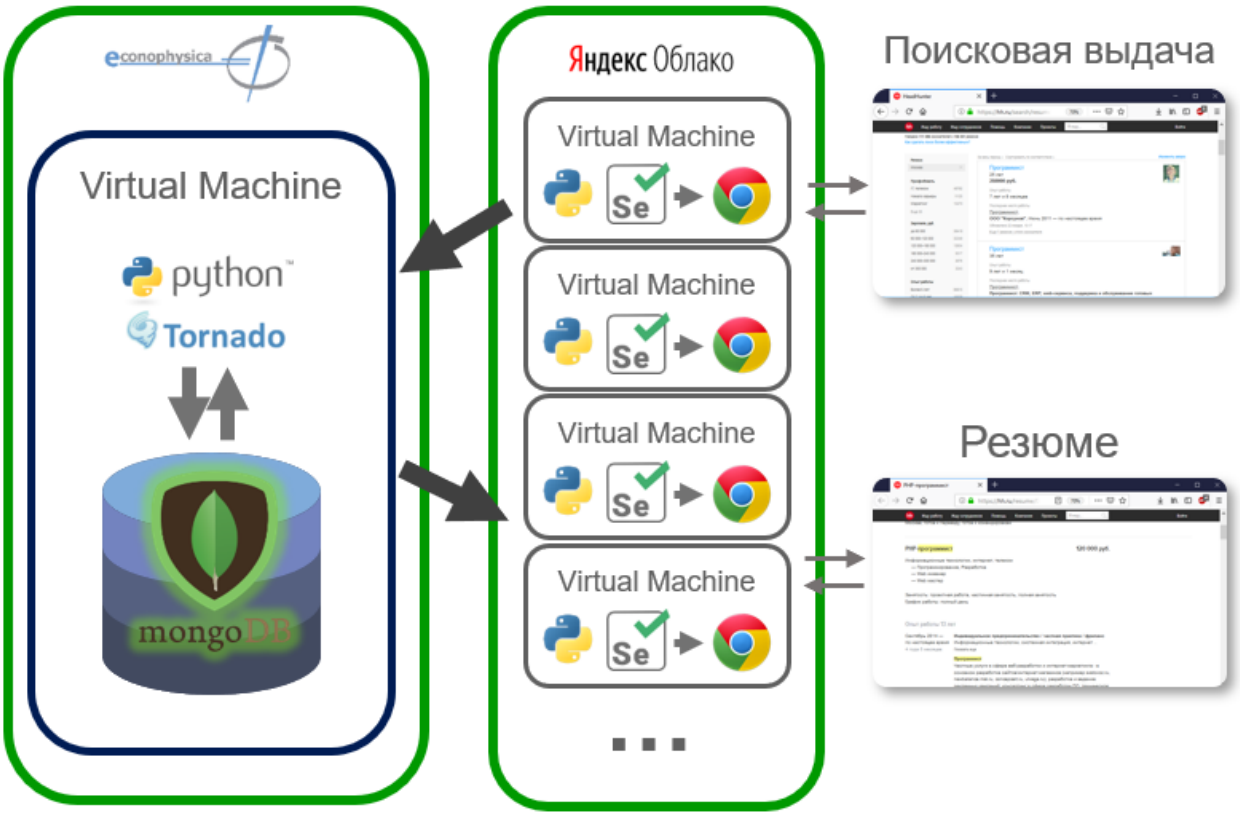

Рисунок 3.2 – Схема сбора данных резюме соискателей.

Для составления словарей поисковый робот находил интересующую должность, после чего разбирал поисковую выдачу. Результаты поисковой выдачи – ссылки на конкретные профили-резюме соискателей, которые передавались с микросервис очереди, реализованные с использованием Python и Tornado. Другие роботы обращались в очередь за набором ссылок на

конкретные профили, после чего посещали их, разбирали HTML код страницы и передавали результаты обратно в микросервис. После чего разобранные данные помещались в хранилище на основе MongoDB. Поиск осуществлялся по запросам, соответствующим популярным должностям и компетенциям:

1 Менеджер по продажам;
2 Data Scientist;
3 Аналитик Big Data;
4 Специалист SAP;
5 Руководитель проектов;
6 Autodesk;
7 Системный администратор;
8 Инженер-геолог;
9 Инженер по промышленной безопасности;
10 Научный сотрудник;
11 Инженер проекта;
12 Водоснабжение;
13 Архитектор;
14 Журналист;
15 НСИ;
16 Промышленная безопасность;
17 Токарь;
18 Строительное проектирование;
19 Охрана труда;
20 Инженер-электрик;
21 Сбор данных;
22 Грамматики.

В результате было собрано 2 482 707 анкет соискателей. После этого этапа для написания грамматик были составлены словари ключевых навыков и названий должностей. На рисунках 3.3 и 3.4 представлены наиболее

популярные ключевые навыки и должностные позиции, исходя из описания опыта работы соискателей. Стоит отметить, что поисковые запросы для сбора данных содержали не репрезентативную выборку пользователей, а лишь тех, компетенции которых интересовали компанию-заказчика.

```
In [31]: counter.most_common(40)

Out[31]: [('пользователь пк', 13594),
          ('работа в команде', 11100),
          ('autocad', 10709),
          ('управление проектами', 10704),
          ('ведение переговоров', 10551),
          ('организаторские навыки', 10510),
          ('adobe photoshop', 8494),
          ('деловая переписка', 7499),
          ('управление персоналом', 6959),
          ('грамотная речь', 6856),
          ('руководство коллективом', 6038),
          ('водительское удостоверение категории b', 5959),
          ('ms powerpoint', 5848),
          ('английский язык', 5740),
          ('деловое общение', 5216),
          ('ms outlook', 5021),
          ('ms office', 4840),
          ('ms excel', 4488),
          ('заключение договоров', 4397),
          ('ms word', 3943),
          ('обучение персонала', 3841),
          ('работа с большим объемом информации', 3397),
          ('проектная документация', 3357),
          ('archicad', 3135),
          ('coreldraw', 3083),
          ('охрана труда и техника безопасности', 2799),
          ('internet', 2793),
          ('проведение презентаций', 2760),
```

Рисунок 3.3 – Наиболее популярные ключевые навыки / компетенции.

```
In [34]: counter.most_common(40)

Out[34]: [('архитектор', 6498),
          ('инженер', 5829),
          ('системный администратор', 5596),
          ('менеджер по продажам', 5488),
          ('руководитель проекта', 4764),
          ('главный инженер', 3943),
          ('токарь', 3743),
          ('руководитель проектов', 3438),
          ('журналист', 2810),
          ('архитектор-дизайнер', 2101),
          ('токарь-универсал', 2087),
          ('продавец-консультант', 2073),
          ('инженер пто', 1997),
          ('менеджер по работе с клиентами', 1932),
          ('ведущий архитектор', 1873),
          ('инженер-конструктор', 1748),
          ('корреспондент', 1733),
          ('директор', 1618),
          ('ведущий инженер', 1508),
          ('инженер-проектировщик', 1504),
          ('дизайнер', 1451),
          ('главный инженер проекта', 1434),
          ('технический директор', 1337),
          ('менеджер', 1330),
          ('начальник участка', 1249),
          ('менеджер проектов', 1222),
          ('генеральный  директор', 1157),
          ('бухгалтер', 1088),
```

Рисунок 3.4 – Наиболее популярные должности исходя из описаний опыта работы.

На основе полученных словарей составлялись правила, позволяющие извлекать должности, а также навыки и компетенции из резюме. Большая часть грамматик реализована непосредственно с помощью библиотеки Yargy, однако для извлечения фактов с предыдущих мест работы соискателей, соответствующих связкам «Организация-должность-период» извлекаются данные с использованием предложенного комбинированного подхода.

3.3 Описание приложения

Разработанное приложение представляет собой страницу, через которую можно как загрузить файл в формате doc / docx / pdf, так и напрямую скопировать текст резюме во вкладку «Исходный текст». При загрузке файла поле заполняется автоматически. Разработанные шаблоны и правила позволяют извлекать следующие атрибуты из анкет:

- ФИО, Контактные данные, претендуемая должность, ЗП;
- опыт работы (организация, должность, период работы);
- Навыки (технологии и личностные качества);
- Знание языков;
- Дополнительная информация (разрешение на работу, готовность к командировкам, военный билет и т.д.).

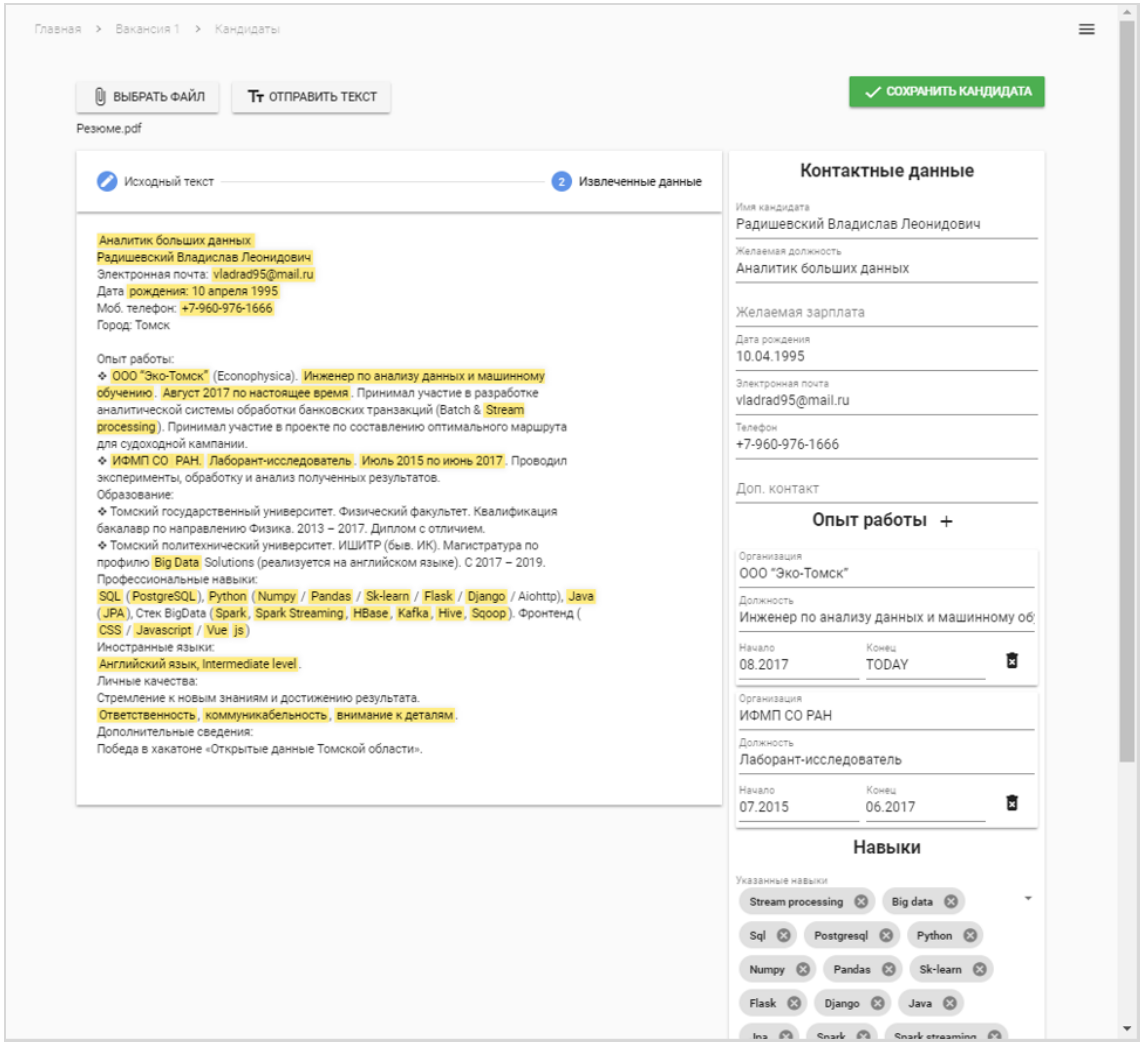

Рисунок 3.5 – Страница с извлеченным резюме.

После небольшого ожидания работы алгоритма во вкладке «Исходный текст» появится тот-же текст, но уже с подсвеченными областями, в которых нашлись упоминания (рисунок 3.5). Поля справа также заполняются автоматически. Однако для того, чтобы пользователь мог дополнить информацию, либо исправить недочеты алгоритма, содержимое полей можно редактировать. После редактирования пользователь приложения нажимает на кнопку «Сохранить кандидата». Затем его данные загружаются на сервер в базу данных для целей дальнейшего анализа и других целей.

Выводы

В разработке конечного приложения применялись следующие инструменты и технологии: Python, Flask, PostgreSQL, Vue.js, Selenium. Для составления правил, словарей, и их проверки корректно использовались данные из 2 482 707 анкет соискателей. В результате с помощью разработанного приложения возможно извлекать и заполнять базу данных всей основной информацией: контактные данные, опыт работы, компетенции и т. д.

4 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

В рамках настоящей работы был предложен новый метод к извлечению структурированных данных для языков со свободным порядком слов, а также разработана программная библиотека, предназначенная для составления специальных правил по извлечению информации из текста. Алгоритм предполагает использование контекстно-свободных грамматик и современных моделей синтаксического анализа текста с последующей постобработкой выделения ключевых фраз и смыслов из текста.

Планирование работ и работа над рисками дает возможность обеспечить высокую вероятность успешного достижения целей и выступает предпосылкой эффективной реализации проекта. Перед тем, как представить продукт на рынке информационных систем, необходимо оценить данную разработку с точки зрения ее востребованности, а также ресурсоэффективности и ресурсосбережения. Для достижения данной цели решались следующие задачи: определение потенциальных потребителей, анализ конкурентоспособности технического решения, планирование проекта, формирование его бюджета, а также определение финансовой эффективности и разработка реестра рисков.

4.1 Предпроектный анализ

4.1.1 Потенциальные потребители разрабатываемого решения

Для определения потенциальных потребителей требуется выявить целевой рынок. Для того, чтобы определить организации, которым необходима данная разработка, было проведено сегментирование целевого рынка. Сегментация проводилась по следующим критериям: размер организации и задачи, которые данные организации решают. Карта сегментирования представлена в таблице 4.1.

Таблица 4.1 – Сегментация рынка

Задачи / Размер организации	Анализ внутренних документов	Анализ документов из открытых источников	Анализ товаров	Анализ тематических новостей	Бренд аналитика
Крупные	+				+
Средние		+	+	+	+
Мелкие		+	+	+	

Конечными потребителями разрабатываемого решения могут быть как любые крупные отечественные компании, имеющие большой внутренний документооборот и испытывающие потребность в проведении семантического анализа больших массивов текстовых данных, при которых работа аналитиков становится слишком ресурсоемкой и неэффективной.

Кроме того, потенциальными потребителями могут быть мелкие, а также средние компании в области консалтинга, рейтинговых агентств, хедж-фондов, интернет-магазинов и другие. Настоящее решение может применяться в таких задачах как анализ текстов из открытых источников, анализ наличия определенных товаров, анализ тематических новостей, бренд аналитика и т.д.

Таким образом целевыми потребителями являются средние и крупные предприятия, которые, как правило, располагают большими массивами текстовой информации, либо малые и средние, занимающиеся задачами по сбору и анализу данных из открытых источников.

4.1.2 Анализ конкурентоспособности технического решения

Анализ конкурентоспособности технического решения был проведен с помощью оценочной карты. В качестве конкурентных систем

рассматривались решения RCO Fact Extractor SDK ($К_1$) и Abby Compreno ($К_2$). Результаты анализа представлены в таблице 4.2.

Позиция разработки и конкурентов оценивается по каждому показателю экспертным путем по пятибалльной шкале, где 1 – наиболее слабая позиция, а 5 – наиболее сильная. Веса показателей, определяемые экспертным путем, в сумме должны составлять 1.

Таблица 4.2 – Результаты анализа конкурентных систем анализа текста

Критерии оценки	**Вес крите-рия**	**Баллы**			**Конкурентоспособ-ность**		
		$Б_ф$	**$Б_{к1}$**	**$Б_{к2}$**	**$К_ф$**	**$К_{к1}$**	**$К_{к2}$**
1. Требуемая квалификация для возможности конфигурирования решения	0,15	3	4	4	0,45	0,6	0,6
2. Возможность конфигурирования для решения задач в каждой предметной области	0,15	5	5	4	0,75	0,6	0,6
3. Возможность написания правил для фактов, описываемых в виде свободного порядка слов	0,15	5	3	3	0,75	0,45	0,45
4. Отказоустойчивость решения	0,15	5	5	5	0,75	0,75	0,75
5. Потребность в ресурсах памяти	0,14	5	5	5	0,7	0,7	0,7
6. Скорость обработки данных	0,1	2	3	3	0,2	0,3	0,3
7. Безопасность данных	0,09	4	4	4	0,36	0,36	0,36
8. Цена	0,07	4	3	3	0,28	0,21	0,21
Итого	**1**				**4,24**	**3,97**	**3,97**

Из таблицы сравнения собственного сервиса обработки документов, с решениями RCO Fact Extractor SDK и Abby Compreno, можно сделать вывод, что уязвимость конкурентных решений связана ограниченным функционалом представленных решений и значительной сложности обработки фактов, описываемых свободным порядком слов, с точки зрения возможности составления правил по извлечению информации.

Таким образом, разработанное решение является более предпочтительным для решения широкого класса задач благодаря новизне метода. Исходя из этого, во-первых, появляется возможность повысить качество результата извлечения фактов, а во-вторых, снижается количество кода, необходимого для построения правил благодаря работе не только с контекстно-свободными грамматиками, а также с грамматиками зависимостей и представление приложений в виде синтаксического дерева.

4.1.3 SWOT-анализ

SWOT – Strengths (сильные стороны), Weaknesses (слабые стороны), Opportunities (возможности) и Threats (угрозы) – это комплексный анализ научно-технического проекта. Такой анализ применяют для исследования внешней и внутренней среды проекта. Ниже представлена матрица SWOT.

Таблица 4.3 – Матрица SWOT разработки

Внутренняя среда / **Внешняя среда**	**Сильные стороны:** S_1. Использование современных моделей синтаксического анализа текста S_2. Возможность построения правил с использованием грамматики зависимостей S_3. Более низкая стоимость технологии S_4. Расширяемый функционал	**Слабые стороны:** W_1. Для работы с решением необходим специалист с опытом в области обработки естественных языков W_2. Данная библиотека не является конечным решением актуальных задач, а лишь конструктором этого решения
Возможности: O_1. Потребность рынка решении задач по извлечению информации O_2. Наличие других языков, в которых синтаксические модели показывают высокое качество	– Добавление поддержки новых языков позволит выйти на зарубежный рынок – Добавления новых конструкций для правил с учетом потребностей пользователей сделает конфигурирование более гибким – Привлечение новых клиентов благодаря конфигурированию продукта под их предметную область и задачу	– Разработка различных обучающих материалов: подробная документации с примерами, разработка курсов, проведение семинаров – Привлечение специалистов, внешних консультантов, а также профессиональное сообщество для масштабирования с точки зрения функционала и других языков

Продолжение таблицы 4.3

Угрозы: T_1. Появление на рынке новых игроков с аналогичной разработкой T_2. Появление новых, принципиально отличающихся методов решения	– Регулярное исследование целевых потребителей, выявление и работа над их новыми потребностями – Работа над оптимизацией производительности решения – Работа по снижению порога вхождения для разработчиков конечных решений	– Оказание услуг по конфигурации под конкретную предметную область – Поддержание стабильной ценовой политики

Необходимо отметить, что особенностью данного решения является его универсальность и возможность конфигурирования под каждую конкретную предметную область. Однако данная библиотека не является конечным решением, поэтому для решения каждой задачи по отдельности требуется дополнительное конфигурирование, либо предоставление обучающих материалов для специалистов, разрабатывающие конечное решение. Расширение функционала со временем позволит сделать процесс конфигурирования наиболее удобным и эффективным. Поддержание стабильной ценовой политики и оказание услуг по настройки под конкретную предметную область будет напрямую способствовать конкурентоспособности разрабатываемого решения.

4.2 Инициация проекта

В процессе инициации проекта определяются начальные цели и содержание, а также фиксируются финансовые ресурсы. Определяются внутренние и внешние заинтересованные стороны проекта, которые будут взаимодействовать и влиять на общий результат научного проекта.

Таблица 4.4 – Заинтересованные стороны проекта

Заинтересованные стороны	**Ожидание сторон**
Организация-заказчик	Решение по анализу текстовых данных
Научный руководитель, инженер	Готовая магистерская диссертация
Пользователи	Повышение удобства и снижение времени работы с документами

В таблице 4.5 представлена цель проекта, а также критерии ее достижения.

Таблица 4.5 – Критерии и цели проекта

Цель проекта:	Реализовать программную библиотеку для составления правил по извлечению фактов из текста на русском языке
Ожидаемые результаты:	Возможность применять разработанную библиотеку для решения следующих задач: • Named Entity Recognition • Key-Value Extraction • Fact Extraction
Критерии приемки результата проекта:	Прохождение функционального тестирования решения на стороне заказчика
Требования к результату проекта:	1. Выполнены все пункты технического задания 2. Разработанный функционал полностью соответствует проектным решениям

4.2.1 Ограничения и допущения проекта

Ограничения проекта представлены в таблице 3.6.

Таблица 4.6 - Ограничения проекта

Фактор	**Ограничения**
Бюджет проекта	260000 рублей
Источник финансирования	ООО «Эко-Томск»
Сроки проекта	21.01.2019 – 18.05.2019
Дата утверждения плана управления проектом	21.01.2019
Дата завершения проекта	18.05.2019

Максимальный бюджет настоящего проекта установлен в сумме 260000 рублей, а сроки составляют с 21 января по 18 мая 2019.

4.3 Планирование управления проектом

Планирование проекта предполагает определение условий выполнения всех этапов и задач для установления порядка и последовательности.

Основные этапы планирования:

- определение структуры работ в рамках научно-технического проекта;
- определение участников каждой работы;
- установление продолжительности работ;
- построение графика выполнения проекта.

4.3.1 Структура работ в рамках проекта

Для составления структуры работ определяются ключевые события проекта, затем детальный перечень этапов и работ. На каждый вид работ определяется исполнитель. Распределение исполнителей по данным видам работ приведено в таблице 4.7.

Таблица 4.7 – Распределение исполнителей по работам

Основные этапы	№ этапа (код работ)	Содержание работ	Исполнители
Разработка задания	1	Постановка задачи	Радишевский В.Л. Губин Е.И.
Выбор направления исследования	2	Обзор научно-технической базы	Радишевский В.Л.
	3	Разработка и утверждение ТЗ	Радишевский В.Л. Губин Е.И.
	4	Составление календаря проекта	Радишевский В.Л.
	5	Разработка вариантов исполнения проекта	Радишевский В.Л. Губин Е.И.
Разработка продукта	6	Разработка модуля предобработки текстовых данных	Радишевский В.Л.
	7	Подбор модели синтаксического парсера для русского языка	Радишевский В.Л.
	8	Разработка модуля постобработки и визуализации данных синтаксической структуры текста	Радишевский В.Л
	9	Разработка библиотеки для описания синтаксических структур	Радишевский В.Л.
	10	Проверка работы библиотеки на тестовых данных и правилах	Радишевский В.Л. Губин Е.И.
Оформление отчетной документации	11	Составление пояснительной записки	Радишевский В.Л.

4.3.2 Определение трудоемкости выполнения работ

Для определения ожидаемых сроков выполнения проекта необходимо оценить его трудоемкость. Воспользуемся формулой:

$$t_{ожi} = \frac{3 * t_{\min i} + 2 * t_{\max i}}{5}$$

где $t_{ожi}$ – ожидаемая трудоемкость выполнения *i*-ой работы чел.-дн.;

$t_{\min i}$ – минимально возможная трудоемкость выполнения заданной *i*-ой работы (оптимистическая оценка: в предположении наиболее благоприятного стечения обстоятельств), чел.-дн.;

$t_{\max i}$ – максимально возможная трудоемкость выполнения заданной *i*-ой работы (пессимистическая оценка: в предположении наиболее неблагоприятного стечения обстоятельств), чел.-дн.

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях Tpi, учитывающая параллельность выполнения работ несколькими исполнителями:

$$t_{pi} = \frac{t_{ожi}}{Ч_i}$$

где t_{pi} – продолжительность одной работы раб.-дн.; $t_{ожi}$ – ожидаемая трудоемкость выполнения одной работы, чел.-дн; $Ч_i$ – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел. Для удобства составления календарного плана и графика работ необходимо перевести длительность каждого из этапов из рабочих дней в календарные дни. Для этого воспользуемся следующей формулой:

$$T_{кi} = T_{pi} * k_{кал}$$

где $t_{кi}$ – продолжительность выполнения i-й работы в календарных днях; t_{pi} – продолжительность выполнения i-й работы в рабочих днях; $k_{кал}$ – коэффициент календарности. Коэффициент календарности определяется по следующей формуле:

$$Ткал = \frac{T_{кал}}{T_{кал} - T_{вых} - T_{пр}} = \frac{365}{365 - 66} = 1{,}22$$

где $T_{кал}$ – количество календарных дней в году; $T_{вых}$ – количество выходных дней в году; $T_{пр}$ – количество праздничных дней в году. В соответствии с производственным календарем (для 6-дневной рабочей недели) в 2019 году 365 календарных дней, 299 рабочих дней, 66 выходных/праздничных дней. В таблице 4.8 представлены подробные временные расчеты этапов отдельных видов работ.

Таблица 4.8 – Временные показатели проведения научного-технического проекта

Наименование работы	**Исполнители работы**	**Трудоемкость работ, чел-дни**			**Длительность работ, дни**	
		t_{min}	**t_{max}**	**$t_{ож}$**	**$T_{рабоч}$**	**$T_{кален}$**
Постановка задачи	Инженер	2	4	3,2	3	4
	Научный руководитель	1	3	1,8	2	2
Обзор научно-технической базы	Инженер	7	9	8,2	8	10
Разработка и утверждение ТЗ	Инженер	9	11	9,8	10	12
	Научный руководитель	1	3	1,8	2	2
Составление календаря проекта	Инженер	1	3	1,8	2	2
Разработка вариантов исполнения проекта	Инженер	5	7	6,2	6	8
	Научный руководитель	3	5	4,2	4	5
Разработка модуля предобработки текстовых данных	Инженер	11	13	12,2	12	15
Подбор модели синтаксического парсера для русского языка	Инженер	11	13	12,2	12	15
Разработка модуля постобработки и визуализации данных синтаксической структуры текста	Инженер	11	13	12,2	12	15
Разработка библиотеки для описания синтаксических структур	Инженер	14	16	15,2	15	19
Проверка работы библиотеки на тестовых данных и правилах	Инженер	6	8	7,2	7	9
	Научный руководитель	1	3	1,8	2	2
Составление пояснительной записки	Инженер	6	8	7,2	7	9

Таблица 4.9 – Календарный план-график проекта

Код работы (из ИСР)	Состав участников	Длительность, дни	Дата начала работ	Дата окончания работ	21.01-27.01	28.01-03.02	04.02-10.02	11.02-17.02	18.02-24.02	25.02-03.03	04.03-10.03	11.03-17.03	18.03-24.03	25.03-31.03	01.04-07.04	08.04-14.04	15.04-21.04	22.04-28.04	29.04-05.05	06.05-12.05	13.05-19.05
1	Радишевский В.Л.	4	21.01.2019	24.01.2019																	
	Губин Е.И.	2	21.01.2019	22.01.2019																	
2	Радишевский В.Л.	10	25.01.2019	03.02.2019																	
3	Радишевский В.Л.	12	04.02.2019	15.02.2019																	
	Губин Е.И.	2	04.02.2019	05.02.2019																	
4	Радишевский В.Л.	2	16.02.2019	17.02.2019																	
5	Радишевский В.Л.	8	18.02.2019	25.02.2019																	
	Губин Е.И.	5	18.02.2019	22.02.2019																	
6	Радишевский В.Л.	15	26.02.2019	12.03.2019																	
7	Радишевский В.Л.	15	13.03.2019	27.03.2019																	
8	Радишевский В.Л.	15	28.03.2019	11.04.2019																	
9	Радишевский В.Л.	19	12.04.2019	30.04.2019																	
10	Радишевский В.Л.	9	01.05.2019	09.05.2019																	
	Губин Е.И.	2	01.05.2019	02.05.2019																	
11	Радишевский В.Л.	9	10.05.2019	18.05.2019																	

4.4 Бюджет проекта

Для полноты и достоверности учета всех расходов сгруппируем все затраты по следующим статьям

– затраты на материалы;

– затраты на амортизацию;

– основная заработная плата исполнителей;

– дополнительная заработная плата исполнителей темы;

– отчисления во внебюджетные фонды (страховые отчисления);

– накладные расходы.

4.4.1 Материальные затраты

В расчет взяты только затраты на канцелярские товары в размере 1000 рублей.

4.4.2 Амортизационные отчисления

Для работы над проектом использовался ноутбук. Амортизацию рассчитаем линейным способом.

Первоначальная стоимость ПК 60000 рублей; срок полезного использования для машин офисных код 330.28.23.23 составляет 2-3 года, берем 3 года; планируется использовать ПК для написания работы в течение 4 месяцев. Тогда:

- месячная норма амортизации:

$$Ан = \frac{1}{n} * 100\% = \frac{1}{12 \times 3} \times 100\% = 2{,}8\%$$

где *n* - количество месяцев полезного срока эксплуатации ОС.

- Ежемесячные амортизационные отчисления:

$$Аг = 60000 \times 2{,}8 = 1\,680 \text{ рублей}$$

- Итоговая сумма амортизации основных средств:

$$А = 1680 \times 4 = 6720 \text{ рублей}$$

Таким образом, в материальные затраты необходимо включить сумму амортизации основных средств в сумме 6720 руб.

4.4.3 Заработная плата исполнителей проекта

Заработная плата рассчитывается из суммы заработной платы исполнителя и научного руководителя исходя из трудоемкости каждого этапа и занятости каждого из них на данном этапе по формуле

$$З_{зп} = З_{осн} + З_{доп}$$

где $З_{осн}$ – основная заработная плата; $З_{доп}$ – дополнительная заработная плата.

Рассчитаем основную заработную плату:

$$З_{осн} = З_{дн} \times Т_р \times (1 + К_{пр} + К_д) \times К_р$$

$З_{дн}$ – среднедневная заработная плата, руб.

$К_{пр}$ – премиальный коэффициент (т.е. 30% от $З_{дн}$);

$К_д$ – коэффициент доплат и надбавок составляет примерно 0,2 – 0,5 (в НИИ и на промышленных предприятиях – за расширение сфер обслуживания, за профессиональное мастерство, за вредные условия: 15-20% от $З_{дн}$);

$К_р$ – районный коэффициент (для Томска 1,3);

$Т_р$ – продолжительность работ, выполняемых работником, раб. Дни

Рассчитаем среднедневную заработную плату по формуле:

$$З_{дн} = \frac{З_м \times М}{F_д}$$

$З_м$ – оклад работника за месяц, руб.

М – количество месяцев работы без отпуска в течение года:

при отпуске в 48 раб. дней М=10,4 месяца, 6-дневная неделя;

$F_д$ – действительный годовой фонд рабочего времени персонала, раб. дн.

Таблица 4.10 – Баланс рабочего времени (для 6-дневной недели)

Показатели рабочего времени	Дни
Календарные дни	365
Нерабочие дни (праздники/выходные)	66
Потери рабочего времени (отпуск/невыходы по болезни)	56
Действительный годовой фонд рабочего времени	243

Для расчета основной заработной платы инженера берем оклад, равный окладу 21760 руб. Для расчета основной заработной платы руководителя в расчет возьмем оклад, равный 33664 руб.

Таблица 4.11 – Расчет основной заработной платы

Исполнители	$З_{дн}$, руб.	$К_{пр}$	$К_{д}$	$К_{р}$	$Т_{р}$	$З_{осн}$, руб.
Инженер	931	0	0	1,3	94	113 768
Научный руководитель	1440	0,3	0,2	1,3	10	28 080

В дополнительную заработную плату входят суммы выплат, предусмотренные трудовым кодексом, например, оплата ежегодных и дополнительных отпусков, оплата времени, связанного с выполнением государственных и общественных обязанностей и т.д. запланируем дополнительную заработную плату в размере 15 % от основной заработной платы исполнителей,

В таблице 4.12 представлен расчет затрат на заработную плану исполнителей.

Таблица 4.12 – Затраты на заработную плату без отчислений

Исполнители	$З_{осн}$, руб.	$З_{доп}$, руб.	$З_{зп}$, руб.
Инженер	113 768	17 065	130 833
Научный руководитель	28 080	4 218	32 292
Итого	141 848	21 283	163 125

4.4.4 Отчисления во внебюджетные фонды (страховые отчисления)

Общие тарифы страховых взносов в 2019 году в ИФНС:

22% — на пенсионное страхование;

2,9% — страхование по временной нетрудоспособности;

5,1% — медицинское страхование.

Величина отчислений во внебюджетные фонды определяется исходя из формулы:

$$З_{внеб} = k_{внеб} * (З_{осн} + З_{доп})$$

где $k_{внеб}$ – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

Таким образом, с учетом результатов расчета зарплат на заработную плату величина отчислений во внебюджетные фонды составляет:

$$З_{внеб} = 0{,}3 * 163125 = 48937 \text{ руб}$$

4.4.5 Накладные расходы

При выполнении проекта могут возникнуть косвенные издержки – накладные расходы, возникающие дополнительно к основным затратам, например, на консультационные услуги, оплату коммунальных услуг, расходы на услуги связи (телефон, интернет) и так далее.

Их величина определяется по следующей формуле:

$$З_{накл} = (\text{сумма статей расходов}) \cdot k_{нр};$$

где $k_{нр}$ – коэффициент, учитывающий накладные расходы.

Величину коэффициента накладных расходов можно взять в размере 16%.

$$З_{накл} = (1\,000 + 6\,720 + 163\,125 + 48\,937) \cdot 0{,}16 = 35\,166 \text{ рублей.}$$

4.4.6 Формирование бюджета

После выполнения всех расчетов по статьям можно определить плановую общую себестоимость проекта «Комбинированный подход к

извлечению структурированных данных для языков со свободным порядком слов».

Таблица 4.13 – Бюджет затрат

Наименование	Сумма, руб.	Удельный вес, %
Затраты на материалы	1 000	0,4
Затраты на амортизацию	6 720	2,6
Затраты на основную заработную плату	141 848	55,7
Затраты на дополнительную заработную плату	21 283	8,3
Страховые взносы	48 937	19,2
Накладные расходы	35 166	13,8
Общий бюджет	255 518	100%

Исходя из расчета бюджета затрат следует, что наибольшая его часть приходится на основную и дополнительную заработную плату исполнителей (64 %). Также необходимо отметить, что расходы на страховые взносы (19,2 %) составляют немаловажную часть расходов. Затраты на амортизацию, материалы и накладные расходы составляют небольшую долю (суммарно 16,8 %). Это связано с отсутствие необходимости использования значительно дорогостоящего оборудования и материалов.

4.4.7 Риски

Риски в реализации проекта включают в себя возможные неопределенные события, которые могут возникнуть в проекте и вызвать последствия, которые повлекут за собой нежелательные эффекты. Оценка рисков проекта представлена в таблице 4.14. Для каждого из них даны рекомендации по смягчению их воздействия.

Таблица 4.14 – Реестр рисков

Реестр рисков	Риск	Потенциальное воздействие	Вероятность наступления	Влияние риска	Уровень риска	Способы смягчения	Условия наступления
1	Несоответствие разработанной и требуемой функциональности	Недостаточная функциональность может привести к неконкурентоспособности устройства	2	3	средний	Прототипирование, разработка сценариев использования, участие потенциальных пользователе	Ошибки при постановке задачи, недостаточный анализ качества разработки и ее перспективности на рынке
2	Постоянный поток изменений требований	Задержки выполнения работ	2	2	низкий	Установка ограничений для внесения изменений, итеративность разработки (внесения изменений в следующих итерациях)	Ошибки при постановке задачи
3	Технологическое отставание	Неконкурентоспособность устройства	2	2	низкий	Технический анализ, анализ стоимости, прототипирование	Не достаточная оценка существующих аналогов
4	Недостаточная производительность	Неконкурентоспособность устройства	1	3	средний	Проведение сравнительного тестирования, прототипирование	Ошибки при постановке задачи, недостаточный анализ качества разработки и ее перспективности на рынке

В результате данного этапа были рассмотрены возможные риски при реализации настоящей работы. Основная часть рисков может привести к неконкурентоспособности разработанного решения. Однако их воздействие можно минимизировать благодаря проведению прототипирования, итеративности разработки, проведению технического анализа стоимости и проведению сравнительного тестирования.

4.4.8 Интегральный финансовый показатель эффективности

Общая трудоемкость разработки решения составила 133 человеко-дня. Общий бюджет проекта составил 255 518 рублей. Исходя из ограничений, накладываемых на проект, максимальный бюджет не должен превышать 260000 рублей. Расчет интегральный финансовый показатель разработки рассчитывает по следующей формуле:

$$I_{\text{фин}} = \frac{\Phi_{\text{р}}}{\Phi_{max}}$$

Где $I_{фин}$ – интегральный финансовый показатель разработки, $\Phi_р$ – стоимость исполнения работ, Φ_{max} – максимально допустимая стоимость исполнения проекта. Таким образом значения финансового показателя составляет:

$$I_{\text{фин}} = \frac{255518\ \text{руб}}{260000\ \text{руб}} = 0{,}98$$

4.5 Выводы

Результаты оценки востребованности разработки можно считать положительными, поскольку, во-первых, были выявлены потенциальные потребители настоящего решения. Во-вторых, в результате анализа конкурентоспособности выяснилось, что разработанное решение является более предпочтительным для широкого класса задач и обладает достаточными конкурентными преимуществами благодаря новизне метода. В-третьих, проведенный SWOT анализ показал перспективность разработки. Расширение функционала, оказание услуг по настройки или консультированию под каждую конкретную предметную область совместно с поддержанием стабильной ценовой политики позволит сохранять свою конкурентоспособность.

Кроме того, в данной главе был разработан план и сформирован бюджет технического решения. Продолжительность проекта составила 94 рабочих дня, а общий бюджет затрат составил 255 518 рублей. Таким образом план-график и бюджет проекта успешно укладываются в ограничения заказчика. Разработанный реестр рисков отражает потенциальные пути преодоления внешних и внутренних рисков и способствует успешной реализации проекта, а также его дальнейшее существование, а рассчитанный интегральный финансовый показатель эффективности ($I_{фин}$ = *0,98*) свидетельствует о возможности реализации настоящего проекта.

5 Социальная ответственность

В рамках настоящей работы был предложен новый метод к извлечению структурированных данных для языков со свободным порядком слов, а также разработана программная библиотека, предназначенная для составления специальных правил по извлечению информации из текста. Алгоритм предполагает использование контекстно-свободных грамматик, а также современных моделей синтаксического анализа текста с последующей постобработкой выделения ключевых фраз и смыслов из текста.

В разделе будут рассмотрены опасные и вредные факторы, оказывающие влияние на производственную деятельность инженера-программиста. Исследовано рабочее место программиста и помещение, в котором он находится. Разработка осуществлялась в компьютерном классе Кибернетического центра ТПУ. Основные средства работы – персональный компьютер и локальная вычислительная сеть с выходом в Интернет. Рассмотрены воздействия объекта исследования на окружающую среду, правовые и организационные вопросы, а также мероприятия в чрезвычайных ситуациях.

5.1 Правовые и организационные вопросы обеспечения безопасности.

Нормативное регулирование охраны труда при осуществлении трудовой деятельности за компьютерами осуществляется посредством следующих документов:

- Трудовой кодекс РФ;
- Приказ Минздравсоцразвития России "Об утверждении перечней вредных и (или) опасных производственных факторов и работ, при выполнении которых проводятся обязательные предварительные и периодические медицинские осмотры (обследования), и Порядка проведения обязательных предварительных и периодических медицинских осмотров (обследований) работников, занятых на тяжелых

работах и на работах с вредными и (или) опасными условиями труда" от 12.04.2011 N 302н;

- Федеральный закон "О специальной оценке условий труда" от 28 декабря 2013 г. N 426;
- СаНПиН 2.2.2/2.4.1340-03"Гигиенические требования к персональным электронно-вычислительным машинам и организации работы" (с изменениями на 21 июня 2016 года);
- Типовая инструкция по охране труда при работе на персональном компьютере (ПК, ПЭВМ) ТОИ Р-45-084-01 и др.

Согласно "Трудовому кодексу Российской Федерации" от 30.12.2001 N 197-ФЗ работодатель обязан обеспечить нормальные условия для выполнения работниками норм выработки. К таким условиям, в частности, относятся:

- исправное состояние помещений, сооружений, машин, технологической оснастки и оборудования;
- своевременное обеспечение технической и иной необходимой для работы документацией;
- надлежащее качество материалов, инструментов, иных средств и предметов, необходимых для выполнения работы, их своевременное предоставление работнику;
- условия труда, соответствующие требованиям охраны труда и безопасности производства;
- создание и функционирование системы управления охраной труда;
- соответствующие требованиям охраны труда условия труда на каждом рабочем месте;
- режим труда и отдыха работников в соответствии с трудовым законодательством и иными нормативными правовыми актами, содержащими нормы трудового права;
- обучение безопасным методам и приемам выполнения работ и оказанию первой помощи пострадавшим на производстве, проведение

инструктажа по охране труда, стажировки на рабочем месте и проверки знания требований охраны труда;

- недопущение к работе лиц, не прошедших в установленном порядке обучение и инструктаж по охране труда, стажировку и проверку знаний требований охраны труда;
- организацию контроля за состоянием условий труда на рабочих местах, а также за правильностью применения работниками средств индивидуальной и коллективной защиты;
- информирование работников об условиях и охране труда на рабочих местах, о риске повреждения здоровья, предоставляемых им гарантиях, полагающихся им компенсациях и средствах индивидуальной защиты;
- принятие мер по предотвращению аварийных ситуаций, сохранению жизни и здоровья работников при возникновении таких ситуаций, в том числе по оказанию пострадавшим первой помощи;
- санитарно-бытовое обслуживание и медицинское обеспечение работников в соответствии с требованиями охраны труда, а также доставку работников, заболевших на рабочем месте, в медицинскую организацию в случае необходимости оказания им неотложной медицинской помощи.

Федеральный закон "О специальной оценке условий труда" от 28.12.2013 N 426-ФЗ регламентирует проведение спецоценки, если деятельность работников предприятия предусматривает непрерывную работу за компьютеризированными системами. Результаты проведенной спецоценки влияют на установление гарантий и компенсаций работникам согласно Трудовому кодексу РФ. Так, сотрудники, условия труда на рабочих местах, которых признаны вредными, в зависимости от степени вредности имеют право на сокращенную рабочую неделю не более 36 часов, дополнительный отпуск не менее семи календарных дней и/или компенсацию в размере 4% от оклада.

Нормативные положения СанПиНа 2.2.2/2.4.1340-03 предъявляют определенные требования к оснащению рабочего места, предусматривающего длительную работу за ПК:

Таблица 5.1 – Нормы оборудования рабочих мест

Высота перегородок, разделяющих рабочие места	Не менее 1,5 метров
Ширина рабочего стола	От 80 до 140 см
Глубина рабочего стола	От 80 до 100 см
Высота рабочего стола	7,25 см
Расстояние от глаз до монитора	От 60 до 70 см
Расстояние клавиатуры от края стола	От 10 до 30 см
Сидение	Должно позволять регулировку по высоте, повороту и углу наклона спинки (регулировки должны быть независимыми друг от друга)
Подставка для ног	Ширина – от 30 см, глубина – от 40 см, с углом наклона до 20 градусов

5.2 Производственная безопасность.

5.2.1. Анализ выявленных вредных и опасных факторов и обоснование мероприятий по снижению воздействия

Идентифицируем вредные и опасные факторы, возникающие при разработке проекта, в соответствии с ГОСТом 12.0.003-2015 «Опасные и вредные производственные факторы. Классификация».

Таблица 5.2 – Вредные и опасные факторы

Факторы (ГОСТ 12.0.003-2015)	Этапы работ		Нормативные документы
	Разработка	Эксплуатация	
Отклонение показателей микроклимата	+	+	СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений
Превышение уровня шума	+	+	СанПиН 2.2.4.3359-16 "Санитарно-эпидемиологические требования к физическим факторам на рабочих местах"
Отсутствие или Недостаток естественного света	+	+	СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95
Повышенная напряженность магнитного поля	+	+	СаНПиН 2.2.2/2.4.1340-03 "Гигиенические требования к персональным электронно-вычислительным машинам и организации работы" (с изменениями на 21 июня 2016 года)
Психофизиологический фактор	+	+	СаНПиН 2.2.2/2.4.1340-03 "Гигиенические требования к персональным электронно-вычислительным машинам и организации работы" (с изменениями на 21 июня 2016 года)

5.2.2 Микроклимат

Длительное воздействие на человека неблагоприятных метеорологических условий резко уменьшает его самочувствие, снижает производительность труда и часто приводит к заболеваниям.

Влажность воздуха оказывает значительное влияние на терморегуляцию организма человека. Высокая относительная влажность воздуха при его высокой температуре способствует перегреванию организма. Низкая влажность вызывает пересыхание слизистых оболочек дыхательных путей. Подвижность воздуха весьма эффективно способствует теплоотдаче,

что является положительным явлением при высокой температуре окружающей среды и отрицательным – при низкой.

Оптимальные и допустимые показатели температуры, относительной влажности и скорости движения воздуха в рабочей зоне производственных помещений должны соответствовать значениям, согласно СанПиН 2.2.4.548 – 96 для категории тяжести работ 1а (к категории 1а относятся работы с интенсивностью энерготрат до 120 ккал/ч (до 139 Вт), производимые сидя и сопровождающиеся незначительным физическим напряжением)

Таблица 5.3 – Оптимальные нормы микроклимата.

Период года	Температура,°С	Температура поверхностей,°С	Относительная влажность, %	Скорость движения воздуха, м/с
Холодный	22-24	21-25	60-40	0,1
Теплый	23-25	22-26	60-40	0,1

Таблица 5.4 − Допустимые величины показателей микроклимата на рабочих местах производственных помещений

Период года	Температура воздуха, °С		Относительная влажность воздуха, %	Скорость движения воздуха, не более, м/с	
	Диапазон ниже оптимальных величин	Диапазон выше оптимальных величин		при температуре воздуха ниже оптимальной	при температуре воздуха выше оптимальной
Холодный	20,0 – 21,9	24,1 – 25,0	15 – 75	0,1	0,1
Тёплый	21,0 – 22,9	25,1 – 28,0	15 – 75	0,1	0,2

При обеспечении допустимых величин микроклимата на рабочих местах перепад температуры воздуха по высоте должен быть не более 3°С, перепад температуры воздуха по горизонтали, а также ее изменения в течение смены не должны превышать – 4°С. При этом абсолютные значения температуры воздуха не должны выходить за пределы оптимальных величин.

При температуре воздуха на рабочих местах 25 °С и выше максимально допустимые величины относительной влажности воздуха не должны выходить за пределы:

- 70% – при температуре воздуха 25 °С,
- 65% – при температуре воздуха 26 °С,
- 60% – при температуре воздуха 27 °С,
- 55% – при температуре воздуха 28 °С.

Помещение, где выполнялась работа, было обследовано на соответствие требованиям СанПиН 2.2.4.548 –96. Измерения производились при помощи портативного термогигрометра «ИВТМ – 7» (абсолютная погрешность при измерении температуры ±0,2 – ±0,5, основная погрешность измерения влажности ±0,2). Результаты обследования приведены в таблице ниже:

Таблица 5.5 – Результаты измерений параметров макроклимата

Период года	Температура воздуха, °С	Относительная влажность воздуха, %	Скорость движения воздуха, не более, м/с
Холодный (январь)	0,1м – 22,1 1,5м – 22,1	60	0
Теплый (июль)	0,1м – 23,0 0,5м – 23,0	59	0

Результаты измерений соответствуют допустимым значениям нормативов, следовательно микроклимат помещения удовлетворяет требованиям санитархных норм и правил.

5.2.3 Уровень шума

Шум на рабочем месте оказывает раздражающее влияние на работника, повышает его утомляемость, а при выполнении задач, требующих внимания и сосредоточенности, способен привести к росту ошибок и увеличению продолжительности выполнения задания. Согласно СанПиН 2.2.4.3359-16 нормативным эквивалентным уровнем звука на рабочих местах является 50 дБА.

Источником шумовых помех на рабочем месте могут выступать вентиляционные установки, кондиционеры, ЭВМ и его периферийные устройства, а также серверные комнаты.

В рабочем помещении производились измерения многофункциональным шумомер, виброметром и анализатором спектра «ЭКОФИЗИКА-110А». погрешность измерений составляет ± 0,7 дБА. Результаты измерений представлены в таблице 5.6:

Таблица 5.6 – Результаты измерений уровня звука

Рабочая операция	Уровень звука, дБА		Продолжительность операции, мин	
	Результаты измерений (не менее трех)	Эквивалентный уровень за операцию	Результаты наблюдений	Средняя
Работа за ПК	45; 47; 45	45,6	20, 20, 20	20

В результате вычисления на рабочем месте измеренных величин показателей шума эквивалентный уровень звука за 8 - часовой рабочий день составляет 45,6 дБА, что соответствует нормативным значениям.

Снизить уровень шума в помещениях можно использованием звукопоглощающих материалов с максимальными коэффициентами звукопоглощения в области частот 63-8000 Гц для отделки стен и потолка помещений. Дополнительный звукопоглощающий эффект создают однотонные занавески из плотной ткани, повешенные в складку на расстоянии 15-20 см от ограждения. Ширина занавески должна быть в 2 раза больше ширины окна.

5.2.4 Отсутствие или недостаток естественного света

В помещении при работе с ПК должно быть естественное и искусственное освещение. Естественное освещение обеспечивается через оконные проемы с коэффициентом естественного освещения КЕО не ниже 1,2% в зонах с устойчивым снежным покровом и не ниже 1,5% на остальной

территории. Световой поток из оконного проема должен падать на рабочее место оператора с левой стороны.

Искусственное освещение в помещениях эксплуатации компьютеров должно осуществляться системой общего равномерного освещения.

Освещенность на поверхности стола в зоне размещения документа должна быть 300-500 лк. Допускается установка светильников местного освещения для подсветки документов. Местное освещение не должно создавать бликов на поверхности экрана и увеличивать освещенность экрана более 300 лк. Прямую блескость от источников освещения следует ограничить. Яркость светящихся поверхностей (окна, светильники), находящихся в поле зрения, должна быть не более 200 кд/м^2.

Отраженная блескость на рабочих поверхностях ограничивается за счет правильного выбора светильника и расположения рабочих мест по отношению к естественному источнику света. Яркость бликов на экране монитора не должна превышать 40 кд/м^2. Показатель ослепленности для источников общего искусственного освещения в помещениях должен быть не более 20, показатель дискомфорта в административно-общественных помещениях не более 40. Соотношение яркости между рабочими поверхностями не должно превышать 3:1 – 5:1, а между рабочими поверхностями и поверхностями стен и оборудования 10:1.

Для искусственного освещения помещений с персональными компьютерами следует применять светильники типа ЛПО36 с зеркализованными решетками, укомплектованные высокочастотными пускорегулирующими аппаратами. Допускается применять светильники прямого света, преимущественно отраженного света типа ЛПО13, ЛПО5, ЛСО4, ЛПО34, ЛПО31 с люминисцентными лампами типа ЛБ. Допускается применение светильников местного освещения с лампами накаливания. Светильники должны располагаться в виде сплошных или прерывистых линий сбоку от рабочих мест параллельно линии зрения пользователя при разном расположении компьютеров. При периметральном расположении —

линии светильников должны располагаться локализованно над рабочим столом ближе к его переднему краю, обращенному к оператору. Защитный угол светильников должен быть не менее 40 градусов. Светильники местного освещения должны иметь непросвечивающийся отражатель с защитным углом не менее 40 градусов.

В рабочем помещении производились измерения уровня искусственного освещения люксометром-яркометром-пульметром «Эколайт-01 DIN». При этом погрешность измерения освещенности составляла ±8%, яркости ±10%, а коэффициент пульсации ±10%.

В результате измерений освещенность рабочих поверхностей составила 350 лк, что соответствуют допустимым значениям нормативов.

Таблица 5.7 – Характеристика осветительного оборудования

Наименование рабочей зоны	Тип светильников	Тип ламп	Мощность ламп, Вт	Высота подвеса, м	Доля негорящих ламп, %
Компьютерная аудитория кафедры	ЛПО	ЛБ	40	3	0

Для обеспечения нормативных значений освещенности в помещениях следует проводить чистку стекол оконных проемов и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

5.2.5 Повышенная напряженность магнитного поля

Электромагнитные поля, характеризующиеся напряженностями электрических и магнитных полей, наиболее вредны для организма человека. Основным источником этих проблем, связанных с охраной здоровья людей, использующих в своей работе автоматизированные информационные системы на основе персональных компьютеров, являются дисплеи (мониторы), они представляют собой источники наиболее вредных излучений, неблагоприятно влияющих на здоровье человека.

Предельно допустимые значения излучений от ЭВМ в соответствии с СанПиН 2.2.2/2.4.1340-03 приведены в таблице 5.8.

Таблица 5.8 Допустимые уровни ЭМП, создаваемых

Наименование параметров		ВДУ ЭМП
Напряженность электрического поля	В диапазоне частот 5 Гц – 2 кГц	25 В/м
	В диапазоне частот 2 кГц – 400 кГц	2,5 В/м
Плотность магнитного потока	В диапазоне частот 5 Гц – 2 кГц	250 нТл
	В диапазоне частот 2 кГц – 400 кГц	25 нТл
Электростатический потенциал экрана видеомонитора		500 В

На рабочем месте были проведены исследования измерителем напряженности электрических и магнитных полей ПЗ-80ЕН (погрешность измерений ±15%).

Таблица 5.9 Фактические значения уровня ЭМП

Места измерений	Расстояние от источника, м	Высота от пола, м	Напряженность электрического поля	Плотность магнитного потока
В диапазоне частот 5 Гц – 2 кГц	0,5	1,0	9	10
В диапазоне частот 2 кГц – 400 кГц	0,5	1,0	0,27	0

В результате исследований уровень электромагнитного поля, создаваемое на рабочем месте не превышают допустимые уровни и соответствует требованиям СаНПиН 2.2.2/2.4.1340-03 "Гигиенические требования к персональным электронно-вычислительным машинам и организации работы" (с изменениями на 21 июня 2016 года).

5.2.6 Психофизиологический фактор

Настоящей работе сопутствует ряд вредных психофизиологических факторов: напряжение зрения и внимания; эмоциональные, интеллектуальные и длительные статические нагрузки; большая монотонность труда; большой объем обрабатываемой информации в единицу времени; нерациональная организация рабочего места.

В результате воздействия данных факторов к концу рабочего дня рабочий испытывает неприятные ощущения: переутомление глаз, головная боль, тянущие боли в мышцах шеи, рук и спины, снижение концентрации внимания.

В целом продолжительность непрерывной работы за компьютером не должна превышать 2-х часов. Основная работа за компьютером предусматривает не менее 50 % времени в течение рабочей смены или рабочего дня нахождения за ним. Время перерыва зависит от вида и сложности осуществляемой работы путем деления на группы. Выделяют 3 группы: А (работа по считыванию информации с экрана компьютера с предварительным запросом), Б (работа по вводу информации), В (творческая работа в режиме диалога с компьютером). Во время перерывов следует выполнять специальную гимнастику для снятия напряжения с глаз. Рекомендуемый комплекс упражнения представлен в Приложении 8 к СанПиН 2.2.2/2.4.1340-03. Выполнять какую-либо работу, не связанную с компьютером, во время перерыва нельзя. Потому как перерыв приравнивается к времени отдыха. А в соответствии со ст. 106 ТК РФ время отдыха – это свободное от исполнения трудовых обязанностей время, которое работник может использовать по своему усмотрению.

5.3 Экологическая безопасность

Деятельность по разработке ПО не связана с производством, поэтому влияние на окружающую среду минимально. При работе над проектом применяются рекомендации по минимизации влияния на окружающую среду.

С учетом того, что наиболее значительную часть твердых бытовых отходов (до 40 % в развитых странах) составляет бумага и картон — бумага для печати, упаковка и упаковочные материалы, актуальным является осуществление их утилизации для повторного использования в промышленном производстве, что является наиболее экономически эффективным способом обращения с отходами согласно ГОСТ Р 55090-2012

«Ресурсосбережение. Обращение с отходами. Рекомендации по утилизации отходов бумаги» [51].

При завершении срока службы ПК их можно отнести к отходам электронной промышленности. Переработка таких отходов осуществляется разделением на однородные компоненты, химическим выделением пригодных для дальнейшего использования компонентов и направлением их для дальнейшего использования согласно ГОСТ Р 55102-2012 «Ресурсосбережение. Обращение с отходами. Руководство по безопасному сбору, хранению, транспортированию и разборке отработавшего электротехнического и электронного оборудования, за исключением ртутьсодержащих устройств и приборов» [52].

Перечень элементов и содержащее их отработанное электротехническое и электронное оборудование, которые должны быть раздельно собраны при выводе отработавшего электротехнического и электронного оборудования из эксплуатации:

- конденсаторы, содержащие ПХБ;
- печатные платы и других устройств с площадью поверхности больше 10 $см^2$;
- картриджи;
- пластик;
- электронно-лучевые трубки;
- элементы отработавшего электротехнического и электронного оборудования;
- газоразрядные лампы;
- жидкокристаллические экраны (если необходимо, вместе с корпусом) с поверхностью более 100 $см^2$ и все экраны с подсветкой газоразрядными лампами;
- внешние электрические кабели; - элементы, содержащие огнеупорные керамические слои;

- конденсаторы, содержащие электролит (размер хотя бы одной из сторон конденсатора должен быть 25 мм или более).

Люминесцентные лампы относят к ртутьсодержащим отходам, и для их утилизации действует Постановление Правительства РФ от 03.09.2010 № 681 (ред. от 01.10.2013) «Об утверждении Правил обращения с отходами производства и потребления в части осветительных устройств, электрических ламп, ненадлежащие сбор, накопление, использование, обезвреживание, транспортирование и размещение которых может повлечь причинение вреда жизни, здоровью граждан, вреда животным, растениям и окружающей среде» [53]. Согласно постановлению, устанавливается порядок обращения с отходами производства и потребления в части осветительных устройств, электрических ламп, ненадлежащие сбор, накопление, использование, обезвреживание, транспортирование и размещение которых может повлечь причинение вреда жизни, здоровью граждан, вреда животным, растениям и окружающей среде.

Не допускается самостоятельное обезвреживание, использование, транспортирование и размещение отработанных ртутьсодержащих ламп потребителями отработанных ртутьсодержащих ламп, а также их накопление в местах, являющихся общим имуществом собственников помещений многоквартирного дома, за исключением размещения в местах первичного сбора и размещения и транспортирования до них.

Сбор отработанных ртутьсодержащих ламп у потребителей осуществляют специализированные организации.

Отходы, не подлежащие переработке и вторичному использованию подлежат захоронению на полигонах.

5.4 Безопасность в чрезвычайных ситуациях

5.4.1 Перечень возможных ЧС на объекте

Компьютерная аудитория кафедры, согласно НПБ 105-03 «Определение категорий помещений, зданий и наружных установок по

взрывопожарной и пожарной опасности» относится к категории В3 по пожароопасности, содержит вещества и материалы, способные при взаимодействии с водой, кислородом воздуха или друг с другом только гореть.

Для минимизации возможности возникновения фактора пожара необходимо проводит пожарную профилактику. Пожарная профилактика представляет собой комплекс организационных и технических мероприятий, направленных на обеспечение безопасности людей, на предотвращении пожара, ограничение его распространения, а также создание условий для успешного тушения пожара. Для профилактики пожара чрезвычайно важна правильная оценка пожароопасности, определение опасных факторов и обоснование способов и средств пожар предупреждения и защиты.

Одно из условий обеспечения пожаробезопасности - ликвидация возможных источников воспламенения.

Обогревание помещения открытыми электронагревательными приборами могут привести к пожару, т.к. в помещении находятся бумажные документы и справочная литература. Следовательно, использование открытого нагревательного прибора неприемлемо.

5.4.2 Меры по предотвращению и ликвидации ЧС и их последствий

В целях предотвращения пожара предлагается:

- проводить с сотрудниками противопожарный инструктаж;
- проводить плановый осмотр и своевременно устранять все неисправности в электроприборах;
- предотвращать небезопасное хранение легковоспламеняющихся жидкостей;
- оснащение помещения автоматической системой обнаружения пожара;
- оснащение помещения автоматической системой оповещения о пожаре.

Согласно СП 5.13130.2009 «Системы противопожарной защиты. Установки пожарной сигнализации и пожаротушения автоматические.

Нормы и правила проектирования (с Изменением N 1)» [54] и СП 3.13130.2009 «Системы противопожарной защиты. Система оповещения и управления эвакуацией людей при пожаре. Требования пожарной безопасности» [55] помещения с категорией В3 должны оснащаться системой автоматической пожарной сигнализации и система оповещения и управления эвакуацией людей при пожарах.

Компьютерная аудитория кафедры оснащена одним пожарным, а также двумя дымовыми извещателями, для оповещения о пожаре установлена звуковая сирена.

5.5 Выводы

В результате проведенного анализа были выявлены вредные и опасные производственные факторы для компьютерной аудитории кафедры, и работы, связанной с проектом.

Для вредных и опасных факторов, из нормативных документов, были определены значения нормативных показателей, которые сравнивались со значениями, действующими при разработке проекта. Приведены рекомендации по улучшения условий труда и минимизации влияния на работника вредных и опасных факторов. Проведен анализ воздействия на окружающую среду и обозначены проблемы утилизации отходов. Определена категория помещения по пожароопасности, а также действия по минимизации риска возникновения пожара.

Также были рассмотрены правовые нормы трудового законодательства, применимые к условиям настоящего проекта, определены основные требования к организации рабочего места.

После проведенного анализа можно сделать вывод, что рабочее место соответствует всем нормативным требованиям производственной безопасности и охраны труда.

ЗАКЛЮЧЕНИЕ

В ходе данной магистерской диссертации был предложен комбинированный подход для извлечения информации из текста. Данный подход заключается в использовании контекстно-свободных грамматик совместно с синтаксическими шаблонами, извлекаемых из деревьев на основе грамматик зависимости. На основе предложенного подхода разработана программная библиотека, включающая собственные шаблоны для извлечения подграфов из деревьев зависимостей с учетом морфологических, синтаксических признаков и результатов извлечения с помощью контекстно-свободных грамматик.

В работе показано, что разработанная библиотека успешно применяется на примере задачи по извлечению информации из резюме соискателей. Кроме того, было создано веб-приложение для анализа документов и текстов резюме. Получены шаблоны для извлечения всей необходимой информации, включающей контактные данные, описание опыта работы, различные навыки, компетенции, опыт работы с технологиями.

В качестве дальнейших шагов планируется расширение функционала для составления шаблонов, применение готовых общедоступных тезаурусов / словарей для возможности использования синонимов, публикация проекта в открытый доступ, а также разработка банка основных шаблонов.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Moens, M.-F. Information Extraction: Algorithms and Prospects in a Retrieval Context. – Netherlands: Springer. – 2009. – 255 p.
2. Большакова Е. И. И др. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика: учеб.пособие – М.: МИЭМ, 2011. — 272 с.
3. Jurafsky D., Martin J. H. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition // Prentice Hall series in artificial intelligence. – 2009.
4. Chowdhury G. G. Natural language processing // Annual review of information science and technology. – 2003. – Vol. 37. – №. 1. – P. 51-89.
5. Кольцова Д. А., Кольцов С. В. История и развитие машинного перевода // Русский язык и культура в зеркале перевода: IX Международная научная конференция «Русский язык и культура в зеркале перевода». – 2019. – Т. 10. – С. 130.
6. Wortzel A. ELIZA REDUX: A Mutable Iteration // Leonardo. – 2007. – Vol. 40. – №. 1. – P. 31-36.
7. Winograd T. Procedures as a representation for data in a computer program for understanding natural language. – Massachusetts. Inst. Of Tech. – Cambridge. – 1971. – AI Technical Report № 84.
8. Thuraisingham B. A primer for understanding and applying data mining // It Professional. – 2000. – Vol. 2. – №. 1. – P. 28-31.
9. Waldrop M. M. The chips are down for Moore’s law // Nature News. – 2016. – Vol. 530. – №. 7589. – P. 144.
10. Sha F., Pereira F. Shallow parsing with conditional random fields // Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Vol. 1. – Association for Computational Linguistics. –2003. – P. 134-141.
11. Collins M., Koehn P., Kučerová I. Clause restructuring for statistical machine translation // Proceedings of the 43rd annual meeting on association for

computational linguistics. – Association for Computational Linguistics. – 2005. – P. 531-540.

12. Kaelbling L. P., Littman M. L., Moore A. W. Reinforcement learning: A survey //Journal of artificial intelligence research. – 1996. – Vol. 4. – P. 237-285.

13. Hofmann T. Unsupervised learning by probabilistic latent semantic analysis //Machine learning. – 2001. – Vol. 42. – №. 1-2. – P. 177-196.

14. Goldberg Y. A primer on neural network models for natural language processing //Journal of Artificial Intelligence Research. – 2016. – Vol. 57. – P. 345-420.

15. Jozefowicz R. et al. Exploring the limits of language modeling //arXiv preprint arXiv:1602.02410. – 2016.

16. Manning C. et al. The Stanford CoreNLP natural language processing toolkit //Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations. – 2014. – C. 55-60.

17. Vinyals O. et al. Grammar as a foreign language //Advances in neural information processing systems. – 2015. – C. 2773-2781.

18. Cui P. et al. A survey on network embedding // IEEE Transactions on Knowledge and Data Engineering. – 2018.

19. Szabó Z. Compositionality. – Metaphysics Research Lab: Stanford University. – First published Thu Apr 8, 2004. – Substantive revision May 24, 2017.

20. Lin B. Y. et al. Multi-channel bilstm-crf model for emerging named entity recognition in social media // Proceedings of the 3rd Workshop on Noisy User-generated Text. – 2017. – P. 160-165.

21. Allahyari M. et al. A brief survey of text mining: Classification, clustering and extraction techniques // arXiv preprint arXiv:1707.02919. – 2017.

22. Akbik A., Blythe D., Vollgraf R. Contextual string embeddings for sequence labeling // Proceedings of the 27th International Conference on Computational Linguistics. – 2018. – P. 1638-1649.

23. Hershcovich D. et al. Syntactic Interchangeability in Word Embedding Models // arXiv preprint arXiv:1904.00669. – 2019.

24. Hardeniya N. et al. Natural Language Processing: Python and NLTK. – Packt Publishing Ltd. – 2016.

25. Burtsev M. et al. DeepPavlov: Open-Source Library for Dialogue Systems // Proceedings of ACL 2018, System Demonstrations. – 2018. – P. 122-127.

26. Козеренко Е. Б., Кузнецов К. И., Романов Д. А. Семантическая обработка неструктурированных текстовых данных на основе лингвистического процессора PullEnti // Информатика и её применения. – 2018. – Т. 12. – №. 3. – С. 91-98.

27. Russian language models for spaCy [electronic resource]. – URL: https://github.com/buriy/spacy-ru – Accessed 02.05.2019.

28. Горкун О. П. Подходы к извлечению объектов и фактов из неструктурированных текстов //Advanced Science. – 2019. – С. 70-72.

29. Jiang R., Banchs R. E., Li H. Evaluating and combining name entity recognition systems // Proceedings of the Sixth Named Entity Workshop. – 2016. – P. 21-27.

30. Ng V. Machine learning for entity coreference resolution: A retrospective look at two decades of research // Thirty-First AAAI Conference on Artificial Intelligence. – 2017. – P. 4877-4884.

31. Ng V. Entity Coreference Resolution // IEEE Intelligent Informatics Bulletin. – 2016. – Vol. 17. – №. 1. – P. 7-13.

32. Emani C. K., Cullot N., Nicolle C. Understandable big data: a survey // Computer science review. – 2015. – Vol. 17. – P. 70-81.

33. Ермакова Л. М. Методы извлечения информации из текста // Вестник Пермского Университета. Математика. Механика. Информатика. – Вып. 1(9). – 2012. – С. 77-84.

34. Hogenboom F. et al. A survey of event extraction methods from text for decision support systems // Decision Support Systems. – 2016. – Vol. 85. – P. 12-22.

35. Tomita-parser tool for extraction structured data from texts [electronic resource]. – URL: https://tech.yandex.ru/tomita/ – Accessed 02.05.2019.

36. Yargy – documentation [electronic resource]. – URL: https://yargy.readthedocs.io/ru/latest/ – Accessed 02.05.2019.

37. Chernyak E., Ilvovsky D. Lecture 4. Parsing [electronic resource]. – URL: http://wiki.cs.hse.ru/Lecture_4._Parsing – Accessed 02.05.2019.

38. Andor D. et al. Globally normalized transition-based neural networks // arXiv preprint arXiv:1603.06042. – 2016.

39. Rule-based named entity recognition library for russian language [electronic resource] – URL: https://github.com/natasha/natasha – Accessed 02.05.2019.

40. Poibeau, Thierry; Kosseim, Leila (2001). "Proper Name Extraction from Non-Journalistic Texts". Language and Computers. 37 (1): 144–157.

41. Straka M., Straková J., Hajic J. Prague at EPE 2017: The UDPipe System // EPE 2017. – 2017. – P. 65.

42. Stenetorp P. et al. BRAT: a web-based tool for NLP-assisted text annotation //Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics. – Association for Computational Linguistics, 2012. – P. 102-107.

43. Nivre J. et al. Universal Dependencies v1: A Multilingual Treebank Collection // LREC. – 2016.

44. Грановский Д.В., Бочаров В.В., Бичинева С.В. Открытый корпус: принципы работы и перспективы // Компьютерная лингвистика и развитие семантического поиска в Интернете: Труды научного семинара XIII Всероссийской объединенной конференции «Интернет и современное общество». Санкт-Петербург, 19–22 октября 2010 г. — СПб., 2010. — 94 с.

45. Национальный корпус русского языка [Электронный ресурс]. – URL: http://www.ruscorpora.ru – Дата обращения: 02.05.2019.

46. Chen D., Manning C. A fast and accurate dependency parser using neural networks // Proceedings of the 2014 Conference on Empirical Methods in Natural

Language Processing (EMNLP) – Association for Computational Linguistics. – Doha, Qatar. – 2014. – P. 740–750.

47. Zhang Y., Nivre J. Transition-based dependency parsing with rich non-local features. // Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers – Vol. 2, Stroudsburg, PA, USA, – 2011 – P. 188–193.

48. Korobov M: Morphological Analyzer and Generator for Russian and Ukrainian Languages // Analysis of Images, Social Networks and Texts. – 2015. – P 320-332.

49. Radishevskii V. L. et al. Distributed GLR-Parser for Natural Language Processing. Proceedings of the VIII International Conference "Distributed Computing and Grid-technologies in Science and Education" (GRID 2018), Dubna, Moscow region, Russia, September 10 – 14. – 2018.

50. Zeman D. et al. CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies //Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. – 2018. – P. 1-21.

51. ГОСТ 55090-2012. Ресурсосбережение. Обращение с отходами. Рекомендации по утилизации отходов бумаги // Электронный фонд правовой и нормативно-технической документации. [Электронный ресурс]. URL: http://docs.cntd.ru/document/1200103182. – Дата обращения: 02.05.2019 г.

52. ГОСТ 55102-2012. Ресурсосбережение. Обращение с отходами. Руководство по безопасному сбору, хранению, транспортированию и разборке отработавшего электротехнического и электронного оборудования, за исключением ртутьсодержащих устройств и приборов // Электронный фонд правовой и нормативно-технической документации. [Электронный ресурс]. URL: http://docs.cntd.ru/document/1200104723. – Дата обращения: 02.05.2019 г.

53. Постановление Правительства РФ от 03.09.2010 N 681 (ред. от 01.10.2013) "Об утверждении Правил обращения с отходами производства и потребления в части осветительных устройств, электрических ламп,

ненадлежащие сбор, накопление, использование, обезвреживание, транспортирование и размещение которых может повлечь причинение вреда жизни, здоровью граждан, вреда животным, растениям и окружающей среде // Государственная система правовой информации [Электронный ресурс]. URL: http://pravo.gov.ru/proxy/ips/?docbody=&nd=102141053 – Дата обращения: 02.05.2019 г.

54. СП 5.13130.2009 Системы противопожарной защиты. Установки пожарной сигнализации и пожаротушения автоматические. Нормы и правила проектирования (с Изменением N 1) // Электронный фонд правовой и нормативно-технической документации. [Электронный ресурс]. URL: http://docs.cntd.ru/document/1200071148. – Дата обращения: 02.05.2019 г.

55. СП 3.13130.2009 Системы противопожарной защиты. Система оповещения и управления эвакуацией людей при пожаре. Требования пожарной безопасности // Электронный фонд правовой и нормативно-технической документации. [Электронный ресурс]. URL: http://docs.cntd.ru/document/1200071145. – Дата обращения: 02.05.2019 г.

Список публикаций и основных научный достижений

1. Radishevskii V. L., Kulnevich A. D., Chugunov R. A., Shevchuk A. A. Distributed GLR-parser for Natural Language Processing // CEUR Workshop Proceedings. - 2018 - Vol. 2267. - p. 374-377.
2. Kulnevich A. D., Radishevsky V. L., Chugunov R. A., Shevchuk A. A. Application of russian named entity recognition and coreference resolution in the oil industry // CEUR Workshop Proceedings. - 2018 - Vol. 2267. - p. 378-382.
3. Именная стипендия Сбербанка "12UP" на 2018-2019 уч.г.
4. Стипендия Президента РФ по ПНР на 2018-2019 уч.г.
5. Повышенная государственная академическая стипендия III степени в номинации «За достижения в научно-исследовательской деятельно» в осеннем семестре 2017/2018 учебного года.
6. Повышенная государственная академическая стипендия III степени в номинации «За достижения в научно-исследовательской деятельно» в осеннем семестре 2018/2019 учебного года.
7. Повышенная государственная академическая стипендия III степени в номинации «За достижения в научно-исследовательской деятельно» в весеннем семестре 2017/2018 учебного года.
8. Диплом победителя в номинации "Лучшее приложение на основе открытых данных", команда "No_name". Второй региональный конкурс "Открытые данные Томской области", г. Томск, декабрь 2017.
9. Диплом победителя, команда "No_name". Хакатон Energy Hack, г. Новосибирск, май 2018.