

Федеральное государственное бюджетное учреждение науки  
Санкт-Петербургский институт информатики и автоматизации  
Российской академии наук (СПИИРАН)

На правах рукописи

Смирнов Сергей Владимирович

ТЕХНОЛОГИЯ И СИСТЕМА АВТОМАТИЧЕСКОЙ  
КОРРЕКТИРОВКИ РЕЗУЛЬТАТОВ ПРИ РАСПОЗНАВАНИИ  
АРХИВНЫХ ДОКУМЕНТОВ

Специальность 05.13.11 – Математическое и программное обеспечение  
вычислительных машин, комплексов и компьютерных сетей

Диссертация на соискание ученой степени  
кандидата технических наук

Научный руководитель:  
доктор технических наук  
Кулешов Сергей Викторович

Санкт-Петербург – 2015

## Оглавление

Оглавление .....	2
Введение .....	5
Глава 1. Аналитический обзор предметной области и постановка задачи исследования.....	12
1.1    Назначение системы распознавания архивных документов .....	12
1.2    Характеристики обрабатываемых документов .....	16
1.3    Варианты использования результатов распознавания.....	18
1.4    Обзор и сравнительный анализ OCR систем .....	19
1.4.1    OCR системы.....	20
1.4.2    Результаты сравнения.....	23
1.4.3    Выводы.....	24
1.5    Классификация ошибок распознавания.....	25
1.6    Методы корректировки ошибок в тексте .....	28
1.6.1    Методы корректировки орфографических ошибок .....	28
1.6.2    Методы корректировки ошибок распознавания.....	32
1.6.3    Вывод .....	36
1.7    Выводы по первой главе .....	37
Глава 2. Автоматическая корректировка ошибок оптического распознавания на основе рейтинго-ранговой модели текста .....	39
2.1    Описание используемых методов .....	39
2.1.1    Расстояние Левенштейна .....	39
2.1.2    Метод анаграмм .....	41
2.2    Общий алгоритм корректировки .....	46
2.3    Подготовка структур данных.....	47
2.3.1    Предварительная обработка.....	47
2.3.2    Структуры для отбора корректировок.....	48

2.3.3 Структуры для ранжирования корректировок.....	50
2.3.4 Структуры для обнаружения ошибок.....	51
2.4 Генерация корректировок .....	51
2.5 Ранжирование корректировок .....	53
2.6 Формирование результата .....	56
2.7 Выводы по второй главе.....	57
 Глава 3. Реализация системы распознавания архивных документов .....	59
3.1 Интеграция с электронным архивом.....	59
3.2 Технология распознавания и корректировки результатов .....	61
3.3 Архитектура и компонентная модель системы .....	65
3.4 ПК подготовки к работе .....	67
3.4.1 ПМ настройки профилей.....	67
3.4.2 ПМ сравнительного анализа .....	68
3.4.3 ПМ подготовки структур данных .....	69
3.5 ПК пакетного распознавания .....	75
3.5.1 ПМ управления заданиями .....	75
3.5.2 ПМ планирования выполнения .....	76
3.5.3 ПМ обработки задания .....	76
3.6 ПК распознавания изображения.....	77
3.6.1 ПМ предобработки .....	77
3.6.2 ПМ OCR .....	78
3.6.3 ПМ корректировки результатов .....	78
3.6.4 ПМ оценки качества .....	81
3.7 Инструментарий настройки и выбора наиболее эффективной конфигурации .....	82

3.8	Морфологический анализ .....	83
3.9	API системы.....	83
3.10	Выводы по третьей главе .....	84
Глава 4.	Апробация системы и оценка полученных результатов .....	86
4.1	Условия проведения испытаний.....	86
4.2	Описание корпуса документов .....	88
4.3	Порядок проведения испытаний .....	90
4.4	Критерии оценки .....	91
4.4.1	Базовые критерии оценки .....	91
4.4.2	Критерии оценки качества поиска .....	92
4.5	Выбор OCR систем .....	93
4.6	Распознавание всего корпуса документов.....	96
4.7	Оценка метода автоматической корректировки .....	98
4.7.1	Подготовка тестовых наборов изображений .....	99
4.7.2	Анализ результатов.....	99
4.8	Оценка результатов автоматической корректировки всего корпуса документов .....	101
4.9	Выводы по четвертой главе .....	104
Заключение .....	106	
Список сокращений и условных обозначений.....	108	
Словарь терминов.....	109	
Список литературы .....	110	
Приложение А. Примеры графического интерфейса системы .....	123	
Приложение Б. Свидетельства о государственной регистрации .....	125	
Приложение В. Акты внедрения.....	127	

## Введение

**Актуальность темы диссертации.** В наше время сохранение исторического наследия является актуальной задачей во всем мире, в стратегии развития информационного общества Российской Федерации одним из основных направлений является сохранение культурного наследия России и обеспечение его доступности для граждан [29].

Повсеместно запускаются проекты по массовой оцифровке фондов библиотек, музеев, архивов. Отличительными чертами данных проектов являются большие объемы обрабатываемой информации, достигающие размеров от сотен тысяч до миллионов документов за год, высокая стоимость работ, отсутствие временного ресурса на проведение полноценного контроля качества человеком и, как следствие, потребность в автоматизации всего цикла работ.

После перевода документов на бумажных носителях в электронный вид требуется обеспечить возможность оперативного поиска и навигации. Эффективность поисковых инструментов во многом зависит от результатов, полученных на выходе применяемой системы оптического распознавания символов (OCR — optical character recognition).

Достоверность результатов оптического распознавания сильно зависит от качества исходного изображения, лексикона, используемого при написании текста, особенностей шрифтов, наличия сторонних объектов, шумов и многих других факторов. Высокая точность достигается в случае распознавания изображений, где текст размещен на монотонно ровном фоне с хорошей контрастностью; словарь, используемый при написании текста, является современным и не содержит редких слов и словоформ; начертание букв и слов позволяет однозначно произвести сопоставление с шаблоном. К данной категории относятся все современные печатные документы.

Существующие коммерческие системы распознавания текста («Abbyy Finereader» [35], «Nuance OmniPage» [81] и др.), а также системы с открытыми исходными кодами («Cuneiiform» [47], «Tesseract» [108] и др.) достигают высокой точности результатов при обработке современных качественных печатных

документов. В случае же распознавания архивных документов, происхождение которых датируется десятками лет назад, количество допущенных ошибок в результатах распознавания значительно возрастает и эффективность применения средств автоматизации снижается. Результаты, получаемые на выходе систем распознавания необходимо подвергать последующей корректировке.

Методы автоматической корректировки ошибок распознавания во многом основываются на адаптации известных подходов корректировки орфографических ошибок, использующих скрытые Марковские модели, нейронные сети, n-граммы слов и символов, конечные автоматы. Также применяются методы, объединяющие результаты нескольких систем распознавания, использующие дополнительную информацию о контексте и эвристические алгоритмы. Большой вклад в теорию и практику корректировки ошибок в текстах внесли Philips L., Brill E., Kolak O., Mays E., Fossati D., Kukich K., Reynaert M. [43,53,73,74,77,91,98] и другие зарубежные ученые. Среди отечественных авторов в области автоматической обработки результатов оптического распознавания изображений можно выделить труды Арлазарова В.Л., Славина О.А., Шоломова Д.Л., Постникова В.В. [3,31-33,94] и других.

Во многих случаях существующие методы требуют привлечения ручного труда, предназначены для обработки современных текстов и не подходят в чистом виде для обработки архивных документов, отличающихся обилием узкоспециализированных терминов и значительным отличием в качестве результатов распознавания.

Решению описанных проблем и разработке системы распознавания архивных документов с применением методов автоматической корректировки и посвящена данная диссертационная работа.

**Объектом исследования** является распознавание изображений архивных документов с целью последующего поиска по ним.

**Предметом исследования** являются технология, методы и алгоритмы автоматической корректировки результатов распознавания архивных документов.

**Цель работы и задачи исследования.** Основной целью диссертационной работы является разработка технологии и системы автоматического обнаружения и корректировки допущенных ошибок при распознавании архивных документов различной тематики.

Для достижения поставленной цели в диссертационной работе поставлены и решены следующие задачи:

1. Сравнение качества результатов существующих систем оптического распознавания и классификация основных видов допускаемых ошибок при распознавании архивных документов.
2. Анализ существующих подходов к корректировке ошибок распознавания.
3. Разработка метода автоматического обнаружения и корректировки ошибок распознавания архивных документов, на основе пополняемых корпусных и тематических тезаурусов.
4. Проектирование, разработка и апробация системы распознавания документов архивного фонда, использующей алгоритмы и методы, предложенные в работе.

**Методы исследования.** Для решения поставленных задач в работе используются методы теории множеств, теории вероятности, статистического анализа, корпусной и компьютерной лингвистики. Реализация разработанных алгоритмов проводилась в соответствии с объектно-ориентированной методологией разработки программного обеспечения.

**Положения, выносимые на защиту.** На основе проведенных теоретических работ и их экспериментальной апробации на защиту выносятся следующие положения:

1. Метод автоматической корректировки ошибок распознавания текста на основе рейтинго-ранговой модели текста.
2. Алгоритм извлечения из результатов распознавания и тематических текстов корпусных тезаурусов и структур данных, применяемых при корректировке.

3. Правила ранжирования корректировок, основанные на частотных характеристиках и статистической вероятности сочетаемости с предшествующими словами.
4. Архитектура и компонентная модель разработанной системы распознавания, реализующей предложенный метод корректировки и предоставляющей эксперту инструментарий для настройки параметров обработки архивных документов различных тематических областей.

**Научная новизна** работы состоит в следующем:

1. Разработан метод автоматической корректировки ошибок распознавания текста на основе рейтинго-ранговой модели текста.
2. Разработаны правила ранжирования корректировок, основанные на предварительно проведенном статистическом n-грамм анализе корпуса результатов распознавания и тематических текстов.
3. Разработана система распознавания архивных документов с автоматической корректировкой результатов и инструментарием, позволяющим эксперту ограничивать пространство конфигураций для повышения качества распознавания.

**Обоснованность и достоверность** научных положений обеспечены аналитическим обзором исследований и разработок в данной области, подтверждаются положительными итогами практического использования результатов диссертации, а также апробацией основных научно-практических положений в печатных трудах и докладах на всероссийских и международных конференциях.

**Практическая ценность** работы заключается в создании программной системы, реализующей теоретические результаты работы, которая может использоваться в проектах массовой оцифровки и распознавания документов фондов государственных архивов, библиотек, музеев, судов, ЗАГС и других учреждений.

Разработанная в диссертационной работе технология и система автоматического распознавания позволяет значительно повысить скорость

обработки документов, сократить потребность трудоемкой дорогостоящей ручной работы и как следствие существенно расширить электронное информационное пространство, а также увеличить его доступность для граждан с минимальными временными и финансовыми затратами.

Предложенные в диссертационной работе подходы, методы и алгоритмы автоматического обнаружения и корректировки ошибок оптического распознавания позволяют значительно повысить качество конечных результатов.

**Реализация результатов работы.** Представленные в работе методы и алгоритмы были реализованы на языке программирования Java в виде программных модулей системы оптического распознавания текста и введены в промышленную эксплуатацию в составе государственной информационной системы «Государственные архивы Санкт-Петербурга» (государственный контракт №0172200006113000229\_146076 от 24.12.2013)

**Апробация результатов работы.** Основные положения и результаты диссертационной работы представлялись на конференциях: I Всероссийская электронная научно-практическая конференция-форум молодых ученых и специалистов «Современная российская наука глазами молодых исследователей - 2011»; IV Всероссийская научно-практическая конференция "Научное творчество XXI века" с международным участием (Красноярск, 2011); XVI Международная научно-практическая конференция «Перспективы развития информационных технологий» (Новосибирск, 2013); XXI Международная научно-практическая конференция «Перспективы развития информационных технологий» (Новосибирск, 2014); XIV Санкт-Петербургская международная конференция «Региональная информатика (РИ-2014)» (Санкт-Петербург, 2014); X Всероссийская научно-практическая конференция «Электронные ресурсы библиотек, музеев, архивов» (Санкт-Петербург, 2014); XVII Всероссийская объединенная научная конференция «Интернет и современное общество» (Санкт-Петербург, 2014).

Разработанное программное обеспечение было апробировано на документах фондов центральных государственных архивов Санкт-Петербурга в составе

государственной информационной системы «Государственные архивы Санкт-Петербурга», регистрационный номер системы в реестре государственных информационных систем Санкт-Петербурга №2053 от 11.03.2013г.

**Публикации.** Основные результаты по материалам диссертационной работы опубликованы в 13 печатных работах, среди них 6 работ в рецензируемых изданиях из перечня ВАК, получено 2 свидетельства о государственной регистрации программы для ЭВМ.

**Структура и объем работы.** Диссертационная работа включает введение, четыре главы, заключение, список использованных источников (115 наименований) и 3 приложения. Объем работы – 129 страниц машинописного текста, включая 34 рисунка и 16 таблиц.

**Во введении** обоснована важность и актуальность темы диссертации, сформулированы цели диссертационной работы и решаемые задачи, определяется научная новизна работы, а также ее практическая значимость. Приводится краткое содержание работы по главам.

**В первой главе** приводится аналитический обзор предметной области и существующих систем оптического распознавания, определяется степень их пригодности к распознаванию архивных документов, выявляется необходимость корректировки допускаемых ошибок распознавания, приводится классификация ошибок по видам и анализ существующих подходов к корректировке, уточняются требования к разрабатываемой системе.

**В второй главе** содержится описание используемых методов и разработанного метода автоматической корректировки ошибок распознавания на основе рейтинго-ранговой модели текста.

**В третьей главе** приводится описание архитектуры и программной реализации системы распознавания архивных документов, определяется порядок ее взаимодействия с системой электронного архива, описывается технология распознавания и корректировки результатов, предоставляется информация об инструментарии для настройки параметров обработки архивных документов различных тематических областей.

**В четвертой главе** даются сведения об условиях и порядке проведения испытаний разработанной системы распознавания архивных документов, приводится описание экспериментального корпуса документов, критериев оценки качества распознавания. Представлены результаты экспериментальной оценки предложенного метода корректировки и результаты автоматической корректировки всего корпуса документов.

**В заключении** подводятся итоги работы, приводятся основные результаты исследований и пути дальнейшего развития научных исследований.

# Глава 1. Аналитический обзор предметной области и постановка задачи исследования

## 1.1 Назначение системы распознавания архивных документов

Сфера деятельности государственных архивов включает в себя широкий спектр задач, связанных с комплектованием, учетом, использованием и обеспечением сохранности документов. На рисунке 1.1 представлен типовой набор рабочих процессов, протекающих в архиве.

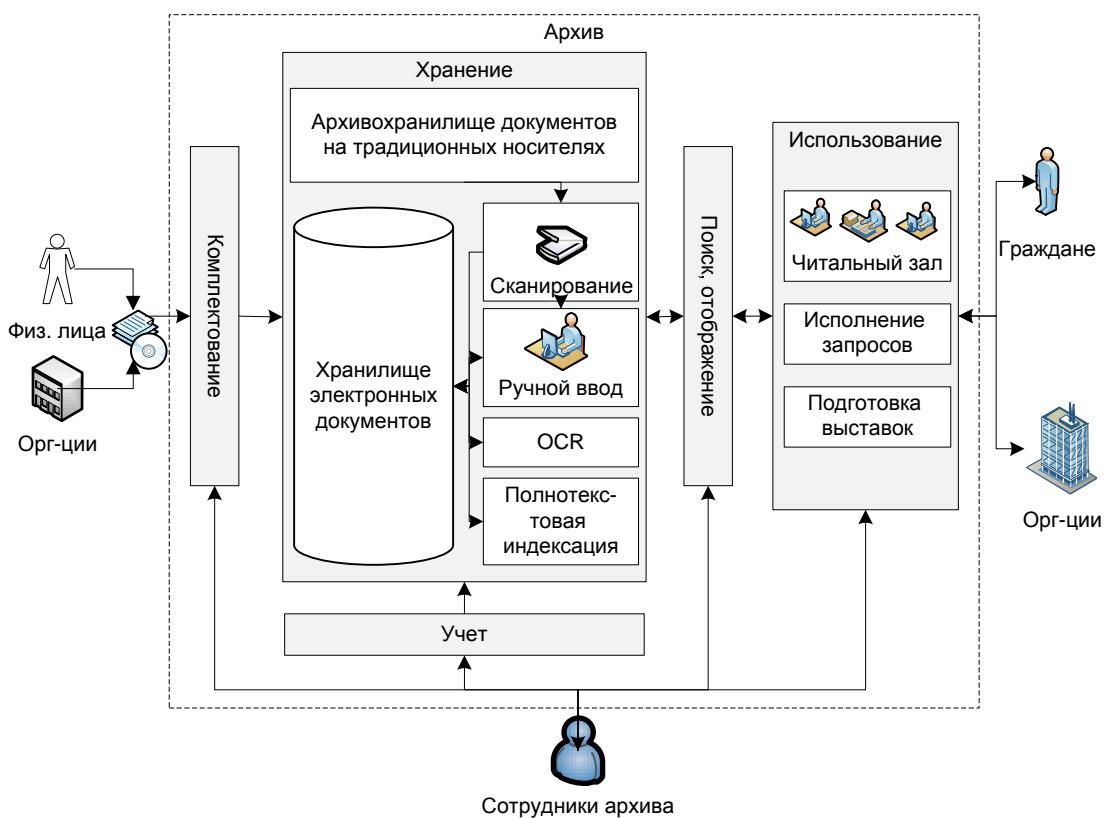


Рисунок 1.1. Общая схема рабочих процессов архива

Эффективность выполнения каждой задачи имеет сильную зависимость от скорости нахождения и получения доступа к нужным документам. Поиск документов является своего рода «узким» местом во всех рабочих процессах и накладывает серьезные ограничения на время выполнения ежедневных задач архива.

На данный момент в информационных системах центральных государственных архивов Санкт-Петербурга, поиск производится лишь по документам, обладающим текстовым описанием. Текстовое описание вручную

заносится в систему операторами и сотрудниками архива в процессе составления научно-справочного аппарата и оцифровки бумажных документов.

Данный подход к наполнению и построению поискового механизма обладает рядом существенных ограничений:

1. Малое покрытие — лишь малая часть документов попадает в поисковый индекс и как следствие остается недоступной для автоматического поиска и скрытой от конечного пользователя.
2. Низкая скорость наполнения поисковой базы — ручной ввод данных не может обеспечить должной скорости роста поисковой базы. В условиях постоянного пополнения базы данных отсканированными образами документов, разрыв между количеством отсканированных документов и количеством документов, включенных в поисковый индекс, экспоненциально возрастает.

Очевидно, что для снижения влияния данных ограничений необходимо автоматизировать процессы пополнения поисковой базы и развивать поисковые механизмы, использующиеся в архивах.

В работе предлагается решение, предоставляющее пользователям архива возможность оперативного поиска по содержимому электронных образов документов без необходимости предварительного ручного ввода поисковых метаданных.

Предлагаемое решение представляет собой программный комплекс, состоящий из трех подсистем:

1. подсистема распознавания и корректировки ошибок;
2. подсистема полнотекстовой индексации результатов распознавания;
3. подсистема поиска по распознанным изображениям документов.

На рисунке 1.2 изображена схема взаимодействия подсистем, на примере процесса обработки и поиска по электронному образу документа.

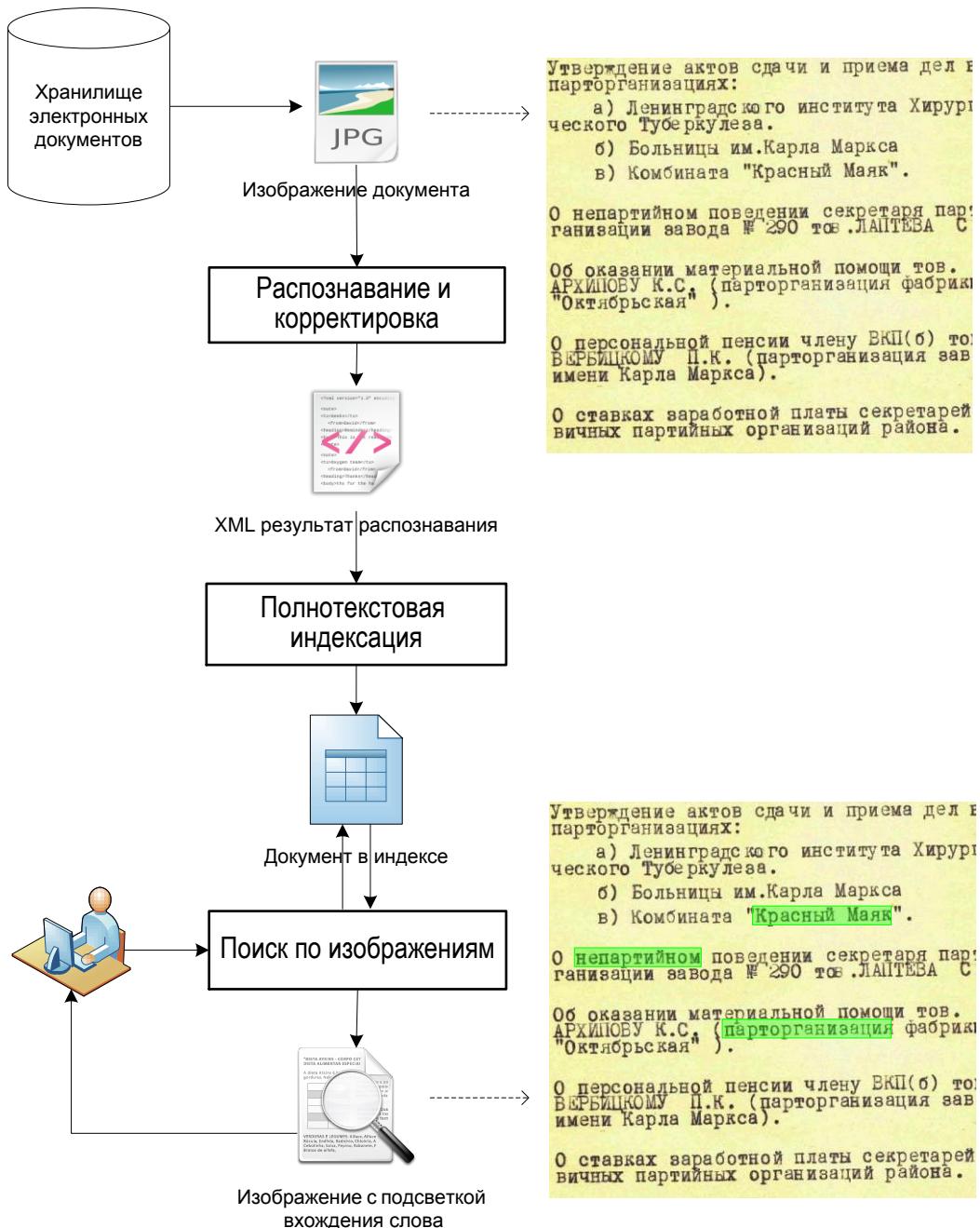


Рисунок 1.2. Процесс обработки и поиска изображений

Рассмотрим последовательность шагов данного процесса:

1. Изображение выбирается из хранилища электронных документов.
2. Изображение передается на вход подсистемы распознавания и корректировки.
3. В результате оптического распознавания формируется xml документ, содержащий распознанный текст, с указанием координат расположения слов и набором возможных вариантов написания (в тех случаях, когда однозначное соответствие установить не удалось).

4. Далее производится полнотекстовое индексирование результата распознавания, на выходе которого формируется ряд индексных документов для помещения в индексное хранилище. Индексируется каждый вариант написания слова с учетом особенностей морфологии русского языка. В качестве системы полнотекстовой индексации и поиска используется библиотека Apache Lucene [41], реализованная на языке программирования Java [67].
5. Изображение готово к поиску.
6. Пользователь вводит поисковую фразу и передает команду подсистеме поиска по изображениям.
7. Поисковая фраза проходит анализ и из индекса выбираются документы, удовлетворяющие критериям поиска.
8. На исходном изображении документа цветом выделяются искомые слова, и результаты отображаются пользователю.

Ключевым элементом в предложенном программном комплексе является подсистема распознавания и корректировки ошибок, разработке которой и посвящена данная диссертационная работа.

Отличительными особенностями массового распознавания архивных документов являются [25]:

- сверхбольшие объемы обрабатываемых документов;
- разбиение всего объема документов на большие тематические группы, обладающие общими свойствами;
- высокие требования к пропускной способности системы;
- отсутствие практической возможности проведения ручной верификации и корректировки всех результатов распознавания;
- важность проведения автоматической оценки и контроля качества результатов распознавания.

При разработке системы следует учитывать ряд особенностей внедрения и эксплуатации в государственных архивах, обусловленных отсутствием

достаточного количества времени и ресурсов у сотрудников архивов для настройки и администрирования:

1. Отсутствие времени и ресурсов на ручное распознавание и ручную корректировку результатов распознавания
2. Отсутствие времени и ресурсов на ручной отбор и поиск документов, пригодных для распознавания.
3. Отсутствие времени и ресурсов на постановку в очередь на обработку документов, пригодных к распознаванию.
4. Отсутствие времени и ресурсов на ручной контроль качества распознавания каждого документа.
5. Отсутствие времени и ресурсов на ручное обучение.

Особое внимание на этапах проектирования и разработки системы массового распознавания следует обратить на следующие проблемные области [39]:

- характеристики обрабатываемых документов;
- варианты использования результатов распознавания;
- выбор OCR систем;
- корректировка ошибок распознавания;
- оценка качества распознавания.

## 1.2 Характеристики обрабатываемых документов

Документы государственных архивов Санкт-Петербурга, подлежащие обработке в рамках данной диссертационной работы, подразделяются на дела (единицы хранения) и научно-справочный аппарат (НСА): описи, указатели, картотеки, каталоги, путеводители. НСА содержит в себе полную информацию обо всех хранящихся в архиве документах в сжатой компактной форме и является основным поисковым инструментом по фондам архива [27].

При внедрении систем автоматического распознавания текста, в первую очередь следует обрабатывать именно документы НСА. Текст документов НСА

является более однородным по виду написания (рукописный или машинописные), типу шрифта и структуре расположения, чем текст оригиналов единиц хранения.

Все машинописные документы НСА по своему качеству можно разделить на четыре категории:

1. Документы, напечатанные на печатной машинке низкого качества. Текст таких документов характеризуется расплывчатыми очертаниями, блеклыми чернилами,искаженными углами наклона, наличием большого количества ручных исправлений и второстепенных помарок и трудно воспринимается даже человеческим глазом. Пример изображения проиллюстрирован на рисунке 1.3.

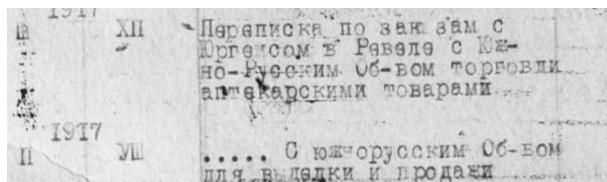


Рисунок 1.3. Печатная машинка, низкое качество

2. Документы, напечатанные на печатной машинке, среднего качества – более ровное расположение строк, более четкие очертания и контрастность, но с нарушениями в междустрочных и межбуквенных пространствах. Пример изображения проиллюстрирован на рисунке 1.4.

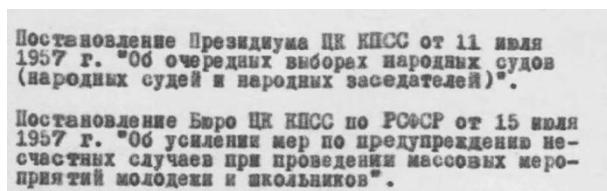


Рисунок 1.4. Печатная машинка, среднее качество

3. Документы, напечатанные на печатной машинке, высокого качества. Пример изображения проиллюстрирован на рисунке 1.5.

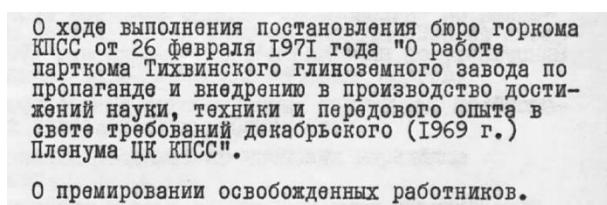


Рисунок 1.5. Печатная машинка, высокое качество

4. Документы, напечатанные на принтере, очень высокого качества. Пример изображения проиллюстрирован на рисунке 1.6.

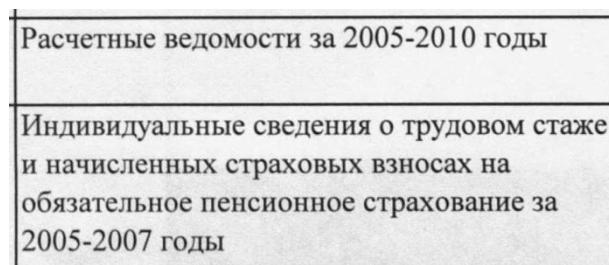


Рисунок 1.6. Принтер, очень высокое качество

### 1.3 Варианты использования результатов распознавания

Существует множество вариантов использования результатов распознавания, и они далеко не ограничиваются созданием лишь полностью идентичной копии оригинала документа. Результаты распознавания могут использоваться для решения следующих задач [107]:

- Полнотекстовое индексирование — результат распознавания рассматривается как простой текст и в дальнейшем подается на вход поисковой системы. Текст используется как основа для полнотекстового поиска. Причем, конечному пользователю в результате поиска отображается найденный образ документа без обозначения вхождения поисковой фразы.

Данный вид не требует точности распознавания и одновременно предоставляет хорошие поисковые возможности.

- Отображение с подсветкой результатов на образе — в данном режиме распознанный текст обрабатывается также как и в предыдущем случае, а отличие заключается в подсистеме отображения поисковых результатов. В результатах поиска пользователю предоставляется изображение с выделенными фрагментами вхождений поисковой фразы. Очевидно, что в данном случае требования к качеству распознавания возрастают, но одновременно с этим увеличивается и эффективность поисковой системы в отличие от предшествующего способа отображения результатов.
- Выдача результатов в виде неразмеченного текста — поисковым результатом является непосредственно текст, полученный в результате распознавания, а оригинальное изображение документа не отображается. Если распознанные слова будут сильно искажены, то пользователь не

сможет получить искомой информации, и потеряет доверие к системе. Таким образом, точность должна быть очень высокой, что практически не может быть достигнуто без привлечения человеческого труда, и, как следствие, ведет к значительным времененным и финансовым затратам.

- Воссоздание оригинального документа — отображение результатов распознавания редко производится без форматирования и разметки текста, с целью сохранения исходной структуры и деталей расположения элементов. В дополнение, размеченный xml документ может содержать дополнительные атрибуты, тэги или ссылки на родственные документы.

В рамках данной диссертационной работы результаты распознавания планируется использовать лишь на промежуточном этапе полнотекстового индексирования. Пользователю поисковый результат будет предоставляться в виде подсвеченных областей на изображении.

Выбранный вариант использования результатов распознавания снижает требования к OCR системам в части качества проведения структурного анализа документа [17], что существенно увеличивает круг систем подходящих под задачи исследования. Обязательными требованиями являются лишь способность обрабатывать русскоязычные тексты и наличие в результатах распознавания «x,y» координат найденных слов.

#### **1.4 Обзор и сравнительный анализ OCR систем**

Самостоятельная разработка OCR систем представляет собой довольно сложную научную и техническую задачу и не может являться обоснованной для большинства проектов по оцифровке. Особенно при условии того, что на рынке присутствует порядка десятка различных OCR систем, отличающихся условиями распространения, стоимостью, предоставляемыми функциями и, разумеется, качеством генерируемых результатов.

Наиболее актуальной задачей становится выбор подходящей для конкретного проекта OCR системы. Самым надежным подтверждением правильности выбора является проведение сравнительного анализа результатов

распознавания. При проведении сравнения необходимо опираться на показатели, которые наиболее полно отвечают будущим целям использования полученных результатов распознавания.

Сравнительный анализ и выбор OCR систем будет производиться в контексте решения задачи распознавания русскоязычных документов архивного фонда, за период с 1917 года по настоящее время [27].

#### **1.4.1 OCR системы**

Современные системы оптического распознавания можно разделить на коммерческие и свободно распространяемые системы с открытыми исходными кодами.

По своей архитектуре системы подразделяются на приложения для персонального использования, серверные решения для проектов массовой обработки документов и онлайн сервисы распознавания образов.

Онлайн сервисам трудно удовлетворять требованиям крупных проектов по оцифровке архивных документов из-за ограничений по максимальному количеству сеансов распознавания, пропускной способности каналов связи, а также обеспечения конфиденциальности передаваемой информации. К тому же данные сервисы строятся поверх существующих движков распознавания и, как следствие, не представляют самостоятельного интереса для участия в сравнительном анализе.

В контексте задач массовой оцифровки интерес представляют как коммерческие системы по причине своего заявленного высокого качества, так и открытые системы по причине своей доступности и гибкости в настройке. Поскольку целью данной работы является обработка русскоязычных документов для последующего поиска с подсветкой вхождения поисковых фраз, то интерес представляют системы с поддержкой распознавания русского языка, а также выдающие информацию о координатах расположения распознанных слов/символов на изображении.

В таблицах 1.1 и 1.2 представлен перечень наиболее популярных OCR систем с отметками интересующих характеристик.

Таблица 1.1. Коммерческие OCR системы

<b>Наименование системы</b>	<b>Поддержка русского языка</b>	<b>Информация о координатах</b>	<b>Участие в сравнении</b>
ABBYY FineReader [35]	да	да	да
AnyDoc Software [40]	нет сведений	нет сведений	нет
Cvision ocr [49]	да	нет	нет
Dynamsoft OCR SDK (Tesseract engine) [51]	да	да	нет
ExperVision TypeReader & RTK [52]	нет	нет	нет
IRIS Readiris [65]	да	да	да
LEADTOOLS OCR SDK [75]	да	да	нет
Nuance OmniPage [81]	да	да	да
Transym OCR [111]	нет сведений	нет сведений	нет

Таблица 1.2. OCR системы с открытым исходным кодом

<b>Наименование системы</b>	<b>Поддержка русского языка</b>	<b>Информация о координатах</b>	<b>Участие в сравнении</b>
Clara OCR [45]	нет	нет	нет
Cuneiform Linux [47]	да	да	да
Cuneiform Windows [48]	да	нет	да
GOCR [58]	нет	нет	нет
Javaocr [68]	нет сведений	нет	нет
LOCR [76]	нет	нет	нет
Ocrad [82]	нет	нет	нет
OCRchie [83]	нет	нет	нет
Ocre [84]	нет	нет	нет
OCRFeeder [85]	да	нет	нет
Ocropus [86]	нет	да	нет
SimpleOCR [103]	нет	нет	нет
Tesseract [108]	да	да	да

Для участия в сравнении выберем несколько признанных лидеров из коммерческих систем (“Abbyy Finereader”, “Nuance Omnipage”), одну менее популярную коммерческую систему (“IRIS Readiris ”) и все свободно распространяемые системы, поддерживающие распознавание русского языка (“Cuneiform Linux”, “Cuneiform Windows”, “Tesseract”).

**“ABBYY FineReader”** — система оптического распознавания символов, разработанная российской компанией ABBYY. Является признанным лидером на рынке. Распространяется на коммерческой основе.

Система позволяет извлекать текстовую информацию из цифровых изображений (фотографий, результатов сканирования, PDF-файлов), распознает около двух сотен языков, в том числе, русский, и предоставляет результаты распознавания в разнообразных форматах, включая xml формат с информацией о координатах распознанного текста.

В сравнении принимает участие версия “Abbyy Finereader 11”.

**“IRIS Readiris”** — коммерческая система оптического распознавания символов, также как и “Abbyy Finereader” представлена во всех видах от персонального приложения до инструментария разработчика.

Распознает более 130 языков, включая русский, принимает файлы и сохраняет результаты во всех возможных форматах.

В сравнении принимает участие версия “IRIS Readiris 14”.

**“Nuance OmniPage”** — коммерческая система оптического распознавания символов, представлена во всех видах от персонального приложения до инструментария разработчика.

Поддерживает распознавание более 120 различных языков, включая русский, принимает файлы и сохраняет результаты во всех возможных форматах, в том числе в собственном xml формате с указанием координат.

В сравнении принимает участие версия “Nuance OmniPage 19”.

**“CuneiForm”** — свободно распространяемая открытая система оптического распознавания текстов российской компании Cognitive Technologies [46].

В Windows версии информация о координатах распознанного текста может быть получена только из бинарного формата вывода, который может быть прочитан только самой программой.

В Linux версии системы результаты распознавания могут быть сохранены с координатами каждого распознанного символа в формате hocr [109].

В сравнении принимают участие версии “Cuneiform Windows 12” “Cuneiform Linux 1.1.0”.

**“Tesseract”** — свободно распространяемая программа для распознавания текстов.

Результаты распознавания могут быть сохранены в формате hocr с указанием координат слов. В сравнении принимает участие версия “Tesseract 3.02.02”, собранная из исходных кодов на ОС Linux.

#### 1.4.2 Результаты сравнения

Минимальные и максимальные показатели точности распознавания на уровне слов отображены в таблице 1.3, диаграмма сравнения по данному критерию отображена на рисунке 1.7. Более детальные результаты сравнения приводятся в главе 4.5.

Качество распознавания напрямую зависит от качества исходных изображений, каждая система улучшала свои результаты последовательно от одного набора изображений к другому.

“Abbyy Finereader” достигает максимальных показателей на всех наборах данных и является бесспорным лидером.

“Cuneiform Linux” наоборот показывает наихудшие результаты распознавания. Применение данной системы в промышленных масштабах не целесообразно.

Остальные системы занимают промежуточную позицию с незначительными отклонениями относительно друг друга. Стоит выделить, что на наборе «ПМ-2» лучший результат достигает система “Nuance Omnipage”, а на наборе «ПМ-3» лучшие результаты показывает “Tesseract”. Система “IRIS Readiris” занимает

последнюю позицию на наборе «ПМ-2». Качество результатов распознавания системой “Cuneiform Windows” является средним на всех наборах данных.

Таблица 1.3. Точность распознавания

<b>Качество документов (источник), набор</b>	<b>Максимум</b>	<b>Минимум</b>
среднее ( <i>печатная машинка</i> ), ПМ-2	76,80%	22,01%
высокое ( <i>печатная машинка</i> ), ПМ-3	90,55%	30,28%
очень высокое ( <i>принтер</i> ), ПР-4	99,25%	90,20%

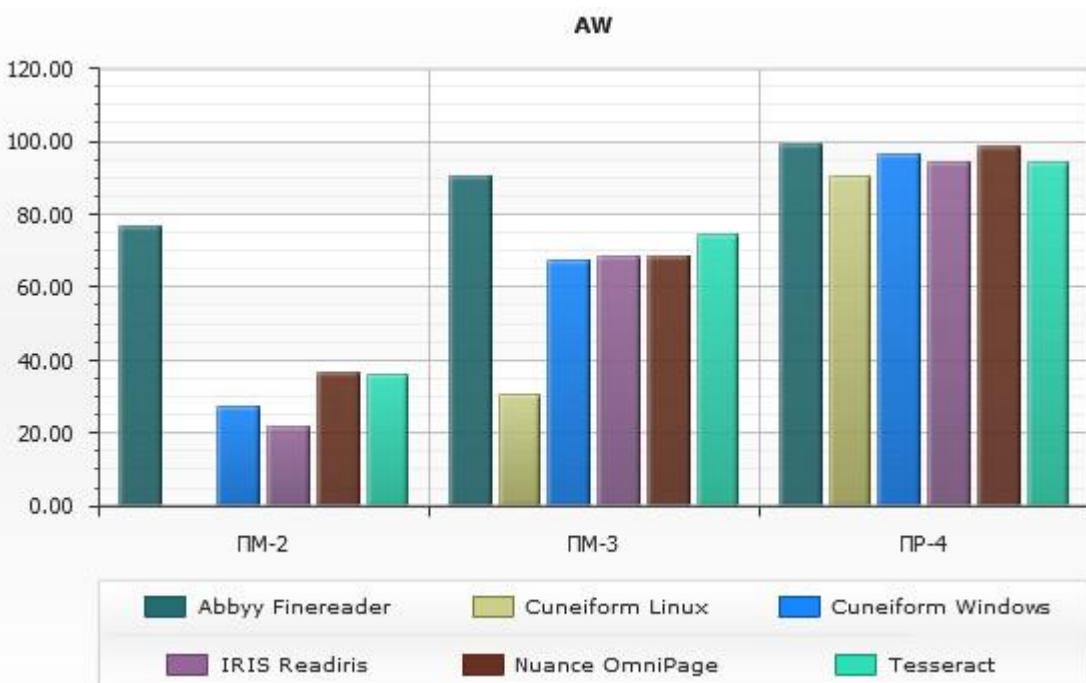


Рисунок 1.7. Диаграмма по критерию «Точность в словах»

#### 1.4.3 Выводы

1) Если бюджет проекта позволяет использовать “Abbyy Finereader”, то, несомненно, выбор следует остановить на этой системе.

Для задач построение поисковой системы по изображениям выбирать другие рассмотренные коммерческие системы не целесообразно с экономической точки зрения, так как показатели свободно распространяемых систем ничуть не отстают.

Если бюджет проекта ограничен, но требуется охватить все виды изображений, то возможно применение дифференцированного подхода, при котором каждому виду изображений будет соответствовать своя система распознавания. Такой подход накладывает дополнительные затраты на классификацию документов по видам.

2) Система ‘Tesseract’ из всех рассмотренных свободно распространяемых систем единственная предоставляет информацию о координатах слов на изображении и показывает высокое качество, относительно других систем, за исключением “Abbyy Finereader”. Таким образом, система “Tesseract” является хорошим выбором для задач построения поисковой системы по изображениям и будет использоваться как основная в данной работе.

3) Сравнительный анализ выявил наличие ошибок в результатах распознавания архивных документов среди всех систем оптического распознавания на всех наборах данных, что указывает на необходимость разработки и применения средств автоматической корректировки ошибок.

## **1.5 Классификация ошибок распознавания**

Качество процесса корректировки во многом зависит от точности нахождения ошибок и их верной классификации. Все типы ошибок можно разделить на две категории: словарные и несловесные ошибки [74].

*Несловесные ошибки* – это ошибки, приводящие к словам, которые не встречаются ни в одном из словарей. Например, «книго» вместо «книга». Большинство средств проверки орфографии нацелены на исправление именно данного вида ошибок.

*Словарные ошибки* – это ошибки, в результате которых формируются существующие слова с правильным написанием, но с некорректным применением. Ошибки данной категории могут быть исправлены только с использованием знаний о контексте.

Приведенная классификация является достаточно грубой и ее определенно не достаточно для покрытия всех видов ошибок, встречающихся в результатах OCR. Под вопросом остаются корректные слова, которые не встречаются в использующихся словарях, к таким словам могут относиться устаревшие термины, слова с историческими правилами написания, имена собственные и другие узкотематические и специфичные для конкретной предметной области слова.

Приведем более детальную классификацию ошибок, относящуюся, непосредственно, к системам оптического распознавания текста [71]:

- *Некорректная сегментация*

Недетерминированное расстояние между строками, словами или символами приводят к некорректному распознаванию пробелов, приводящих к ошибкам сегментации.

Большинство методов обнаружения и корректировки ошибок определяют границы слов по пробельным символам (пробел, табуляция, отступы и прочие), что зачастую приводит к ошибочному выделению слов. Системам оптического распознавания более свойственно именно некорректное разделение слов на несколько [70], в то время как для текстов, набранных человеком, более типично объединение нескольких слов в одно [74].

- *Ошибочное определение переносов слов*

В случае, когда слово ввиду своей длины не помещается на одной строке, часть его переносится на новую строку. Данное разделение приводит также к увеличению ошибок сегментации.

- *Некорректное распознавание символов*

Шумы, изменчивость начертания символов и нестандартные шрифты приводят к неточному распознаванию символов, что в свою очередь приводит к формированию ошибочных слов.

- *Замена символов*

Типичными случаями замены правильных символов на некорректные являются: замена цифр на буквы, ошибочная подстановка одних букв вместо других.

- *Вставка и удаление символов*

Вставка и удаление символов менее типично для OCR систем, чем замена символов, но в случае плохо различимого начертания символов может происходить разбиение одного символа на несколько или слияние нескольких различных символов в один.

- *Ошибки пунктуации*

Плохое качество сканирования приводит к появлению шума, который зачастую воспринимается как символы пунктуации, такие как точки, запятые, многоточия и другие.

- *Некорректное определение регистра*

Из-за вариабельности шрифтов зачастую происходит некорректное определение регистра символов.

- *Ошибки, изменяющие смысл слова*

В некоторых случаях ошибочное определение символов может приводить к формированию существующих слов, но некорректных в данном контексте.

В таблице 1.4 приведены распространенные примеры ошибок в результатах распознавания.

Таблица 1.4 . Примеры OCR ошибок

<b>Вид ошибки</b>	<b>Ошибка распознавания</b>	<b>Правильный вариант</b>
Некорректная сегментация	слово форма	словоформа
Некорректная сегментация	ин сти тут	институт
Вставка символа переноса	ли- тература	литература
Некорректное распознавание символа	избранных	избранных
Подстановка цифры вместо символа	06ъ	Объ
Вставка лишнего символа	проток'ол	протокол
Изменение смысла слова	Ситников	Сотников
Некорректное определение регистра	БыстРоВ	Быстров
Пунктуация	план.мероприятий	план мероприятий
Разрушение	гъЛо! пнтро_в ъ	г.Ленинграде

Наиболее трудны для корректировки ошибки, связанные с полным разрушением исходного слова. По большому количеству ошибок данного вида

можно судить о чрезмерно плохом качестве исходных изображений, а корректировка такого рода ошибок возможна только за счет улучшения изображений и применения методов оптимизации качества на этапах предобработки.

## **1.6 Методы корректировки ошибок в тексте**

В последнее время задача корректировки ошибок правописания и распознавания вызывает в научном мире все больший интерес. Появляется много публикаций с описанием различных подходов, методов и алгоритмов. В данном разделе приводится обзор основных работ в данной области.

Вначале будут рассмотрены методы корректировки орфографических ошибок. Далее будет описан ряд работ использующих методы корректировки ошибок оптического распознавания.

### **1.6.1 Методы корректировки орфографических ошибок**

Корректировка ошибок распознавания во многом схожа с корректировкой орфографических ошибок. Поскольку тема корректировки орфографических ошибок является намного более глубоко исследованной, то вначале стоит обратиться к классификации ее методов и рассмотреть основные направления работ.

#### **Корректировка несловесных ошибок**

Несловесная ошибка выражается в полученной символьной последовательности, отсутствующей в известных системе словарях, таким образом, задачей корректировки становится нахождение наиболее подходящего (схожего) слова корректировки из словаря. Схожесть слов может определяться различными способами:

1. Расстояние Левенштейна [18]: минимальное число операций вставки, удаления и замены символов, которое необходимо произвести для того чтобы преобразовать одну строку в другую. Более подробно алгоритм вычисление расстояния Левенштейна описан в главе 2.

2. Расстояние Дамерау-Левенштейна [50]: в добавок к операциям вставки, удаления и замены добавляется операция перестановки двух соседних символов. В контексте корректировки OCR ошибок данное расширение не представляет большого значения, так как ошибка в перестановке символов не типична для систем оптического распознавания.
3. N-граммы на уровне символов: символьная n-грамма представляет собой последовательность из n символов. Отношение количества n-грамм, которые содержаться в обоих словах, и уникального количества всех n-грамм, может быть использовано в качестве меры определения схожести слов. Метод корректировки, основанный на n-граммах, применяется в работе [37].

Поиск корректировок в словарях путем перебора всех слов с вычислением выбранной меры схожести является очень трудоемкой операцией, сильно замедляющей работу системы. Увеличение скорости выборки достигается через выборку слов по его ключу (хэшу). Поиск осуществляется путем вычисления хэша ошибочного слова и поиска слов в словаре с таким же значением хэша.

Самыми распространенными методами являются методы SOUNDDEX [100] и Double Metaphone [91], они оба определяют схожие слова на основе их произношения и подходят для корректировки ошибок допущенных исключительно в распознавании текстов напечатанных человеком.

Другой метод использует структуру слова в процессе вычисления хэша [92]. Хэш строится из наиболее важных букв, являющихся основополагающими в формировании слова. В зависимости от корпуса текста от 56% до 77% всех ошибок могут быть откорректированы, используя описанную методику в купе с другими методами.

Еще одним подходом к корректировке является обучение на допущенных ошибках и дальнейшая тренировка системы для подбора наиболее релевантных заместителей. Главным требованием является наличие большого количества подверженных ошибкам слов и соответствующих им корректных словоформ. Обучение производится на основе нейронных сетей [10] или других

вероятностных методов. В работе [43] предложен подход, называемый улучшенной моделью “noisy channel”, суть которого заключается в том, что система предварительно собирает информацию о вероятностях замены символьных n-грамм. Вероятности дальше используются для формирования списка корректировок по словарю. Для выборки наиболее подходящей корректировки применяется языковая модель, учитывающая контекст. На наборе размером 10000 слов, 80% которых использованы для обучения и 20% для испытаний, достигается снижение количества ошибок на 74%. Данный подход представляет интерес при корректировке OCR ошибок, но требует достаточного количества предварительно сформированных эталонных текстов для обучения. Поскольку в текущей работе отсутствует возможность формирования эталонных текстов, данный алгоритм не может быть применен.

### **Корректировка словарных ошибок**

Методы корректировки словарных ошибок основываются на анализе контекста, окружающего потенциально ошибочное слово. В зависимости от метода в качестве контекста может выступать синтаксическая структура предложения, часть речи анализируемого слова или семантика предложения, текста, предметной области.

Важным методом является корректировка ошибок на основе n-грамм модели на уровне слов. N-грамма на уровне слов представляет собой последовательность из n слов, а модель n-грамм содержит информацию о частоте повторения каждой отдельной n-граммы в тексте.

Простой алгоритм, использующий триграммы слов [77], основывается на расчете вероятности целого предложения, путем разбиения его на триграммы и вычислении их суммарной частоты. Затем слова в предложении заменяются кандидатами, и вероятность предложения вычисляется вновь. В конечном итоге вариант с наибольшей вероятностью считается корректным.

Описанный подход производит чрезмерно большое количество замен слов, что влияет на скорость обработки. В дополнение, построение модели n-грамм на уровне слов требует большого объема исходного текста, чтобы избежать проблем,

связанных с разреженностью данных. Если триграммы отсутствуют в модели или их частота недостаточна высокая, то ошибочные корректировки будут иметь место. Одним из способов исправления данной ситуации является использование набора триграмм Google Web 1T [59], насчитывающего более миллиарда триграмм. Данный метод вначале генерирует набор кандидатов, используя частотные показатели триграмм, а затем выбирает наилучший вариант, оценивая сходство слов методом вычисления наибольшей общей последовательности.

Преодолеть проблему разреженности данных в работе [53] попытались путем построения модели смешанных триграмм, состоящих из слов и/или частей речи. Для каждого слова в предложении определяется часть речи, далее составляются смешанные триграммы, например («слово», глагол, существительное). Из всего набора корректировок для замены ошибочного слова выбирается корректировка, которая лучше укладывается в грамматическую модель предложения, предварительно построенную на основе вероятности появления смешанных триграмм. Подбор списка корректировок осуществляется по словарю с учетом следующих показателей: расстояние Левенштейна, длина слова, фонетическая схожесть SOUNDEX или Double Metaphone. Построение грамматической модели триграмм по результатам распознавания исторических текстов является довольно трудной задачей из-за большого количества ошибочных последовательностей. Поэтому в диссертационной работе будет использоваться модель n-грамм только на уровне слов.

Контекстная корректировка ошибок может также использовать алгоритмы определения смысла слов [54]. В рамках данного алгоритма, определяются слова, которые не подходят по смыслу и формируется список наиболее подходящих замен.

Еще одним важным аспектом данной категории методов является определение тематики корректируемого текста, что позволяет в дальнейшем ранжировать слова заместители схожей тематике выше, чем слова заместители сторонней тематики [115]. Применение данного подхода требует наличия адекватного набора тематических словарей. Поскольку специфические термины

не содержатся в обычных словарях, необходимым становится применение дополнительных словарей. Существует ряд работ, в которых данные словари формируются динамически на основе анализа текстов с различных источников интернета [105].

Для достижения наилучших результатов в работе [102] был применен комбинированный подход, использующий алгоритм Soundex на фонетическом уровне, модель “noisy channel” на уровне символов, биграмм модель на уровне слов, грамматическую модель на синтаксическом уровне и модель совместного вхождения слов на смысловом уровне. На каждом уровне формируется вектор из слов корректировок, на основе которых в дальнейшем происходит формирование финального списка. Данный подход позволяет корректировать как несловесные, так и словарные ошибки.

Более подробный обзор работ, связанных с корректировкой орфографических ошибок, изложен в работе [74].

### **1.6.2 Методы корректировки ошибок распознавания**

Все методы корректировки OCR ошибок можно разделить на две группы:

1. Корректировка на основе сравнения результатов нескольких OCR систем.

В работе [113] результат одной из систем принимается за основной, а результат вспомогательной системы используется для корректировки ошибок. Трудность реализации алгоритма объединения результатов заключается в сопоставлении одних результатов другим, обусловленная различиями результатов сегментации документа. Данный метод дает положительный результат в случае схожего уровня качества распознавания каждой системы. Точность результатов распознавания с использованием данного метода корректировки достигает максимума в 99,49% для книг второй половины 20 века и минимума в 94,38% для книг конца 19 века.

2. Корректировка, использующая результаты распознавания одной OCR системы.

Методы данной группы будут рассмотрены применяться в текущей работе, и рассмотрены далее.

При разработке программ корректировки результатов оптического распознавания необходимо учитывать тот факт, что несловесные ошибки преобладают над словарными ошибками [74]. Таким образом, большинство методов корректировки орфографических несловесных ошибок подходят и для корректировки OCR ошибок.

Как описано в предыдущем разделе, модель “noisy channel” очень широко используется при корректировке несловесных ошибок. Данную модель можно построить, проведя обучение системы на базе текстов с ошибками оптического распознавания, как это было выполнено в работе [73]. Модель реализована в виде конечного автомата, требующего обучения, и позволяет исправить до 80% ошибок по оценке точности распознавания на уровне слов. Поскольку текущая работа направлена на корректировку распознанных текстов архивов, которые не обладают базой эталонных текстов для обучения, описанный подход не может быть применен.

В другой работе [110] используется статистический подход, основанный на символьных n-граммах, биграммах слов и механизме изучения вероятностей замены символов. Символьные n-граммы используются для извлечения списка корректировок из словаря. Извлекаются корректировки, обладающие достаточным количеством общих n-грамм с ошибочным словом. Извлеченные корректировки затем упорядочиваются по взвешенному расстоянию Левенштейна, где весом является вероятность вставки, замены или удаления отдельного символа. Далее для каждого предложения производится поиск оптимальной последовательности слов с учетом корректировок по алгоритму Витерби [112]. В данной реализации алгоритм Витерби учитывает вероятность появления биграмм слов и подсчитанный ранг каждой корректировки. После корректировки определенного количества результатов распознавания, на основе этих результатов собирается информация о вероятностях символов (весах расстояния Левенштейна). Предложенный метод позволяет снизить количество

ошибок до 60%, но ключевой проблемой остается высокая вероятность некорректного самообучения системы при наличии неточного алгоритма корректировки.

Работа [90] нацелена на корректировку результатов распознавания рукописных имен собственных. Для отбора корректировок применяется стохастический конечный автомат, который используется для построения оптимального пути нахождения корректировок. Поиск пути реализован через расширенный алгоритм Витерби и заключается в подборе корректировок по словарю, где за вес слова принимается взвешенное расстояние Левенштейна, но без явного вычисления веса для каждого слова. Тренировочная база используется для определения весов автомата. В качестве тренировочной базы применяются словари с большим количеством имен. Описанный подход позволяет провести корректировку 95% ошибочных имен и является довольно перспективным решением для обработки узкоспециализированных текстов.

Помимо рукописных документов, технологии оптического распознавания часто применяются и для обработки исторических документов. Особенностями исторических документов являются вариабельность стилей, словарей, используемых шрифтов. Для корректировки таких текстов современные словари малопригодны. Работа [60] основывается на использовании модели “noisy channel”, построенной по ряду специализированных словарей старонемецких текстов. Примеры для обучения модели генерируются из базы, содержащей исторические варианты написания слов. Предварительные эксперименты показали довольно низкое сокращение ошибок на тестовых примерах, так как исторические тексты очень трудны для обработки.

Тренировочные примеры требуются как для построения модели “noisy channel”, так и для расчета весов расстояния Левенштейна. Поскольку подобные тренировочные базы требуют больших накладных расходов на их подготовку, в работе [61] был предложен алгоритм самостоятельного обучения системы. Для каждого слова из корпуса распознанных текстов выбирается ряд схожих слов из словаря, находящихся в заданном пределе расстояния Левенштейна. Выбранные

слова далее используются для самообучения. Сформированные таким образом весовые характеристики являются более достоверными, чем весовые характеристики, сформированные только лишь с использованием расстояния Левенштейна, но уступают по своей достоверности сгенерированным по вручную подготовленным эталонным текстам.

В работе [79] реализовано несколько алгоритмов нечеткого полнотекстового поиска по результатам оптического распознавания. Среди них способ, основанный на классификации символов. Все символы или группы символов разбиты по категориям, каждая категория обладает определенной канонической формой записи. Для каждого слова в тексте вычисляется каноническая форма записи (хэш-код) и формируется хэш-таблица всех слов в тексте. При поиске достаточно вычислить хэш-код искомого слова и выбрать из хэш-таблицы все соответствующие результаты.

Также направление классификации символов и построения хэшей слов исследуется в работе [71]. Данные разработки дают неплохие результаты, если заранее известны всевозможные варианты трансформации и ошибочного распознавания символов.

В системе полуавтоматической корректировки результатов распознавания OCRspell [106], помимо описанных методов, используется механизм ручного обучения. Оператор системы вручную корректирует ошибочные слова, а система динамически производит сравнение ошибочного слова с его корректным заместителем. В процессе сравнения формируются сопоставления ошибочно распознанных символов их корректной форме, например для замены «оризик@» -> «физика» будут сформированы два сопоставления: «ор» -> «ф» и «@» -> «а». Сопоставления, которые встречаются чаще, обладают большим весом. С ростом количества сопоставлений отмечается заметное падение производительности, так как при поиске корректировок приходится производить перебор большого количества возможных сочетаний.

Другая система корректировки ошибок распознавания AfterScan [36] осуществляет проверку орфографии, анализ текста и обеспечивает следующие

возможности: автоматическое исправление ошибок распознавания и ошибок ручного ввода; чистку отступов, пробелов и пунктуации; приведение к типографским нормам; переформатирование старых текстов с фиксированными переносами строк, переносами слов и отбивкой пробелами; автоматическую работу без вмешательства пользователя в пакетном режиме; возможность легкой проверки и исправления ошибок через журнал исправлений. Данная система распространяется на коммерческих условиях, функционирует под ОС Windows, не обладает программным интерфейсом для запуска пакетной обработки, предназначена для решения типовых пользовательских задач и, как следствие, не подходит для полноценного решения задач диссертационной работы.

Следующая работа [98] посвящена корректировке результатов распознавания исторических документов на датском и английском языках. Процесс корректировки запускается после полного окончания распознавания всего корпуса документов. Вместо поиска корректировок для ошибочно написанных слов, производится поиск всех встречающихся форм написания для каждого слова по словарю и высокочастотному списку слов, сформированному по корпусу текстов. Отбираются только те слова, расстояние Левенштейна до которых не превышает заданный порог. Алгоритм поиска схожих слов основывается на вычислении хэша по методу анаграмм, первоначально описанного Мартином Рейнартом в работе по корректировке орфографических ошибок [96] и доработанного позднее для обработки OCR ошибок.

### **1.6.3 Вывод**

Из рассмотренных методов в диссертационной работе будет использоваться алгоритм нахождения минимального расстояния между словами (расстояние Левенштейна [18]) и алгоритм поиска схожих слов методом анаграмм [98], предложенный Мартином Рейнартом, адаптированный под корректировку ошибок распознавания текста. Выбранные алгоритмы позволяют обрабатывать ошибки типичные для систем оптического распознавания, не требуют проведения

предварительного обучения и могут применяться для обработки текстов независимо от языка написания.

## 1.7 Выводы по первой главе

Анализ предметной области указывает на необходимость разработки решения, позволяющего проводить потоковое распознавание архивных документов в полностью автоматическом режиме.

Система должна обеспечивать распознавание архивных документов различных тематических областей и различных категорий: от высококачественных документов, напечатанных на современных принтерах до документов среднего качества, напечатанных на печатной машинке в середине 20 века. Для этого требуется обеспечить возможность подключения различных коммерческих и свободно распространяемых OCR систем, обладающих способностью к распознаванию русскоязычных текстов. Дополнительным требованием к OCR системам является наличие в возвращаемых результатах информации о координатах распознанных слов.

Анализ точности результатов распознавания показал, что каждая из рассмотренных в главе OCR систем допускает ошибки. Таким образом, актуальной задачей становится разработка и программная реализация алгоритмов корректировки результатов распознавания.

Из обзора существующих методов, проектов и систем корректировки ошибок следует, что в общем случае неплохо решается ряд задач по обработке результатов распознавания с использованием словарей, статистических моделей языка, хорошо развита тематика обнаружения и коррекции ошибок в тексте. Тем не менее, во многих случаях указанные методы предназначены для обработки современных текстов и не подходят в чистом виде для обработки исторических текстов, содержащих большое количество специализированных терминов, имен собственных, географических наименований и т.п. В большинстве работ корректировка основана на предварительном ручном обучении системы или участии человека на этапе финального выбора корректировки. Также стоит

отметить, очень малое количество работ нацеленных на корректировку именно русскоязычных текстов. Это вызывает потребность разработки алгоритма корректировки, учитывающего особенности русского языка и позволяющего обрабатывать корпуса текстов больших объемов в полностью автоматическом режиме.

Практическая задача заключается в распознавании электронных образов научно-справочного аппарата центральных государственных архивов Санкт-Петербурга, специализирующихся на хранении документов различной направленности: историко-политические документы, документы по литературе и искусству, документы по личному составу ликвидированных государственных предприятий, научно-техническая документация.

Отдельно стоит отметить необходимость разработки критерия оценки качества распознавания, учитывающего требования поисковой системы по изображениям.

## Глава 2. Автоматическая корректировка ошибок оптического распознавания на основе рейтинго-ранговой модели текста

### 2.1 Описание используемых методов

Предлагаемый в диссертационной работе метод [26] использует расстояние Левенштейна [18] в качестве критерия оценки близости корректировки и ошибочного слова, а также алгоритм поиска схожих слов методом анаграмм [98].

#### 2.1.1 Расстояние Левенштейна

Для оценки степени близости (сходства) двух слов может быть применено расстояние Левенштейна [18].

*Расстояние Левенштейна* (также редакционное расстояние или дистанция редактирования) — это минимальное количество операций вставки, удаления и замены символов, необходимых для превращения одной строки в другую [20].

Значительный вклад в изучение и развитие вопроса вычисления расстояния Левенштейна внёс Дэн Гасфилд [9].

*Вес операции* (вставка, удаление, замена) может отличаться в зависимости от вида операции и набора обрабатываемых символов, отражая разную вероятность появления ошибки распознавания текста:

- $\text{cost}(a,b)$  — вес операции замены символа  $a$  на символ  $b$ ;
- $\text{cost}(\varepsilon,b)$  — вес операции вставки символа  $b$ , где  $\varepsilon$  — пустой символ;
- $\text{cost}(a,\varepsilon)$  — вес операции удаления символа  $a$ .

Для вычисления расстояния Левенштейна требуется определить последовательность замен с минимальным суммарным весом. Для классического расстояния Левенштейна используется следующая формула вычисления веса:

$$\text{cost}(a,b) := \begin{cases} 0, & \text{если } a = b \\ 1, & \text{если } a \neq b \end{cases}.$$

Вычисление классического значения, значения при произвольных  $\text{cost}(a,b)$ , решается по алгоритму Вагнера-Фишера [114]. Пусть  $x = x_1 \dots x_n$  и  $y = y_1 \dots y_m$  —

две строки с длинами  $n$  и  $m$  соответственно, тогда расстояние Левенштейна  $ld(x, y)$  можно посчитать по следующей рекуррентной формуле:

$$\begin{aligned}
 ld(x, y) &= D(n, m), \\
 D_{0,0} &= 0, \\
 D_{i,0} &= D_{i-1,0} + \text{cost}(x_i, \varepsilon), 1 \leq i \leq n, \\
 D_{0,j} &= D_{0,j-1} + \text{cost}(\varepsilon, y_j), 1 \leq j \leq m, \\
 D_{i,j} &= \min \left( \begin{array}{l} D_{i-1,j} + \text{cost}(x_i, \varepsilon) \\ D_{i,j-1} + \text{cost}(\varepsilon, y_j) \\ D_{i-1,j-1} + \text{cost}(x_i, y_j) \end{array} \right)
 \end{aligned}$$

где  $\min(a, b, c)$  возвращает наименьшее из трёх аргументов.

Справедливы следующие утверждения:

$$\begin{aligned}
 D(x, y) &\geq \|x\| - \|y\|, \\
 D(x, y) &\leq \max(\|x\|, \|y\|). \\
 D(x, y) = 0 &\Leftrightarrow x = y
 \end{aligned}$$

Используя вышеприведённую формулу можно вычислить матрицу  $D$ , причем вычисление можно проводить по строкам или по столбцам.

Матрица расчета расстояния Левенштейна для слов «разговор» и «расовоор» представлена на рисунке 2.1. Как видно из рисунка, значение расстояния Левенштейна равняется значению  $D_{8,8} = 3$ .

x \ y	r	a	c	o	v	o	o	r	
x	0	1	2	3	4	5	6	7	8
r	1	0	1	2	3	4	5	6	7
a	2	1	0	1	2	3	4	5	6
z	3	2	1	1	2	3	4	5	6
г	4	3	2	2	2	3	4	5	6
о	5	4	3	3	2	3	4	5	6
в	6	5	4	4	3	2	3	4	5
о	7	6	5	5	4	3	2	3	4
р	8	7	6	6	5	4	3	3	3

2.1. Матрица расчета расстояния Левенштейна для слов «разговор» и «расовоор»

### 2.1.2 Метод анаграмм

Метода анаграмм [97-99] предназначается для поиска вариантов корректировки ошибочного слова по заранее подготовленному словарю, содержащему наиболее высокочастотные слова из всего корпуса распознанных текстов. Вместо простого перебора всех слов выборка осуществляется по хэшу из предварительно сформированной хэш-таблицы анаграмм  $H^{anagram}$ . Ключом таблицы является «плохая» хэш-функция, которая выдает числовое значение одинаковое для всех слов, являющихся анаграммами. Значениями таблицы выступают соответствующие ключу списки слов из словаря.

Под анаграммой понимают перестановку букв, посредством которой образуются новые слова и фразы [34]. Таким образом, слова «ТОК», «КОТ», «ОТК» будут являться словами анаграммами.

Значение хэш-функции будем называть анаграммным ключом. Каждому символу  $c_i$  используемого алфавита  $A = c_1, \dots, c_{|A|}$ , сопоставим уникальный числовой идентификатор из кодовой таблицы UTF-8. Значение хэш-функции для слова  $w = c_1, \dots, c_{|w|}$  вычисляется по следующей формуле:

$$\text{hash}(w) = \sum_{i=1}^{|w|} \text{int}(c_i)^n,$$

где  $\text{int}(c)$  — значение числового кода символа  $c$  в кодовой таблице UTF-8, а  $n$  — значение степени, в которую возводится числовой код. Возведение числового кода в степень  $n$  необходимо для устранения появления коллизий, в работе [99] установлено достаточное значение  $n = 5$ .

Введем ряд определений для описания поиска корректировок по методу анаграмм.

*Ключевое слово* — слово, требующее корректировки, для которого происходит отбор корректировок на замену.

*Алфавит*  $A_{List}^k$ , содержащий  $n$ -граммы всех символов и знаков, входящих в состав списка слов  $List$ , с добавлением символа пробела в начало и конец каждого слова:

$$A_{List}^k = \{ngrams(sws, n) \mid w \in List, 1 \leq n \leq k\},$$

где  $s$  — символ пробела,  $ngrams(w, n)$  — функция, возвращающая список  $n$ -грамм слова  $w$ .

*Строгий алфавит*  $A_{List}^{k-}$ , содержащий  $n$ -граммы всех символов и знаков, входящих в состав списка слов  $List$ , без добавления символа пробела:

$$A_{List}^{k-} = \{ngrams(w, n) \mid w \in List, 1 \leq n \leq k\}.$$

*Поисковый хэш-алфавит*  $X_{List}^k$ , состоящий из анаграммных ключей символьных  $n$ -грамм списка слов  $List$ :

$$X_{List}^k = \{hash(g) \mid g \in A_{List}^k\}.$$

*Ключевой хэш-алфавит*  $\Phi_{\{w\}}^k$ , состоящий из анаграммных ключей символьных  $n$ -грамм ключевого слова  $w$ :

$$\Phi_{\{w\}}^k = \{hash(g) \mid g \in A_{\{w\}}^{k-}\}.$$

Отметим, что ключевой алфавит строится на основе строгого алфавита. Рассмотрим правила построения алфавитов на практическом примере.

Пусть список слов состоит из двух слов:

$$List = \{\text{трудовые, договоры}\},$$

а  $k = 2$ .

Тогда алфавит  $A_{List}^2$  будет содержать 10 одиночных символов и 16 биграмм:

$$A_{List}^2 = \{ , m, p, y, \partial, o, e, \text{ы}, \text{и}, \text{е}, \text{з}, \text{м}, \text{т}, \text{р}, \text{у}, \text{у}, \text{д}, \text{о}, \text{о}, \text{в}, \text{ы}, \text{ы}, \text{е}, \text{е}, \text{,}, \text{д}, \text{о}, \text{г}, \text{з}, \text{о}, \text{в}, \text{о}, \text{р}, \text{п}, \text{ы}, \text{ы} \}$$

Поисковый хэш-алфавит  $X_{List}^2$  будет состоять из хэшей всех символьных последовательностей алфавита  $A_{List}^2$ . Причем мощность множества  $X_{List}^2$  будет меньше мощности множества  $A_{List}^2$  на 2, так как  $hash(o\partial) = hash(g\partial)$  и  $hash(o\text{в}) = hash(\text{в}o)$ .

Строгий алфавит ключевого слова  $w = \text{трубоые}$  может быть вычислен с другим значением  $k = 3$  и будет состоять из 7 одиночных символов, 7 биграмм и 6 триграмм:

$$A_{\{w\}}^{3-} = \{m, p, y, \varepsilon, o, \varepsilon, e, mp, py, y\varepsilon, \varepsilon o, o\varepsilon, \varepsilon y, \varepsilon e, tpy, rpy, ywo, wo\varepsilon, owo, owo\varepsilon, wo\varepsilon e\}.$$

Ключевой хэш-алфавит  $\Phi_{\{w\}}^3$  будет содержать хэши всех элементов алфавита  $A_{\{w\}}^{3-}$ , но его длина будет меньше на 1, так как  $\text{hash}(o\varepsilon) = \text{hash}(wo)$ .

Главной задачей метода анаграмм является нахождение списка корректировок  $W$ , схожих по написанию с заданным ключевым словом. Схожие слова могут быть получены путем вставки, удаления, замены или перестановки символов [50].

Выделим основную характеристику хэш-функции:

$$\text{hash}(w) = \sum_{i=1}^{|w|} \text{hash}(c_i) = \text{hash}(c_1 \dots c_{k-1}) + \text{hash}(c_k) + \text{hash}(c_{k+1} \dots c_{|w|}), \forall 1 \leq k \leq |w|,$$

из которой следует, что общее значение может быть рассчитано, как сумма значений элементов произвольной длины.

Данная характеристика позволяет определить четыре пути по извлечению всего набора схожих слов для ключевого слова  $w$ :

1. Перестановка. Для извлечения всех слов, полученных путем перестановки символов, достаточно однократной выборки из хэш-таблицы по анаграммному ключу исходного слова:

$$\text{hash}(\text{слово}) = \text{hash}(\text{слоов}) = \text{hash}(\text{волос}).$$

2. Удаление. Каждый символ или символьную последовательность ключевого алфавита  $\Phi_{\{w\}}^k$  необходимо вычесть из анаграммного ключа ключевого слова:

$$\text{hash}(\text{слово}) - \text{hash}(l) = \text{hash}(\text{слово}).$$

3. Вставка. Каждый символ или символьную последовательность поискового хэш-алфавита  $X_{List}^k$  необходимо прибавить:

$$\text{hash}(\text{слово}) + \text{hash}(o) = \text{hash}(\text{слово}).$$

4. Замена. Каждый символ или символьную последовательность ключевого хэш-алфавита  $\Phi_{\{w\}}^k$  необходимо вычесть, а каждый символ или символьную последовательность поискового хэш-алфавита  $X_{List}^k$  прибавить:

$$\text{hash}(\text{слого}) - \text{hash}(z) + \text{hash}(e) = \text{hash}(\text{слово}).$$

На практике, операция «замена» является общей для оставшихся трех операций:

- a) для получения операции «удаление» достаточно прибавить 0 и вычесть элемент ключевого алфавита;
- b) для получения операции «вставка» достаточно вычесть 0 и прибавить элемент поискового алфавита;
- c) для получения операции «перестановка» достаточно ничего не вычитать и не прибавлять.

Рассмотрим псевдокод алгоритма отбора корректировок по методу анаграмм:

```
#w - ключевое слово
#ldLimit - пороговое значение расстояния Левенштейна
#searchAlph - поисковый алфавит
#keyAlph - ключевой алфавит
Function anagramSearch(w, ldLimit, searchAlph, keyAlph) {
    W ← Ø;
    R[] ← Ø;
    keyWordHash ← hash(w);
    for all ck ∈ keyAlph do
        for all cs ∈ searchAlph do
            simulatedWordHash ← keyWordHash + cs - ck;
            W' ← getWords(simulatedWordHash);
            for all w' ∈ W' do
                if LD(w, w') ≤ ldLimit then
                    W ← W ∪ w';
                    R[w'] = R[w'] + 1;
```

```

        end if
    end for
end for
return W;
}

```

Функция  $getWords(simulatedWordHash)$  возвращает список слов-анаграмм по хэшу  $simulatedWordHash$  из хэш-таблицы анаграмм  $H^{anagram}$ .

Между каждой выбранной корректировкой и ключевым словом вычисляется расстояние Левенштейна, и если оно больше заданного порога  $ldLimit$ , то корректировка исключается из результирующего списка.

Метод обладает особенностью саморанжирования отбираемых слов: некоторые слова могут быть выбраны из хэш-таблицы несколько раз, чем чаще слово было получено, тем выше его ранг. Количество повторений сохраняется для каждого слова в ассоциативном массиве  $R$ . Особенность саморанжирования проиллюстрирована на рисунке 2.2.

ОБЪЕМ - ключевое слово

Перестановка n-грамм ключевого слова:

в результатах нет слова «ОБЪЕМ»

Удаление юниграмм ключевого слова:

<span style="border: 1px solid black; padding: 2px;">ОБЪЕМ</span>	-	<span style="border: 1px solid black; padding: 2px;">О</span>	=	<span style="border: 1px solid black; padding: 2px;">ББЬЕМ</span>
<span style="border: 1px solid black; padding: 2px;">ОБЪЕМ</span>	-	<span style="border: 1px solid black; padding: 2px;">Б</span>	=	<span style="border: 1px solid black; padding: 2px; background-color: green;">ОБЬЕМ</span>
<span style="border: 1px solid black; padding: 2px;">ОБЪЕМ</span>	-	<span style="border: 1px solid black; padding: 2px;">Б</span>	=	<span style="border: 1px solid black; padding: 2px; background-color: green;">ОБЬЕМ</span>
<span style="border: 1px solid black; padding: 2px;">ОБЪЕМ</span>	-	<span style="border: 1px solid black; padding: 2px;">ТЬ</span>	=	<span style="border: 1px solid black; padding: 2px;">ОББЕМ</span>
<span style="border: 1px solid black; padding: 2px;">ОБЪЕМ</span>	-	<span style="border: 1px solid black; padding: 2px;">Е</span>	=	<span style="border: 1px solid black; padding: 2px;">ОББ'ЕМ</span>
<span style="border: 1px solid black; padding: 2px;">ОБЪЕМ</span>	-	<span style="border: 1px solid black; padding: 2px;">М</span>	=	<span style="border: 1px solid black; padding: 2px;">ОББ'Е</span>

Удаление биграмм ключевого слова:

в результатах нет слова «ОБЪЕМ»

Вставка n-грамм:

в результатах нет слова «ОБЪЕМ»

Замена. Вычитание n-грамм и добавление n-грамм:

<span style="border: 1px solid black; padding: 2px;">ОБЪЕМ</span>	+	<span style="border: 1px solid black; padding: 2px;">Б</span>	-	<span style="border: 1px solid black; padding: 2px;">БВ</span>	=	<span style="border: 1px solid black; padding: 2px; background-color: green;">ОБЬЕМ</span>
			...			
<span style="border: 1px solid black; padding: 2px;">ОБЪЕМ</span>	+	<span style="border: 1px solid black; padding: 2px;">О</span>	-	<span style="border: 1px solid black; padding: 2px;">OB</span>	=	<span style="border: 1px solid black; padding: 2px; background-color: green;">ОБЬЕМ</span>
			...			
<span style="border: 1px solid black; padding: 2px;">ОБЪЕМ</span>	+	<span style="border: 1px solid black; padding: 2px;">ТЬ</span>	-	<span style="border: 1px solid black; padding: 2px;">БЬ</span>	=	<span style="border: 1px solid black; padding: 2px; background-color: green;">ОБЬЕМ</span>

Вычислительная сложность метода зависит от размеров алфавитов, большие алфавиты будут приводить к увеличению количества запросов в хэш-таблицу, а алфавиты меньшего размера будут приводить к их сокращению. Время выборки значений из хэш-таблицы является неизменной величиной  $\Theta(1)$ .

## 2.2 Общий алгоритм корректировки

Разделим весь процесс корректировки результатов распознавания на четыре основных этапа (рисунок 2.3):

1. Подготовка структур данных.
2. Генерация корректировок.
3. Ранжирование корректировок.
4. Формирование результата.



2.3. Общий алгоритм корректировки

В ходе предварительного этапа подготовки структур данных производится сбор статистической информации по всему корпусу распознанных документов и тематических текстов, формируется целый ряд тезаурусов, словарей и хэш-таблиц, содержащих необходимые данные для этапа генерации корректировок.

Этап генерации корректировок является основным этапом обработки, на котором для каждого ошибочно распознанного слова формируются списки корректировок на замену. Все множество ошибок распознавания можно разделить на множество ошибок 1-го рода (пропущенные слова) и множество ошибок 2-го рода (ошибочно распознанные слова). На этапе генерации корректировок обработке подвергаются только ошибки 2-го рода.

Далее каждой корректировке присваивается ранг и производится финальное упорядочивание корректировок по убыванию их ранга.

На последнем этапе производится выборка наиболее вероятных корректировок и сохранение финального результата распознавания в формате XML.

## **2.3 Подготовка структур данных**

На первом шаге необходимо произвести анализ всего корпуса распознанных документов для формирования статистической информации по встречающимся словам.

### **2.3.1 Предварительная обработка**

Назовем лексемой последовательность символов, разделенных пробелом или символами  $\{.,;()\"&[]?!\'{ }/+=<>%\}$ , либо определенных системой распознавания как слова.

Выразим весь набор лексем, полученных в результате распознавания документов, в виде упорядоченного по порядку следования элементов множества  $L^{source} = \{s_1, s_2, \dots, s_m\}$ .

Под процедурой нормализации будем понимать преобразование последовательности  $L^{source}$  в нормализованную последовательность лексем  $L = \{s_1, s_2, \dots, s_n\}$ .

Процедура нормализации состоит из следующих шагов:

1. Очистка лексем.

В начале и конце каждой лексемы удаляются все неалфавитные символы.

Неалфавитными символами будем считать символы, не входящие во множество символов русского алфавита {а-я, А-Я}.

2. Замена символов.

Все производные символа тире заменяются символом обычного тире.

Все производные пробельного символа заменяются символом обычного пробела.

3. Объединение лексем, разделенных знаком переноса.

Две лексемы  $s_1$  и  $s_2$  объединяются в одну, в том случае если:

- a) Лексемы расположены на разных строках.
- b) Лексема  $s_1$  заканчивается символом «-» после удаления всех неалфавитных символов в конце лексемы.
- c) В лексеме  $s_1$  перед символом «-», стоит символ в нижнем регистре
- d) Лексема  $s_2$  обладает длиной более 2-х символов и начинается с символа в нижнем регистре после удаления всех неалфавитных символов в начале лексемы.

### 2.3.2 Структуры для отбора корректировок

Получив нормализованную последовательность  $L$ , сформируем множество лексем в нижнем регистре символов  $L^{low}$  и его рейтинговое распределение  $\zeta_{L^{low}}$ :

$$L^{low} = \{lower(s) \mid s \in L\},$$

где  $lower(s)$  — функция перевода строки в нижний регистр.

Исходя из предположения, что наиболее часто встречающиеся лексемы с наибольшей вероятностью не содержат ошибок, а также с целью уменьшения

поискового пространства проведем сокращение множества  $L^{low}$  и его рейтингового распределения  $\xi_{L^{low}}$  до множества  $L^{lowpruned}$  и рейтингового распределения  $\xi_{L^{lowpruned}}$ :

$$L^{lowpruned} = \{s \mid s \in L^{low}, \xi_{L^{low}}(s) \geq \alpha\}$$

где  $\alpha$  — минимально допустимое количество повторений одной лексемы.

Выбор значения  $\alpha$  является своего рода компромиссом. При низком значении может остаться большее количество ошибочных лексем, а при высоком могут быть потеряны редкие имена собственные, географические наименования и т.п.

Далее проведем сбор статистической информации о вхождении парных лексем. Для этого сформируем множество биграмм  $L^{bigram}$  и его рейтинговое распределение  $\xi_{L^{bigram}}$ :

$$L^{bigram} = \{(lower(s_1), lower(s_2)) \mid s_1, s_2 \in L; (seq(s_1, s_2) \vee seq(s_2, s_1))\},$$

где функция  $seq(a_1, \dots, a_z)$  возвращает значение «истина», если элементы  $a_1 - a_z$  следуют строго друг за другом, и «ложь» в противном случае. Порядок следования лексем в паре не имеет значения, то есть пары  $seq(s_1, s_2)$  и  $seq(s_2, s_1)$  считаются равными.

Сбор биграмм производится без учета знаков препинания. Это обусловливается тем, что в результатах распознавания может присутствовать большое количество ошибочных знаков препинания, полученных из-за наличия «шума» на исходном изображении. Главной задачей является сбор максимального количества биграмм для избегания проблем с разреженностью данных и корректировки ошибочно объединенных слов.

Проведем сокращение множества биграмм  $L^{bigram}$  и его рейтингового распределения  $\xi_{L^{bigram}}$  до множества  $L^{bipruned}$  и его рейтингового распределения  $\xi_{L^{bipruned}}$ :

$$L^{bipruned} = \left\{ (s_1, s_2) | (s_1, s_2) \in L^{bigram}; len(s_1), len(s_2) > 1; \xi_{L^{bigram}}(s_1, s_2) \geq \beta \right\},$$

где  $len(s)$  — количество символов в строке  $s$ , а  $\beta$  — минимальное пороговое значения количества повторений одной биграммы. Ограничение по длине лексемы в биграмме введено для того, чтобы избежать нежелательного разбиения слов при корректировке и сократить пространство поиска.

Сформируем основные структуры данных для генерации кандидатов на замену ошибочных слов: множество корректировок  $L^{corr}$  и хэш-таблицу анаграмм  $H^{anagram}$ .

$$L^{corr} = L^{lowpruned} \cup \left\{ "s_1 s_2" | (s_1, s_2) \in L^{bipruned} \right\},$$

$$H^{anagram} = \left\{ \langle hash(s), \langle s, \xi_{L^{corr}}(s) \rangle \rangle | s \in L^{corr} \right\},$$

где  $\xi_{L^{corr}}$  — рейтинговое распределение множества всех лексем для выбора корректировок  $L^{corr}$ .

Для каждого элемента множеств  $L^{corr}$  вычисляется значение хэш-функции  $hash(s)$  и производится добавление записи в хэш-таблицу  $H^{anagram}$ , ключом которой является значение хэш-функции, а значением — список всех элементов с их рейтингом, обладающих соответствующим значением хэш-функции. Описание алгоритма вычисления значения хэш-функции представлено выше.

### 2.3.3 Структуры для ранжирования корректировок

Произведем нормализацию морфологической формы каждой лексемы множества  $L$ , используя функцию морфологического анализа  $morph$ :

$$morph(s) = \sum_b, \quad b \in \sum_b, \quad s \in \sum_s \rightarrow b \in \sum_s,$$

где  $\sum_b$  — множество лемм (основных форм) лексемы  $s$ ,  $\sum_s$  — множество словоформ лексемы  $s$ .

Функция морфологического анализа  $morph$  обладает следующими свойствами:

$$\text{morph}(b) = b, \forall s \notin \Sigma_s \rightarrow \text{morph}(s) = b, b \notin \Sigma_s.$$

В результате перевода лексем в нормальную форму получим множество лексем:

$$L^{\text{lemm}} = \left\{ \text{morph}(\text{lower}(s)) \mid s \in L, s \notin D^{\text{stop}} \right\},$$

где  $D^{\text{stop}}$  — список стоп-слов.

Реализация функции перевода лексем в нормальную форму с помощью внешнего модуля морфологического анализа системы «АОТ» [1], позволяет генерировать нормальные (базовые) формы на основе морфологических предсказаний [28] даже для лексем, отсутствующих в словаре.

Сформируем отношения  $L_1^{\text{lemm}}, L_2^{\text{lemm}}$  для связок лексем множества  $L^{\text{lemm}}$  и их рейтинговые распределения  $\xi_{L_1^{\text{lemm}}}, \xi_{L_2^{\text{lemm}}}$ :

$$L_1^{\text{lemm}} = L^{\text{lemm}},$$

$$L_2^{\text{lemm}} = \left\{ (b_1, b_2) \mid b_1, b_2 \in L^{\text{lemm}}; \text{seq}(b_1, b_2) \right\}.$$

#### 2.3.4 Структуры для обнаружения ошибок

Опишем следующую структуру данных — корпусный тезаурус  $D^{\text{corpus}}$ , который будет впоследствии применяться для ограничения области лексем, подлежащих корректировке:

$$D^{\text{corpus}} = L^{\text{lowpruned}} \cap (D^{\text{general}} \cup D^{\text{special}}),$$

где  $D^{\text{general}}$  — словарь общих слов русского языка,  $D^{\text{special}}$  — тематические тезаурусы, содержащие специфические термины предметной области документов (имена собственные, географические наименования, аббревиатуры и т.п.).

### 2.4 Генерация корректировок

Пусть последовательность  $\text{Lex}^{\text{source}} = \{s_1, s_2, \dots, s_m\}$  — упорядоченный по порядку следования набор лексем, полученных в результате распознавания отдельного изображения документа.

Тогда  $\text{Lex} = \{s_1, s_2, \dots, s_n\}$  — результат нормализации последовательности  $\text{Lex}^{\text{source}}$ .

Разделим все множество лексем  $\text{Lex}$  на множество лексем  $\text{Lex}^{\text{error}}$ , подлежащих корректировке, и множество лексем  $\text{Lex}^{\text{correct}}$ , которые будем считать корректно распознанными:

$$\text{Lex} = \text{Lex}^{\text{error}} \cup \text{Lex}^{\text{correct}},$$

$$\text{Lex}^{\text{error}} = \text{Lex} \setminus \text{Lex}^{\text{correct}}.$$

В область лексем  $\text{Lex}^{\text{correct}}$ , не подлежащих корректировке, отнесем лексемы, для которых найдено соответствие в словаре  $D^{\text{corpus}}$  или длина которых меньше порогового значения  $\varphi$ :

$$\text{Lex}^{\text{correct}} = \{s \mid s \in \text{Lex}; (\text{len}(s) \leq \varphi \vee s \in D^{\text{corpus}})\}.$$

Задача генерации корректировок сводится к отбору множества корректировок  $W_i \subset L^{\text{corr}}$  для замены каждой лексемы  $s_i \in \text{Lex}^{\text{error}}$ , где  $i \in |1 \dots \text{Lex}^{\text{error}}|$ .

Отбор множества  $W$  осуществляется по методу анаграмм, описанному в разделе 2.1.2. Рассмотрим особенности адаптации метода анаграмм в данной работе.

Алфавиту  $A_{\text{List}}^k$  установим значения  $k = 2$  и  $\text{List} = L^{\text{lp}}$ . Введем дополнительное ограничение на построение алфавита  $A_{L^{\text{lp}}}^2$  такое, что в его состав могут входить только те символы и символьные последовательности, из которых формируются корректные слова:

$$C = \{a..я\} \cup \{-, s\},$$

$$A_{L^{\text{lp}}}^2 = C \cup \{c_1, c_2 \mid \text{seq}(c_1, c_2) \in \text{sfs}; w \in L^{\text{lowpruned}}; c_1, c_2 \in C\},$$

где  $s$  — пробельный символ. Добавление пробельного символа в начало и конец слова  $w$  необходимо для обеспечения возможности корректировки ошибочно объединенных слов.

В поисковый алфавит и ключевой алфавит добавим нули. Таким образом, основная формула для вычисления корректировок ключевого слова  $w$  выглядит следующим образом:

$$\text{simulatedWordHash} \leftarrow \text{keyWordHash} + cs - ck,$$

$$cs \in \{0\} \cup X_{L^{\text{lowpruned}}}^2, ck \in \{0\} \cup \Phi_{\{w\}}^2.$$

## 2.5 Ранжирование корректировок

После получения множества корректировок  $W$ , необходимо определить вероятность каждой из них и провести их ранжирование.

Ранжирование будем производить в два шага:

$$W \xrightarrow{1} \bar{W} \xrightarrow{2} \hat{W}.$$

На первом шаге производится оценка каждой корректировки, отобранный методом анаграмм, и формирование множества  $\bar{W}$ , упорядоченного по убыванию оценки.

На втором шаге вычисляется финальный ранг и формируется множество  $\hat{W}$ , упорядоченное по значению финального ранга.

Шаг 1. Инвариантная оценка соответствия корректировки  $w$  для замены лексемы  $s$  [2]:

$$\text{score}(s, w) = \ln(\xi_{L^{\text{corr}}}(w)) \times (|w| - LD(s, w)) \times R(w) \times d_{\text{factor}},$$

$$d_{\text{factor}} = \begin{cases} \pi, & \text{если } s \in D^{\text{corpus}} \\ 1, & \text{если } s \notin D^{\text{corpus}} \end{cases}$$

где  $\xi_{L^{\text{corr}}}(w)$  — частота повторения слова  $w$  во всем корпусе слов  $L^{\text{corr}}$ ;  $|w|$  — длина корректировки  $w$  в символах;  $LD(s, w)$  — расстояние Левенштейна между словами  $s$  и  $w$ ;  $R(w)$  — количество повторений корректировки  $w$  в ходе отбора

методом анаграмм;  $d_{factor}$  - словарный фактор, увеличивает вес корректировки, если она встречается в корпусном тезаурусе.

Вычисление оценки  $score(s, w)$  основывается на трех предположениях:

1. Исходное множество корректировок  $L^{corr}$  может содержать ошибки распознавания, так как оно было получено из корпуса распознанных документов. Вследствие этого, будем считать, что корректировки, обладающие наибольшей частотой повторения  $\xi_{L^{corr}}(w)$ , наиболее вероятно являются корректными и будем отдавать им предпочтение.

2. По той же причине будем придавать больший вес корректировкам, которые встречаются в предварительно сформированном корпусном тезаурусе  $D^{corpus}$ .

3. Чем меньше расстояние Левенштейна  $LD(s, w)$  между исходной лексемой и корректировкой, тем более предпочтительным будем его считать.

В итоге, на данном шаге производится вычисление оценки  $score(s, w)$  для каждой корректировки из множества  $W$  и формируется упорядоченное по убыванию вычисленной оценки множество:

$$\vec{W} = \{w \mid w \in W, score(s, w_k) \geq score(s, w_{k+1}), 1 \leq k \leq |W|\}.$$

### Шаг 2. Вычисление финального ранга.

Сократим размер множества  $\vec{W}$  до  $n$  элементов,  $|\vec{W}| = \min(n, |\vec{W}|)$ , и вычислим значение финального ранга  $Rank(s, w)$  для каждой корректировки  $w$ :

$$Rank(s, w) = \frac{score(s, w)}{\sum_{j=1}^{|W|} score(s, w_j)} \times P(w),$$

где  $P(w)$  — статистическая вероятность нахождения в тексте корректировки  $w$  на позиции лексемы  $s$ .

Выразим  $P(w_i)$  через языковую модель, основанную на n-граммах, следующим образом:

$$P(w_i) = P(w_i \mid w_{1,i-1}),$$

где  $P(w_i | w_{1,i-1})$  — вероятность появления слова  $w_i$  при наличии предшествующей ему последовательности слов  $w_1, w_2, \dots, w_{i-1}$ .

В случае языковой модели, основанной на биграммах, данное определение можно упростить:

$$P(w_i) = P(w_i | w_{1,i-1}) = \frac{f(w_{i-1}, w_i)}{f(w_{i-1})},$$

где  $f(w_{i-1})$  — частота повторения слова,  $f(w_{i-1}, w_i)$  — частота повторения биграммы  $(w_{i-1}, w_i)$ .

В данной работе, предшествующая лексема может являться ошибочной, поэтому, вместо слова  $w_{i-1}$  будем использовать множество корректировок  $\vec{W}_{i-1}$ , информацию о частоте повторения слов и биграмм будем получать из рейтинговых распределений лексем в нормальной форме  $\xi_{L_1^{lemm}}, \xi_{L_2^{lemm}}$ .

Формула расчета вероятности принимает вид:

$$P(w_i^k) = \frac{\sum_{j=1}^{|\vec{W}_{i-1}|} \xi_{L_2^{lemm}}(\text{morph}(w_{i-1}^j), \text{morph}(w_i^k))}{\sum_{j=1}^{|\vec{W}_{i-1}|} \xi_{L_1^{lemm}}(\text{morph}(w_{i-1}^j))},$$

$$1 \leq k \leq |\vec{W}_i|,$$

где  $w_i^k$  —  $k$ -ая по порядку корректировка лексемы  $s_i$ ,  $w_{i-1}^j$  —  $j$ -ая по порядку корректировка лексемы  $s_{i-1}$ .

В итоге для каждой лексемы  $s \in Lex^{error}$  формируется упорядоченное по убыванию финального ранга множество наиболее вероятных корректировок:

$$\widehat{W} = \left\{ w \mid w \in \vec{W}, Rank(s, w_k) \geq Rank(s, w_{k+1}), Rank(s, w) \in [0..1], 1 \leq k \leq |\vec{W}| \right\},$$

$$|\widehat{W}| = \begin{cases} |\vec{W}|, & \text{если } |\vec{W}| < k \\ k, & \cdots \text{если } |\vec{W}| \geq k \end{cases}.$$

## 2.6 Формирование результата

Результат распознавания представляет собой множество:

$$RES = \left\{ \langle s, \lambda, w^{best}, W^{alternate} \rangle \mid s \in Lex \right\},$$

где признаком  $\lambda$  обозначается, требует лексема корректировки или нет:

$$\lambda = \begin{cases} 0, & \text{если } s \in Lex^{cor} \\ 1, & \text{если } s \in Lex^{err} \end{cases},$$

$w^{best}$  — наилучшая корректировка,  $W^{alternate}$  — дополнительные корректировки.

Выбор наилучшей корректировки  $w^{best}$  производится по следующим правилам:

- Если больше половины символов в лексеме  $s$  являются прописными и среди корректировок  $\hat{W}$ , есть корректировки из словаря аббревиатур  $D^{abbr}$ , то среди них выбирается корректировка с наивысшим рангом  $Rank(s, w)$ :

$$w^{best} = \operatorname{Argmax}_{w \in (\hat{W} \cap D^{abbr})} Rank(s, w).$$

- Если первый символ лексемы  $s$  прописной, а остальные строчные и в списке корректировок  $\hat{W}$ , есть корректировки из словаря фамилий  $D^{surname}$  или имён  $D^{name}$ , то среди них выбирается корректировка с наивысшим рангом  $Rank(s, w)$ :

$$w^{best} = \operatorname{Argmax}_{w \in (\hat{W} \cap (D^{surname} \cup D^{name}))} Rank(s, w).$$

- Если по предыдущим правилам наилучшая корректировка не была выявлена, то выбирается самая первая корректировка из списка  $\hat{W}$ :

$$w^{best} = \hat{w}_1,$$

$$\hat{W} = \{\hat{w}_1 \dots \hat{w}_{|\hat{W}|}\}.$$

В случае если правила 1 и 2 возвращают множество корректировок с одинаковым рангом, то наилучшая корректировка выбирается среди них произвольным образом.

Во множество дополнительных корректировок  $W^{alternate}$  включаются все корректировки  $\hat{W}$  за исключением наилучшей  $w^{best}$ :

$$W^{alternate} = \hat{W} \setminus \{w^{best}\}.$$

## 2.7 Выводы по второй главе

Описанный в главе метод предназначается для автоматической корректировки результатов массового оптического распознавания архивных документов.

Отбор корректировок осуществляется по специальным тезаурусам, построенным на основе частотных характеристик вхождений слов и словосочетаний со всего корпуса распознанных материалов. Использование таких тезаурусов, позволяет производить корректировку текстов различных предметных областей, содержащих узкоспециализированную терминологию, имена собственные, географические наименования и т.п.

Финальное ранжирование отобранных слов-кандидатов производится с учетом контекста и основывается на результатах статистического n-грамм анализа всего корпуса текста в нормальной форме. Реализация функции перевода слова в нормальную форму на основе морфологических предсказаний позволяет генерировать нормальные (основные) формы даже для слов русского языка, отсутствующих в словаре.

При корректировке корпусов большого объема существенное значение имеет скорость выполнения обработки. В данной работе для увеличения скорости производится корректировка только слов с некорректным написанием, не подвергаются корректировке слова с малым количеством символов. Поиск корректировок производится по заранее подготовленным хэш-таблицам, что также обеспечивает высокую скорость обработки.

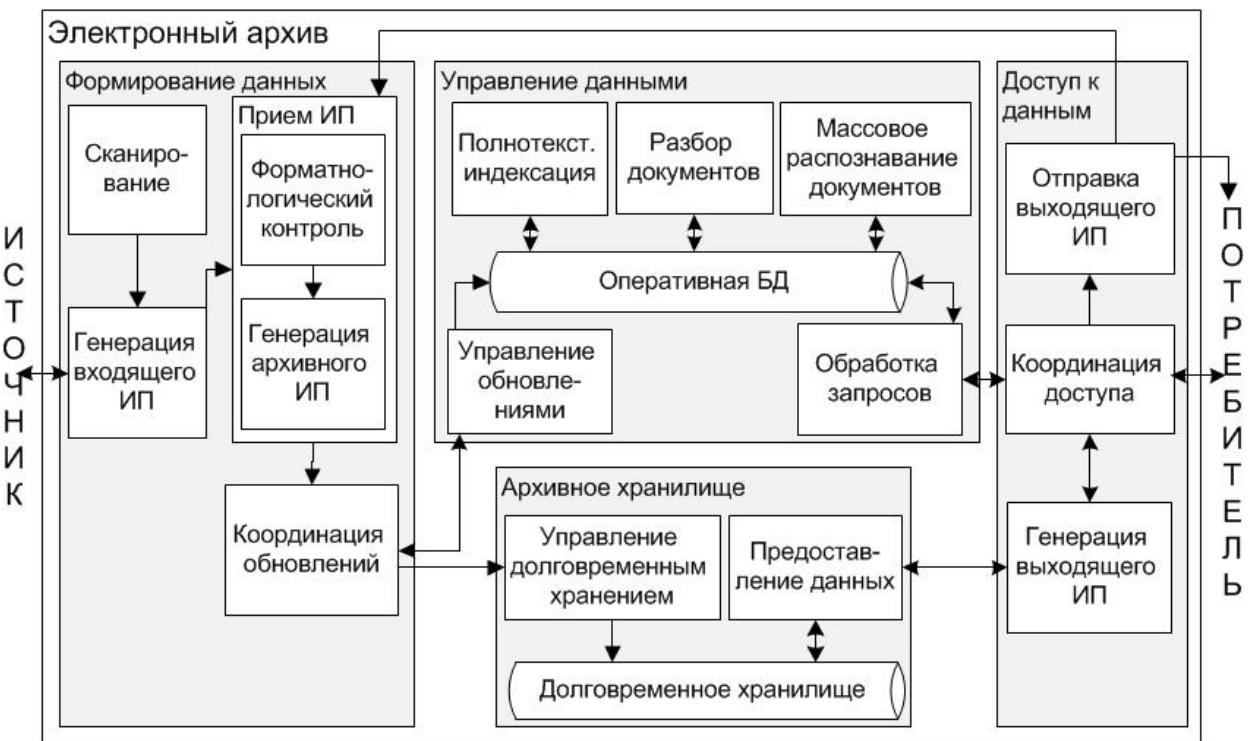
Данный подход не может обеспечить корректировку слов, присутствующих в словаре, но некорректно примененных в текущем контексте.

Наличие в откорректированных результатах распознавания списка дополнительных корректировок, помимо наилучшей корректировки, предоставляет возможность включить их в поисковый индекс, для увеличения полноты поисковой выдачи.

## Глава 3. Реализация системы распознавания архивных документов

### 3.1 Интеграция с электронным архивом

Разрабатываемая система распознавания должна учитывать особенности и отвечать требованиям основных процессов, протекающих в ЭА. Рассмотрим типовую компонентную модель построения ЭА, основанную на международном стандарте ISO 14721:2003 [66], с входящей в ее состав подсистемой массового распознавания документов. На рисунке 3.1 представлена схема организации и взаимодействия основных компонентов электронного архива, вовлеченных в процесс массового потокового распознавания.



3.1. Схема организаций компонентов электронного архива

На вход генератора входящего информационного пакета (ИП, submission information package [66,24]) поступают электронные образы архивных документов с этапа сканирования и от других источников информации. В процессе формирования входящего информационного пакета, поступающие образы снабжаются уменьшенными копиями, для предварительного просмотра и объединяются в группы. Каждая группа изображений помимо содержательной

информации и данных о ее физическом размещении содержит в себе описательную и справочно-поисковую информацию.

Далее сформированный входящий ИП подвергается форматно-логическому контролю и классификации: проверяются форматы представления информации, качество передаваемых изображений (разрешение, размеры, цветность и т.п.), достаточность описательной информации, принадлежность к тем или иным категориям и типам информации, хранимой в архиве. После прохождения проверок и классификаций входящий ИП преобразуется в набор архивных ИП, которые уже непосредственно предназначаются для размещения в оперативных базах данных электронного архива.

Таким образом, поступившие на хранение в ЭА данные сразу могут быть доступны для поиска и просмотра потребителю, но в большинстве случаев первоначальной справочно-поисковой информации оказывается недостаточно для организации высококачественного и оперативного обнаружения информации. Как следствие, полученный набор документов обрабатывается модулями разбора, ввода, распознавания и индексации для расширения поисковой базы, упорядочивания и увеличения вероятности быть найденным среди всей россыпи поступившей информации.

В задачи системы распознавания не входит построение идентичной электронной копии документа, которая бы могла использовать без прикрепленных к ней образов. В качестве поискового инструмента достаточным является привязка текстового представления к области изображения по заданным координатам. Данное условие существенно снижает требования к сложности реализации модуля оптического распознавания, необходимости проведения затратных по времени ручных верификаций и корректировок, а также позволяет избежать осуществления процедур детального анализа макета изображения.

Работа с распознанным текстом, неотделенным от изображений, в свою очередь снижает порог поступления документа в рабочую базу системы, за счет нестрогих требований к качеству распознавания (никакая информация не будет утеряна, так как всегда будет доступна для изучения непосредственно с образа).

Таким образом, можно приступать к работе с документом по достижению установленного порога качества распознавания, а, в последствие, после ручной верификации и корректировки лишь обновлять базу данных. Применение данной стратегии увеличивает скорость пополнения поисковой базы архива, что в свою очередь дает прирост производительности и скорости поиска документов.

После этапов разбора, распознавания и тщательной проверки администраторами ЭА, электронные версии документов могут группироваться в ИП и отправляться вновь на вход компонента приема данных, но уже не в качестве поисковых материалов, а как проверенные и готовые для размещения в долговременном хранилище документы.

### **3.2 Технология распознавания и корректировки результатов**

Пользователи системы подразделяются на 2 группы:

1. Эксперты, осуществляющие обработку всего корпуса документов.
2. Пользователи ЭА, осуществляющие поиск по проиндексированным результатам распознавания изображений документов. Данная группа пользователей не взаимодействует напрямую с системой распознавания, но должна быть описана для полноты восприятия.

Рассмотрим технологию обработки документов ЭА экспертом, представленную в виде последовательности действий на рисунке 3.2.

1. Вначале работы эксперту необходимо произвести анализ хранящихся электронных образов документов в ЭА. Документы следует подвергать анализу на предмет качества сканирования и принадлежности определенной тематике. В случае документов центральных государственных архивов тематику может определять специфика архива, принадлежность к определенному фонду, вид НСА.



### 3.2. Последовательность обработки документов ЭА экспертом

2. В результате анализа эксперт формирует несколько тематических групп документов и имеет представление о качестве электронных образов внутри этих групп. Для каждой из тематических групп должны быть отобраны тестовые образы и вручную введен текст, изображенный на них. Ручная

подготовка текстовых эталонов необходима для проведения оценки качества и точности распознавания.

3. Далее в задачу эксперта входит подготовка набора профилей, содержащих правила обработки, для первичного распознавания документов. В ходе этого процесса эксперт производит распознавание отдельных тестовых образов и оценивает качество распознавание по показателям точности, возвращаемым системой.
4. Следующим этапом является проведение сравнительного анализа результатов распознавания тестовых изображений. Для этого запускается процесс распознавания каждого изображения с каждым профилем.
5. Результаты сравнительного анализа позволяют выбрать наиболее подходящий профиль для распознавания отдельной группы изображений.
6. Далее эксперту необходимо произвести запуск пакетного распознавания всех документов каждой тематической группы. Данный процесс может продолжаться от нескольких часов до нескольких суток в зависимости от количества документов.
7. После окончания процесса распознавания эксперт может произвести запуск процесса построения структур данных, необходимых для процедуры автоматической корректировки результатов. Структуры данных должны быть сгенерированы для каждой тематической группы в отдельности. Структуры данных строятся по всему корпусу распознанных документов группы, но могут быть дополнительно расширены путем добавления к результатам распознавания перечня текстов, относящихся к той же тематической группе.

При запуске корректировки эксперту необходимо выбрать набор тематических тезаурусов и словарей, которые будут использованы для формирования корпусного тезауруса. Тезаурусы могут быть загружены как из предустановленного списка, так и вручную.

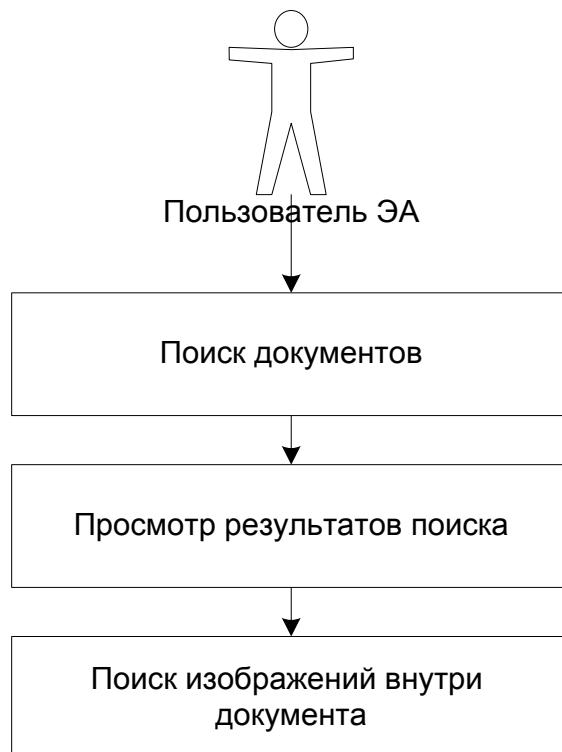
Повторные перестроения структур данных с новыми пороговыми значениями или наборами тематических тезаурусов можно производить на

основе ранее сформированных структур, что позволяет избежать длительного процесса чтения и разбора исходных результатов распознавания и текстов.

8. Последними действиями являются настройка профилей для корректировки результатов распознавания, их сравнительный анализ, выбор наиболее эффективного и запуск процесса автоматической корректировки.

В результате подготовительной работы проведенной экспертом при помощи инструментария, обеспечивающего настройку системы, и проведения процессов распознавания и корректировки, распознанный корпус документов подвергается полнотекстовому индексированию.

Пользователю ЭА предоставляется механизм поиска по распознанным изображениям документов (рисунок 3.3).



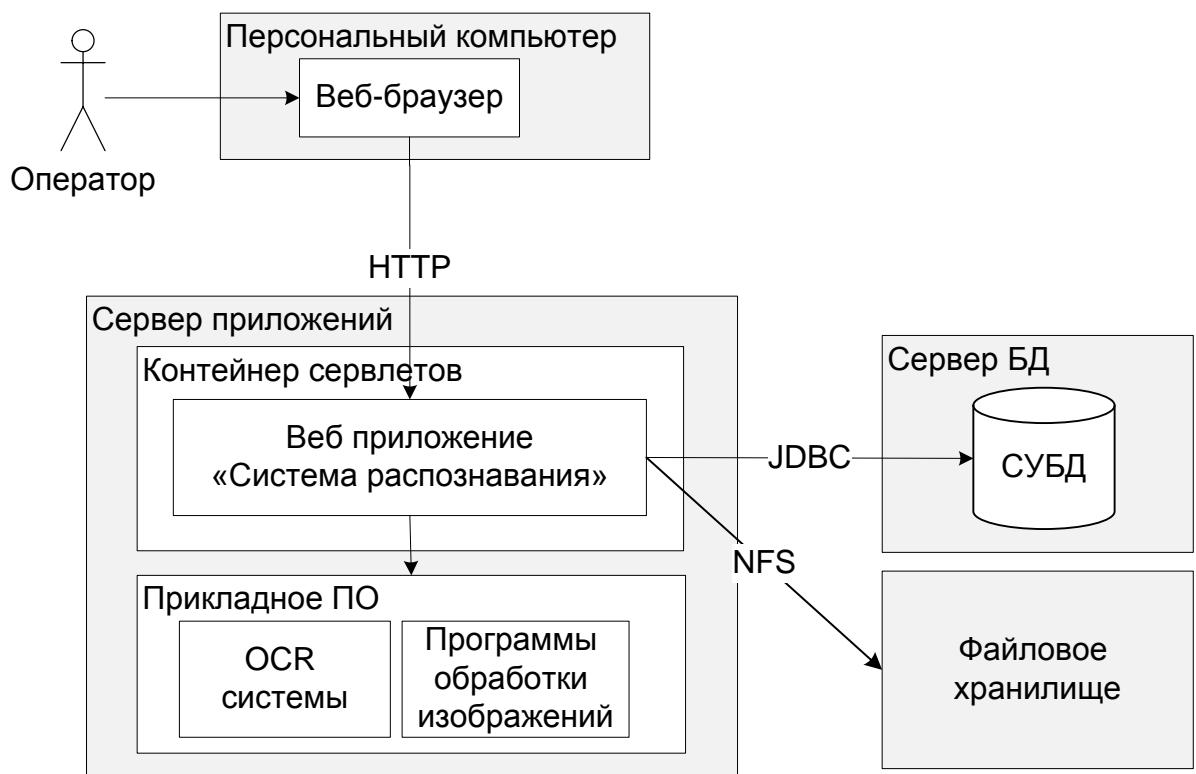
3.3. Поиск документов пользователем ЭА

Через графический интерфейс системы ЭА пользователь может произвести сквозной поиск архивных документов, изображения которых содержат искомую фразу. Далее при детальном просмотре изображений пользователю

предоставляется возможность уточненного поиска отдельных изображений внутри документа.

### 3.3 Архитектура и компонентная модель системы

Архитектура разработанной системы, изложенная на рисунке 3.4, построена по классической трёхзвенной модели [30].



3.4. Архитектура системы

Программная реализация системы состоит из веб-приложения «Система распознавания», набора прикладных программ и утилит, базы данных.

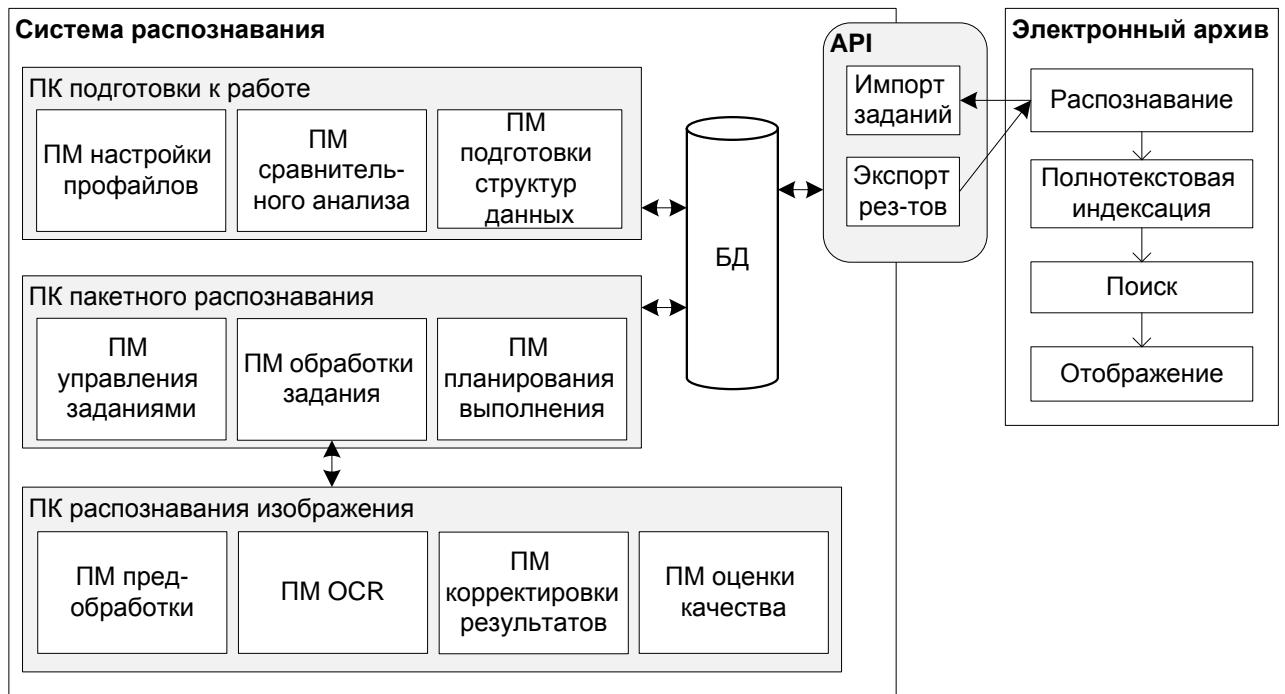
Работа с системой осуществляется через веб приложение, разработанное на языке программирования Java [67] с использованием Spring Framework [104]. Приложение устанавливается в виде WAR [78] файла на контейнер сервлетов [78] (Glassfish [56], Apache Tomcat [42], JBoss [69] и т.п.).

БД системы может быть развернута на СУБД, поддерживающей реляционную модель хранения данных [16] (Oracle Database [87], PostgreSQL [93], MySQL [80] и т.п.).

Вызов прикладных программ осуществляется через программную оболочку веб-приложения, что позволяет производить увеличение вычислительной мощности за счет горизонтального масштабирования [72] серверов приложений.

В основе программной реализации системы лежит свободно распространяемое ПО, что делает систему потенциально более доступной для применения в других проектах.

Компонентная модель разработанной системы представлена на рисунке 3.5.



3.5. Компонентная модель системы

Разработанная система состоит из трех программных комплексов (ПК) связанных между собой единой БД и программного интерфейса для взаимодействия с внешними подсистемами:

1. ПК подготовки к работе предназначен для ручного тестирования процесса распознавания на единичных изображениях. В результате тестирования для разных типов изображений создаются профили, содержащие в себе настройки каждого этапа распознавания и общую технологическую схему обработки. После этого необходимо произвести сравнительный анализ созданных профилей на различных наборах данных и выявить наиболее подходящий.

Также в задачи подготовки к работе входит предварительный разбор всего корпуса распознанных текстов и построение структур данных, необходимых для автоматической корректировки ошибок.

2. ПК пакетного распознавания предназначен для управления ходом выполнения заданий на обработку изображений документов. Его основными задачами являются: предоставление возможности просмотра журнала заданий, управление приоритетами заданий, вызов процедур распознавания отдельных изображений, сбор результатов и запись их в БД.
3. ПК распознавания изображения отвечает за процесс обработки отдельного изображения в соответствии с заданной в профиле технологической схемой.
4. Программный интерфейс системы предоставляет ряд API вызовов для постановки на распознавание отдельных документов ЭА, опроса состояния и получения результатов.

### **3.4 ПК подготовки к работе**

ПК подготовки к работе состоит из трёх программных модулей (ПМ), рассмотренных ниже.

#### **3.4.1 ПМ настройки профилей**

Через графический интерфейс данного модуля оператор системы (эксперт) может производить настройку процесса распознавания различных групп документов.

Система предоставляет возможность группировать отобранные тестовые изображения в произвольной иерархии и прикреплять эталонные тексты к ним.

Помимо подготовки наборов изображений для тестирования данный модуль предоставляет графический интерфейс для настройки параметров распознавания. Управляя набором параметров каждого этапа обработки изображения, эксперт может наблюдать за изменениями результата.

Результат отображается в отдельной области экрана, через которую можно просмотреть:

- исходное изображение;
- результат предобработки изображения;
- результат, полученный на выходе OCR системы;
- результат корректировки с подсвеченными словами, которые система определила как ошибочные, и варианты их замены;
- финальный результат распознавания в формате XML;
- таблицу с вычисленными критериями оценки качества результата распознавания до и после этапа корректировки;
- детальный лог хода обработки изображения.

Для сохранения подходящих параметров распознавания в ПМ имеется возможность создать профиль, представляющий из себя строку свойств вида:

```
<параметр>=<значение>
...
<параметр>=<значение>
```

### **3.4.2 ПМ сравнительного анализа**

Подготовка и выбор профиля, для обработки определенной группы документов, является непростой задачей. Для объективной оценки профиля необходимо проводить анализ точности результатов распознавания на множестве изображений. В дополнение к этому желательно иметь результаты сравнения точности нескольких профилей.

Программный модуль предоставляет возможность запустить процесс пакетного распознавания выбранных наборов тестовых изображений  $N$  с различными профилями  $P$ . После окончания пакетной обработки на экране будет отображаться сводная таблица результатов распознавания  $T = N \times P$ , содержащая усредненные значения критериев оценки точности распознавания набора  $n \in N$  с профилем  $p \in P$ . Пример сводной таблицы представлен на рисунке 3.6.

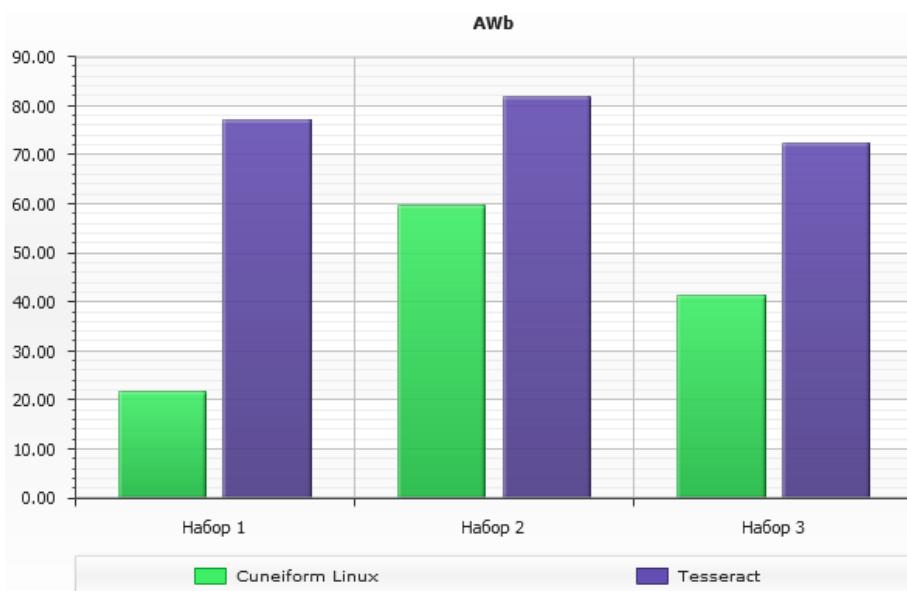
Набор...	Профайл	За...	tOCR	stage	Tw	Tc	rTw	rTc	AD	AC	AW	AWb
Набор 1	Cuneiform	69...	0:00:06	postcorre...	185	954	69.29 %	52.88 %	64.32 %	40.74 %	4.49 %	21.72 %
Набор 1	Tesseract	69...	0:00:04	postcorre...	226	1630	84.64 %	90.35 %	80.97 %	89.08 %	76.78 %	77.15 %
Набор 2	Cuneiform	69...	0:00:08	postcorre...	358	2208	92.03 %	79.03 %	67.32 %	74.55 %	47.56 %	59.64 %
Набор 2	Tesseract	69...	0:00:05	postcorre...	375	2623	96.40 %	93.88 %	75.73 %	91.12 %	77.63 %	81.75 %
Набор 3	Cuneiform	69...	0:00:11	postcorre...	257	1478	88.01 %	67.15 %	59.92 %	57.25 %	29.45 %	41.44 %
Набор 3	Tesseract	69...	0:00:06	postcorre...	292	1997	100.00 %	90.73 %	72.26 %	84.55 %	64.04 %	72.26 %

3.6. Пример сводной таблицы результатов сравнительного анализа

Также результаты сравнительного анализа отображаются в графическом представлении, а именно в виде столбиковой диаграммы, построенной по трем измерениям:

1. набор изображений,
2. профиль,
3. критерий оценки.

Эксперту предоставляется возможность выбора проекции построения. Пример графического представления результатов проиллюстрирован на рисунке 3.7.



3.7. Пример графического представления результатов сравнительного анализа

Описанный модуль позволяет эксперту проводить качественное сравнение результатов распознавания несколькими профилями, и определять для распознавания какой группы документов их лучше применять.

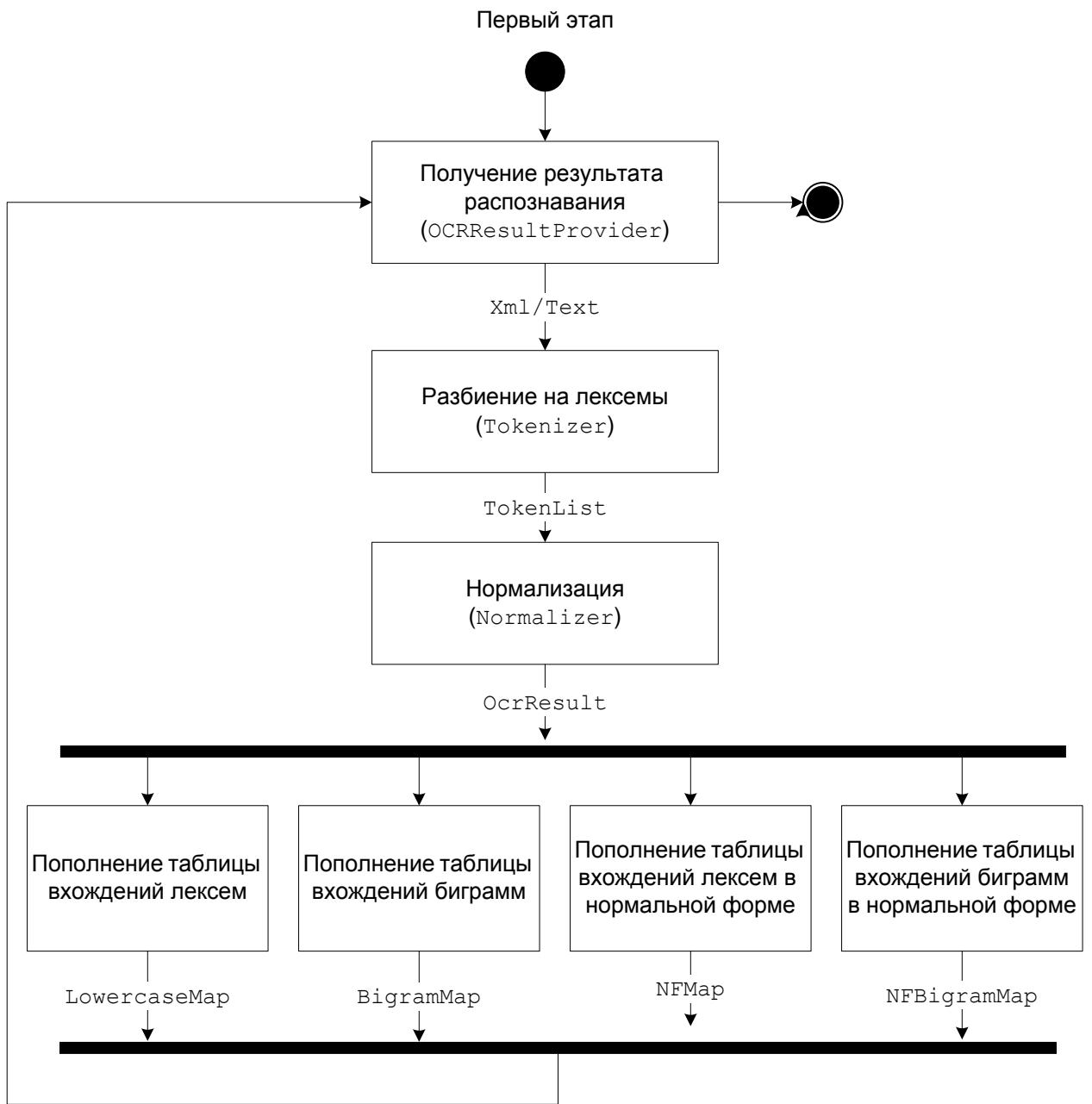
### **3.4.3 ПМ подготовки структур данных**

Программный модуль реализует задачу подготовки структур данных для корректировки ошибок распознавания, формализованную в главе 2.3.

Схематично процесс подготовки структур данных представлен на рисунках 3.8 и 3.9.

Структуры данных для корректировки собираются на основе результатов распознавания документов определенной тематической группы, так как основной

задачей ставится сбор статистической информации о встречающихся словах и их последовательностях. Если для подготовки структур данных использовать тексты произвольных тематик, множества слов которых практически не пересекаются, то собранные статистические данные будут малодостоверны.



3.8. Подготовка структур данных для корректировки. Первый этап.

На вход ПМ передаются параметры, указывающие на источник предварительно распознанных документов одного тематического направления, в случае архивных документов это могут быть описи, указатели, дела определенных фондов, текст путеводителя и т.п.

В зависимости от полученного источника, ПМ используя шаблон фабричного метода [8] создает экземпляр класса `OCResultProvider`, отвечающего за предоставление результатов распознавания из хранилища данных указанного источника.

Первый этап процесса подготовки структур данных заключается в циклической обработке всего набора результатов распознавания, имеющегося в источнике (рисунок 3.8).

1. Из источника выбирается очередной результат распознавания в том формате, в котором он был получен на выходе OCR системы.

2. Далее полученный результат проходит этап разбиения на лексемы. Причем для каждого из возможных форматов используется один из классов `HOCRTokenizer` или `PlainTextTokenizer`, реализующих интерфейс `Tokenizer`.

Класс `HOCRTokenizer` применяется для разбиения на лексемы результата в формате hOCR [109]. Данный формат основан на языке гипертекстовой разметки HTML. Строки определяются по тэгу `span` со значением атрибута `class=ocr_line`, лексемы же отделяются друг от друга тегом `span` со значением атрибута `class=ocrx_word`. Пример результата распознавания в формате hOCR генерированного OCR системой Tesseract представлен ниже:

```
<?xml version="1.0" encoding="UTF-8"?>
<head>
    <title/>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name='ocr-system' content='tesseract 3.02.02' />
    <meta name='ocr-capabilities' content='ocr_page ocr_carea
ocr_par ocr_line ocrx_word' />
</head>
<body>
    <div class='ocr_page' id='page_1' title='image
"/storage/ocrarena_workdir/tesseract/052825be-356d-4fff-a358-
1c92f1c34c78/200-image390363.jpg"; bbox 0 0 1682 1605; pageno 0'>
        <div class='ocr_carea' id='block_1_1' title="bbox 50 60 1472
186">
            <p class='ocr_par' dir='ltr' id='par_1' title="bbox 50 61
1472 186">
                <span class='ocr_line' id='line_1' title="bbox 59 61 1449
113">
                    <span class='ocrx_word' id='word_1' title="bbox 59 61
119 104">Об</span>

```

```

<span class='ocrx_word' id='word_2' title="bbox 158 67
459 103">изменени</span>
    <span class='ocrx_word' id='word_3' title="bbox 501 66
906 102">наименования</span>
        <span class='ocrx_word' id='word_4' title="bbox 946 68
1449 113">партнерганизации</span>
    </span>
</p>
</div>
</div>
</body>
</html>

```

Класс `PlainTextTokenizer` отвечает за формирование лексем из результатов распознавания в текстовом формате. Для разбиения строк используется разделитель «`\r?\n`» (формат регулярного выражения Java [89]). Для разбиения строк на лексемы — разделитель «`[\\p{Zs}\\t\\\\(\\)\\\\{\\}\\\\'\"\\\\%\\\\.\\\\,\\\\&#!?=;<>/]`».

3. Полученный набор лексем далее подвергается процедуре нормализации, описанной в главе 2.3.1 (класс `Normalizer`).

С начала и конца каждой лексемы отделяются все некириллические символы, используя регулярное выражение:

```
«(^[^\p{IsCyrillic}]*) (.*) ([^\p{IsCyrillic}]*$)».
```

Далее все варианты написания символа тире «`\p{Pd}`» заменяются символом «`-`», и все пробельные символы «`\p{Zs}`» заменяются символом обычного пробела.

В приведенных выше регулярных выражениях используются классы символов стандарта Unicode [95]: `\p{IsCyrillic}`, `\p{Pd}`, `\p{Zs}`.

После этого производится корректировка возможных ошибок расстановки переносов. Ошибочно перенесенные на разные строки лексемы объединяются по эвристическому правилу, описанному в главе 2.3.1.

В результате нормализации формируется объект класса `OcrResult`, содержащий результаты распознавания в формате пригодном для последующей обработке системой полнотекстовой индексации.

4. По нормализованному результату распознавания производится сбор статистической информации о вхождениях лексем (класс `LowercaseMap`), биграмм лексем (класс `BigramMap`), лексем в нормальной форме (класс `NFMap`) и биграмм лексем в нормальной форме (класс `NFBigramMap`). Описание классов приведено в таблице 3.1.

Таблица 3.1. Описание классов

Класс	Родительский класс	Описание
<code>LowercaseMap</code>	<code>HashMap&lt;String, Integer&gt;</code>	<code>String</code> — лексема <code>Integer</code> — кол-во вхождений
<code>BigramMap</code>	<code>HashMap&lt;List&lt;String&gt;, Integer&gt;</code>	<code>List&lt;String&gt;</code> — биграмма лексем <code>Integer</code> — кол-во вхождений
<code>NFMap</code>	<code>HashMap&lt;String, Integer&gt;</code>	<code>String</code> — лексема в нормальной форме <code>Integer</code> — кол-во вхождений
<code>NFBigramMap</code>	<code>HashMap&lt;List&lt;String&gt;, Integer&gt;</code>	<code>List&lt;String&gt;</code> — биграмма лексем в нормальной форме <code>Integer</code> — кол-во вхождений

Первый этап подготовки структур данных заканчивается после обработки всех имеющихся результатов распознавания. Далее начинается второй этап обработки (рисунок 3.9).

1. Первым шагом является сокращение таблицы вхождений лексем (класс `LowercaseMap`) и таблицы вхождений биграмм лексем (класс `BigramMap`) по пороговым значениям  $\alpha$  и  $\beta$  соответственно.

В итоге формируются урезанные таблицы `PrunedLowercaseMap` и `PrunedBigradMap`, содержащие лексемы и биграммы лексемы, число повторений которых больше чем пороговые значения.

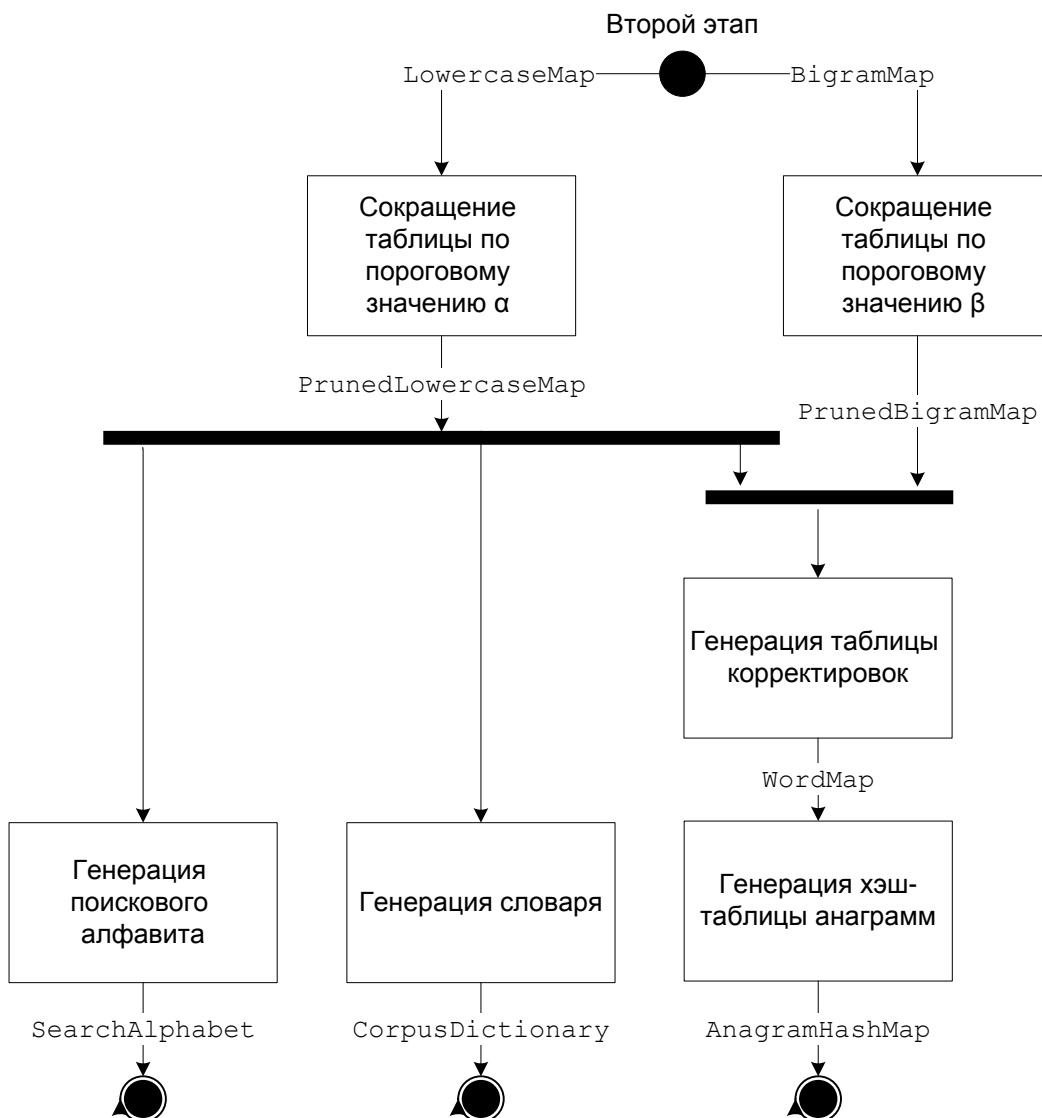
2. Далее на основе `PrunedLowercaseMap` генерируется поисковый алфавит `SearchAlphabet`.

Из каждой лексемы таблицы `PrunedLowercaseMap` извлекаются последовательности одиночных, сдвоенных и строенных символов, состоящих только из символов «абвгдежзийклмнопрстуфхцшъыэюя -». Для каждой последовательности вычисляется анаграмм-хэш, который помещается в объект класса `SearchAlphabet`.

3. В словарь CorpusDictionary помещаются все лексемы из PrunedLowercaseMap, которые состоят из кириллических символов, символа тире или пробела (`<[\p{IsCyrillic}\p{Pd}\p{Zs}]>`) и содержатся в словаре.

4. Список лексем WordMap, в котором будет производиться поиск корректировок, формируется путем объединения лексем из PrunedLowercaseMap и биграмм лексем из BigramMap. Биграммы объединяются через символ пробела.

5. Последней генерируемой структурой является хэш-таблица анаграмм AnagramHashMap. Каждый элемент списка WordMap помещается в хэш-таблицу, в качестве ключа используется вычисленный анаграмм-хэш.



3.9. Подготовка структур данных для корректировки. Второй этап.

Описание классов второго этапа подготовки структур представлено в таблице 3.2.

Таблица3.2. Описание классов

<b>Класс</b>	<b>Родительский класс</b>	<b>Описание</b>
SearchAlphabet	HashSet<Long>	Long — значение хэша символьной последовательности
CorpusDictionary	HashSet<String>	String — слово, прошедшее словарную проверку
WordMap	HashMap<String, Integer>	String — лексема Integer — кол-во вхождений
AnagramHashMap	HashMap<Long, Set<String>>	Long — значение хэша Set<String> — список лексем

### 3.5 ПК пакетного распознавания

Программный комплекс предназначен для упорядочивания процесса массовой обработки документов.

Все изображения архива разбиваются по группам (пакетам), где каждый пакет соответствуетциальному документу. Далее для каждого пакета создается отдельное задание (класс Task), которое обрабатывается в порядке своей очередности.

Задание состоит из набора изображений и профиля распознавания, также каждому заданию назначается приоритет обработки.

#### 3.5.1 ПМ управления заданиями

Программный модуль предоставляет графический интерфейс для просмотра списка созданных заданий.

В данном модуле оператор может:

- отслеживать статус выполнения заданий;
- управлять приоритетами и порядком обработки;
- просматривать промежуточные и финальные результаты;
- производить поиск заданий по идентификатору, наименованию профиля, статусу обработки, результату выполнения.

### **3.5.2 ПМ планирования выполнения**

Ключевым элементом данного модуля является планировщик выполнения заданий (класс `BatchRecognitionScheduler`). Главной задачей планировщика является выбор наиболее приоритетного задания и запуск процесса его обработки.

Обработка заданий выполняется в многопоточном режиме и может быть масштабирована на произвольное количество серверов.

Для балансировки нагрузки в системе помимо количества серверов устанавливается максимальное пороговое количество одновременно обрабатываемых заданий на каждом сервере (предпочтительно задавать это значение равным количеству ядер процессора).

Экземпляр класса `BatchRecognitionScheduler` в фоновом режиме производит опрос базы данных на наличие новых заданий, из БД выбирается самая приоритетная задача и запускается процесс ее распознавания на наименее загруженном сервере. Если все сервера заняты, то задание ожидает освобождения ресурсов.

### **3.5.3 ПМ обработки задания**

Центральным компонентом данного модуля является класс `RecognitionServiceImpl`, который обеспечивает выполнение задания по строго указанной в профиле схеме обработки.

В общем случае процесс обработки состоит из следующих шагов:

1. Предобработка и оптическое распознавание всех изображений задания.
2. Вычисление критериев оценки качества результатов распознавания для каждого изображения в отдельности и для всего задания суммарно.
3. Корректировка результатов распознавания.
4. Повторное вычисление критериев оценки качества после корректировки ошибок распознавания для каждого изображения в отдельности и для всего задания суммарно.

На каждом шаге обработки производится детальное журналирование времени выполнения, промежуточных результатов и причин возникновения исключительных ситуаций.

### **3.6 ПК распознавания изображения**

Процесс распознавания разделяется на четыре уровня: предварительная обработка изображения, оптическое распознавание символов, постобработка результатов оптического распознавания и оценка точности проведенного распознавания. Программная реализация каждого из уровней выполнена в виде обособленных автономных модулей.

Каждый модуль implements базовые интерфейсы программного комплекса и может иметь несколько вариантов реализации. Выбор модуля для обработки конкретного изображения производится динамически в процессе обработки.

ПМ обработки задания производит обращение к программным модулям распознавания через вызовы базовых сервисов. Такая сервис ориентированная организация основных вычислительных модулей обеспечивает возможности расширения, масштабируемости, подключения новых модулей и простоты управления вычислительными ресурсами системы, что является немаловажным фактором для дальнейшего развития системы и возможности ее адаптации к различным средам эксплуатации и развертывания.

Рассмотрим более детально каждый из уровней обработки.

#### **3.6.1 ПМ предобработки**

Программный модуль предобработки предоставляет набор сервисов подготовки изображения к процессу извлечения текста.

Под подготовкой подразумевается проведение операций по устранению «шума», сглаживанию фоновых структур, увеличению резкости изображения, выравниванию угла наклона, удалению «темных» и поврежденных областей в местах переплета, бинаризации и т.п. [13,55,62].

Параметры для каждой операции устанавливаются в профиле, передаваемом на вход программного модуля вместе с изображением.

Реализация основана на последовательных вызовах программных средств библиотеки ImageMagick [64].

### **3.6.2 ПМ OCR**

Программный модуль оптического распознавания символов отвечает за перевод изображений текста в машиночитаемую и редактируемую форму.

В общем случае, принципы работы OCR основываются на структурном анализе изображений и разбиении этих изображений на более мелкие участки с целью обнаружения текстовых зон [4]. Среди этих зон OCR идентифицирует отдельные строки текста, а среди строк выделяет отдельные слова и символы. Далее происходит поиск совпадения найденного символа с символами из предустановленного набора шрифтов и так далее для всех символов слова. После этого полученное слово подвергается словарной проверке, с целью обнаружения совпадающих кандидатов [38]. Таким образом обрабатывается весь текстовый отрезок изображения и в результате формируется результат распознавания в виде файла с описанием координат текстовых регионов, распознанным вариантом и дополнительными реквизитами, такими как степень достоверности отдельных слов и символов.

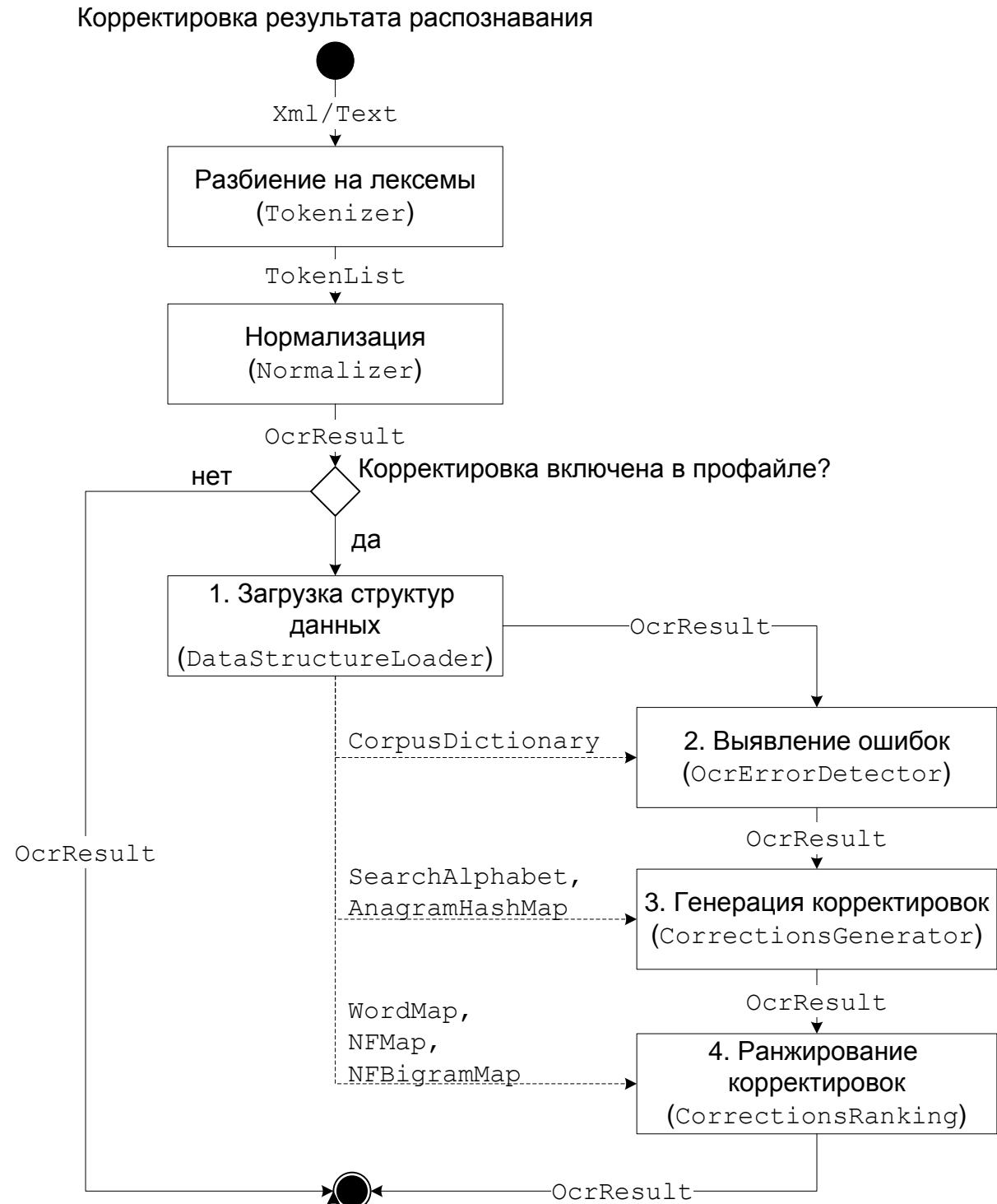
Программная реализация представляет собой набор классов реализующих интерфейс `OCRService`. Каждый класс является Java оболочкой над OCR системой и осуществляет делегирование операции распознавания конкретной системе.

Были реализованы Java оболочки для вызова следующих систем распознавания: “Tesseract”, “Cuneiform Linux”, “Cuneiform Windows”, “Abbyy Finereader”, “Nuance Omnipage”, “IRIS Readiris ”.

### **3.6.3 ПМ корректировки результатов**

После прохождения этапа оптического распознавания результаты поступают на вход программного модуля корректировки, где они проходят

процедуры разбиения на лексемы и нормализации, подробно описанные в главе 3.4.3. После этого проверяется необходимость проведения корректировки ошибок на основе рейтинго-ранговой модели текста и осуществляется дальнейшая обработка. Алгоритм работы программного модуля изображен на рисунке 3.10



3.10. Корректировка результата распознавания.

1. Если проверка настроек профиля, показала необходимость корректировки, то производится загрузка в оперативную память структур данных, подготовленных ранее (см. главу 3.4.3).

Загрузка в оперативную память производится единожды при первом обращении. Стоит отметить, что процесс загрузки может занимать продолжительный период времени. Для предотвращения проблем с конкурирующими запросами на загрузку данных данный программный код выделяется в синхронизированную область.

2. Далее производится поиск ошибочных лексем в результате распознавания `OcrResult`. Формализованное описание данного процесса приведено в главе 2.4.

Если длина лексемы меньше порога  $\varphi$  или лексема присутствует в словаре `CorpusDictionary`, то лексема считается корректной в ином случае — ошибочной.

Далее из ошибочной лексемы удаляются все символы, соответствующие регулярному выражению «`[\^\p{IsCyrillic}\p{Pd}\p{Zs}]`» и она проходит повторную проверку по словарю. Если в словаре находится соответствие, то сгенерированный вариант лексемы добавляется в список корректировок.

3. Далее производится поиск корректировок по алгоритму, приведенному в главе 2.4.

В результате каждой лексеме добавляется список корректировок (класс `ArrayList<Correction>`).

4. После отбора корректировок для всех ошибочных лексем запускается процедура ранжирования. Главной задачей ставится вычисление ранга для каждой корректировки. Ранг должен отражать степень применимости корректировки для замены ошибочной лексемы. Формализованное описание правил ранжирования приведено в главе 2.5.

На первом этапе каждой корректировке `Correction` присваивается значение переменной `score`, отражающей степень схожести корректировки и исходной лексемы. Далее список корректировок сокращается до `n` значений с наибольшим значением `score`.

На втором этапе происходит вычисление финального ранга. Для этого определяется вероятность появления корректировки вместе с предшествующей по тексту лексемой. Если предшествующая лексема тоже ошибочная, то вычисляется вероятность для всех ее вариантов корректировки. Для вычисления вероятности используются ранее подготовленные таблицы частоты вхождений лексем (класс `NfMap`) и биграмм лексем (класс `NFBigramMap`) в нормальной форме.

Если текущая и предшествующая корректировки состоят более чем из одного слова, то вероятность рассчитывается для слов, расположенных ближе к началу и концу корректировки соответственно.

После упорядочивания списка корректировок по значению финального ранга, производится выбор наилучшей корректировки и формирование списка дополнительных корректировок, по правилам, описанным в главе 2.6.

В дальнейшем при индексировании результатов распознавания могут быть использованы как наилучшие, так и дополнительные корректировки.

### **3.6.4 ПМ оценки качества**

Последним этапом обработки является проведение оценки точности распознавания.

Если к распознанному изображению был прикреплен эталонный текст, то ПМ вычисляет весь набор критериев оценки. Если же эталонного текста нет, то вычисляются только критерии, которые не требуют наличия эталонного текста. Подробное описание критериев оценки качества распознавания приводится в главе 4.

Вычисление коэффициентов ошибок является не тривиальной операцией. Рассмотрим пример: «молоко» — слово в эталоне, «ллолоко» — слово в результатах. Можно предположить, что количество ошибочных символов равно 7, а коэффициент ошибок =  $7/6 = 1,16$ . Однако оба слова имеют общую последовательность символов «олоко». Следовательно, достаточно удалить первый символ «л» и заменить второй символ «л» на символ «м», что будет составлять 2 операции и иметь коэффициент = 0,33.

### 3.7 Инструментарий настройки и выбора наиболее эффективной конфигурации

Программные модули настройки профилей и сравнительного анализа представляют в своей совокупности инструментарий, позволяющий эксперту ограничивать пространство конфигураций для наиболее эффективного решения поставленных задач.

Определим все множество параметров конфигураций  $\Omega$ :

$$\Omega = \Omega_{PRE} \cup \Omega_{OCR} \cup \Omega_{POST} \cup \Omega_{QA},$$

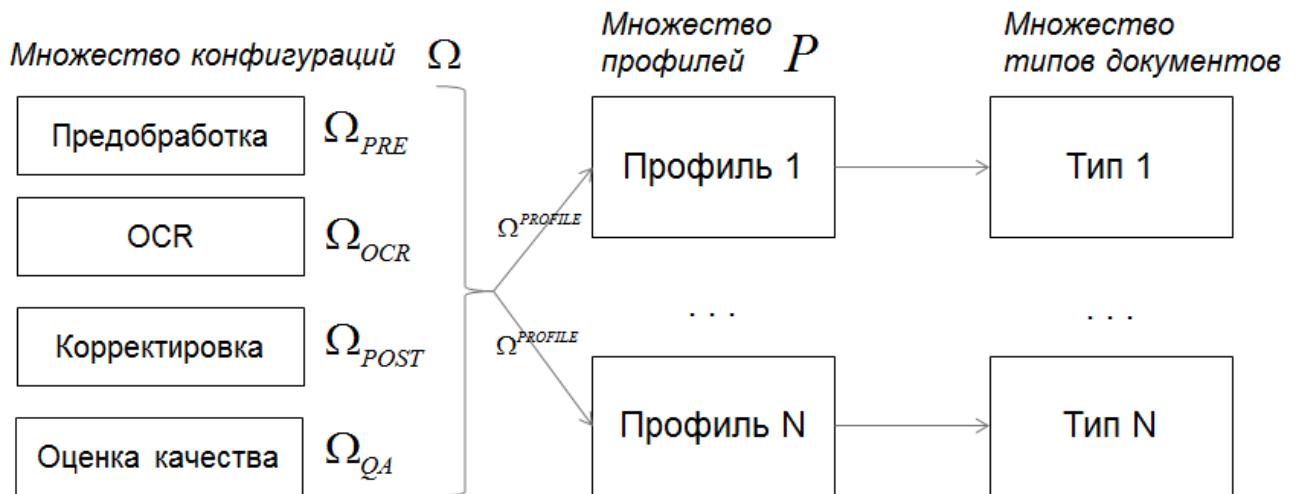
где  $\Omega_{PRE}$  — множество параметров настройки стадии предварительной обработки изображения;  $\Omega_{OCR}$  — множество параметров настройки стадии оптического распознавания изображения;  $\Omega_{POST}$  — множества параметров настройки стадии автоматической корректировки результатов распознавания изображения;  $\Omega_{QA}$  — множества параметров настройки стадии оценки качества итогового результата распознавания.

Задачей эксперта на этапе настройки профилей является формирование множества профилей  $P$  (рисунок 3.11):

$$P = \{ p \mid p = \Omega^{PROFILE}, \Omega^{PROFILE} \subset \Omega \},$$

где  $p$  — профиль, содержащий множество допустимых параметров из всего множества параметров пространства конфигураций, наиболее подходящий для распознавания отдельных изображений.

Используя ПМ сравнительного анализа, эксперт может определить профиль, который наиболее эффективно решает задачу распознавания группы изображений. Для определения эффективности эксперту предоставляются рассчитанные значения критериев оценки качества и точности результатов распознавания  $p$ .



3.11. Иллюстрация процесса настройки профилей

### 3.8 Морфологический анализ

Задача приведения слов к нормальной форме решается классом Morpholizer. Данный класс использует библиотеку «RussianMorphology for Lucene» [101], основанную на системе морфологического анализа «АОТ» [1].

Используемый в системе русский морфологический словарь базируется на грамматическом словаре Зализняка А.А. и включает 161 тысячу лемм.

Главной особенностью данной библиотеки является поиск нормальных форм для словоформ необнаруженных в словаре, на основе морфологических предсказаний.

Одной словоформе может соответствовать много морфологических интерпретаций. Например, у словоформы «СТАЛИ» две интерпретации: {СТАЛЬ, существительное} и {СТАТЬ, глагол}.

### 3.9 API системы

Взаимодействие системы с внешними приложениями осуществляется через API интерфейс, предоставляющий следующие функции:

1. Импорт задания для распознавания.
2. Проверка статуса выполнения задания по идентификатору.
3. Экспорт результатов распознавания задания.

### 3.10 Выводы по третьей главе

Электронный архив, с включенной в его состав подсистемой массового автоматического распознавания документов, обладает рядом существенных преимуществ над архивными системами, в которых распознавание либо отсутствует, либо осуществляется вручную. Данными преимуществами являются: высокие темпы перевода документов в электронную форму, возможности автоматического построения эффективного поискового аппарата, высокая скорость поиска и доступа к электронным образам документов.

Основной особенностью разработанной системы является гибкая архитектура, позволяющая подключать различные коммерческие и свободно распространяемые OCR системы и библиотеки предобработки изображений.

Система может быть настроена на распознавание документов различных категорий качества. Для одной категории потребуется подключение дорогостоящих движков распознавания, для другой хорошие результаты будет выдавать бесплатная OCR система.

Преимуществом системы является наличие процедур автоматической корректировки ошибок распознавания, позволяющих выявлять и исправлять ошибки даже в текстах, изобилующих специфическими терминами, именами собственными, узкоспециализированным лексиконом. Это особенно важно для исторических, архивных документов.

В связи с тем, что главной чертой систем массового распознавания является сверхбольшой объем документов и отсутствие возможности произвести проверку каждого документа вручную, важнейшим процессом, реализованным в системе, является автоматическое определение критериев качества результатов распознавания. Наличие такой оценки позволяет установить определенную шкалу градации, по которой будут определяться дальнейшие варианты использования документа.

Инструментарий для выбора наилучшей конфигурации для распознавания определенной группы архивных документов позволяет эксперту производить настройку системы и проводить сравнительный анализ достоверности

распознавания документов различными профилями. Причем выбор наиболее подходящего профиля, основывается на анализе широкого спектра автоматически рассчитываемых критериев и показателей качества и точности.

В заключение стоит отметить, что система разработана в виде автономного программного комплекса и может быть интегрирована с другими информационными системами.

Примеры графического интерфейса программных модулей разработанной системы представлены в приложении А.

## Глава 4. Апробация системы и оценка полученных результатов

### 4.1 Условия проведения испытаний

Апробация разработанной системы распознавания производилась в составе государственной информационной системы (ГИС) «Государственные архивы Санкт-Петербурга» [21]. Основной задачей ГИС является автоматизация семи центральных государственных архивов и Архивного комитета Санкт-Петербурга с целями повышения эффективности организации архивного дела города, повышения качества и сокращения сроков оказания государственных услуг, содействия обеспечению сохранности документов.

В ходе эксплуатации ГИС регулярно производится оцифровка бумажных документов, полученные электронные образы загружаются в подсистему хранения информации. Объем хранящихся документов измеряется несколькими миллионами изображений. Разработанная система массового оптического распознавания (подсистема распознавания) использовалась для распознавания накопленного массива изображений документов, полученные результаты распознавания в дальнейшем были проиндексированы и использованы в системе полнотекстового поиска.

ГИС представляет собой распределенную систему: в каждом архиве, установлен отдельный экземпляр подсистемы «Автоматизированное рабочее место архивиста» (АРМ).

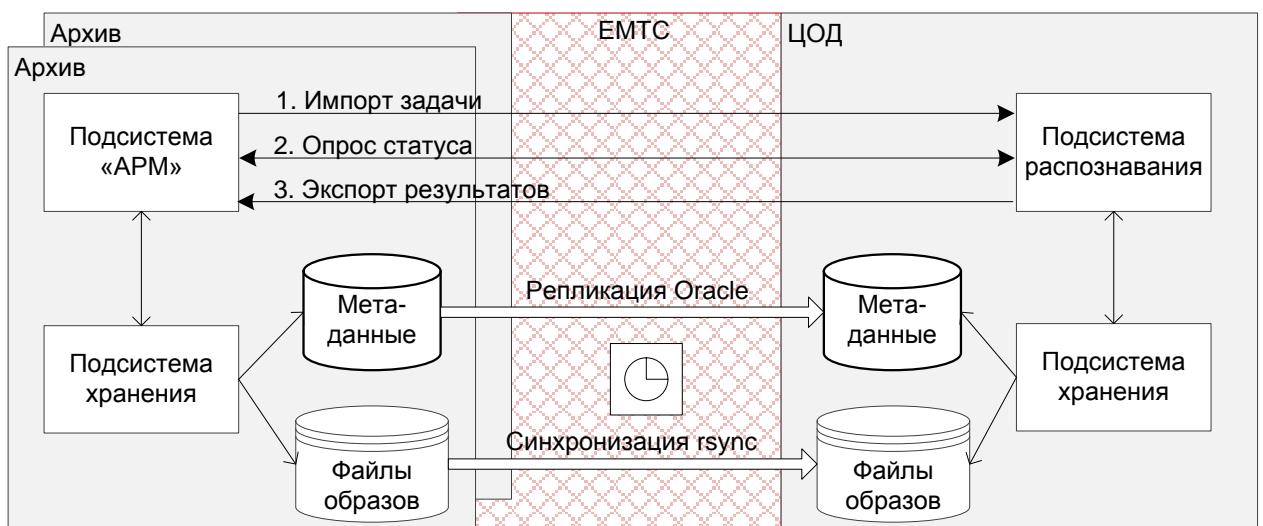
Подсистема распознавания была развернуты на базе созданного в Санкт-Петербурге центра обработки данных (ЦОД). Передача данных между ЦОД и архивами осуществляется по защищенным каналам связи единой мультисервисной телекоммуникационной сети (ЕМТС) [19].

Описание серверов, на которых была развернута подсистема распознавания, представлено в таблице 4.1.

## 4.1. Сервера подсистемы распознавания

Наименование	Системные характеристики	Назначение
Сервер приложений / распознавания	ОС: Debian 6 ОП: 32Gb CPU: Intel(R) Xeon(R)@2.40GHz 32 cores	Функционирование веб-приложения «Система распознавания», выполнение всех этапов обработки изображений
Сервер распознавания	ОС: Windows 2008 ОП: 16Gb CPU: Intel(R) Xeon(R)@2.40GHz 16 cores	Распознавание изображений OCR системами, разработанными под ОС Windows
Сервер базы данных	ОС: Oracle Linux 6.3 ОП: 16Gb CPU: QEMU Virtual CPU @3Ghz 4 cores	Хранение результатов распознавания и всех данных подсистемы распознавания

Схема взаимодействия подсистем представлена на рисунке 4.1



4.1. Схема взаимодействия подсистем

Подсистема хранения обеспечивает централизованную репликацию метаданных (реквизиты документов) и копирование файлов электронных образов документов со всех серверов в единое хранилище в ЦОДе по расписанию. Полная копия всех электронных документов архивов в ЦОДе служит резервной копией и дополнительно предоставляет возможность обрабатывать изображения без необходимости повторной передачи данных по сети, что минимизирует нагрузку на сеть и увеличивает скорость обмена информацией.

При импорте задачи в подсистему распознавания из подсистемы «АРМ» передаются лишь идентификаторы изображений, далее подсистема распознавания обращается за файлами в подсистему хранения данных, расположенную в ЦОДе.

## **4.2 Описание корпуса документов**

В ходе апробации системы распознавания были обработаны изображения документов пяти центральных государственных архивов Санкт-Петербурга [5]:

- Центральный государственный архив Санкт-Петербурга (ЦГА).

Архив хранит дела, свидетельствующие о работе органов власти Ленинграда - Санкт-Петербурга, истории развития экономики, городского хозяйства, образования, здравоохранения, социальной защиты населения за 1917-2002 года. Многие документы содержат сведения о развитии соседних регионов — Архангельской, Мурманской, Новгородской, Псковской, Вологодской областей и Республики Карелия; в архиве сосредоточены материалы за годы Великой Отечественной Войны.

- Центральный государственный архив историко-политических документов Санкт-Петербурга (ЦГАИПД).

В архиве сосредоточены фонды органов коммунистической партии — Обкома, Горкома, райкомов, парткомов ведущих предприятий и организаций города — и комсомола Ленинграда и области за 1917-1991 года. Представлены документы о создании Красной армии, частей особого назначения, о деятельности продотрядов, полков бедноты и посылке партийных агитаторов на работу в деревню. Среди материалов партийных органов периода Второй мировой войны имеются фонды Ленинградского штаба партизанского движения и его отделов, партизанских отрядов, полков, бригад и политотдела Ленинградской армии народного ополчения.

- Центральный государственный архив литературы и искусства Санкт-Петербурга (ЦГАЛИ).

В архиве сосредоточены фонды государственных учреждений, общественных организаций литературы, искусства и культурно-

просветительной работы, а также фонды личного происхождения деятелей культуры С.-Петербурга с 1917 г. по настоящее время.

- Центральный государственный архив документов по личному составу ликвидированных государственных предприятий, учреждений, организаций Санкт-Петербурга (ЦГАЛС).

Архив хранит документы по личному составу ликвидированных предприятий, организаций и учреждений, не имеющих правопреемников. К документам по личному составу относятся следующие документы: приказы по личному составу, личные карточки, личные дела, лицевые счета, трудовые книжки, книги списочного состава и другие.

- Центральный государственный архив научно-технической документации Санкт-Петербурга (ЦГАНТД).

В фондах архива хранятся документы ведущих научно-исследовательских, проектных и конструкторских организаций Ленинграда - Санкт-Петербурга с 1917 года по настоящее время. В архиве представлена научно-техническая (проектная, конструкторская, научно-исследовательская, картографическая) и управленческая документация по отраслям промышленности (топливодобывающей, энергетической, металлургической, машиностроительной, химической, электротехнической, электронной, текстильной, пищевой и другой), транспорту, сельскому и лесному хозяйству, строительству, здравоохранению, геологии, метеорологии.

Все изображения научно-справочного аппарата архивов, участвующие в аprobации системы, представлены в сети Интернет на сайте «Архивы Санкт-Петербурга» [15].

В процессе испытаний системы было обработано более 35 тысяч документов научно-справочного аппарата пяти архивов, объемом более миллиона изображений. Точные сведения о количестве обработанных документов представлены в таблице 4.2.

Таблица 4.2. Размер корпуса обработанных документов

Набор данных		Количество документов	Количество изображений	Среднее кол-во изображений в документе
архив	вид			
ЦГА	описи	22 066	342 319	15
	указатели	200	8 251	41
ЦГАИПД	указатели	3 800	207 254	54
ЦГАЛИ	описи	1 415	28 251	19
ЦГАЛС	описи	3 501	59 064	17
ЦГАНТД	описи	1 626	63 524	39
<b>Всего:</b>		<b>32 608</b>	<b>708 663</b>	

### 4.3 Порядок проведения испытаний

Испытания разработанной системы распознавания архивных документов и автоматической корректировки результатов производились в следующем порядке:

1. Вначале были проанализированы существующие способы оценки качества распознавания и предложены дополнительные критерии оценки качества поиска по распознанным документам, наиболее подходящие для оценки результатов исследования.

2. Следующим этапом стал выбор наиболее подходящих OCR систем для распознавания всего корпуса архивных документов и отдельной оценки разработанного метода автоматической корректировки.

3. После выбора OCR систем было произведено распознавание всего корпуса документов в каждом архиве и проанализированы получившиеся результаты.

4. На основе распознанных документов были подготовлены структуры данных, настроены профили автоматической корректировки и проведена оценка эффективности разработанного метода автоматической корректировки.

5. Последним этапом стало проведение автоматической корректировки всего корпуса распознанных документов в каждом архиве и анализ получившихся результатов.

Все измерения и анализ результатов проводились при помощи разработанного в диссертационной работе инструментария.

## 4.4 Критерии оценки

### 4.4.1 Базовые критерии оценки

Рассмотрим существующие критерии оценки качества результатов распознавания [23].

1. *Коэффициент распознанных символов:*

$$rT_c = \frac{c}{n_c},$$

где  $c$  — общее количество символов в эталоне,  $n_c$  — количество символов в результате распознавания.

2. *Коэффициент распознанных слов:*

$$rT_w = \frac{w}{n_w},$$

где  $w$  — общее количество слов в эталоне,  $n_w$  — количество слов в результате распознавания.

3. *Словарная точность*

$$A_D = 1 - \frac{n_{error}}{n_w},$$

где  $n_w$  — общее количество лексем (слов) в результате распознавания,  $n_{error}$  — количество слов в результате распознавания, отсутствующих в словаре, состав словаря может отличаться в зависимости от тематики распознанного текста.

4. *Точность в символах:*

$$A_C = 1 - \frac{i_c + s_c + d_c}{c},$$

где  $i_c$ ,  $s_c$ ,  $d_c$  — количество ошибочно вставленных, замещенных и удаленных символов соответственно.

5. *Точность в словах:*

$$A_W = 1 - \frac{i_w + s_w + d_w}{w},$$

где  $i_w$ ,  $s_w$ ,  $d_w$  — количество ошибочно вставленных, замещенных и удаленных слов соответственно.

6. Точность в словах без учета порядка:

$$A_{W_b} = 1 - \frac{n_{missed} + n_{wrong}}{w},$$

где  $n_{missed}$  — количество пропущенных слов,  $n_{wrong}$  — количество ошибочно распознанных слов.

#### 4.4.2 Критерии оценки качества поиска

Рассмотренные выше базовые критерии оценки точности основываются на точном совпадении слов. При автоматическом выборе варианта корректировки бывает довольно трудно отдать предпочтение той или иной словоформе, поэтому в финальном результате могут содержаться корректные слова, написанные в некорректной форме (падеж, склонение и т.п.). Однако, для задач полнотекстового поиска по результатам распознавания с подсветкой вхождения поисковой фразы на изображении, данными неточностями можно пренебречь без ущерба качеству поиска.

Поисковый инструментарий системы ГИС «Государственные архивы Санкт-Петербурга» позволяет пользователю производить настройку параметров поиска. Статистический анализ поисковых запросов за предыдущие три года показал следующее:

1. пользователи производят поиск по ключевой фразе по совпадению всех слов или вхождению одного из слов;
2. пользователи не производят поиск по точному вхождению ключевой фразы;
3. в подавляющем большинстве случаев пользователи пользуются морфологическим поиском.

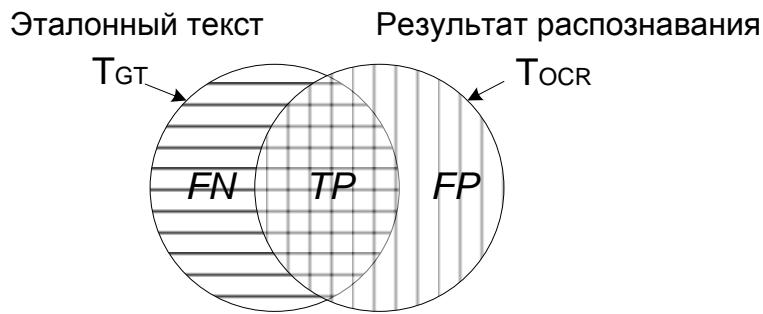
На основе результата анализа статистики поисковых запросов сформируем критерии оценки качества распознавания, которые будут наиболее точно отвечать задачам поиска по результатам распознавания пользователями.

Пусть  $T_{GT}$  — множество поисковых токенов эталонного текста, а  $T_{OCR}$  — множество поисковых токенов результата распознавания (рисунок 4.2).

$FN = T_{GT} / T_{OCR}$  — множество токенов, по которым будет выдан ложноотрицательный поисковый результат.

$FP = T_{OCR} / T_{GT}$  — множество токенов, по которым будет выдан ложноположительный поисковый результат.

$TP = T_{GT} \cap T_{OCR}$  — множество токенов, по которым будет выдан истинно положительный поисковый результат.



4.2. Отношение поисковых токенов эталона и результата распознавания

Для оценки качества будем использовать метрики точности *Precision*, полноты *Recall* и среднего гармонического *F* (*F-measure*), вычисляемые по следующим формулам:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \quad F = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

## 4.5 Выбор OCR систем

Выбор OCR систем для участия в процедуре оценки качества результатов распознавания основывался на сравнительном анализе их результатов.

Таблица 4.3. Наборы данных для сравнительного анализа OCR систем

Набор	Количество изображений	Количество слов	Количество символов
ПМ-2 (печатная машинка, среднее качество)	45	7380	52671
ПМ-3 (печатная машинка, высокое качество)	46	8875	64869
ПР-4 (принтер, очень высокое качество)	11	1858	13976

О перенесении вопроса "О политработе в свете Указа правительства на зав. Металлоизделий, ф-ке им. В. Слуцкой, Истеххоз и зав. Молотова с повестки дня заседания Бюро РК, имеющего быть 11 октября на повестку дня заседания Бюро РК 18/X-С.Г.

а) Набор «ПМ-2» (печатная машинка, среднее качество)

О выполнении постановления бюро РК КПСС от 9 марта 1973 года "О работе партийных организаций сельскохозяйственных предприятий по повышению роли специалистов в развитии общественного производства в свете постановления ЦК КПСС по Алтайской краевой партийной организации".

б) Набор «ПМ-3» (печатная машинка, высокое качество)

**1999 год**  
Документы (договоры, соглашения, справки, переписка) о передаче средств федерального бюджета для закупки и поставки сельскохозяйственной продукции в Федеральный продовольственный фонд, в Федеральный фонд зерна, о финансировании ярмарки «Российский фермер»

в) Набор «ПР-4» (принтер, очень высокое качество)

Рисунок 4.3. Примеры изображений

Сравнительный анализ производился путем распознавания трех наборов изображений различного качества. Описание наборов приведено в таблице 4.3, примеры изображений приведены на рисунке 4.3.

Каждому изображению был сопоставлен эталонный текст, введенный вручную. Формат изображений — JPEG, разрешение - 300dpi.

Результаты вычисления наиболее значимых критериев оценки качества распознавания различными OCR системами представлены в таблице 4.4, описание участующих OCR систем приведено в главе 1.4.1.

Для оценки словарной точности  $A_D$  использовался русскоязычный словарь GNU Aspell 0.60.6 [57].

Таблица 4.4. Результаты сравнительного анализа OCR систем

Набор	OCR система	$rT_w$	$rT_c$	$A_D$	$A_C$	$A_W$
<b>ПМ-2 (печатная машинка, среднее качество)</b>						
ПМ-2	Abbyy Finereader	101,12%	98,14%	82,14%	92,45%	76,80%
<b>ПМ-2</b>	<b>Cuneiform Linux</b>	<b>85,33%</b>	<b>50,14%</b>	<b>57,89%</b>	<b>18,19%</b>	<b>0,00%</b>
ПМ-2	Cuneiform Windows	121,06%	99,12%	69,70%	69,01%	27,50%
ПМ-2	IRIS Readiris	95,88%	80,01%	58,78%	56,88%	22,01%
ПМ-2	Nuance OmniPage	121,60%	102,79%	68,05%	82,51%	36,54%
ПМ-2	Tesseract	100,04%	88,77%	74,98%	65,74%	36,06%
<b>ПМ-3 (печатная машинка, высокое качество)</b>						
ПМ-3	Abbyy Finereader	100,86%	99,99%	80,88%	98,08%	90,55%
ПМ-3	Cuneiform Linux	97,65%	79,40%	67,02%	64,68%	30,28%
ПМ-3	Cuneiform Windows	109,07%	100,74%	74,33%	91,93%	67,37%
ПМ-3	IRIS Readiris	108,88%	102,02%	71,86%	91,33%	68,34%
ПМ-3	Nuance OmniPage	108,51%	101,09%	75,44%	94,01%	68,65%
ПМ-3	Tesseract	104,12%	99,25%	75,37%	92,59%	74,68%
<b>ПР-4 (принтер, очень высокое качество)</b>						
<b>ПР-4</b>	<b>Abbyy Finereader</b>	<b>99,68%</b>	<b>100,29%</b>	<b>93,90%</b>	<b>99,55%</b>	<b>99,25%</b>
ПР-4	Cuneiform Linux	99,41%	98,05%	92,15%	95,22%	90,20%
ПР-4	Cuneiform Windows	100,70%	100,16%	93,48%	99,09%	96,72%
ПР-4	IRIS Readiris	100,54%	99,90%	93,74%	97,72%	94,40%
ПР-4	Nuance OmniPage	100,59%	100,20%	94,38%	99,46%	98,39%
ПР-4	Tesseract	102,48%	100,31%	93,17%	98,59%	94,13%

Полученные результаты сравнительного анализа свидетельствуют о том, что коммерческая система «Abbyy Finereader» достигает максимального уровня качества. Другие коммерческие системы не превосходят по качеству свободно распространяемые системы. Среди свободно распространяемых систем явный лидер отсутствует, но самые низкие результаты показывает система «Cuneiform Linux».

Для распознавания всего корпуса архивных документов была выбрана система с открытыми исходными кодами «Tesseract». Первой причиной ее выбора послужило то, что она является свободно распространяемой, второй причиной

является формат представления результатов распознавания, содержащий информацию о координатах распознанных слов, что полностью удовлетворяет требованиям поискового механизма.

Для оценки качества разработанного метода автоматической корректировки помимо системы «Tesseract» была выбрана коммерческая система «Abbyy Finereader».

#### **4.6 Распознавание всего корпуса документов**

Первым шагом обработки всего корпуса документов было распознавание изображений и последующий анализ полученных результатов. Распознавание производилось свободно распространяемой системой «Tesseract».

Оценка качества результатов первичного распознавания производилась, путем вычисления словарной точности  $A_D$  с использованием словаря, состав которого описан в таблице 4.5.

Таблица 4.5. Состав словаря, используемого при вычислении словарной точности

Источник словоформ	Количество словоформ
Объединенные словари Зализняка [11] и Hunspell [63]	5 498 345
Фамилии	918 659
Имена	105 560
Отчества	231 313
Аббревиатуры	1 413
<b>Всего:</b>	<b>6 755 290</b>

Применяемый словарь был сформирован из различных источников и включает все словоформы исходных слов. Словоформы из словаря Зализняка были получены с помощью библиотеки «RussianMorphology for Lucene» [101], из словаря Hunspell с помощью утилиты Hunspell-tools [88], для склонения фамилий, имен и отчеств была использована java версия библиотеки padeg.dll [22]. Перечень уникальных фамилий, имен и отчеств был получен из государственного регистра населения Санкт-Петербурга. Итоговый размер словаря составил 6 755 290 словоформ.

Перед проведением словарной проверки результаты распознавания проходили процедуру нормализации, описанную в разделе 2.3.1.

Характеристики полученных результатов первичного распознавания представлены в таблице 4.6.

Таблица 4.6. Характеристики результатов первичного распознавания

Набор данных		Общее количество символов	Общее количество лексем $n_w$	Количество ошибочных лексем $n_{error}$	Словарная точность $A_D$
архив	вид				
ЦГА	описи	256 229 691	46 267 201	17 706 507	0,62
	указатели	6 776 594	1 289 212	625 751	0,51
ЦГАИПД	указатели	149 249 090	22 094 096	10 647 194	0,52
ЦГАЛИ	описи	17 907 428	3 200 646	1 091 795	0,66
ЦГАЛС	описи	57 193 957	9 353 017	4 529 334	0,52
ЦГАНТД	описи	36 089 001	5 837 216	1 854 795	0,68
<b>Всего:</b>		<b>523 445 761</b>	<b>88 041 388</b>	<b>36 455 376</b>	<b>0,59</b>

Словарная точность позволяет приблизительно оценить качество на большом объеме распознанных документов. Из полученных результатов следует, что 41% лексем не содержатся ни в одном из словарей. Данный результат объясняется тем, что в общем корпусе документов содержатся рукописные документы, а также документы с низким качеством печати.

Для повышения качества результатов распознавания в диссертационной работе был реализован метод автоматической корректировки. В ходе его разработки необходимо было проанализировать весь корпус материалов на предмет встречающихся символов. Перечень всех уникальных символов встретившихся в результатах распознавания архивных документов представлен в таблице 4.7.

Всего было обнаружено 122 уникальных символа. Стоит отметить наличие разнообразных по написанию пробельных символов и символов тире, кавычек, скобок, что было учтено в программной реализации путем применения специальных шаблонов регулярных выражений, обозначающих тип символов.

Также можно выделить ряд специальных символов и символов пунктуации, которые возможно были ошибочно получены в процессе распознавания.

Таблица 4.7. Уникальные символы в результатах распознавания

Группа символов	Значения
<b>Алфавитные</b>	Ё А Б В Г Д Е Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ ъ й Ъ Э Ю Я а б в г д е ж з и й к л м н о п р с т у ф х ц ч ш щ ъ ў є ю я ё
<b>Цифровые</b>	0 1 2 3 4 5 6 7 8 9
<b>Математические</b>	+ < = >
<b>Специальные</b>	^ ` © ® °
<b>Денежные единицы</b>	\$
<b>Пунктуация:</b>	
тире	- —
открывающая скобка	( [ { , ,,
закрывающая скобка	) ] }
открывающая кавычка	« ‘ “ ‘
закрывающая кавычка	» ’ ” ’
подчеркивание	—
другие символы пунктуации	! " # % & ' * , . / : ; ? @ \

#### 4.7 Оценка метода автоматической корректировки

Перед запуском процесса массовой корректировки ошибок, допущенных при распознавании всего корпуса документов, была произведена оценка качества предложенного метода автоматической корректировки.

Для проведения оценки были подготовлены тестовые наборы изображений по каждому архиву в отдельности. Далее наборы изображений были распознаны различными OCR системами с включенной и отключенной автоматической корректировкой результатов распознавания. Сравнительный анализ полученных результатов, позволил оценить эффективность предложенного в работе метода.

#### **4.7.1 Подготовка тестовых наборов изображений**

Изображения для проведения экспериментальной оценки были вручную отобраны в каждом архиве. Отбиралось несколько десятков изображений с равномерным распределением по всему множеству изображений. Главным критерием отбора служило наличие на изображениях печатного текста, содержащего лексикон специфичный для тематики конкретного архива.

Все изображения проходили предварительную обрезку в графическом редакторе, в результате которой на всех изображениях оставались только области, содержащие текстовую информацию. Данная обработка необходима для, того чтобы сделать сравнительный анализ независимым от способностей OCR систем выполнять структурный анализ.

Для каждого изображения был вручную подготовлен эталонный текст. Формат тестовых изображений — JPEG, разрешение - 300dpi.

Все отобранные изображения прошли процесс распознавания двумя OCR системами («Tesseract», «Abbyy Finereader») и оценку точности на уровне слов  $A_w$ . Далее для каждой OCR системы изображения были отсортированы в порядке убывания точности распознанного текста на уровне слов  $A_w$  и распределены по пяти тестовым наборам, каждый из которых отвечал определенному диапазону точности.

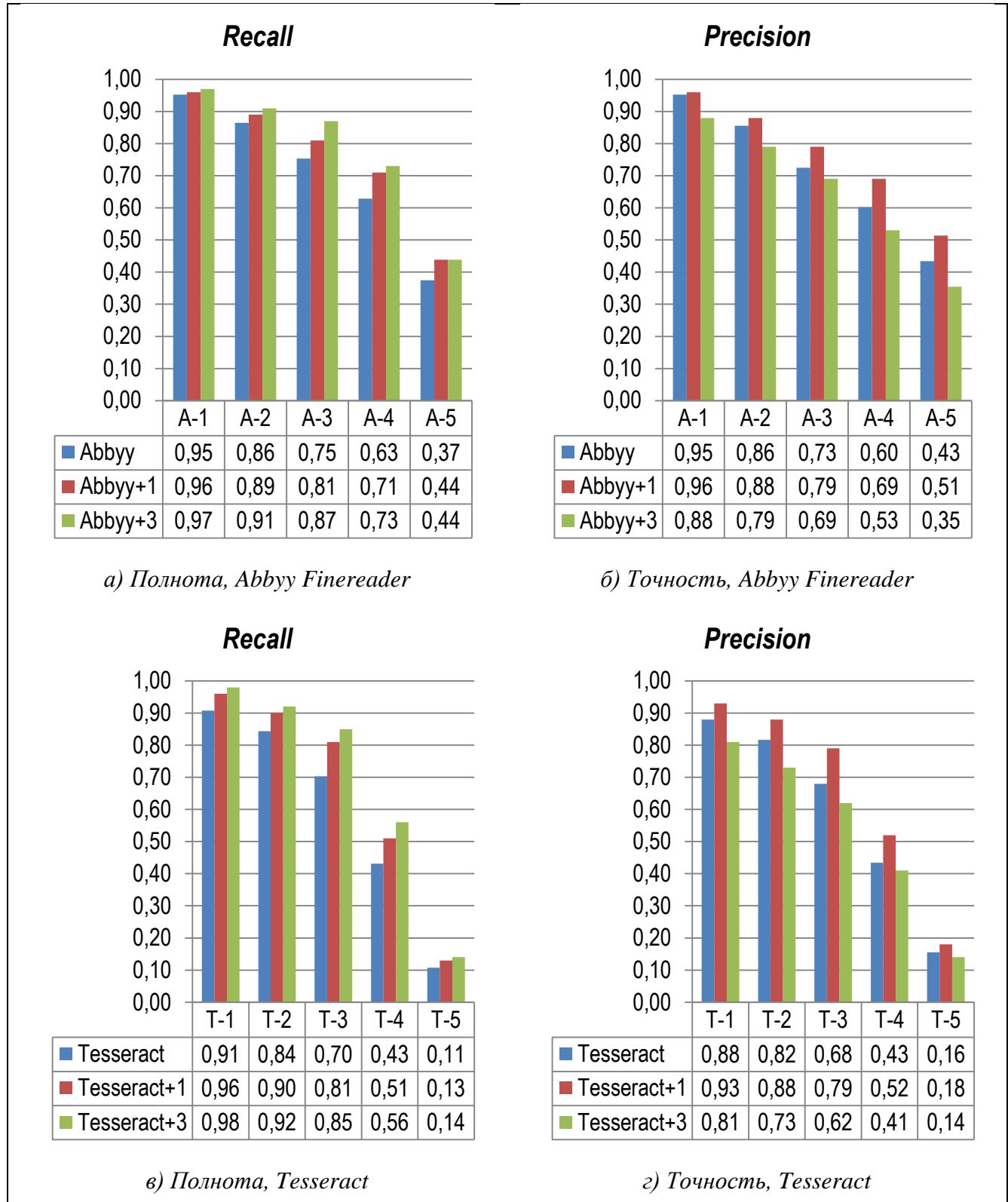
#### **4.7.2 Анализ результатов**

Рассмотрим результаты оценки метода автоматической корректировки на основе тестовых наборов изображений документов архива ЦГАИПД.

Результаты распознавания были получены коммерческой OCR системой «Abbyy Finereader» (Abbyy) и свободно распространяемой системой «Tesseract». Для оценки качества эталонный текст и результат распознавания изображения разбивались на поисковые токены, далее вычислялась полнота *Recall* и точность *Precision* наличия токенов эталона в результате распознавания.

Сравнение значений полноты и точности результатов распознавания тестовых наборов изображений без корректировки, с результатами распознавания,

содержащими один (наилучшая корректировка: «+1») и три варианта (наилучшая корректировка плюс две альтернативных: «+3») замены ошибочных слов, представлено на рисунке 4.4.



4.4. Сравнение оценок полноты и точности до и после корректировки. Соответствие наборов диапазонам точности на уровне слов: «A-1», «T-1» — 100% – 80%; «A-2», «T-2» — 80% – 60%; «A-3», «T-3» — 60% – 40%; «A-4», «T-4» — 40% – 20%; «A-5», «T-5» — 20% – 0%.

Сравнительный анализ показал, что разработанный метод корректировки повышает качество результатов распознавания как коммерческих, так и свободно распространяемых OCR систем. Наибольшие **приращения (до +15%)** показателей полноты и точности отмечаются на результатах распознавания, находящихся в диапазоне словарной точности от 80 до 20%, что объясняется малым количеством «простых» ошибок в верхнем диапазоне и низким качеством результатов в нижнем диапазоне.

Увеличение значения полноты результатов распознавания при учете альтернативных корректировок свидетельствует о том, что верные корректировки не всегда определяются как наилучшие, но присутствуют в списке альтернативных корректировок, что указывает на возможность и необходимость дальнейшей доработки алгоритма и правил ранжирования корректировок.

#### **4.8 Оценка результатов автоматической корректировки всего корпуса документов**

Последним этапом испытаний разработанной системы стало проведение автоматической корректировки всего корпуса результатов распознавания архивных документов (**32 608** документов объемом **708 663** изображений), состоящего из **88 041 388** лексем.

Характеристики результатов распознавания до и после автоматической корректировки представлены в таблице 4.8

Таблица 4.8. Оценка результатов распознавания до и после автоматической корректировки (« $\Delta-$ » — количество исправленных ошибок, « $\Delta+$ » — увеличение словарной точности).

<b>Набор данных</b>		<b>Количество ошибок <math>n_{error}</math></b>			<b>Словарная точность <math>A_D</math></b>		
<b>архив</b>	<b>вид</b>	<b>до</b>	<b>после</b>	<b><math>\Delta -</math></b>	<b>до</b>	<b>после</b>	<b><math>\Delta +</math></b>
ЦГА	описи	17 706 507	11 465 500	6 241 007	0,62	0,75	0,13
	указатели	625 751	477 872	147 879	0,51	0,63	0,12
ЦГАИПД	указатели	10 647 194	5 126 549	5 520 645	0,52	0,77	0,25
ЦГАЛИ	описи	1 091 795	742 025	349 770	0,66	0,77	0,11
ЦГАЛС	описи	4 529 334	920 825	3 608 509	0,52	0,9	0,38
ЦГАНТД	описи	1 854 795	1 224 657	630 138	0,68	0,79	0,11
<b>Всего:</b>		<b>36 455 376</b>	<b>19 957 428</b>	<b>16 497 948</b>	<b>0,59</b>	<b>0,77</b>	<b>0,18</b>

Количество ошибочных слов  $n_{error}$  после автоматической корректировки **сократилось на 45%, было исправлено 16 497 948 ошибочных лексем, значение словарной точности в среднем по всем архивам увеличилось на 18%.**

**Максимальное увеличение словарной точности на 38% и сокращение количества ошибочных слов на 80%** было достигнуто при корректировке результатов распознавания документов научно-справочного аппарата ЦГАЛС, а минимальное увеличение словарной точности на 11% было получено для ЦГАЛИ и ЦГАНТД. Данное различие можно объяснить более качественными структурами данных, используемыми при корректировке документов ЦГАЛС. Структуры данных ЦГАЛС, содержат меньшее количество некорректных слов, так как качество первичных результатов распознавания документов ЦГАЛС превосходит другие архивы, поскольку большая часть документов создана на печатной машинке или принтере. В дополнение, структуры данных ЦГАЛС являются более полноценными в связи с тем, что объем материалов ЦГАЛС, используемых для построения структур данных, превосходит суммарный объем материалов ЦГАНТД и ЦГАЛИ.

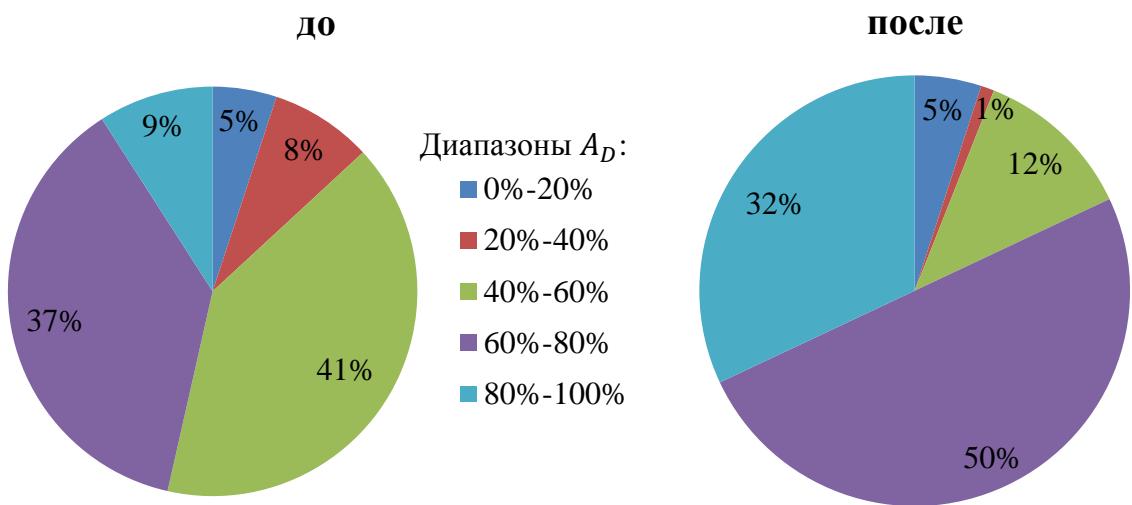
Распределения количества изображений по диапазонам словарной точности результатов распознавания до и после корректировки представлены в абсолютных значениях в таблицах 4.9 и 4.10 и в относительных значениях на рисунке 4.5.

Таблица 4.9. Распределение по диапазонам словарной точности до корректировки

Набор данных		Количество изображений в диапазонах словарной точности $A_D$ до автоматической корректировки				
		0%-20%	20%-40%	40%-60%	60%-80%	80%-100%
ЦГА	описи	14 695	9 375	133 053	148 236	36 960
	указатели	172	1 002	4 782	2 049	246
ЦГАИПД	указатели	10 832	35 705	106 201	51 368	3 019
ЦГАЛИ	описи	3 852	1 710	13 817	34 327	9 275
ЦГАЛС	описи	5 079	633	6 251	13 863	2 705
ЦГАНТД	описи	1 523	11 625	21 571	13 410	10 932
<b>Итого:</b>		<b>36 153</b>	<b>60 050</b>	<b>285 675</b>	<b>263 253</b>	<b>63 137</b>

Таблица 4.10. Распределение по диапазонам словарной точности после корректировки

Набор данных		Количество изображений в диапазонах словарной точности $A_D$ после автоматической корректировки				
		0%-20%	20%-40%	40%-60%	60%-80%	80%-100%
ЦГА	описи	15 145	2 367	50 423	181 830	82 880
	указатели	138	496	3 384	3 609	624
ЦГАИПД	указатели	7 715	2 713	21 343	100 845	74 350
ЦГАЛИ	описи	3 961	891	6 364	31 719	20 046
ЦГАЛС	описи	5 079	275	2 685	14 148	6 229
ЦГАНТД	описи	2 580	315	2 993	16 728	36 448
<b>Итого:</b>		<b>34 618</b>	<b>7 057</b>	<b>87 192</b>	<b>348 879</b>	<b>220 577</b>



4.5. Распределение количества изображений по диапазонам словарной точности до и после автоматической корректировки

Представленные распределения наглядно демонстрируют, что проведенная корректировка улучшила качество результатов распознавания значительного количества изображений. Корректировка не принесла положительного эффекта лишь для результатов распознавания, словарная точность которых находится в диапазоне от 0% до 20%.

Необходимо отметить существенное увеличение (на 23%) количества результатов распознавания в диапазоне словарной точности высокого уровня и значительное уменьшение (на 36%) количества результатов распознавания в диапазонах словарной точности низкого уровня (0%-60%).

#### **4.9 Выводы по четвертой главе**

Предложенные в диссертационной работе методы и алгоритмы были реализованы, апробированы и введены в промышленную эксплуатацию в составе государственной информационной системы «Государственные архивы Санкт-Петербурга» в качестве подсистемы распознавания архивных документов, подсистемы автоматической корректировки результатов и подсистемы поиска по изображениям документов архивного фонда. Результаты экспериментальных исследований успешно подтвердили выдвинутые ранее теоретические положения.

В ходе апробации было успешно проведено распознавание накопленного массива электронных образов документов научно-справочного аппарата архивов с последующей автоматической корректировкой полученных результатов. Весь процесс обработки документов производился по предложенной в работе технологии.

Проведение настройки системы и сравнительного анализа эффективности обработки документов различными конфигурационными профилями при помощи предложенного в работе инструментария подтвердило его применимость и удобство в использовании.

Достоверность, значимость и практическая применимость результатов диссертационной работы подтвердились успешными результатами распознавания документов пяти центральных государственных архивов Санкт-Петербурга и значительным повышением их качества за счет применения предложенного метода автоматической корректировки.

Применение разработанных систем распознавания и корректировки в процессах электронного комплектования архивов новыми документами и перевода бумажных документов в цифровой вид позволило значительно увеличить скорость и объем обрабатываемых документов. Основываясь на статистических данных проекта «Государственные архивы Санкт-Петербурга», ручной ввод описей размером в 500 тысяч изображений составил бы около 70 человеко-лет, с помощью предложенных в работе систем их обработка заняла

несколько недель. Таким образом, применение средств автоматического распознавания и корректировки позволило сократить годы ручного труда.

Полученные результаты распознавания архивных документов были впоследствии проиндексированы и использованы в системе поиска по изображениям документов архивного фонда, что позволило значительно расширить поисковое пространство для работы граждан, исследователей и сотрудников архивов.

## Заключение

Полученные в диссертационном исследовании результаты представляют собой решение актуальной задачи повышения качества распознавания изображений архивных документов. Применение результатов при распознавании исторических архивных документов способствует сохранению культурного наследия страны и обеспечения его доступности для граждан. В ходе исследования получены следующие основные результаты:

1. Разработан метод автоматической корректировки ошибок распознавания текста на основе рейтинго-ранговой модели текста, позволяющий производить корректировку результатов распознавания документов различных предметных областей, содержащих узкоспециализированную терминологию.
2. Разработан алгоритм извлечения из результатов распознавания и тематических текстов корпусных тезаурусов и структур данных, применяемых при корректировке.
3. Разработаны правила ранжирования корректировок, основанные на частотных характеристиках и статистической вероятности сочетаемости с предшествующими словами.
4. Разработан инструментарий, позволяющий эксперту производить настройку системы для обработки архивных документов различных тематических областей.
5. Разработана система распознавания документов архивного фонда, реализующая предложенные методы и алгоритмы автоматической корректировки результатов.

Результаты диссертационного исследования внедрены в СПб ГУП «Санкт-Петербургский информационно-аналитический центр», семи центральных государственных архивах и Архивном комитете Санкт-Петербурга в составе государственной информационной системы «Государственные архивы Санкт-Петербурга». Копии свидетельств о государственной регистрации программ для ЭВМ и акты внедрения представлены в приложениях Б и В соответственно.

Дальнейшее развитие научных исследований в данной области может быть проведено в следующих направлениях:

- Повышение качества распознавания за счет разработки и применения методов, алгоритмов и средств предварительной обработки изображений, что позволит достигать результатов высокой точности даже при обработке документов низкого исходного качества.
- Разработка критериев оценки качества распознавания документов, учитывающих семантическую связность результатов на основе различных мер ассоциации [12], что позволит выявлять и исключать из поискового индекса документы с большим количеством словесных ошибок распознавания и корректировки, которые невозможно определить, используя лишь словарную оценку точности.
- Применение механизмов автоматического или полуавтоматического пополнения/построения тематических тезаурусов [6,7] и выделения терминов и терминологических словосочетаний из текстов предметных областей обрабатываемых документов для повышения эффективности предложенного метода автоматической корректировки.

Результаты диссертационного исследования также могут быть успешно применены в проектах по оцифровке фондов библиотек, музеев и других учреждений и предприятий.

Разработанную систему можно использовать как самостоятельный автономный сервис распознавания изображений, интегрирующийся с информационными системами различных ведомств, а также с приложениями стационарных компьютеров и мобильных устройств. Взаимодействие с данным сервисом может осуществляться по сети интернет через разработанный программный интерфейс API.

## Список сокращений и условных обозначений

- БД - база данных.
- ИС - информационная система.
- НСА - научно-справочный аппарат архива.
- ОС - операционная система.
- ПК - программный комплекс.
- ПМ - программный модуль.
- ПО - программное обеспечение.
- СУБД - система управления базой данных.
- ЭА - электронный архив.
- API - application programming interface [14] (интерфейс программирования приложений).
- OCR - optical character recognition (оптическое распознавания символов)
- XML - eXtensible Markup Language.

## Словарь терминов

- биграмма: Последовательность из двух последовательных элементов (символов или слов).
- профиль: Множество допустимых параметров для распознавания документов определенного типа, выбранных из всего множества параметров конфигурации процесса распознавания.
- тезаурус: Отобранные слова или понятия; тезаурус выполняет роль специализированного словаря заданной предметной области, например медицина, музыка. [2].
- триграмма: Последовательность из трех последовательных элементов (символов или слов).
- n-грамма: Последовательность из n последовательных элементов (символов или слов).

## Список литературы

1. Автоматическая обработка текста [Электронный ресурс]. – Режим доступа: <http://www.aot.ru>, свободный. – Загл. с экрана (дата обращения: 10.05.2014).
2. Александров, В.В. Интеллект и компьютер / В.В. Александров. – СПб. : Издательство «Анатолия», 2004. – 251 с.
3. Арлазаров, В.Л. Адаптивное распознавание / В.Л. Арлазаров, Н.В. Котович, О.А. Славин // Информационные технологии и вычислительные системы. – 2002. – № 4. – С. 11–23.
4. Арлазаров, В.Л. Алгоритмы распознавания и технологии ввода текстов в ЭВМ / В.Л. Арлазаров, О.А. Славин // Информационные технологии и вычислительные системы. – 1996. – №1. – С. 48-54.
5. Архивы - Архивы Санкт-Петербурга [Электронный ресурс] : офиц. сайт. – СПб., 2011-2015. – Режим доступа: <http://spbarchives.ru/web/group/archives>, свободный. – Загл. с экрана (дата обращения 24.10.2014).
6. Бессмертный, И.А. Метод автоматического построения тезаурусов на основе статистической обработки текстов на естественном языке / И.А. Бессмертный, А.Б. Нугуманова // Известия Томского политехнического университета. – 2012. – т.321, № 5. – С. 125-130.
7. Боярский, К.К. Проблемы пополнения семантического словаря / К.К. Боярский, Е.А. Каневский // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. – 2011. – Выпуск 2(72). – С. 132-137.
8. Гамма, Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования = Design Patterns: Elements of Reusable Object-Oriented Software / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. – СПб. : Питер, 2007. – 366 с.
9. Гасфилд, Д. Строки, деревья и последовательности в алгоритмах / Д. Гасфилд. – СПб. : БВХ-Петербург, 2003. – 654 с.

10. Городецкий, А.Е. Управление и нейронные сети / А.Е. Городецкий, И.Л. Тарасов. – СПб. : Издательство Политехнического университета, 2005. – 312 с.
11. Зализняк, А.А. Грамматический словарь русского языка / А.А. Зализняк. – М. : Русский язык. – 1980. – 880 с.
12. Захаров, В.П. Выделение терминологических словосочетаний из специальных текстов на основе различных мер ассоциации / В.П. Захаров, М.В. Хохлова // Сборник научных статей XVII Всероссийской объединенной конференции «Интернет и современное общество». – СПб.: Учреждение «Университетские телекоммуникации», 2014. – С. 290–293.
13. Зиняков, В.Ю. Восстановление двумерных изображений с дефектами / В.Ю. Зиняков, А.Е. Городецкий, А.Ю. Кучмин, Е.И. Зеленев, Н.В. Алферова // Информационно-управляющие системы. – 2013. – № 3(64). – С. 8-15.
14. Интерфейс программирования приложений – Википедия [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Интерфейс\\_программирования\\_приложений](https://ru.wikipedia.org/wiki/Интерфейс_программирования_приложений), свободный. – Загл. с экрана (дата обращения: 28.09.2014).
15. Информационные ресурсы - Архивы Санкт-Петербурга [Электронный ресурс]. – Режим доступа: [http://spbarchives.ru/web/group/information\\_resources](http://spbarchives.ru/web/group/information_resources), свободный. – Загл. с экрана (дата обращения: 24.10.2014).
16. Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли, К. Бегг. – 3-е изд. – М. : Вильямс.– 2003. – 1436с.
17. Кулешов, С.В. Методы сегментации OCR систем в задачах автоматической обработки архивных документов / С.В. Кулешов, С.В. Смирнов // Труды СПИИРАН. – 2011. – Выпуск 1(16). – С. 110–122.
18. Левенштейн, В. Двоичные коды с исправлением выпадений, вставок и замещений символов / В. Левенштейн // Доклады Академии Наук СССР. – 1965. –т.163, № 4. – С. 845-848.

19. Об организации деятельности исполнительных органов государственной власти Санкт-Петербурга по развитию, подключению и эксплуатации единой мультисервисной телекоммуникационной сети исполнительных органов государственной власти Санкт-Петербурга и создании государственной информационной системы Санкт-Петербурга «Учет ресурсов единой мультисервисной телекоммуникационной сети исполнительных органов государственной власти Санкт-Петербурга» [Электронный ресурс] : [Постановление Правительства Санкт-Петербурга от 01.07.2011. № 884]. – Электрон. текстовые дан. – Доступ из информ.-справоч. системы «Кодекс».
20. Расстояние Левенштейна – Википедия [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Расстояние\\_Левенштейна](https://ru.wikipedia.org/wiki/Расстояние_Левенштейна), свободный. – Загл. с экрана (дата обращения: 27.09.2014).
21. Реестр государственных информационных систем Санкт-Петербурга [Электронный ресурс]. – Режим доступа: <http://reestr-gis.spb.ru/#regis:is2053>, свободный. – Загл. с экрана (дата обращения: 09.10.2014).
22. Склонение фамилий, имен и отчеств по падежам Библиотека функций [Электронный ресурс]. – Режим доступа: <http://www.delphikingdom.com/asp/viewitem.asp?catalogid=412>, свободный. – Загл. с экрана (дата обращения: 16.11.2014).
- 23 Смирнов, С.В. Критерии оценки качества результатов оптического распознавания / С.В. Смирнов // Сборник материалов XVI Международной научно-практической конференции «Перспективы развития информационных технологий». – Новосибирск: Издательство ЦРНС, 2013. – С. 33–38.
24. Смирнов, С.В. Логическая модель представления информации в электронном архиве / С.В. Смирнов // Сборник научных трудов IV Всероссийской научно-практической конференции с международным участием «Научное творчество XXI века». – Красноярск: Научно-инновационный центр, 2011. – выпуск 2. – С. 93-94.

25. Смирнов, С.В. Подсистема массового распознавания изображений архивных документов / С.В. Смирнов // Труды СПИИРАН. – 2012. – выпуск 3(22). – С. 234–248.
26. Смирнов, С.В. Корректировка ошибок оптического распознавания на основе рейтинго-ранговой модели текста / С.В. Смирнов // Труды СПИИРАН. – 2014. – выпуск 4(35). – С. 64-82.
27. Смирнов, С.В. Сравнительный анализ OCR систем в контексте построения системы поиска по изображениям архивных документов / С.В. Смирнов // Информационно-измерительные и управляющие системы. – 2014. – т. 12, №12. – С. 62–69.
28. Сокирко, А.В. Морфологические модули на сайте [www.aot.ru](http://www.aot.ru) / А.В. Сокирко // Материалы конференции «Диалог-2004». – 2004.
29. Стратегия развития информационного общества в Российской Федерации от 7 февраля 2008 г. N Пр-212 // Российская газета. –2008. –16 фев.
30. Трёхуровневая архитектура приложений – Википедия [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Трёхуровневая\\_архитектура](https://ru.wikipedia.org/wiki/Трёхуровневая_архитектура), свободный. – Загл. с экрана (дата обращения: 29.09.2014).
31. Шоломов, Д.Л. Коррекция распознанного текста с использованием методов классификации / Д.Л. Шоломов // Труды ИСА РАН. – 2007. – т. 29. – С. 356–371.
32. Шоломов, Д.Л. Синтаксический подход к пост-обработке нечетко распознанного текста / Д.Л. Шоломов // Сборник трудов ИСА РАН «Документооборот. Концепции и инструментарий». – М.: Едиториал УРСС, 2004. – С. 193–207.
33. Шоломов, Д.Л., Постников В.В., Марченко А.А., Усков А.В. Пост-обработка результатов OCR распознавания, использующая частично определенный синтаксис / Д.Л. Шоломов, В.В. Постников, А.А. Марченко, А.В. Усков // Труды ИСА РАН. – 2005. – т.16. – С. 146–163.

34. Энциклопедический словарь Брокгауза и Эфрона. – СПб.: Типография АО "Брокгауз и Эфрон", 1890. – т. 1а. – 690 с.
35. ABBYY FineReader [Электронный ресурс]. – Режим доступа: <http://www.abbyy.ru/finereader/>, свободный. – Загл. с экрана (дата обращения: 29.04.2014).
36. AfterScan – post-OCR text proofing, advanced spell-checking, automatic correction [Электронный ресурс]. – Режим доступа: <http://www.afterscan.com/ru/>, свободный. – Загл. с экрана (дата обращения: 12.11.2014).
37. Ahmed, F. MultiSpell: an N-Gram Based Language-Independent Spell Checker / F. Ahmed., Ernesto William De Luca, A. Nurnberger // Proceedings of Eighth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2007). – 2007.
38. Anderson N. IMPACT Briefing Paper: Optical Character Recognition [Электронный ресурс] / N. Anderson. – 2010. – Режим доступа: <http://www.impact-project.eu/uploads/media/IMPACT-ocr-bp-pilot-1b.pdf>, свободный. – Загл. с экрана (дата обращения: 06.06.2012)
39. Anderson, N. IMPACT Best Practice Guide: Optical Character Recognition – Part 1 [Электронный ресурс] / N. Anderson. – 2010. – Режим доступа: <http://www.impact-project.eu/uploads/media/IMPACT-ocr-bpg-pilot-s1.pdf>, свободный. – Загл. с экрана (дата обращения: 06.06.2012)
40. AnyDoc Software [Электронный ресурс]: офиц. сайт. – Режим доступа: <http://www.anydocsoftware.com/>, свободный. – Загл. с экрана (дата обращения: 29.04.2014).
41. Apache Lucene [Электронный ресурс]. – Режим доступа: <http://lucene.apache.org>, свободный. – Загл. с экрана (дата обращения: 18.09.2014).
42. Apache Tomcat - Welcome! [Электронный ресурс]. – Режим доступа: <http://tomcat.apache.org/>, свободный. – Загл. с экрана (дата обращения: 29.09.2014).

43. Brill, E. An Improved Error Model for Noisy Channel Spelling Correction / E. Brill, R. Moore // Proceedings of the 38<sup>th</sup> Annual Meeting on Association for Computational Linguistics (ACL '00). – 2000. –pp. 286–293.
44. Carrasco, R.C. Text Digitisation [Электронный ресурс] / R.C. Carrasco. – Режим доступа: <https://sites.google.com/site/textdigitisation/home>, свободный. – Загл. с экрана (дата обращения: 27.12.2013)
45. Clara OCR [Электронный ресурс]. – Режим доступа: <http://freecode.com/projects/claraocr>, свободный. – Загл. с экрана (дата обращения: 29.04.2014).
46. CuneiForm - Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/CuneiForm>, свободный. – Загл. с экрана (дата обращения: 18.09.2014).
47. Cuneiform Linux [Электронный ресурс]. – Режим доступа: <https://launchpad.net/cuneiform-linux>, свободный. – Загл. с экрана (дата обращения: 29.04.2014).
48. Cuneiform Windows [Электронный ресурс]. – Режим доступа: [http://cognitiveforms.com/ru/products\\_and\\_services/](http://cognitiveforms.com/ru/products_and_services/) cuneiform, свободный. – Загл. с экрана (дата обращения: 29.04.2014).
49. Cvision ocr [Электронный ресурс]. – Режим доступа: <https://www.cvisiontech.com>, свободный. – Загл. с экрана (дата обращения: 29.04.2014).
50. Damerau, F.J. A technique for computer detection and correction of spelling errors / F.J. Damerau // Commun. ACM. – 1964. – vol. 7, no. 3. – pp. 171–176.
51. Dynamsoft OCR SDK [Электронный ресурс]. – Режим доступа: <http://www.dynamsoft.com>, свободный. – Загл. с экрана (дата обращения: 29.04.2014).
52. ExperVision TypeReader & RTK [Электронный ресурс]. – Режим доступа: <http://www.experimentation.com>, свободный. – Загл. с экрана (дата обращения: 29.04.2014).

53. Fossati, D. A Mixed Trigrams Approach for Context Sensitive Spell Checking / D. Fossati, Barbara Di Eugenio // Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing '07). – 2007. – pp. 623–633.
54. Ginter, F. New Techniques for Disambiguation in Natural Language and Their Application to Biological Text / F. Ginter, J. Boberg, J. Jarvinen, T. Salakoski // J. Mach. Learn. Res. – 2004. – vol. 5. – pp. 605–621.
55. Gupta, M.R. OCR binarization and image pre-processing for searching historical documents / M.R. Gupta, N.P. Jacobson, E.K. Garcia // Pattern Recognition. – 2007. – no. 2. – pp. 389-397.
56. GlassFish Server [Электронный ресурс]. – Режим доступа: <https://glassfish.java.net/>, свободный. – Загл. с экрана (дата обращения: 29.09.2014).
57. GNU Aspell [Электронный ресурс]. – Режим доступа: <http://aspell.com>, свободный. – Загл. с экрана (дата обращения: 16.11.2014).
58. GOCR [Электронный ресурс]. – Режим доступа: <http://jocr.sourceforge.net>, свободный. – Загл. с экрана (дата обращения: 29.04.2014).
59. Google Web 1T Data [Электронный ресурс]. – Режим доступа: <http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T13>, свободный. – Загл. с экрана (дата обращения: 06.06.2012).
60. Hauser, A.W. OCR-Postcorrection of Historical Texts : thesis /Andreas W. Hauser. – München: 2007. – 90 p.
61. Hauser, A.W. Unsupervised Learning of Edit Distance Weights for Retrieving Historical Spelling Variations / A.W. Hauser, K.U. Schulz // Proceedings of the First Workshop on Finite-State Techniques and Approximate Search. – 2007. – pp. 1–6.
62. He, J. A comparison of binarization methods for historical archive documents / J. He, Q.D.M. Do, A.C. Downton, J.H. Kim // Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05). – 2005. – pp. 538-542.

63. Hunspell [Электронный ресурс]. – Режим доступа: <http://sourceforge.net/projects/hunspell/files/Hunspell/Documentation/>, свободный. – Загл. с экрана (дата обращения: 16.11.2014).
64. ImageMagick: Convert, Edit, Or Compose Bitmap Images [Электронный ресурс]. – Режим доступа: <http://wwwimagemagick.org/>, свободный. – Загл. с экрана (дата обращения: 05.10.2014).
65. IRIS Readiris [Электронный ресурс]. – Режим доступа: <http://www.irislink.com>, свободный. – Загл. с экрана (дата обращения: 29.04.2014).
66. ISO 14721:2003. Space data and information transfer systems – Open archival information system – Reference model [Электронный ресурс] : стандарт ISO. – Режим доступа: <https://www.iso.org/obp/ui/#iso:std:iso:14721:ed-2:v1:en>. – Загл. с тит. страницы (дата обращения: 25.11.2014).
67. Java [Электронный ресурс]. – Режим доступа: <http://java.com>, свободный. – Загл. с экрана (дата обращения: 18.09.2014).
68. Javaocr [Электронный ресурс]. – Режим доступа: <http://sourceforge.net/projects/javaocr/>, свободный. – Загл. с экрана (дата обращения: 29.04.2014).
69. JBoss Application Server 7 - JBoss Community [Электронный ресурс]. – Режим доступа: <http://jbossas.jboss.org/>, свободный. – Загл. с экрана (дата обращения: 29.09.2014).
70. Jones, M.A. Integrating multiple knowledge sourcesm a Bayesian OCR post-processor / M.A. Jones, G.A. Story, B.W. Ballard // Proceedings of IDCAR-91. – 1991. – pp. 925–933.
71. Kai, N. Unsupervised Post-Correction of OCR Errors : диссертация / Niklas Kai. – Hannover: Leibniz University. –2010. – 111 p.
72. Khare A. A Fresh Graduate’s Guide to Software Development Tools and Technologies, Chapter 6 Scalability [Электронный ресурс] / A. Khare, Y. Huang, H. Doan, M.S. Kanwal. – 2012. – 24 p. – Режим доступа: <http://www.comp.nus.edu.sg/~seer/book/2e/Ch06.%20Scalability.pdf>, свободный. – Загл. с тит. страницы (дата обращения: 29.10.2014).

73. Kolak, O. A Generative Probabilistic OCR Model for NLP Applications / O. Kolak, W. Byrne, P. Resnik // HLT-NAACL. – 2003. – pp. 55–62.
74. Kukich, K. Techniques for automatically Correcting Words in Text / K. Kukich // ACM computing survey Computational Linguistic. – 1992. – vol. 24, no. 4. – pp. 377–439.
75. LEADTOOLS OCR SDK [Электронный ресурс]. – Режим доступа: <http://www.leadtools.com>, свободный. – Загл. с экрана (дата обращения: 29.04.2014).
76. LOCR [Электронный ресурс]. – Режим доступа: <http://www.math.northwestern.edu/~mlerma/locr/>, свободный. – Загл. с экрана (дата обращения: 29.04.2014).
77. Mays, E. Context Based Spelling Correction / E. Mays, F.J. Damerau, R.L. Mercer // Inf. Process. Manage. – 1991. – vol. 27, no. 5. – pp. 517–522.
78. Mordani, R. Java Servlet Specification Version 3.0 [Электронный ресурс] / R. Mordani. – USA. –2009. – Доступ с сайта Oracle, свободный. – Загл. с тит. экрана (дата обращения: 20.10.2014).
79. Myka, A. Fuzzy Full-Text Searches in OCR Databases / A. Myka, U. Güntzer // Proceedings of the ADL '95. – 1996. – pp. 131–145.
80. MySQL Database [Электронный ресурс]. – Режим доступа: <http://www.mysql.com/>, свободный. – Загл. с экрана (дата обращения: 29.09.2014).
81. Nuance OmniPage [Электронный ресурс]. – Режим доступа: <http://www.nuance.com>, свободный. – Загл. с экрана (дата обращения: 29.04.2014).
82. Ocrad [Электронный ресурс]. – Режим доступа: <http://www.gnu.org/software/ocrad/>, свободный. – Загл. с экрана (дата обращения: 29.04.2014).
83. OCRchie [Электронный ресурс]. – Режим доступа: <http://www.eecs.berkeley.edu/~fateman/kathey/ocrchie.html>, свободный. – Загл. с экрана (дата обращения: 29.04.2014).

84. Ocre [Электронный ресурс]. – Режим доступа: <http://lem.eui.upm.es/ocre.html>, свободный. – Загл. с экрана (дата обращения: 29.04.2014).
85. OCRFeeder [Электронный ресурс]. – Режим доступа: <https://wiki.gnome.org/action/show/Apps/OCRFeeder>, свободный. – Загл. с экрана (дата обращения: 29.04.2014).
86. Ocropolis [Электронный ресурс]. – Режим доступа: <https://code.google.com/p/ocropolis/>, свободный. – Загл. с экрана (дата обращения: 29.04.2014).
87. Oracle Database [Электронный ресурс]. – Режим доступа: <http://www.oracle.com/ru/products/database/overview/index.html/>, свободный. – Загл. с экрана (дата обращения: 29.09.2014).
88. Package: hunspell-tools [Электронный ресурс]. – Режим доступа: <https://packages.debian.org/sid/text/hunspell-tools>, свободный. – Загл. с экрана (дата обращения: 16.11.2014).
89. Pattern (Java Platform SE 7) [Электронный ресурс]. – Режим доступа: <http://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html>, свободный. – Загл. с экрана (дата обращения: 04.10.2014).
90. Perez-Cortes, J. Stochastic Error-Correcting Parsing for OCR Post-Processing / J. Perez-Cortes, J. Amengual, J. Arlandis, R. Llobet // Proceedings of the International Conference on Pattern Recognition (ICPR '00). – 2000. – pp. 405–408.
91. Philips, L. The Double Metaphone Search Algorithm / L. Philips // C/C++ Users Journal. – 2000. – vol. 8, no. 6. – pp. 38–43.
92. Pollock, J. Automatic Spelling Correction in Scientific and Scholarly Text / J. Pollock, A. Zamora // Commun. ACM. – 1984. – vol. 27, no. 4. – pp. 358–368.
93. PostgreSQL Database [Электронный ресурс]. – Режим доступа: <http://www.postgresql.org/>, свободный. – Загл. с экрана (дата обращения: 29.09.2014).
94. Postnikov, V.V. Post-processing of OCR Results Using Automatically Constructed Partially Defined Syntax / V.V. Postnikov, D.L. Sholomov // Proceedings of the

- International Conference on Machine Learning, Technologies and Applications. – 2004. – pp. 814–820.
95. Regex Tutorial - Unicode Characters and Properties [Электронный ресурс]. – Режим доступа: <http://www.regular-expressions.info/unicode.html>, свободный. – Загл. с экрана (дата обращения: 04.10.2014).
96. Reynaert M. Text Induced Spelling Correction : диссертация / Martin William Christian Reynaert. – Enschede: PrintPartners Ipskamp, 2005. – 203 p.
97. Reynaert, M. Corpus-Induced Corpus Clean-up / M. Reynaert // Fifth International Conference on Language Resources and Evaluation (LREC '2006). – 2006.
98. Reynaert, M. Non-interactive OCR Post-correction for Giga-Scale Digitization Projects / M. Reynaert // Computational Linguistics and Intelligent Text Processing. – 2008. – pp. 617–630.
99. Reynaert, M. Text Induced Spelling Correction / M. Reynaert // Proceedings of the 20th international conference on Computational Linguistics (COLING '04). – 2004. – pp. 834–841.
100. Rusell, R.C. Patent Numbers, 1,261,167 (1918) and 1,435,663 (1922): Technical report / R.C. Rusell, M.K. Odell. – Washington : Patent Office. – p. 67.
101. Russian morphology for lucene - Google Project Hosting [Электронный ресурс]. – Режим доступа: <https://code.google.com/p/russianmorphology/>, свободный. – Загл. с экрана (дата обращения: 05.10.2014).
102. Schaback, J. Multi-Level Feature Extraction for Spelling Correction / J. Schaback, F. Li // Workshop on Analytics for Noisy Unstructured Text Data (IJCAI-2007). – 2007. – pp. 79–86.
103. SimpleOCR [Электронный ресурс]. – Режим доступа: <http://www.simpleocr.com/>, свободный. – Загл. с экрана (дата обращения: 29.04.2014).
104. Spring Framework [Электронный ресурс]. – Режим доступа: <http://projects.spring.io/spring-framework/>, свободный. – Загл. с экрана (дата обращения: 29.09.2014).

105. Strohmaier, C.M. Methoden der lexikalischen Nachkorrektur OCR-erfasster Dokumente : Ph.D. thesis / Christian M. Strohmaier. – Munich, 2004. – 158 p.
106. Taghva, K. OCRSpell: an interactive spelling correction system for OCR errors in text / K. Taghva, E. Stofsky // International Journal of Document Analysis and Recognition. – 2001. – vol. 3. – pp. 125–137.
107. Tanner S. Deciding Whether Optical Character Recognition is Feasible. 2004 [Электронный ресурс]. – Режим доступа: [http://www.odl.ox.ac.uk/papers/OCRFeasibility\\_final.pdf](http://www.odl.ox.ac.uk/papers/OCRFeasibility_final.pdf), свободный. – Загл. с экрана (дата обращения: 06.06.2012)
108. Tesseract-ocr [Электронный ресурс]. – Режим доступа: <http://code.google.com/p/tesseract-ocr/>, свободный. – Загл. с экрана (дата обращения: 29.04.2014).
109. The hOCR Embedded OCR Workflow and Output Format [Электронный ресурс]. – Режим доступа: [https://docs.google.com/document/d/1QQnIQtvdAC\\_8n92-LhwPcjtaUFwBlzE8EWnKAxlgVf0/preview#](https://docs.google.com/document/d/1QQnIQtvdAC_8n92-LhwPcjtaUFwBlzE8EWnKAxlgVf0/preview#), свободный. – Загл. с экрана (дата обращения: 18.09.2014)
110. Tong, X. A Statistical Approach to Automatic OCR Error Correction In Context / X. Tong, D. Evans // Proceedings of the Fourth Workshop on Very Large Corpora (WVLC-4). – 1996. – pp. 88–100.
111. Transym OCR [Электронный ресурс]. – Режим доступа: <http://www.transym.com>, свободный. – Загл. с экрана (дата обращения: 29.04.2014).
112. Viterbi, A.J. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm / A.J. Viterbi // IEEE Transactions on Information Theory. – 1967. – vol. 13, no. 2. – pp. 260–269.
113. Volk, M. Strategies for reducing and correcting OCR error / M. Volk, L. Furrer, R. Sennrich // Language Technology for Cultural Heritage. – Berlin: Springer Berlin Heidelberg. – 2011. – pp. 3–22.

114. Wagner, R.A. The String-to-String Correction Problem / R.A. Wagner, M.J. Fischer // J. ACM. – 1974. – vol. 21, no. 1. – pp. 168–173.
115. Wick, M. Context-Sensitive Error Correction: Using Topic Models to Improve OCR / M. Wick, M. Ross, E. Learned-Miller // Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR '07). – 2007. pp. – 1168–1172.

## Приложение А. Примеры графического интерфейса системы

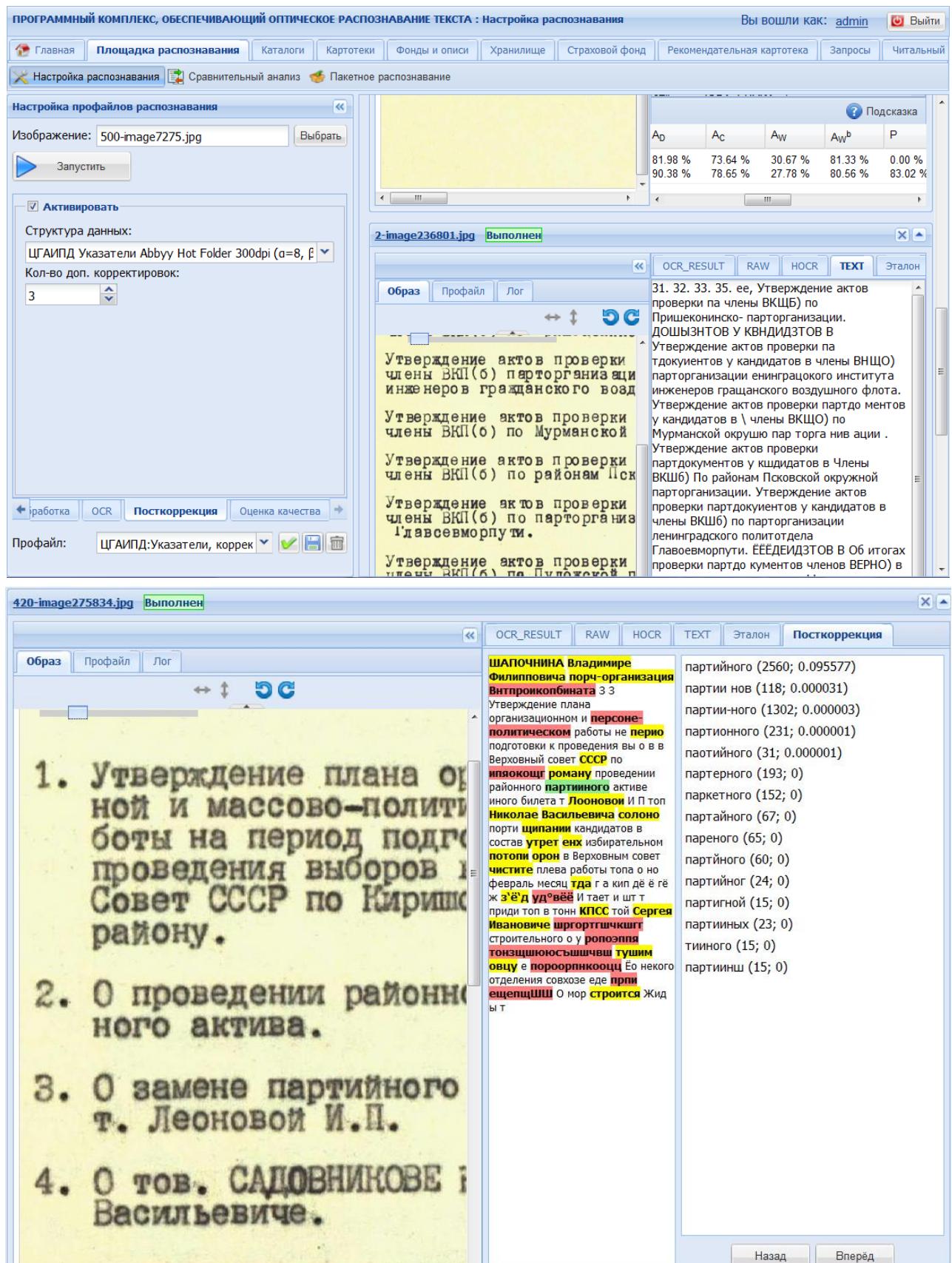


Рисунок А.1. Графический интерфейс программного модуля настройки профайлов

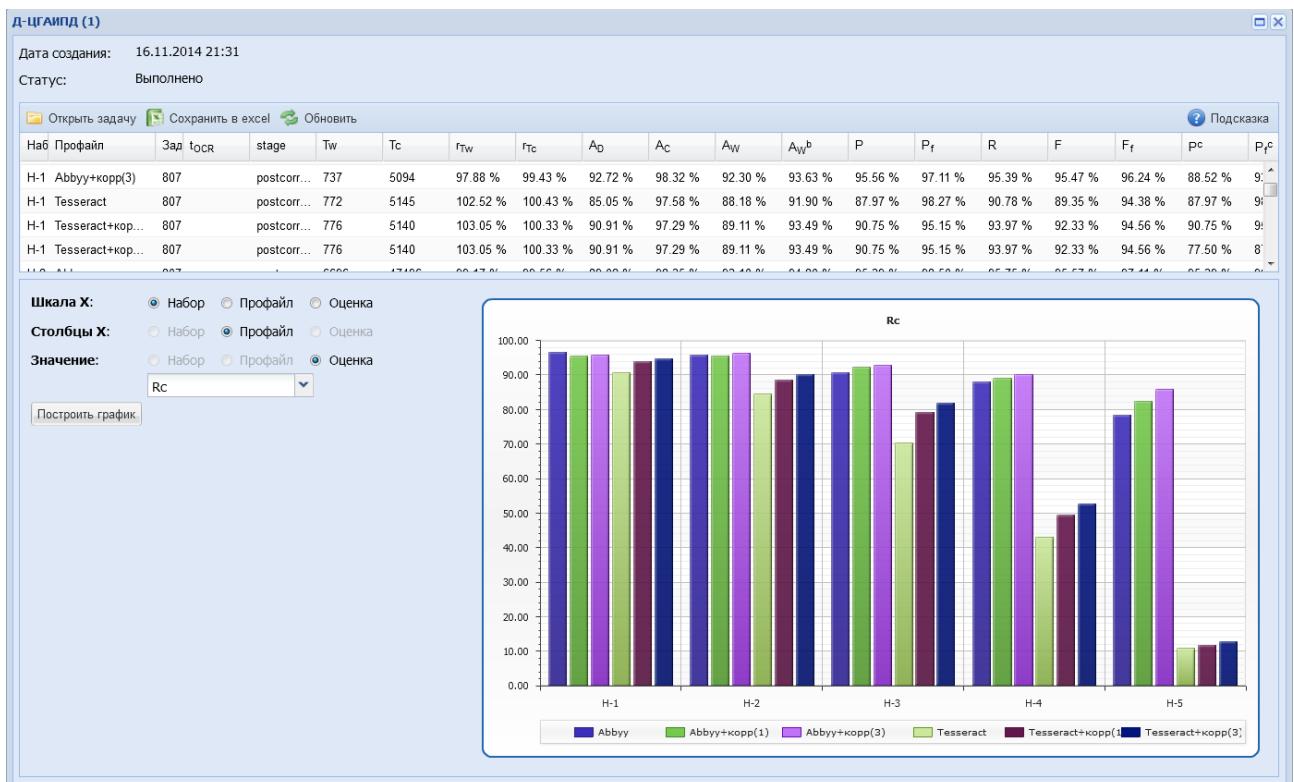


Рисунок А.2. Графический интерфейс программного модуля сравнительного анализа

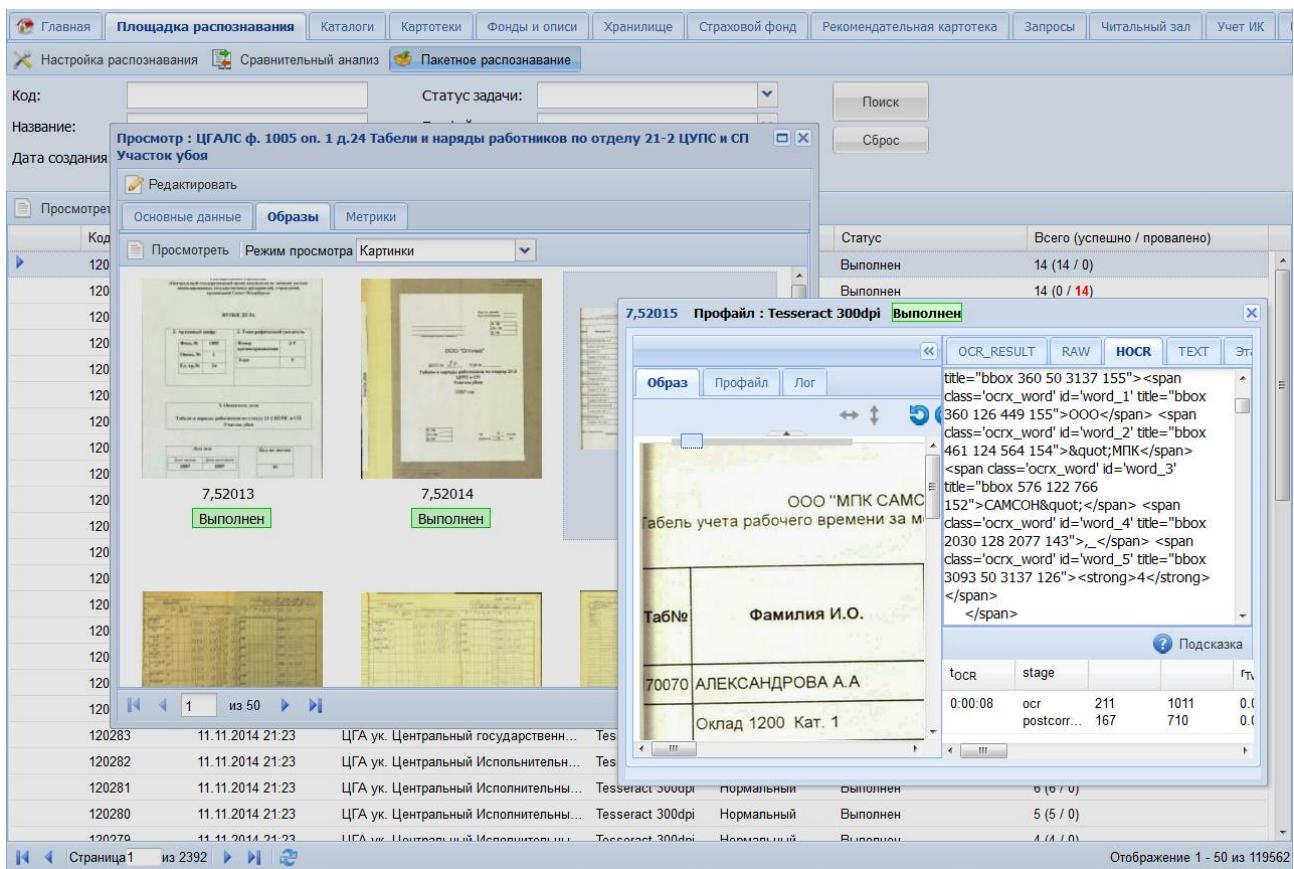


Рисунок А.3. Графический интерфейс программного модуля пакетного распознавания

**Приложение Б. Свидетельства о государственной регистрации**

**РОССИЙСКАЯ ФЕДЕРАЦИЯ**



**СВИДЕТЕЛЬСТВО**

о государственной регистрации программы для ЭВМ

**№ 2014662557**

**Программный комплекс «Формирование метаданных» ГИС  
«Государственные архивы Санкт-Петербурга»**

Правообладатель: *Санкт-Петербург, от имени которого выступает  
Комитет по информатизации и связи (RU)*

Авторы: *Смирнов Сергей Владимирович (RU), Кожин Александр  
Владимирович (RU), Воронцов Артем Валерьевич (RU),  
Белозерова Марина Вячеславовна (RU)*

Заявка № **2014660387**

Дата поступления **14 октября 2014 г.**

Дата государственной регистрации

в Реестре программ для ЭВМ **03 декабря 2014 г.**

Врио руководителя Федеральной службы  
по интеллектуальной собственности

Л.Л. Кирай



Государственная служба по интеллектуальной собственности Российской Федерации

Рисунок Б.1. Программный комплекс «Формирование метаданных»  
ГИС «Государственные архивы Санкт-Петербурга»

# РОССИЙСКАЯ ФЕДЕРАЦИЯ



## СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

**№ 2014662676**

**Программный комплекс «Информационно-лингвистическое обеспечение» ГИС «Государственные архивы Санкт-Петербурга»**

Правообладатель: *Санкт-Петербург, от имени которого выступает Комитет по информатизации и связи (RU)*

Авторы: *Смирнов Сергей Владимирович (RU), Кожин Александр Владимирович (RU), Воронцов Артем Валерьевич (RU), Белозерова Марина Вячеславовна (RU)*

Заявка № **2014660310**

Дата поступления **14 октября 2014 г.**

Дата государственной регистрации

в Реестре программ для ЭВМ **05 декабря 2014 г.**

*Врио руководителя Федеральной службы  
по интеллектуальной собственности*

A handwritten signature in black ink, appearing to read "Л.Л. Кирий".

Л.Л. Кирий



Рисунок Б.2. Программный комплекс «Информационно-лингвистическое обеспечение»  
ГИС «Государственные архивы Санкт-Петербурга»

## Приложение В. Акты внедрения



ПРАВИТЕЛЬСТВО САНКТ-ПЕТЕРБУРГА

АРХИВНЫЙ КОМИТЕТ САНКТ-ПЕТЕРБУРГА

АКТ

№



УТВЕРЖДАЮ

Председатель Комитета

С.В. Штукова  
2014г

### АКТ

о внедрении результатов диссертационных исследований

Смирнова Сергея Владимировича на тему

«Технология и система автоматической корректировки результатов при  
распознавании архивных документов»

Настоящим подтверждается, что результаты диссертационного исследования внедрены в составе государственной информационной системы «Государственные архивы Санкт-Петербурга», функционирующей в Архивном комитете Санкт-Петербурга и семи подведомственных ему центральных государственных архивах, в качестве программных комплексов, обеспечивающих формирование метаданных, информационно-лингвистическое обеспечение и оптическое распознавание текста.

Результаты диссертационного исследования, используемые в системе, позволили произвести массовое распознавание накопленного массива электронных образов научно-справочного аппарата архивов и используются в настоящее время для потокового распознавания вновь загружаемых изображений архивных документов в автоматическом режиме с целью расширения поискового пространства и пополнения мета базы данных.

Применение предложенных в диссертационной работе подходов к распознаванию и поиску архивных документов предоставило сотрудникам и посетителям архивов и Архивного комитета Санкт-Петербурга новые возможности для эффективного поиска информации.

Заместитель председателя Комитета

М.В. Мишенкова

Начальник сектора автоматизированных  
архивных технологий и информатизации

П.А. Крылов

Рисунок В.1. Акт о внедрении в Архивном комитете СПб



## АРХИВНЫЙ КОМИТЕТ САНКТ-ПЕТЕРБУРГА

САНКТ-ПЕТЕРБУРГСКОЕ ГОСУДАРСТВЕННОЕ  
КАЗЕННОЕ УЧРЕЖДЕНИЕ  
«ЦЕНТРАЛЬНЫЙ ГОСУДАРСТВЕННЫЙ АРХИВ  
ДОКУМЕНТОВ ПО ЛИЧНОМУ СОСТАВУ  
ЛИКВИДИРОВАННЫХ ГОСУДАРСТВЕННЫХ  
ПРЕДПРИЯТИЙ, УЧРЕЖДЕНИЙ,  
ОРГАНИЗАЦИЙ САНКТ-ПЕТЕРБУРГА»  
(ЦГАЛС СПб)

Днепропетровская ул., 9-А, Санкт-Петербург, 191119  
Тел. (812) 572-12-57, Факс (812) 572-12-57.  
ОКПО 80569659 ОГРН 1027809227445  
ИНН / КПП 7842013557/784201001

УТВЕРЖДАЮ

«16» декабря 2014г.

## АКТ

о внедрении результатов диссертационных исследований на тему  
«Технология и система автоматической корректировки результатов при  
распознавании архивных документов»

Полученные в диссертационном исследовании Смирнова Сергея Владимировича «Технология и система автоматической корректировки результатов при распознавании архивных документов» результаты внедрены в рамках работ по автоматизации архивной деятельности и адаптации государственной информационной системы «Государственные архивы Санкт-Петербурга» под потребности архива, проводимых СПб ГУП «Санкт-Петербургский информационно-аналитический центр».

Специфика архива подразумевает наличие в содержании дел и научно-справочного аппарата большого количества наименований предприятий, специальностей, должностей, имен собственных и других узкоспециализированных терминов, в связи, с чем предложенные в диссертационном исследовании методы и алгоритмы автоматической корректировки ошибок, основанные на тематических тезаурусах, являются особенно актуальными и эффективными для обработки документов архива.

Внедрение предложенных в исследовании систем распознавания и поиска архивных документов значительно облегчило процессы обнаружения необходимой информации, что положительно отразилось на эффективности исполнения запросов, поступающих от граждан и организаций.

Заведующий отделом комплектования,  
обеспечения сохранности и использования  
документов (1-Д)

О.Н. Пейбо

Рисунок В.2. Акт о внедрении в ЦГАЛС СПб



Комитет по информатизации и связи  
Санкт-Петербургское государственное  
унитарное предприятие  
**“Санкт-Петербургский**  
**информационно-аналитический центр”**  
191040, Санкт-Петербург,  
Транспортный пер., д.6, литер А, пом. 7Н 8Н  
Тел. (812) 764-3957,  
факс (812) 764-9548,  
e-mail: secretar@iac.spb.ru

### УТВЕРЖДАЮ

Директор СПб ГУП «СПб ИАЦ»

Е.В. Корабельников



2014г.

### АКТ

#### внедрения результатов диссертационных исследований на тему: «Технология и система автоматической корректировки результатов при распознавании архивных документов»

Полученные в диссертационном исследовании Смирнова Сергея Владимировича «Технология и система автоматической корректировки результатов при распознавании архивных документов» результаты внедрены по следующим пунктам:

1) В рамках работ по развитию государственной информационной системы «Государственные архивы Санкт-Петербурга» использованы результаты диссертационного исследования в полном объеме. Реализованные системы распознавания архивных документов, автоматической корректировки результатов и поиска изображений архивного фонда позволили успешно произвести обработку более миллиона электронных образов архивных документов, увеличить размер поискового покрытия на несколько порядков и предоставить пользователям эффективный поисковый инструментарий, работоспособность которого не требует проведения трудоемкой и дорогостоящей подготовительной ручной работы.

2) В рамках работ по автоматизации процесса комплектования Архивного фонда Санкт-Петербурга для повышения качества результатов распознавания архивных документов и сокращения количества допускаемых ошибок используется предложенный в диссертационной работе инструментарий, позволяющий экспертам производить настройку системы для наиболее эффективного решения задачи распознавания, путем создания профилей под каждый класс документов. Применение данного инструментария позволяет использовать для корректировки ошибок различные тематические тезаурусы и производить сравнительный анализ качества результатов различными профилями, опираясь на широкий спектр автоматически вычисляемых показателей и критериев точности и качества распознавания. Применение предложенных в диссертационной работе методов автоматического обнаружения ошибок, генерации, ранжирования и отбора корректировок способствовало существенному снижению затрат на ручную корректировку, а для некоторых классов документов и полному отсутствию необходимости в ней.

Первый заместитель директора,  
кандидат технических наук, профессор

Ю.Н. Захаров

Начальник отдела разработки и проектирования  
электронных архивов

М.В. Белозёрова

Рисунок В.3. Акт о внедрении в СПб ГУП «СПб ИАЦ»