

## Инкрементная разметка рукописных архивных дневников XIX века

Л.М. Местецкий<sup>1, 2✉</sup>, В.П. Зыков<sup>1</sup>

<sup>1</sup> МГУ имени М.В. Ломоносова, г. Москва, 119991, Россия

<sup>2</sup> Высшая школа экономики, г. Москва, 101000, Россия

### Ссылка для цитирования

Местецкий Л.М., Зыков В.П. Инкрементная разметка рукописных архивных дневников XIX века // Программные продукты и системы. 2025. Т. 38. № 4. С. 668–681. doi: 10.15827/0236-235X.152.668-681

### Информация о статье

Группа специальностей ВАК: 2.3.5

Поступила в редакцию: 07.07.2025

После доработки: 28.07.2025

Принята к публикации: 30.07.2025

**Аннотация.** В статье рассматривается проблема подготовки данных для машинного обучения алгоритмов распознавания старых рукописных текстов. Исследование построено на архивных материалах дневников XIX века, составляющих национальное культурное наследие. В современных архивах хранятся дневники, содержащие тысячи электронных копий страниц, полученных путем сканирования бумажных рукописей. Особые свойства этих документов – высокий культурный уровень авторов, большие объемы рукописей, единый почерк, единый стиль форматирования текста. Для обучения алгоритмов распознавания требуется подготовить разметку рукописи, состоящую в максимально точном дословном переводе некоторой части текста. Такой перевод выполняется экспертом-ассессором высокой квалификации, специалистом по творческому наследию автора дневника. Но даже для него разметка – весьма трудоемкий, наиболее затратный по времени элемент процесса автоматизации работы с архивными рукописными текстами. Целью данного исследования является разработка новой технологии работы эксперта-ассессора, позволяющей существенно сократить трудозатраты на разметку рукописных дневников. Предлагается подход, основанный на итерационном наращивании массива размеченных данных небольшими последовательными порциями. Точная разметка очередной порции рукописи получается в два этапа. Сначала выполняется ее автоматическая расшифровка обученным ранее алгоритмом, после чего эксперт делает точный подстрочный перевод, исправляя в режиме редактирования ошибки алгоритма расшифровки. Полученная экспертом точная разметка используется далее для очередной итерации обучения алгоритма. Предложенный подход реализован в виде программного комплекса «Подстрочник», предназначенного для работы специалиста-гуманитария. Программа поддерживает полный цикл операций по разметке данных для обучения алгоритмов расшифровки рукописного текста: автоматический перевод и последующее его редактирование для получения точной разметки, дообучение. Программа прошла практическую проверку при работе с дневниками Ф.П. Литке и А.В. Сухова-Кобылина.

**Ключевые слова:** машинное обучение, инкрементная разметка, рукописный текст, расшифровка рукописи, подстрочный перевод

**Благодарности.** Работа выполнена в рамках проекта РНФ № 22-68-00066

**Введение.** В российских архивах хранятся тысячи страниц рукописей дневников писателей, ученых, военачальников, составляющих важную часть национального культурного наследия. Высокий культурный уровень авторов определяет постоянный интерес к дневникам со стороны гуманитарных исследователей – филологов и историков. Через их труд культурное наследие становится доступным для широкого круга заинтересованных читателей. Работа с рукописными дневниками является весьма трудоемкой из-за большого объема текстов, старой орфографии, сложных почерков, дефектов, присущих древним бумажным документам. Поэтому проблема автоматизации перевода рукописных документов в редактируемые электронные версии является весьма актуальной. Такое преобразование текста будем называть расшифровкой. Оно значительно упрощает и ускоряет понимание, поиск и навигацию при работе с рукописями.

Постановка задачи расшифровки архивных рукописных документов существенно зависит от нескольких важных условий, связанных с разнообразием почерков и форматов записей, а также с объемом словаря. Например, для проведения генеалогических исследований, в которых заинтересованы многие организации и частные лица, выполняется расшифровка метрических книг, ревизских сказок. Записи в этих документах выполнены разными почерками в одинаковом формате, они используют относительно небольшой словарь. Другой вариант постановки – расшифровка, аннотирование, хранение персональных архивов выдающихся людей, например, Петра Первого, А.С. Пушкина. Здесь тексты написаны одним автором, они имеют разнообразие форматов и большой словарь.

С этой точки зрения рукописные дневники являются особым материалом. С одной сто-

роны, это текст большого объема, написанный одним почерком, имеющий единый формат, что упрощает задачу распознавания. С другой стороны, высокий культурный уровень автора и особенности его стиля приводят к тому, что разбирать и расшифровывать текст может только профессиональный исследователь, работающий с наследием данного автора. Кроме того, такого рода исследования имеют исключительно научные некоммерческие цели и мотивируются в основном научным энтузиазмом исследователя.

Таким образом, особенностями работы с дневниками являются огромный объем рукописного текста и очень узкий круг специалистов, способных в этом тексте разобраться и, главное, желающих это сделать. Особые свойства дневников – большие объемы рукописей, написанных одним почерком и имеющих единый стиль форматирования, – дают надежду на создание обучаемых алгоритмов распознавания их текстов. Обучение алгоритмов расшифровки может быть выполнено только самими специалистами. Поэтому разработка программных средств для расшифровки таких рукописей должна быть направлена на создание универсальной программы, настраиваемой на конкретные дневники с помощью машинного

обучения. При этом обучение и эксплуатация программы осуществляется высококвалифицированными исследователями.

Необходимым элементом машинного обучения является разметка данных. Для расшифровки рукописных текстов такой разметкой является точный перевод некоторой части рукописи. На основе разметки осуществляется обучение алгоритмов распознавания, то есть настройка алгоритмов на конкретные свойства рукописи – почерк, стиль, словарь и т.п. Постоянство этих свойств в рукописных дневниках дает основание для успешного решения задачи расшифровки всего дневника. На рисунке 1 показано место разметки в общем процессе автоматической расшифровки рукописного дневника. Результатом разметки фрагмента рукописного текста является дословный перевод. В данной статье он называется подстрочным, или подстрочником, что согласуется с традиционным использованием этого термина [1]. Подстрочный перевод исторических документов в науке широко распространен, поскольку в переводе важна точность, а не стиль.

Главной фигурой процесса разметки дневников является эксперт-ассессор (филолог или историк, специалист по творчеству, эпохе и среде общения автора дневника), чья квалификация



Рис. 1. Схема процесса обучения алгоритма расшифровки рукописного текста

Fig. 1. Flowchart of the handwritten text recognition algorithm training process

позволяет расшифровывать рукопись. Выполненную экспертом расшифровку какого-либо рукописного фрагмента будем называть точной разметкой. Расшифровка с помощью компьютера неизбежно содержит ошибки, которых особенно много на первых порах обучения алгоритма, для удобства терминологии будем называть ее слабой разметкой.

Подготовка качественного подстрочника весьма трудоемкая, она требует высокой квалификации эксперта-ассессора. Опыт показывает, что для обучения алгоритма распознавания в случае хорошего каллиграфического почерка автора достаточно подготовить качественный подстрочный перевод тысячи строк [2]. Однако если почерк менее разборчив, может потребоваться намного больше, до десяти тысяч строк [3]. Такая работа занимает очень много времени. В [4] приводится оценка доли трудозатрат на разметку данных в общем объеме работ при решении исследовательских задач. Из общего времени, затраченного на решение конкретной задачи с помощью нейронной сети, до 90 % может уйти на подготовку данных вручную для их последующего использования в рамках реализации корректного процесса обучения.

До настоящего времени проблеме разметки рукописных текстов не уделялось достаточного внимания. Главные усилия разработчиков ПО были сосредоточены на методах обучения нейросетей, на повышении качества расшифровки и скорости работы. При этом предполагалось, что качественные данные для обучения существуют в достаточном объеме. Однако опыт успешных решений показал, что обученная нейросеть, позволившая расшифровать рукописи одного автора, требует нового обучения для рукописей других авторов. При этом разметка рукописи для такого обучения выполняется экспертом вручную и занимает основную часть времени всего процесса расшифровки. Поэтому задача разработки методов автоматизации разметки рукописи выходит на первый план при создании информационных технологий для работы с архивными рукописными документами.

В статье предлагается новый подход, который обеспечивает минимизацию трудозатрат эксперта-ассессора при разметке рукописных дневников. Подход основан на итерационном наращивании массива размеченных данных небольшими последовательными порциями. Сокращение трудозатрат достигается за счет использования экспертом-ассессором компьютерной расшифровки, полученной с помощью

алгоритма распознавания, ранее обученного на предыдущей порции текста.

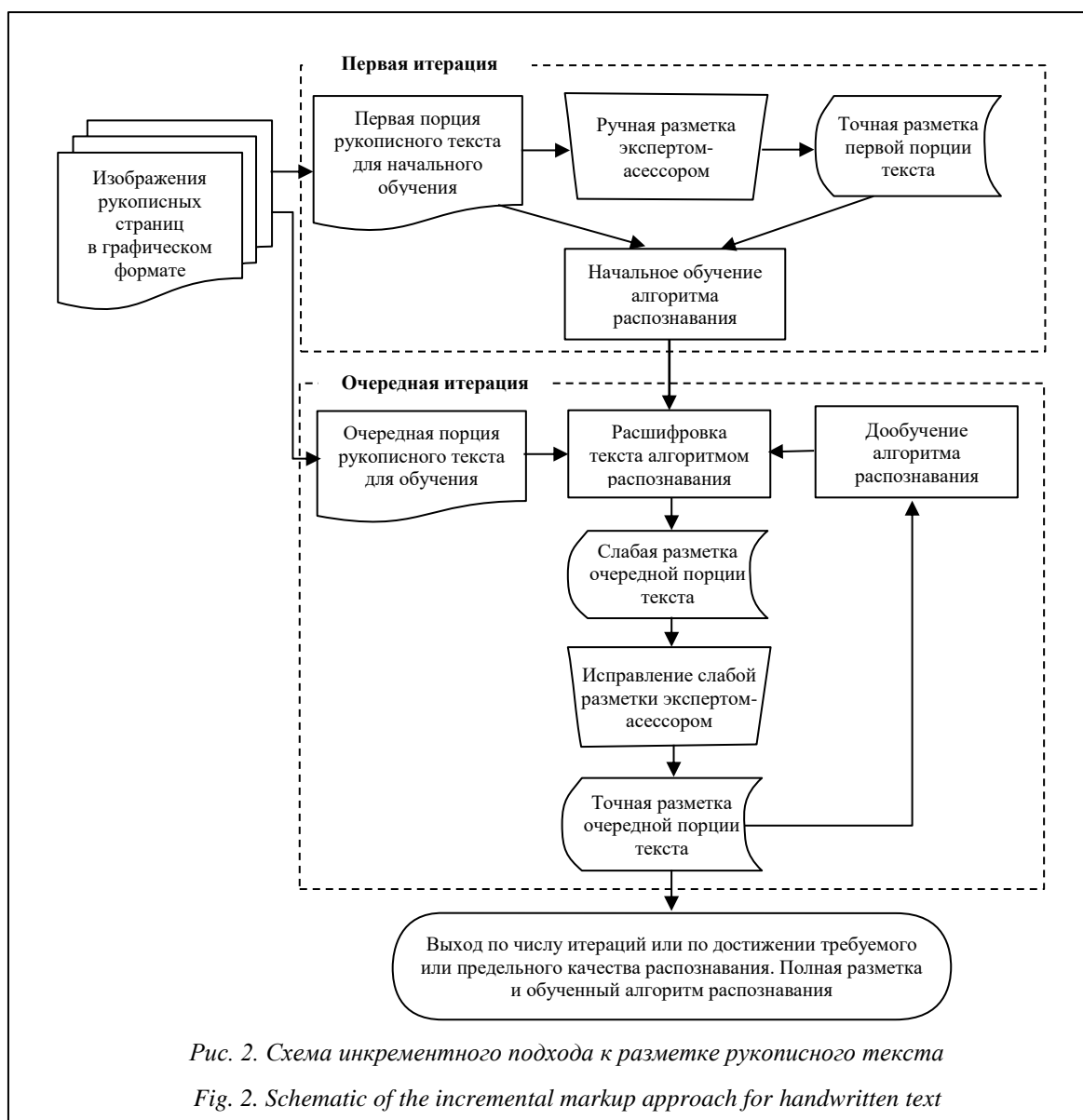
Для обоснования и проверки предлагаемого решения использованы реальные архивные документы: наборы электронных сканированных страниц рукописных дневников Ф.П. Литке [4] и А.В. Сухова-Кобылина [5]. Результатами разметки являются расшифрованные страницы дневника, форматированные синхронно с исходными изображениями страниц и строк, представленные в виде текстовых файлов.

Работа с архивами рукописных документов обычно выполняется небольшими группами профессиональных исследователей – историков и филологов. При этом важную роль играет непосредственный визуальный анализ оригиналов рукописных страниц или высококачественных электронных копий, полученных с помощью сканеров. Кроме собственно текста записей, большое значение для исследователя имеют многочисленные детали рукописи: форматирование строк и слов, вставки, зачеркивания, подчеркивания, цвет чернил и карандашей и многие другие авторские элементы стиля, присущие рукописному тексту. Анализ текста с учетом всей совокупности таких деталей осложняется большим объемом рукописей. Поэтому задача автоматизации такого анализа с использованием информационных технологий является весьма актуальной при работе с большими массивами рукописных документов.

### Инкрементный подход

Предлагаемый подход к разметке состоит в последовательном наращивании массива размеченных данных небольшими порциями. Сокращение трудозатрат эксперта-ассессора достигается за счет комбинированного использования слабой и точной разметки в процессе подстрочного перевода рукописи. Общая структура подхода представлена на рисунке 2.

Первая порция текста расшифровывается экспертом полностью вручную. Эта расшифровка является точной разметкой и служит для начального обучения алгоритма. Далее выполняется несколько итераций наращивания размеченного текста. На каждой итерации новая порция рукописного текста размечается в два этапа. На первом этапе выполняется ее автоматическая расшифровка с помощью уже обученного алгоритма. Полученный подстрочный перевод является слабой разметкой, которую на втором этапе корректирует эксперт-ассессор. Он исправляет ошибки, допущенные алгорит-



мом, в результате чего создается точная разметка этой порции текста. После этого выполняется дообучение алгоритма с использованием полученной точной разметки. В результате с каждой очередной итерацией уровень ошибок алгоритма распознавания уменьшается. При этом работа эксперта по корректировке слабой разметки требует существенно меньших затрат, чем расшифровка рукописного текста с нуля.

Инкрементный подход позволяет последовательно наращивать объем разметки и снижать уровень ошибок. Кроме того, он предоставляет простой критерий для определения необходимого числа итераций обучения алгоритма распознавания. Итерационный процесс разметки завершается, когда очередной цикл

обучения алгоритма не дает значимого снижения уровня ошибок.

### Строчная технология разметки

Модели распознавания рукописного текста имеют две разновидности – страничную и строчную. Страничная модель принимает на вход изображение всей страницы текста целиком, а строчная – изображение рукописной строки, на выходе выдает ее подстрочный перевод в виде текстовой строки. Применительно к рассматриваемой задаче разметки строчная модель имеет несколько преимуществ. Во-первых, некоторые строки на рукописной странице не удается разобрать и разметить даже эксперту (или разметка этих строк для него просто

не представляет научного интереса), в то время как остальные строки на этой странице хорошо читаются. В этом случае непонятно, как использовать такую не до конца размеченную страницу для обучения страничной модели. Строчная модель такой проблемы не имеет, так как использует для обучения отдельные размеченные строки. Во-вторых, страничная модель не всегда справляется с проблемой сегментации строк в том случае, когда они имеют сложную криволинейную форму. Строчная модель может в этом случае игнорировать плохо сегментированные строки на изображении страницы. В-третьих, для эксперта-ассессора страница является слишком большим фрагментом текста, обычно в процессе получения подстрочного перевода он оперирует понятиями строк и отдельных слов.

В проводимых авторами экспериментах для распознавания рукописного текста используется строчный вариант модели VAN (*Vertical Attention Network*, вертикальная сеть внимания), которая описана в [6]. Эта модель показывает лучшие результаты на открытых датасетах и хорошо себя зарекомендовала при распознавании дневников Ф.П. Литке и А.В. Сухово-Кобылина.

Базовые элементы подстрочного перевода рукописного текста составляют понятия рукописной и текстовой строк (рис. 3). Рукописной строкой является фрагмент изображения страницы исходной рукописи в графическом формате. Текстовая строка – это результат расшифровки рукописной строки в формате текстового файла. Подстрочный перевод заключается в получении текстовой строки для каждой рукописной. Такой перевод может быть получен разными путями: ассессором вручную, обученным алгоритмом расшифровки, а также комбинацией алгоритмической и экспертной расшифровки. Для удобства терминологии будем называть эти варианты ручной, автоматической и редакторской разметкой соответственно.

При ручной разметке ассессор, имея перед глазами рукописную строку, расшифровывает

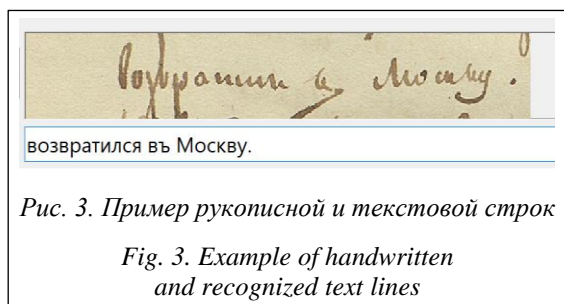


Рис. 3. Пример рукописной и текстовой строк

Fig. 3. Example of handwritten and recognized text lines

ее содержание и вводит текстовую строку с клавиатуры компьютера с помощью текстового редактора. Результатом является точная разметка строки.

Автоматическая разметка состоит в формировании текстовой строки с помощью алгоритма расшифровки. При этом получаемая текстовая строка может содержать различные ошибки и неточности. Такая разметка называется слабой.

Редакторская разметка выполняется ассессором на основе анализа рукописной строки и текстовой, полученной алгоритмом расшифровки. Найденные ошибки ассессор исправляет вручную в текстовом редакторе. Полученная разметка также является точной.

Источником повышения эффективности процесса разметки в рамках предлагаемого инкрементного подхода является использование редакторской разметки вместо традиционной ручной.

Таким образом, процесс разметки состоит из следующих элементов:

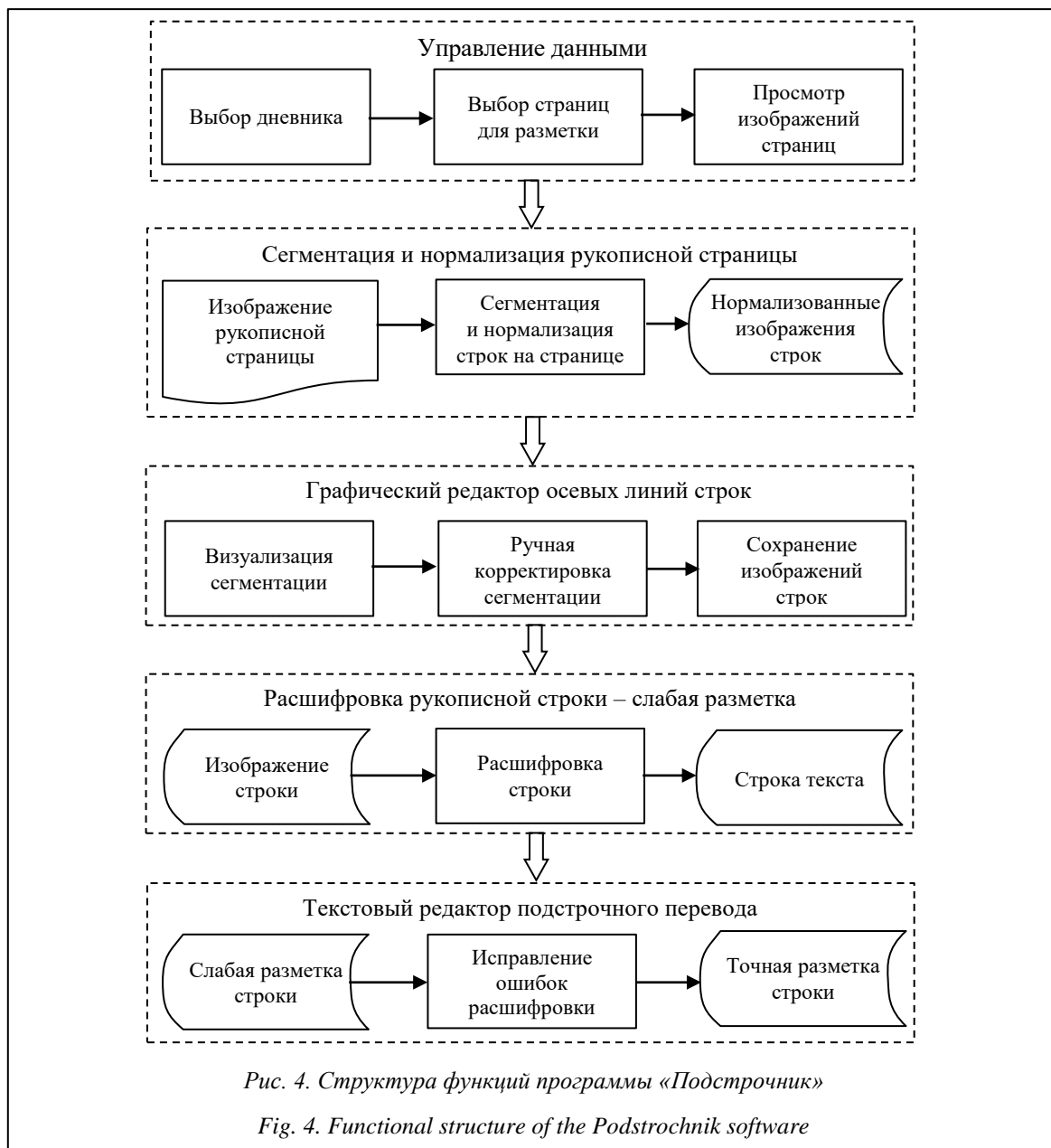
- 1) сегментация строк на исходных изображениях страниц – выделение в исходном изображении страницы изображений отдельных рукописных строк;
- 2) ручная разметка рукописных строк на первой начальной итерации инкрементного подхода;
- 3) автоматическая разметка строк на последующих итерациях;
- 4) редакторская разметка рукописных строк путем исправления ошибок автоматической разметки в текстовых строках.

Выполнение пунктов 1 и 3 предполагает построение алгоритмов решения соответствующих задач. Выполнение пунктов 2 и 4 осуществляется экспертом-ассессором. Для этого разработана программа-редактор «Подстрочник», обеспечивающая работу ассессора с изображениями страниц и текстовых расшифровок. Программа поддерживает реализацию всего процесса итерационного наращивания разметки в соответствии с предложенным инкрементным подходом.

### Описание программы «Подстрочник»

Программа «Подстрочник» представляет собой редактор для работы эксперта-ассессора при подготовке подстрочника в формате текстовых файлов.

Основные функции программы, обеспечивающие операционные потребности эксперта-ассессора, иллюстрирует рисунок 4:



– синхронизация – обеспечение строгого соответствия страниц и строк подстрочного перевода с исходным изображением рукописного текста, необходимого для адекватного анализа и исправления ошибок экспертом;

– сегментация страниц рукописного текста, включающая выделение строк на изображении, формирование отдельных растровых изображений строк и нормализацию этих изображений путем приведения к стандартному формату, который определяется требованиями алгоритма автоматической расшифровки рукописи;

– редактирование – вставка и удаление строк, внесение правки в текст подстрочного

перевода в случае обнаружения ошибок сегментации и слабой расшифровки; при этом следует обеспечить использование алфавита старорусского письма, соответствующего оригиналу дневника, поскольку подстрочный перевод осуществляется в том алфавите, который использовал автор;

– автоматическая расшифровка строчек рукописи: на каждой итерации инкрементного подхода выполняется дообучение алгоритма расшифровки, поэтому расшифровка на очередной итерации процесса разметки осуществляется с помощью той версии алгоритма, которая была получена в результате обучения на предыдущей итерации;

– визуализация всех этапов процесса разметки для обеспечения прозрачности и интерпретируемости каждой из операций, выполняемых экспертом при подготовке подстрочного перевода;

– общие технологические возможности текстового редактора: ввод, сохранение, коррекция данных, приведение текстов и изображений в формат, принятый в алгоритмах расшифровки, визуализация всех операций по подготовке подстрочного перевода для обеспечения прозрачности и интерпретируемости разметки.

В список функций «Подстрочника» не входит обучение нейросетевого алгоритма расшифровки, используемого в блоке 4. Обучение выполняется на каждой итерации инкрементного подхода. Этот процесс может потребовать значительных вычислительных ресурсов, поэтому он реализуется автономно на сервере. А программа «Подстрочник» ориентирована на работу исследователя на персональном компьютере или ноутбуке. Кроме того, сами алгоритмы расшифровки и механизмы их обучения могут быть различными. Чтобы обеспечить возможность использования разных алгоритмов, в «Подстрочнике» предусмотрен интерфейс взаимодействия с алгоритмом расшифровки. Интерфейс основан на фиксированных форматах обмена данными между «Подстрочником» и алгоритмом обучения нейросети. Со стороны «Подстрочника» формируется датасет размеченных рукописных строчек, а со стороны программы обучения в редактор передается набор данных, описывающий структуру и весовые коэффициенты нейросети. Таким образом, алгоритм расшифровки, используемый в блоке 4, присутствует в «Подстрочнике» в виде массива весовых коэффициентов нейросети, полученного от сервера, на котором происходит обучение.

Программа «Подстрочник» разработана в интегрированной среде Delphi Pascal [7] с широким использованием инструментов для визуального проектирования пользовательского интерфейса. Мощност и эффективность этих инструментов позволили выполнить быстрое проектирование, разработку и отладку программы. Программа работает с операционной системой Windows 10.

### Сегментация строк рукописи

Строчная технология разметки включает в себя этап сегментации – выделение всех строк

на исходном изображении рукописного текста с тем, чтобы далее сделать подстрочный перевод каждой строки по отдельности. Сегментация строк выглядит как рутинная техническая операция, требующая достаточно большой работы. Во многом она может быть выполнена в автоматическом режиме с помощью соответствующего алгоритма [8, 9]. Однако разнообразие почерков, стилей письма, форматирования рукописей настолько велико, что вряд ли можно рассчитывать на получение универсального алгоритма, который будет безошибочно сегментировать любой рукописный текст. Поэтому в «Подстрочнике» реализован комбинированный двухэтапный подход к сегментации рукописной страницы. Сначала сегментация выполняется алгоритмом, а после этого эксперт просматривает полученные строки и вносит правку, если хочет скорректировать результат. Для такого подхода важным является способ описания сегментации, то есть визуальное представление выделенных строк, пригодное для ручной коррекции экспертом-ассессором.

Обычно результат сегментации представляется в виде полигональных областей, охватывающих строки [9]. Однако в рукописях часто встречаются строки с пересечением выпадающих элементов букв. В таких случаях полигональное описание строк становится неоднозначным и слабо поддается строгой оценке качества. Кроме того, ручная корректировка полигонального описания является слишком сложной. В «Подстрочнике» используется другой способ представления строковой сегментации – в виде осевых линий строк. Осевая линия проходит слева направо от начала к концу строки, пересекая все буквы примерно посередине между нижней и верхней границами базовых линий строки. Автоматическая сегментация строк сводится к вычислению осевых линий строк на изображении страницы. В случае идеального аккуратного письма эти линии представляют собой горизонтальные прямые. Однако в реальной рукописи очень часто строки сильно отклоняются от горизонталей и к тому же имеют криволинейную форму. Это связано как с особенностями авторского стиля и почерка, так и с искажениями, вносимыми при сканировании страниц. Для адекватного представления осей строк в «Подстрочнике» используется описание их в виде ломаных линий, состоящих из отдельных прямолинейных отрезков в количестве от 1 до 16 в зависимости от длины строки. Пример такого представления показан на рисунке 5.

Сегментация строк осуществляется алгоритмом автоматически, что весьма удобно для эксперта. Однако разнообразие индивидуальных форматов письма и качества сканированных рукописей не позволяет гарантированно правильно выполнить автоматическую нарезку строчек во всех случаях. Иногда возможно появление ошибок, таких как пропуски строк, неверное определение длины строк, неточная аппроксимация осей. Для подобных случаев предусмотрен режим ручной коррекции осевых линий строк с помощью нескольких простых операций графического редактирования: перемещение узловых точек осей, а также вставка и удаление оси. Операции используют общепринятую технику графических редакторов.

### Нормализация строк рукописи

Под нормализацией данных при распознавании изображений обычно понимается приведение их к стандартному виду, который определяется требованиями используемых алгорит-

мов распознавания. Нормализация данных для алгоритмов машинного обучения состоит в так называемом распрямлении строк в изображении рукописного текста.

Полученные в результате сегментации страницы осевые линии строк дают возможность нормализовать строки путем преобразования исходного изображения каждой строки, при котором кусочно-линейная ломаная линия оси превращается в горизонтальный прямолинейный отрезок. Для этого с каждым отрезком оси связывается параллелограмм с двумя вертикальными сторонами, длина которых равна расстоянию между осевыми линиями соседних строк. Этот параллелограмм преобразуется в прямоугольник вертикальным сдвигом столбцов пикселей растрового изображения, как показано на рисунке 6.

Преобразование вытягивает звенья ломаной в одну горизонтальную прямую, а вместе с этим превращаются в прямоугольники параллелограммы, для которых отрезки ломаной являются средними линиями.

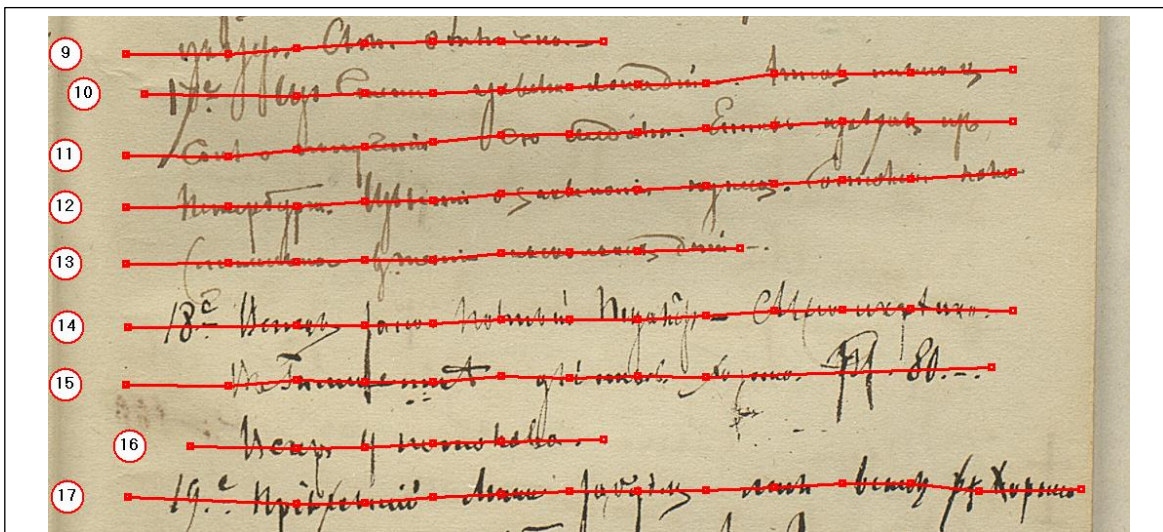


Рис. 5. Построение осевых линий строк в виде ломаных линий

Fig. 5. Construction of text line axes using polylines

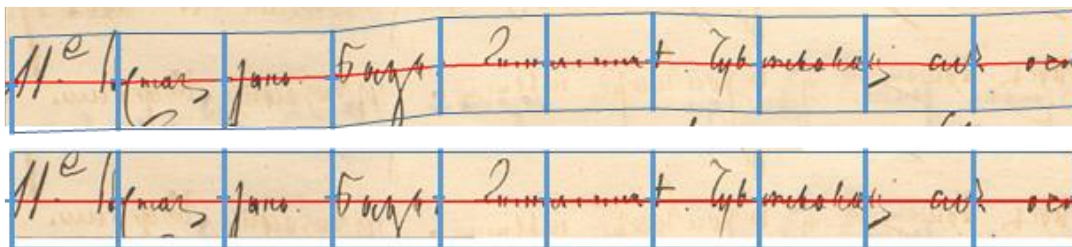


Рис. 6. Нормализация строки: ось вытягивается в горизонтальную прямую

Fig. 6. Normalizing a text line: the result shows the axis transformed into a horizontal line

### Расшифровка строк

Алгоритм распознавания рукописного текста представляет собой отдельный блок, который является сменным относительно программы. Это сделано для того, чтобы обеспечить в процессе исследований возможность экспериментов с разными методами и алгоритмами расшифровки. Взаимодействие с этим блоком осуществляется через стандартный интерфейс: на его вход подается отдельная строчка рукописного текста, а на выходе получается перевод этой строчки в виде текстовой строки в обычном алфавите, расширенном несколькими буквами старорусской орфографии, принятой в XIX веке.

Обучение используемого алгоритма распознавания осуществляется внешней программой. При этом принят стандартный формат представления обучающих данных, который не зависит от конкретного алгоритма распознавания и механизма обучения, принятого для этой программы. Формат устроен следующим образом. Элементарной единицей обучающей информации является отдельная размеченная строка текста, представленная в виде двух элементов: растрового изображения и текстовой строки разметки. В рамках инкрементного подхода очередной цикл обучения алгоритма выполняется на основе порции размеченных строк.

Такой интерфейс между программой и блоком распознавания определяет состав функций программы. Во-первых, это подготовка размеченных строк очередной порции рукописи для получения слабой разметки. А во-вторых, редактирование слабой разметки с целью получения точной.

В настоящей работе используется строчная архитектура VAN [6], схематично показанная на рисунке 7. Она представляет собой сверточную нейронную сеть. На рисунке 8 представлена структура входящего в нее кодировщика, который включает десять блоков (по три свертки в каждом) и одномерный сверточный декодировщик, преобразующий внутреннее представление модели в набор вероятностей.

Модель VAN выдает вероятности символов в строке, к ним применяется жадное декодирование CTC (*Connectionist Temporal Classification*, коннекционистская временная классификация), при обучении используется функция потерь CTC-loss [10]. Отметим, что CTC-loss – это инструмент для обучения нейронных сетей при решении задач, требующих сопоставления последовательностей разной длины. В данном случае входные данные (изображение рукописной строки) и выходные (текстовая строка) не выровнены и имеют разную длину.

Таким образом, при обучении решается задача оптимизации

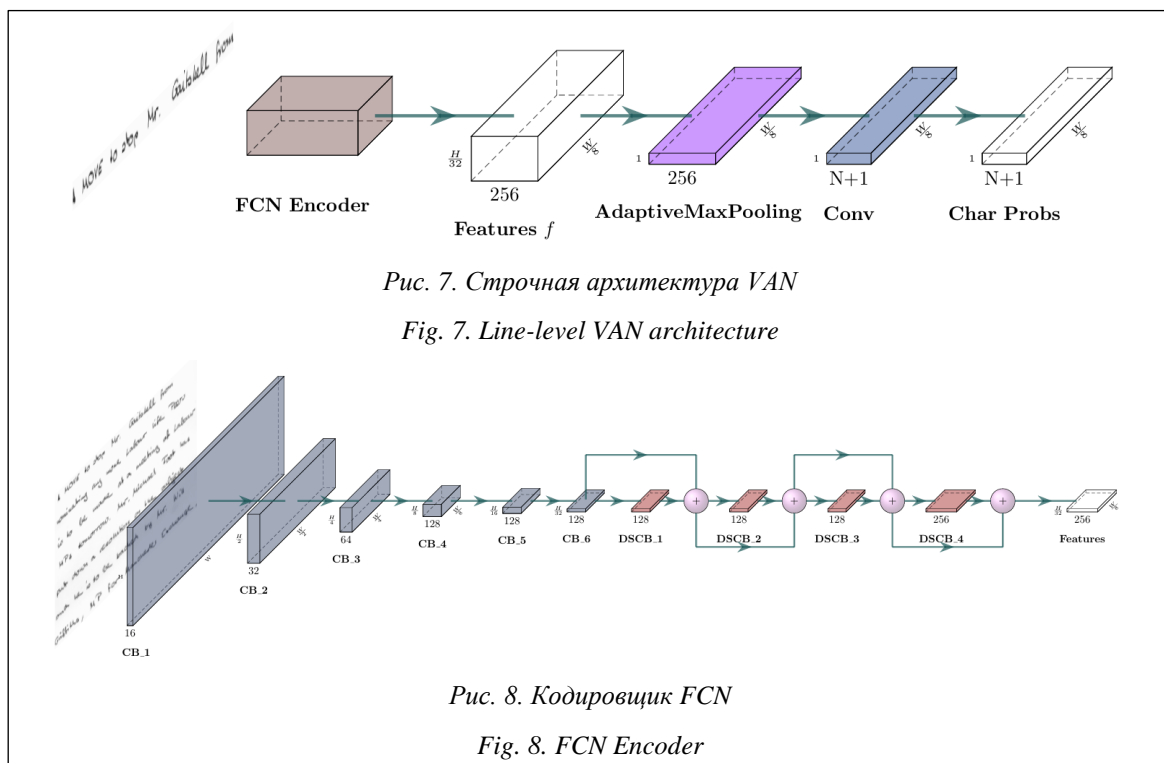


Рис. 7. Строчная архитектура VAN

Fig. 7. Line-level VAN architecture

Рис. 8. Кодировщик FCN

Fig. 8. FCN Encoder

$$\hat{\theta} = \arg \min_{\theta} L_{CTC}(y, \hat{y}(\theta)),$$

где  $y$  – правильная последовательность символов;  $\hat{y}(\theta)$  – предсказанная последовательность модели с параметрами  $\theta$ ;  $L_{CTC}(y, \hat{y}(\theta))$  – функция потерь.

### Вычислительный эксперимент

Целью эксперимента является проверка работоспособности и эффективности итерационного процесса, описанного выше. Были использованы реальные архивные данные рукописного дневника писателя-драматурга XIX века А.В. Сухова-Кобылина и их расшифровки, сделанные ранее при издании книг, посвященных творчеству писателя. Эти расшифровки не могут рассматриваться в качестве точной разметки поскольку представляют собой перевод рукописного текста XIX века на современный русский язык. Однако наличие такой расшифровки существенно облегчило точный подстрочный перевод рукописи квалифицированным экспертом-ассессором. В результате был получен качественный подстрочный перевод 114 рукописных страниц, содержащих 2 876 размеченных строк дневника. Перед обучением строчной модели VAN-данные были разбиты на три группы: 2 667 строк вошли в обучающую выборку, 123 строки – в валидационную, и 86 строк – в тестовую.

В качестве основного критерия качества расшифровки использовался показатель CER (*Character Error Rate*), процент ошибок в буквах. В качестве вспомогательного критерия – WER (*Word Error Rate*), процент ошибок в словах.

Все 2 667 строк обучающей выборки были разбиты на пять порций: четыре порции по 500 строк и пятая 667 строк. Размер пятой порции увеличен, чтобы использовать все имеющиеся данные. Валидационная и тестовая выборки оставались неизменными – 123 и 86 строк соответственно. На первом шаге итерационного процесса обучающая выборка состоит из 500 строк с точной разметкой. На этой ограниченной выборке производится обучение строчной модели VAN, веса кодировщика которой инициализировались значениями, полученными в процессе предобучения на открытом англоязычном датасете IAM [11]. При этом в эксперименте не используются предобученные веса

декодировщика, так как на разных датасетах размерности этих весов не совпадают ввиду различных алфавитов. Обучение останавливается после того, как достаточно большое количество эпох не приводит к снижению CER на валидации. Лучшей эпохой считается та, на которой наблюдается самая низкая ошибка CER на валидации. Веса, полученные на этой эпохе, используются далее на очередной итерации процесса разметки.

На каждой следующей итерации процесса в обучающую выборку добавляется очередная порция строк (по 500 строк в порциях со второй по четвертую и 667 в пятой порции). Веса кодировщика инициализируются числами, полученными на предыдущей итерации обучения. Декодировщик обучается с нуля, поскольку алфавит при увеличении обучающей выборки расширяется. Обучение останавливается после того, как достаточно большое количество эпох не приводит к снижению CER на валидации. После этого в обучающую выборку добавляется очередная порция строк, и запускается очередная итерация обучения. Таким образом, в эксперименте промоделирован итерационный процесс наращивания размера обучающей выборки и дообучения модели. Значения показателей качества расшифровки, полученные в ходе эксперимента, приведены в таблице 1. Динамика значений показателей CER и WER на тестовой выборке отображена на рисунке 9.

Как видно из полученных результатов, ошибки CER и WER на тестовых данных значительно снижаются на каждой итерации процесса. При этом на пятой итерации снижение CER произошло лишь на 1.14 %, это говорит о том, что дальнейшее увеличение обучающей выборки имеет мало практического смысла. Следовательно, 2 667 размеченных строк в обучающей выборке достаточно, и процесс разметки можно завершить.

Данные эксперимента дают возможность приближенно оценить, насколько использование инкрементного подхода снизило трудозатраты ассессора на разметку массива 2 667 строк. В таблице 2 объемы текста оцениваются количеством строк: во втором столбце представлены объемы порций, в третьем – ошибки слабой разметки в процентах от объема порций, взятые из таблицы 1. Исправление этих ошибок осуществляется ассессором вручную обычной коррекцией текста. Для простоты расчетов предположим, что трудозатраты ассессора на перевод строк и на исправление ошибок слабой разметки одинаковы. Это значит, что если

Таблица 1  
**Метрики CER и WER (в %) при разных размерах обучающей выборки, число эпох обучения, номер лучшей эпохи с точки зрения CER на валидации**

Table 1

**CER and WER (%) for different training set sizes, number of training epochs, and epoch with the best CER on the validation set**

Число строк на обучении	Обучение		Валидация		Тестирование		Всего эпох	Лучшая эпоха
	CER	WER	CER	WER	CER	WER		
500	0.15	0.23	22.70	58.71	19.62	48.30	18 881	14 144
1 000	0.10	0.23	17.92	50.70	15.15	39.60	2 851	2 222
1 500	0.11	0.35	15.39	45.39	13.51	38.07	2 678	2 040
2 000	0.37	0.84	13.46	42.12	12.07	32.90	3 402	2 066
2 667	0.27	0.87	12.32	37.84	10.93	30.55	5 108	4 562

слабая разметка 100 строк исходной рукописи содержит 20 % ошибок CER, то время ассессора на исправление этих ошибок составит 20 % от времени, которое он должен был бы затратить на ручной перевод этих 100 строк. Это предположение позволяет оценить объем ручной работы по переводу первой порции и по коррек-

тировке ошибок в остальных порциях. Первая порция требует ручного перевода 500 строк. Автоматическая расшифровка второй порции, тоже включающей 500 строк, содержит 19.62 % ошибок CER, что эквивалентно 98 строкам. Эти ошибки исправляются экспертом вручную. Поэтому работа над ошибками может быть оценена как эквивалентная ручному переводу 98 строк. Таким образом, трудозатраты на коррекцию ошибок можно оценить в количестве строк ручного перевода. Полученные данные представлены в четвертом столбце таблицы 2 в виде числа строк, которые переводятся экспертом вручную.

Таблица 2

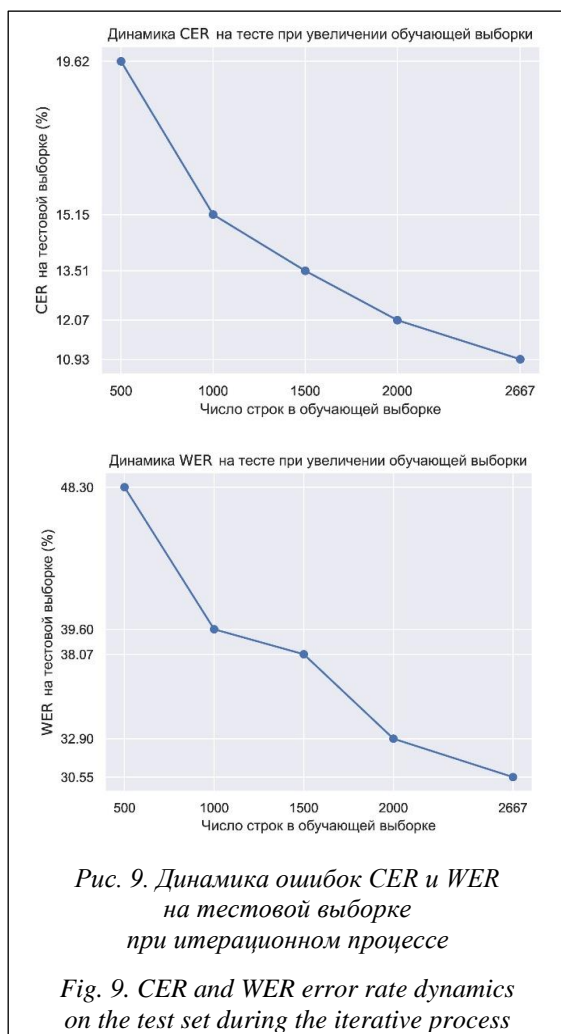
**Трудозатраты ассессора при построении инкрементной разметки рукописного текста**

Table 2

**Assessor labor costs for incremental markup of handwritten text**

Номер порции	Объем порции (строки)	Ошибки слабой разметки (%)	Объем работы над ошибками (строки)
1	500	–	500
2	500	19.62	98
3	500	15.15	76
4	500	13.51	68
5	667	12.07	81
Всего	2 667		823

Очевидно, что трудозатраты ассессора на ручную разметку текста объемом 2 667 строк соотносятся с затратами на инкрементную разметку того же текста как  $2\ 667 : 823 = 3.24$ . Это значит, что в рассмотренном примере время работы эксперта-ассессора сократилось больше чем в три раза. Следует отметить, что эта цифра



представляет собой нижнюю оценку сокращения трудозатрат. Она не учитывает эффекта подсказок, которые получает эксперт от слабой расшифровки (несмотря на ошибки значительная часть текста переводится правильно). Поэтому, как показывает опыт, часто возникают ситуации, когда человеческий взгляд эксперта не позволяет разобрать слова и строчки рукописи, а алгоритм расшифровки справляется с переводом. В этом случае даже одно–два правильно расшифрованных алгоритмом слова открывают смысл текста и являются подсказкой для понимания и точного перевода.

Поскольку на каждой итерации качество распознавания увеличивается, эксперту нужно исправлять все меньше и меньше ошибок слабой расшифровки на новых порциях данных, то есть такой процесс существенно упрощает работу эксперта-ассессора, снижает трудозатраты и существенно сокращает время разметки. Эти результаты экспериментально подтверждают корректность, эффективность и практическую применимость инкрементного подхода к разметке рукописных дневников.

### Заключение

Предлагаемый инкрементный подход к разметке рукописных архивных дневников позволяет существенно сократить трудозатраты эксперта-ассессора на обучение алгоритмов расшифровки и обеспечить тем самым возможности для автоматизации процессов распознавания, поиска, навигации и индексирования при работе с большими рукописными массивами документов.

Разметка рукописных страниц дневника осуществляется на основе подстрочного перевода большого объема. При этом современные

текстовые редакторы не предназначены для такой работы, они не рассчитаны на выполнение буквального, дословного перевода. Поэтому разработка специализированного редактора является актуальной задачей. Редактор для подстрочного перевода рукописного текста должен обеспечить возможности структурного анализа, сегментации и нормализации изображений страниц рукописи, строгую синхронизацию рукописи и подстрочного перевода, локальное редактирование подстрочного перевода. Разработанная в ходе исследования программа «Подстрочник» обеспечивает эти возможности. Программа может рассматриваться в качестве прототипа универсального редактора для разметки архивных рукописных текстов. Программно-редактор «Подстрочник» ориентирована на потребности небольших исследовательских коллективов при работе с историческим российским культурным наследием.

Программа прошла практическую проверку при подготовке рукописных дневников Ф.П. Литке и А.В. Сухова-Кобылина. Разметка состояла в подготовке подстрочного перевода около 100 рукописных страниц для каждого дневника, что позволило создать обучающие наборы данных с переводом более 2 000 строк каждый. Опыт использования этих данных для обучения алгоритмов расшифровки показал хорошее качество полученной разметки.

Дальнейшее развитие программы «Подстрочник» связано с усовершенствованием алгоритмов сегментации страниц рукописей. В частности, представляет интерес локализация и сегментация маргиналий, фрагментов на иностранных языках, форматных особенностей текста (даты, заголовки, шрифты), цвета чернил, типа пера или карандаша и т.п. Кроме того, предметом дальнейших исследований может стать совершенствование алгоритма расшифровки и механизма его обучения.

### Список литературы

1. Ефремова Т.Ф. Современный толковый словарь русского языка. М: Астрель, 2006.
2. Lomov N., Kropotov D., Stepochkin D., Laptev A. Handwritten text recognition and browsing in archive of prisoners letters from Smolensk Convict Prison. In: LNCS. Proc. AIST, 2024, vol. 14486, pp. 227–240. doi: 10.1007/978-3-031-54534-4\_16.
3. Базарова Т.А., Проскурякова М.Е. Автографы Петра I: чтение технологиями искусственного интеллекта и создание электронного архива // Историческая информатика. 2022. № 4. С. 179–190. doi: 10.7256/2585-7797.2022.4.39224.
4. Болтунова Е.М., Лаптев А.К. Распознавание рукописного текста и интеллектуальный анализ: возможности нейронных технологий (на примере работы с «Дневником» Ф.П. Литке) // Имагология и компаративистика. 2025. № 23. С. 358–379. doi: 10.17223/24099554/23/17.
5. Пенская Е.Н., Купцова О.Н. Невидимая величина. А.В. Сухова-Кобылин: театр, литература, жизнь. М.: Изд. дом ВШЭ, 2024. 472 с.
6. Coquenot D., Chatelain C., Paquet T. End-to-end handwritten paragraph text recognition using a vertical attention network. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, vol. 45, no. 1, pp. 508–524. doi: 10.1109/TPAMI.2022.3144899.

7. Язык программирования Паскаль живет всех живых. URL: <https://overclockers.ru/blog/Beltar/print/232144/Paskal-zhivye-vseh-zhivyh> (дата обращения: 01.07.2025).
8. Полухин Д.А., Сальников И.И. Методы и этапы распознавания рукописного текста // Научное обозрение. Педагогические науки. 2019. № 3–2. С. 71–74.
9. Barakat B.K., Cohen R., Droby A. et al. Learning-free text line segmentation for historical handwritten documents. *Appl. Sci.*, 2020, vol. 10, no. 22, art. 8276. doi: 10.3390/app10228276.
10. Graves A., Fernandez S., Gomez F., Schmidhuber J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. *Proc. ICML*, 2006, pp. 369–376. doi: 10.1145/1143844.1143891.
11. Liwicki M., Bunke H. IAM-OnDB - an on-line English sentence database acquired from handwritten text on a whiteboard. *Proc. ICDAR*, 2005, vol. 2, pp. 956–961. doi: 10.1109/ICDAR.2005.132.

Software &amp; Systems

doi: 10.15827/0236-235X.152.668-681

2025, 38(4), pp. 668–681

### Incremental markup of 19<sup>th</sup>-century handwritten archival diaries

Leonid M. Mestetskiy <sup>1, 2✉</sup>, Valeriy P. Zykov <sup>1</sup><sup>1</sup>Lomonosov Moscow State University, Moscow, 119991, Russian Federation<sup>2</sup>Higher School of Economics,  
Moscow, 101000, Russian Federation

#### For citation

Mestetskiy, L.M., Zykov, V.P. (2025) 'Incremental markup of 19<sup>th</sup>-century handwritten archival diaries', *Software & Systems*, 38(4), pp. 668–681 (in Russ.). doi: 10.15827/0236-235X.152.668-681

#### Article info

Received: 07.07.2025

After revision: 28.07.2025

Accepted: 30.07.2025

**Abstract.** The paper considers the problem of preparing data for machine learning based training of recognition algorithms for old handwritten texts. The research analyzes archival documents – 19th-century diaries that comprise the national cultural heritage. Modern archives preserve diaries that comprise thousands of digital page images. These images result from scanning physical handwritten documents. The special properties of these documents are the high cultural level of the authors, large volumes of manuscripts, uniform handwriting, and uniform text formatting style. Training recognition algorithms requires annotated manuscript data. This markup involves creating a precise, verbatim transcription of a text portion. A highly qualified expert assessor, a specialist in the creative heritage of the diary author, performs such verbatim transcription. However, even for a specialist, markup requires significant effort. This process constitutes the most time-consuming element in automating archival handwritten text workflows. The objective of this research is to develop a novel methodology for expert assessors. This methodology substantially decreases the labor required for handwritten diary markup. The authors propose an approach based on iterative expansion of the marked dataset with small sequential increments. Accurate manuscript markup proceeds through two sequential phases. An existing trained algorithm first produces an automatic transcription. Subsequently, a specialist edits the algorithmic output to obtain an accurate verbatim transcript. The expert's precise markup subsequently serves for the next algorithm training iteration. The proposed approach is implemented as the Podstrochnik (Interlinear) software package. The software supports the complete data markup cycle for training handwriting recognition algorithms. This cycle includes automatic transcription, subsequent editing to obtain precise markup, and model retraining. The software underwent practical testing using diaries from F.P. Litke and A.V. Sukhovo-Kobylin.

**Keywords:** machine learning, incremental markup, handwritten text, manuscript transcription, interlinear transcription

**Acknowledgements.** The work is supported by Russian Science Foundation grant no. 22-68-00066 project "Cultural Heritage of Russia: Intellectual Analysis and Thematic Modeling of the Corpus of Handwritten Texts"

#### References

1. Efremova, T.F. (2006) *Modern Explanatory Dictionary of the Russian Language*. Moscow (in Russ.).
2. Lomov, N., Kropotov, D., Stepanchkin, D., Laptev, A. (2024) 'Handwritten text recognition and browsing in archive of prisoners letters from Smolensk Convict Prison', in *LNCS. Proc. AIST*, 14486, pp. 227–240. doi: 10.1007/978-3-031-54534-4\_16.
3. Bazarova, T.A., Proskuryakova, M.E. (2022) 'Autographs of Peter the Great: Reading with artificial intelligence technologies and creating an electronic archive', *Historical Informatics*, (4), pp. 179–190 (in Russ.). doi: 10.7256/2585-7797.2022.4.39224.

4. Boltunova, E.M., Laptev, A.K. (2025) 'Handwriting recognition and data mining: Possibilities of neural network technologies (based on admiral Fyodor Lutke's diary)', *Imagology and Comparative Studies*, (23), pp. 358–379 (in Russ.). doi: 10.17223/24099554/23/17.
5. Penskaya, E.N., Kuptsova, O.N. (2024) *The Invisible Quantity. A.V. Sukhovo-Kobylin: Theater, Literature, Life*. Moscow, 472 p. (in Russ.).
6. Coquenet, D., Chatelain, C., Paquet, T. (2023) 'End-to-end handwritten paragraph text recognition using a vertical attention network', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1), pp. 508–524. doi: 10.1109/TPAMI.2022.3144899.
7. *The Pascal Programming Language is Alive and Well*, available at: <https://overclockers.ru/blog/Beltar/print/232144/Paskal-zhivee-vseh-zhivyyh> (accessed July 01, 2025) (in Russ.).
8. Polyukhin, D.A., Salnikov, I.I. (2019) 'Methods and stages of recognition of the hand-written text', *Scientific Review. Pedagogical Sci.*, (3–2), pp. 71–74 (in Russ.).
9. Barakat, B.K., Cohen, R., Droby, A. et al. (2020) 'Learning-free text line segmentation for historical handwritten documents', *Appl. Sci.*, 10(22), art. 8276. doi: 10.3390/app10228276.
10. Graves, A., Fernandez, S., Gomez, F., Schmidhuber, J. (2006) 'Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks', *Proc. ICML*, pp. 369–376. doi: 10.1145/1143844.1143891.
11. Liwicki, M., Bunke, H. (2005) 'IAM-OnDB - an on-line English sentence database acquired from handwritten text on a whiteboard', *Proc. ICDAR*, pp. 956–961. doi: 10.1109/ICDAR.2005.132.

**Авторы**

**Местецкий Леонид Моисеевич**<sup>1,2</sup>, д.т.н.,  
профессор, ведущий научный сотрудник,  
mestlm@mail.ru

**Зыков Валерий Павлович**<sup>1</sup>, магистрант,  
valera\_zykov\_2003@mail.ru

<sup>1</sup> МГУ имени М.В. Ломоносова, г. Москва,  
119991, Россия

<sup>2</sup> Высшая школа экономики,  
г. Москва, 101000, Россия

**Authors**

**Leonid M. Mestetskiy**<sup>1,2</sup>, Dr.Sci. (Engineering),  
Professor, Leading Researcher,  
mestlm@mail.ru

**Valeriy P. Zykov**<sup>1</sup>, Graduate Student,  
valera\_zykov\_2003@mail.ru

<sup>1</sup> Lomonosov Moscow State University,  
Moscow, 119991, Russian Federation

<sup>2</sup> Higher School of Economics,  
Moscow, 101000, Russian Federation