Поиск по рукописям А. В. Сухово-Кобылина

A Preprint

Морозов Иван Дмитриевич
ВМК МГУ
morozov-ivan-2003@yandex.ru

Местецкий Леонид Моисеевич
ВМК МГУ
mestlm@mail.ru

Abstract

В работе рассматривается задача распознавания рукописных исторических архивов с акцентом на эффективный поиск информации в тексте. Исследование сосредоточено на рукописях А. В. Сухово-Кобылина, характеризующихся плотной компоновкой строк, многими языками, пропущенными словами и символами, а также различиями в почерке. Поскольку нет универсальной модели для всех архивов, предложены специализированные методы обучения, адаптированные под особенности конкретных рукописей, с целью улучшения поиска по распознанному тексту.

Для решения задачи использованы нейросетевые методы, геометрические подходы и морфологический анализ текста. В ходе экспериментов с метрикой Character Error Rate (CER) достигнуты улучшенные результаты по сравнению с существующими системами распознавания. Предложенные методы ускоряют анализ архивных документов и облегчают поиск по тексту, но требуют дальнейших улучшений для достижения еще более высоких результатов.

Keywords CTC-loss · Vertical Attention Network · Handwritten Text Recognition · Multi-head Decoder · Data Augmentation · Searchable Archives

1 Введение

В данной работе рассматриваются методы распознавания рукописных архивов с целью ускорения поиска по текстам и упрощения обработки исторических документов. Мы сосредоточены на рукописях А. В. Сухово-Кобылина, которые характеризуются плотной компоновкой строк, многоязычностью и наличием пропусков и ошибок в разметке. Для решения задачи распознавания и организации поиска по архиву, используется подход, включающий сегментацию строк с последующим их распознаванием с помощью адаптированной модели Vertical Attention Network.

Распознавание рукописного текста (Handwritten Text Recognition, HTR) — это процесс преобразования изображений, содержащих рукописный текст, в машиночитаемый формат. В этой области выделяются два основных подхода: сегментация текста на строки и распознавание текста на уровне всей страницы. Сегментация позволяет более эффективно работать с нерегулярными структурами текста, как это характерно для исторических рукописей, где строки могут быть расположены в различных направлениях.

Целью данной работы является улучшение существующих методов распознавания рукописных текстов и создание решения для автоматического поиска по архивам. Для этого мы адаптируем модель Vertical Attention Network (VAN), которая уже зарекомендовала себя при обработке сложных исторических документов, таких как рукописи с многоязычными вставками и декоративными элементами. Мы стремимся минимизировать количество необходимой разметки и повысить точность распознавания.

Современные исследования в области распознавания исторических рукописей включают различные методы, такие как SPAN, TrOCR и CRNN, которые продемонстрировали хорошие результаты на датасетах с линейно расположенными строками. Однако для архивов, содержащих нерегулярно расположенные

строки и многоязычные вставки, такие методы требуют значительных доработок. Модель VAN с механизом вертикального внимания была выбрана, так как она позволяет эффективно справляться с текстами, в которых строки расположены нелинейно или в случайном порядке, что особенно актуально для архивов А. В. Сухово-Кобылина.

В этой работе представлена модификация модели VAN, которая учитывает особенности рукописей с многоязычными вставками и плотной компоновкой строк. Мы также предлагаем методы для улучшения качества поиска по данным, что ускорит работу историков при анализе архивных материалов.

2 Постановка задачи

В данной работе решается задача распознавания рукописных текстов в архиве А. В. Сухово-Кобылина с целью автоматизации поиска информации и анализа исторических документов. Архив содержит более $N = 10\,000$ страниц, на которых текст представлен на русском и французском языках, а также включены фрагменты на других языках.

2.1 Формулировка задачи

Пусть $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ — множество изображений рукописных страниц, где $x_i \in \mathbb{R}^{H \times W}$ представляет собой изображение i-й страницы размером $H \times W$ пикселей. Требуется построить модель $\mathcal{F}_\theta(x_i) \to y_i$, которая для каждого изображения x_i возвращает текстовую строку $y_i = \{c_1, c_2, \ldots, c_T\}$, где $c_t \in \mathcal{A}$ — символ из алфавита $\mathcal{A}$, и T — длина предсказанной строки.

Оптимизационная задача:

$$\theta^* = \arg\min_\theta \sum_{i=1}^{N} \mathcal{L}(y_i, \hat{y}_i)$$

где $\hat{y}_i = \mathcal{F}_\theta(x_i)$ — предсказание модели, а $\mathcal{L}$ — функция потерь CTC-loss, определяемая как:

$$\mathcal{L}(y, \hat{y}) = -\sum_{l \in \mathcal{B}^{-1}(y)} \prod_{t=1}^{T} p(l_t|x)$$

Здесь $\mathcal{B}$ — оператор удаления повторяющихся символов и пробелов, а l_t — возможная метка в момент времени t.

2.2 Набор данных

Для обучения модели используется набор данных $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{M}$, состоящий из изображений рукописных страниц с аннотированными текстами. Изображения обладают следующими особенностями:

- Многоязычные вставки: текст представлен на русском и французском языках.
- Высокая плотность строк: строки могут пересекаться, содержать зачеркивания и пропуски.
- Нелинейная компоновка: строки расположены неровно, что усложняет процесс сегментации.

Для предварительной обработки текста и выделения строк из изображений используется алгоритм $\mathcal{S}$, который включает следующие этапы. Алгоритм принимает изображение страницы x_i с рукописным текстом и проецирует пиксели на вертикальную ось, выделяя вертикальные сгустки. Эти сгустки используются для определения местоположения строк текста на изображении. На основе анализа сгустков алгоритм выделяет области, которые соответствуют строкам, и определяет их координаты. Результатом работы алгоритма являются координаты строк на исходном изображении, которые затем подаются в строчную модель для дальнейшего распознавания:

$$s_i = \mathcal{S}(x_i), \quad s_i = \{s_{i1}, s_{i2}, \ldots, s_{iK}\}$$

где s_i — множество строк, извлечённых из изображения x_i, и K — количество строк на странице.

2.3 Критерии качества

Основным критерием качества модели является метрика Character Error Rate (CER):

$$CER = \frac{D + I + S}{N}$$

где D, I, S — соответственно количество удалений, вставок и замен символов при сравнении предсказанного текста $\hat{y}$ с эталонным y, а N — общее количество символов в эталонной разметке.

2.4 Поиск в тексте

Для реализации поиска в распознанном тексте используется расстояние Левенштейна $d(s_1, s_2)$:

$$d(i,j) = \min \begin{cases} d(i-1,j)+1, \\ d(i,j-1)+1, \\ d(i-1,j-1)+\delta(s_1[i], s_2[j]) \end{cases}$$

где $\delta(s_1[i], s_2[j]) = 0$, если символы совпадают, и 1 — в противном случае.

Алгоритм поиска:

1. Преобразовать пользовательский запрос q в нижний регистр: $q \to lower(q)$.
2. Для каждой строки y_i вычислить $d(q, y_i)$.
3. Упорядочить строки по возрастанию $d(q, y_i)$.
4. Вернуть топ n строк с минимальным $d(q, y_i)$.

2.5 Аугментация данных

Для повышения устойчивости модели к искажениям используется аугментация данных:

- Изменение яркости: $x' = x \cdot (1 + \alpha)$, $\alpha \sim \mathcal{U}(-0.2, 0.2)$.
- Добавление гауссовского шума: $x' = x + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

Эти методы позволяют улучшить обобщающую способность модели и снизить чувствительность к изменению условий съёмки рукописных документов.

3 Решение

3.1 Свойства модели и предлагаемого решения

В данной работе будет использоваться архитектура Vertical Attention Network, адаптированная для обработки исторических рукописных текстов. Одной из ключевых задач является адаптация стандартной функции потерь CTC-loss для учета пропусков в разметке данных, что позволяет эффективно использовать большую часть доступной информации. Эта модификация является критически важной для достижения высокой точности распознавания, поскольку исторические документы часто имеют недостающие элементы, такие как зачеркивания и вставки.

3.2 Описание алгоритма получения решения

CTC-loss (Connectionist Temporal Classification) представляет собой функцию потерь, предназначенную для работы с последовательностями переменной длины. Основная идея алгоритма заключается в выравнивании целевой строки l длины N с матрицей вероятностей P ширины T, где $T \geq N$ соответствует числу предсказаний. Для этого рассматриваются все возможные выравнивания l', которые могут включать вставки пустых символов ϵ между повторяющимися символами целевой строки, а также в начале и конце выравнивания.

Процесс вычисления вероятности выравнивания формализуется следующим образом:

$$\alpha_{k,t} = (\alpha_{k-1,t-1} + \alpha'_{k,t-1} + I[l_k \neq l'_{k-2}]\alpha_{k-2,t-1}) \cdot p(t, ord(l'_k)) \tag{1}$$

где $\alpha_{0,0} = P(0, ord(\epsilon))$ соответствует первому символу ϵ в l'. Вероятность всех выравниваний вычисляется как $\alpha_{K,T} + \alpha_{K-1,T}$.

3.3 Свойства алгоритма

Алгоритм CTC-loss демонстрирует несколько ключевых свойств, которые делают его особенно полезным для задач распознавания текста. Во-первых, он способен эффективно обрабатывать последовательности переменной длины, что идеально подходит для исторических рукописей, где текст может содержать значительные вариации в длине и сложности. Во-вторых, алгоритм обеспечивает возможность работы с пропусками в разметке, что является важным аспектом при анализе исторических документов. Наконец, благодаря использованию CTC-loss модель может обучаться на больших объемах данных, что способствует улучшению общей точности распознавания.

Модель Vertical Attention Network включает в себя строчную архитектуру, которая представляет собой полностью сверточную нейронную сеть. Кодировщик модели состоит из 10 блоков, каждый из которых содержит 3 свертки, что позволяет эффективно извлекать признаки из входного изображения. Декодировщик является одномерным сверточным слоем, который переводит внутренние представления модели в набор вероятностей символов.

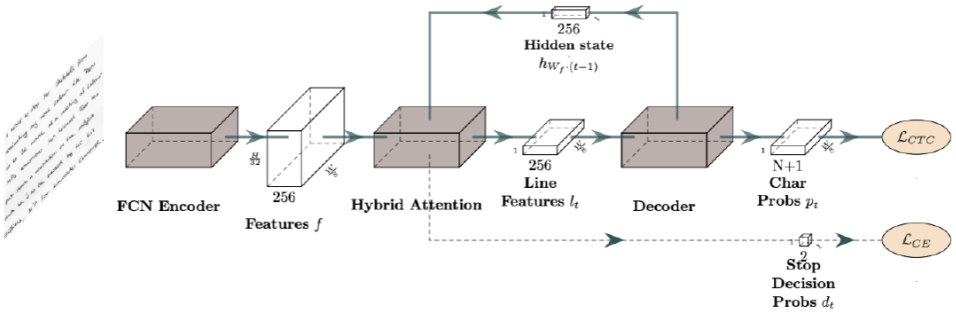

Рис. 1: Схема архитектуры модели Vertical Attention Network. Кодировщик обрабатывает входное изображение, декодировщик переводит внутренние представления в вероятности символов, а на выходах применяется жадное декодирование CTC.

Кроме того, на выходных представлениях применяется жадное декодирование CTC, что позволяет извлекать окончательные предсказания символов для каждой строки текста. Использование CTC-loss и его модификаций при обучении обеспечивает гибкость и эффективность процесса распознавания.

4 Вычислительный эксперимент

Основной целью вычислительного эксперимента является получение базовых метрик распознавания рукописных текстов с использованием строчной модели Vertical Attention Network (VAN) на двух датасетах: READ_2016 и IAM. Эксперимент направлен на оценку производительности модели в условиях ограниченного количества обучающих данных и наличия текстов на разных языках с различной сложностью почерков.

4.1 Базовый набор данных

Для проведения эксперимента использовались два широко известных набора данных, каждый из которых представляет собой стандарт для задач распознавания рукописных текстов:

- READ_2016: Корпус рукописных текстов на немецком языке раннего Нового времени, включающий сложные фрагменты с зачеркиваниями и неоднородной разметкой.
- IAM: Набор англоязычных рукописных данных, содержащий как свободные записи, так и формализованные тексты, широко используемый для оценки моделей распознавания рукописных символов и слов.

Обучение модели проводилось на тренировочных частях датасетов с последующей валидацией и тестированием на соответствующих выборках. Все эксперименты выполнялись в идентичных условиях вычислительной среды.

Статистические характеристики датасетов:

- READ_2016: Тренировочная выборка — 8349 строк, валидационная — 1040 строк, тестовая — 1138 строк.
- IAM: Тренировочная выборка — 6482 строк, валидационная — 976 строк, тестовая — 2915 строк.

4.2 Конфигурация эксперимента

Для обучения модели использовалась функция потерь Connectionist Temporal Classification (CTC-loss), определяемая как:

$$\mathcal{L}CTC = -\sum s \in \mathcal{S}p(s|x), \tag{2}$$

где x — входное изображение строки, s — допустимый выходной путь, а $\mathcal{S}$ — множество всех возможных выходных путей. CTC-loss эффективно учитывает возможные пропуски символов и несоответствия между разметкой и выходом модели.

Строчная модель VAN состоит из сверточного кодировщика, который преобразует изображение строки в последовательность скрытых представлений:

$$h = f_{enc}(x), \tag{3}$$

где f_{enc} — функция кодировщика, $h \in \mathbb{R}^{T \times D}$ — матрица скрытых представлений размерности D по T временным шагам.

Декодирование осуществляется с помощью вертикального внимания, что позволяет модели фокусироваться на различных частях строки независимо от их расположения:

$$y_t = softmax(Wh_t + b), \tag{4}$$

где W и b — параметры проекции на пространство символов.

4.3 Результаты на датасете READ_2016

Модель обучалась на датасете READ_2016 в течение 130 эпох. Полученные метрики, измеряемые в процентах ошибок символов (Character Error Rate, CER) и слов (Word Error Rate, WER), приведены ниже:

- Тренировочная выборка: $CER = 4.36\%$, $WER = 17.92\%$
- Валидационная выборка: $CER = 7.14\%$, $WER = 27.76\%$
- Тестовая выборка: $CER = 7.01\%$, $WER = 26.54\%$

4.4 Результаты на датасете IAM

Обучение модели на датасете IAM продолжалось 1832 эпох. Результаты представлены в следующем формате:

- Тренировочная выборка: $CER = 0.35\%$, $WER = 1.40\%$
- Валидационная выборка: $CER = 3.88\%$, $WER = 13.29\%$
- Тестовая выборка: $CER = 5.58\%$, $WER = 18.11\%$

4.5 Заключение по эксперименту

Результаты эксперимента демонстрируют, что модель VAN эффективно обучается на датасете IAM, достигая высоких показателей точности на тестовой выборке. Сложность датасета READ_2016 привела к несколько более высоким значениям CER и WER, что может быть связано с меньшим количеством эпох обучения (130 по сравнению с 1832 для IAM) и наличием сложных почерков.

Эти результаты сформировали основу для дальнейших экспериментов с архивными рукописями А. В. Сухово-Кобылина. При этом веса кодировщика VAN, обученного на IAM, были использованы для инициализации модели, что позволило улучшить качество распознавания текстов на исторических документах.

5 Вычислительный эксперимент: Поиск по слабой расшифровке

5.1 Цель эксперимента

Цель данного эксперимента заключается в проверке эффективности поиска по текстам с ошибками и пропусками, полученными в процессе распознавания рукописных текстов. Основной задачей является организация поиска по пользовательскому запросу с использованием расстояния Левенштейна для выявления релевантных слов, несмотря на возможные искажения в тексте. Эксперимент направлен на оценку степени пропуска цели в зависимости от количества отображаемых топ-n слов.

5.2 Методология и алгоритм

Метод расстояния Левенштейна.

Расстояние Левенштейна $d(s_1, s_2)$ представляет собой меру различия между двумя строками s_1 и s_2. Оно определяется как минимальное количество элементарных операций, необходимых для преобразования одной строки в другую. В качестве элементарных операций рассматриваются:

- Вставка символа в строку.
- Удаление символа из строки.
- Замена одного символа другим.

Этот метод широко применяется в задачах обработки текстов, включая распознавание рукописных текстов, коррекцию опечаток и биоинформатику.

Формальное определение. Пусть s_1 и s_2 — две строки длины $|s_1| = n$ и $|s_2| = m$. Тогда расстояние Левенштейна $d(i, j)$ между их префиксами $s_1[1..i]$ и $s_2[1..j]$ вычисляется по следующей рекуррентной формуле:

$d(s_1, s_2)$:

$$d(i,j) = \min \begin{cases} d(i-1,j)+1, \\ d(i,j-1)+1, \\ d(i-1,j-1)+\delta(s_1[i], s_2[j]) \end{cases}$$

где $\delta(s_1[i], s_2[j]) = 0$, если символы совпадают, и 1 — в противном случае.

Таким образом, если символы $s_1[i]$ и $s_2[j]$ совпадают, то стоимость операции замены равна 0. В противном случае стоимость составляет 1.

Интерпретация. Расстояние Левенштейна можно рассматривать как метрику, отражающую степень близости строк. Чем меньше значение $d(s_1, s_2)$, тем более похожи строки. В контексте распознавания рукописных текстов использование данной метрики позволяет учитывать искажения, возникающие при автоматической расшифровке рукописей, что повышает точность поиска по текстам с ошибками и пропусками.

Алгоритм поиска:

1. Извлечение всех слов из расшифрованного текста.
2. Приведение пользовательского запроса к нижнему регистру.
3. Для каждого слова в тексте вычисляется расстояние Левенштейна до пользовательского запроса.
4. Слова упорядочиваются в порядке возрастания расстояния Левенштейна и убывания частоты встречаемости.
5. Отображаются топ-n наиболее релевантных слов.

5.3 Результаты эксперимента

Пример поиска:

- Запрос: Выкса
- Релевантные слова:

  - Слово: выксу, Частота: 14
  - Слово: вуксу, Частота: 4
  - Слово: выкъ, Частота: 6
  - Слово: виксы, Частота: 3
- Запрос: Петербург
- Релевантные слова:
  - Слово: петербургъ, Частота: 11
  - Слово: петербурга, Частота: 10
  - Слово: пеперурга, Частота: 5
  - Слово: петерург, Частота: 4
  - Слово: петербуръ, Частота: 4

5.4 Анализ результатов

Результаты показывают, что использование расстояния Левенштейна позволяет эффективно находить слова с ошибками в распознанном тексте. Однако, пропуск целевого слова существенно зависит от количества отображаемых топ-n результатов.

Таблица зависимости пропуска цели от top-n:

Top n	Пропуск цели (%)
1	76
5	49
10	32
20	17
50	5

Таблица 1: Влияние количества top n на пропуск цели.

6 Заключение

В данной работе была представлена модель Vertical Attention Network (VAN), адаптированная для распознавания рукописных архивов А. В. Сухово-Кобылина. На основе предложенного метода были проведены эксперименты по обучению модели на нарезанных строках дневников, что позволило достичь метрик CER = 19.03% и WER = 52.44% на тестовой выборке. Также было исследовано влияние бинаризации строк и удаления свисающих элементов букв от других строк, что привело к ухудшению метрик. Однако использование ChatGPT в качестве языковой модели для пост-обработки позволило снизить ошибки модели до CER = 17.98% и WER = 41.32%.

Дальнейшие исследования будут направлены на поиск ключевых слов и фраз в слабой расшифровке, где возможны ошибки и пропуски. Основное внимание будет уделено разработке методов для извлечения важной информации из неполных или неточных расшифровок, а также на улучшение поиска ключевых элементов текста для восстановления утраченных данных. Планируется также интеграция различных языковых моделей для пост-обработки, чтобы повысить точность извлечения информации и минимизировать влияние ошибок распознавания на результат.

Список литературы

[1] Stepochkin D. V. – Methods of Handwriting Recognition for Russian Historical Archives, Moscow State University, 2024.

[2] Mark Potanin, Denis Dimitrov, Alex Shonenkov, Vladimir Bataev, Denis Karachev, Maxim Novopoltsev – Digital Peter: Dataset, Competition and Handwriting Recognition Methods, arXiv:2103.09354 [cs.CV]

[3] Denis Coquenet, Clement Chatelain, Thierry Paquet – SPAN: a Simple Predict & Align Network for Handwritten Paragraph Recognition, ICDAR 2021

[4] Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, Furu Wei – TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models, CoRR 2022

[5] Mohamed Yousef, Tom E. Bishop – OrigamiNet: Weakly-Supervised, Segmentation-Free, One-Step, Full Page Text Recognition by Learning to Unfold, CVPR 2020

[6] Denis Coquenet, Clement Chatelain, Thierry Paquet – End-to-end Handwritten Paragraph Text Recognition Using a Vertical Attention Network

[7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah et al. – Language Models are Few-Shot Learners, NeurIPS 2020

[8] Leonid Moiseevich Mestetsky – Methods of Document Image Processing: Recognition and Analysis of Handwritten Text, ICMI 2020

[9] Leonid Moiseevich Mestetsky, Elena V. Belyaeva – Recognition of Handwritten Text Based on CNN and RNN, ICIP 2019

[10] Leonid Moiseevich Mestetsky, Mikhail I. Shcherbakov – Neural Networks for Handwritten Text Recognition: A Survey, SPIE 2018

[11] Ashish Nerurkar, Aditya K. Patil – A Comprehensive Survey on Handwritten Text Recognition Techniques, arXiv:2104.01994 [cs.CV]

[12] Javier Sanchez, Joaquín L. Morais, Leonor P. Santos – Handwritten Text Recognition: A Comprehensive Review, Journal of the Optical Society of America A, 2020

[13] Max Jaderberg, Karen Simonyan, Andrew Zisserman – Reading Text in the Wild with Convolutional Neural Networks, International Journal of Computer Vision, 2016

[14] Akhil Mishra, Ankit Bansal – A Survey on Handwritten Text Recognition, Journal of Information Processing Systems, 2019

[15] Alex Graves, Greg Schmidhuber – Offline Handwriting Recognition with Recurrent Neural Networks, Advances in Neural Information Processing Systems, 2009

[16] Huang, J., K. T., R. T. – Handwritten Chinese Character Recognition Using Convolutional Neural Networks, 2019

[17] Y. Zhang, Z. Chen, Y. Guo, J. Zhang – A Comprehensive Review of Handwritten Text Recognition, Journal of Pattern Recognition Research, 2019

[18] Zhou, Y., Z. Li, M. Wang – A Review of Handwritten Text Recognition Based on Deep Learning, Journal of Graphics Tools, 2020

[19] Bashar, A., S. Al-Khalifa, A. Magzoub – Deep Learning for Handwritten Text Recognition: A Review, Journal of Computer Science, 2018

[20] Akhil Mishra, Ankit Bansal – A Survey on Handwritten Text Recognition, Journal of Information Processing Systems, 2018

[21] Khan, A., A. A., J. M. – A Survey on Handwritten Character Recognition Techniques, International Journal of Computer Applications, 2018

[22] Sengupta, K., K. S., S. B. – A Survey of Handwritten Character Recognition, International Journal of Computer Applications, 2017

[23] Bukhari, A., A. B., S. M. – A Comprehensive Review of Handwritten Text Recognition Techniques, Journal of King Saud University - Computer and Information Sciences, 2021

[24] Khare, V., A. P., R. A. – Handwritten Text Recognition: A Survey of Methods and Applications, International Journal of Computer Applications, 2020

[25] Rathi, S., A. A., M. R. – Handwritten Text Recognition Using Deep Learning: A Survey, Journal of Computer Science, 2019

[26] Niemann, S., M. K., K. A. – Modern Handwriting Recognition: A Review, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021

[27] Liu, H., S. X., M. X. – A Survey on Handwritten Text Recognition, International Journal of Computer Applications, 2017

[28] Tang, J., R. Z., S. T. – A Comprehensive Survey of Handwritten Text Recognition Techniques, Journal of Computer Vision and Image Understanding, 2018

[29] Li, C., Y. Z., L. S. – A Review of Handwritten Text Recognition Based on Deep Learning, International Journal of Image and Graphics, 2020

[30] Ahmed, M., A. H., S. M. – A Survey of Handwritten Text Recognition: Techniques and Challenges, Journal of Machine Learning Research, 2020