

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
имени М. В. Ломоносова  
Факультет вычислительной математики и кибернетики

---

*Козлов В. Д.*

МЕТОДЫ  
ОФФЛАЙН-РАСПОЗНАВАНИЯ  
РУКОПИСНОГО ТЕКСТА  
(ОБЗОР)

Москва, 2014

# Введение

В настоящее время сканирование и сохранение в памяти компьютера текста с твердого носителя является решенной задачей. Это существенно облегчает задачу хранения рукописных и печатных текстов и предоставления доступа к ним различных пользователей. Однако полученный в результате сканирования текст хранится в памяти компьютера в виде изображения, чаще всего растрового, что делает работу с ним весьма сложной: затруднено ориентирование, практически невозможны редактирование, форматирование и поиск по тексту. Для решения этих задач необходимо провести процесс распознавания текста на изображении с созданием файла в том или ином текстовом формате. Перевод изображений рукописного, машинописного или печатного текста в текстовые данные называется распознаванием текста.

Задача распознавания машинописного текста носит название оптического распознавания символов (*optical character recognition, OCR*). В настоящее время существуют высокоточные системы для распознавания машинописных и рукопечатных текстов (например, FineReader фирмы АBBYY). Распознавание же рукописных текстов является гораздо более сложной и на данный момент не решённой задачей.

Задача распознавания рукописного текста носит название HWR (*handwriting recognition*). Существуют два различных класса задач HWR:

- онлайн-распознавание — распознавание текста ведётся параллельно с вводом текста;
- оффлайн-распознавание — распознавание текста ведётся на уже синтезированном изображении.

При онлайн-распознавании процесс формирования изображения текста и процесс его ввода в систему распознавания совмещены, что позволяет системе отслеживать процесс начертания символов. Это даёт возможность получать помимо графической информации ещё и информацию о структуре входных изображений, например, о направлении и скорости движения пера или о его нажиме при написании символа. На данный момент онлайн-системы распознавания широко используются на планшетных ПК.

В задаче оффлайн-распознавания, в отличие от предыдущей, системе доступна только графическая информация. Уже это делает её значительно труднее онлайн-вой. Неполный список проблем, типичных при распознавании рукописного текста оффлайн в общем случае, включает в себя:

- высокая вариативность начертания символов — по размеру, наклону, набору составных частей, связям между ними и др.;
- орфографические ошибки в тексте;
- специфические особенности начертания, не позволяющие уверенно разделять символы;
- пересечение элементов текста, наложение частей текста друг на друга;
- помарки, кляксы, исправления, дефекты носителя (бумаги), а также артефакты, возникающие при сканировании;

- непараллельность строк текста;  
и другие.

На данный момент коммерческие пакеты HWR могут уверенно распознавать только машиночитаемые формы (анкеты, заполненные бланки и тому подобное), поскольку при использовании таких структурированных документов и уменьшении диапазона возможных вводимых символов качество распознавания резко повышается. К этому случаю относятся распознавание почтовых адресов при автоматической сортировке почты, подписей на чеках, цифр и др.

Стандартное оффлайн-распознавание рукописного текста проводится по следующей схеме:

- 1) предобработка изображения, выделение области интереса;
- 2) сегментация и нормализация текста из области интереса;
- 3) распознавание сегментированного текста тем или иным методом.

## 1 Предобработка изображения

На этапе предобработки изображения выполняются следующие задачи:

1. Повышение качества изображения (*preprocessing*), выполняющееся методами обработки изображений (фильтрация, шумоподавление и другие) и имеющая своей целью повысить качество изображения;
2. Выделение области интереса на изображении, использующее методы анализа изображений и имеющее своей целью избавиться от нетекстовой информации (например, пятна, помарки, изображения в тексте и другие)

### 1.1 Препроцессинг

На этом этапе происходит очистка изображения от дефектов сканирования. Осуществляется это стандартными методами обработки изображений (например, применением различных фильтров). В частности, в самом начале работы к изображению в целях шумоподавления часто применяется фильтр Гаусса [24].

Важную роль в препроцессинге занимает т.н. пороговая бинаризация (*threshold binarization*) — перевод изображения в чёрно-белый формат из цветного или оттенков серого. Это позволяет резко разделить текст и фон, упрощает в дальнейшем применение многих алгоритмов, а также избавляет от некоторых шумов на изображении.

Обычно на гистограмме яркости изображения текста наблюдается два пика: высокий пик в области светлых пикселей (соответствует фону, то есть бумаге) и пик пониже в области тёмных (соответствует тексту). Поэтому задача поиска порогового значения яркости, т. е. такого, что пиксели с яркостью выше этого значения (фон) будут считаться чёрными, а ниже (текст) — белыми (такое «инвертирование» цвета делается в целях упрощения применения многих алгоритмов в дальнейшем), является задачей поиска оптимального значения между двумя пиками гистограммы. Для

решения этой задачи существуют хорошо изученные методы, например, метод Оцу [23] и его вариации.

## 1.2 Выделение региона интереса

На этом этапе на бинаризованном изображении выделяется непосредственно область, в которой находится распознаваемый текст, и отбрасываются элементы, текстом не являющиеся. К ним относятся такие объекты, как кляксы, пятна на бумаге, не удалённые в процессе бинаризации, картинки и др. Для их удаления можно, например, выделять компоненты связности на изображении, вычислять геометрические признаки и на их основе классифицировать компоненту связности как часть текста или дефект, используя методы машинного обучения или эвристики<sup>1</sup>.

## 2 Сегментация и нормализация текста

На этом этапе текст разделяется, или сегментируется, на удобные для анализа составные части. Наиболее естественными действиями на данном этапе является разделение текста на отдельные строки (*сегментация строк*) и разделение строк на слова (*сегментация слов*), а также, теоретически, разделение слов на элементарные составные части. Кроме того, на данном этапе проводится *нормализация* текста — приведение выделенных составных частей к некоторому стандартному виду для снижения вариативности и упрощения распознавания.

Среди методов сегментации текста можно выделить два основных подхода: «снизу вверх» (*bottom-up approach*) и «сверху вниз» (*top-down approach*). Основная идея подхода «снизу вверх» заключается в выделении компонент связности в тексте с последующим их объединением в более крупные образования, такие как слова, строки и разделы, используя отношения между отдельными частями текста. Подход «сверху вниз» заключается в последовательном разделении всего документа на всё более мелкие составные части, используя свойства документа как целого.

### 2.1 Сегментация строк

Задача сегментации (разделения) строк в машинопечатных документах на сегодняшний день считается полностью решённой [25]. Но в задаче НWR при разделении строк в общем случае возникают сложности, не позволяющие напрямую применять алгоритмы, пригодные для машинопечатных текстов [26]:

- 1) строки не только могут не являться параллельными, но и могут изгибаться;
- 2) различные строки могут быть слишком близки, а элементы текста, принадлежащего различным строкам, могут налагаться друг на друга.

---

<sup>1</sup>например, «если коэффициент формы области (отношения квадрата её периметра к площади) меньше некоторого значения, а площадь — больше некоторого значения, то это с большой вероятностью дефект» (т.к. рукописный текст обычно является некоторой кривой)

Среди методов разделения строк в тексте можно выделить [21]:

1. Метод горизонтальной проекции. Этот метод использует top-down подход и широко используется в задаче разделения строк в машинопечатных документах. Заключается он в следующем: для изображения считается так называемый *горизонтальный профиль проекции* (HPP, *horizontal projection profile*), представляющий собой сумму всех пикселей изображения вдоль горизонтального направления, затем на нём ищутся локальные минимумы. Так как рассматривается бинарное изображение, на котором пиксели текста имеют значение 1, а пиксели фона — 0, эти минимумы будут соответствовать межстрочным интервалам.

Для рукописного текста с достаточно большими межстрочными интервалами такой подход также подходит, но, как уже было сказано, в общем случае это предположение не выполняется. Тем не менее, возможно использование модифицированных методов, основанных на НРР, например, с использованием локального НРР [22].

2. Группировочные методы. Эти методы основаны на bottom-up подходе: компоненты связности объединяются в строки в зависимости от их геометрических свойств, таких как размер, форма и расстояние между ними. Также широко используются для разделения строк в машинопечатном тексте [?].

В рукописном тексте их использование ограничено в связи с возможным объединением частей текста из разных строк в одну связную компоненту. Однако возможно использовать модификации этих методов, например, использующие вероятностные карты документа [26] вместо строго связных компонент.

3. Выделение *базовых линий* (*baseline*). Эти методы основаны на идее, что человек пишет либо по, либо поверх некоторой воображаемой линии. Данные методы пытаются аппроксимировать эту линию, а затем восстановить по ней строку.

В [29] описывается подход, основанный на *преобразовании Хафа* (*Hough transform*) [?]. Поскольку преобразование Хафа выделяет прямые, его можно использовать для выделения базовых линий в случае, если они не слишком искривлены. В работе [29] преобразование Хафа применяется к центрам компонент связности пикселей текста. Такой подход требует, чтобы строки текста были близки к прямым, но зато позволяет выделять строки, расположенные в произвольном месте и идущие под произвольными углами.

В [?] описывается *метод отталкивания-притягивания* (*attractive-repulsive approach*). При проходе по странице сверху вниз базовые линии строятся одна за другой, так, что на каждую следующую линию действуют силы отталкивания со стороны уже построенных линий и силы притяжения со стороны пикселей текста.

В [18] описывается метод локальных минимумов. В нём компоненты связности пикселей текста рассматриваются как кривые линии, у которых ищутся локальные точки минимума. Затем эти точки минимума кластеризуются в прямые.

Пересечение элементов различных строк представляет собой проблему не только сегментации строк, но и распознавания текста, так как отнесение элемента к непра-

вильной строке очевидно ухудшает его распознаваемость. Пересекающиеся компоненты являются проблемой для методов горизонтальной проекции (так как они увеличивают значение профиля проекции в тех местах, где должен быть его минимум) и группировочных методов (так как они используют связные компоненты пикселей текста для построения строк), но слабо влияют на некоторые методы выделения базовых линий [21], в частности, [?], [18].

Для поиска пересекающихся элементов из различных строк можно использовать такие признаки, как размер компонент связности текста, факт отнесения одной компоненты к нескольким строкам или, напротив, неотнесения ни к какой строке. После нахождения таких «сомнительных» компонент нужно определить, относятся ли они к какой-то строке или же их нужно декомпозировать на элементы, относящиеся к разным строкам. Такая вертикальная декомпозиция компонент — сложная задача. Простое решение заключается в разрезании компоненты на части горизонтальными линиями, как это сделано в работах [19], [20], но можно применить и более тонкие подходы, например, выделение отдельных штрихов [17].

## 2.2 Сегментация слов

На этом этапе работы системы распознавания выделенные строки текста разделяются на отдельные слова. В отличие от машинописного текста, в котором расстояние между словами более-менее постоянно, а интервалы между символами внутри слова гораздо меньше, чем интервалы между словами, в рукописном тексте размер интервалов между словами может варьироваться в очень широких пределах.

Обычно при решении задачи сегментации слов действуют в рамках bottom-up подхода. Компоненты связности текста, отнесённые к одной строке на предыдущем этапе работы системы распознавания, объединяются в слова на этом этапе.

В работе [15] описывается использование лапласиана гауссиана (*Laplacian of Gaussian, LOG*) для выделения выделения слов на изображении и метод подбора параметров оператора. После применения оператора LOG близкие компоненты связности текста начинают сливаться вместе, что используется для выделения слов на исходном изображении.

Можно поставить альтернативную задачу: определить, является ли пробел между компонентами связности текста пробелом между словами или интервалом между буквами в слове, и решать эту задачу методами классификации. Поскольку расстояние между частями одного слова обычно меньше, чем между словами, распространёно использование так называемых «метрик разрыва» (*gap metric*): для интервала между компонентами связности вычисляется некоторая величина и сравнивается с порогом.

В работе [13] предлагается метод сегментации, основанный на трёхуровневой нейронной сети. В этом методе задача сегментации слов в предварительно выделенной строке рассматривается как задача классификации пустоты между двумя объектами с двумя классами: «пробел между словами» и «пробел внутри слова». Классификация проводится на основе геометрических признаков компонент связности текста в строке.

В работе [16] производится анализ штрихов в компонентах связности. Часть штрихов после этого отбрасывается, и для интервалов в получившемся изображении вы-

числяется метрика разрыва, которая затем сравнивается с порогом.

## 2.3 Нормализация

В силу высокой вариативности начертания слов их распознавание является очень сложным процессом. Нормализация служит для приведения слова к некоторому стандартному виду без значительной потери информации, необходимой для распознавания. Одними из наиболее часто употребляемых методов нормализации являются *slope correction* и *slant correction*.

*Slope* — это величина (угол) отклонения слова от горизонтальной линии. *Slant* — величина отклонения тех элементов слова, что должны быть вертикальными, от, собственно, вертикали [31]. Понятно, что угол наклона слова на его смысл не влияет, но может ухудшать его распознаваемость.

Простейший метод коррекции *slope* состоит в выполнении следующих шагов:

- 1) Ввести функционал качества на горизонтальном профиле проекции изображения слова (например, стандартное отклонение), зависящий от угла поворота изображения;
- 2) Максимизировать его на некотором диапазоне углов.

Фактически метод повторяет метод «горизонтализации» изображения в задаче OCR. Принцип работы можно видеть на рис. 1.

В работе [32] предложен метод коррекции *slope* методом выделения *центрального региона* (*core region*) слова. Центральным регионом слова называется область, в которой находятся основные части прописных букв слова. Для поиска центрального региона используется анализ горизонтального профиля проекции. Именно, на профиле проекции отыскивается глобальный максимум, и берутся его края. Далее минимумами штрихов, лежащих внутри центрального региона, приближается базовая линия слова. После этого изображение поворачивается так, чтобы базовая линия была горизонтальной. В работе [31] предлагается модификация метода выделения центрального региона путём анализа распределения значений горизонтального профиля проекции.

Существуют и другие методы, например, основанные на сглаживании (*smearing*) и линейной регрессии (см. рис. 2).

Существуют и другие методы нормализации, например, коррекция размера и выделение скелета текста, но они применяются реже.

## 3 Распознавание слова

После выделения слов в документе и приведения их к стандартному виду начинается процесс распознавания.

Распознавание рукописной последовательности символов можно определить как процесс поиска последовательности символов, наиболее подходящей под данные изображения.

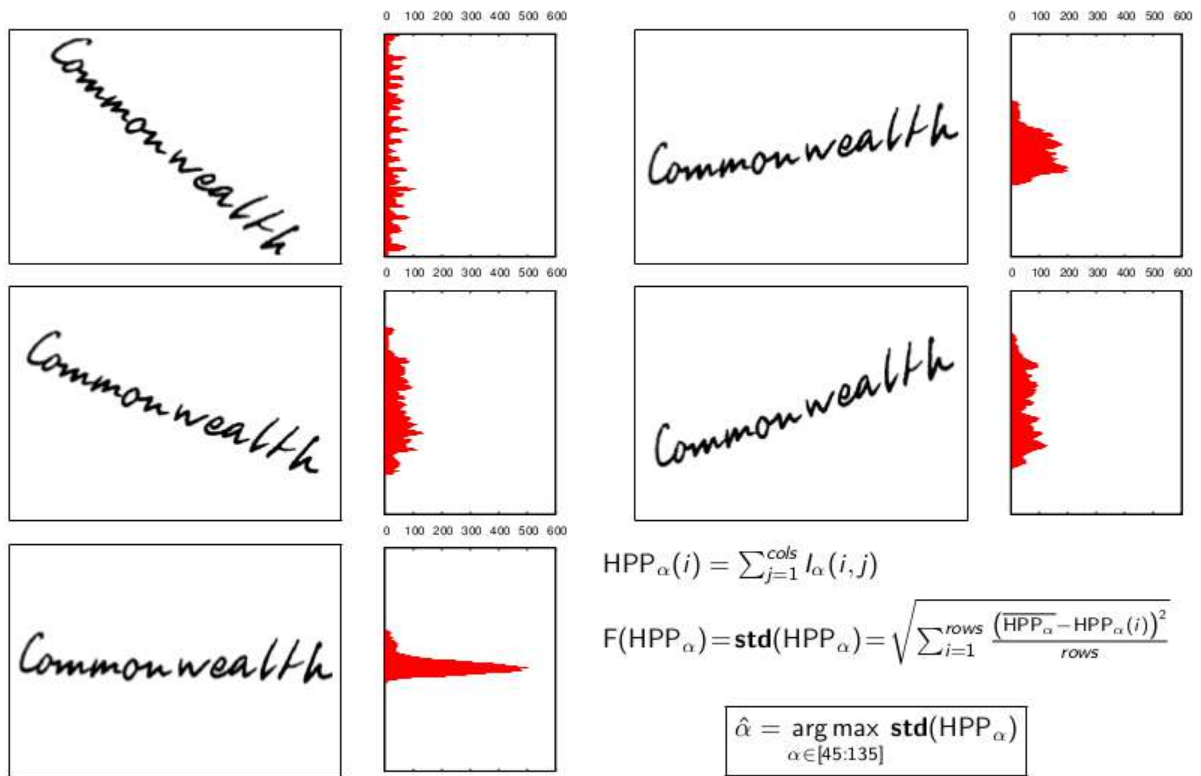


Рис. 1: Коррекция slope с использованием горизонтального профиля проекции

### 3.1 Распознавание слова целиком

В случае, если имеется сильно ограниченный словарь, слово можно попытаться распознать «как есть» — извлечь признаки непосредственно из всего слова и попытаться решить задачу классификации. Но даже в этом случае легко может оказаться, что признаков слишком много, и изменяются они в слишком большом диапазоне, чтобы можно было провести уверенную классификацию или даже построить модель. Поэтому почти всегда слово разбивается на какие-то более примитивные составные части, которые просто анализировать. Сегментация слов бывает явная (*explicit*) и неявная (*implicit*).

#### Явная сегментация слова

В этом подходе слово разбивается на более-менее интерпретируемые части. В идеальном случае это рукописные символы, но могут также быть, например, части букв или даже отдельные штрихи. Это является главным препятствием для использования методов распознавания, основанных на явной сегментации — на сегодняшний день не существует эффективного метода явного разбиения слов [1]. Стоит, однако, отметить, что если текст изначально устроен так, что символы отделены друг от друга (напр., каждый символ занимает одну клетку на бланке), эти подходы сработают.

Предположим сначала, что возможно представить слово как последовательность



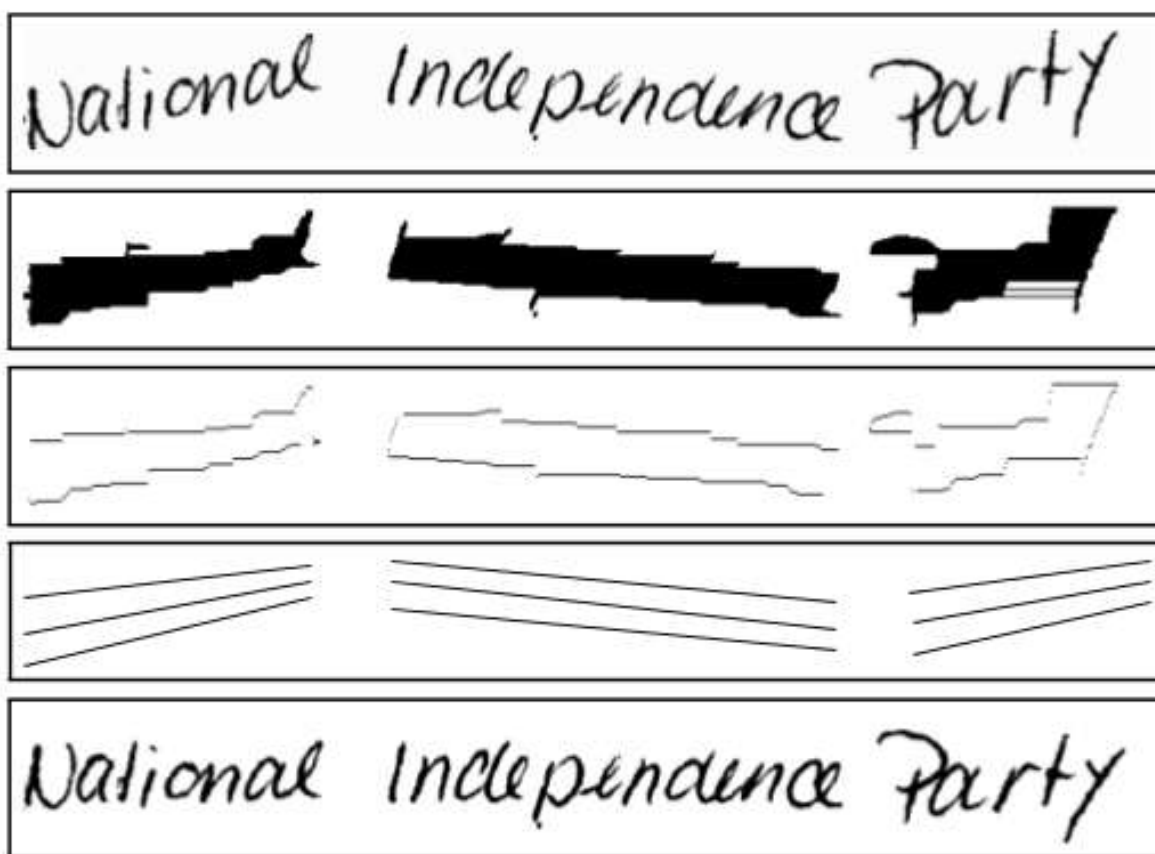


Рис. 2: Коррекция slope с использованием RLSA и линейной регрессии

отдельных символов. Тогда каждый символ можно представить как некоторый вещественный вектор признаков (обычно среди них выделяют *стохастические* (*частотные*) *признаки*, такие как доля белых пикселей, размеры символа, символ как вытянутая в строку матрица пикселей и др., и *геометрические*, такие как набор штрихов, составляющих символ, их геометрические характеристики и др.), после чего запустить какой-либо алгоритм классификации, где классы — все символы, которые могут встретиться в тексте. Поскольку классов очень много, и часть из них похожи (напр., «o» и «0»), рекомендуется использовать нейронную сеть типа MLP.

Сразу на символы разделить слово, скорее всего, не получится. В этом случае обычно к сегментации ставится требование: выделенные части должны быть значимыми элементами символов. После этого обычно задача переформулируется в терминах задач динамического программирования.

### Динамическое программирование

Рассмотрим процедуру динамического программирования. Пусть имеется слово, представленное в виде последовательности из  $T$  сегментов. В данный момент распознано  $n_1$  первых букв, составляющих  $t_1$  сегментов. Мы хотим набрать следующую букву

из следующих сегментов, и имеется некоторая *оценка штрафа*  $\xi(t; n_1, t_1)$  — какой штраф даётся за классификацию следующих  $t$  фрагментов как буквы при условии, что первые  $n_1$  букв уже классифицированы. Тогда фактически нужно решить задачу дискретной оптимизации — найти, какой путь (т.е. какая интерпретации последовательности фрагментов) минимизирует этот штраф.

Оценка штрафа должна получаться, во-первых, на основании анализируемых  $t$  сегментов, т.е. нужна какая-либо система распознавания образов (напр., нейросеть), которая принимает на вход признаки фрагмента и вместо привычного класса будет возвращать некоторый штраф, во-вторых, на основании уже полученной последовательности букв, т.е. стоит использовать какую-либо модель языка (например, марковская языковая модель или просто словарь).

## Неявная сегментация слова

В этом подходе слово разбивается на большое количество достаточно мелких (гораздо меньших символа) фрагментов, которые зачастую накладываются друг на друга [1]. Фактически, при анализе используется метод «скользящего окна». Одним из основных методов анализа таких образований являются марковские модели. Почти всегда в системах неявного распознавания используется комбинация из модели письма и модели распознавания.

## Марковские модели письма

Поставим задачу следующим образом: описать каждый символ некоторой скрытой марковской моделью. Архитектура модели обычно принимается линейной с возможностью для каждого состояния перейти либо на одно состояние вперёд, либо в себя. Пространство наблюдаемых переменных полагается непрерывным, обычно  $\mathbb{R}^N$ .

Пусть символ разделён на большое число вертикальных «полосок». Тогда каждую такую «полоску» можно перевести в признаковое описание, которое, в свою очередь, можно интерпретировать как некоторый наблюдаемый параметр в скрытой марковской модели, описывающей данный символ. Тогда можно обучить эту модель, используя, например, алгоритм Баума-Велша.

В связи с тем, что для модели с дискретным набором наблюдаемых параметров алгоритмы обучения и распознавания работают очень быстро, часто предлагается проводить векторное квантование входных данных. Альтернативой является комбинирование дискретной модели с нейронной сетью (особенно с RBF-сетью, т.к. в таком случае и модель, и сеть по обучаются по сути вариациями EM-алгоритма) [1].

## Марковские модели языка

Для улучшения качества распознавания символов обычно составляется т.н. модель языка. Связано это с тем, что в языке может встретиться далеко не любая последовательность символов, а те, которые могут, не равновероятны. Поэтому некоторое понимание того, что может встретиться в тексте, обычно позволяет улучшать качество распознавания, а также, возможно, исправлять ошибки, допущенные автором.

Самой простой и жёсткой моделью языка является его словарь — список всех допустимых слов языка. Для решения задачи распознавания в общем случае он, впрочем, неприменим, т.к. начинает очень быстро разрастаться.

Более гибкой альтернативной моделью является  $n$ -граммная модель, которая представляет собой случайный процесс с дискретными состояниями — символами алфавита, для которого вероятность перехода из текущего состояния в следующее зависит от  $n$  предыдущих:  $P(w_{k+1}) = P(w_{k+1}|w_k, w_{k-1}, \dots, w_{k-n})$ . При  $n = 1$  модель превращается в марковский процесс с дискретным множеством наблюдаемых состояний (символы алфавита), совпадающим с множеством скрытых состояний. Такой модели зачастую оказывается достаточно для практических целей, более того, её можно расширить, введя в модель специальные состояния, соответствующие широко распространённым в языке диграммам и триграммам.

Комбинация марковских моделей письма и языка обычно проводится следующим образом:  $P(w_n) = P(w_n|w_{n-1})p(w_n)^\rho$ , где  $P(w_n)$  — вероятность того, что текущий символ — буква  $w$  в общей модели,  $P(w_n|w_{n-1})$  — вероятность того, что текущий символ — буква  $w$  в языковой модели,  $p(w_n)$  — вероятность того, что текущий символ — буква  $w$  в модели письма,  $\rho$  — весовой параметр [1].

## Выводы

Задача оффлайн-распознавания рукописного текста на данный момент полностью не решена и, по всей видимости, решена ещё долго не будет.

Залогом успешного распознавания текста являются успешные предобработка и сегментация.

Для распознавания слов наиболее многообещающим выглядит метод неявной сегментации и использования марковских моделей для распознавания.

# Литература

- [1] *Plötz, Thomas and Fink, Gernot A.* Markov models for offline handwriting recognition: a survey // International Journal on Document Analysis and Recognition (IJ DAR), **12**, 4 (269–298), 2009.
- [2] *Brakensiek, Anja and Willett, Daniel and Rigoll, Gerhard.* Improved degraded document recognition with hybrid modeling techniques and character n-grams // Pattern Recognition, IEEE, 2000. Proceedings. 15th International Conference on, **4**, (438–441), 2000.
- [3] *El Abed, Haikal and Margner, Volker.* Comparison of different preprocessing and feature extraction methods for offline recognition of handwritten arabic words, Document Analysis and Recognition, 2007. IEEE. ICDAR 2007. Ninth International Conference on, **2**, (974–978), 2007.
- [4] *Feng, Bing and Ding, Xiaoqing and Wu, Youshou.* Chinese handwriting recognition using hidden Markov models // Pattern Recognition, 2002. (IEEE). Proceedings. 16th International Conference on, **3**, (212–215), 2002.
- [5] *Bunke, H and Bengio, Samy and Vinciarelli, A.* Offline recognition of unconstrained handwritten texts using HMMs and statistical language models // Pattern Analysis and Machine Intelligence, IEEE Transactions on, **26**, 6, (709–720), 2004.
- [6] *Wienecke, Markus and Fink, Gernot A and Sagerer, Gerhard.*, Toward automatic video-based whiteboard reading // International Journal of Document Analysis and Recognition (IJ DAR), **7**, 2–3, (188–200), 2005.
- [7] *Natarajan, Prem and Saleem, Shirin and Prasad, Rohit and MacRostie, Ehry and Subramanian, Krishna.* Multi-lingual offline handwriting recognition using hidden Markov models: A script-independent approach. — Arabic and Chinese Handwriting Recognition, (231–250), Springer, 2008.
- [8] *Pastor, Moisés and Toselli, Alejandro and Vidal, Enrique.* Projection profile based algorithm for slant removal. — Image Analysis and Recognition, (183–190), Springer, 2004.
- [9] *Côté, Myriam and Lecolinet, Eric and Cheriet, Mohamed and Suen, Ching Y.* Automatic reading of cursive scripts using a reading model and perceptual concepts // International Journal on Document Analysis and Recognition, **1**, 1, (3–17), 1998.

- [10] Kavallieratou, Ergina and Fakotakis, Nikos and Kokkinakis, G. Slant estimation algorithm for OCR systems // *Pattern Recognition*, **34**, 12, (2515–2522), 2001.
- [11] *Uchida, Seiichi and Taira, Eiji and Sakoe, Hiroaki*. Nonuniform slant correction using dynamic programming // *Document Analysis and Recognition*, 2001. Proceedings. Sixth International Conference on, (434–438), 2001.
- [12] *Taira, Eiji and Uchida, Seiichi and Sakoe, Hiroaki*. Nonuniform slant correction for handwritten word recognition // *IEICE TRANSACTIONS on Information and Systems (The Institute of Electronics, Information and Communication Engineers)*, **87**, 5, (1247–1253), 2004,
- [13] *Huang, Chen and Srihari, Sargur N*. Word segmentation of off-line handwritten documents. *Electronic Imaging 2008 (International Society for Optics and Photonics)*, (68150E–68150E), 2008.
- [14] *Kim, Gyeonghwan and Govindaraju, Venu*. Handwritten phrase recognition as applied to street name images // *Pattern Recognition*, **31**, 1, (41–51), 1998.
- [15] *Manmatha, Raghavan and Srimal, Nitin*. Scale space technique for word segmentation in handwritten documents. — *Scale-Space Theories in Computer Vision*, (22–33), 1999, Springer.
- [16] *Park, Jaehwa and Govindaraju, Venu and Srihari, Sargur N*. Efficient word segmentation driven by unconstrained handwritten phrase recognition, *Document Analysis and Recognition*, 1999. (IEEE), ICDAR’99. Proceedings of the Fifth International Conference on, (605–608), 1999.
- [17] *Nakajima, Yu and Mori, Shunji and Takegami, Shuuki and Sato, S*. Global methods for stroke segmentation // *International Journal on Document Analysis and Recognition*, **2**, 1, (19–23), 1999,
- [18] *Kim, Gyeonghwan and Govindaraju, Venu and Srihari, Sargur N*. An architecture for handwritten text recognition systems // *International Journal on Document Analysis and Recognition*, **2**, 1, (37–44), 1999,
- [19] *Likforman-Sulem, Laurence and Faure, Claudie*. Extracting text lines in handwritten documents by perceptual grouping. — *Advances in handwriting and drawing: a multidisciplinary approach*, (117–135), 1994, Europa, Paris.
- [20] *Likforman-Sulem, Laurence and Faure, Claudie*. Une méthode de résolution des conflits d’alignements pour la segmentation des documents manuscrits // *Traitement du signal*, **12**, 6, (541–549), 1995.
- [21] *Likforman-Sulem, Laurence and Zahour, Abderrazak and Taconet, Bruno*. Text line segmentation of historical documents: a survey // *International Journal of Document Analysis and Recognition (IJDAR)*, **9**, 2-4, (123–138), 2007.

- [22] *Bar-Yosef, Itay and Hagbi, Nate and Kedem, Klara and Dinstein, Itshak.* Line segmentation for degraded handwritten historical documents // Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on, (1161–1165), 2009.
- [23] *Otsu, Nobuyuki.* A threshold selection method from gray-level histograms // Automatica, **11**, 285-296, (23–27), 1975.
- [24] *Шануро, Л. and Стокман, Дж.* Компьютерное зрение // Лаборатория знаний, **752**, (8), 2006.
- [25] *Plamondon, Réjean and Srihari, Sargur N.* Online and off-line handwriting recognition: a comprehensive survey // Pattern Analysis and Machine Intelligence, IEEE Transactions on, **22**, 1, (63–84), 2000.
- [26] *Li, Yi and Zheng, Yefeng and Doermann, David and Jaeger, Stefan.* Script-independent text line segmentation in freestyle handwritten documents // Pattern Analysis and Machine Intelligence, IEEE Transactions on, **30**, 8, (1313–1329), 2008.
- [27] *O’Gorman, Lawrence.* The document spectrum for page layout analysis // Pattern Analysis and Machine Intelligence, IEEE Transactions on, **15**, 11, (1162–1173), 1993.
- [28] *Duda, Richard O and Hart, Peter E.* Use of the Hough transformation to detect lines and curves in pictures // Communications of the ACM, **15**, 1, (11–15), 1972,
- [29] *Likforman-Sulem, Laurence and Hanimyan, Anahid and Faure, Claudie.* A Hough based algorithm for extracting text lines in handwritten documents // Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on, **2**, (774–777), 1995.
- [30] *Öztop, E and Mülayim, Adem Yasar and Atalay, Volkan and Yarman-Vural, Fatos.* Repulsive attractive network for baseline extraction on document images // Signal Processing, **75**, 1, (1–10), 1999.
- [31] *Vinciarelli, Alessandro and Luettin, Juergen.* A new normalization technique for cursive handwritten words // Pattern Recognition Letters, **22**, 9, (1043–1050), 2001.
- [32] *Bozinovic, Radmilo M. and Srihari, Sargur N.* Off-line cursive script word recognition // Pattern Analysis and Machine Intelligence, IEEE Transactions on, **11**, 1, (68–83), 1989.
- [33] *de Zeeuw, Frank* Slant Correction Using Histograms // Undergraduate Thesis [http://www. ai. rug. nl/~ axel/teaching/bachelorprojects/zeeuw\\_slant\\_correction. pdf](http://www.ai.rug.nl/~axel/teaching/bachelorprojects/zeeuw_slant_correction.pdf), 2006
- [34] *Vinciarelli, Alessandro and Luettin, Juergen.* Offline cursive handwriting: From word to text recognition // Pattern Recognition Letters, 2003, IDIAP.

- [35] *Sayre, Kenneth M.* Machine recognition of handwritten words: A project report // Pattern recognition, **5**, 3, (213–228), 1973.
- [36] *Lavrenko, Victor and Rath, Toni M and Manmatha, R.* Holistic word recognition for handwritten historical documents // Document Image Analysis for Libraries, 2004. Proceedings. First International Workshop on, (278–287), 2004.
- [37] *Adamek, Tomasz and O'Connor, Noel E and Smeaton, Alan F.* Word matching using single closed contours for indexing handwritten historical documents // International Journal of Document Analysis and Recognition (IJ DAR), **9**, 2-4, (153–165), 2007.
- [38] *Алгоритм Баума — Велша* <http://ru.wikipedia.org/?oldid=63080905>, urldate = 2014-11-22.
- [39] *Алгоритм Витерби* <http://ru.wikipedia.org/?oldid=62968928> urldate = 2014-11-22.
- [40] *Kuchuganov, Aleksander Valeryevich.* The Description and Analysis of Graphical Information in a Description Logic Formal Language 2013
- [41] *Зеленцов, И.А., Филиппович, Ю.Н.* Распознавание образов на основе структурных фреймовых описаний в скорописных текстах XVII в // Наука и образование: электронное научно-техническое издание, 2011,