DOI: 10.15514/ISPRAS-2020-2(3)-32

Logical structure extraction from scanned documents

[1,2]*Bogatenkova A. O., ORCID: 0000-0001-8679-1568 <nastyboget@ispras.ru>*
[1]*Kozlov I. S., ORCID: 0000-0002-0145-1159 <kozlov-ilya@ispras.ru>*
[1]*Belyaeva O. V., ORCID: 0000-0002-6008-9671 <belyaeva@ispras.ru>*
[1,2]*Perminov A. I., ORCID: 0000-0001-8047-0114 <perminov@ispras.ru>*
[1]*Ivannikov Institute for System Programming of the RAS,*
25, Alexander Solzhenitsyn Str., Moscow, 109004, Russia
[2]*Lomonosov Moscow State University,*
GSP-1, Leninskie Gory, Moscow, 119991, Russian Federation

Abstract. Logical structure extraction from various documents has been a longstanding research topic because of its high influence on a wide range of practical applications. A huge variety of different types of documents and, as a consequence, the variety of possible document structures make this task particularly difficult. The purpose of this work is to show one of the ways to represent and extract the structure of documents of a special type. We consider scanned documents without a text layer. This means that the text in such documents can not be selected or copied. Moreover, you can not search for the content of such documents. However, a huge number of scanned documents exist that one needs to work with. Understanding the information in such documents may be useful for their analysis, e. g. for the effective search within documents, navigation and summarization. To cope with a large collection of documents the task should be performed automatically. The paper describes the pipeline for scanned documents processing. The method is based on the multiclass classification of document lines. The set of classes includes textual lines, headers and lists. Firstly text and bounding boxes for document lines are extracted using OCR methods, then different features are generated for each line, which are the input of the classifier. We also made available dataset [1] of documents, which includes bounding boxes and labels for each document line; evaluated the effectiveness of our approach using this dataset and described the possible future work in the field of document processing.

Keywords: machine learning; document structure; natural language processing; OCR.

For citation: Bogatenkova A. O., Kozlov I. S., Belyaeva O. V., Perminov A. I. Logical structure extraction from scanned documents. Trudy ISP RAN/Proc. ISP RAS, 2020, vol. 1, issue 2, pp. 3–4. 10.15514/ISPRAS-2020-2(3)-32

Извлечение логической структуры из сканированных документов

[1,2]*Богатенкова А. О., ORCID: 0000-0001-8679-1568 <nastyboget@ispras.ru>*
[1]*Козлов И. С., ORCID: 0000-0002-0145-1159 <kozlov-ilya@ispras.ru>*
[1]*Беляева О. В., ORCID: 0000-0002-6008-9671 <belyaeva@ispras.ru>*
[1,2]*Перминов А. И., ORCID: 0000-0001-8047-0114 <perminov@ispras.ru>*
[1]*Институт системного программирования им. В.П. Иванникова РАН, 109004, Россия, г. Москва, ул. А. Солженицына, д. 25*
[2]*Московский государственный университет имени М.В. Ломоносова, 119991, Россия, Москва, Ленинские горы, д. 1.*

Аннотация. Большое количество информации представлено в виде сканированных документов без текстового слоя, то есть текст документа не копируется и по нему нельзя осуществлять поиск. Однако зачастую требуется осуществлять быстрый поиск по их содержимому. Знание структуры документов может способствовать более эффективному их анализу. В статье предложен пайплайн обработки сканированных документов, а также разработан метод извлечения структуры из них. Данный метод основан на многоклассовой классификации строк документа, в том числе классификации на заголовки и списки. Пайплайн состоит из извлечения текста и рамок строк документов с помощью методов OCR, формирования признаков и обучения классификатора на данных признаках. Кроме того, размечен и доступен для изучения [1] корпус документов, проведена экспериментальная проверка реализованного метода на данном корпусе и описаны возможности для дальнейшей работы и исследований.

Ключевые слова: машинное обучение; структура документа; обработка естественного языка; OCR.

Для цитирования: Богатенкова А. О., Козлов И. С., Беляева О. В., Перминов А. И. Извлечение логической структуры из сканированных документов. Труды ИСП РАН, 2020, том 1 вып. 2, с. 3–4. 10.15514/ISPRAS-2020-2(3)-32

1. Введение

Документы имеют определённую логическую структуру. Например, законы делятся на главы, статьи, разделы. Научные статьи состоят из аннотации, введения, обзора существующих работ и других секций. Информация о логической структуре полезна для автоматического анализа документа.

Задача автоматического извлечения логической структуры из документов осложняется следующими факторами:

Во-первых, большое количество документов представляет из себя сканированные изображения с бумажных носителей. Такие изображения не содержат текстовый слой, для его извлечения необходимо использовать методы обработки изображений.

Во-вторых, зачастую логические части документа выделяются с помощью форматирования: увеличенного размера шрифта, жирности, отступов. Такая информация помогает читателю лучше понимать структуру документа, автоматическая система также должна учитывать эти признаки.

В-третьих, разные типы документов организованы отличным образом. Например, научные статьи, финансовые отчёты, законы могут состоять из разных структурных элементов (законы имеют главы, статьи, пункты, подпункты; научные статьи состоят из введения, аннотации, списка литературы). Форматирование и язык документов также могут быть различны (законы, как правило, пишут в 1 колонку, научные статьи – в 1-2 колонки). Задача создать систему, способную обработать любой тип документа может оказаться слишком сложной, задача построения своей системы для каждого типа документов "с нуля"может оказаться слишком трудоёмкой. Таким образом, наша система должна позволять добавлять поддержку новых типов документов, причём такое добавление не должно быть слишком трудоёмким.

Многие типы документов имеют требования по оформлению, однако эти требования не являются полностью формализованными, кроме того составители документов могут от них отклоняться. В связи с описанным выше разнообразием можно сделать вывод, что для выделения логической структуры лучше подходят методы машинного обучения.

В данной статье описан метод извлечения структуры документа в виде заголовков, элементов списков и текстовых строк. Каждая строка документа относится к одному из этих трех типов на основе определённых признаков. Для выделения признаков может быть необходима метаинформация, такая как размер и тип шрифта, отступы, междустрочные интервалы и т. д. Поэтому извлечение логической структуры целесообразно делать на этапе анализа изображений.

Данный анализ предполагает следующее: с помощью методов OCR из изображений извлекаются строки документа с текстом и координатами их рамок на изображении. Следующим шагом пайплайна является выделение признаков на основе извлечённых данных. Далее выбранным алгоритмом машинного обучения проводится многоклассовая классификация строк.

Статья организована следующим образом: глава 2 содержит обзор различных подходов, с помощью которых решается задача выделения структуры документа; в главе 3 раскрывается процесс составления обучающего набора данных, в частности описывается набор документов, используемый при реализации и проверке метода и манифест для разметки данных; в главе 4 рассматривается реализованный метод; в главе 5 показаны результаты экспериментальной проверки метода, сравнение различных методов машинного обучения, анализ ошибок и анализ важности признаков, а в главе 6 представлены краткие выводы и предлагаются возможности для дальнейшей работы и исследований.

2. Обзор аналогичных работ

Применяется множество разнообразных подходов [2—4], которые позволяют выделять в тексте заголовки и распознавать логическую структуру документов. Среди подходов можно выделить следующие:

- на основе оглавления;
- на основе правил;
- на основе машинного обучения.

2.1 Извлечение структуры из документов на основе оглавления и правил

По анализу содержимого и структуре изображений документов проводятся соревнования ICDAR [5—7]. В одном из таких соревнований [5] производилось извлечение структуры из книг, содержимое которых было получено с помощью оптического распознавания символов. Структура книг в виде разбиения на страницы, параграфы, главы извлекалась с использованием оглавления, которое присутствовало в большинстве книг.

В 2019 году проводились соревнования FinTOC [8], где из финансовых документов извлекалась структура в виде иерархии уровней заголовков документов. Максимальная глубина уровней равна пяти. Одна из команд-участниц [2] извлекала необходимую структуру используя оглавление документов, а также систему правил, которые применялись для определения иерархии заголовков. Сначала идентифицировались страницы, содержащие текст оглавления, затем в документе находились страницы, соответствующие заголовкам, указанным в оглавлении. Последним шагом являлось выделение иерархии найденных заголовков, основанное на применении правил: анализировались такие признаки, как междустрочный интервал, отступ, шрифт, символы нумерации. Использованный подход позволил получить достаточно высокую точность, но низкую полноту, так как некоторые оглавления документов были неполными.

Извлечение структуры документов на основе оглавления имеет ряд недостатков. Во-первых, невозможно обрабатывать документы, в которых нет оглавления. Во-вторых, при использовании этого метода в структуру документа не будут включаться заголовки, которые не вошли в оглавление, например, заголовки более низкого уровня. В-третьих, данный метод не позволяет извлекать элементы маркированных и нумерованных списков, которые не включаются в оглавление документа.

2.2 Извлечение структуры из документов на основе машинного обучения

В соревнованиях [8] кроме извлечения иерархической структуры документов решалась задача определения, является ли конкретный блок документа заголовком. Командам был дан набор pdf-документов, xml-файлов с выделенными блоками до-

кументов, а также набор признаков для каждого блока: является ли шрифт блока жирным, курсивом, состоит ли текст из заглавных букв, начинается с заглавной буквы или с нумерации. Кроме данных признаков, каждая из команд использовала различные дополнительные морфологические, семантические, лингвистические признаки. На основе этих признаков обучались различные классификаторы: SVM, MNB, Extra Tree, Decision Tree, Gradient Boosting. Для оценки результатов использовалась F1-мера, максимальный score в соревновании – 0,982.

Победители соревнования [3] создали новый датасет для обучения с помощью аугментации данных, перевели новые сгенерированные текстовые блоки в векторное представление, а затем использовали рекуррентные нейронные сети LSTM и BiLSTM для решения задачи классификации.

Для того, чтобы классифицировать строки документа, логично использовать такие признаки, как жирность шрифта, отступы, высоту текста и т. д. Эти признаки сильно отличаются друг от друга диапазонами значений, поэтому нейронные сети LSTM и BiLSTM, как правило, плохо подходят для решения данной задачи. Кроме того, описанный подход применялся для определения заголовков и не распространялся на элементы списков.

В статье [4] структура документа извлекалась с использованием методов машинного обучения, включая глубокое обучение. Цель данной работы – автоматически идентифицировать и классифицировать различные секции документов и понять их смысл в рамках документа (назначить семантическую метку).

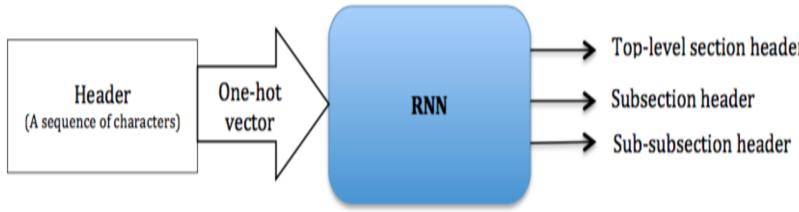


Рис. 1. Вход и выход классификатора заголовков
Fig. 1. Input and output of the section classifier

Классификатор (рис. 1), который был использован при решении задачи, состоит из нескольких частей. Сначала строки документа подаются на вход классификатору (классификатор строк), который определяет, является ли строка заголовком, затем строки-заголовки классифицируются точнее другими классификаторами (классификаторы секций). В этом решении структура документа имела вложенность 3, то есть предполагалось выделение секций, подсекций, подподсекций. Кроме того, в

данной работе был размечен датасет, на котором происходило обучение модели. Метрика качества - F1-мера, при идентификации заголовков итоговый score – 0,96; при классификации секций средний F1-score – 0,81.

Данный подход ограничивает число уровней вложенности извлекаемой структуры и так же не позволяет извлекать элементы списков. Разработанный нами метод позволяет извлекать структуру без ограничения уровней вложенности.

3. Набор данных и манифест

3.1 Описание данных

Датасет представляет собой набор документов в виде изображений в формате JPEG, скачанный с сайта zakupki.gov.ru [9]. Набор данных доступен для изучения [1].

Анализируемые документы являются сканированными копиями страниц текстов государственных закупок. Каждое изображение будем считать отдельным документом. Из рассмотрения удалены документы, содержащие таблицы, рисунки, рамки и прочие нетекстовые элементы.

Данный корпус имеет ряд специфических особенностей. Текст расположен в одной колонке, не выделен цветом, шрифт не меняется (меняется только его начертание или размер). Большая часть всех текстов написана на русском языке, редко встречаются латинские буквы. Так как содержимое документов представляет собой в основном договоры предприятий, в текстах встречается большое количество элементов списков (нумерованных и маркированных), зачастую списки имеют очень глубокий уровень вложенности (четвертый, пятый). Заголовков относительно немного, они могут быть пронумерованы и также иметь глубокий уровень вложенности, поэтому их можно спутать с элементами списков.

Поскольку сканированная копия может быть сделана с любой страницы документа, а не только первой, на некоторых страницах (которые мы считаем отдельным документом) могут отсутствовать заголовки или элементы списков.

3.2 Разметка данных

Для создания обучающего набора данных часто используют ручную разметку – предлагают выполнить задачу классификации человеку-аннотатору.

В книге [10] предлагается выполнять разметку обучающего корпуса в следующем порядке:

1. Спецификация задания – формальное определение задания и формата данных, используемое ПО и так далее.
2. Составление Манифеста – инструкции для аннотаторов.
3. Разметка данных. Непосредственно разметка данных с учётом Манифеста
4. Измерение согласованности аннотаторов. На этом шаге проверяется то, что размечающие понимают задание одинаково. Если это не так, то производится

возврат к шагу 2

5. Вынесение решения по аннотациям – если производилась разметка с перекрытием (один и тот же пример размечался более чем одним анотатором), возникает необходимость объединить их результаты. Мы пропустили этот шаг, так как аннотаторы производили разметку без перекрытия.

Опишем эти шаги подробнее применительно к нашей задаче.

3.2.1 Спецификация задания

Мы поставили нашу задачу как многоклассовую классификацию строк. Аннотаторам последовательно показывались изображения сканированного документа, одна из строк которого обведена в красную рамку. Аннотатору было необходимо отнести текст в рамке к одному из наперёд заданных классов. В нашей задаче выполнялась классификация на следующие классы:

- Заголовок;
- Элемент списка;
- Простой текст;
- Другое (не текст).

Таким образом, аннотатору ставилась задача классификации изображения. Для выполнения этого задания использовалась система собственной разработки [11]. После разметки данные сохранялись в формате JSON, формат описан в Приложении (рис. 4).

3.2.2 Манифест

В предыдущей секции мы определили задание для разметки как классификацию изображения на один из нескольких классов. Теперь необходимо определить, чем должны руководствоваться аннотаторы, относя изображение к тому или иному классу, в этом им помогает *манифест* или инструкция для разметки (основную часть манифеста можно посмотреть в Приложении, в [1] содержится полный текст манифеста).

Как правило, невозможно полностью формально описать правила классификации (в противном случае нам нет необходимости использовать машинное обучение). Поэтому в манифесте допустимо использование нестрого определённых понятий и правил (например: *жирный текст, выравнивание по центру, изображение синей печати*). При этом важно, чтобы все аннотаторы понимали задание одинаково. Для того, чтобы убедиться в этом, вычисляется согласованность. Низкая согласованность может означать то, что манифест написан недостаточно хорошо/понятно и требует доработки.

3.2.3 Согласованность

Для проверки одинакового и правильного понимания задания аннотаторами необходимо измерить их согласованность:

1. Предложить нескольким аннотаторам независимо выполнить разметку одного и того же множества заданий.
2. Вычислить специальную статистику, показывающую, насколько согласована разметка.
3. В случае низкой согласованности рекомендуется разобрать спорные ситуации и обновить манифест, лучше прописав правила для спорных ситуаций и добавив примеров.

Хороший обзор о методах вычисления согласованности можно найти в статьях [12] и [13]. Для проверки правильности разметки была посчитана специальная статистика Cohen's kappa, принимающая значения ≤ 1. Чем ближе κ к 1, тем выше согласованность. После разметки десяти документов (407 строк) двумя аннотаторами значение статистики κ оказалось равным 0.975, что считается высоким уровнем согласованности.

После чего было решено размечать остальной корпус документов. В результате было размечено 600 документов (21350 строк) и отдельные JSON файлы были объединены в один (рис. 4 в Приложении).

Стоит отметить, что высокая согласованность ещё не означает, что задание выполнено "правильно". Так, если все аннотаторы будут всегда относить каждое изображение к классу *"Простой текст"*, не обращая внимания на признаки и манифест, то κ будет равна 1, но вряд ли это можно назвать правильной разметкой. Такая проблема актуальна при использовании краудсорсинга. Обзор методов краудсорсинга можно найти в статье [14].

4. Описание решения

4.1 Описание реализованного метода

Первым шагом в решении задачи является выделение текста и рамок для строк документа с помощью методов OCR. Вторым шагом является составление вектора признаков для каждой строки документа. С помощью выделенной текстовой информации извлекаются различные текстовые признаки, описанные в главе 4.2. Для визуальных признаков используется информация из координат рамок (для отступов и высоты). В определении жирности шрифта используется само изображение документа и координаты рамок строк внутри изображения.

Таким образом, каждый документ представлен набором векторов признаков для строк, а тренировочные данные являются объединением данных для документов. Следующим шагом в решении задачи является применение алгоритма машинного обучения, который на основе выделенных признаков распределит строки по клас-

сам. Схема пайплайна показана на рис. 2

Рис. 2. Пайплайн обработки документов
Fig. 2. Pipeline for documents processing

4.2 Выделение признаков

Среди признаков, характеризующих строки документа, можно выделить следующие группы:

- Признаки, основанные на регулярных выражениях.
  Данная группа признаков основывается на анализе начала и конца каждой строки. Такие признаки очень важны для выявления элементов списков различных типов, а также могут сигнализировать о конце заголовка или начале списка.
  Регулярные выражения позволяют выделить следующие признаки:
  - начинается ли строка с цифры или буквы со скобкой или точкой (также анализируются иерархические выражения вида 1.1.1);
  - начинается ли строка с тире (и других символов, характерных для маркированного списка);
  - состоит ли строка целиком из заглавных букв (характерно для некоторых заголовков);
  - начинается ли строка с заглавной (строчной) буквы;
  - начинается ли строка с конкретных слов типа «Раздел», «Секция», «Глава» и т. д.;
  - оканчивается ли строка символами вида «. , ; :»;
  - оканчивается ли строка строчной буквой.
- Текстовые признаки.
  Данная группа признаков связана с подсчетом некоторых строковых характеристик, а именно:
  - количество букв в первом и втором словах строки;
  - количество слов в строке (строка разбивается на слова по пробелам);
  - количество символов в строке (длина строки).
- Визуальные признаки.

Данная группа признаков связана с графическим представлением текста в документе. То есть при анализе строки рассматривается не ее текст, а следующие признаки (первые три признака измеряются в пикселях):

- отступ от левого края страницы;
- высота текста строки (точнее высота ограничивающей ее рамки);
- отступ от верхнего края страницы;
- жирность шрифта различных уровней (подробнее см. в главе 4.2.1).

Важно рассматривать не одну строку, а ее окрестность, для увеличения точности классификатора строк. Поэтому к признакам каждой строки добавляются признаки четырех предыдущих и последующих строк.

Для строк, которые начинаются с нумерации, определяется, есть ли в документе строка, предшествующая данной с нумерацией, меньшей данной на единицу.

И, наконец, для каждого документа вычисляется средний отступ от левого края страницы, средняя высота шрифта, средняя длина строки, среднее число слов в строках, среднее значение для жирности шрифта (ядро свертки 5), среднее число букв в первом слове каждой строки. Данные значения добавляются к признакам каждой строки документа.

4.2.1 Определение жирности шрифта

Рассмотрим более подробно способ определения жирности шрифта. Для этого использовались морфологические операции Dilation и Erosion [15] из библиотеки OpenCV [16]. Данные операции с помощью комбинаций увеличения и сужения границ на изображении позволяют детектировать жирность текста.

При работе с текстом в качестве исходного изображения используется bounding box конкретной строки, полученный с помощью tesseract [17]. В данной работе функции erode и dilate запускались с параметром kernel, равным от 2 до 7 включительно (6 раз).

5. Экспериментальная проверка метода

5.1 Подбор классификатора

При решении задачи было опробовано множество методов машинного обучения, для лучших из них проведен анализ результатов. В анализе участвовало 4 классификатора:

- алгоритм k ближайших соседей (KNeighborsClassifier);
- логистическая регрессия (LogisticRegression);
- градиентный бустинг (GradientBoostingClassifier);
- "extreme"градиентный бустинг (XGBClassifier).

Набор размеченных документов тремя способами был разбит на тренировочное и тестовое множества. Разбиение производилось по документам, то есть группа строк, относящихся к одному документу попадала целиком либо в тренировочное, либо в тестовое множество. На каждом разбиении было проведено обучение классификаторов и вычисление F1-score (с макро-усреднением). Усредненные значения F1-score для каждого классификатора указаны в табл. 1.

Табл. 1. Сравнение классификаторов
Table 1. Classifier comparison

Классификатор	**F1-score**
Nearest Neighbors	0.89
Logistic Regression	0.9
Gradient Boosting	0.92
XGBoost	**0.95**

В целом, все рассмотренные классификаторы показали хороший результат (табл. 1), наилучший результат показал XGBClassifier, поэтому было решено выбрать его.

5.2 Анализ значимости признаков

Анализ значимости признаков был проведен с помощью библиотеки xgbfir [18] . В табл. 2 представлены первые 10 признаков с наивысшей значимостью (information gain). Это признаки, которые имеют наибольший вес при вычислении предсказания классификатора.

Табл. 2. Значимость признаков
Table 2. Features importances

Признак	**Information gain**
Число символов первого слова в строке	22089
Индикатор, является ли строка продолжением списка	2400
Жирность шрифта	2368
Число слов в строке	2263
Отступ от левого края страницы	1557
Признак начала строки с выражения вида 1.1.1 (произвольный уровень вложенности, вместо цифр могут быть буквы)	1519
Жирность шрифта (менее жирный шрифт)	811
Индикатор, заканчивается ли предыдущая строка буквой	611
Число букв в начале строки	361
Тире в начале строки	359

С использованием данных о важности признаков, в признаковое пространство были добавлены новые признаки. Например, вместо одного признака, отвечающего за жирность шрифта, была добавлена целая группа признаков, отвечающая за различные уровни жирности шрифта. Для самых важных признаков к вектору признаков каждой строки документа были добавлены усреднённые значения данных признаков по документу. В табл. 2 представлен итоговый список признаков после добавления и значения их важности.

5.3 Результаты

После настройки параметров XGBClassifier способом, описанным в [19] на валидационном датасете (*learning_rate=0.1, n_estimators=1000, max_depth=7, min_child_weight=2, gamma=0, subsample=1, colsample_bytree=1, alpha=0.01*) итоговый F1-score, полученный в результате кросс-валидации (разбиение данных на 3 части) оказался равным 0.98995.

5.4 Анализ ошибок

На рис. 3 показана матрица ошибок для полученного классификатора. Матрица получена на валидационном датасете. По вертикальной оси расположены правильные классы, по горизонтальной - классы, которые предсказал классификатор. В клетках на пересечении расположены значения количества строк, у которых совпали данные классы.

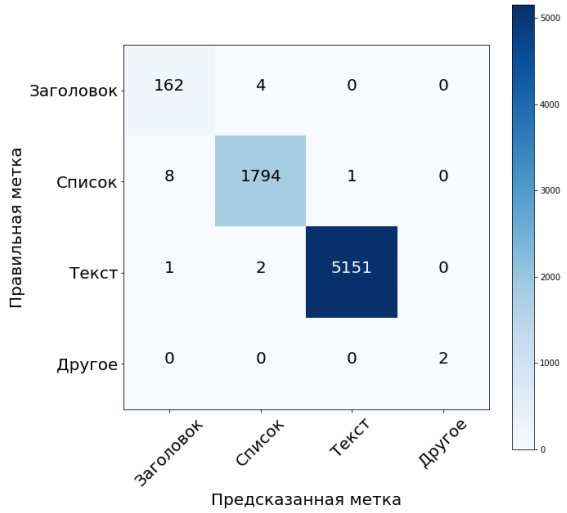


Рис. 3. Матрица ошибок без нормализации
Fig. 3. Confusion matrix without normalization

На рисунке видно, что 8 строк, относящихся к типу «Список», были проклассифицированы как «Заголовок», 4 строки, напротив, вместо метки «Заголовок» получили метку «Список». Аналогичную статистику можно посмотреть и для других пар классов.

Таким образом, больше всего классификатор путает классы «Заголовок» и «Список». Это можно объяснить тем, что некоторые признаки данных классов очень похожи, например, многие заголовки начинаются с нумерации, а элементы списков имеют большой отступ от левого края страницы.

6. Заключение

В данной статье разработан метод выделения логической структуры документа, основанный на классификации строк документа, определяющий в документах заголовки и списки разных уровней вложенности. Реализован пайплайн, состоящий из обработки документов с помощью программы Tesseract [17] и извлечения текста и рамок строк, выделения векторов признаков и обучения классификатора. Кроме того, с помощью ручной разметки получен размеченный датасет, доступный для изучения [1]. Для эффективной классификации списков и заголовков помогает извлечение признаков, указанных в табл. 2 и использование XGBClassifier [20]. Итоговый F1-score на данных кросс-валидации, полученный при настройке параметров классификатора, равен 0.98995. Однако, в силу особенностей датасета классификатор может путать заголовки и элементы списков.

Дальнейшие исследования могут быть направлены на выделение более подробной структуры. Помимо классификации каждой строки можно определять её уровень вложенности по отношению к документу.

Список литературы / References

[1]. А. О. Богатенкова. Dataset, ИСП РАН. URL: `https://github.com/NastyBoget/document_structure_extraction` (дата обращения 27.05.2020).

[2]. E. Giguet and G. Lejeune. Daniel@ fintoc-2019 shared task: toc extraction and title detection. In *Proceedings of the Second Financial Narrative Processing Workshop (FNP 2019)*, pages 63–68, 2019.

[3]. K. Tian and Z. Peng. Finance document extraction using data augmentation and attention. In *Proceedings of the Second Financial Narrative Processing Workshop (FNP 2019), September 30, Turku Finland*, number 165, pages 1–4. Linköping University Electronic Press, 2019.

[4]. M. M. Rahman and T. Finin. Deep understanding of a document's structure. In *Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies*, pages 63–73, 2017.

[5]. A. Doucet, G. Kazai, S. Colutto, and G. Mühlberger. Icdar 2013 competition on book structure extraction. In *12th International Conference on Document Analysis and Recognition*, pages 1438–1443. IEEE, 2013.

[6]. L. Gao, X. Yi, Z. Jiang, L. Hao, and Z. Tang. Icdar2017 competition on page object detection. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1417–1422. IEEE, 2017.

[7]. C. Clausner, A. Antonacopoulos, and S. Pletschacher. Icdar2017 competition on recognition of documents with complex layouts-rdcl2017. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1404–1410. IEEE, 2017.

[8]. R. Juge, I. Bentabet, and S. Ferradans. The fintoc-2019 shared task: financial document structure extraction. In *Proceedings of the Second Financial Narrative Processing Workshop (FNP 2019)*, pages 51–57, 2019.

[9]. Единая информационная система в сфере закупок, ЕИС. URL: `https://zakupki.gov.ru/` (дата обращения 27.05.2020).

[10]. J. Pustejovsky and A. Stubbs. *Natural Language Annotation for Machine Learning: A guide to corpus-building for applications*. " O'Reilly Media, Inc.", 2012, pages 105–139.

[11]. А. И. Перминов. Paragraph labeler application, ИСП РАН. URL: `https://github.com/dronperminov/ParagraphLabelerApp` (дата обращения 10.06.2020).

[12]. R. Artstein and M. Poesio. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596, 2008.

[13]. P. S. Bayerl and K. I. Paul. What determines inter-coder agreement in manual annotations? a meta-analytic investigation. *Computational Linguistics*, 37(4):699–725, 2011.

[14]. Р. А. Гилязев и Д. Ю. Турдаков. Активное обучение и краудсорсинг: обзор методов оптимизации разметки данных. *Труды Института системного программирования РАН*, 30(2), 2018.

[15]. J. Liang, J. Piper, and J.-Y. Tang. Erosion and dilation of binary images by arbitrary structuring elements using interval coding. *Pattern Recognition Letters*, 9(3):201–209, 1989.

[16]. Opencv, Intel Corporation. URL: `https://opencv.org` (visited on 05/27/2020).

[17]. R. Smith. An overview of the tesseract ocr engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE, 2007.

[18]. B. Kostenko. XGBoost feature interactions reshaped. URL: `https://github.com/limexp/xgbfir` (visited on 05/27/2020).

[19]. A. Jain. Complete guide to parameter tuning in xgboost with codes in python. URL: `https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python` (visited on 05/27/2020).

[20]. XGBoost documentation, The XGBoost Contributors. URL: `https://xgboost.readthedocs.io/en/latest/index.html` (visited on 05/27/2020).

Приложение / Appendix

```
[
    {
    "name": name of the first document,
    "width": image width (pixels),
    "height": image height,
    "entities": [
        {
            "label": first line label,
            # bounding box for the first line
            "x": first line left indent,
            "y": first line top indent,
            "width": first line width,
            "height": first line height,
            "text": first line text
        }, ...
    }, ...
]
```

Рис. 4. Результат разметки
Fig. 4. Labeling result

Манифест

Выделяются следующие типы строк: заголовок, элемент списка, текст.

1. Заголовок – это название главы, секции, подглавы, параграфа. Строка помечается заголовком, если:
    - текст визуально (полностью) выделяется жирностью (рис. 5);
    - текст полностью выделяется шрифтом (курсив, подчеркнутый, другой шрифт, другой размер шрифта) (рис. 6);
      при этом если текст строки выделен шрифтом частично, то заголовком

Рис. 5. Пример заголовка №1
Fig. 5. Header example №1

Рис. 6. Пример заголовка №2
Fig. 6. Header example №2

это не считается;

- текст выделяется отступом (расположен по центру) (рис. 7);

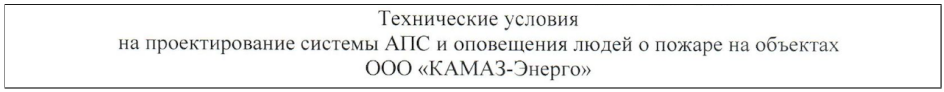


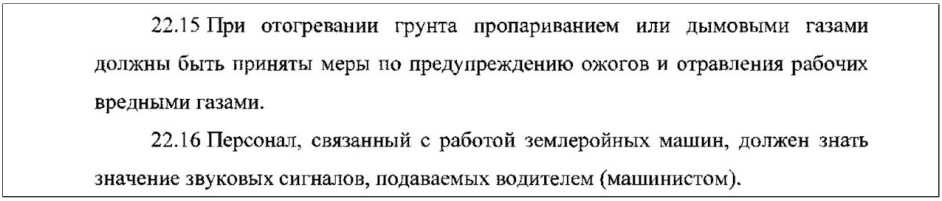
Рис. 7. Пример заголовка №3
Fig. 7. Header example №3

- если заголовок занимает несколько строк, остальные строки тоже относятся к типу «заголовок».

2. Элемент списка - это начало нумерованного или маркированного списка. Строка помечается как элемент списка, если:
   - строка наряду с несколькими другими строками пронумерована («1. 1) а) 1.1» и т. д. в начале строки) или выделена некоторым маркером (точка, тире и т. д.);
   - если элемент списка визуально занимает несколько строк, все строки кроме первой помечаются как текст. Также к списку не относятся строки, помеченные как заголовок (выделенные шрифтом, жирностью и т. д.). На рис. 8 как элемент списка будут помечены только две строки, остальные помечаются как текст.
3. Текстовые строки – это все остальные строки, содержащие текст документа.
4. Other – при разметке могут попадаться выделенные области, не содержащие текста, такие области помечаются как «Other».

Информация об авторах / Information about authors

Анастасия Олеговна БОГАТЕНКОВА является бакалавром кафедры Системного


22.15 При отогревании грунта пропариванием или дымовыми газами должны быть приняты меры по предупреждению ожогов и отравления рабочих вредными газами.

22.16 Персонал, связанный с работой землеройных машин, должен знать значение звуковых сигналов, подаваемых водителем (машинистом).

Рис. 8. Пример элементов списка
Fig. 8. List example

программирования Московского государственного университета имени М.В. Ломоносова. Научные интересы: распознавание структуры документов, цифровая обработка изображений.

Anastasiya Olegovna BOGATENKOVA is a bachelor of the Department of system programming of CMC of Lomonosov Moscow State University. Research interests: document layout analysis, digital image processing.

Илья Сергеевич КОЗЛОВ является стажером-исследователем Института Системного Программирования РАН. Научные интересы: распознавание структуры документов, цифровая обработка изображений, нейросетевая обработка данных, распознавание образов.

Ilya Sergeevich KOZLOV — researcher at ISP RAN. Research interests: document layout analysis, digital image processing, neural network data processing, image pattern recognition.

Оксана Владимировна БЕЛЯЕВА является стажером-исследователем Института Системного Программирования РАН. Научные интересы: распознавание структуры документов, цифровая обработка изображений, нейросетевая обработка данных, распознавание образов.

Oksana Vladimirovna BELYAEVA — researcher at ISP RAN. Research interests: document layout analysis, digital image processing, neural network data processing, image pattern recognition.

Андрей Игоревич ПЕРМИНОВ является магистром кафедры Системного программирования Московского государственного университета имени М.В. Ломоносова. Научные интересы: цифровая обработка сигналов, нейросетевая обработка данных, создание искусственных данных.

Andrey Igorevich PERMINOV is a master of the Department of system programming of CMC of Lomonosov Moscow State University. Research interests: digital signal processing, neural network data processing, generation of artificial data.