DOI: 10.15514/ISPRAS-2000-1(2)-33

Logical structure extraction from scanned documents

[1,2]*Bogatenkova A. O. <nastyboget@ispras.ru>*
[1]*Kozlov I. S. <kozlov-ilya@ispras.ru>*
[1]*Belyaeva O. V. <belyaeva@ispras.ru>*
[1,2]*Perminov A. I. <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. There are many scanned documents without a text layer. Understanding the information in such documents may be useful for their analysis, e. g. for the effective search within documents. This paper describes pipeline for scanned documents processing. The method based on multiclass classification including headers and lists classes. Text and bounding boxes for lines were extracted using OCR methods, different features were generated for each line, which were 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, 2000, vol. 1, issue 2, pp. 3–4. 10.15514/ISPRAS-2000-1(2)-33

Извлечение логической структуры из сканированных документов

[1,2]*Богатенкова А. О. <nastyboget@ispras.ru>*
[1]*Козлов И. С. <kozlov-ilya@ispras.ru>*
[1]*Беляева О. В. <belyaeva@ispras.ru>*
[1,2]*Перминов А. И. <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. Зачастую требуется осуществлять быстрый поиск по их содержимому. Знание структуры документов может способствовать более эффективному их анализу. В статье описан процесс построения пайплайна обработки сканированных документов. 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, 2000, том 1 вып. 2, с. 3–4. 10.15514/ISPRAS-2000-1(2)-33

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 может быть 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 OCR.

Во-вторых, зачастую логические части документа выделяются с помощью форма-

тирования: 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 по-разному. Так, 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. Поэтому для выделения логической структуры желательно использовать 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 таких признаков может быть необходима метаинформация, такая как размер и тип шрифта, отступы, междустрочные интервалы и т. д. Поэтому извлечение логической структуры логично делать на этапе анализа сканированных документов.

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 документов проводится очень много соревнований ICDAR, например [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 документов, а так же 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 использова-

ла 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. Процесс аугментации показан на рис. 1.

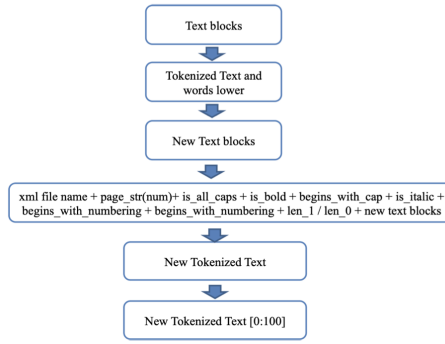

Рис. 1. Аугментация данных для LSTM и BiLSTM
Fig. 1. Data augmentation for LSTM and BiLSTM

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] 2017 года 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).

Классификатор (рис. 2), который был использован при решении задачи, состоит из нескольких частей. Сначала строки документа подаются на вход классификатору (классификатор строк), который определяет, является ли строка заголовоком, 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.

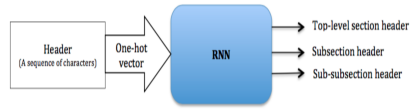

Рис. 2. Вход и выход классификатора заголовков
Fig. 2. Input and output of the section classifier

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.

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 изображения и таблицы.

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.

Первоначально набор данных состоял из 5000 изображений, однако на большей части этих изображений присутствовали таблицы, рисунки, рамки и прочие нетекстовые элементы, которые не рассматриваются в данной работе. Поэтому в итоговый набор вошли только 600 изображений, удовлетворяющие необходимым требованиям.

3.2 Манифест

Манифест - это инструкция для разметки данных, в которой указывается цель разметки, что и по какому принципу нужно размечать. Он нужен для формализации задачи машинного обучения и для объяснения людям, которые будут размечать данные, какая разметка считается правильной. Как правило, в манифесте описывается, что будет поступать на вход аннотатору и что должно получиться на выходе, при этом объяснение должно сопровождаться множеством примеров.

Текст манифеста для нашей задачи можно посмотреть в приложении.

3.3 Разметка данных

Документы были размечены с использованием специальной системы, разработанной в ИСП РАН [10]. Процес разметки проходил следующим образом: в каждом документе последовательно обводились в рамки текстовые строки, которым аннотатор присваивал метку одного из классов. Данные рамки и текст каждой строки распознавались с помощью программы Tesseract [11]. Результатом разметки стал набор JSON файлов, каждый из которых содержит название документа, размеры изображения и список текстовых строк. Для каждой строки указаны координаты заключающей её рамки (bounding box) и метка принадлежности строки к одному из описанных классов.

3.4 Вычисление уровня согласия между аннотаторами

Для проверки правильности разметки была посчитана специальная статистика Cohen's kappa. После разметки десяти документов (407 строк) двумя аннотаторами значение статистики κ оказалось равным 0.975 (чем ближе значение κ к единице, тем больший уровень согласия достигнут между аннотаторами), 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).

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 классам.

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:
  - 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).

Кроме того, к признакам, перечисленным выше, для каждой строки были добавлены аналогичные признаки четырех предыдущих и следующих строк. Это нужно для анализа продолжения блока строк конкретного типа.

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

И, наконец, для каждого документа вычислялся средний отступ от левого края страницы, средняя высота шрифта, средняя длина строки, среднее число слов в строках, среднее значение дли жирности шрифта (ядро свертки 5), среднее число букв в первом слове каждой строки. Данные значения добавлялись 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 из библиотеки OpenCV. Они предполагают применение свертки к изображению с некоторым ядром. Среди пикселей изображения, которые перекрываются ядром, в случае операции dilate вычисляется максимальное значение цвета пикселей, а в случае erode - минимальное.

При работе с текстом в качестве исходного изображения используется bounding box конкретной строки. При применении к изображению операции dilate увеличивается размер светлых областей, поэтому шрифт становится менее жирным (а может и вовсе пропасть). При применении операции erode наоборот увеличивается размер тёмных областей (в нашем случае шрифт становится более жирным). Таким образом, применяя последовательно сначала операцию dilate, а затем erode, можно добиться исчезновения текста, написанного нежирным шрифтом, более жирный шрифт останется практически без изменений. Далее можно вычислить средний цвет измененного изображения (bounding box-а), значение которого является искомым признаком.

В зависимости от размера ядра свертки можно варьировать размер тех регионов изображения, на которых вычисляется максимум или минимум, а значит можно определять жирность шрифта различных уровней. В данной работе размер ядра свертки для функций erode и dilate варьировался от 2 до 7 включительно.

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);
- экстра-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. 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, наилучший результат показал XGBClassifier, поэтому было решено выбрать его.

5.2 Анализ значимости признаков

Анализ значимости признаков был проведен с помощью библиотеки xgbfir [12] . В табл. 2 представлены первые 10 признаков с наивысшей значимостью (information gain). Это признаки, которые имеют наибольший вес при вычислении предсказания классификатора.

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 и значения их важности.

5.3 Результаты

После настройки параметров XGBClassifier [13] ($learning\_rate = 0.1$, $n\_estimators = 1000$, $max\_depth = 7$, $min\_child\_weight = 2$, $gamma = 0$, $subsample = 1$, $colsample\_bytree = 1$, $alpha = 0.01$) итоговый F1-score, полученный в результате кросс-валидации (разбиение данных на 3 части) оказался равным

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	1157
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

0.98995.

5.4 Анализ ошибок

На рис. 3 показана матрица ошибок для полученного классификатора. 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.

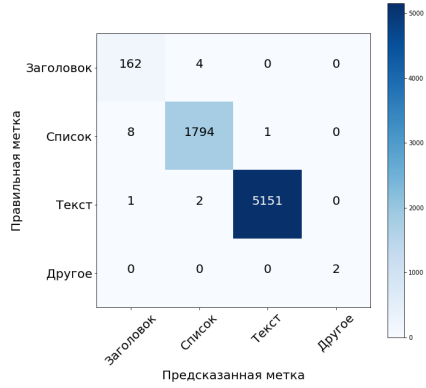

Рис. 3. Матрица ошибок без нормализации
Fig. 3. Confusion matrix without normalization

На рисунке видно, что 8 строк, относящихся к типу «Список», были проклассифи-

цированы 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. Выводы

В данной статье реализован метод выделения логической структуры документа, основанный на классификации строк документа, определяющий в документах заголовки и списки разных уровней вложенности.

Был обработан с помощью программы Tesseract и размечен набор документов, используемый в качестве тренировочных данных. Для эффективной классификации списков и заголовков может помочь извлечение признаков, указанных в табл. 2 и использование XGBClassifier. Итоговый F1-score на 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/tree/master/docs` (дата обращения 27.05.2020). / A. O. Bogatenkova. Dataset, ISP RAS. URL: `https://github.com/NastyBoget/document_structure_extraction/tree/master/docs` (visited on 05/27/2020).

[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]. А. И. Перминов. Paragraph labeler application, ИСП РАН. URL: `https://github.com/dronperminov/ParagraphLabelerApp` (дата обращения 10.06.2020). / A. I. Perminov. Paragraph labeler application, ISP RAS. URL: `https://github.com/dronperminov/ParagraphLabelerApp` (visited on 06/10/2020).

[11]. 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.

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

[13]. 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).

Приложение / Appendix

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);

```
[
    {
    "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

15.3 Проведение временных огневых работ

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

- текст полностью выделяется шрифтом (курсив, подчеркнутый, другой шрифт, другой размер шрифта) (рис. 6);

- *Привести в текстовой части*

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

при этом если текст строки выделен шрифтом частично, то заголовком это не считается;

- текст выделяется отступом (расположен по центру) (рис. 7);
- если заголовок занимает несколько строк, остальные строки тоже относятся к типу «заголовок».

2. Элемент списка - это начало нумерованного или маркированного списка.

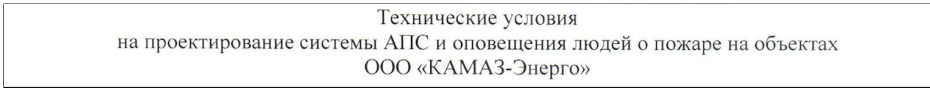

Рис. 7. Пример заголовка №3
Fig. 7. Header example №3

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.

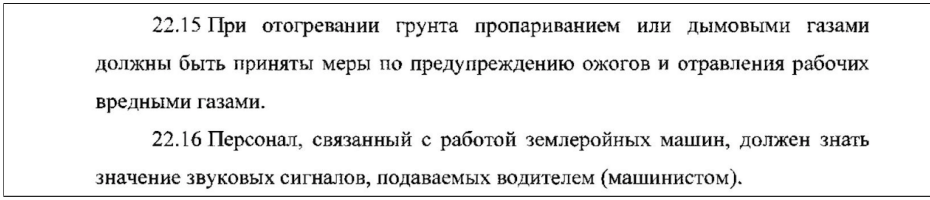

Рис. 8. Пример элементов списка
Fig. 8. List example

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».