Извлечение логической структуры из сканированных документов

А.О. Богатенкова

Институт системного программирования им. В.П. Иванникова
Российской академии наук,
Москва, Россия
bogatenkova.anastasiya@mail.ru

***Аннотация*— Большое количество текстовой информации представлено в виде pdf-документов, у которых может отсутствовать текстовый слой. Зачастую требуется осуществлять быстрый поиск по их содержимому. Знание структуры документов может способствовать более эффективному их анализу. В статье рассмотрены существующие решения задачи извлечения структуры документа в виде иерархии заголовков, а также предложен другой метод решения данной задачи, позволяющий отличать заголовки и элементы списков от остальных строк документа. Кроме того, размечен 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**

I. ВВЕДЕНИЕ

Как правило, документы имеют логическую структуру и содержат название, разбиение на главы, подглавы и т. д., нумерованные и маркированные списки. Выделение такой структуры документа может помочь при решении задач автоматизированного анализа документов, а также при поиске по документам.

Применяется множество разнообразных подходов [5], [6], [7], которые позволяют выделять в тексте заголовки и распознавать логическую структуру документов. Однако данные подходы подразумевают работу с ограниченным количеством уровней вложенности и не принимают во внимание элементы списков.

Ограничим класс рассматриваемых документов сканированными документами без текстового слоя. Будем считать, что в документах не содержатся изображения и таблицы. 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.

Статья состоит из следующих частей: глава 2 содержит обзор различных подходов, с помощью которых решается задача выделения структуры документа; в главе 3 описывается датасет, используемый при реализации и проверке метода; в главе 4 рассматривается реализованный метод; в главе 5 показаны результаты экспериментальной проверки метода, а в главе 6 представлены краткие выводы и предлагаются возможности для дальнейшей работы и исследований.

II. ОБЗОР АНАЛОГИЧНЫХ РАБОТ

По извлечению структуры из документов существуют несколько подходов:

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

A. Извлечение структуры из документов на основе оглавления и правил

По анализу документов проводится очень много соревнований на ICDAR, например [1], [2], [3]. В одном из таких соревнований [1] 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 [4], где из финансовых документов извлекалась структура в виде иерархии уровней заголовков документов. Максимальная глубина уровней равна пяти. Одна из команд-участниц [5] 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.

B. Извлечение структуры из документов на основе машинного обучения

В соревнованиях [4] 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 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 [6] 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.

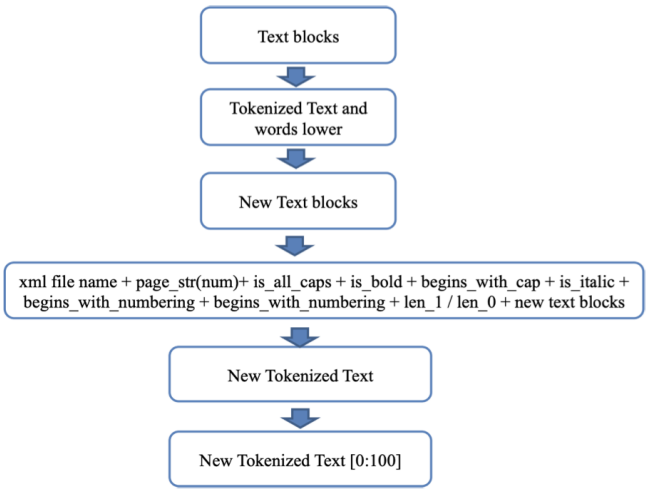

Рис. 1. Аугментация данных для LSTM и BiLSTM

В статье [7] 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

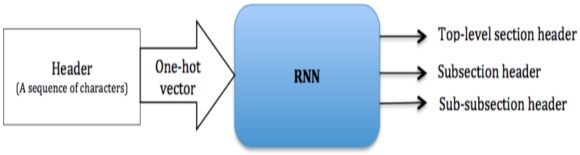

Рис. 2. Вход и выход классификатора заголовков

(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. Метрика качества - F1-мера, при идентификации заголовков итоговый score – 0,96; при классификации секций средний F1-score – 0,81.

III. ДАТАСЕТ

Датасет представляет собой набор документов в виде изображений в формате .jpeg, скачанный из интернета. Документы являются сканированными копиями текстов договоров различных предприятий (СТО, РД ЭО и др.). Каждое изображение будем считать отдельным документом.

Данный корпус имеет ряд специфических особенностей. Документы не содержат таблиц и рисунков, текст расположен строго по-горизонтали, 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.

Документы были размечены с использованием системы для разметки, разработанной в ИСП РАН [8]. Текст документов распознавался с помощью программы Tesseract [9]. Результатом разметки стал набор .json файлов, каждый из которых содержит название документа и список текстовых строк. Для каждой строки указаны координаты заключающей её рамки (bounding box) и правильная метка.

Для проверки правильности разметки была посчитана специальная статистика Cohen's kappa. После разметки десяти документов 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 и отдельные .json файлы были объединены в один.

IV. ОПИСАНИЕ РЕШЕНИЯ

A. 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.

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, среднее число букв в первом слове каждой строки. Данные значения добавлялись к признакам каждой строки документа.

B. Подбор классификатора

При решении задачи было опробовано множество классификаторов, 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).

Множество документов тремя способами было разбито на тренировочное и тестовое множества (разбиение по документам), на каждом разбиении было проведено обучение классификаторов и вычисление F1-score. Усредненные значения F1-score для каждого классификатора указаны в таблице 1.

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

ТАБЛИЦА I
Сравнение классификаторов

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

C. Анализ значимости признаков

В таблице 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 таблице 2 представлен итоговый список признаков и значения их важности.

V. РЕЗУЛЬТАТЫ

После настройки параметров XGBClassifier ($learning\_rate$, $n\_estimators$, max_depth, min_child_weight, $gamma$, $subsample$, $colsample\_bytree$, $alpha$) итоговый F1-score, полученный в результате кросс-валидации оказался равным 0.98995.

На рис. 3 показана матрица ошибок для полученного классификатора. По вертикальной оси расположены

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

ТАБЛИЦА II

Значимость признаков

правильные классы, по горизонтальной - классы, которые предсказал классификатор. В клетках на пересечении расположены значения количества строк, у которых совпали данные классы.

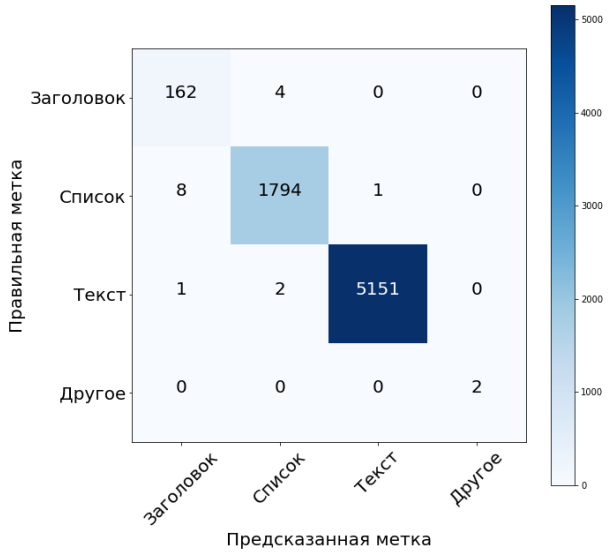

Рис. 3. Матрица ошибок без нормализации

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.

VI. 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 размечен копус документов, используемый в качестве тренировочных данных, выявлены признаки, наиболее подходящие для классификации строк документа. Для классификации использован XGBClassifier, для которого были подобраны оптимальные параметры. Итоговый F1-score, полученный при настройке параметров, равен 0.98995. В силу особенностей датасета классификатор чаще всего путает 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

[1] Doucet A. et al. ICDAR 2013 competition on book structure extraction //2013 12th International Conference on Document Analysis and Recognition. – IEEE, 2013. – С. 1438-1443.

[2] Gao L. et al. ICDAR2017 competition on page object detection //2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). – IEEE, 2017. – Т. 1. – С. 1417-1422.

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

[4] Juge R., Bentabet I., Ferradans S. The fintoc-2019 shared task: Financial document structure extraction //Proceedings of the Second Financial Narrative Processing Workshop (FNP 2019). – 2019. – С. 51-57.

[5] Giguet E., Lejeune G. Daniel@ FinTOC-2019 Shared Task: TOC Extraction and Title Detection //Proceedings of the Second Financial Narrative Processing Workshop (FNP 2019). – 2019. – С. 63-68.

[6] Tian K., Peng Z. Finance document Extraction Using Data Augmentation and Attention //Proceedings of the Second Financial Narrative Processing Workshop (FNP 2019), September 30, Turku Finland. – Linköping University Electronic Press, 2019. – №. 165. – С. 1-4.

[7] Rahman M. M., Finin T. Deep Understanding of a Document's Structure //Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies. – 2017. – С. 63-73.

[8] ParagraphLabelerApp

[9] Smith R. An overview of the Tesseract OCR engine //Ninth International Conference on Document Analysis and Recognition (ICDAR 2007). – IEEE, 2007. – Т. 2. – С. 629-633.

[10] Complete Guide to Parameter Tuning in XGBoost with codes in Python