

Извлечение иерархической логической структуры из текстовых документов в формате docx

Автор: Богатенкова Анастасия Олеговна

Научный руководитель: Гомзин Андрей Геннадьевич

Научный консультант: Козлов Илья Сергеевич

28 апреля 2021 г.

ИСП **РАН**

- В мире каждый год создаётся большое количество документов.
- Для автоматической обработки (например, поиска по документам, суммаризации и т. д.) необходимо представить документ в удобном для обработки виде, т. е. выделить его структуру.
- Для этих целей в ИСП РАН разрабатывается программный модуль docreader, основанный на проекте dedoc¹.

¹<https://github.com/ispras/dedoc>

Для документов можно определить три основных типа структуры:

1. **Физическая структура** – внешний вид документа (**жирность** шрифта, две колонки).
2. **Логическая структура** описывает разбиение документа на компоненты (например, законы состоят из глав, главы разбиваются на статьи и т. д.).
3. **Семантическая структура** связана с задачей понимания содержимого текста.

В данной работе рассматривается логическая структура документа.

Компоненты логической структуры зависят от домена. Например, научные статьи состоят из аннотации, введения, обзора существующих работ и других секций.

Acknowledgments	iii
Abstract	v
Résumé	vii
1 Introduction	1
1.1 Context	1
1.2 Contribution	3
1.3 Thesis Structure	4
2 Document Production and Understanding	7
2.1 Printed Documents	7
2.2 Document Production	9
2.2.1 Logical Document	9
2.2.2 Physical Document	11
2.2.3 Rendered and Paper Documents	12
2.3 Document Understanding	12
2.3.1 Image Preprocessing	13
2.3.2 Physical Structure Recognition	14
2.3.3 Logical Structure Recognition	14
2.4 Document Models	15

- Анализируются документы с требованиями к проектам – технические задания.
- Технические задания имеют определённую логическую структуру и описываются ГОСТами², однако на практике от них часто отходят.
- Для автоматической обработки технических заданий необходимо выделять из них логическую структуру, которую нужно формализовать.

²http://technicaldocs.ru/гост19/шаблоны/техническое_задание

- Большое количество технических заданий создаётся и хранится в формате docx.
- Формат docx в основном ориентирован на отображение физической структуры документа.
- Библиотеки по работе с форматом существуют³, однако не позволяют выделить в документах все стили и текст нумерации элементов списков.

³Рассматривается язык программирования python

- Формально описать извлекаемую иерархическую логическую структуру;
- Реализовать метод построения иерархической логической структуры из технических заданий в формате docx;
- Провести оценку качества реализованного метода;
- Реализовать извлечение текста и необходимых метаданных из документов в формате docx. Встроить реализованный метод в открытый проект по обработке документов dedoc⁴.

⁴<https://github.com/ispras/dedoc>

По итогам изучения работ лингвистов и анализа примеров технических заданий, предложена структура ТЗ следующего вида: дерево произвольной глубины, в котором каждому узлу соответствует параграф документа с типом (заголовок, содержание, часть, элемент списка или текст).



Ставится задача построения дерева с типизированными узлами. Задача построения дерева требует решения следующих подзадач:

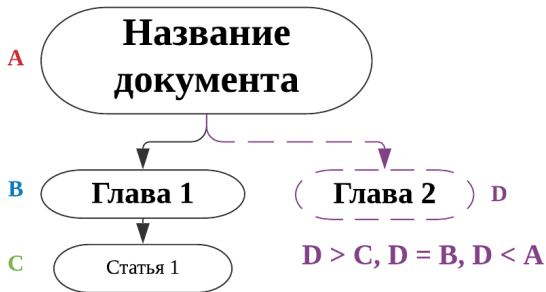
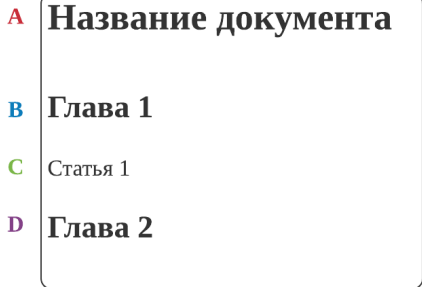
Определение типа параграфов

Каждому параграфу сопоставляется один из 5 классов: *заголовок*, *содержание*, *часть*, *элемент списка* или *текст*.

Построение дерева

Определение вложенности параграфов на основе попарного сравнения.

- Добавляем параграф «Глава 2» в дерево.
- Сравниваем параграфы попарно для определения места вставки.



Название документа (0 уровень)

Глава 1 (1 уровень)

Содержимое главы 1 (2 уровень)

Статья 1 (2 уровень)

Содержимое статьи 1 (3 уровень)

- Первый элемент маркированного списка (4 уровень)
- Второй элемент маркированного списка (4 уровень)

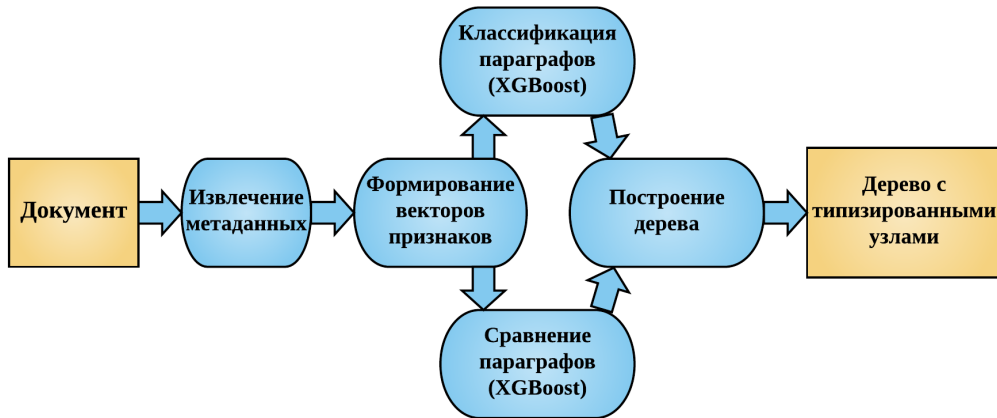
Статья 2 (2 уровень)

Содержимое статьи 2 (3 уровень)

Глава 2 (1 уровень)

1. Первый элемент нумерованного списка (2 уровень)
2. Второй элемент нумерованного списка (2 уровень)





Классификация параграфов

- Классификация параграфов на 5 классов: заголовок, содержание, часть, элемент списка или текст.
- Формирование векторов признаков для параграфов, обучение классификатора (XGBoost) и корректировка ответов классификатора.

Сравнение параграфов

- Классификация **пар** параграфов документа на 3 класса: больше, меньше или равно.
- Формирование векторов признаков для **пар** параграфов и обучение классификатора пар параграфов (XGBoost).

Задача	Количество признаков	Алгоритм	Достоверность (accuracy)	F-мера
Классификация параграфов	234	XGBoost	0.96	0.94
Существующий метод классификации параграфов	8	RandomForest	0.91	0.86
Сравнение параграфов	28	XGBoost	0.98	0.97

- Реализовано построение деревьев документов типа «Техническое задание».
- На тестовой выборке из 5 документов усреднённое расстояние Робинсона-Фулдса между построенными и размеченными деревьями оказалось равным 0.073.

```
Root
├── title: ТЕХНИЧЕСКОЕ ЗАДАНИЕ
│   ├── item: 1. Кормление альпаки
│   │   ├── item: - Качество корма должно быть хорошим
│   │   └── item: - Количество корма должно быть достаточным
│   ├── item: 2. Уход за альпаками
│   │   ├── item: - Альпаку надо пасти на высоте около 3500 метров над уровнем моря.
│   │   └── item: - Альпаку надо стричь
│   └── title: На разведение и уход за альпаками. Принято министерством по делам альпаководства.
└── toc: СОДЕРЖАНИЕ
    ├── toc: 1 КОРМЛЕНИЕ АЛЬПАКА 1
    └── toc: 2 УХОД ЗА АЛЬПАКАМИ 2
```

Данные для сравнения пар параграфов

Для каждого параграфа рассматриваются 4 нижележащих параграфа. Каждой паре нужно сопоставить один из трех классов: больше, меньше или равно.

- 22 документа, скачанных с сайта государственных закупок⁵;
- 1405 параграфов;
- 4438 пар параграфов.

⁵<https://zakupki.gov.ru>

На языке python написан
обработчик docx документов.
Код обработчика встроен в
проект dedoc. Пример
извлечения метаданных
(жирность, размер шрифта,
выравнивание и т. д.) для
параграфа документа.

```
"text": "Header 1",  
"annotations": [  
  {  
    "start": 0,  
    "end": 8,  
    "name": "indentation",  
    "value": "0"  
  },  
  {  
    "start": 0,  
    "end": 8,  
    "name": "alignment",  
    "value": "left"  
  },  
]
```

- Исследована структура технических заданий, предложена структура в виде дерева произвольной глубины с типизированными узлами.
- Разработан и реализован метод по извлечению предложенной структуры из технических заданий, представленных в формате docx.
- Составлен и размечен набор технических заданий.
- Проведено сравнение с адаптацией существующего метода к поставленной задаче.
- Реализовано извлечение текста и метаданных из документов в формате docx. Код обработчика docx документов встроен в проект dedoc.

Процесс создания обучающего набора документов:

1. Создание изображений для разметки из docx файла. На каждом из изображений один из параграфов обведён в рамку.
2. Разметка каждого изображения аннотаторами с помощью системы разметки⁶.
3. Получение меток для параграфов, соответствующих размеченным изображениям.

⁶<https://github.com/dronperminov/ImageClassifier>

Страницы docx документа преобразуются в изображения с параграфами, обведенными в рамку.

Header 1

Header 3

Header 2

Header 2

Header 1

Header 3

Simple text

1. Bullet list point 1
2. Bullet list point 2
3. Bullet list point 3
4. Bullet list point 4

Some simple text again

1. Numeric list point 1
2. Numeric list point 2
- 2.1. Numeric list point 3
- 2.2. Numeric list point 4
3. Numeric list point 5
4. Numeric list 2 point 1
- 4.1. Numeric list 2 point 2
5. Numeric list 2 point 3

Start of a little table

First row column 1	First row column 2	First row column 3
gdrrnhnjhfg	Vgh thyk krt h	
234	5fem grthy kr th	123
456/8	Dsvfmrk -567 70.678 gôkrnk devmrt	Fghjkh Dfghjk dfghnj
Test merged cells		Test
		Merged rows
		rows

End of a little table

- Признаки, полученные с помощью регулярных выражений для начала и конца строки (в том числе для списков).
- Номер параграфа, длина текста параграфа, число слов в тексте параграфа.
- Индикаторы того, если ли в параграфе жирный, курсивный, подчёркнутый текст, текст из заглавных букв.
- Выравнивание, отступ от левого края, размер шрифта текста.
- Признаки для названий стилей («heading», «title», «list item», «contents»).
- Признаки трех предыдущих и трех следующих параграфов.

- Разница значений размеров шрифта и отступов от левого края.
- Разница значений индикаторов жирности, курсива, подчеркивания.
- Разница выравниваний и типов параграфов (каждому типу поставлено в соответствие число по приоритетам типов).
- Разница индикаторов соответствия параграфов стилям.
- Индикатор начала списка после строки, заканчивающейся двоеточием.
- Разница индикаторов соответствия параграфов шаблонам конкретных типов списков.