Извлечение иерархической логической структуры из текстовых документов в формате docx

---

Автор: Богатенкова Анастасия Олеговна

Научный руководитель: Козлов Илья Сергеевич

12 апреля 2021 г.

ИСП РАН

Актуальность задачи

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, основаный на проекте dedoc[1].

[1] https://github.com/ispras/dedoc

Типы структуры документа

Для документов можно определить три основных типа структуры:

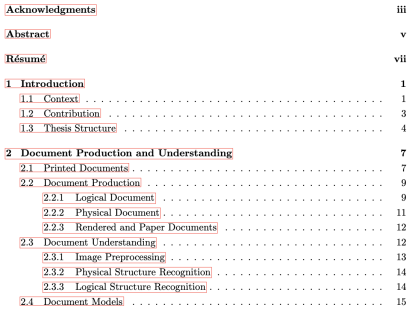
1. **Физическая структура** описывает то, как выглядит документ.
2. **Логическая структура** описывает разбиение документа на компоненты (например, законы состоят из глав, главы разбиваются на статьи и т. д.).
3. **Семантическая структура** связана с задачей понимания содержимого текста.

В данной работе рассматривается логическая структура документа.

Логическая структура документа

ИСП РАН

Логическая структура документа описывает разбиение документа на компоненты. Например, научные статьи состоят из аннотации, введения, обзора существующих работ и других секций.



Технические задания

ИСП РАН

- Распространённым типом документов являются документы с требованиями к проектам – технические задания.
- Технические задания имеют определённую логическую структуру.
- Точного определения структуры технического задания не было найдено.
- Определение возможной структуры и автоматическое извлечение этой структуры может быть использовано для упрощения работы с техническими заданиями.

Особенности docx

ИСП РАН

- 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 существуют[2], однако не позволяют выделить в документах все стили и текст нумерации элементов списков.

[2] Рассматривается язык программирования python

Постановка задачи

ИСП РАН

- 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 dedoc[3].

[3] https://github.com/ispras/dedoc

Структура технического задания

ИСП РАН

Предложена структура ТЗ следующего вида: дерево неограниченной глубины, в котором каждому листу соответствует параграф документа с типом (заголовок, содержание, часть, элемент списка или текст).

Построение дерева на основе сравнения пар строк

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)

Описание метода

Задача построения дерева требует решения следующих задач:

1. Разработка метода определения типа параграфов технического задания;
2. Разработка метода сравнения параграфов технического задания.

Классификатор параграфов

ИСП РАН

1. Решается задача классификации параграфов документа на 5 классов: заголовок, содержание, часть, элемент списка или текст.
2. Формирование вектора признаков на основе метаданных параграфов.
3. Обучение классификатора (XGBoost) и корректировка ответов классификатора.
4. Точность (accuracy) классификации на кросс-валидации 0.96.

Классификатор пар параграфов

ИСП РАН

1. Решается задача классификации пар параграфов документа на 3 класса: больше, меньше или равно.
2. Формирование вектора признаков на основе метаданных пар параграфов.
3. Обучение классификатора пар параграфов (XGBoost).
4. Точность (accuracy) классификации на кросс-валидации 0.98.

Построение дерева

ИСП РАН

- 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 дожно 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
```

Создание набора данных

Процесс создания обучающего набора документов:

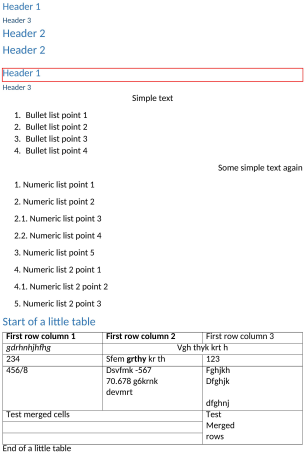
1. Создание изображений для разметки из docx файла. На каждом из изображений один из параграфов обведён в рамку.
2. Разметка каждого изображения аннотаторами с помощью системы разметки[4].
3. Получение меток для параграфов, соответствующих размеченным изображениям.

[4] https://github.com/dronperminov/ImageClassifier

Создание изображений из docx документов

ИСП РАН

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
gdrhnhjhfhg	Vgh thyk krt h	
234	Sfem **grthy** kr th	123
456/8	Dsvfmk -567 70.678 g6krnk devmrt	Fghjkh Dfghjk dfghnj
Test merged cells		Test Merged rows

End of a little table

Извлечение метаданных

ИСП РАН

На языке python написан обработчик docx документов. Код обработчика встроен в проект dedoc. Пример извлечения метаданных для параграфа документа


```
"node_id": "0.0",
"text": "Header 1",
"annotations": [
  {
    "start": 0,
    "end": 8,
    "name": "indentation",
    "value": "0"
  },
  {
    "start": 0,
    "end": 8,
    "name": "alignment",
    "value": "left"
  },
  {
    "start": 0,
    "end": 8,
    "name": "size",
    "value": "16.0"
  },
  {
    "start": 0,
    "end": 8,
    "name": "style",
    "value": "heading 1"
  }
],
```

Сравнение с существующим методом

- Не найдено методов, решающих такую же задачу.
- Есть метод классификации параграфов docx документов.
- Существующий метод не предназначался для выделения содержания в документах, поэтому при сравнении методов производилась классификация на 4 класса (заголовок, часть, список и текст).
- Существующий метод был реализован в соответствии с его описанием и изменен для классификации параграфов на 4 класса.
- Точность (accuracy) существующего метода на кросс-валидации – 0.91, предложенного метода 0.95.

Заключение

ИСП РАН

- Исследована структура технических заданий, предложена структура в виде дерева неограниченной 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.