# Извлечение метаданных из текстов нормативно-правовых актов на русском языке

ВМК МГУ

Октябрь 2020

«I always thought something was fundamentally wrong with the universe»

D. Adams. The Hitchhiker's Guide to the Galaxy

## 1 Введение

Каждый день органами власти Российской Федерации выпускается множество законов, указов, приказов, постановлений и других нормативно-правовых актов (НПА). Кроме собственно текста каждый НПА содержит также и метаданные: кем, когда и под каким номером он был выпущен, какое у него название, когда и кем он подписан и так далее. Законодательство устроено иерархично, поэтому каждый НПА обязательно содержит ссылки на вышестоящие по отношению к нему. Кроме того, могут быть и «горизонтальные ссылки» на НПА, регулирующие затронутую в каких-то конкретной статье или пункте тему.

Текст НПА с одной стороны достаточно формален, но с другой стороны написан на русском языке, с присущей ему вариативностью. Без качественных метаданных никакая база НПА невозможна: нельзя будет ни найти документ по его реквизитам, ни разобраться на какой документ проставлена ссылка. Поэтому задание практикума по обработке текстов в этом году будет связано с извлечением метаданных из текстов НПА.

## 2 Постановка задачи

Требуется реализовать алгоритм, принимающий на вход текстовый файл с НПА и возвращающий JSON объект, содержащий извлеченные метаданные.

Алгоритм должен поддерживать следующие типы метаданных:

1. тип документа;
2. номер документа;
3. дата принятия;
4. название документа;
5. орган, принявший акт.

Например, для текста с рисунка 1 необходимо построить JSON как на рисунке 2.

```
УКАЗ

ПРЕЗИДЕНТА РОССИЙСКОЙ ФЕДЕРАЦИИ

О заместителе Министра внутренних дел
Российской Федерации - начальнике Следственного
департамента Министерства внутренних дел
Российской Федерации

1. Назначить Алексеева Юрия Федоровича заместителем
Министра внутренних дел Российской Федерации - начальником
Следственного - департамента - Министерства - внутренних - дел
Российской Федерации.

2. Присвоить Алексееву Юрию Федоровичу специальное звание
генерал-майора юстиции.

3. Настоящий Указ вступает в силу со дня его подписания.

и
ж ЕЫ
э й

#ской Федерации - В.Путин

Москва, Кремль
14 июня 2012 года
№ 831

ЛАИ
```

Рис. 1: Указ Президента РФ № 831

```
{
    "type": "указ",
    "name": "О заместителе Министра внутренних дел
Российской Федерации - начальнике Следственного
департамента Министерства внутренних дел
Российской Федерации",
    "number": "831",
    "authority": "Президент Российской Федерации",
    "date": "14.06.2012"
}
```

Рис. 2: Метаданные указа Президента РФ № 831

Так как метаданные содержатся или на первой, или на последней странице, то для упрощения задачи текстовые файлы содержат не полные тексты НПА, а только их первые и последние страницы. Текстовые файлы получены из pdf-файлов без текстового слоя с помощью системы распознавания символов tesseract, поэтому тексты могут содержать ошибки распознавания.

Более подробно о каждом типе метаданных написано в секциях ниже.

## 2.1 Тип документа

Тип документа указывается в шапке документа (в верхней части документа).

В рамках задания документ может относиться к одному из следующих типов:

- «федеральный закон»,
- «постановление»,
- «приказ»,
- «распоряжение»,
- «закон»,
- «указ».

Обратите внимание, что в перечне присутствуют два похожих, но отдельных типа: «закон» и «федеральный закон».

Поле `type` должно содержать строку, обозначающую какой-то один из указанных типов, это задача классификации.

## 2.2 Номер документа

Номер документа часто указывается после символа № или N и может состоять из цифр, букв и спецсимволов. Примеры номеров:

- 67-ФЗ,
- 324-01-ЗМО,
- 824н.

Номер документа необходимо извлечь из текста «как есть», не подвергая дополнительной обработке.

Поле `number` должно содержать извлеченную строку. Чтобы скомпенсировать влияние ошибок распознавания символов в исходных текстах, при сравнении предсказанного и настоящего номера проверяющей системой регистр букв не учитывается.

## 2.3 Дата принятия

Термин «дата принятия» в рамках задачи используется не совсем юридически строго. Под датой принятия мы имеем в виду дату, которая используется, когда в ссылке на документ пишут: «во исполнение приказа № 123 от ...». Для законов это и будет дата принятия соответствующим законодательным органом, но для приказов или указов это будет дата подписания соответствующим уполномоченным лицом. У других типов НПА могут быть свои особенности.

Для некоторых типов документов вам могут встретиться сразу несколько дат. Например федеральные законы сначала принимаются Государственной Думой, потом одобряются Советом Федерации, а затем подписываются президентом. Извлекать нужно только дату принятия.

Поле `date` должно содержать дату принятия НПА, в указанном формате: `dd.mm.yyyy`.

## 2.4 Название документа

Название НПА – это краткий текст, описывающий основную суть и тему документа.

Название часто начинается с предлогов «о» или «об». В тексте НПА название может встречаться в кавычках или без кавычек. Название не включает в себя тип НПА.

Для примера на рисунке 1 название документа – это «О заместителе Министра внутренних дел Российской Федерации – начальнике Следственного департамента Министерства внутренних дел Российской Федерации».

Поле `name` должно содержать строку с названием в кавычках или без, в зависимости от того, как оно встретилось в тексте НПА. При сравнении предсказанного и настоящего номера в проверяющей системе не учитывается регистр букв, пробелы и символы перевода строки не различаются.

### 2.5 Орган, принявший акт

Часто автор документа указывается после слова «Принят», но далеко не всегда, например в указе на Рис. 1 автор - «Президент Российской Федерации».

В рамках задания необходимо для каждого документа определить орган, его принявший. Наименование органов необходимо извлекать в «нормализованном виде»:

- Ярославская областная дума
- Федеральная служба по надзору в сфере образования и науки
- Правительство Удмуртской Республики
- Департамент охраны окружающей среды и природопользования Ярославской области

В обучающей и тестовой выборке около 200 уникальных органов. В тестовой выборке есть органы, которых нет в обучающей.

Поле `authority` должно содержать наименование органа в нормализованном виде. При сравнении предсказанного и настоящего органа не учитывается регистр букв, пробелы и символы перевода строки не различаются.

## 3 Оценка качества

Качество выделения метаданных оценивается по отдельности по каждому элементу по разным метрикам.

1. **Тип документа.** Требуется предсказать один из шести классов. Метрика качества – макро f1-мера.
2. **Номер документа.** Требуется точное совпадение выделенного номера и указанного в разметке. Метрика качества – accuracy.
3. **Дата принятия.** Требуется точное совпадение выделенной и размеченной даты в формате `dd.mm.yyyy`. Метрика качества – accuracy.
4. **Название.** Допускается нечеткое совпадение. Метрика качества – средний коэффициент Жаккара (подробности ниже).
5. **Автор.** Допускается нечеткое совпадение. Метрика качества – средний коэффициент Жаккара (подробности ниже).

Метрика качества выделения имени документа (`name`) и автора (`authority`) L – это коэффициент Жаккара, вычисляется следующим образом:

$$L(s_{extr}, s_{gold}) = \frac{intersect}{offset_{start} + intersect + offset_{end}}, \quad (1)$$

где $intersect$ - длина самой длинной общей подстроки у выделенной строки и строки из «золотой метки», $offset_{start}$ - наибольшее количество символов до начала общей подстроки в выделенной строке и в строке из «золотой метки», $offset_{end}$ - наибольшее количество символов от конца общей подстроки до конца строки в выделенной строке и в строке из «золотой метки».

Эта метрика принимает значения от 0 до 1 и штрафует как за выделение более короткой строки чем строка из «золотой метки», так и за выделение более длинной строки.

## 4 Формат решения

Решения проверяются удаленно. Интерфейс автоматической системы находится по адресу: `https://2020.tpc.ispras.ru`. Решения должны быть написаны на языке Python (версия 3.6.9). Можно использовать все стандартные библиотеки, а также:

- scikit-learn==0.23.2 - алгоритмы машинного обучения
- pymorphy2[fast]==0.9.1 – библиотека для лемматизации

- numpy==1.18.5 – работа с многомерными массивами
- scipy==1.5.2 – работа с многомерными массивами
- nltk==3.5 - инструменты для обработки текстов
- tensorflow==2.3.1 – библиотеки для работы с искусственными нейронными сетями
- torch==1.6.0 – библиотеки для работы с искусственными нейронными сетями
- Keras==2.4.3 – библиотеки для работы с искусственными нейронными сетями
- pandas==1.1.3 – библиотека для работы с табличными данными
- pymystem3==0.2.0 – библиотека для лемматизации
- gensim==3.8.3 – библиотека для обработки текста

В случае необходимости использования дополнительных библиотек, сообщите об этом организаторам практикума (библиотеки будут добавлены для всех студентов). Доступ в Интернет на проверяющей машине закрыт.

## 4.1 Тестирование

На личной странице (`https://2020.tpc.ispras.ru/submissions/metadata`) находится статистика со всеми результатами в т.ч. результатами последнего тестирования (дата, метрики качества).

На странице `https://2020.tpc.ispras.ru/results` доступны результаты всех участников. Таблица обновляется раз в неделю.

Данные для обучения и локальной проверки доступны тут:
`http://tpc.at.ispras.ru/prakticheskoe-zadanie-2020/`.

## 4.2 Загрузка решения

Загружаемый файл должен представлять собой zip архив с любым именем. Архив должен обязательно содержать:

- Решение в файле `solution.py`. В файле должен содержаться класс Solution. В классе должны присутствовать методы:
  - `def train(self , train: List[Tuple[str, dict]]) -> None`. Метод берет на вход пары (текст, словарь из 5 меток: type, date, number, authority, name) и обучает на них модели. train – python представление тренировочных данных по ссылке.
  - `def predict(self , test: List[str]) -> List[dict]`. Метод предсказывает по списку текстов метки для каждой подзадачи (в виде словаря).

  Пример решения, возвращаюего пустые результаты для всех подзадач доступен на странице с заданием `http://tpc.at.ispras.ru/prakticheskoe-zadanie-2020/`.
- Описание применяемых методов в файле `description.txt`. Пожалуйста, напишите подробное описание, какие методы и признаки использовались. Это описание будет выложено вместе с решением после завершения курса.
- Все используемые ресурсы, необходимые для корректной работы метода.

## 4.3 Ограничения

- Каждую неделю можно послать не более 7 решений.

  **Внимание!** Итоговое тестирование будет проводиться на последнем загруженном решении.
- Размер загружаемого архива не должен превышать 15Мб.
- Время тестирования одного решение (обучение + предсказание) не должно превышать 30 минут.
- На проверяющей машине доступно 16Гб оперативной памяти.

В связи с первым ограничением, для тестирования на локальной машине рекомендуется использовать метод перекрестной проверки.

### 4.4 Baseline

Так как НПА написаны формальным языком, на основе методических рекомендаций, регулярные выражения могут показывать неплохие результаты в этой задаче. Поэтому в качестве baseline используется решение на основе правил. Приведем его краткое описание:

Алгоритм работы:

1. Текст примера разбивается на отдельные строки;
2. В цикле по всем строкам документа строки передаются в функции, которые пытаются извлечь каждый из элементов метаданных;
3. Название документа и автор могут находиться на нескольких подряд идущих строках. Для них начало и конец по шаблонам ищутся в подряд идущих строках. Остальные типы метаданных ищутся в пределах одной строки по заданным шаблонам.