**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский Авиационный Институт»**

**(Национальный Исследовательский Университет)**

**Институт: №8 «Информационные технологии и прикладная математика»**
**Кафедра: 806 «Вычислительная математика и программирование»**

Отчет по лабораторной работе №1

по предмету «Информационный поиск»

# «Добыча корпуса документов»

Группа: М8О-412Б-22

Студент(ка): Кайдалова А. А.

Оценка:

Дата сдачи:

Москва, 2025

## Задание

Необходимо проанализировать корпус документов, который будет использован при выполнении остальных лабораторных работ:

- Скачать примеры документов к себе на компьютер. В отчёте нужно указать источник данных. Источников в итоговом индексе должно быть не менее двух;
- Ознакомиться с ним, изучить его характеристики. Из чего состоит текст? Есть ли дополнительная мета-информация? Если разметка текста, какая она?
- Выделить текст.
- Найти существующие поисковики, которые уже можно использовать для поиска по выбранному набору документов (встроенный поиск Википедии, поиск Google с использованием ограничений на URL или на сайт). Если такого поиска найти невозможно, то использовать корпус для выполнения лабораторных работ нельзя!
- Привести несколько примеров запросов к существующим поисковикам, указать недостатки в полученной поисковой выдаче.

В результатах работы должна быть указаны статистическая информация о корпусе:

- Размер примеров «сырых» документов.
- Количество документов.
- Размер текста, выделенного из «сырых» данных.
- Средний размер документа, средний объём текста в документе.

## Журнал выполнения задания

**Тема.** Я выбрала тему для корпуса документов, который буду использовать во всех лабораторных работах. Тема – **материнство**. Мой выбор пал на данную сферу, так как в интернете есть большое количество статей, форумов и других источников для парсинга, также это важная социальная область.

**Источники** данных, которые я нашла: letidor.ru 7ya.ru, mama.ru.

**Поисковики.** Чтобы удостовериться, что выбранные источники подходят для лабораторных, в поисковой строке Google я ввела такой запрос: «первый прикорм схема site: letidor.ru OR site:7ya.ru OR site:mama.ru». При этом использовался оператор site: и логический оператор OR для объединения результатов с нескольких сайтов. Google выдал страницы, соответствующие запросу, но только с сайтов, которые я указала. Значит, мой корпус документов уже проиндексирован и доступен для поиска через мощную существующую систему. Посмотрим еще один запрос: «режим дня ребенка 2 лет» site:mama.ru Google возвращает релевантные статьи, но в выдаче присутствуют форумные темы и страницы категорий, что снижает точность. Можно

сказать, что существующие системы (Google, внутренние поиски сайтов) позволяют находить документы, но выдача содержит шум, отсутствует ранжирование по качеству контента.

**Характеристики документов из разных источников.**

Источник **7ya.ru:** семейный сайт. В html конкретной публикации есть сама статья, ее содержание, статья дня, похожие статьи, читайте также, поделитесь, похожие темы и тд. Мета-информация и разметка текста: в <head id="Head1"> <title> тема статьи, в блоке <main id="article" role="main" class="article">  <p> идет текст, внутри которого встречаются картинки <img itemprop="image" class="sberimg" src=", заголовки h2, h3 и тд, перечисления <p>. Далее идет автор <a href=" </a> и дата публикации и изменений <meta itemprop="datePublished" content=""/> <meta itemprop="dateModified" content="" и в  31.03.2025 Обновлено 31.03.2025 . Также могут встречаться статьи только с датой публикации, с ссылками в тексте <a title="" class="wordlink" href=" </a>, без картинок. Два примера размера сырого документа на этом сайте – 351 КБ, 348 КБ. В извлеченном тексте из первого примера всего символов (включая пробелы, знаки препинания): 5368, всего букв (без учета знаков препинания и пробелов): 8262. В извлеченном тексте из второго примера всего символов (включая пробелы, знаки препинания): 5013, всего букв (без учета знаков препинания и пробелов): 8006. **Итог**:

- Статья находится в ;
- Метаданные: <title>, <meta itemprop="datePublished">, автор в ссылке;

Источник **mama.ru**: сайт со статьями на разные темы материнства и беременности. На нем вкладки с разными темами статей, можно перейти на любую из них и выбрать конкретную публикацию для анализа. Рассмотрим html: <title> с темой, есть указания на другие статьи, вход, регистрация и тд. В бдоке <h4 class="fw-bold text-secondary mb-0"> ссылки на темы статьи. В <h1 class="fw-bold" itemprop="headline"> также указана тема,  время прочтения, <p dir="ltr"> тут уже идет текст статьи, в нем встречаются заголовки <h2 dir="ltr" id="period-novorozhdennosti-0-1-mesyac">, перечисления <ul><li dir="ltr" aria-level="1">, картинки <figure class="image"><img src=, автор <h5 class="fw-semibold small-text-on-mobile">, <h5 class="fw-semibold small-text-on-mobile"> дата публикации, ссылки на литературу <ol><li><p dir="ltr">. В статьях может не быть указание автора, картинок. Два примера размера сырого документа на этом сайте – 587 КБ, 587 КБ. В извлеченном тексте из первого примера всего символов (включая пробелы, знаки препинания): 6491, всего букв (без учета знаков препинания и пробелов): 5352. В извлеченном тексте из второго примера всего символов (включая пробелы, знаки препинания): 7425, всего букв (без учета знаков препинания и пробелов): 6038. **Итог**:

- Текст — в блоках ;
- Метаданные: <h1 itemprop="headline">, автор в <h5>, дата публикации;

Источник **letidor.ru:** сайт со статьями на темы для родителей: психология, образование, здоровье, право, звезды и дети и другие. На нем есть навигационное меню с рубриками, можно перейти на любую из них и выбрать конкретную публикацию для чтения. Рассмотрим html: `<title>` содержит название статьи и название сайта, есть мета-теги с описанием (description), указания на социальные сети, рекламные скрипты и т.д. В шапке сайта присутствует логотип, меню основных разделов, кнопки входа и регистрации. Данные о статье, включая текст, подгружаются динамически и отображаются в блоке с классами `jsx-...`. Заголовок статьи представлен в элементе `<h1>`. Анонс или введение статьи можно найти в мета-теге `og:description` или в атрибутах JSON-данных, встроенных в страницу (в теге `<script>` с `__REACT_QUERY_STATE__`). Текст статьи разбит на виджеты (widgets) типа `markdown`, `image`, `incut`, `readMore`. Непосредственный текст статьи находится в `widgets` с `"type":"markdown"` внутри атрибута `"body"`. В тексте встречаются подзаголовки, оформленные как `### Заголовок`, перечисления, могут быть картинки (`"type":"image"` с атрибутами `versions` и `credits`), блоки "Читайте также" (`"type":"readMore"`), врезки (`"type":"incut"`). Информация об авторе, дате публикации, времени чтения содержится в JSON-данных (в `attributes` статьи: `published_at`, `read_duration`, а также в связанных данных автора `person`). В статьях может не быть указания конкретного автора (например, если материал редакционный). В конце статьи могут быть ссылки на использованные источники или фото. Примерный размер сырого HTML-документа на этом сайте – 498 КБ, 587 КБ. В извлеченном чистом тексте из примера всего символов (включая пробелы, знаки препинания): 8510, всего букв (без учета знаков препинания и пробелов): примерно 3043. В извлеченном тексте из второго примера всего символов (включая пробелы, знаки препинания): примерно 3896, всего букв (без учета знаков препинания и пробелов): примерно 5708. **Итог**:

- Основной текст содержится в JSON-объектах внутри <script> под ключом "widgets" с "type":"markdown";
- Дата, автор, время чтения — в структурированных данных (published_at, person);

## Результаты

После выполнения 2 лабораторной был собран корпус из 54 543 документов.

| | source | article_count |
|---|---|---|
| 1 | 7ya.ru | 8 417 |
| 2 | letidor.ru | 40 685 |
| 3 | mama.ru | 5 441 |

Дальнейшие значения получены потем запросов к бд (брала рандомные 1000 документов для ускорения работы запросов).

| | |
|---|---|
| Количество документов | 54 543 |
| Источники | 7ya.ru, mama.ru, letidor.ru |
| Средний размер сырого HTML | 304.625 КБ |
| Средний размер извлечённого текста | 3 192 символов, 5.66 КБ |
| Общий объём бд | 8 ГБ |

## Выводы

Корпус документов по теме материнство:

- Уникален;
- Достаточно велик (54 543 документов);
- Проиндексирован существующими поисковиками;
- Содержит структурированные метаданные (автор, дата, заголовок).

## Приложения

Код из 2 лабороторной для извлечения текста:

```
def extract_article_text(html, source_url):

    soup = BeautifulSoup(html, 'html.parser')
    for tag in soup(["script", "style", "nav", "footer", "aside", "header"]):
        tag.decompose()

    try:
        for script in soup.find_all("script", type="application/ld+json"):
            data = json.loads(script.string)
            if isinstance(data, dict) and "articleBody" in data:
                return data["articleBody"].strip()
    except (ValueError, KeyError, TypeError):
        pass

    content = None

    if "7ya.ru" in source_url:
        content = soup.select_one('.articlebody')
    elif "mama.ru" in source_url:
```

```
        lead = soup.select_one('.article-lead--text.mt-2')
        lead_text = ""
        if lead:
            p_tag = lead.find('p')
            if p_tag:
                lead_text = p_tag.get_text(strip=True)
        article_block = soup.select_one('.article-block')
        main_text = ""
        if article_block:
            for tag in article_block(["script", "style", "nav", "footer", "aside", "header"]):
                tag.decompose()
            main_text = ' '.join(article_block.get_text().split())
        full_text = ' '.join([lead_text, main_text]).strip()
        return full_text if full_text else ""
    elif "letidor.ru" in source_url:
        texts = soup.select('div[data-qa="text"]')
        if texts:
            full_text = ' '.join(t.get_text().strip() for t in texts)
            return full_text
    else:
        content = soup

    return ' '.join(content.get_text().split()) if content else ''
```

Код из 2 лабороторной для извлечения метаданных:

```
def extract_article_metadata(html, source_url):
    soup = BeautifulSoup(html, 'html.parser')
    title = ""
    author = ""
    publish_date = ""

    try:
        for script in soup.find_all("script", type="application/ld+json"):
            data = json.loads(script.string)
            if isinstance(data, dict):
                title = data.get("headline", "").strip()
                author_obj = data.get("author", {})
                if isinstance(author_obj, dict):
                    author = author_obj.get("name", "").strip()
                elif isinstance(author_obj, str):
                    author = author_obj.strip()
                dt = data.get("datePublished", "")
                if dt:
                    publish_date = dt.split("T")[0]
                if title and author and publish_date:
                    return title, author, publish_date
    except (ValueError, KeyError, TypeError):
        pass

    if "7ya.ru" in source_url:
        title_tag = soup.select_one('h1[itemprop="headline"]')
        author_tag = soup.select_one('div.type a[href*="club.7ya.ru"]')
        date_tag = soup.find('meta', {'itemprop': 'datePublished'})
        title = title_tag.get_text(strip=True) if title_tag else ""
        author = author_tag.get_text(strip=True) if author_tag else ""
        publish_date = date_tag['content'] if date_tag and date_tag.get('content') else ""

    elif "mama.ru" in source_url:
        title_tag = soup.select_one('h1[itemprop="headline"]')
        author_tag = soup.select_one('h5.fw-semibold.small-text-on-mobile')
        date_meta = soup.find('meta', {'itemprop': 'datePublished'})
        publish_date = ""
        if date_meta and date_meta.get('content'):
            publish_date = date_meta['content'].split('T')[0]
        title = title_tag.get_text(strip=True) if title_tag else ""
```

```
        author = author_tag.get_text(strip=True) if author_tag else ""
    elif "letidor.ru" in source_url:
        title_tag = soup.select_one('h1[data-qa="lb-topic-header-texts-title"]')
        title = title_tag.get_text(strip=True) if title_tag else ""
        author_tag = soup.select_one('div.jsx-1196406913 a.link')
        author = author_tag.get_text(strip=True) if author_tag else ""
        date_div = soup.select_one('div.J57431KZ.M7gu8GJc')
        if date_div:
            date_text = date_div.get_text().strip()
            match = re.search(r'(\d{1,2})\s+([а-яА-Я]+)\s+(\d{4})', date_text)
            if match:
                day, month_ru, year = match.groups()
                months = {
                    "января": "01", "февраля": "02", "марта": "03", "апреля": "04",
                    "мая": "05", "июня": "06", "июля": "07", "августа": "08",
                    "сентября": "09", "октября": "10", "ноября": "11", "декабря": "12"
                }
                month = months.get(month_ru, "01")
                publish_date = f"{year}-{month}-{day.zfill(2)}"
    return title, author, publish_date
```