Известия СПбГЭТУ «ЛЭТИ». 2022. Т. 15, № 4. С. 35–43
LETI Transactions on Electrical Engineering & Computer Science. 2022. Vol. 15, no. 4. P. 35–43

УДК 004.622 Научная статья

https://doi.org/10.32603/2071-8985-2022-15-4-35-43

Краулер для формирования датасета пользовательских соглашений на использование персональных данных

М. Д. Кузнецов[1]✉, **Е. С. Новикова**[2]

[1] Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина), Санкт-Петербург, Россия

[2] Санкт-Петербургский Федеральный исследовательский центр Российской академии наук, Санкт-Петербург, Россия

mkuznetsov7991@gmail.com✉

Аннотация. Сбор и использование персональных данных для удовлетворения цифровых потребностей пользователей сегодня являются крайне распространенными сценариями. Пользователи активно предоставляют свои персональные данные для улучшения качества цифровых сервисов. В то же время, пользовательские соглашения – единственный инструмент информирования о том, какие персональные данные и как используются. Существуют разные подходы к повышению прозрачности пользовательских соглашений, однако для большинства этих подходов требуются данные для проведения экспериментов и для обучения моделей искусственного интеллекта. В настоящее время датасетов для исследования пользовательских соглашений немного, а те, которые имеются, не покрывают рынок умных устройств. Умные устройства генерируют огромный трафик, состоящий из персональных данных, поэтому их пользовательские соглашения заслуживают не меньшего внимания. В данной работе авторы предлагают новый способ формирования датасета пользовательских соглашений, а также представляют соответствующий инструмент, обладающий помимо основных функций рядом улучшений для обхода блокировок и captcha.

Ключевые слова: соглашения об использовании персональных данных, краулер, датасет, сбор данных, очистка данных

Для цитирования: Кузнецов М. Д., Новикова Е. С. Краулер для формирования датасета пользовательских соглашений на использование персональных данных // Изв. СПбГЭТУ «ЛЭТИ». 2022. Т. 15, № 4. С. 35–43. doi: 10.32603/2071-8985-2022-15-4-35-43.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Original article

The Crawler Forming the Dataset of User Agreements for the Use of Personal Data

M. D. Kuznetsov[1]✉, **E. S. Novikova**[2]

[1] Saint Petersburg Electrotechnical University, Saint Petersburg, Russia

[2] Saint Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS), Saint Petersburg, Russia

mkuznetsov7991@gmail.com✉

Abstract. The collection and use of personal data to meet the digital needs of users are extremely common scenarios today. Users actively provide their personal data to improve the quality of digital services. At the same time, user agreements are the only tool for informing which personal data is used and how. There are different approaches to make user agreements more transparent, but most of these approaches require data to both experiment and train deep learning models. Currently, there are few datasets for researching of user

agreements, and those that are available do not cover the internet-of-things market. Smart devices generate a huge amount of personal data traffic, so their user agreements deserve just as much attention as agreements of the websites. In this paper, the authors propose a new way of forming a dataset of user agreements, and also presents a corresponding tool that, in addition to the main functions, has a number of improvements for bypassing blocks, bans and captcha.

Keywords: personal data agreements, crawler, dataset, data collection, data cleaning

For citation: Kuznetsov M. D., Novikova E. S. The Crawler Forming the Dataset of User Agreements for the Use of Personal Data // LETI Transactions on Electrical Engineering & Computer Science. 2022. Vol. 15, no. 4. P. 35–43. doi: 10.32603/2071-8985-2022-15-4-35-43.

Conflict of interest. The authors declare no conflicts of interest.

Введение. Рынок Интернета и, в частности, интернета вещей самая быстрорастущая область бизнеса. Такие темпы развития образуют непокрытые законодательством средства и способы обработки персональных данных, и, по сути, только пользовательские соглашения об использовании и обработке персональных данных регулируют законность их оборота. В связи с рядом больших утечек персональных данных внимание сообщества было обращено на способы регулирования их оборота, в результате чего были приняты меры по улучшению законного регулирования сбора и обработки персональных данных. К таким документам относится General Data Protection Regulation (GDPR), в Российской Федерации действует Федеральный Закон № 152. Эти законодательные меры обязали поставщиков цифровых услуг четко прописывать правила оборота персональных данных. Однако пользовательские соглашения на обработку данных, призванные прояснить ситуацию с персональными данными, зачастую написаны крайне сложно, так что конечные пользователи не всегда могут понять, с какими правилами они соглашаются. Поэтому проблема прозрачности пользовательских соглашений стоит как никогда остро.

Специалисты всего мира ведут исследования по указанной проблеме. К самым важным относятся работы [1]–[5]. В них исследователи применяют различные методы текстового анализа, но самыми точными из них пока остаются методы, основанные на моделях глубокого обучения. Несмотря на их точность, они все же обладают рядом важных особенностей, затрудняющих и замедляющих исследования. Например, такие модели нуждаются в качественных экспериментальных данных и обучающих выборках, разработка и формирование которых – крайне трудоемкая и рутинная задача, и на данный момент отсутствуют такие обучающие выборки для пользовательских соглашений «умных» устройств. В [1], [2] для получения результатов авторы провели большую работу по сбору пользовательских соглашений, а также провели аннотирование при поддержке квалифицированных юристов. Здесь важно то, что исследования велись до принятия законов, регулирующих обработку и хранение персональных данных, и то, что исследуемые ими пользовательские соглашения были предназначены для веб-сайтов.

Помимо данного раздела статья содержит еще три. В следующем разделе представлена постановка задачи, в третьем обсуждаются особенности и потенциальные проблемы, которые могут возникнуть при формировании датасета и разработке краулера. Четвертый раздел посвящен непосредственно инструменту формирования датасета и методике, вложенной в него. В последнем разделе обсуждаются полученные результаты и перспективы дальнейших исследований в данной области.

Постановка задачи. Для того чтобы продолжить разработки и исследование пользовательских соглашений для «умных» устройств, необходима новая обучающая выборка, которая покроет и опишет состояние рынка «умных» устройств с учетом ныне действующих законодательных мер. Задача формирования обучающей выборки пользовательских соглашений может быть решена разными способами, в данной работе предложены методики, позволяющие сформировать требуемую выборку, а также их программные реализации. Решение проблемы позволит получить необходимую выборку данных для дальнейших исследований. Кроме того, выборка станет важным шагом в сторону формализации и структуризации пользовательских соглашений на обработку персональных данных. В свою очередь формализованные представления пользовательских соглашений могут быть использованы для разработки

систем поддержки принятия решений при управлении персональными данными пользователей.

Потенциальные проблемы. Еще до решения задачи были выделены потенциальные проблемы, способные замедлить процесс разработки и сбора датасета. Потенциально возможные проблемы при реализации приложений подобного типа следующие:

1. Блокировка из-за подозрительных заголовков браузера.

2. Блокировка из-за слишком частого обращения с запросами.

3. Как следствие двух предыдущих пунктов – требование подтвердить, что это не попытка автоматического доступа (ввод captcha).

4. Невидимые элементы разметки.

5. Динамически формируемые страницы торговых площадок и пользовательских соглашений.

6. Промахи при сборе данных из-за частично некорректных результатов поиска на торговых площадках и в поисковых движках.

Проблемы 1–3 решаются использованием разных заголовков браузера между запросами. Также отправка запросов ограничена по частоте от 2 до 6 с, ограничение выбирается случайным образом. Такие решения позволяют крайне редко попадать под подозрения, потому что в таком случае поведение максимально похоже на поведение реального пользователя, соответственно, доля успеха при попытке получить данные с веб-страницы значительно повышается. Стоит отметить, что данные ограничения обходятся за счет использования прокси-серверов, которые позволяют менять IP-адреса. Еще один важный и эффективный инструмент – профиль браузера. Он позволяет запускать браузер в режиме удаленного управления (безголовый) с определенной историей использования – будь то куки-файлы, история запросов или аутентификация в различных сервисах. Наличие такой предыстории у браузера для некоторых сайтов является доказательством, что его работа не автоматизирована.

Проблема 4 решается следующим образом. Попав на страницу пользовательского соглашения, можно исполнить javascript-код, который загрузит на страницу библиотеку для работы с деревом DOM и удалит невидимые элементы разметки.

Проблема 5 решается использованием «безголового» браузера, который полнофункционален с точки зрения воспроизведения контента, так как поддерживает исполнение javascript-кода на странице. Таким образом, страница будет загружена и динамические элементы будут созданы, после чего можно будет их обработать. Однако на некоторых веб-сайтах для того, чтобы получить ту или иную информацию, необходимо заполнить определенную форму. С такими обстоятельствами сложно бороться – разметка всегда различается, но таких случаев крайне мало, поэтому исключение их из рассмотрения будет оправданным.

Проблема 6 может отчасти решиться конкретизацией поискового запроса посредством прибавления к названию производителя ключевых слов и названий продукции, которая им производится. Хотя этот вариант и показал качественные результаты, более точным оказался поиск производителя «как есть» (поиск исключительно по названию фирмы производителя), но иногда все же попадаются некорректные результаты.

Методология решения. Начальный этап решения задачи – это первичная декомпозиция. Ее результат – выделение подзадач различной важности, которые должны быть решены для доведения цикла разработки до конца. В данном случае можно выделить следующие подзадачи:

- определение источника информации о различной IoT-продукции,
- отправка поискового запроса,
- получение результатов запроса (список IoT-продуктов),
- определение производителей IoT-продукции,
- поиск официальных сайтов производителей в сети Интернет,
- поиск раздела «пользовательское соглашение» на сайтах производителей,
- скачивание пользовательских соглашений,
- очистка скачанных веб-документов от лишних элементов разметки,
- слияние тегов и оборачивание сырого текста,
- нормализация пунктуации и отступов,
- извлечение текста из тегов.

Исходя из результатов декомпозиции, эффективным подходом выглядит представление приложения в виде последовательно выполняющихся подпрограмм так, что входом модуля становится результат работы предыдущего модуля, т. е. в виде конвейера.

Важным вопросом остается выбор источника данных, например в [6] был использован репозиторий веб-ссылок DMOZ, в [1] для получения данных – сервис Amazon Alexa, в [2] средством получения данных выступал api сервиса Google Play. Действия авторов этих работ понятны и ло-

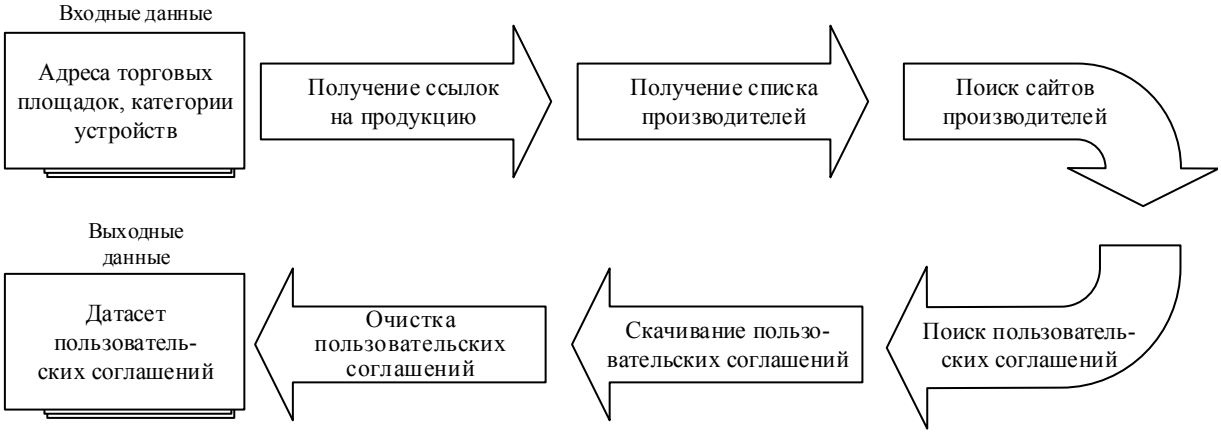

Рис. 1. Алгоритм сбора данных
Fig. 1. Data collection algorithm

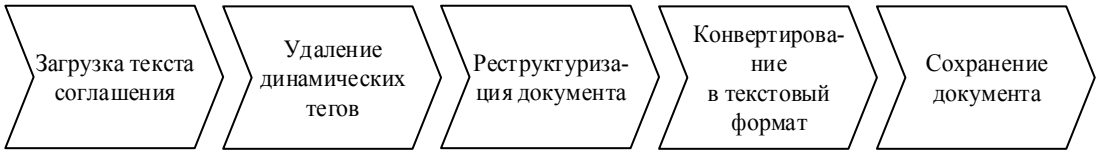

Рис. 2. Алгоритм очистки
Fig. 2. Document sanitization algorithm

гичны – они берут информацию из источников, которые агрегируют в себе необходимую им информацию. К сожалению, в случае с «умными» устройствами такой подход не работает, так как отсутствует централизованный сервис или репозиторий, содержащий данные об «умных» устройствах, поэтому авторам пришлось предложить свой подход к сбору данных.

Торговые интернет-площадки нацелены на сбыт «умных» устройств, обладают информацией о производителе и ряде других параметров. Html-разметка страниц «умных» устройств редко отличается от устройства к устройству, поэтому получить данные, применяя определенный алгоритм, не составляет большого труда. Далее, воспользовавшись поисковым сервисом Google, можно определить веб-страницы производителей «умной» продукции. Затем, пользуясь глобальной навигацией веб-сайтов, легко определить ссылку на пользовательское соглашение. Таким образом, авторы пришли к методике формирования, представленной на рис. 1. Сначала ищутся ссылки на страницы устройств, находящихся в продаже на различных торговых площадках, затем из разметки страниц извлекаются данные о производителе, осуществляется поиск веб-сайтов производителей, поиск соглашений на этих сайтах, скачивание и очистка пользовательских соглашений.

В разработанной методике следует уделить внимание очистке скачанных пользовательских соглашений. Она необходима для проведения исследований текстов пользовательских соглашений, так как сырые веб-страницы содержат нежелательный код, который должен быть удален, чтобы извлечь полезные текстовые данные. Для этого авторы предложили следующий порядок действий, который представлен на рис. 2.

Для дальнейшего продвижения работ по реализации краулера была построена композиционная модель приложения; на ней присутствуют все необходимые для решения задач модули. Схема представлена на рис. 3.

Для реализации приложения были выбраны следующие средства:

- python 3.9;
- браузер Firefox;
- драйвер geckodriver для управления браузером в удаленном режиме;
- библиотека html-sanitizer для очистки скачанных веб-документов.

Выбор «безголового» браузера обусловлен потребностью в отрисовке страниц, так как на некоторых веб-страницах разметка генерируется с помощью javascript. Это делает невозможным использование простого скачивания, необходима страница именно с исполненными скриптами, в противном случае будет невозможно получить требуемую информацию. В то же время браузер лишен графического интерфейса, вследствие чего снижается потребление вычислительных ресурсов.

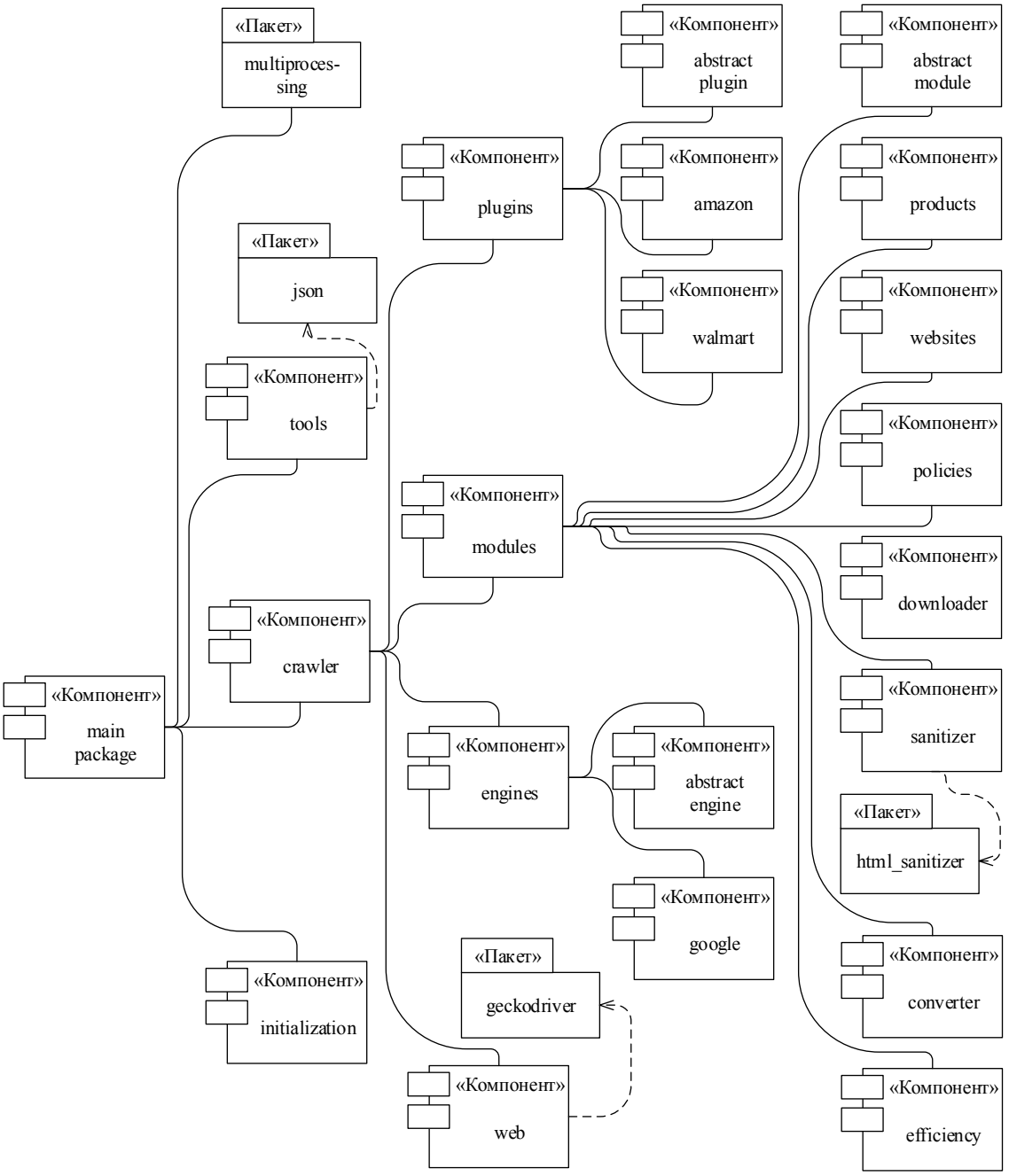

Рис. 3. Композиционная модель приложения в нотации UML
Fig. 3. Component model of application in UML notation

Приложение для формирования обучающей выборки пользовательских соглашений на использование и обработку персональных данных построено на четырех основных абстракциях.

Концепция модуля – одна из основополагающих, так как модулем в данном случае выступает любая подпрограмма, участвующая в сборе данных, принимающая входные данные в виде json-файла, и на выходе дающая также json-файл, чтобы следующий в очереди модуль мог отработать. Модули могут быть написаны с нуля, а могут расширять возможности уже существующих посредством механизма наследования. Таким образом можно, не переписывая существующий код, изменять поведение программы и адаптировать ее под разные задачи сбора данных.

Концепция конвейера – этот элемент поочередно вызывает модули и передает данные из одного модуля в другой. В результате отработки всех модулей поэтапно решается поставленная задача, т. е. сбор данных из интернет-источников. Конвейер может быть сконфигурирован, в него могут быть помещены любые модули, реализующие соответствующий интерфейс. Также может

быть сконфигурирована последовательность запуска модулей сбора данных.

Концепция поискового движка – данная концепция порождена в связи с необходимостью сделать приложение как можно более гибким. Такой абстрактный элемент позволяет менять используемые поисковые движки, применять к результатам поиска алгоритмы для определения, какие результаты удовлетворяют условиям поиска, а какие нет.

Концепция плагина – плагин обеспечивает сбор данных с целевой торговой площадки. Данная концепция использована также для обеспечения гибкости приложения – для устранения привязки к набору конкретных торговых площадок. Используя механизм наследования, можно переопределить поведение плагина для работы с любой другой торговой площадкой.

На рис. 3 модуль «main» отвечает за запуск программы и развертывание основных ее частей. Там же происходит инициализация пула процессов для мультипроцессинга затратных задач – таких, как, например, взаимодействие с браузером. Он также отвечает за последовательное исполнение подпрограмм элементов конвейера, осуществляет прием выходных и передачу входных данных модулей:

– модуль «initialization» проводит проверку файловой системы и создает необходимые директории в папке ресурсов;

– модуль «tools» содержит вспомогательные функции, в частности для ввода и вывода данных в формате json;

– модуль «crawler» отвечает за получение данных с веб-страниц, в нем агрегированы все инструменты для сбора и очистки данных;

– модуль «plugins» включает в себя набор плагинов, каждый из которых адаптирован для получения требуемой информации с определенного шаблона веб-страничной разметки. Некоторое поведение инкапсулировано в абстрактном плагине для увеличения повторной применимости программного кода. Получая адрес на вход, данный плагин скачивает страницу и с помощью набора шаблонов пытается извлечь информацию;

– данные, полученные с помощью модулей «products», «websites», «policies», «downloader», «anitizer», «converter» и «efficiency», записываются в json-файлы для большей прозрачности и возможности сохранения результатов между запусками приложения, например при пропуске какого-либо из этапов и использования его сохраненных результатов работы;

– модуль «products» реализует получение производителей IoT-продуктов;

– модуль «websites» получение официальных сайтов производителей,

– модуль «policies» получение веб-ссылок на пользовательские соглашения;

– модуль «downloader» отвечает за скачивание страниц и их сохранение в отведенную для этого директорию;

– модуль «sanitizer» отвечает за очистку скачанных веб-страниц от ненужных тегов и ссылок;

– модуль «converter» производит перевод пользовательских соглашений из веб-страничного вида в текстовое представление;

– модуль «efficiency» рассчитывает статистику по датасету;

– модуль «web» отвечает за взаимодействие с веб-сайтами, будь то торговые площадки или сайты производителей IoT-продуктов; в нем используется geckodriver для управления «безголовым» браузером;

– модуль «proxy» содержит инструменты для скачивания и автоматического применения бесплатных прокси-серверов. Однако ввиду ненадежности бесплатных прокси-серверов есть также возможность задать список выделенных прокси-серверов.

Для обеспечения наиболее гибкой настройки как можно больше настроек выведено в отдельный конфигурационный файл. В нем задаются:

– параметры для библиотеки html-sanitizer, в частности набор допустимых тегов и атрибутов;

– параметры браузера, в том числе количество повторных попыток при сбоях, появлениях captcha и т. д., набор юзерагентов для перебора, флаги использования кэширования, флаг запуска браузера в режиме без графического интерфейса, флаг использования прокси, пути для логов, а также путь до профиля браузера в файловой системе;

– список директорий и файлов, в которые происходит сохранение результатов сбора данных;

– количество процессов для одновременного сбора данных на многоядерных конфигурациях.

Для настройки работы заменяемых элементов – поисковых движков плагинов и модулей, предусмотрены отдельные файлы, в которых создаются те или иные конфигурируемые объекты.

Учитывая конвейерную организацию и передачу результатов из модуля в модуль посредством json-файлов, структура датасета следующая: каждый модуль имеет свой json-файл для записи результатов. По сути, результаты – это массив из python-словарей, каждый словарь является своего

рода кортежем, эти кортежи обладают избыточностью данных, однако таким образом достигается максимальная простота формализации данных. Каждый элемент – IoT-устройство, обладающее набором информационных полей: идентификатор; ссылка на страницу на торговой площадке; наименование производителя; ключевое слово, по которому было найдено устройство; ссылка на сайт производителя; ссылка на пользовательское соглашение; путь к сохраненной оригинальной странице пользовательского соглашения; путь к очищенному пользовательскому соглашению; путь к текстовой версии пользовательского соглашения; хеш, сгенерированный по тексту соглашения; блок статистики по структурным элементам – нумерованным и ненумерованным спискам, элементам списков, таблиц, параграфов, длины соглашения в символах и т. п. Пример такой разметки можно увидеть на рис. 4.

```
{
  "id": 1,
  "url": "https://www.walmart.com/ip/GreaterGoods-Smart-Scale-BT-Connected-Body-Weight-Bathroom-Scale-BMI-Body-Fat-Muscle-Mass-Water-Weight-FSA-HSA-Approved/696264102",
  "manufacturer": "greater goods",
  "keyword": "smart scale",
  "website": "http://greatergoods.com",
  "policy": "http://greatergoods.com/legal/privacy-policy",
  "original_policy": "D:\\source\\repos\\iot-crawler\\resources\\original_policies\\greatergoods.com-legal-privacy-policy.html",
  "processed_policy": "D:\\source\\repos\\iot-crawler\\resources\\processed_policies\\greatergoods.com-legal-privacy-policy.html",
  "plain_policy": null,
  "policy_hash": "9d63c3eeb2a4ef4ad0b4428ad56d4be5",
  "statistics": {
    "length": 24315,
    "table": 0,
    "ol": 0,
    "ul": 7,
    "li": 27,
    "p": 39,
    "br": 5
  }
},
```

Рис. 4. Пример кортежа датасета
Fig. 4. Dataset tuple sample

В веб-краулере также предусмотрена возможность явного указания адресов для скачивания пользовательских соглашений, для чего существует отдельный json-файл, содержащий элементы со схожей структурой. В нем можно указывать любые из полей – они будут заполнены соответствующе, а незаполненные поля останутся равными «null». Явно заданные для скачивания политики считываются непосредственно на этапе скачивания, таким образом данные о названии производителя и другие данные, которые участвуют в более ранних стадиях сбора, несут сугубо справочный характер. Статистические показатели пользовательских соглашений рассчитываются на последнем этапе работы приложения, что означает их перезапись после каждого запуска при условии, что модуль расчета статистики активен.

Выводы и заключение. В данной статье было обращено внимание на неразработанную область регулирования персональных данных – «умные» устройства. Авторы предложили методики формирования обучающих выборок, основанные на последовательном сборе данных из разных источников: сначала сбор данных об «умной» продукции из открытых источников (торговые площадки), затем поиск веб-сайтов производителей «умной» продукции, далее поиск пользовательских соглашений на веб-сайтах производителей.

Тестирование инструмента, реализующего предложенные методики, показало его достаточную эффективность. В результате работы краулера удалось собрать внушительное количество пользовательских соглашений [7] – 592 соглашения, что более чем в 5 раз больше, чем в [1]. Сформированная обучающая выборка – важный шаг в исследованиях пользовательских соглашений «умных» устройств на обработку персональных данных. Данная выборка уже используется для проведения различных исследований, таких как в [8]. В дальнейшем такая обучающая выборка позволит разработать методы формализации и структурирования пользовательских соглашений, что значительно повысит их прозрачность для конечных пользователей. Кроме того, методы формализации и структуризации являются переходными, благодаря им становится возможной разработка систем поддержки принятия решений при управлении персональными данными пользователей.

Список литературы

1. The creation and analysis of a website privacy policy corpus / S. Wilson, F. Schaub, A. A. Dara, F. Liu, S. Cherivirala, P. G. Leon, M. S. Andersen, S. Zimmeck, K. M. Sathyendra, N. C. Russell, T. B. Norton, E. Hovy, J. Reidenberg, N. Sadeh // Proc. of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin, Germany, 2016. P. 1330–1340. doi: 10.18653/v1/P16-1126.

2. MAPS: scaling privacy compliance analysis to a million apps / S. Zimmeck, P. Story, D. Smullen, A. Ravichander // Proc. on Privacy Enhancing Technologies. Stockolm, Sweden, 2019. Vol. 3. P. 66–86. doi: 10.2478/popets-2019-0037.

3. PrivOnto: a semantic framework for the analysis of privacy policies, Semantic Web / A. Oltramari, P. Piraviperumal, F. Schaub, S. Wilson, N. Sadeh, J. Reidenberg // 2018. Vol. 9. P. 185–203. doi: 10.3233/SW-170283.

4. Polisis: automated analysis and presentation of privacy policies using deep learning / H. Harkous, K. Fawaz, R. Lebret, F. Schaub, K. G. Shin, K. Aberer // USENIX Security, 2018. P. 1–22. doi: 10.48550/arXiv.1802.02561.

5. Novikova E., Doynikova E., Kotenko I. P2Onto: Making Privacy Policies Transparent, CyberICPS 2020, SECPRE 2020, ADIoT 2020 // Lecture Notes in Computer Science, Springer. 2020. Vol. 12501. P. 235–252. doi: 10.1007/978-3-030-64330-0_15.

6. Zaeem R. N., Barber K. S. A large publicly available corpus of website privacy policies based on DMOZ // Proc. of the Eleventh ACM Conf. on Data and Appl. Security and Privacy (CODASPY '21). Association for Computing Machinery, 2021. P. 143–148. doi: 10.1145/3422337.3447827.

7. Privacy policies of IoT devices: Collection and analysis / M. Kuznetsov, E. Novikova, I. Kotenko, E. Doynikova // MDPI Sensors. 2022. Vol. 22. P. 1–23. doi: 10.3390/s22051838.

8. Kuznetzov M., Novikova E. Towards application of text mining techniques to the analysis of the privacy policies // 10th Mediterranean Conf. on Embedded Computing (MECO). Budva: Institute of Electrical and Electronics Engineers Inc 2021. P. 1–4. doi: 10.1109/MECO52532.2021.9460130.

Информация об авторах

Кузнецов Михаил Дмитриевич – аспирант кафедры информационных систем СПбГЭТУ «ЛЭТИ».
E-mail: mkuznetsov7991@gmail.com
https://orcid.org/0000-0002-0970-8473

Новикова Евгения Сергеевна – канд. техн. наук, старший научный сотрудник лаборатории проблем компьютерной безопасности СПб ФИЦ РАН.
E-mail: esnovikova@comsec.spb.ru
https://orcid.org/0000-0003-2923-4954

References

1. Wilson S., Schaub F., Dara A. A., Liu F., Cherivirala S., Leon P. G., Andersen M. S., Zimmeck S., Sathyendra K. M., Russell N. C., Norton T. B., Hovy E., Reidenberg J., Sadeh N. The Creation and Analysis of a Website Privacy Policy Corpus // Proc. of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin, Germany, 2016. P. 1330–1340. doi: 10.18653/v1/P16-1126.

2. Zimmeck S., Story P., Smullen D., Ravichander A. MAPS: Scaling Privacy Compliance Analysis to a Million Apps // In Proc. on Privacy Enhancing Technologies. Stockolm, Sweden, 2019. Vol. 3. P. 66–86. doi: 10.2478/popets-2019-0037.

3. Oltramari A., Piraviperumal P., Schaub F., Wilson S., Sadeh N., Reidenberg J. PrivOnto: a Semantic Framework for the Analysis of Privacy Policies, Semantic Web, 2018. Vol. 9. P. 185–203. doi: 10.3233/SW-170283.

4. Harkous H., Fawaz K., Lebret R., Schaub F., Shin K. G., Aberer K. Polisis: Automated Analysis and Presentation of Privacy Policies Using Deep Learning. USENIX Security, 2018. P. 1–22. doi: 10.48550/arXiv.1802.02561.

5. Novikova E., Doynikova E., Kotenko I. P2Onto: Making Privacy Policies Transparent, CyberICPS 2020, SECPRE 2020, ADIoT 2020 // Lecture Notes in Computer Science, Springer. 2020. Vol. 12501. P. 235–252. doi: 10.1007/978-3-030-64330-0_15.

6. Zaeem R. N., Barber K. S. A Large Publicly Available Corpus of Website Privacy Policies Based on DMOZ // Proc. of the Eleventh ACM Conf. on Data and Appl. Security and Privacy (CODASPY '21). Association for Computing Machinery, 2021. P. 143–148. doi: 10.1145/3422337.3447827.

7. Kuznetsov M., Novikova E., Kotenko I., Doynikova E. Privacy Policies of IoT Devices: Collection and Analysis // MDPI Sensors. 2022. Vol. 22. P. 1–23. doi: 10.3390/s22051838.

8. Kuznetzov M., Novikova E. Towards Application of Text Mining Techniques to the Analysis of the Privacy Policies // 10th Mediterranean Conf. on Embedded Computing (MECO). Budva: Institute of Electrical and Electronics Engineers Inc 2021. P. 1–4. doi: 10.1109/MECO52532.2021.9460130.

Information about the authors

Mikhail D. Kuznetsov – postgraduate student, Department of Information Systems, Saint Petersburg Electrotechnical University.
E-mail: mkuznetsov7991@gmail.com
https://orcid.org/0000-0002-0970-8473

Evgenia S. Novikova – Cand. Sci. (Eng.), senior researcher of the Laboratory of Computer Security Problems of SPC RAS (St. Petersburg Federal Research Center of the Russian Academy of Sciences).
E-mail: esnovikova@comsec.spb.ru
https://orcid.org/0000-0003-2923-4954

Статья поступила в редакцию 01.04.2022; принята к публикации после рецензирования 07.04.2022; опубликована онлайн 28.04.2022.

Submitted 01.04.2022; accepted 07.04.2022; published online 28.04.2022.