2024-09-10

Статические и динамические веб-данные

Статические веб-данные обычно* не меняются со временем и не зависят от клиента. Например, видео на видеохостинге: загруженное единожды видео не меняет содержимое видеопотока «на лету», оно обычно загружается 1 раз в определенный момент времени и остается неизменным вне зависимости от того, сколько раз или с каких устройств вы с ним взаимодействуете.

*но видео может быть удалено, некоторые хостинги (например, YouTube) позволяет части своих клиентов менять видео без его перезагрузки на платформу, в видео могут быть ошибки, которые исправлены не в видеопотоке (а, например, в посте в социальных сетях) и т.п.

Но есть и неплохие примеры — скажем, библиотека Мошкова; загруженный файл, например, «Преступления и наказания» не меняется многие годы (Lib.ru/Классика: Достоевский Федор Михайлович. Преступление и наказание).

Динамические веб-данные либо меняются в зависимости от клиента («черные списки» или пейволл — доступ к полной версии только после оплаты, могут быть и другие виды ограничений — например, региональные, по подсетям и т.п.), либо без предупреждения регулярно меняются со временем. Например, описание под видео с видеохостинга: после

загрузки оно может быть изменено без уведомлений или даже следов изменения сколько угодно раз; можно менять даже название видео и обложку.

Разумеется, динамических данных сейчас гораздо больше. Поэтому важно при сборе и анализе данных учитывать временной атрибут: когда данные были получены. Отсюда же идут понятия версионности данных (сама история изменений), множества состояний (возможность получить любую из предыдущих версий в полном объеме) и среза данных (единовременный сбор и фиксация текущего состояния веб-ресурса и размещенных на нем данных без сравнения с предыдущими версиями напрямую). Без учета фактора времени корректный анализ динамических веб-данных чаще всего невозможен.

Множество срезов данных позволяет со временем сформировать свои наборы данных с возможностью уже сравнения и версионирования. Но сама процедура среза (или снятия дампа, снятия слепка) это не подразумевает, это просто следствие накопления данных у сборщика.

Посмотреть на то, как работает такой механизм, можно через Wayback Machine https://archive.org/web/

Example

Как можно использовать Wayback Machine

При помощи хроники срезов пользователи оценили

кадровую ситуацию в публичной компании на основании того, как со временем менялась страница с персоналом (сотрудник + его должность).
Анализ динамики веб-данных позволил отследить изменения в должностях, а также найти возможные даты начала и конца работы в компании.
Через сопоставление с другими источниками (социальные сети) удалось также оценить тональность высказываний бывших сотрудников и в целом сделать ряд предположений об условиях труда (частично подтвердившихся).

Сбор, хранение и обработка веб-данных

Основные сложности/факторы риска:

1. Доступ: многие веб-ресурсы ограничивают или блокируют автоматизированный доступ к своим данным (даже публичным)
2. Полнота: не все данные можно собрать в полном (достаточном для корректного и эффективного анализа) объеме
3. Повторы (дубликаты) и (что особенно вредно) повторы семантические (технически и/или формально данные могут быть различны, но с точки зрения сутевого анализа представляют собой дубликаты; пример — картинка в разных форматах)

4. Неструктурированность (слабоструктурированность): данные не имеют четкой структуры, она (структура) может меняться без предупреждения, структура никак не документируется

> **Note**
>
> Но бывает и наоборот, в том числе в довольно нишевых областях.
> Например — база данных карт игры Magic: The Gathering Scryfall

5. Избыточность: избыточные и/или нерелевантные данные, которые никак не способствуют решению задачи, но которые невозможно или слишком трудоемко отделить от полезных данных
6. Некачественные данные: «мусорные» данные в одном наборе, сильно отличающиеся друг от друга по содержанию, формату, полноте информации; анализ таких данных затруднен из-за их изначального состояния
7. Нечеткие определения сущностей: разные веб-ресурсы могут по-разному определять одни и те же сущности. Оценка в рецензии как пример: она может быть по 10-балльной, 100-балльной, 4 звезды из 4, 5 звезд из 5, она может быть лингвистической переменной (плохо-нормально-хорошо-великолепно), она может вообще не быть формально обозначенной на ресурсе, она может быть обозначена цветом или изображением (медаль-

орден-корона). Но при анализе N веб-ресурсов с рецензиями сущность «оценка рецензента» должна быть приведена к единому виду. Сразу возникает вопрос: к какому виду и как? Если, например, брать «негативная»/ «нейтральная»/ «позитивная», то какие правила будут применяться к каждому конкретному случаю, чтобы конвертировать их в данные категории?

8. Запись в хранилище: данные, которые собираются параллельно и/или асинхронно, могут вызывать сложности с записью в хранилище в хронологическом порядке.

Веб-данные необходимо хранить с сохранением связей между ними. Для этого активно применяются документоориентированные и графовые базы данных (TypeDB). GraphQL — язык запросов к графовым базам данных.

Одно из крупнейших хранилищ, основанное на графовой структуре: https://commoncrawl.org/

Практика 1

- воспользоваться Common Crawl - Get Started; CommonCrawl with Python - Get All Pages from a Domain - JC Chouinard для освоения доступа к базе Common Crawl
- можно визуально поизучать Common Crawl - Overview

- можно посмотреть наработки CmonCrawl · PyPI и GitHub - michaelharms/comcrawl: A python utility for downloading Common Crawl data
- после этого собрать консольное приложение, которое осуществляет поиск по Common Crawl и выводит перечень связанные с запросом страниц
- поискать там упоминания г. Перми, Пермского Политеха, кафедры ИТАС; МГУ им. Ломоносова, МФТИ им. Баумана; Бориса Пастернака в контексте г. Перми
- представить результаты в виде текстового вывода