

Powered by **zyte**

Can ChatGPT solve Web Data Extraction?

Extract Summit 2023

Can ChatGPT solve Web Data Extraction?

What LLMs are capable of?

Paradigms of using LLMs

About me: Konstantin Lopukhin

Head of Data Science at Zyte

https://github.com/lopuhin

konstantin@zyte.com



It used to be good



Tell me the product name, description, current price and image urls from the webpage page http://books.toscrape.com/catalogue/a-light-in-the-attic_1000/index.html



Product name: A Light in the Attic

Description: A Light in the Attic is a collection of humorous and imaginative poems for children, written by Shel Silverstein. The book features over 100 poems that cover a variety of topics, including animals, food, family, and imagination. Each poem is accompanied by Silverstein's trademark black-and-white illustrations, which add to the humor and charm of the book.

Now it's at least honest



Tell me product name, description, price and image URLs for a webpage page http://books.toscrape.com/catalogue/a-light-in-the-attic_1000/index.html



I'm unable to access the internet to fetch real-time data or specific information from external websites, including the URL you provided. However, I can provide you with a template on how you can scrape product information from a webpage like the one you mentioned using Python and libraries like BeautifulSoup and requests.

You can use the following Python code as an example to scrape product information from the given URL:

It can even "browse"



G

Tell me product name, description, price and image URLs for a webpage page http://books.toscrape.com/catalogue/a-light-in-the-attic_1000/index.html



I can provide you with the information you asked for based on the content from the webpage:

- Product Name: A Light in the Attic
- Price: £51.77
- Description: It's hard to imagine a world without A Light in the Attic. This now-classic

Powered by **zyte**



What is missing?

- "Browsing" is not available via the API
- Web pages can be bigger than the context size of the model
- We also need to handle crawling and collecting all the data
- Processing every page with GPT4 can cost up to 1\$/page

ChatGPT/LLMs are a promising start:

• they can "understand" our requirements and web page text

But we need to build on top of them!



What we want

I need art books from books.toscrape.com, include book image as "mainImage"





Why we want this?

As a person who needs data:

- I have too many websites
- I have have too much on my plate
- I don't know how to extract data in a traditional way

As a company that provides data services:

- When clients have different needs, "traditional" ML is not enough
- No one wants to miss on efficiency improvements



Focus of the rest of the talk

LLM helping us do extraction from a web page for an arbitrary set of attributes:

```
{
    "is_puzzle": "is this a jigsaw puzzle",
    "number_of_pieces": int,
    "location": "location show in a puzzle",
    ...
}

{
```



What LLMs can do for us?

- Take a limited amount of text as input (usually up to 1k 10k words)
- Understand text and follow instructions
- Generate text as output
- Expensive at scale: \$1.5 to \$60 for 1M tokens (100 to 10k web pages)
- Best off-the-shelf models are available only as comercial APIs, but open-source models are catching up

How to use LLMs for Web Data Extraction

- 1. Input: HTML or text or?
- 2. Output: code or extracted data?
- 3. Model: commercial third-party or self-hosted open source?

Web page representation: HTML or text

HTML is large: ~150k tokens, much larger than 4k/16k limit for gpt-3.5 Solutions:

- cleaning HTML: 90k tokens
- extracting only the web page text from HTML: 5k tokens
- text only from the main part of the web page: 1.5k tokens Example costs with gpt-3.5:
 - 150k tokens / web page \rightarrow \$450 per 1k pages
 - > 1.5k tokens / web page \rightarrow \$3 per 1k pages

LLM generating code

Main idea: do LLM generation just once per website or layout, and efficiently run this code, without LLM invocation at runtime.

If LLM generates the full code, it must be tested or reviewed.

But we can also constrain what LLM can do, e.g. ask it to generate XPath or CSS selectors, and select from a predefined library of post-processing functions.

LLM generating selectors example

Here is the HTML of a web page: <!DOCTYPE html>

```
<!--[if lt IE 7]> ...
```

Please generate semantically meaningful CSS selectors for the following attributes:

- price
- description
- availability
- SKU

LLM generating selectors example

Here is the HTML of a web page: <!DOCTYPF html>

<!--[if lt IE 7]> ...

Please generate semantically meaningful CSS selectors for the following attributes:

- \rightarrow .product main .price color - price
- description \rightarrow #product_description + p

- SKU

- availability \rightarrow .product main .instock.availability
 - → table.table.table.striped th:contains('UPC') + td

LLM generating code pros and cons

Pros:

- fast and cheap at runtime
- we can verify if generated code makes sense

Cons:

- challenging to implement on large HTML and multiple layouts
- need to be updated when a website changes
- selectors won't handle all use-cases

Limitations of selectors



LLM outputting extracted data

LLM is invoked for every page we need to handle, e.g.:

Below is a web page text of product:

Please extract the following attributes:

- is_puzzle: is this a jigsaw puzzle
- number_of_pieces

...

location: location show in a puzzle

LLM outputting extracted data pros and cons

Pros:

- easy to implement, stateless
- can handle complex use-cases which are not possible with selectors or a human-written spider (e.g. product color mentioned in a description, summarization, etc.)

Cons:

- expensive to run, e.g. \$1 \$5 per 1K pages with gpt-3.5
- harder to QA and control

Self-hosted models pros and cons

Pros:

- can be used with more sensitive data
- can be cheaper to run
- can be more reliable and faster to respond

Self-hosted models pros and cons

Pros:

- can be used with more sensitive data
- can be cheaper to run
- can be more reliable and faster to respond

Cons:

- yes they "can" but this is all non-trivial
- are not yet at GPT-4 quality level, so not suitable for some tasks

Commercial or Open Source models?

My recommendation:

- start by prototyping with a quality commercial model (e.g. OpenAI) to validate the idea and establish required model quality level
- open-source models can reach gpt-3.5-turbo quality with task-specific fine-tuning and lower costs at high enough volume

LLMs usage and research at Zyte

- Using OpenAI APIs for a few custom projects, after extensive legal review together with the customers legal teams
- Experimenting with generating selectors and spider code
- Experimenting with in-house models which can do attribute extraction at runtime

DEMO

https://llm-extraction-demo.zyte.com

Try it out!

https://llm-extraction-demo.zyte.com extract-summit extract-summit-2023

Available to try until Oct 30th



Powered by **zyte**





Thank you!