

Web Analyzer Tool

Overview

It is designed to streamline competitor analysis by automating data collection and analysis. It extracts primary keywords, analyzes search performance, audits content, gathers traffic metrics, and evaluates call-to-action (CTA) strategies.

Features

1. **Extract Primary Keywords** - Uses web scraping to retrieve relevant keywords from competitor websites.
2. **Analyze Search Performance** - Gathers top-ranking sites for specific keywords.
3. **Perform Content Audits** - Analyzes blog posts, backlink profiles, and content structure.
4. **Gather Traffic Metrics** - Estimates visitor statistics and engagement.
5. **Evaluate CTA Strategies** - Identifies call-to-action placements and effectiveness.

Working

The script follows these steps:

1. **Initialize Search Providers** - Loads Google and DuckDuckGo search providers from [search_providers.py](#).
2. **Extract Keywords** - Uses web scraping (via [requests](#) and [BeautifulSoup](#)) and Selenium (for JavaScript-rendered content) to extract primary keywords from the competitor's site.
3. **Analyze Search Performance** - Searches for extracted keywords on supported search engines, retrieves top-ranking websites, and stores relevant details.
4. **Perform Content Audits** - Scrapes the competitor's website to analyze blog structure, backlink profiles, and on-page SEO elements.
5. **Gather Traffic Metrics** - Estimates visitor traffic based on content structure, links, and historical trends.
6. **Evaluate CTA Strategy** - Detects call-to-action elements like signup buttons, CTAs, and promotional links.
7. **Store Results** - Saves extracted data to [web_analyzer_20250216_221002.json](#) for further analysis.

Installation and Dependencies

Prerequisites

- Python 3.12+

- Google Chrome (for Selenium WebDriver)
- Required Python packages:

```
Python  
pip install requests beautifulsoup4 selenium pandas
```

Setting Up Selenium WebDriver

Ensure that **Google Chrome** and the **Chrome WebDriver** are installed and accessible in your system PATH.

Running the Script

Step 1: Configure Search Providers

The script supports Google and DuckDuckGo search. Ensure [search_providers.py](#) is correctly configured.

Step 2: Execute the Script

Run the script from the command line:

```
Python  
python main.py
```

Step 3: Provide Input URL

The script prompts for a competitor URL. You can also modify [web_analyzer_20250216_221002.json](#) to predefine input.

Step 4: View Output

The extracted data is stored in a JSON file and printed on the console.