

An Open-Source Google ReRanker: Transforming Google Results with AI Magic

Metehan Yesilyurt

AI Search & Information Retrieval Researcher · metehan.ai

Published: November 19, 2024 ·

Canonical: <https://metehan.ai/blog/an-open-source-google-reranker-transforming-google-results-with-ai-magic/>

*I am a seasoned SEO professional and developer passionate about leveraging advanced AI technologies to solve complex challenges in digital marketing. Drawing on years of experience in search engine optimization, I developed a basic Google ReRanker Tool, an open-source application designed to analyze and re-rank Google search results based on multiple advanced factors like relevance, content quality, and domain authority. *You can fork it on [GitHub](#)^[1] and adjust any function and weight. It's open-source, you can use it anywhere, any time.**

The **Google ReRanker Tool** is an open-source Python-based application designed to help SEOs analyze and re-rank Google search results based on advanced factors like relevance, content quality, domain authority, and user intent. This tool leverages **machine learning models** (like BERT), **NLP processing** (via spaCy), and **custom ranking algorithms** to provide actionable insights for better search result evaluation. I'm using VSCode, this script works in your terminal and saves the result in a txt file.

This guide explains:

1. **How the tool works** (technical insights).
2. **How to use it effectively** (step-by-step instructions).
3. **Planned future improvements** for expanded functionality.

How Does It Work?

The tool functions in three primary steps:

1. Fetching Search Results

The tool queries Google Search for a user-provided query and extracts the following:

- **Title:** The page title of each result.
- **URL:** The link to the result.
- **Snippet:** A short summary of the page content, as shown in Google Search.
- **Position:** The result's initial ranking in Google SERPs.

Key Component: `get_search_results()` uses the `requests` library to make a search query and parses results using `BeautifulSoup`. It extracts the titles, URLs, and snippets for further analysis.

2. Analyzing Results

Each result is evaluated using multiple factors, which are calculated through various algorithms:

3. Re-ranking Results

The tool combines all factor scores into a **weighted total score**. Weights are preconfigured but can be adjusted in the `_initialize_weights()` method.

Each result is re-ranked based on its total score, providing insights into which pages deserve top positions.

Step-by-Step Usage Guide

1. Requirements

Before using the tool, ensure your environment meets the following requirements:

Python 3.8+

Required Python libraries:

TEXT

```
`pip install numpy scikit-learn spacy requests beautifulsoup4 torch transformers pandas`
```

NLP Model: Download the spaCy `en_core_web_sm` model:

TEXT

```
`python -m spacy download en_core_web_sm`
```

2. Running the Tool

1. Save the code as `google_reranker.py`.
2. Open a terminal and run the script: `python google_reranker.py`
3. Enter a search query when prompted: `Enter search query: best smartphones 2024`

3. Reviewing Results

Once analysis is complete:

The top results are displayed in the terminal:

TEXT

```
Total search results: 10
Top 3 results:
1. Title of Result 1 (Score: 0.91)
2. Title of Result 2 (Score: 0.87)
3. Title of Result 3 (Score: 0.85)
```

A detailed re-ranking report is saved as `rerank_analysis.txt`.

4. Analyzing the Report

The `rerank_analysis.txt` file contains:

Original Search Results with positions, titles, and URLs.

Re-ranked Results with detailed scores for each factor:

TEXT

```
1. Result Title (Original Position: 2)
  URL: https://example.com
  Snippet: This is a sample snippet.
  Total Score: 0.92

  Score Details:
  - relevance: 0.95
  - content_quality: 0.87
  - domain_authority: 0.93
  - intent_match: 0.89
  - freshness: 0.70
  - technical_seo: 0.88
  - user_signals: 0.76
```

Use this report to identify high-value pages that Google might have overlooked or poorly ranked.

Who Should Use This Tool?

- **SEO Professionals:** Analyze SERP results and optimize web pages for ranking factors.
- **Content Marketers:** Understand content gaps and improve relevancy.
- **Developers:** Customize weights and integrate with other SEO tools for automation.
- **Data Scientists:** Extend functionality for deeper analytics or integrate with AI-driven systems.

Customizing the Tool

1. Adjusting Factor Weights

The `_initialize_weights()` the method allows customization of factor importance:

TEXT

```
return {
    'relevance_score': 0.30, # Increase relevance weight
    'content_quality': 0.20,
    'domain_authority': 0.10,
    'user_intent_match': 0.15,
    'freshness': 0.10,
    'technical_seo': 0.10,
    'user_signals': 0.05
}
```

2. Extending Functionality

- **Fetch Full Page Content:** Add logic to scrape and evaluate full-page text.
- **Integrate Real User Metrics:** Use Google Analytics or custom engagement data for accurate user signals.
- **Refine Models:** Experiment with advanced embeddings like **OpenAI's GPT** for better context understanding.

3. Debugging Tips

If Google blocks requests:

- Use rotating proxies or CAPTCHA solvers for scraping.
- Implement exponential backoff for repeated requests.

Planned Improvements

1. **Fetch Full Page Content:** Analyze beyond snippets for better quality scoring.
2. **Rich Data Integration:** Use APIs like Ahrefs or Moz for real-time domain authority and backlinks data.
3. **Keyword Optimization Suggestions:** Recommend actions based on analyzed gaps.
4. **Interactive Dashboard:** Build a web-based GUI for better usability.

Conclusion

The Google ReRanker is a robust, extensible framework for SEOs looking to analyze and understand Google's ranking patterns. With room for further development and customization, it provides a valuable starting point for building advanced SEO tools.

Once again; here is the GitHub repo URL; <https://github.com/metehan777/google-rerank-tool/>^[1]

I also want to share a useful link here; <https://medium.com/google-cloud/reranking-3b5f351cb398>

Open-source rerankers are the cheapest AEO test bench. They mimic the reranking stage that decides AI citations in ChatGPT, Perplexity and Gemini. Answer engine optimization experiments scale when reranking is local.

REFERENCES & SOURCES

[1] fork it on GitHub — <https://github.com/metehan777/google-rerank-tool/>