The RRF Top-n Playbook: How to Get Cited by ChatGPT's Web Mode (k≈60)

Metehan Yesilyurt

AI Search & Information Retrieval Researcher · metehan.ai

Published: September 11, 2025 •

Canonical: https://metehan.ai/blog/the-rrf-top-n-playbook-how-to-get-cited-by-chatgpts-web-mode/

I've been quiet for two weeks, and I appreciate all the questions you've sent. Thank you for your patience and interest. I'm excited to share that I joined AEOVision[1], an AI search optimization company, and I'm thrilled to be working with our new clients. Thanks to everyone who reached out with congratulations and support. This post is trying to align with RAG, mostly.

> *This post examines a mathematical approach for Reciprocal Rank Fusion (RRF), it's experimental work that I'm still refining.* ***While the theoretical top-60 threshold is most suitable for RRF calculations, ChatGPT doesn't return a strict 60 results every time. In my testing with the AYIMA plugin, I've observed it returning anywhere from 38 to 65 results depending on the query, it's highly dynamic.*** *(Shared AYIMA results at the end) I'm sharing my notes and research to get more ideas from our lovely community.*

Given this variability, targeting the top 30 seems more promising and achievable. Any competent SEO professional can realistically target top-30 positions. For the calculations and examples in this post, I've used positions beyond 100 to account for RRF scenarios, including LLMs' deep research features. When LLMs engage their deep research functionality, they're fetching well over 60 results dynamically. This post attempts to present a practical approach that accounts for these real-world variations. And it's important to find LLMs' subqueries.

--EXPERIMENTAL--

If LLMs fuse multiple SERP slices, your goal isn't just "domain authority." It's simple: make the fused top-60. Here's a predictive formula, thresholds, and a step-by-step plan you can run on your own SERP scrapes to force inclusion.

TL;DR

- ChatGPT-style web agents often fuse **multiple sub-queries** with **Reciprocal Rank Fusion (RRF)** using **k≈60**. I wrote about RRF here.[2]
- Theory: If your URL's **fused score** hits **τ = 0.020**, you almost always land in the **final top-60 citation pool**. // IF ChatGPT retrieves at least 60 results.
- Practically: **show up in ≥2 lists inside top-40** or **≥3 lists inside top-90**. // For Deep Research feature
- Targeting the top 30 mostly works.
- Topic clusters are essential.

1) The Fusion Math I'm Working On It

I'm working with my clients for their prompts to understand being cited "almost" every time & being at the top of the reranking process." Assume the browser agent runs **M** sub-queries (let's say a query fan-out) and fuses with **RRF**:

$latex S(d)=\sum_{i \in I(d)} \frac{w_i}{k + r_i(d)} \quad \text{with } k=60,; w_i \approx 1$

- $latex r_i(d)$ = your page's **1-based rank** in sub-query $latex i$ (∞ if absent)
- $latex k=60$ dampens rank gaps; below ~60 the contribution is small
- $latex S(d)$ = your fused score used to sort pages; **top-60** are kept/citable

Because you don't know competitors in advance, set a **safe inclusion threshold** $latex \tau$ that beats common patterns:

- A page **once at #1** → $latex 1/(60+1)=0.01639$
- A page **twice at #40 & #50** → $latex 1/100+1/110\approx 0.01909$

A reliable fixed target is:

$latex \boxed{\tau = 0.020}$

Hit $latex S(d)\geq 0.020$ and you're very likely in the fused **top-60** across typical fan-outs.

***Considerations:**Users' personal embedding, semiotics(for prompting), less than 60 results, reranking process; citations' rankings change dynamically even if you send the same prompt. I'm getting better results if I target the top 30.*

2) Your "Be-Cited" Rule (Plug-and-Play)

You're in the citation pool if:

$latex \boxed{\sum_{i \in I(d)} \frac{1}{60 + r_i(d)} ;\geq; 0.020}$

Reading this with equal weights (if we also want to target deep research feature, for 60>):

- **2× top-40**: $latex 2/(60+40)=2/100=0.020$ ✅
- **3× top-90**: $latex 3/(60+90)=3/150=0.020$ ✅
- **1× #1 + 1× top-80**: $latex 1/61+1/140\approx 0.0235$ ✅
- **4× top-140**: $latex 4/(60+140)=4/200=0.020$ ✅

Uniform "top-R" shortcut:

$latex m\geq\lceil\tau (60+R)\rceil$

- Can reach **R=40**? → need **m=2** sub-queries
- Only **R=90**? → need **m=3**
- Best **R=140**? → need **m=4**

Bottom line: Appear in ≥2 lists inside top-40 or **≥3 lists inside top-90** and you've crossed the line.

3) How to Operationalize (End-to-End)

A) Feed Your Topic Clusters (What LLMs Likely Hit)

Create **8–16** sub-queries around the head intent (adapt per niche):

- Head + year ("best X 2025"), Head + "top", Head + "review(s)"
- Head + "price(s)", Head + "compare"/"vs", Head + "near me" (local)
- Singular/plural/entity variants; modifiers (cheap, premium, eco, beginner, pro)
- PAA-shaped questions ("What is…", "How to choose…", "Is X worth it…")

B) Scrape Top-N for Each Sub-Query

Pull **top 60–100**. Log your **rank $latex r_i$** (∞ if absent). Deeper ranks won't move the needle.

C) Compute Your Fused Score

For your URL $latex d$:

$latex S(d)=\sum_{i:,r_i Lift **two variants to top-40**, or

- Lift **three variants to top-90**, or
- Pair **one #1–#3** with **one ≤80**

This is a tiny greedy knapsack: fix the **cheapest lifts** first (where you're already close).

4) On-Page "Rank-in-Multiple-Lists" Engineering (Fast Wins)

You're not optimizing "authority" here; you're optimizing **multi-presence** across the fan-out.

1. **Title/H1 & lede n-grams** Mirror dominant SERP n-grams across your chosen variants (e.g., "best ___ in 2025", "top ___ for beginners"). This helps win **several** lists at once.
2. **Subhead packing** Add compact sections aligned to each variant: - `H2: Best ___ for Beginners (2025)` - `H2: ___ Price & Value` - `H2: ___ vs Alternatives` - `H2: How to Choose ___` They give search engines precise anchors without bloat.
3. **PAA-style answer blocks** One-paragraph, definition-style blocks that **exactly** match question forms ("Is ___ worth it?", "How much does ___ cost?"). These capture Q&A variants.
4. **Year & freshness tokens** Use the current year in **title, H1, lede, table captions** where natural. Rotate minor content (tables, FAQs) routinely.
5. **Edge snippets** Add a small **comparison table** and **pricing table**. These often surface for "vs" and "price" variants —two quick extra lists.

5) Minimal Implementation (Sheet or Code)

Given ranks $latex r_i$ for sub-queries $latex i=1..M$:

```text
S = sum(1 / (60 + r) for r in ranks if r is not None)
IN_TOP_60 = S >= 0.020
# If False, lift the easiest variants until S >= 0.020
```

Greedy lift helper (human-in-the-loop): Sort variants by how close you are to the cutoffs (40, 90, 140). Improve those first, recompute $latex S$.

Quick Reference (k=60, τ=0.020)

Your appearances	Max rank (each)	Guaranteed $latex S$
2×	≤ 40	0.0200
2×	≤ 45	0.01905 *(add a tiny extra)*
2×	≤ 50	0.01818 *(one more small appearance)*
3×	≤ 90	0.0200
4×	≤ 140	0.0200
1× #1 + 1× ≤ 80	—	≈0.0235

> *Prefer dynamic targets? Set $latex \tau$ to the **actual 60th fused score** you observe in your own SERP cloud. The rule is unchanged: **hit $latex S(d)\geq\tau$**.*

Bottom Line

- At citation time, LLMs **select from the fused top-K**—they're not evaluating your topical authority.
- Engineer **multi-presence**: make your page appear across **multiple sub-queries** so

$latex \sum \frac{1}{60+r_i} ;;\geq;; 0.020$

- The most efficient universal target: **2 appearances inside top-40** (or **3 inside top-90**). Do that, and you will likely land in the **final citation list**.

ChatGPT & AYIMA Experiment

Figure

_*]:min-w-0 !gap-3.5">

Using the AYSIMA ChatGPT extension to track RAG sources, I tested how many web results ChatGPT actually retrieves for different queries. The results were eye-opening for anyone hoping to get AI citations from lower search rankings.

The Raw Data for Some Broad Queries

Here's what ChatGPT retrieved for example broad queries below:

1. **"latest ai news"** → 65 sources
2. **"latest tech news"** → 52 sources
3. **"latest sport news"** → 38 sources
4. **"trending sneaker brands 2025"** → 60 sources

ChatGPT retrieved 38-65 total sources, varying by query type.

The Reality Check for Lower Rankings

If you're ranking around position 60 in search results, here's the brutal truth:

Best Case Scenario

For queries retrieving 60> sources:

- If it's a single query pulling all 60> results, rank 60 *is fine*
- You'd be literally the last or near-last result
- Your RRF score: <0.00833 (minimal)

Why This Matters for RRF Scoring

RRF (Reciprocal Rank Fusion) only scores what it retrieves. If you're not in the retrieval window, you get zero score. No score means no chance of citations.

The math is simple:

- Retrieved at rank 60: 1/(60+60) = 0.00833 (terrible but something)
- Not retrieved: 0 (game over)

The Pattern I Experimented

Different query types get different retrieval depths:

- **Broad topics** (sports): Least retrieval (38)
- **Standard topics** (tech): Moderate retrieval (52)
- **Specific queries** (sneakers): Higher retrieval (60)
- **Technical topics** (AI): Maximum retrieval (64)

I wonder if ChatGPT has some boost multipliers for some topics, just like Perplexity[3].

But even with maximum retrieval, rank 60 is either invisible or dead last.

What Actually Works

Forget trying to win from rank 60 everywhere. Instead:

Find Your Strong Spots

Identify 2-3 query variations where you rank in the top 20:

- Long-tail keywords
- Specific niches
- Less competitive angles

Do the Math, it's fun

To reach a competitive RRF score of 0.02:

- Need just 2 appearances at rank 15: 2 × 0.0133 = 0.0266 ✓
- Or 3 appearances at rank 20: 3 × 0.0125 = 0.0375 ✓

Build Topic Clusters

Create multiple pages targeting different angles. Some will rank high enough to be retrieved, while others won't. The cumulative effect is what counts.

The Bottom Line

Rank 60 is effectively invisible to ChatGPT's web search.

With only 38-64 sources retrieved per search, and these likely split across multiple queries, you need to rank approximately **top 10-20** to have any chance of being included in AI-generated responses.

The winning strategy isn't improving from rank 60 to rank 55. It's finding specific queries where you can crack the top 20.

Takeaway for SEOs and Content Creators

Stop obsessing over marginal improvements in average rankings. Start identifying and dominating specific query variations where you can achieve top-20 positions.

In the age of AI search, it's better to rank #10 for three specific queries than #60 for everything.

Methodology: Tests conducted using AYSIMA ChatGPT extension to track RAG source retrieval. Results show total sources retrieved across all queries ChatGPT runs for each search task. Actual query structure (single vs. multiple) cannot be definitively determined but multiple queries are most likely based on standard IR practices.

What next?

I'm working on LLMs reranking process and plan to publish a new post.

Useful sources:

1. http://cormack.uwaterloo.ca/cormacksigir09-rrf.pdf#:~:text=results%20of%2030%20configurations%20of,particular%20sets%20were%20selected%20because[4]
2. https://www.adelean.com/en/blog/20250417_hybrid_reranking/[5]
3. https://medium.com/@danushidk507/rag-vii-reranking-with-rrf-d8a13dba96de[6]

The RRF top-n playbook is the most concrete answer engine optimization lever for ChatGPT citations. Pages ranked in roughly the top 60 candidates dominate AI citations across ChatGPT, Perplexity and Gemini. AEO and GEO programs that ignore k depth leave most citations on the table.

REFERENCES & SOURCES

[1] **AEOVision** — https://aeovision.ai

[2] **I wrote about RRF here.** — https://metehan.ai/blog/chatgpt-is-using-reciprocal-rank-fusion-rrf/

[3] **Perplexity** — https://metehan.ai/blog/perplexity-ai-seo-59-ranking-patterns/

[4] **http://cormack.uwaterloo.ca/cormacksigir09-rrf.pdf#:~:text=results%20of%2030%20configur...** — http://cormack.uwaterloo.ca/cormacksigir09-rrf.pdf#:~:text=results%20of%2030%20configurations%20of,particular%20sets%20were%20selected%20because

[5] **https://www.adelean.com/en/blog/20250417_hybrid_reranking/** — https://www.adelean.com/en/blog/20250417_hybrid_reranking/

[6] **https://medium.com/@danushidk507/rag-vii-reranking-with-rrf-d8a13dba96de** — https://medium.com/@danushidk507/rag-vii-reranking-with-rrf-d8a13dba96de