Tuning a search pool: content beat weights

On a Trawl pool of 18 public repos, every search-time setting I tried scored exactly the same, because the reranker sets the final order. What moved the scores was fixing what was in the index, measured with 15 questions whose answers I knew.



Published
3 November 2026

Reading time
3 minutes

About
Mullet

Tags
Search, AI

More on: Mullet

If every weight setting gives the same score, stop tuning weights and fix what's in the index. Three rounds of tuning a Trawl pool showed exactly that: none of the search-time settings I tried changed the score at all, and changes to the content took MRR from 0.489 to 0.542 and hit@10 from 0.733 to 0.867.

Trawl is Mullet's search. It grew out of Trove, and a pool is a collection of documents and code it indexes and searches by keyword and by meaning together.

The pool and the test

The pool holds 18 public repos of tools for running several Claude Code accounts, plus Mullet's own code. To score it I wrote 15 questions with known answers. Each answer is a piece of text that only appears where a tool actually implements the thing asked about, so a hit means the search found the code, not a page that mentions it.

Each round gets four figures. MRR is the average of 1 divided by the rank of the first right answer, so 1.0 means every answer came first. hit@1, hit@5 and hit@10 are the share of questions with a right answer in the top 1, 5 or 10.

Round	Change	MRR	hit@1	hit@5	hit@10
1	Wizard config plus exclusions	0.489	0.333	0.733	0.733
2	Missing languages indexed; UI, CSS, tests, site pages and design notes out	0.533	0.400	0.667	0.800
3	Synonyms; two test questions corrected	0.542	0.400	0.667	0.867

Round 1: the weights changed nothing

The first round used the config Mullet's setup wizard writes, plus exclusions. Then I tried these search-time variants, and every one scored exactly the same as round 1:

- reciprocal rank fusion (RRF) instead of weighted merging
- BM25-heavy, keyword weighted above vector
- vector-heavy, vector weighted above keyword
- code priority off
- MMR off

The reranker sets the final order. The weights only shape the list it reads, so on this pool they made no difference to the score.

Round 2: fix what's in the index

About half the tools had none of their code in the index. Trawl's code indexer only parses TypeScript, Python, JavaScript, CSS, Rust, YAML, JSON and HTML, and ignores the `code.extensions` setting, so only the Go, shell and Swift tools' docs were in the index until those extensions were listed under documents instead. Three other changes went in the same round:

- Two repos made up 45% of all the chunks, 17,844 of 39,666, mostly UI code, styles and tests. Their UI code, CSS, tests and site pages came out.
- The pool's own design notes came out too. They matched the questions' wording too closely and pushed real code down the results.
- One tool's main script has no file extension, so the pool links to it under a `.sh` name.

MRR rose to 0.533 and hit@10 to 0.800. hit@5 fell, from 0.733 to 0.667: one answer fewer in the top 5, with more of the tools' code now competing for those places. The index was 36,924 chunks after this round.

Round 3: synonyms, and the test itself

Search synonyms for handoff, workstate, fallback and login. And two test questions were corrected, because the code they pointed at had changed since they were written.

That gave MRR 0.542 and hit@10 0.867. Two questions still miss the top 10. I accepted that: for one, the right files are indexed but outranked, and for the other, another tool's docs answer the question instead.

What I'd do again

- Write the questions first, with answers you can check as plain text, and re-score after every change.
- When every weight setting scores the same, stop there. Look at what's in the index: what's missing, what's noise, and what echoes your own questions.
- Check that every language in the pool is actually being indexed.