ON-PAGE.AI RESEARCH

The Best and Worst Entity Models for SEO: Which Find the Words Google Uses, and Which Feed You False Positives

Eric Lancheres · On-Page.ai Research | July 20, 2026

ABSTRACT

Entity optimization advice is only as good as the entity extractor behind it, and tools differ enormously in what they extract. We ran five extractors over the same 166 page-one ranking pages across 48 SERPs: our own production model (disclosed: we sell it), Google Cloud Natural Language, TextRazor, a GPT-5-series budget LLM (gpt-5-mini) prompted the way inexpensive tooling typically does it, and spaCy's standard English pipeline. The referee is external to every contestant: Google's own words for each query, taken from the AI Overview, People Also Ask, and related searches captured with the SERPs. Every model is scored on the two axes a practitioner cares about: accuracy, does it find the important words, the ones Google itself uses for the query, and false positives, does it hand you words nothing in the query's ecosystem supports. The five models disagree profoundly. Median entities per page ranged from 15 (spaCy) to 172 (Google Cloud NL). Of each model's top-30 entities per page, 48.6% of ours appeared in Google's words, against 33.7% for Google Cloud NL, 21.2% for TextRazor, 17.8% for gpt-5-mini, and 16.4% for spaCy. Between 46.6% and 65.1% of what the four generic extractors surfaced was false positives, entities appearing in neither Google's words nor any competing ranking page, against 13.4% for our model. At matched top-10 size, recommendation lists built from each extractor hit Google's words at 64.2% (ours), 61.5% (Google Cloud NL), 53.4% (gpt-5-mini), 49.7% (spaCy, producible on only 71 of 166 pages), and 40.0% (TextRazor). The LLM was additionally unstable: identical runs on identical text overlapped only 61%. Which extractor sits under an entity-SEO tool is not an implementation detail. It decides most of what the tool tells you to do.

KEY FINDINGS

- Five extractors read the same pages and produced five different realities: from 15 entities per page to 172. The spread shows on both axes that matter: accuracy (finding the words Google actually uses) and false positives (words with no support anywhere).
- Roughly half to two-thirds of what generic extractors surface is false positives: entities that appear in neither Google's words for the query nor any competing ranking page. That is optimization effort pointed at nothing.
- A naive budget-LLM extractor is the least Google-aligned of the five and does not agree with itself: two in five of its entities change between identical runs.
- Disclosure up front: one of the five models is ours, and it wins these referees, including against Google Cloud Natural Language. The referees are external and the method is described fully enough to check.

1. Background

“Cover the entities Google expects” has become standard advice in semantic SEO, and a small industry of tools now surfaces entity lists and entity gaps for any page. Underneath that advice sits an unexamined dependency: some model has to decide what the entities are. Tools build on general-purpose extractors (Google Cloud Natural Language, TextRazor, open-source NER) or increasingly on LLM prompting, and the resulting lists are treated as interchangeable ground truth.

They are not interchangeable, and this study measures how far apart they sit. It is also, plainly, a vendor study: one of the five contestants is the entity model behind our own products, and we publish the comparison because it wins. Two design choices keep the comparison honest. The referee is external to every contestant: not similarity to any model, but presence in what Google itself writes about each query. And every model is scored by one identical procedure, described in full below, so the test can be rerun against us [1].

2. Method

Population. From the 50-keyword, ten-vertical SERP capture used in our earlier studies [1] (July 15, 2026; US, English), we crawled the page-one organic results and kept the 275 pages that yielded clean article text; 166 pages across 48 SERPs were processable by all five models and form the comparison set.

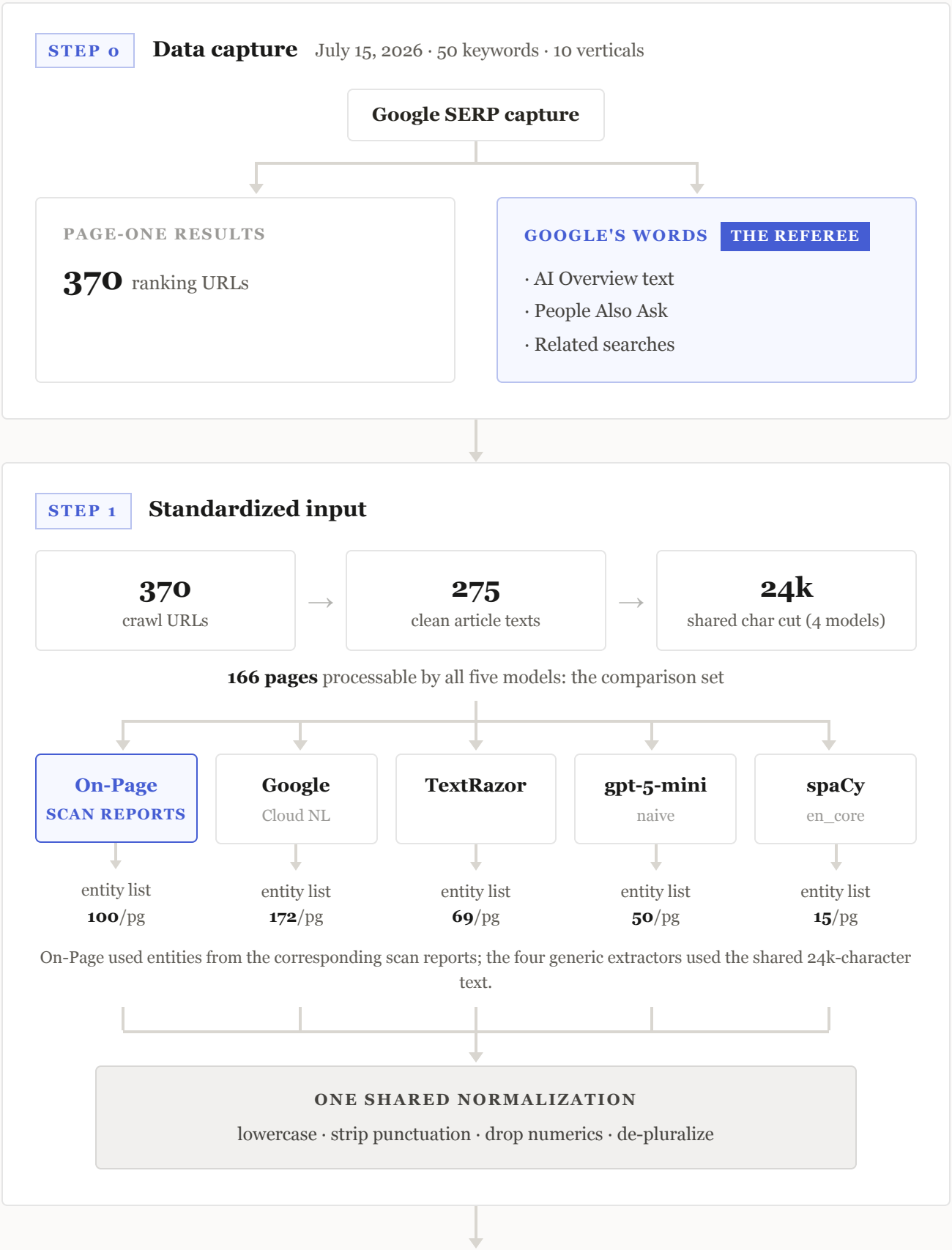
Contestants. Four generic extractors ran on the same locally extracted text, truncated identically for every model: Google Cloud Natural Language's entity analysis [3], TextRazor's entity extractor [4], spaCy's en_core_web_sm English pipeline [5], and OpenAI's gpt-5-mini [6] given a single naive instruction to list the entities in the text, which is how inexpensive tooling typically wires an LLM. Our own model's entities come from the same scan reports our customers see, filtered to entities the report marks as present on the page; its internals are not described here beyond that, and this asymmetry in text source is noted in §5.

Referee. For each keyword we assembled Google's own query-side words from the same capture: the AI Overview's full text (shown on 49 of 50 SERPs) [2], every People Also Ask question, and every related search. An entity “appears in Google's words” when its normalized form occurs in that text. One normalization applies to every model: lowercasing, whitespace and edge punctuation cleanup, removal of purely numeric strings and strings under three characters, and crude depluralization. Matching is deliberately surface-level and identical for all contestants.

Measures. Extraction quality: the share of a model's entities (full set, and top-30 to neutralize verbosity differences) appearing in Google's words, and the false-positive rate, the share appearing in neither Google's words nor the text of any competing ranking page for the keyword. Recommendations: for our model, the entities its report recommends adding; for the others, the strongest available simulation of a gap analysis, entities the model finds on at least two other ranking pages but not on yours, ranked by cohort frequency; both cut to top-10 so list sizes match. Stability: gpt-5-mini was run three times on identical text for 50 pages. We also tested a rank-prediction referee (does entity coverage separate positions within a SERP); it was weak for every model, consistent with our earlier finding that little separates positions within page one [1], and we exclude it rather than publish a measure no model passes.

How accuracy and false positives were measured

One capture, five extractors, one external referee, identical rules for every model.



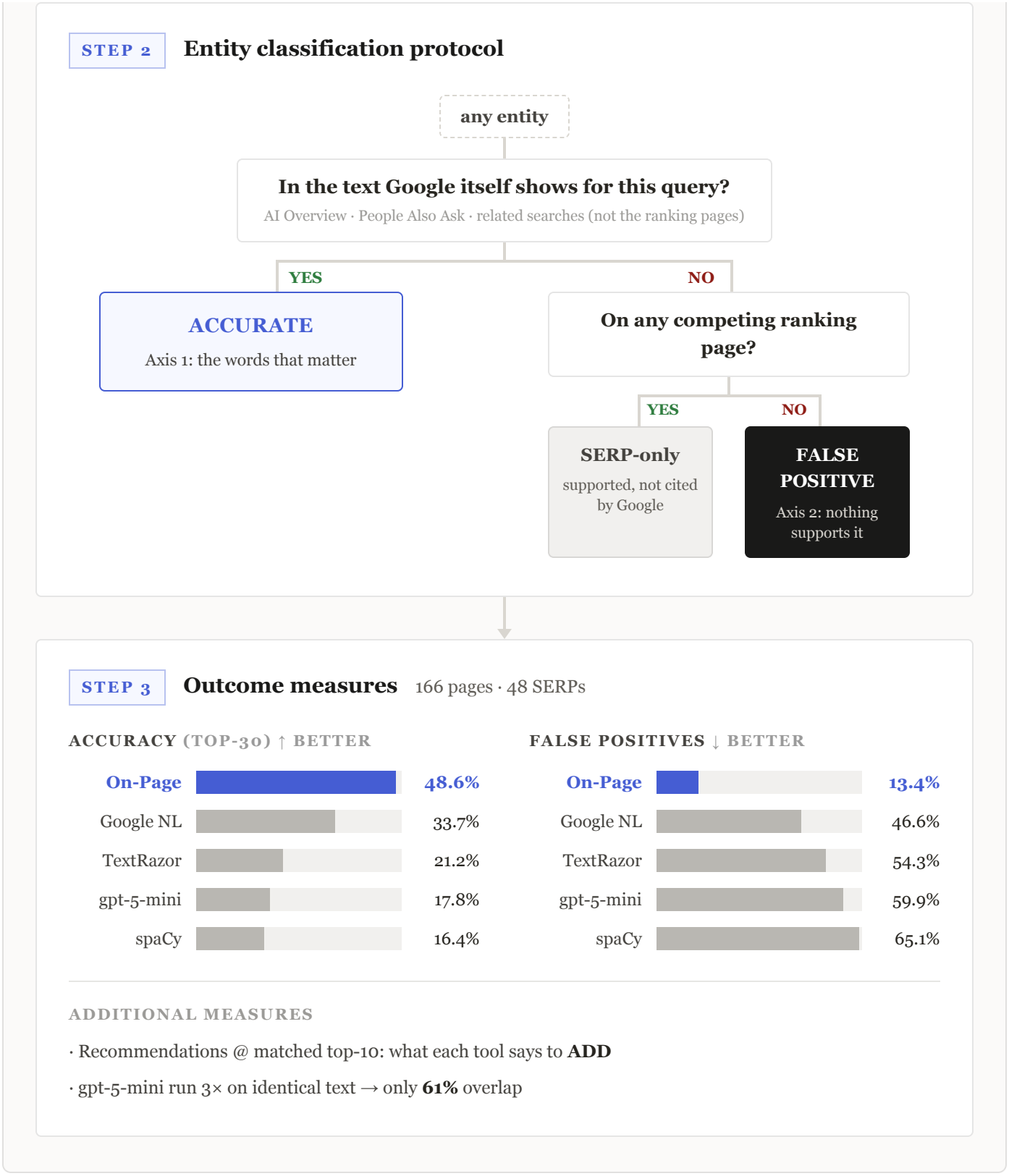


Figure 1. Study design. A single capture supplies both the pages and the referee, so there is no time drift between what ranked and what Google said. All five models are compared on the same page set under identical scoring and normalization rules, and an entity is only counted as a false positive after failing two checks: absent from Google's words and absent from every competing ranking page.

3. Results

Everything below scores the five models on those two axes. Accuracy is what you want more of: an extractor that surfaces the words that matter, the vocabulary Google itself uses when it answers the query. False positives are what you want less of: plausible-looking words with no support anywhere in the query's ecosystem, the ones that send optimization effort down the wrong path.

3.1 Accuracy: which models find the words that matter

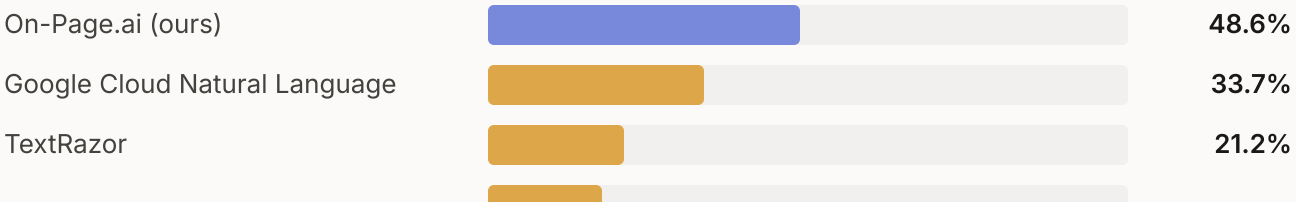
The five models produce very different amounts of material: median entities per page run from 15 (spaCy) through 50 (gpt-5-mini), 69 (TextRazor), and 100 (ours, page-present entities) to 172 (Google Cloud NL). Volume alone says nothing about quality, so the primary comparison uses each model's top 30 entities per page.

Model	Ents/page (median)	In Google's words, all (%)	Top-30 (%)	False positives (%)
On-Page.ai (ours)	**100**	**37.2**	**48.6**	**13.4**
Google Cloud Natural Language	172	19.1	33.7	46.6
TextRazor	69	18.7	21.2	54.3
gpt-5-mini (naive prompt)	50	17.1	17.8	59.9
spaCy (en_core_web_sm)	15	16.1	16.4	65.1

Table 1. Extraction quality on 166 common pages. Hit rates are the mean per-page share of entities appearing in Google's query-side words (AI Overview, People Also Ask, related searches). False-positive rate: share appearing in neither Google's words nor any competing ranking page's text.

One extractor's top entities track Google's own words. The rest don't.

For each page, we took each model's top 30 entities and checked how many appear in what Google itself wrote for that query: the AI Overview, People Also Ask, and related searches.



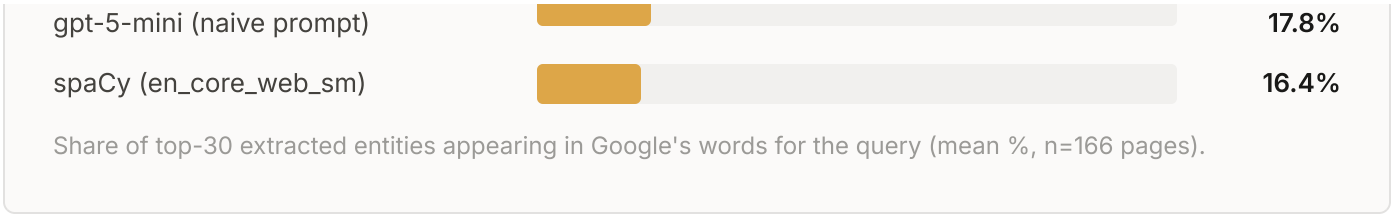


Figure 2. Mean share of each model's top-30 entities that appear in Google's own query-side text, per page, across 166 ranking pages and 48 SERPs. Matching is surface-level after one shared normalization (Method). Blue: our model. Amber: the four generic extractors.

Two things stand out. First, the spread is not subtle: 48.6% versus 16.4% at the top of each model's list is the difference between an entity list that mostly reflects how Google talks about the topic and one that mostly does not. Second, the ordering is not simply “bigger list wins”: Google Cloud NL extracts the most and lands second; spaCy extracts the least and lands last; the LLM sits in between on volume and near the bottom on alignment.

Accuracy, seen directly: how much of Google's answer each model's vocabulary covers

Google's AI Overview for “how long to smoke a brisket” (excerpt), with every word that appears in each model's entity list for one ranking page highlighted. An accurate model reads a page and comes back speaking Google's language.

On-Page.ai (ours)

Smoking a brisket typically takes 1 to 1.5 hours per pound at 225°F to 250°F. A standard 12-to-15-pound packer brisket will take roughly 10 to 15 hours to cook, plus an additional 1 to 4 hours to rest. The Breakdown of Your Cook: The Smoke (Internal Temp to 160°F) typically takes 6 hours.

Highlights: smoking, brisket, packer brisket, cook, rest, smoke, temp

spaCy (en_core_web_sm)

Smoking a brisket typically takes 1 to 1.5 hours per pound at 225°F to 250°F. A standard 12-to-15-pound packer brisket will take roughly 10 to 15 hours to cook, plus an additional 1 to 4 hours to rest. The Breakdown of Your Cook: The Smoke (Internal Temp to 160°F) typically takes 6 hours.

Highlights: smoke. Its 15-entity list touches one word of Google's answer.

Figure 3. Excerpt of Google's AI Overview for the query (captured July 15, 2026), with words covered by each model's entity list for the same ranking page highlighted (word-boundary matching, shared normalization). Our list covers 17% of the excerpt's characters; spaCy's covers 2%.

3.2 False positives: the words that lead you down the wrong path

The inverse question matters more for practitioners: how much of what an extractor surfaces has no support anywhere in the query's ecosystem? We count an entity as a false positive when it appears in neither Google's words for the query nor the text of any competing ranking page. A false positive is not necessarily wrong about the page; it is unsupported as an optimization target, because nothing Google shows for the query gives evidence it matters.

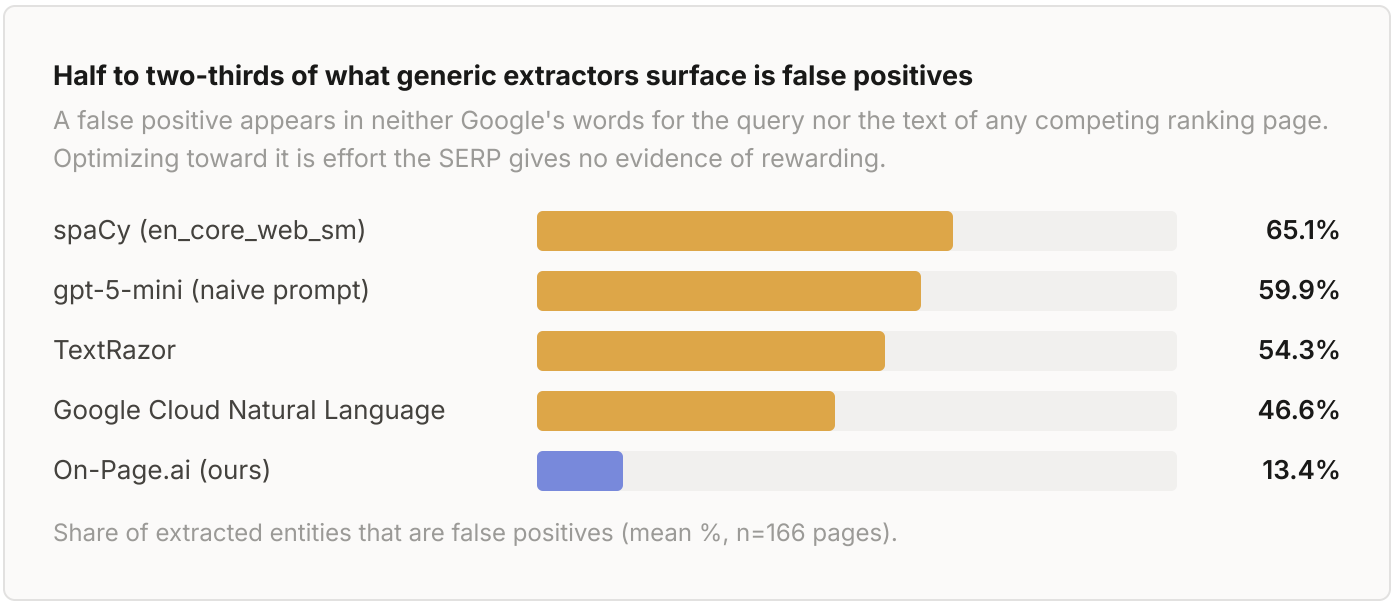
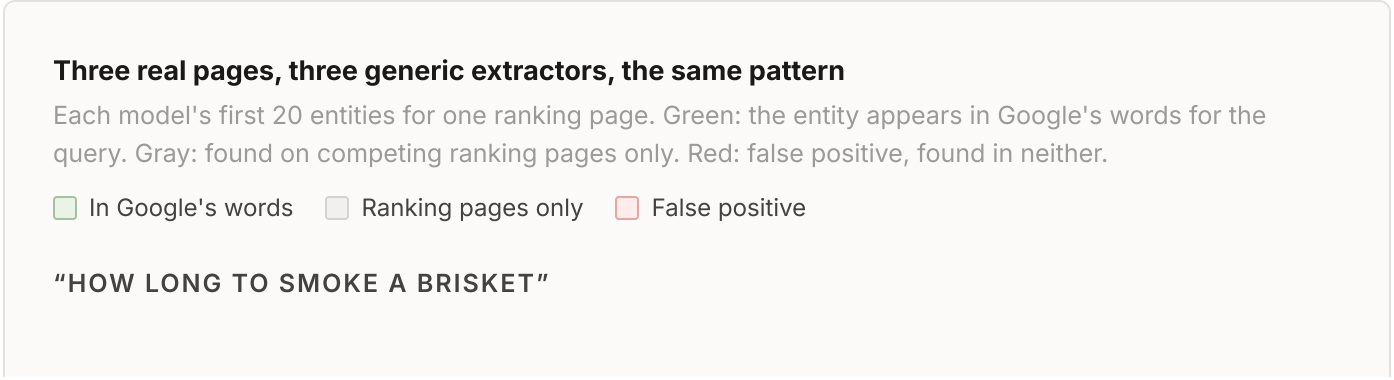


Figure 4. Mean false-positive rate per page: the share of a model's entities found in neither Google's query-side text nor any competing ranking page's content for the same keyword. spaCy 65.1%, gpt-5-mini 59.9%, TextRazor 54.3%, Google Cloud Natural Language 46.6%, our model 13.4%.

For the four generic extractors, between 46.6% and 65.1% of extracted entities were false positives. If a tool built on one of them hands you an entity checklist, roughly half of the list, or more, points at material with no footprint in the SERP you are trying to win. Our model's rate is 13.4%, which is expected behavior rather than magic: it is built to extract entities relevant to search cohorts, and this referee measures exactly that. The practical point is not that generic extractors are bad NLP; it is that they answer a different question than the one entity SEO asks.



gpt-5-mini (naive prompt)

30%
in Google's words (6/20)

55%
false positives (11/20)

Smoking brisket | Brisket | Smoked brisket | Texas style | American BBQ | Low-and-slow | Texas Crutch | Stall (smoking stall) | Probe test | Meat thermometer | Thermometer probe | Pellet grill | Pellet smoker | Offset smoker | Electric smoker | Wood grill | Full-packer brisket | Whole-packer brisket | Brisket flat | Point cut

On-Page.ai (ours)

75%
in Google's words (15/20)

0%
false positives (0/20)

brisket | temperature | smoker | butcher paper | process | smoking | flavor | moisture | recipe | thermometer | seasoning | beef brisket | black pepper | juices | results | meat thermometer | pellet grill | pepper | cutting board | method

The LLM's false positives look plausible. That is what makes them expensive: they read like sensible optimization targets, and nothing in the query's ecosystem supports them.

"BEST MAGNESIUM SUPPLEMENT"

TextRazor

30%
in Google's words (6/20)

50%
false positives (10/20)

2026-07-22T00:00:00.000+00:00 | Bioavailability | Digestion | Stomach | Information | Magnesium | Citric acid | Magnesium (medication) | Magnesium aspartate | Magnesium glycinate | Health | Disease | Psychological stress | Average | Amazon (company) | Keyboard shortcut | Ritual | Stress (biology) | Racing thoughts | Energy

On-Page.ai (ours)

60%
in Google's words (12/20)

10%
false positives (2/20)

supplements | magnesium | magnesium glycinate | magnesium oxide | health | products | magnesium citrate | blood pressure | mineral | content | citrate | absorption | benefits | glycinate | function | magnesium supplement | constipation | account | bone health | magnesium levels

Here the noise is unmistakable: a raw timestamp, interface boilerplate, and Wikipedia-style disambiguations extracted as if they were topics a supplement page should cover.

"BEST CODE EDITOR FOR PYTHON"

Google Cloud Natural Language

60%
in Google's words (12/20)

15%
false positives (3/20)

Python | Visual Studio Code | VS Code | PyCharm | Sublime Text | Pieces | Cursor | Integration | GitHub | Spyder | Specialized Python | support | computing | Pricing | AI context | developers | development workflow | codebase | frameworks | streamline coding

On-Page.ai (ours)

85%
in Google's words (17/20)

0%
false positives (0/20)

python | development | editor | features | support | version | spyder | windows | interface | pycharm | debugger | debugging | visual studio code | integration | code completion | projects | beginners | code editor | thonny | vs code

The best of the generic extractors is genuinely close on named tools, and it still invents phrases no one searching this query uses. Zero false positives on our side of this page.

Figure 5. Three page-one results (left unnamed), each read by one generic extractor and by our model: the first 20 multiword or specific entities in each model's own output order, with the same readability cut applied to every panel. Full-page false-positive rates for these pages: gpt-5-mini 50%, TextRazor 63%, Google Cloud NL 21%, and ours 15%, 8%, and 6%, against sample means of 59.9%, 54.3%, 46.6%, and 13.4% (Table 1). Representative pages, not hand-picked extremes.

3.3 Recommendation quality at matched size

Extraction is the input; what a tool tells you to add is the output. We compared top-10 recommendation lists: entities a model would surface as missing from your page but present in the competitive landscape. To keep the comparison fair, competitor recommendations were constructed by the strongest method their extractors support (entities found on at least two other ranking pages, absent from yours, ranked by frequency), and all lists were cut to ten.

Model	Top-10 recs in Google's words (%)	Pages with a usable list
On-Page.ai (ours)	**64.2**	**166/166**
Google Cloud Natural Language	61.5	159/166
gpt-5-mini (naive prompt)	53.4	119/166
spaCy (en_core_web_sm)	49.7	71/166
TextRazor	40	132/166

Table 2. Top-10 recommendation lists scored by presence in Google's query-side words. "Pages" counts pages (of 166) where the model produced at least three recommendations.

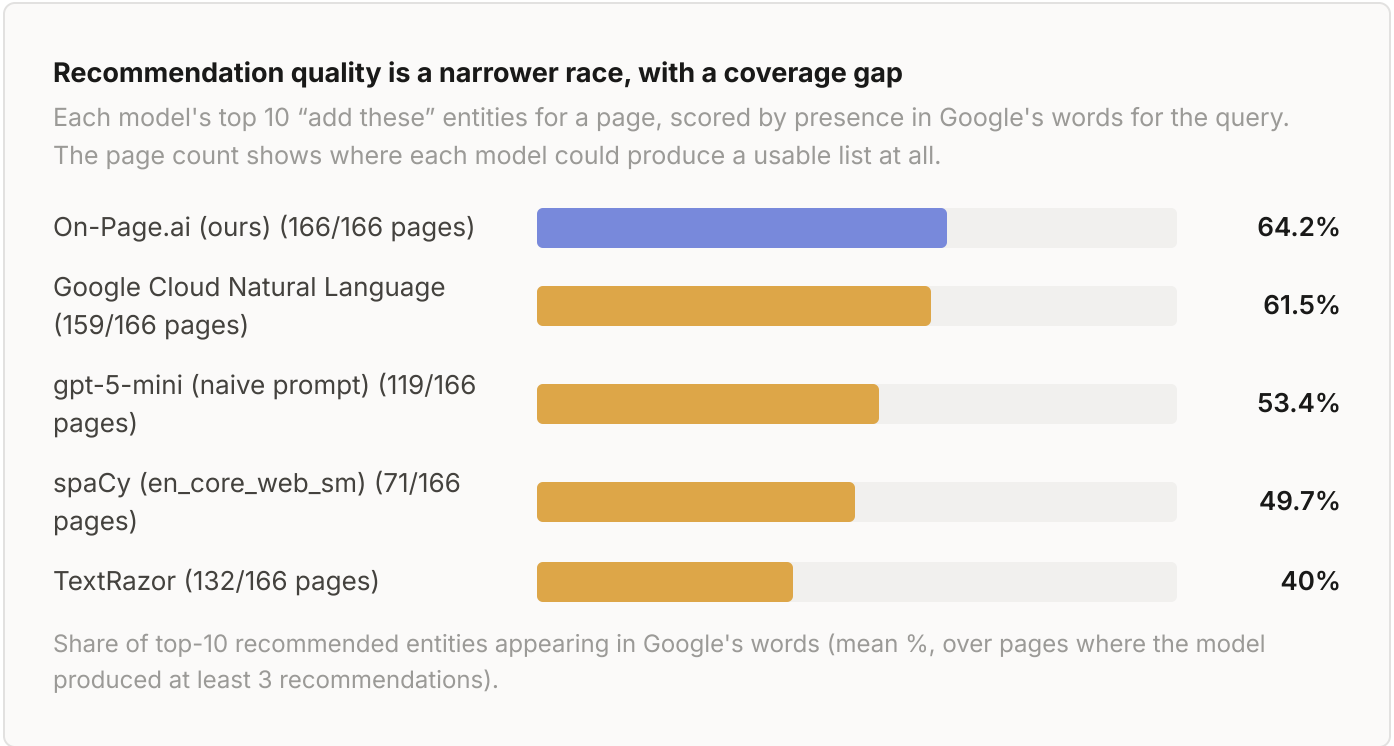


Figure 6. Top-10 recommendation lists scored against Google's query-side text. Competitor recommendations are simulated by the strongest available method: entities that model finds on at least two other ranking pages but not on yours, ranked by cohort frequency (Method). Parentheses: pages (of 166) where the model produced at least three recommendations.

This race is narrower at the top: 64.2% (ours) against 61.5% for a Google Cloud NL-based gap analysis. Two differences still separate the field. Coverage: our model produced a usable list on all 166pages, Google Cloud NL on 159, and spaCy on only 71, because a sparse extractor often cannot find enough consensus entities to recommend anything. And floor: a TextRazor-based gap analysis lands at 40.0%, meaning six of its top ten suggestions have no support in Google's own words for the query.

3.4 The LLM does not agree with itself

Determinism matters for a metric you track over time. Four of the five extractors return identical output for identical input. The budget LLM does not: across three identical runs on identical text, gpt-5-mini's entity sets overlapped an average of 61%. An entity checklist that rewrites two of every five items when you refresh it is not a stable optimization target, and any score built on it will move without the page changing.

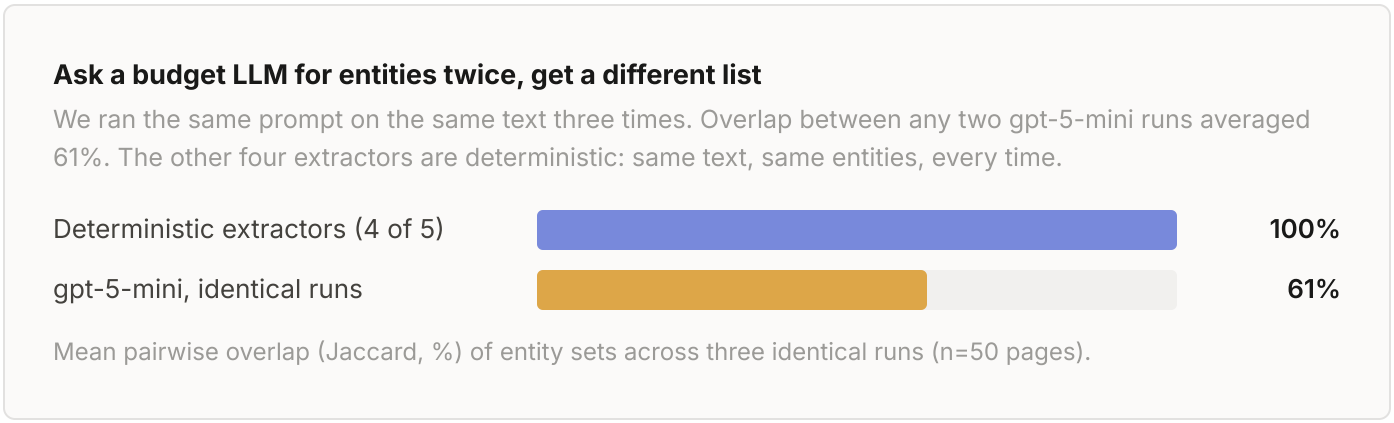


Figure 7. Run-to-run stability. gpt-5-mini's entity sets overlap 61% on average between identical runs on identical text: roughly two in five entities change if you simply ask again. Our model, Google Cloud Natural Language, TextRazor, and spaCy return identical output for identical input.

4. Discussion

The result is best read as a statement about fit, not about NLP quality in the abstract. Google Cloud Natural Language, TextRazor, and spaCy are competent general-purpose systems, and gpt-5-mini is a capable model given an underspecified job. But entity SEO asks a narrow question: of everything on and around this page, which concepts does the search ecosystem for this query actually traffic in? A model tuned to that question beats models tuned to Wikipedia-style entity linking or generic salience, on the referee that matters to practitioners, by margins

large enough to change what a user does next. That tuned model is ours, and the margin is not an accident: we have spent years building the extractor specifically for search, against ranking cohorts rather than encyclopedias, precisely because general-purpose NLP kept answering the wrong question. This study is the first time we have put that investment against the field under an external referee, and it is why a purpose-built model from a small team outscores a general-purpose API from Google itself.

The stakes are larger than tool trivia. In semantic SEO, the entity list is upstream of everything: content briefs, gap analyses, optimization checklists, and increasingly the prompts fed to writing tools all start from some extractor's idea of what the topic contains. A list that is half false positives does not merely waste effort; it steers a page away from the language the query's ecosystem actually uses, one plausible-looking wrong word at a time. A list aligned with Google's own vocabulary concentrates the same effort on relevance the ecosystem demonstrably traffics in. Whether that alignment also makes pages rank faster is the industry's operating assumption, not something this study measures; what it measures is the precondition, and the precondition is where the five models separate.

The false-positive rate is the number we would put in front of anyone buying entity-gap tooling. It bounds how much of a tool's to-do list can possibly pay off, whatever Google's ranking systems do internally. What this study shows is that the lists themselves, the starting point of all such work, differ radically in how much of Google's own language they reflect. The winning model is not a lab artifact: it is the production extractor behind our scan reports and API, and the availability note below describes how to run the same inspection on any page.

5. Limitations

One of the five models is ours, and we are publishing a comparison we won; readers should weigh that, and the method is specified so the test can be rerun against us. The referee is surface matching against Google's rendered words, not Google's internal entity understanding; paraphrases and synonyms are not credited, which penalizes all models equally but rewards extractors whose vocabulary matches Google's phrasing. Our model's entities come from our scan pipeline's own text extraction while competitors ran on an independent local crawl of the same URLs, an asymmetry we could not fully remove; competitor text used an identical truncation across models. Competitor recommendation lists are simulated gap analyses, not

those vendors' commercial products, and are built only from crawlable ranking pages. The population is one day, one locale, 50 hand-built keywords, and the 166 pages all five models could process. gpt-5-mini represents one budget LLM under one naive prompt; better prompting or newer models may do better, and we make no claim beyond the configuration tested. A rank-prediction referee was tested and excluded because every model failed it (§2).

6. SEO Implications

For teams doing entity or semantic optimization:

- Ask what extractor sits under your entity tool. It is the single biggest determinant of what the tool tells you, and “we use NLP” can mean anything from a sparse NER pipeline to an unstable LLM prompt.
- Sanity-check any entity checklist against Google's own surfaces for your query: the AI Overview, People Also Ask, related searches, and the pages that rank. If a recommended entity appears in none of them, you are likely looking at a false positive.
- Distrust entity metrics that change when nothing changed. A stable page should produce a stable entity list; if your tool's list churns between refreshes, its extractor may be guessing.
- Keep expectations honest: an aligned entity list tells you what the query's ecosystem talks about. Whether covering it moves rankings, and how fast, is a separate question this study does not answer.

References

1. Lancheres, E. (2026). “Do AI Assistants Cite Original Content? 793 AI Citations Measured Against the Pages That Rank.” On-Page.ai Research (source of the SERP and AI Overview capture reused here). api.on-page.ai/research/ai-citation-study
2. “AI features and your website,” Google Search Central documentation. developers.google.com/search/docs/appearance/ai-features
3. “Analyzing entities,” Google Cloud Natural Language documentation. cloud.google.com/natural-language/docs/analyzing-entities
4. “Entity extraction,” TextRazor documentation. textrazor.com/docs/rest
5. spaCy, en_core_web_sm English pipeline. spacy.io/models/en

6. OpenAI API documentation (gpt-5-mini). platform.openai.com/docs/models

Cite this study: Lancheres, E. (2026). The Best and Worst Entity Models for SEO: Which Find the Words Google Uses, and Which Feed You False Positives. On-Page.ai Research. https://api.on-page.ai/research/entity-model-study