Jentic

Discoverable by Design:

How API Description Quality Drives AI Agent Performance

Whitepaper

Authors: Emma Murphy, Dr. Adam Bermingham

Date: August 2026

Executive Summary

Agent tool selection accuracy is driven by API quality, not LLM intelligence

We find that the quality of API descriptions is the dominant factor determining AI agent accuracy at tool selection, and that even moderately cheap LLMs can achieve performance on-par with expensive models if given high-quality API descriptions to select from.

AI agents are rapidly becoming an execution layer across enterprise systems; depending not just on prompts, but on tools to execute actions across API catalogs and enterprise systems. The ability to reliably discover and execute the right tool at runtime is critical to agent performance.

The challenge of finding the right tool is fundamentally a search problem, and API description quality is a critical and often overlooked factor. Incomplete, ambiguous, or poorly structured descriptions cause incorrect tool selection, increasing inference costs and operational risk.

Jentic addresses this problem through three complementary capabilities:

1. **AI Discoverability (AID) score**
   A dimension in Jentic's AI-Readiness rubric, where APIs are scored according to how optimized they are for agent search.
2. **API Description Improvement**
   A capability for automatically enhancing API operation summaries and descriptions.
3. **Agent-First API Search Infrastructure**
   Semantic search designed for agent intent.

We performed a robust evaluation of agent search performance across a broad range of enterprise agentic use cases. We show that AID score is a reliable indicator of search performance, with higher AID scores consistently corresponding to better agent search success. We demonstrate that improving API descriptions produced significant gains in agent tool search success and show how this can be done in a scalable and systematic way.

When combined, these capabilities increase retrieval accuracy from 59% to 94% across search configurations and index quality levels. They also reveal that data quality can matter more than model choice. For enterprises deploying AI agents in production, this delta represents the difference between experimental tooling and dependable execution.

Introduction

Tool Description quality directly affects agent reliability; and it's fixable. We measure this effect and introduce practical capabilities to measure, improve and help agents to find the right tool for their intent.

An agent finds it more difficult to find the right tool as the tools available to it grow. For the simplest cases, an agent can bring all of the tools available into its context and select what it thinks is the best tool. For everything beyond the narrowest scope, an agent must search for the right tool across a surface of API and MCP tools, often retrieving candidates first and then selecting the right tool from the shortlist. The ability to retrieve and select the right tool is directly related to overall agent reliability. If an agent cannot find the right tool, it may burn tokens and time retrying, or eventually fail.

All search systems, whether semantic or lexical search, or hybrid systems fusing both, rely on matching the agent query to the records in the index. However many APIs today are not designed with agent usability in mind. API operation description fields can vary greatly from tool to tool in terms of style, domain language and verbosity. Quality also varies widely and agents may struggle where descriptions are inconsistent, vague or absent entirely.

To help understand this challenge of agent API usability and how to overcome it, Jentic has developed the Jentic AI-Readiness (AIR) rubric [1]. One of the six AIR dimensions is *AI Discoverability (AID)*. The *AID score* measures how effectively an API description can be discovered by an AI agent. Jentic provides an agent skill that improves API AI-readiness [2], which generally increases the AID score by enhancing API descriptions.

To investigate this, we evaluate the effect of improving API description quality on agent search performance across a large representative enterprise index. We examine how AID score tracks retrieval outcomes by examining the relationship between AID and search success. We evaluate agent performance across a broad range of agent tool-use scenarios and assess whether gains are consistent across different search approaches and model tiers.

Data and Methodology

API Index

To support our evaluation we curated a large enterprise tool index from Jentic's public API catalog [3]. We selected SaaS APIs with coverage across a range of common enterprise functions: communication, software development, finance, productivity, CRM, storage, automation and analytics. In these tools there are many adjacent or overlapping functionalities and many idiosyncrasies with domain language making this a large and challenging setting for agent tool disambiguation. In total, the index comprises 8,467 operations from 70 OpenAPI descriptions including for example Slack, Asana, Google, Docker and Zoom.

	Min	Max
Operations	50	400
Median *Summary* Length	34	405
Median *Description* Length	16	59

Analysing the Jentic API Catalog at the start of 2026 prior to indexing, we found that 511% of API Operations have neither Summary nor Description fields and just 43% have both. In the index, the length of summaries and descriptions that were present vary greatly from a few words to lengthy detail instructions. AID scores take into account several factors including the depth and clarity of textual descriptions. In the enterprise index, 12 APIs scored Level 0 "Not Ready", 57 Level 1 "Foundational", just one API scored Level 2 "AI-Aware". No APIs scored Level 3 "AI-Ready" or Level 4 "Agent-Optimized".

Using Jentic's improvement tooling we updated the descriptions and summaries in the index. Throughout the evaluation we refer to three indexes:

1. **base:** An index consisting of original APIs without any improvements.
2. **API-improved**: An index where just the target API description is improved.
3. **index-improved**: An index where all the API descriptions are improved.

Agent Use Cases

We developed 40 agent tool search use cases with coverage across the eight enterprise domains in the index. The process involved two steps. First, we shortlisted operations which represented core user-facing functionality for each API in the index. From these operations we selected 40 diverse and distinct operations ensuring coverage over endpoint method, domain, and operation patterns.

With these target operations in mind we developed a use case for each operation. Each use case intent has a base query that we consider a canonical agent search query. To represent diverse agent behavior we expand the use cases with four query variants with different levels of specificity, structure, and technical precision, giving 200 total queries.

Query Type	Type Description	Example Query
base	Simple and direct intent	*create Revolut payment order*
content	Intent with specific field content	*create Revolut order for €49.99 with customer email buyer@shop.com and line items for product SKU-56124*
natural	High-level natural language	*Create a new order on Revolut to accept payment from a customer via Revolut Pay, card, or digital wallets*
structured	Fixed format with semantic and syntactic structure	*Task: initialise payment order. Capability: payment processing. {{API: Revolut}}*
technical	Explicit API-level instructions	*POST to Revolut orders API with amount, currency, customer details, line_items, shipping, and industry_data parameters*

Search System

All systems use semantic search, using the same lightweight embedding model [4] and embedding based retrieval with cosine similarity relevance scoring. This reflects current practice in agentic tool discovery and is well-suited to multi-domain API surfaces. Lexical approaches, such as BM25, can require domain-sensitive tuning to handle synonymy and terminology variation. Hybrid systems combining semantic and lexical indices offer greater control and potential higher performance ceiling but at a cost of tuning and operational complexity. Semantic search requires minimal configuration and handles cross-domain retrieval without adaptation, representing a strong complexity-quality trade-off in the multi-domain setting.

To establish a baseline, we used Anthropic Tool Search Tool [5], following the instructions in the Tool Search With Embeddings Cookbook [6]. We compare this with two variants of Jentic's tool search via Jentic's hosted MCP. Both Jentic approaches handle weak or missing operation descriptions gracefully, while still rewarding high-quality ones.

System	Search Strategy
base-tool-search	Embedding-based implementation following Anthropic's Tool Search with Embeddings Cookbook. Operations are embedded from a structured text representation of the tool definition.
jentic-keyphrase	Operations are embedded from a distilled keyword representation derived from the operation definition, using corpus-level statistics to surface discriminative terms and phrases. Verbatim description text is not projected directly.
jentic-intent	Extends keyphrase with two additional components: a heuristically derived intent string constructed from operation structure (providing signal where summary or description fields are weak or absent), and truncated, concatenated summary and description text.

Search Performance Measures

We use two evaluation metrics:

1. **Search Success Rate** (SSR) - The agent selects the most relevant operation first time
2. **Hit rate at 10** (HR@10) - The relevant operation appears in the top ten retrieved candidate operations

Search success rate is our end-to-end agent search usability measure for the success of the search and select process; Hit rate@10 measures the retrieval step, on which selection is obviously dependent, and allows us to look at the retrieval approach in isolation. Statistical significance of observed differences in hit rate@10 between base and improved descriptions was assessed using McNemar's Test.

Findings

API Description Enhancement

Enhancing API operation summaries and descriptions meaningfully improves agent search success.

Jentic's description improvement system produced a consistent and measurable uplift in search performance. Agent search success rose from 85% to 91.5%. Agent retrieval measured by hit rate, increases from 0.87 to 0.95 for improving the relevant API and to 0.94 when the full index is improved. This shows how the underlying data improvements yield better search results for the agent to select from, reducing the likelihood of relevant operations being omitted from the results list.

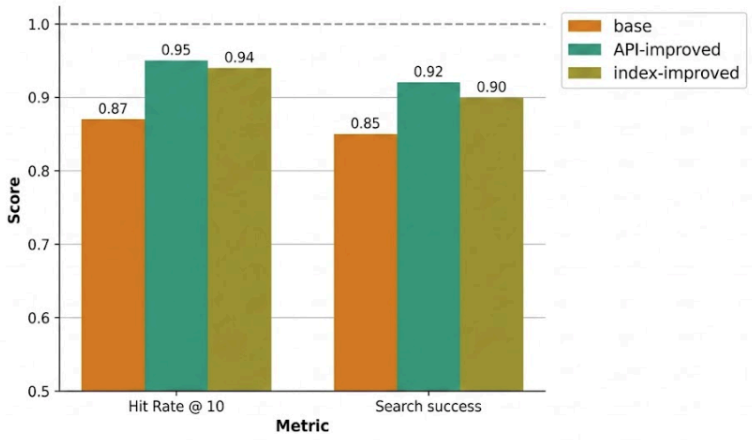

Figure 1.1, Overall Search retrieval and success by index quality (jentic-intent, n=200)

Targeted, API improvement is a practical optimization path. Refining descriptions for individual APIs achieves performance comparable to improving the full index; *API-improved* index achieved a hit rate of 0.95 compared to 0.94 for index-wide improvements. Teams can prioritize their most business-critical APIs with immediate returns rather than a full pass over their API catalog.

Improved performance for search success and retrieval is consistent across base, content, structured and technical query variants and for all index types, with structured queries showing the largest relative improvement. Natural queries showed no improvement, suggesting that explicit and direct agent query formulation is rewarded more in this approach.

Query Type	Agent Search Success Rate					
	base	API-improved	Δrel	index-improved	Δrel	n
all	**0.85**	**0.915**	**+7.7%**	**0.9**	**+5.9%**	**200**
base	0.9	0.975	+8.3%	0.975	+8.3%	40
structured	0.7	0.825	+17.9%	0.75	+7.1%	40
content	0.8	0.875	+9.3%	0.9	+12.5%	40
natural	0.9	0.9	+0.0%	0.875	-2.8%	40
technical	0.95	1.0	+5.3%	1.0	+5.3%	40

Query Type	Retrieval Hit Rate @10					
	base	API-improved	Δrel	index-improved	Δrel	n
all	**0.87**	**0.95**	**+9.2%*** **	**0.94**	**+8.1%** **	**200**
base	0.9	0.975	+8.3%	0.975	+8.3%	40
structured	0.775	0.95	+22.6%*	0.925	+19.4%	40
content	0.825	0.925	+12.1%	0.925	+12.1%	40
natural	0.9	0.9	+0.0%	0.875	-2.8%	40
technical	0.95	1.0	+5.3%	1.0	+5.3%	40

The benefits of improved descriptions hold across search systems. The gap between the search systems is substantial; *jentic-intent* reached a hit rate@10 of 0.94 on the improved index, significantly higher than *jentic-keyphrase* (0.86) and base-tool-search (0.65).

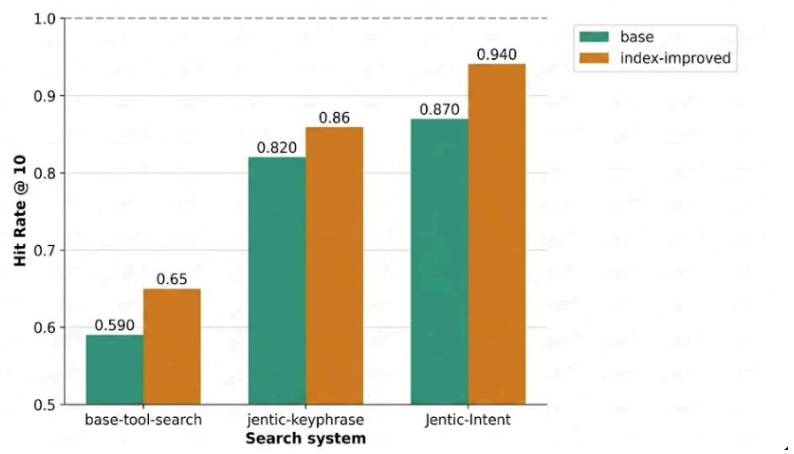

Figure 1.2: Overall retrieval results by search system and index quality.

This gap is most evident when the index contains many semantically similar operations, where small differences in wording can significantly affect ranking. The baseline system showed clear failure modes consistent with this, for example:

1. For the query “Search Mandrill sent messages by content and filters”, the top result was a generic *GET /search* operation , whereas jentic-intent surfaced the more specific and correct *POST /messages/search.json*.
2. For the query “list commits in GitLab repository”, the correct path was found but the wrong method, indicating it did not understand the retrieval intent. Meanwhile, jentic-intent correctly returned the retrieval operation.

Combining improved descriptions with *jentic-intent* search delivered the strongest results but description improvement alone moves the needle, regardless of the underlying search system.

AID Score to Search Correlation

AID score is a reliable indicator of agent search performance

AID scoring focuses on descriptive richness, captured through two sub-metrics: clarity and depth [1]. A higher AID score should predict stronger retrieval performance. In this analysis we use the base index (*base*), an improved index (*index-improved*), and an index with descriptions removed (*no_description*).

AID score showed a statistically significant positive correlation with retrieval performance across all index variants (r=0.185***, n=600). Higher AID scores consistently correspond to stronger retrieval performance, validating AID score as a practical indicator of API searchability.

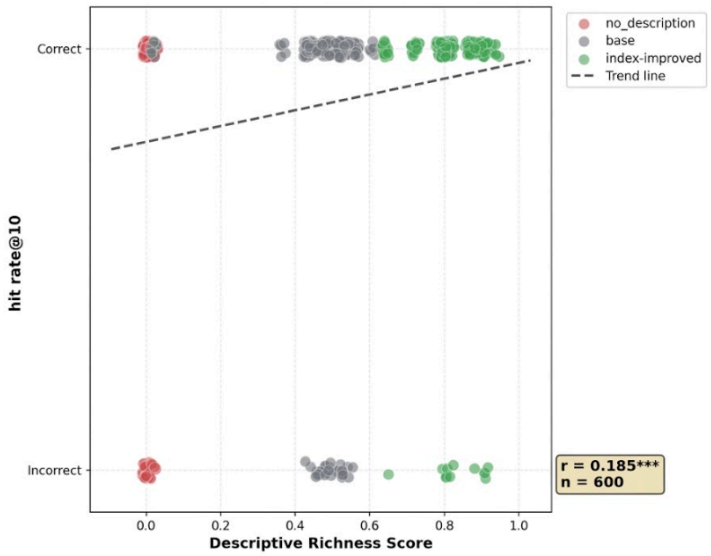

Figure 2.1: AID score vs hit rate@10 correlation across three indices

This correlation holds across query types. We find the strongest signal in content queries (r=0.26**) and technical queries (r=0.24**) where queries rely on discriminative signals: parameters, return fields, and domain terms captured by AID’s depth component.

Query Type	Correlation (r)	n
all	0.19***	600
base	0.13	120
content	0.26**	120
natural	0.15	120
structured	0.18	120
technical	0.24**	120

Overall, AID scoring is operationally useful for businesses because it provides a quantifiable way to evaluate API readiness for agentic use. Internally, teams can use AID score to prioritise API improvement work where it will have the highest impact, and to measure improvements over time. The score provides a direct line between agent reliability and API documentation quality.

Cross-Model LLM Search Performance

Across model tiers, description quality matters more for agent search success than reasoning capability.

We evaluated three Claude 4.5 model tiers - Haiku, Sonnet (our baseline) and Opus - across search success and request body formation. On improved descriptions, all three models achieved identical search success rates (93.5%). For base descriptions, search success averaged ~86%, with negligible variation across models.

Changing the model had no meaningful impact on search performance; improving the descriptions did.

Index	Metric	Haiku 4.5	Sonnet 4.5	Opus 4.5
base	Search Success Rate	0.86	0.865	0.855
base	Valid Request Rate	0.81	0.835	0.83
index-improved	Search Success Rate	0.935	0.935	0.935
index-improved	Valid Request Rate	0.865	0.9	0.905

Lower-cost models are adequate for retrieval and selection. This has significant cost implications; using Haiku instead of Opus for search tasks reduces model costs by 80% with no loss in selection accuracy.

Request body formation is more sensitive to model choice: Haiku achieved 86.5% validation accuracy on improved descriptions versus Sonnet's 90%, suggesting that richer descriptions benefit from a more capable model to fully translate them into correct structured output.

Conclusion

Improving API description quality is a practical, high-impact intervention that delivers measurable discoverability improvements for agent reliability.

Discoverability depends on both the data exposed through API descriptions and the search systems processing that data. Across a range of description qualities and search systems, the pattern is consistent. Our results show that API description quality and search configuration is a strong indicator of agent tool search reliability. AID score provides a quantifiable measure of description quality, giving teams a practical instrument for identifying and prioritising improvement work. Initial results also suggest that for retrieval and selection tasks, model choice matters less than either of these factors.

As agents become an increasingly critical component of enterprise workflows, organizations must treat API description quality as a key consideration in their AI strategy. Jentic's AID score, description improvement system and search infrastructure together provide a measurable practical path for addressing that challenge, with empirically validated gains in agent reliability and associated reductions in operating cost.

Limitations and Future Work

Queryset Coverage

This study evaluated performance using 200 queries derived from 40 base queries, providing a strong initial signal of performance across the APIs in scope. Building on these results, future work will systematically test every document operation, measuring search performance before and after the Jentic improvement tool. This operation-level benchmark will surface edge cases and further validate the generalisability of performance gains.

API Document Dataset

Despite efforts to index only relevant APIs, we risk misrepresenting the document distribution in domain-specific deployments. Future work will evaluate the approach within a focused domain, such as finance, using a curated set of industry specific APIs. In addition, a case study with a real world organisation would provide further validation by measuring impact in an operational environment.

References

[1] Jentic API AI-Readiness (AIR) Rubric Specification:
https://github.com/jentic/api-ai-readiness-framework/blob/main/docs/specification/spec.md

[2] Jentic AIR OpenAPI Improvement Skill jentic-api-improve:
https://github.com/jentic/jentic-api-scorecard#jentic-api-improve

[3] Jentic Public APIs:
https://github.com/jentic/jentic-public-apis

[4] all-MiniLM-L6-v2:
https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[5] Anthropic tool search tool:
https://platform.claude.com/docs/en/agents-and-tools/tool-use/tool-search-tool

[6] Anthropic tool search tool embedding cookbook:
https://platform.claude.com/cookbook/tool-use-tool-search-with-embeddings