Toward Intelligent Skill Discovery in A2A via Fine-Tuned Small Language Models

Elia Pacioni[1,2], Valerio Crocetti[1,3] and Davide Calvaresi[1]

[1]HES-SO Valais-Wallis, 3960 Sierre, Switzerland
[2]University of Extremadura, 06800 Mérida, Spain
[3]Università Politecnica delle Marche, 60121, Ancona, Italy

Abstract

The Agent-to-Agent (A2A) protocol standardizes communication among heterogeneous LLM-based agents through Agent Cards, task lifecycle management, and bindings. However, while Agent Cards describe agent capabilities, A2A does not yet define a canonical registry and search layer for a seamless semantic discovery of the agents' capabilities across distributed or consortium-based catalogs. As a result, discovery is left to ad hoc registries and implementation-specific mechanisms, making it difficult to provide reliable matchmaking across heterogeneous agent ecosystems. The closest historical analogy, the FIPA Directory Facilitator, addressed a similar need but relied on formal service descriptions and shared ontologies that are difficult to scale to open-ended LLM-agent environments. This position paper argues that A2A requires a modern counterpart of the Directory Facilitator: a scalable and auditable discovery layer that maps natural-language intents to machine-readable Agent Cards.

The proposed approach treats A2A agent and skill discovery as a semantic retrieval, re-ranking, and skill-selection task. It combines bi-encoder retrieval for efficient candidate selection with a fine-tuned Small Language Model (SLM) that re-ranks the retrieved Agent Cards, selects the skills supporting each match, and generates human-readable explanations. The paper further outlines an open research agenda for reproducible A2A discovery, including public benchmark construction, realistic query design, lexical and neural baselines, and evaluation protocols combining classical information retrieval metrics with LLM-assisted evaluation of semantic relevance, skill selection, and explanation quality. We argue that such a framework is needed to move A2A discovery beyond operational registries, and beyond the small sub-consortia in which agents currently interoperate, toward experimentally grounded semantic matchmaking.

Keywords

Agentic AI, MAS, Directory Facilitator, Skill Discovery, A2A Registries, Small Language Models

1. Introduction

Multi-agent LLM systems are progressively moving from prototypes to deployed infrastructure, and interoperability standards are becoming a foundational component of this ecosystem. The Agent-to-Agent (A2A) protocol, introduced by Google in April 2025 [1] and later contributed to the Linux Foundation, standardizes communication among heterogeneous LLM-based agents through Agent Cards, task lifecycle management, and bindings. Agent Cards describe what an agent can do, but A2A defines no canonical registry-and-search layer for the *semantic discovery* of agent capabilities across distributed or consortium-based catalogs. How agents find one another is thus left to implementation-specific mechanisms, which makes reproducible matchmaking across heterogeneous ecosystems difficult.

The challenge itself is not new. FIPA-compliant multi-agent systems addressed the same need, introducing the Directory Facilitator (DF) [2], a standardized service agent that maintained a catalog of offered services and answered structured capability queries. The DF is based on shared ontologies and formal service-description languages, which made it semantically robust but too heavy to adopt at scale in open-ended LLM-agent settings.

To date, modern agents are severely affected by the lack of such a discovery layer. Therefore, we argue for rebuilding it with modern tools as an open, reproducible, and auditable component that

24th International Conference of the Italian Association for Artificial Intelligence, October 6–9, 2026, Perugia, Italy
elia.pacioni@hevs.ch (E. Pacioni); valerio.crocetti@hevs.ch (V. Crocetti); davide.calvaresi@hevs.ch (D. Calvaresi)
0000-0002-1557-4870 (E. Pacioni); 0009-0002-3760-6041 (V. Crocetti); 0000-0001-9816-7439 (D. Calvaresi)


maps natural-language intents to machine-readable Agent Cards. We proceed in three steps. First, we ground the difficulty of agent discovery in the mechanics of a single query, showing what must happen when one intent is matched against many candidates (Section 2). Second, we review existing registries and identify the gaps they leave open (Section 3). Finally, we introduce our approach: a two-stage discovery pipeline (Section 4). In the first stage, a bi-encoder embeds the user intent and the serialized Agent Cards, and retrieves a shortlist of candidate agents by similarity. In the second stage, a fine-tuned SLM re-ranks this shortlist; for each candidate it selects the relevant skills and produces a natural-language justification for its decision. What is ultimately at stake is not only retrieval quality but also digital sovereignty. Because the SLM is small enough to run on premises and its explanations are open to inspection, institutions such as public administrations, hospitals, and universities can compose heterogeneous agents without surrendering the matchmaking step to a closed, externally controlled service.

2. The Problem

A client agent holds a natural-language intent and must select, from a catalog of N Agent Cards each exposing up to M skills, the agents whose skills satisfy that intent and identify the relevant skills for each selected agent. A full skill description, carrying a name, prose summary, tags, and input/output examples, is on the order of 150 tokens. Two natural strategies bracket the design space, and neither of them is satisfactory.

The *context-rich* strategy places every full Agent Card in the client's decision prompt: for a modest catalog of $N = 50$ and $M = 20$ this is roughly $50 \times 20 \times 150 \approx 1.5 \cdot 10^5$ tokens of candidates alone, which is costly per call, scales linearly with the catalog, and degrades decision quality as the few relevant skills are diluted among hundreds of irrelevant ones. The *context-lean* strategy keeps only skill names or a syntactic match on tags, cutting the prompt by one to two orders of magnitude. Still, it discards exactly the semantic signal that the decision requires, since a lexical label rarely matches an intent phrased in different words, a failure of off-the-shelf retrievers that has been documented for tool-style descriptions [3]. Neither compactness nor completeness alone is therefore sufficient.

A third approach, faithful to the spirit of A2A, serializes and indexes each Agent Card as a single retrieval item, preserving its metadata and skill descriptions. A bi-encoder retrieves the top-K cards by semantic similarity to the intent; only these candidates are then re-ranked by an SLM, which also selects the relevant skills. This reduces the discovery context from approximately $N \times M \times 150$ to $K \times M \times 150$ tokens, where $K \ll N$, without discarding the semantic information contained in the cards. We therefore formalize agent discovery as a semantic retrieval, re-ranking, and skill-selection task over Agent Cards, leading to the central research question of this paper:

> **RQ.** Can a fine-tuned Small Language Model re-rank A2A Agent Cards more accurately than lexical and dense-retrieval baselines, and how reliably can it identify the most relevant skills for each candidate? Three constraints frame the question. The re-ranker sees only bi-encoder-retrieved candidates. It must justify both decisions with an auditable explanation. It must run on premises.

3. State of the Art and Gaps

This paper fixes its analysis to the A2A specification release v1.0.1, released on 26 May 2026 [4]. A2A standardizes no canonical registry or search API for finding agents by capability across catalogs. Community registries (for example, `a2aregistry.org` [5] and `allenday/a2a-registry` [6]) fill the gap by using syntactic matching on skill names and tags. Industrial efforts such as the Agent Name Service (ANS) [7] introduce capability-based and semantic discovery, but, to the best of our knowledge based on public documentation, they provide neither a reproducible evaluation nor an open benchmark

for the semantic ranking of A2A Agent Cards. The ecosystem itself acknowledges the need: the A2A roadmap now lists the consolidation of registry efforts as an open item [8].

In the adjacent tool-use setting, where the Model Context Protocol (MCP) [9] plays for tools the role A2A plays for agents, skill libraries [10], tool creation and toolset retrieval [11, 12], and tool learning surveys [13] address how a single agent selects its own tools. No comparable open resource exists for discovery across agents via A2A Agent Cards.

From a modeling perspective, neural re-ranking has progressed from pointwise cross-encoders, which jointly score each query-document pair [14], to listwise LLM rankers, which instead order candidates by comparing them against one another [15], and to their distillation into compact open models [16]. This progression suggests that, with suitable supervision, such distilled rankers can approach the effectiveness of substantially larger ones while reducing inference cost. On the resource side, BEIR [17] established the methodological template for heterogeneous retrieval benchmarks: diverse corpora and query types, a shared evaluation interface, and a zero-shot protocol measuring generalization beyond the training domain. Neither line of work, however, has been applied to the joint re-ranking of A2A Agent Cards and the identification of the skills that support each match. Semantic web service matchmakers such as OWLS-MX [18] hybridized logic-based and syntactic matching for OWL-S services, but they inherit the DF's dependence on formal service ontologies.

We identify four gaps:

G1. No open, reproducible benchmark for the semantic discovery of A2A Agent Cards and skills.

G2. No comparable, published baselines and ranking methods for matching natural-language intents to Agent Cards and identifying the skills that support each match.

G3. No lightweight, on-premises alternative to formal ontologies that preserves auditability through human-readable explanations.

G4. No evaluation protocol that reconciles classical Information Retrieval (IR) metrics, skill-level measures, and LLM-assisted judgment under proper reliability controls.

4. The Position: A Reproducible, Sovereign Discovery Layer

We argue that the four gaps can be closed together by a single, modest research program built on five ingredients.

Open benchmark (G1). A curated corpus of A2A Agent Cards, both real and synthetic [19], a set of realistic natural-language queries spanning capability, domain, and constraint-based intents, and a human-annotated relevance ground truth that enables reproducible measurement. BEIR [17] is the methodological template to follow.

Two-stage discovery (G2, G3). A bi-encoder [20, 21] receives the natural-language intent and one representation for each Agent Card, retrieving the top-K candidate Agent Cards efficiently; a fine-tuned SLM [22] then re-ranks them [14], selects the supporting skills for each match, and generates human-readable, auditable explanations of each suggestion, in effect a modern and lightweight counterpart of the FIPA Directory Facilitator.

Synthetic supervision, human evaluation (G2). The re-ranker is trained on synthetic relevance labels and skill-level annotations following generative pseudo-labeling practice [23]: queries are generated from Agent Cards to form positive pairs, hard negatives are mined with the bi-encoder, and label quality is verified before training in the spirit of APIGen [19]. The human-annotated ground truth is reserved strictly for evaluation, so no test signal leaks into training.

Auditability and sovereignty (G3). The SLM is small enough to run on premises, and its explanations enrich the client agent's context with an auditable rationale. Open-weight families such as Gemma [24] and Qwen [25], served with standard local inference runtimes [26], are natural candidates.

Reproducible evaluation (G4). An evaluation protocol that combines classical IR metrics (nDCG [27], MRR, Recall@K) with precision, recall, and F1 for the selected skills, and an LLM-as-judge assessment [28] of semantic relevance and of explanation quality and faithfulness [29], validated against the human ground truth to control judge bias. Lexical (BM25) [30] and cross-encoder rankers

serve as first-class baselines.

An illustrative example. Consider the intent *"extract line items from a scanned invoice and flag any amount above 10,000 CHF"*. A syntactic search over tags such as `pdf` or `invoice` returns superficially plausible but incomplete matches because the constraint on amounts is not expressed anywhere in the tags, and the notion of "line items" is phrased differently across cards. The bi-encoder instead retrieves the top-K cards whose skill descriptions are semantically closest to the intent, and the SLM re-ranker scores them, highlights the supporting skills, and justifies each decision. It might rank first an agent exposing a `document-extraction` skill whose input/output example maps invoice fields to structured records together with a `threshold-alert` skill, explaining that the pair jointly covers both the extraction and the conditional flagging; and rank lower an OCR-only agent, explaining that it recovers raw text but neither structures the line items nor supports the amount condition. The explanation is not decorative: it is the artifact that allows a human operator or a downstream orchestrator to audit *why* a given agent was proposed before delegating a task to it. This is precisely the accountability that syntactic registries cannot offer, and that formal ontologies provide only at the cost of scalability.

5. Open Questions and Conclusion

Realizing this layer surfaces questions we believe the community should debate. *How realistic can an open Agent Card corpus be* while public cards remain scarce, and how should synthetic and collected cards be mixed and provenance-documented without overfitting queries to any single source? *Where does the sovereignty trade-off actually sit*: can an on-premises SLM re-ranker match a large proprietary ranker closely enough that the gains in cost, latency, and auditability outweigh the quality gap? *Can an LLM judge be trusted* as a scalable proxy for human relevance, and under what reliability controls? And *how should the layer be decoupled from the evolving A2A specification* so that transport changes do not break capability matching?

We stress that a rigorous negative result would itself be a contribution: if a fine-tuned SLM re-ranker fails to beat strong bi-encoder or lexical baselines on a well-constructed benchmark, that finding constrains the design of any future discovery layer. Because MCP tool descriptions share the same structure of capability metadata [9], both the benchmark design and the discovery models should generalize beyond A2A, widening the addressable ecosystem.

A2A has standardized how agents communicate, but not how they discover one another. We have argued that the missing discovery layer should be rebuilt as an open, reproducible, and auditable component that reframes the classic FIPA DF through a two-stage retrieve-and-re-rank design, with a fine-tuned SLM deployable on premises acting as both re-ranker and skill selector. The concrete deliverables that would ground this position, namely an open benchmark, comparable baselines, and a validated evaluation protocol, are modest in scope yet reusable by the whole community. Together they are what is needed to move A2A discovery beyond operational registries toward experimentally grounded semantic matchmaking, and they lay the experimental groundwork for a larger effort on trustworthy and sovereign agent ecosystems.

Acknowledgments

This work was supported by the project - Mind the Gap: Adapting Higher Education to the Age of AI (MTG, 140012/RI-PSTRAT25-06).

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT and Claude for grammar, style, and clarity checks and sentence-level paraphrasing. Grammarly was also used to support grammar, spelling, and style checks. After using these tools, the author(s) reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] A2A Project, Agent2Agent (A2A) Protocol Specification, Linux Foundation, https://a2a-protocol.org, 2025. Originally developed by Google, hosted by the Linux Foundation since June 2025.

[2] Foundation for Intelligent Physical Agents, FIPA Agent Management Specification, Standard SC00023K, http://www.fipa.org/specs/fipa00023/, 2004.

[3] Z. Shi, Y. Wang, L. Yan, P. Ren, S. Wang, D. Yin, Z. Ren, Retrieval models aren't tool-savvy: Benchmarking tool retrieval for large language models, in: Findings of the Association for Computational Linguistics: ACL 2025, 2025, pp. 24497–24524.

[4] A2A Project, A2A Protocol v1.0.1 Release Notes, 2026. URL: https://github.com/a2aproject/A2A/releases/tag/v1.0.1, specification release v1.0.1, release notes dated 26 May 2026; accessed 10 July 2026.

[5] A2A Registry community, A2A Registry, https://www.a2aregistry.org, 2025. Community-maintained registry of A2A agents.

[6] A. Day, a2a-registry: A community-driven registry of A2A agents, https://github.com/allenday/a2a-registry, 2025.

[7] K. Huang, V. S. Narajala, I. Habler, A. Sheriff, Agent Name Service (ANS): A Universal Directory for Secure AI Agent Discovery and Interoperability, arXiv preprint arXiv:2505.10609, 2025.

[8] Linux Foundation, A2A Protocol Surpasses 150 Organizations, Lands in Major Cloud Platforms, and Sees Enterprise Production Use in First Year, Press release, https://www.linuxfoundation.org/press/a2a-protocol-surpasses-150-organizations-lands-in-major-cloud-platforms-and-sees-enterprise-production-use-in-first-year, 2026. Lists registry consolidation on the A2A roadmap.

[9] Anthropic, Model Context Protocol (MCP) Specification, https://modelcontextprotocol.io, 2024. Now hosted by the Linux Foundation.

[10] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, A. Anandkumar, Voyager: An open-ended embodied agent with large language models, arXiv preprint arXiv:2305.16291 (2023).

[11] T. Cai, X. Wang, T. Ma, X. Chen, D. Zhou, Large language models as tool makers, in: International Conference on Learning Representations, volume 2024, 2024, pp. 54067–54089.

[12] L. Yuan, Y. Chen, X. Wang, Y. Fung, H. Peng, H. Ji, Craft: Customizing llms by creating and retrieving from specialized toolsets, in: International Conference on Learning Representations, volume 2024, 2024, pp. 40097–40125.

[13] C. Qu, S. Dai, X. Wei, H. Cai, S. Wang, D. Yin, J. Xu, J.-R. Wen, Tool learning with large language models: A survey, Frontiers of Computer Science 19 (2025) 198343.

[14] R. Nogueira, K. Cho, Passage re-ranking with BERT, arXiv preprint arXiv:1901.04085, 2019.

[15] W. Sun, L. Yan, X. Ma, S. Wang, P. Ren, Z. Chen, D. Yin, Z. Ren, Is ChatGPT good at search? investigating large language models as re-ranking agents, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2023, pp. 14918–14937.

[16] R. Pradeep, S. Sharifymoghaddam, J. Lin, RankZephyr: Effective and robust zero-shot listwise reranking is a breeze!, arXiv preprint arXiv:2312.02724, 2023.

[17] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, I. Gurevych, BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models, 2021.

[18] M. Klusch, B. Fries, K. Sycara, OWLS-MX: A hybrid semantic web service matchmaker for OWL-S services, Journal of Web Semantics 7 (2009) 121–133.

[19] Z. Liu, T. Hoang, J. Zhang, M. Zhu, T. Lan, S. Kokane, J. Tan, W. Yao, Z. Liu, Y. Feng, et al., APIGen: Automated pipeline for generating verifiable and diverse function-calling datasets, Advances in Neural Information Processing Systems 37 (2024) 54463–54482.

[20] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using siamese BERT-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 3982–3992.

[21] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W.-t. Yih, Dense passage retrieval for open-domain question answering, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 6769–6781.

[22] P. Belcak, G. Heinrich, S. Diao, Y. Fu, X. Dong, S. Muralidharan, Y. C. Lin, P. Molchanov, Small language models are the future of agentic AI, arXiv preprint arXiv:2506.02153, 2025.
[23] K. Wang, N. Thakur, N. Reimers, I. Gurevych, GPL: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval, in: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), 2022, pp. 2345–2360.
[24] Google DeepMind, Gemma 4 Model Card, https://ai.google.dev/gemma/docs/core/model_card_4, 2026. Accessed: 2026-05-28.
[25] Qwen Team, Qwen3 Technical Report, arXiv preprint arXiv:2505.09388, 2025.
[26] G. Gerganov, llama.cpp contributors, llama.cpp: LLM inference in C/C++, https://github.com/ggml-org/llama.cpp, 2023. Accessed: 2026-06-19.
[27] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of IR techniques, ACM Transactions on Information Systems 20 (2002) 422–446.
[28] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, I. Stoica, Judging LLM-as-a-judge with MT-Bench and chatbot arena, in: Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track, 2023, pp. 46595–46623.
[29] Q. Lyu, S. Havaldar, A. Stein, L. Zhang, D. Rao, E. Wong, M. Apidianaki, C. Callison-Burch, Faithful Chain-of-Thought reasoning, in: Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, 2023, pp. 305–329.
[30] S. Robertson, H. Zaragoza, The probabilistic relevance framework: BM25 and beyond, Foundations and Trends in Information Retrieval 3 (2009) 333–389.