A Multi-agent Approach to Technology Intelligence: From Patents to Market Signals with Large Language Models

Jan Černý
Prague University of Economics and Business / Faculty of Informatics and Statistics, Czechia

jan.cerny@vse.cz

Abstract: Technology Intelligence (TI) draws on knowledge management and competitive intelligence to give decision-makers awareness of technological trends, opportunities and threats. Recent frontier large language models and autonomous agents make it feasible to automate the full TI lifecycle. This paper presents TechIntel-MAS, an eight-role multi-agent system implemented on the CrewAI framework, that transforms a body of disclosed patent filings, supplied alongside a free-text intelligence request, into a structured competitive-intelligence report. The system treats patents as seed signals and searches the open web for corresponding market evidence (products, vendors, deployments, partnerships), and reports the absence of evidence honestly when none is found. Six frontier models (GPT-5.4, Claude Sonnet 4.6, Gemini 3.1 Pro, DeepSeek R1, Moonshot Kimi K2 Thinking, Qwen3 Max Thinking) execute the generative roles in parallel. An independent evaluator pool (Grok 4 and Llama 4 Maverick) operates the Verifier, Telemetry and Judge roles, preserving the principle that the model evaluating an artefact is never the one that produced it. We evaluate the pipeline on four heterogeneous patent filings and report per-model precision, recall, F1, semantic alignment, KIQ-relevancy and a coarse expected calibration error, together with cost and elapsed-time figures from a fully instrumented run. We make three contributions: a TI-specific eight-role architecture aligned with the Kerr three-tier model; a patent-as-seed-signal retrieval pattern that grounds disclosed inventions in open-web market evidence; and an evaluation methodology with strict generator-evaluator independence, evaluated across six frontier LLMs in a pilot study. The paper situates these results within the knowledge-management literature and discusses implications for organisational learning and strategic forecasting in agentic settings.

Keywords*:* Technology intelligence, Multi-agent systems, Large language models, Agentic AI, Patent analytics, Knowledge management, Competitive intelligence, Generator-evaluator independence, CrewAI, Strategic forecasting

1. Introduction

Technology-intensive firms face intense pressure from shortening product cycles, a turbulent competitive environment and constant customer demands for quality. These pressures unfold amid a growing volume of data, increasingly including AI-generated content. Companies increasingly adopt technology intelligence (TI) processes to analyse the market environment, customer demand and competitor activity as an integral part of strategic issue management. Broadly defined as the acquisition and assessment of information on technological trends, opportunities, and threats, TI provides decision-makers with the situational awareness necessary to reduce risks and spot opportunities in an increasingly complex and fast-moving innovation landscape (Kerr et al., 2006; Lichtenthaler, 2003). TI can also be regarded as a subset of competitive intelligence, sharing its defining conditions: it must be a continuous, legal and ethical process. The conceptual model proposed by Kerr et al. (2006) decomposed TI into three tiers: a framework level that maps decision-maker knowledge gaps onto the company's intelligence activities, a system level that configures operational modes (mine, trawl, target, scan) to intelligence needs, and a process level consisting of a six-phase cycle of coordinate, search, filter, analyse, document, and disseminate. This three-tier architecture remains the dominant structural reference for TI system design and has been empirically tested and applied across multiple industry contexts (Mortara et al., 2009, 2010).

Two decades of empirical research document considerable heterogeneity in how large firms organise and coordinate TI. Lichtenthaler traced three successive generations of TI management and showed that effective coordination cannot rely on structural mechanisms alone but requires the simultaneous integration of structural, hybrid and informal forms whose balance reflects firm culture, size, industry dynamism and innovation strategy (Lichtenthaler, 2003, 2004a, 2004b, 2004c, 2005). Practitioner and corporate accounts reinforce this: firms such as DuPont, Motorola and Baxter show that TI must be systematically organised and delivered to those empowered to act (Norling et al., 2000), while implementations at Kodak (Scan & Target) and Deutsche Telekom (the Technology Radar) illustrate systematic external scouting and continuous visualisation of TI outputs (Mortara et al., 2009, 2010; Rohrbeck et al., 2006).

A parallel stream has progressively automated core TI tasks. Porter's QTIP framework argued that database access, analytical software, automated routines and process standardisation can compress analysis timescales through 'tech mining' (Porter, 2005). Later systems combined text mining, morphology and TRIZ-based reasoning to derive technology opportunities and trends from patents (Yoon, 2008; Yoon & Kim, 2012), and semantic-network methods deepened the usability of patent-derived intelligence (An et al., 2018). Applications

span health-IT forecasting (Behkami & Daim, 2012), open innovation (Veugelers et al., 2010) and structured implementation in SMEs (Savioz, 2003).

Taken together, this body of research reveals a persistent structural tension in the TI field. The organisational literature emphasises human judgement, contextual sensitivity, hierarchical coordination and cultural alignment as preconditions for effective TI. The computational literature, by contrast, has shown that automation can substantially improve the speed, coverage and consistency of TI. This tension has been partially reconciled through hybrid approaches, yet the synthesis remains incomplete. The emergence of large language models (LLMs) and autonomous AI agents now introduces a qualitatively new possibility: agentic TI systems that can execute multi-step intelligence workflows, including data retrieval, filtering, semantic analysis, synthesis, and dissemination, with adaptive goal-directed behaviour and minimal human intervention. Recent work has begun to explore related applications, such as retrieval-augmented generation for defence technology intelligence (Zhou et al., 2024).

1.1 Agentic AI and LLM-based Intelligence Systems

A growing body of work studies LLM-based multi-agent systems, in which specialised agents plan, act and interact to solve tasks beyond the reach of a single model (Guo et al., 2024). Within it, agentic retrieval-augmented generation embeds autonomous agents into the retrieval loop, adding reflection, planning and tool use to static RAG (Singh et al., 2025). Multi-agent variants such as MA-RAG decompose retrieval and synthesis across Planner, Extractor and QA agents (Nguyen et al., 2025). These systems, however, target open-domain question answering and are neither grounded in intelligence theory nor evaluated as organisational knowledge processes. A parallel stream examines the LLM-as-judge paradigm and its reliability, documenting biases such as self-preference, in which models over-rate their own outputs (Wataoka et al., 2024; Gu et al., 2026). In applied settings, LLMs have labelled and analysed patents at scale under human supervision (Pelaez et al., 2023) and evaluated strategic alternatives, where aggregating many model judgements approximates expert consensus (Csaszar et al., 2024). TechIntel-MAS combines and extends these lines of work: it adapts a generic agentic-RAG pattern into a TI-specific, Kerr-aligned architecture, grounds that architecture in knowledge-management theory, and makes generator-evaluator independence a core methodological principle rather than studying evaluation in the abstract.

Building on these developments, we designed a TI multi-agent approach that reflects the full intelligence lifecycle: planning, collection, analysis, verification, reporting, dissemination, measurement and judging.

2. Theoretical Framework

We treat TechIntel-MAS as a system for organisational knowledge processing and base it on the three-tier Technology Intelligence model of Kerr et al. (2006), still the standard reference for TI system design. That model sets out a framework tier, which links decision-makers' knowledge gaps to intelligence tasks; a system tier, which sets how information is gathered, through the mine, trawl, target and scan modes; and a process tier, the six-phase cycle that runs from coordination to dissemination. Our pipeline covers all three tiers and adds one step the original does not include: a closing stage that measures each run and compares the models. The Methodology describes how the agents carry out these tiers; this section explains why the design is well founded.

The knowledge-management process view of Alavi and Leidner (2001), revisited for generative AI by Alavi et al. (2024), positions the system as automating knowledge creation, storage and transfer while leaving application to human decision-makers. In the SECI terms of Nonaka (1994), the generative roles accelerate externalisation and combination, converting the tacit content of patent text into explicit, structured intelligence without displacing the tacit judgement on which action still depends (He et al., 2025). Absorptive capacity (Cohen & Levinthal, 1990) provides the link: the system recognises, acquires, assimilates, transforms and applies external signals, turning disclosed inventions into internal strategic knowledge (Pletsch et al., 2026).

The agent layer draws on current work in agentic AI. Role specialisation and orchestration follow established practice in LLM multi-agent systems (Guo et al., 2024), and the patent-seeded, tool-using retrieval loop is a form of agentic retrieval-augmented generation (Singh et al., 2025). Our requirement that no model evaluates its own output rests on the self-preference bias reported in the LLM-as-judge literature (Wataoka et al., 2024; Gu et al., 2026): because language-model evaluators tend to favour their own generations, separating generators from evaluators is a design requirement, not a convenience.

3. Methodology

Our methodology centres on a multi-agent system for Technology Intelligence that mirrors how intelligence work is organised. Each agent takes a specific role (planning, collecting, analysing, verifying, reporting, disseminating, measuring, judging), and the design replicates not only the tasks but the interactions and feedback loops between roles.

To assess how well current frontier models handle these tasks, we ran six models in parallel across the generative agents: GPT-5.4 (OpenAI), Claude Sonnet 4.6 (Anthropic), Gemini 3.1 Pro, DeepSeek R1, Moonshot Kimi K2 Thinking, and Qwen3 Max Thinking (the latter four routed through OpenRouter). Each generative agent therefore produces six independent outputs, one per model, providing a basis for direct comparison. For evaluation and judging, we used Grok 4 and Llama 4 Maverick (both via OpenRouter) as independent assessors, deliberately keeping the evaluating models separate from the generating ones to avoid a model evaluating its own output.

3.1 Agent 1: Planner

The Planner initiates the pipeline. It ingests the organisation's intelligence needs together with a corpus of disclosed patent PDFs, and turns them into a structured intelligence brief. Patents are parsed into patent signals (title, technologies, claims, assignees, IPC/CPC classifications, and filing dates) that seed the brief. The brief contains the Key Intelligence Questions (KIQs), scope, entities of interest, timeframe, search directives, and the verbatim patent signals. Each search directive combines a technology term with a commercial term from a closed set (product, launch, deployment, pilot, partnership, vendor, commercialization, customer, case study); at least five are required, distributed across four archetypes (assignee-anchored, technology-trend, competitive/substitute, temporal). Intelligence needs are ingested automatically from connected systems (board minutes, management requests, strategic reviews) or as manually submitted requests. Before passing the brief downstream, the Planner sends it to the Verifier to confirm that it reflects the request.

3.2 Agent 2: Collector

The Collector receives the validated brief from the Planner and executes its search directives against two open-web search APIs: Exa (semantic) and Brave (broad). Patent ingestion has been moved upstream to the Planner, so the Collector handles web sources only. The agent's generative role is limited: it constructs queries, but retrieval and item content are external to the model and may not be invented. Every item is tagged with source_type (web_exa or web_brave) and source_id (URL) and deduplicated by source_id before emission. A directive that returns nothing may be retried once with a broader rephrasing; further expansion is prohibited. When every directive returns zero hits, the Collector sets no_signal=true and records the attempted queries and channel errors, leaving the items list empty (placeholder hits are forbidden). The Collector then passes the corpus to the Verifier, which checks whether the collection is complete relative to the planning brief.

3.3 Agent 3: Analyst

The Analyst takes the verified corpus and produces a structured analysis: a list of findings, each linked to one or more corpus items and mapped to one or more KIQs. Findings without supporting evidence are excluded.

Once the analysis is complete, both the raw data and the analytical output go to the Verifier. The Verifier compares the two, checking whether the conclusions are present in the collected evidence and whether the synthesis holds together logically. If the Verifier identifies issues, classified into a closed set of five categories (unsupported, logical_gap, factual_error, coverage_gap, scope_drift), it sends the output back to the Analyst with specific annotations. The Analyst then drops findings flagged unsupported or factual_error and reformulates logical_gap items with stronger reasoning. This correction loop can repeat up to three iterations before the output moves to the Reporter.

3.4 Agent V: Verifier

The Verifier acts as a quality gate that sits between the main pipeline agents. It is not a sequential step but a cross-cutting layer: Agents 1, 2, and 3 all call on it at designated checkpoints. Its job is to compare what an agent produced against what it was asked to produce: the brief against the request, coverage against the brief, and conclusions against the collected data. Issues are classified into a closed set of five categories (unsupported, logical_gap, factual_error, coverage_gap, scope_drift), which enforces consistent severity coding across runs and across models. The Verifier also recognises a no-signal exemption. Two evaluators run the verification prompt independently and their reports are merged. Disagreement reduces the report's confidence and is

logged through an evaluator agreement score. The Verifier uses different models from the agents it evaluates, drawing on an independent evaluator pool (Grok 4 and Llama 4 Maverick, via OpenRouter), so the model checking the work is never the one that produced it.

3.5 Agent 4: Reporter

The Reporter takes the verified analytical output and turns it into a structured intelligence report that decision-makers can use. Every report follows the same format to ensure consistency and comparability across intelligence cycles.

Executive Summary: a concise overview of the findings, limited to 500 words.

Key Insights: the ten most significant conclusions, enumerated for quick reference.

Recommendations: three to six actionable recommendations, each naming a concrete decision, action or watchpoint linked to the findings.

3.6 Agent 5: Disseminator

The Disseminator delivers the finalised report to wherever the organisation requires it. This could be email, an internal knowledge management system, a Slack channel, or any other communication platform specified by the requesting team. The agent's role is purely logistical but essential: intelligence that never reaches the right people at the right time is wasted.

3.7 Agent 6: Telemetry

The Telemetry agent measures how well the system performed. While the Verifier checks individual outputs during the pipeline, the Telemetry agent evaluates the end-to-end quality of the entire intelligence cycle after it is complete. It applies four quantitative metrics, computed deterministically from the run artefacts, with the retrieved corpus (the set of collected URLs) serving as the reference set. Cosine similarity measures the semantic alignment between the analytical output and the original collected data, assessing whether the analysis stayed faithful to its source material. Precision measures what proportion of the claims in the final report are supported by the underlying data. Recall measures the proportion of collected corpus items cited at least once across the findings, that is, corpus utilisation rather than completeness against all real-world evidence. F1 is the harmonic mean of precision and recall.

In addition to these data-grounding metrics, the Telemetry agent computes kiq_relevancy: the mean cosine similarity between the report body and each Key Intelligence Question. This matters because a report can be technically accurate yet still fail to answer what the organisation asked. The Telemetry agent produces a structured performance scorecard for each model and each intelligence cycle, which feeds directly into Agent 7.

Agent 7: Judge

The Judge takes the Telemetry scorecard across all six generative models and determines which delivered the most consistent and comprehensive intelligence reports. Consistency here means stable performance across multiple runs and different intelligence tasks. Comprehensiveness means covering the full scope of the intelligence needs without significant gaps.

The Judge does not make this determination in isolation. It consults with the Verifier agent, incorporating the qualitative observations the Verifier accumulated during the pipeline: which model produced the most grounded analysis, which model required the most correction loops, which model's outputs were most frequently flagged. By combining the Telemetry agent's quantitative metrics with the Verifier's qualitative assessments, the Judge produces a final model recommendation that reflects both measurable performance and practical reliability.

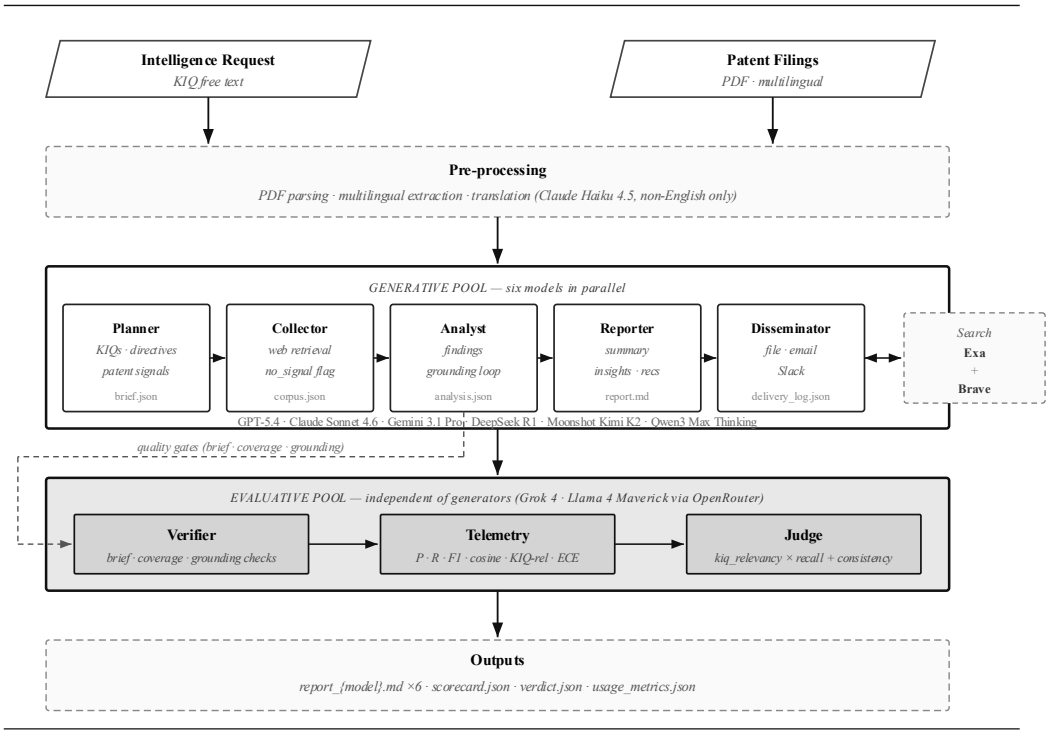

Figure 1: MAS Pipeline Overview

Across the generative agents (Planner, Collector, Analyst, Reporter), every prompt runs in parallel against the six-model generative pool, producing six independent outputs per agent for comparison. The evaluative agents (Verifier, Telemetry, Judge) run on a separate, independent pool of Grok 4 and Llama 4 Maverick (both via OpenRouter), preserving the methodological principle that the model evaluating an artefact is never the same model that produced it. The Disseminator invokes no language model; it is a deterministic distribution function.

4. Results

This section reports the empirical evaluation of the MAS pipeline on a sample of four patent filings of heterogeneous subject matter. Each patent was processed end-to-end through the full multi-agent pipeline, with six generative large language models running in parallel as Planner, Analyst, and Reporter, and two independent evaluators (Grok 4 and Llama 4 Maverick, both via OpenRouter) operating the Verifier, Telemetry, and Judge roles. The pipeline produced 20 to 21 artefacts per run, including a brief, a corpus, six per-model analyses, six per-model reports, a verification report, a telemetry scorecard, a judge verdict, and, on the instrumented run, a per-(agent, model) usage record. Configuration parameters are summarised in Table 1 below.

Table 1: Experimental configuration of the TechIntel-MAS pipeline

Configuration parameter	**Value**
Sample size	n = 4 patent filings
Subject matter	EP-A1: industrial automation, footwear sole structure, inkjet print diagnostics; DE-A1: motor-vehicle frame (German)
Generative model pool	GPT-5.4, Claude Sonnet 4.6, Gemini 3.1 Pro, DeepSeek R1, Moonshot Kimi K2 Thinking, Qwen3 Max Thinking
Evaluator pool	Grok 4, Llama 4 Maverick (both via OpenRouter)
Search backends	Exa Search and Brave Search APIs
Translation agent	Claude Haiku 4.5 (invoked only for non-English filings)

Configuration parameter	Value
Mean wall-clock time per run	54.7 min (range: 43.5 to 61.5 min)
Mean corpus size per run	136 web items, deduplicated by URL
Mean search directives per run	9.5
Translation cost (DE to EN)	< $0.02 per patent
Total LLM cost (instrumented run)	$3.02

Note. Translation agent was invoked only when the source filing was non-English. Cost and elapsed-time figures at per-model granularity are available only for the single fully instrumented run (EP_4345618_A1, Siemens).

4.1 Per-Model Analytic Quality

Table 2 presents the aggregated performance of each generative model across all four runs, ranked by mean F1. A first reading of the table might suggest that Qwen3 Max Thinking achieved the strongest result, given its perfect mean precision of 1.000, a value no other model approached. A closer reading, however, indicates that this figure is largely a consequence of output strategy rather than evidence of superior factual accuracy, and the contrast between Qwen and Claude Sonnet 4.6 illustrates a precision-recall trade-off.

The pattern holds across the full sample. Qwen emitted exactly the prompt's lower bound of six findings on every run and never produced a single no-signal finding (zero across four patents, against eight for GPT-5.4, six for Gemini 3.1 Pro, four for DeepSeek R1, three for Moonshot Kimi K2 Thinking, and two for Claude Sonnet 4.6). Reasoning-tuned models in the pool (Qwen, DeepSeek R1, Moonshot Kimi K2 Thinking, Gemini 3.1 Pro) systematically produced fewer findings and higher precision than the instruction-tuned, coverage-oriented models (Claude Sonnet 4.6, GPT-5.4). The chain-of-thought reasoning phase appears to function as a pre-emission filter, discarding candidate claims that lack sufficient corpus support before they reach JSON output. The cost of this filter is recall: Qwen's mean recall of 0.059 is among the lowest in the experiment, about a third of Claude's 0.178, indicating that Qwen leaves the great majority of the retrieved corpus uncited.

Table 2: Per-model performance aggregated across four patent runs

Model	Wins	F1	Prec.	Recall	Cosine	KIQ	Conf.	ECE	Cost (USD)	Elapsed (s)
Claude Sonnet 4.6	2	0.296 ± 0.04	0.901	0.178	0.701	0.308	0.801	0.100	$0.75	291
GPT-5.4	0	0.245 ± 0.04	0.791	0.146	0.647	0.313	0.839	0.056	$0.68	137
Moonshot Kimi K2	0	0.127 ± 0.10	0.672	0.073	0.367	0.243	0.828	0.270	$0.29	745
Qwen3 Max Thinking	2	0.110 ± 0.04	1.000	0.059	0.447	0.339	0.813	0.187	$0.10	273
Gemini 3.1 Pro	0	0.108 ± 0.03	0.742	0.058	0.605	0.311	0.911	0.214	$0.38	351
DeepSeek R1	0	0.093 ± 0.02	0.845	0.049	0.445	0.273	0.823	0.177	$0.14	921

Note. Wins = number of runs in which the Judge agent declared this model the winner (out of 4). F1, Precision (Prec.), Recall, and Cosine are means across all four runs. KIQ = mean KIQ-relevancy score; Conf. = mean self-reported finding-level confidence; ECE = expected calibration error (lower is better). Cost and Elapsed are from the single instrumented run only. Values in bold indicate the per-column optimum.

Wall-clock completion time ranged from 43.5 to 61.5 minutes across the four runs (mean 54.7 min). The dominant latency source was the per-model Analyst correction loop, executed serially across the six generators. On the single fully instrumented run (patent EP_4345618_A1, Siemens), total LLM expenditure amounted to

$3.02, distributed unevenly across the model pool. Claude Sonnet 4.6 ($0.75) and GPT-5.4 ($0.68) together accounted for nearly half of the spend (47%); each model contributes the same single analysis and single report per run as every other generator in the pool, so the asymmetry reflects per-token pricing and verbosity rather than workload. Qwen3 Max Thinking ($0.10) was the most cost-efficient generator.

4.2 Agentic Web Search and Corpus Quality

Across the four runs, the Collector agent issued 38 search directives and retrieved 543 web items in total (Exa: 326; Brave: 217). After URL-level deduplication, 401 unique domains contributed at least one item, indicating broad bibliographic diversity rather than concentration on a small number of high-frequency hosts. The patent-to-corpus semantic relevancy, namely the cosine similarity computed by sentence-transformers/all-MiniLM-L6-v2 between the patent's full text (translated to English where applicable) and each retrieved web item, varied substantially across the four filings, as shown in Table 3 below.

Table 3: Patent-to-corpus semantic relevancy by filing

Patent	Subject matter	Mean relevancy	Max relevancy
EP_4345618_A1	Industrial automation: virtual control units	0.287	0.550
EP_4721616_A1	Footwear sole structure	0.164	0.321
EP_4714262_A1	Inkjet print diagnostics	0.134	0.303
DE102024130768A1	Motor-vehicle frame (German source)	0.009	0.126

Note. Relevancy is the cosine similarity between the full patent text (translated to English where applicable) and each retrieved web item, averaged across all items in the run corpus. The bold value (0.009) flags the anomalous result discussed in the text.

The three EP patents yielded mean relevancy values in the range 0.134 to 0.287, reflecting adequate thematic alignment between the query vocabulary derived from claim language and the available web literature. The German DE-A1 filing (BMW, motor-vehicle frame construction) yielded a mean relevancy of 0.009, two orders of magnitude below the EP median. Inspection of the retrieval signals confirmed that the translation step succeeded at the surface level: Claude Haiku 4.5 produced grammatical English for both the title (Kraftfahrzeugrahmen rendered as "motor vehicle frame") and the body of the claims. The resulting vocabulary, however, was too generic to distinguish the patent's specific structural contribution from the broader body of general automotive literature. Translation quality is therefore a necessary but insufficient condition for downstream retrieval quality. Non-English filings are likely to benefit from supplementary search-directive strategies that anchor queries to the assignee identity (here, Bayerische Motoren Werke Aktiengesellschaft) and to the IPC subclass (here, B62D 21/02), rather than relying solely on translated natural-language claim vocabulary. The current Planner prompt requires at least one assignee-anchored directive when an assignee is parseable, but the BMW patent's INID-code formatting did not yield a clean assignee field on the first pass; subsequent runs would benefit from a parser-side improvement.

The Exa and Brave channels behaved differently. Exa retrieved more items per run on average (mean 81 versus Brave's 54), while Brave produced higher per-channel domain diversity (0.84 unique domains per result versus 0.69 for Exa). On the Siemens patent, the two channels achieved nearly identical patent-to-corpus relevancy (Brave: 0.288; Exa: 0.287); on the remaining three patents, Exa was marginally higher. The two backends are therefore complementary rather than substitutable: Exa contributes volume and slightly higher topical relevancy on average, while Brave contributes domain diversity that reduces the risk of the corpus collapsing onto a small number of high-frequency hosts.

4.3 Cross-model Agreement

Pairwise comparison of report texts produced two patterns. The Claude Sonnet 4.6 and GPT-5.4 reports had the highest semantic similarity in three of the four runs (mean cosine 0.79), which suggests that these two models converge on the same narrative when given identical evidence. The Jaccard overlap of cited URLs between any two models, by contrast, rarely exceeded 0.50, so even when the narrative converged the citation choices often diverged. The strongest evidence-overlap pairs were GPT-5.4 and Claude Sonnet 4.6 on the Siemens patent (Jaccard 0.58) and Gemini 3.1 Pro and DeepSeek R1 on the footwear patent (Jaccard 0.56).

4.4 Limitations

The conventional precision definition treats no-signal findings identically to citation failures. Reported precision values therefore conflate two methodologically distinct phenomena, and Qwen's perfect precision is partly an artefact of declining to issue no-signal findings rather than evidence of superior citation accuracy. A second limitation concerns evaluator reliability. The Llama 4 Maverick evaluator returned schema-invalid JSON on roughly 50% of grounding-mode invocations, and the Verifier fell back silently to its mock implementation, which degraded effective evaluator independence for about half of all grounding checks. The sample is also small. With four patents the evaluation is best read as a pilot, and the planned extension to ten filings will support more stable model-level estimates and inferential rather than descriptive comparisons.

5. Conclusion

This study showed that a multi-agent system can link the technical content of patent documents to market evidence drawn from the open web. Although the evaluation is a pilot, the results indicate that an autonomous pipeline can reproduce the full intelligence cycle for technology-oriented information. The structured outputs integrate readily with knowledge bases and support knowledge management within an organisation. Human oversight nonetheless remains essential for validating individual outputs. The dense and technical language of patents also proved a key factor in the quality of the results, particularly for non-English filings.

Ethics declaration: The author declares the study does not include any personal or sensitive data.

Declaration on the Use of Generative AI: In preparing this manuscript the author used Anthropic Claude (Sonnet 4.6) as a writing-assistance tool for language editing, structural revision, and consistency checking across sections (for defined terms, table numbering and model lists). The MAS system itself, which is the object of study in the paper, is described in the Methodology section including the analytical outputs. The author takes full responsibility for the content of this manuscript.

References

Alavi, M., & Leidner, D. E. (2001). Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Quarterly, 25*(1), 107–136.

Alavi, M., Leidner, D. E., & Mousavi, R. (2024). A knowledge management perspective of generative artificial intelligence. *Journal of the Association for Information Systems, 25*(1).

An, J., Kim, K., Mortara, L., & Lee, S. (2018). Deriving technology intelligence from patents: Preposition-based semantic analysis. *Journal of Informetrics*, *12*(1), 217–236. https://doi.org/10.1016/j.joi.2018.01.001

Behkami, N. A., & Daim, T. U. (2012). Research forecasting for health information technology (HIT), using technology intelligence. *Technological Forecasting and Social Change*, *79*(3), 498–508. https://doi.org/10.1016/j.techfore.2011.08.015

Cohen, W. M., & Levinthal, D. A. (1990). Absorptive capacity: A new perspective on learning and innovation. *Administrative Science Quarterly, 35*(1), 128–152.

Csaszar, F. A., Katkar, H., & Kim, H. (2024). Artificial intelligence and strategic decision-making: Evidence from entrepreneurs and investors. *Strategy Science*. Advance online publication.

Gu, J., et al. (2026). A survey on LLM-as-a-judge. *The Innovation*.

Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N. V., Wiest, O., & Zhang, X. (2024). Large language model based multi-agents: A survey of progress and challenges. *arXiv*. arXiv:2402.01680.

He, X., et al. (2025). Evolving knowledge management: Artificial intelligence and the dynamics of social interactions. *IEEE Engineering Management Review*.

Kerr, C. I. V., Mortara, L., Phaal, R., & Probert, D. R. (2006). A conceptual model for technology intelligence. *International Journal of Technology Intelligence and Planning*, *2*(1), 73–93. https://doi.org/10.1504/IJTIP.2006.010511

Lichtenthaler, E. (2003). Third generation management of technology intelligence processes. *R&D Management*, *33*(4), 361–375. https://doi.org/10.1111/1467-9310.00304

Lichtenthaler, E. (2004a). Coordination of technology intelligence processes: A study in technology intensive multinationals. *Technology Analysis and Strategic Management*, *16*(2), 197–221. https://doi.org/10.1080/0953732041000168289 2

Lichtenthaler, E. (2004b). Technological change and the technology intelligence process: A case study. *Journal of Engineering and Technology Management*, *21*(4), 331–348. https://doi.org/10.1016/j.jengtecman.2004.09.003

Lichtenthaler, E. (2004c). Technology intelligence processes in leading European and North American multinationals. *R&D Management*, *34*(2), 121–135. https://doi.org/10.1111/j.1467-9310.2004.00328.x

Lichtenthaler, E. (2005). The choice of technology intelligence methods in multinationals: Towards a contingency approach. *International Journal of Technology Management*, *32*(3–4), 388–407. https://doi.org/10.1504/ijtm.2005.007341

Mortara, L., Kerr, C. I. V., Phaal, R., & Probert, D. R. (2009). Technology Intelligence practice in UK technology-based companies. *International Journal of Technology Management*, *48*(1), 115–135. https://doi.org/10.1504/IJTM.2009.024603

Mortara, L., Thomson, R., Moore, C., Armara, K., Kerr, C., Phaal, R., & Probert, D. (2010). Developing a technology intelligence strategy at Kodak European Research: Scan & target. *Research Technology Management*, *53*(4), 27–38. https://doi.org/10.1080/08956308.2010.11657638

Nguyen, T., Chin, P., & Tai, Y.-W. (2025). MA-RAG: Multi-agent retrieval-augmented generation via collaborative chain-of-thought reasoning. *arXiv*.

Nonaka, I. (1994). A dynamic theory of organizational knowledge creation. *Organization Science, 5*(1), 14–37.

Norling, P. M., Herring, J. P., Rosenkrans, W. A., Stellpflug, M., & Kaufman, S. B. (2000). Putting competitive technology intelligence to work. *Research Technology Management*, *43*(5), 23–28. https://doi.org/10.1080/08956308.2000.11671377

Pelaez, S., et al. (2023). Large-scale text analysis using generative language models: A case study in discovering public value expressions in AI patents. *Quantitative Science Studies, 4*(4).

Pletsch, A. L. B., Tonial, G., Matos, F., & Koch, L. L. (2026). Digital transformation: The role of generative AI in the evolution of knowledge management systems. *Journal of Knowledge Management*.

Porter, A. L. (2005). QTIP: Quick technology intelligence processes. *Technological Forecasting and Social Change*, *72*(9), 1070–1081. https://doi.org/10.1016/j.techfore.2005.05.002

Rohrbeck, R., Heuer, J., & Arnold, H. (2006). *The Technology Radar—An instrument of technology intelligence and innovation strategy*. *2*, 978–983. https://doi.org/10.1109/ICMIT.2006.262368

Savioz, P. (2003). *Technology intelligence: Concept design and implementation in technology based SMEs*. Springer. https://doi.org/10.1057/9781403948212

Singh, A., Ehtesham, A., Kumar, S., Talaei Khoei, T., & Vasilakos, A. V. (2025). Agentic retrieval-augmented generation: A survey on agentic RAG. *arXiv*. arXiv:2501.09136.

Veugelers, M., Bury, J., & Viaene, S. (2010). Linking technology intelligence to open innovation. *Technological Forecasting and Social Change*, *77*(2), 335–343. https://doi.org/10.1016/j.techfore.2009.09.003

Wataoka, K., et al. (2024). Self-preference bias in LLM-as-a-judge. *arXiv*.

Yoon, B. (2008). On the development of a technology intelligence tool for identifying technology opportunity. *Expert Systems with Applications*, *35*(1–2), 124–135. https://doi.org/10.1016/j.eswa.2007.06.022

Yoon, J., & Kim, K. (2012). TrendPerceptor: A property-function based technology intelligence system for identifying technology trends from patents. *Expert Systems with Applications*, *39*(3), 2927–2938. https://doi.org/10.1016/j.eswa.2011.08.154

Zhou, L., Yan, S., Li, Z., & Ma, J. (2024). Exploring the application of retrieval-augmented generation technology in defense technology intelligence. In *2024 International Annual Conference on Complex Systems and Intelligent Science (CSIS-IAC)*. IEEE. https://doi.org/10.1109/CSIS-IAC63491.2024.10919351