PLM: Large Language Models for Patent Concept Generation and Innovation Support

Chia-Te Huang
Department of Computer Science and Information Engineering
Fu Jen Catholic University, New Taipei City, Taiwan
414085090@fjucloud.onmicrosoft.com

Abstract

Generating and evaluating patent text is constrained by the complexity of legal knowledge. General-purpose large language models (LLMs) can draft a fluent claim in one pass, but single-pass generation cannot check or improve itself against what patent drafting rewards—coherent linkage between claim elements and consistent terminology. We present PLM (Patent Large Language Model), which uses an LLM as an iterative search-and-revise agent rather than a one-shot drafter: claim revision is framed as a Monte Carlo Tree Search (MCTS*) problem. A process reward model (PRM) scores every intermediate step, and a knowledge-graph agent (KGAgent) anchors each revision to the claim's original technical structure and flags dropped features. On a benchmark of Taiwanese utility-model patents, scored with a Step-wise Claim Quality Score (SCQS)—our search-time application of the completeness/clarity/consistency/linkage rubric of Jiang et al. (2025)—with BERTScore as a secondary fidelity diagnostic, PLM improves the composite SCQS by 11.9% over a GPT-4o baseline (8.21 vs. 7.34) and BERTScore by 3.9 points (0.891 vs. 0.852), with the largest gains on consistency (15.01%) and linkage (14.78%). A compute-matched ablation shows the gain comes mainly from revisions compounding across an iterative, depth-wise search, not from any particular node-selection rule.

Keywords: patent claim revision, large language models, Monte Carlo tree search, process reward model, knowledge graph

1 Introduction

Patents not only grant inventors a period of exclusive rights but also serve as a key source of knowledge throughout research, development, and technological innovation. Patent text is nonetheless long, highly domain-specific, and legally rigorous, which makes it difficult to analyze and reuse. Keyword-based retrieval cannot capture the semantic complexity unique to patent documents, and most existing NLP methods rely on hand-crafted rules or shallow semantic models that fail to capture implicit technical relationships.

Drafting patent documents is time-consuming and costly. Claims, as the legal core of a patent, must be written with high precision and a specific linguistic structure that clearly delineates the scope of protection while avoiding overlap with prior art. Post-grant claim revision—for example, preparing a claim set for licensing, defending validity in an invalidation action, or filing a continuation—involves fine-grained adjustments to claim scope, a process that has traditionally relied entirely on human judgment and is consequently slow and error-prone. This is the setting we target: the dataset in Section 4.1 consists entirely of already-granted claims, and the system does not perform examination-time prosecution.

Taiwan's Intellectual Property Office (TIPO) publishes more than 50,000 utility-model patents annually. A well-crafted independent claim must balance breadth against precision: too narrow and it is easily designed around; too broad and it risks rejection for lack of novelty or inventiveness. A case-summary review of 30 patent administrative judgments handed down by Taiwan's Intellectual Property and Commercial Court between 2020 and 2026 found inventive-step determination to be the central issue in 19 cases (63%), far outweighing all other dispute types combined—illustrating that claim quality carries real stakes. We note, however, that Taiwanese utility-model patents are granted under a formal-examination regime that does not assess novelty or inventive step pre-grant (Taiwan Patent Act, Art. 112; see also Arts. 97, 116–117), and PLM itself does not claim to reproduce an inventive-step judgment; the statistic above is offered only as general motivation for why claim quality matters, and as motivation for future work

Proceedings of the 38th Conference on Computational Linguistics and Speech Processing (ROCLING 2026), pages 284–292
November 6-7, 2026 ©2026 Association for Computational Linguistics

on examination-time drafting support. TIPO's formal-examination requirements do impose constraints on quantitative terminology, open-ended claim language, and hierarchical dependent-claim structure, and manual drafting by patent agents remains costly and uneven in quality.

A further motivation for working in Chinese specifically is linguistic and infrastructural: Chinese claim-drafting conventions (classifier and quantifier usage, antecedent-basis marking) differ from English ones, and Chinese-language patent-drafting datasets remain comparatively scarce next to the English- and USPTO-centred resources most prior work relies on. Chinese-language, Taiwanese-utility-model claim revision is therefore a distinct setting to which LLM techniques have yet to be fully applied, and we treat it as a first-class contribution rather than an incidental choice of dataset.

Our core idea is to change how the LLM is used: rather than generating a claim in one pass, PLM lets the LLM search for a revision, proposing, scoring, and revising candidates until the search converges, drawing on ReST-MCTS* (Zhang et al., 2024) and MCTS-RAG (Hu et al., 2025b). A process reward model (PRM) scores every intermediate step rather than only the final output, and a knowledge-graph agent (KGAgent) anchors each revision to the original component relationships, checking specifically for features a revision has *dropped* (KGAgent does not detect newly added, scope-narrowing limitations; Section 3.2). Our contributions are:

- We recast the completeness/clarity/consistency/linkage rubric of Jiang et al. (2025) as a step-wise search reward (SCQS, Section 3.3) used to score intermediate search states, rather than proposing a new metric.
- We build an MCTS*-based search-and-revise engine combining step-wise PRM scoring with KGAgent's structural checks, avoiding the sparse-reward problem of scoring only a finished output. The ablation in Section 4.3 shows the benefit comes from the iterative, depth-compounding search itself: replacing UCB1*-guided selection with a random rule at the same budget produces no measurable difference, so we do not claim node-selection as a contribution.
- We apply this engine to Chinese-language Taiwanese utility-model claims and release a benchmark test set of annotated claims drawn from TIPO's public database.

2 Related Work

Traditional patent analysis relies mainly on keyword retrieval, citation analysis, and bibliometrics, which struggle with limited semantic understanding, high manual cost, and cross-domain knowledge integration, and typically depend on predefined classification schemes that cannot capture the dynamic evolution of emerging technology fields.

Knowledge fine-tuning and knowledge graphs. Ren et al. (2025a) propose a knowledge fine-tuning (KFT) framework—knowledge-injection pre-training, domain-specific supervised fine-tuning, and reinforcement learning from human feedback—to endow an LLM with the domain knowledge needed for patent concept generation. Knowledge graphs represent domain knowledge as nodes and relations and can support reasoning and decision-making (Mo et al., 2025). Early patent knowledge-graph construction relied on syntactic and semantic rules to structure engineering knowledge as entity–relation–entity triples (Ibrahim et al., 2024), whose survey frames this as one of three paradigms for integrating LLMs and KGs; KGAgent (Section 3.2) falls under this paradigm, using a knowledge structure to constrain and check an LLM's output rather than using an LLM to build the structure. Methods combining graph neural networks for entity alignment and cross-source fusion have also been surveyed (Bian, 2025); we cite this for the general KG-construction problem it surveys, since KGAgent performs simpler, within-document triple extraction rather than cross-source alignment.

Monte Carlo Tree Search for reasoning. Tree of Thoughts (Yao et al., 2023) replaces step-by-step generation with heuristic search, though its evaluation function still requires extra LLM calls; RAP (Hao et al., 2023) combines a world model with MCTS and achieves near-human performance on GSM8K, a grade-school math word-problem benchmark. ReST-MCTS* (Zhang et al., 2024) uses process-reward modeling to guide node expansion along mathematical reasoning chains, and MCTS-RAG (Hu et al., 2025b) integrates retrieval at every node. The tree-search paradigm traces back to UCB1-based bandit planning (Kocsis and Szepesvári, 2006) and was popularized through super-human game play (Silver et al., 2016); recent frameworks similarly search over workflows (Zhang et al., 2025) or iteratively refine a draft with

	ToC (Yu et al., 2025)	PLM
Dataset	1,145 English claims	Chinese TW utility models
Search	Tree search, multi-agent	PRM-scored MCTS*
Reward	Multi-objective	Single scalar (SCQS)
Grounding	Multi-agent critique	KGAgent anti-omission check
Reported gain	∼8–9%	11.9% (not directly comparable)

Table 1: Qualitative comparison with the closest prior work.

self-feedback (Madaan et al., 2023; Fernando et al., 2024).

Yu et al. (2025) apply a tree-of-claims search (ToC) with multi-agent LLMs to patent claim generation. ToC was posted on 21 November 2025, roughly nine months before this submission, and is prior, not concurrent, work. ToC is architecturally close to PLM—both combine an MCTS-style search with an editor/examiner split and a multi-criterion reward, and report a composite gain of similar magnitude—so we position it directly rather than in passing. Table 1 compares the two at the architecture level; we were not able to run ToC's pipeline on our own dataset, so the comparison is qualitative, and a direct quantitative comparison is left to future work. The differences we can state with confidence: dataset and language (Chinese-language Taiwanese utility-model claims rather than ToC's 1,145 English-language claims); KGAgent is used specifically as an anti-omission check rather than a general examiner critique; and PLM's reward is a single step-wise scalar computed per candidate, whereas ToC's is explicitly multi-objective. We do not claim the search-and-revise idea itself is novel relative to ToC; our contribution is the narrower one stated in Section 1.

PLM uses the PRM score as a built-in evaluation function, avoiding extra LLM calls and reducing inference latency relative to methods like ToT that call an LLM to evaluate every candidate.

Retrieval-augmented generation. RAG (Lewis et al., 2020) combines dense retrieval with a generative model; DPR (Karpukhin et al., 2020) uses a dual encoder to align questions and passages. This line of work is dominated by short-text QA, and multi-granularity understanding of long documents such as patent specifications remains limited; patent retrieval additionally requires handling IPC codes, technical language, and cross-lingual queries. Retrieval-augmented pipelines have also been applied directly to IP text, e.g. synthetic multi-angle fine-tuning (Ren et al., 2025b). KGAgent's use of structured triples extracted from the original patent (Section 3) is a complementary, non-retrieval route to the same goal: grounding a revision in evidence external to the LLM's own generation.

Process reward models. Lightman et al. (2023) show that process reward models (PRMs) outperform outcome reward models (ORMs): on a representative subset of MATH, their best process-supervised model solves 78% of problems under best-of-N selection, and their released PRM800K step-level labels let a reward model localize the specific step where a solution's reasoning first goes wrong, rather than only judging the final answer. We adopt a similar step-scoring design, extending the scoring space to the four criteria in Section 3.3. Uesato et al. (2022) separately compare ORM and PRM generalization on multi-step arithmetic and find PRM generalizes better out of distribution.

LLMs in the patent domain. Prior work spans prior-art search (Xiong et al., 2025), patent classification, drafting assistance—including agentic specification drafting (Shea and Yu, 2025), instruction-tuned text editing (Raheja et al., 2023), and multimodal generation of figure descriptions (Shukla et al., 2025; Aubakirova et al., 2023; Zhu et al., 2024)—technology-trend analysis (Lim et al., 2025), and value assessment; key decisions on patentability, scope, and prior-art rejection still require human judgment. Zuo et al. (2024) build an error taxonomy specific to patent generation, finding that generic text-generation metrics do not capture patent-specific errors such as inconsistent antecedent basis or dropped features—consistent with our own view of the judge model as a possible source of error rather than ground truth. LLM adaptation to specialized legal and technical text also extends to factuality-tuned legal QA (Hu et al., 2025a) and retrieval-augmented recommendation in traditional Chinese medicine (Qin et al., 2024).

3 Method

PLM uses a four-layer architecture—data-and-knowledge, retrieval-and-reasoning, scoring-and-search-control, and editing-and-output—connected

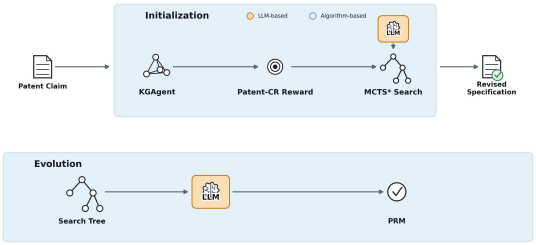

Figure 1: PLM's architecture: initialization (KGAgent, SCQS reward, MCTS* search) and evolution (PRM-guided scoring). The reward box in the image, labelled "Patent-CR Reward", is the SCQS reward of Section 3.3.

through well-defined interfaces. Given a claim or specification draft, the first stage performs text parsing, knowledge-graph construction, and baseline scoring; the second generates revision strategies and computes the reward; the third runs tree search. Three components work together: the **MCTS* search engine**, the **process reward model (PRM)**, and the **knowledge-graph agent (KGAgent)**.

Given a claim and one of six search modes (balanced, completeness-, clarity-, linkage-, terminology-, or scope-oriented), the system runs I expansion iterations to a maximum depth D.[1] Figure 1 shows the overall architecture: initialization builds the knowledge graph and search tree, and the evolution stage scores and prunes the tree with the PRM. The generator ("EditorAgent") is GPT-4o in every condition reported here, including the zero-shot GPT-4o row of Table 3, which differs from PLM only in that search, the PRM, and KGAgent are removed. PLM is thus best read as GPT-4o wrapped in a search-and-revise loop, not a different base model.

3.1 PRM-Guided MCTS* Search

Node selection follows a UCB1*-style criterion adapted for process rewards:

$$\mathrm{UCB1}^*(v) = \bar{Q}(v) + \beta\sqrt{\frac{\ln N(\mathrm{parent})}{N(v)}} \quad (1)$$

where $\bar{Q}(v)$ is the mean PRM score along the path to v (cumulative score over visit count), $N(v)$ is the visit count, and β is an exploration weight (default 0.4). During expansion, the LLM generates k candidate revisions; each is scored by the PRM, and only the best becomes a single child node (Algorithm 1). Because only one child is kept per expansion, the search tree as implemented has width one at every depth; Section 4.3 confirms this empirically—replacing UCB1*-guided selection with uniformly random selection, at the same budget, is statistically indistinguishable from the original. We therefore do not claim the specific selection rule in Equation 1 as a contribution: what matters is the iterative structure itself—propose, score, and condition the next proposal on the best result so far.

[1]These modes reweight emphasis across the four criteria actually scored (Section 3.3); a scope- or novelty-oriented mode only steers the generator's own prompt, since no prior-art signal reaches the judge.

Algorithm 1 PRM-guided expansion

```
root ← original claim; score with PRM
for i = 1 to I do
    v ← node maximizing Eq. 1 along the single-child chain
    if depth(v) < D then
        generate k candidate revisions of v
        score each with the PRM; attach the best as v's child
    end if
end for
return the highest-scoring node visited
```

3.2 KGAgent

KGAgent parses the input specification and claims to extract up to 15 structured triples per patent, restricted to four relation types: component-dependency, component-connection, function-correspondence, and parameter-constraint. These triples act as structural constraints during revision, checking whether a revised claim omits a technical feature disclosed in the original specification but not covered by the claim. KGAgent detects only this kind of omission, not the reverse case of a revision that *adds* a new, scope-narrowing limitation; an "improved" SCQS score should be read as "no disclosed feature dropped and all four criteria improved," not as a guarantee that scope was preserved.

3.3 The SCQS Composite Reward

We score every candidate with a Step-wise Claim Quality Score (SCQS): our name for a search-time, step-wise application of the four-criterion rubric of Jiang et al. (2025). We adopt their four criteria and weights unmodified; our contribution is using the rubric to score intermediate search states rather than only a finished draft. Each sub-score is judged on a 1–10 scale: completeness (Co, whether essential disclosed features are retained, checked against KGAgent's extracted components), clarity

(Cl, penalizing vague functional language and rewarding quantitative, antecedent-grounded claims), consistency (Cn, whether terminology names the same feature identically throughout), and linkage (L, structural coherence of component links per KGAgent's triples). The composite is

$$\text{SCQS} = \frac{4\,Co + 2\,Cl + 2\,Cn + 3\,L}{11} \qquad (2)$$

with each criterion on a 1–10 scale, so the composite likewise ranges over approximately $[1, 10]$ (e.g. PLM's 8.21 in Table 3). The weights are Jiang et al.'s: completeness is weighted highest since a claim missing an essential feature may be invalid; linkage next, since incoherent linkage is a common basis for indefiniteness objections; clarity and consistency lowest, since both affect drafting quality more than validity. The PRM's own step-level aggregate is computed on a separate $[0, 1]$ scale and should not be confused with the 1–10 SCQS scale in Section 4; all values there are means over 150 held-out test patents.

BERTScore is reported throughout as a secondary fidelity diagnostic rather than a second quality metric: it is computed against three fixed novelty/inventive/validity reference passages, so a higher score means closer to those references, not merely less changed from the original. We treat it as secondary for three reasons: Jiang et al. (2025) themselves report that automated metrics including BERTScore correlate only weakly with human judgment on the consistency dimension, which is where our largest gain is concentrated; our own ablation (Section 4.3) finds BERTScore moves in the same direction as SCQS but at roughly a tenth the magnitude; and a per-patent breakdown by IPC section finds BERTScore concentrated in a band only 0.0042 wide, with limited discriminative power (Appendix A).

4 Results

4.1 Experimental Setup

Our dataset is drawn from TIPO's public database: 1,000 utility-model patents granted in December 2025. Table 2 shows the IPC-section distribution; the most frequent subclasses are G06F (64 patents), H01L (49), and G06Q (45). The data is split 70/15/15 into train/validation/test (150 test patents), keeping patents from the same applicant on one side of the split. Claims are annotated along the four SCQS dimensions to serve as ground truth.

IPC Category	Count	Share
G (Physics)	272	27.2%
H (Electricity)	242	24.2%
B (Operations; Transporting)	166	16.6%
A (Human Necessities)	130	13.0%
C (Chemistry; Metallurgy)	84	8.4%
F (Mechanical Engineering)	81	8.1%
E (Fixed Constructions)	20	2.0%
D (Textiles; Paper)	5	0.5%

Table 2: IPC category distribution of the 1,000-patent dataset.

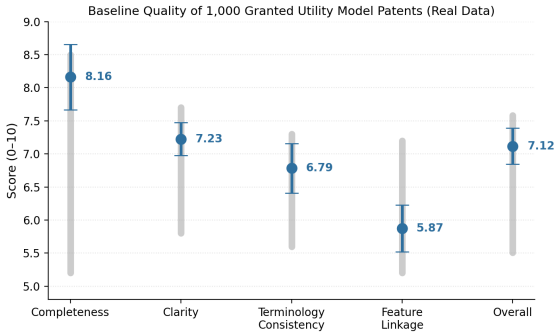

Figure 2: Pre-revision quality-score distribution (mean ± 95% CI) of the 1,000 granted utility-model patents.

The control condition is GPT-4o in zero-shot mode. Table 3's main comparison uses $I = 5$, $D = 3$; the search-procedure ablation in Section 4.3 instead uses $I = 8$ to match the budget of its flat and sequential baselines, and the two should not be read as the same run (full configuration details are in Appendix A). Figure 2 shows the pre-revision quality distribution of the 1,000 granted patents, the reference point against which gains are reported.

4.2 Evaluation and Performance

PLM outperforms all four baseline models on every SCQS dimension and on the composite (Table 3), improving the composite by 11.9% and BERTScore F1 by 3.9 points over GPT-4o. The baseline ranking broadly tracks model scale: GPT-4o leads, then Llama 3.1 70B, with the two smaller open-source models well behind—indicating that, absent domain-specific mechanisms, model scale remains an important factor.

Relative to GPT-4o, the largest gains are on consistency (15.01%) and linkage (14.78%); completeness (10.05%) and clarity (8.00%) gain less. This tracks where the architecture concentrates its effort: knowledge-graph triples target linkage directly, while the PRM's step-wise terminology feed-

Model	Completeness	Clarity	Consistency	Linkage	SCQS	BERTScore
Qwen2-Instruct 7B	6.31	6.17	6.04	5.68	6.05	0.803
GLM-4-Chat 9B	6.48	6.39	6.25	5.94	6.27	0.808
Llama 3.1 70B	7.20	7.11	6.95	6.64	6.98	0.846
GPT-4o	7.56	7.50	7.33	6.97	7.34	0.852
PLM	**8.32**	**8.10**	**8.43**	**8.00**	**8.21**	**0.891**

Table 3: PLM vs. baseline models on the four SCQS dimensions, the SCQS composite, and BERTScore F1.

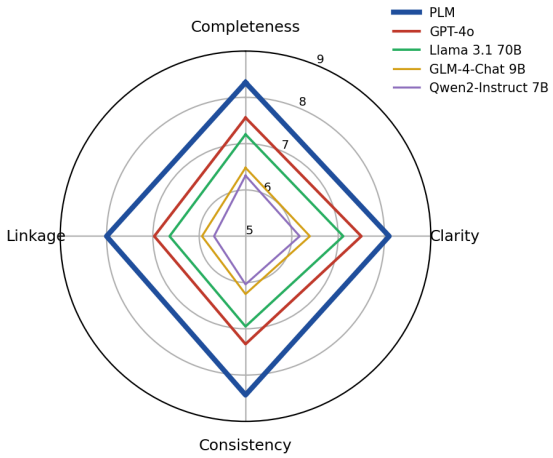

Figure 3: Radar comparison of the four SCQS dimensions across PLM and four baseline models.

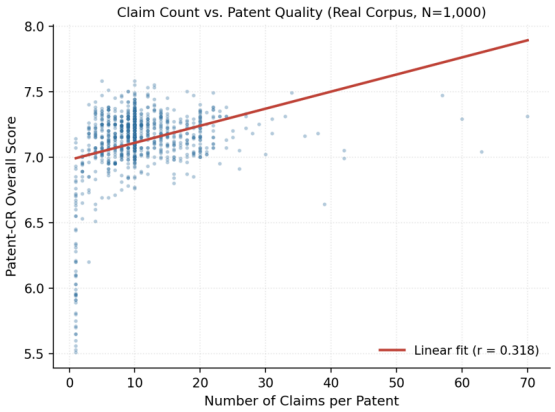

Figure 4: Claim count vs. quality score across the 1,000-patent dataset (Pearson $r = 0.318$; the figure retains its original "Patent-CR" axis label, renamed SCQS in the text, Section 3.3).

back targets consistency; completeness and clarity depend more on the underlying model's own drafting ability. Figure 3 visualizes this comparison.

Figure 4 shows claim count against quality across the full 1,000-patent dataset ($r = 0.318$): a positive correlation of this size is consistent with claim count partly proxying for drafting complexity, and with richer independent/dependent-claim hierarchies; without claim length available to compute a partial correlation, we treat $r = 0.318$ as suggestive only. Figure 5 shows the score distribution across the 150 test patents: PLM has both the highest median and the narrowest interquartile range, indicating more consistent revision quality.

4.3 Search-Procedure Ablation

To isolate whether the gain comes from the search budget, the tree structure, or the node-selection rule, Table 4 compares four search procedures on the same engine and the same 150-patent sample (Default uses $I = 8$ to match the budget of Best-of-N and Self-Refine).

Three findings. First, search beats single-pass generation at every budget tested (Default 7.401 vs. Self-Refine k=1's 7.268). Second, tree structure beats flat sampling at matched budget: Default > Best-of-N (7.333) > single-shot, with Self-Refine at k=8 (7.376, sequential, no branching) in between. Third—and this is the finding that revises a claim from Section 1—replacing UCB1*-guided selection with uniformly random selection, at the same budget, scores 7.398, statistically indistinguishable from Default's 7.401 ($\Delta = 0.003$): the depth-compounding search helps, but the specific node-selection rule tested here adds nothing measurable. Worth noting separately: Self-Refine at k=8 has the highest BERTScore of any condition (0.8572), above Default, suggesting MCTS*'s branch-jumping costs a little surface similarity to the reference passages even as it helps SCQS. Appendix A reports a second, independent sweep over search depth, retrieval, and the exploration weight, directionally consistent with this.

Condition	Score	BERT	PRM
Default (I=8)	7.401	0.8515	0.885
Self-Refine (k=1)	7.268	0.8480	0.871
Best-of-N (N=8)	7.333	0.8468	0.886
Random selection	7.398	0.8507	0.885
Self-Refine (k=8)	7.376	0.8572	0.878

Table 4: Search-procedure ablation, $N = 150$ patents.

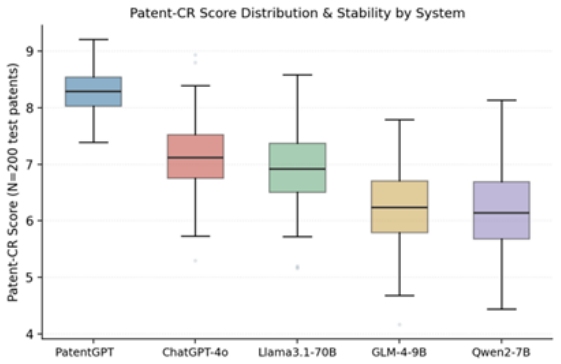

Figure 5: Score distribution and stability by system across the 150 test patents (the figure retains its original "Patent-CR"/"N=200" labels, and "PatentGPT" in the image denotes PLM; the text value is SCQS/N=150, Section 4.1).

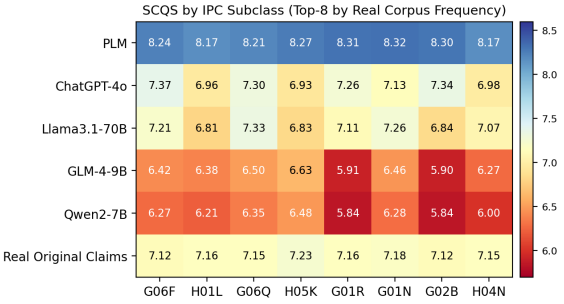

Figure 6: Mean SCQS by IPC subclass for the top-8 subclasses, compared against real original claims.

4.4 Cross-Domain Impact

Figure 6 shows the mean SCQS of five systems across the eight most frequent IPC subclasses, with real original claims as a reference row. PLM leads the original average by roughly 1.0–1.2 points across all eight domains, with a fairly consistent margin. The two smaller open-source baselines fall below the original average in every domain, most sharply in G01R and G02B; ChatGPT-4o and Llama 3.1-70B score slightly above the original average in about half the domains and slightly below in the rest. With only ten to forty samples per domain in the 150-patent test set, this pattern is descriptive rather than conclusive.

One further reading is worth flagging: the judge scores real, already-granted original claims (bottom row, ∼7.1–7.2) at roughly the level of zero-shot GPT-4o's revisions (∼6.9–7.4). This is consistent either with zero-shot GPT-4o already being close to grant-quality, or with the judge not yet being well separated from surface fluency; our data does not distinguish the two, which is itself a reason a human-evaluation study would be valuable.

5 Conclusion

We apply an LLM as an iterative search-and-revise agent, rather than a one-shot drafter, to post-grant revision of Taiwanese utility-model claims, targeting incomplete feature linkage and inconsistent terminology—the two weaknesses that leave single-pass LLM output furthest from what these claims need. Three contributions are validated at the scope stated in Section 1: recasting an existing claim-quality rubric as a step-wise search reward; a search-and-revise engine whose benefit a compute-matched ablation traces to the search structure itself, not node-selection specifics; and an application to Chinese-language Taiwanese utility-model claims with a released benchmark. These show a consistent improvement across a 150-patent held-out sample (11.9% on the SCQS composite relative to GPT-4o); since we have not yet run a significance test over the per-patent deltas, we describe this as consistent rather than statistically significant.

Introducing AI into patent drafting could lower costs and speed up the application process, making it easier for smaller applicants and individual inventors to participate in the patent system, while also potentially affecting roles that have traditionally relied on manual drafting and review—a discussion that belongs alongside closer engagement between AI and IP policy. Future work will extend in four directions: a human-evaluation subset (patent professionals scoring outputs from all five systems, target 100 patents with 2 annotators); a direct quantitative comparison against ToC (Yu et al., 2025); cross-claim dependency analysis together with reinforcement learning to tune the PRM's per-criterion weights; and integration into patent-firm workflows with a human-in-the-loop review interface.

Translation Disclosure

The English text of this manuscript was produced with the assistance of an artificial intelligence (AI) translation tool, which performed the initial translation from the original Chinese draft and subsequent grammatical correction.

References

Dana Aubakirova, Kim Gerdes, and Lufei Liu. 2023. PatFig: Generating short and long captions for patent figures. In *Proceedings of the IEEE/CVF Interna-*

tional Conference on Computer Vision (ICCV) Workshops.

Haonan Bian. 2025. LLM-empowered knowledge graph construction: A survey. ArXiv:2510.20345. Published as a conference paper at ICAIS 2025.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2024. Promptbreeder: Self-referential self-improvement via prompt evolution. In *International Conference on Machine Learning (ICML)*.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of EMNLP*.

Yinghao Hu, Leilei Gan, Wenyi Xiao, Kun Kuang, and Fei Wu. 2025a. Fine-tuning large language models for improving factuality in legal question answering. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4410–4427, Abu Dhabi, UAE. Association for Computational Linguistics.

Yunhai Hu, Yilun Zhao, Chen Zhao, and Arman Cohan. 2025b. MCTS-RAG: Enhancing retrieval-augmented generation with Monte Carlo tree search. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 12581–12597, Suzhou, China. Association for Computational Linguistics.

Nourhan Ibrahim, Samar Aboulela, Ahmed Ibrahim, and Rasha Kashef. 2024. A survey on augmenting knowledge graphs (KGs) with large language models (LLMs): Models, evaluation metrics, benchmarks, and challenges. *Discover Artificial Intelligence*, 4(1):76.

Lekang Jiang, Pascal A. Scherz, and Stefan Goetz. 2025. Patent-CR: A dataset for patent claim revision. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2300–2314, Albuquerque, New Mexico. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of EMNLP*.

Levente Kocsis and Csaba Szepesvári. 2006. Bandit based Monte-Carlo planning. In *European Conference on Machine Learning (ECML)*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step.

Hyunseung Lim, Sooyohn Nam, Sungmin Na, Ji Yong Cho, June Yong Yang, Hyungyu Shin, Yoonjoo Lee, Juho Kim, Moontae Lee, and Hwajung Hong. 2025. PANORAMA: A dataset and benchmarks capturing decision trails and rationales in patent examination. ArXiv:2510.24774.

Aman Madaan, Niket Tandon, Prakhar Gupta, et al. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Belinda Mo, Kyssen Yu, Joshua Kazdan, Joan Cabezas, Proud Mpala, Lisa Yu, Chris Cundy, Charilaos Kanatsoulis, and Sanmi Koyejo. 2025. KGGen: Extracting knowledge graphs from plain text with language models.

Suyang Qin, Yifan Wang, Tangming Cui, Jinge Ma, Xin Zhou, Xinglin Guo, Chuchu Zhang, Chongyun Zhou, Rongjuan Guo, and Haiyan Li. 2024. Intelligent Chinese patent medicine (CPM) recommendation framework: Integrating large language models, retrieval-augmented generation, and the largest CPM dataset. *Pharmacological Research*.

Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. 2023. CoEdIT: Text editing by task-specific instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

Runtao Ren, Jian Ma, and Jianxi Luo. 2025a. Large language model for patent concept generation. *Advanced Engineering Informatics*, page 103301.

Runtao Ren, Jian Ma, and Jianxi Luo. 2025b. Retrieval-augmented generation systems for intellectual property via synthetic multi-angle fine-tuning. ArXiv:2506.00527.

Ryan Shea and Zhou Yu. 2025. AutoSpec: An agentic framework for automatically drafting patent specification. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 12829–12840, Suzhou, China. Association for Computational Linguistics.

Shreya Shukla, Nakul Sharma, Manish Gupta, and Anand Mishra. 2025. PatentLMM: Large multimodal model for generating descriptions for patent figures. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(19):20488–20496.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, pages 484–489.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback.

Qiushi Xiong, Zhipeng Xu, Zhenghao Liu, Mengjia Wang, Zulong Chen, Yue Sun, Yu Gu, Xiaohua Li, and Ge Yu. 2025. Enhancing the patent matching capability of large language models via the memory graph. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 337–347, Padua, Italy. Association for Computing Machinery.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Shuyang Yu, Jianan Liang, and Hui Hu. 2025. ToC: Tree-of-claims search with multi-agent language models. ArXiv:2511.16972. Columbia University; AIGROW. Posted 21 November 2025.

Dan Zhang, Sining Zhoubian, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024. ReST-MCTS*: LLM self-training via process reward guided tree search.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xiong-Hui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. 2025. AFlow: Automating agentic workflow generation. In *International Conference on Learning Representations (ICLR)*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2024. MiniGPT-4: Enhancing vision-language understanding with advanced large language models.

You Zuo, Kim Gerdes, Éric Clergerie, and Benoît Sagot. 2024. PatentEval: Understanding errors in patent generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2687–2710, Mexico City, Mexico. Association for Computational Linguistics.

Explanation of the Title Change

The name PatentGPT is identical to that used in the cited literature (Ren et al., 2025a); it has therefore been changed to "PLM: Large Language Models for Patent Concept Generation and Innovation Support" (Patent Large Language Model).

Config.	Depth	Score	PRM
Default	3	7.413	0.884
No retrieval	3	7.381 (-0.4%)	0.790 (-10.7%)
Shallow	1	7.330 (-1.1%)	0.888 ($+0.4\%$)
Deep	5	7.396 (-0.2%)	0.886 ($+0.1\%$)
$\beta = 0$	3	7.413 ($+0.0\%$)	0.884 (-0.1%)
Best depth	4	7.438 ($+0.3\%$)	0.886 ($+0.1\%$)

Table 5: Depth/retrieval/exploration-weight sweep, $N = 150$.

A Reproducibility and Implementation Notes

This appendix provides reproducibility details together with an additional ablation.

Models and search budget. The generator (EditorAgent) is GPT-4o in every condition in this paper; the exact identity of the judge/PRM component and any fine-tuning procedure are not reported here, since Section 3.3 describes it as a judge model rather than a purely rule-based scorer. Table 3 uses $I = 5$, $D = 3$; the search-procedure ablation (Section 4.3) uses $I = 8$ to match its flat/sequential baselines. k (candidates per expansion) is inferred as 8 from the matched-budget baselines (Best-of-N with $N = 8$, Self-Refine with $k = 8$).

Second ablation: depth, retrieval, exploration weight. Table 5 reports an independent sweep over 150 patents that varies search depth, whether retrieval-augmented grounding is enabled, and β, holding the search procedure fixed at UCB1*-guided MCTS*—a different axis from Table 4, so the two are complementary rather than redundant.

Disabling retrieval hurts the PRM's own grounding signal (-10.7%) far more than the raw score (-0.4%), consistent with retrieval improving groundedness rather than surface fluency, a pattern also seen in patent-domain retrieval more broadly (Xiong et al., 2025; Qin et al., 2024). Setting $\beta = 0$ leaves both columns unchanged to three decimal places, reinforcing that the exploration term has little room to matter under single-child-per-expansion search (Section 3.1). A per-IPC-section breakdown of BERTScore F1 (novelty/inventive/validity references) falls within a band only 0.0042 wide across all eight sections, in contrast to SCQS's sub-scores, which vary meaningfully by IPC section, with linkage consistently weakest.