AWACopilot: A Secure On-Premise Large Language Model-Based Solution for Enhanced Patent Drafting

Mohamad Homam Mawaldi[1,2,*], Zenun Kastrati[1] and Alexander Gustafsson[1]

[1] *Linnaeus University, Universitetsplatsen 1, 352 52 Växjö*
[2] *AWA Sweden AB, Matrosgatan 1 211 18 Malmö, Sweden*

Abstract

Patent drafting is a complex and high-stakes process for securing intellectual property rights. During the patent prosecution phase, maintaining confidentiality is crucial, making cloud-based third-party services inadequate for patent drafting assistance due to data security concerns. This study proposes AWACopilot, a secure, on-premise solution comprising a web service that leverages open-source large language models (LLMs) to assist patent attorneys in the intricate patent application drafting process. AWACopilot generates key patent sections such as background, abstract, detailed description, etc., from human-crafted claims, addressing the data security risks posed by cloud-based AI services. Its modular architecture enables customization and adaptability to different patent tasks. Although challenges remain—including reliance on LLM capabilities and the need for rigorous content verification—this study demonstrates the potential for secure, AI-driven solutions to enhance patent drafting workflows.

Keywords

Intellectual Property, Patents Drafting, LLM, Prompt Engineering, Privacy

1. Introduction

A nation's investment in research and development (R&D) with its research workforce is a key driver of innovation, leading to a robust patent portfolio [1]. This portfolio can subsequently stimulate GDP growth and improve access to international markets. Thus, intellectual property law is not just a legal formality but a vital mechanism for converting R&D into tangible benefits. Notably, a study by the European Patent Office (EPO) [2] revealed that early-stage patent filings enhance startup's chances of obtaining venture capital funding by a factor of 6.4, while also significantly increasing the likelihood of a successful exit for its investors. This underscores the advantages of patents for enterprises of all sizes.

Patent prosecution, the legal and administrative process of obtaining a patent, begins with drafting an application that defines the invention's scope, as later additions of new subject matter are prohibited [3]. Typically handled by patent attorneys due to its complexity, successful prosecution requires technical and legal expertise. A patent application's key sections, including the background (contextualizing the invention), claims (defining legal boundaries), description (detailing the invention), drawings (visual support), and abstract (summary), must be consistent. Strategic claim drafting seeks broad protection while ensuring novelty and non-obviousness, and incorporates narrower claims for litigation defense. The remaining sections of the application are drafted with consistent language from the claims to fully enable the invention without limiting language. This interplay of introduced terms, sought protection in the description sections, with several stakeholders involved, makes patent drafting time-consuming and intellectually demanding. Recognizing these challenges, practitioners and researchers have increasingly turned to software tools and natural language processing (NLP) techniques to assist in the patent prosecution process.

6th Workshop on Patent Text Mining and Semantic Technologies (PatentSemTech) 2025

[*] Corresponding author.

✉ mohamadhomam.mawaldi@lnu.se (M. H. Mawaldi); zenun.kastrati@lnu.se (Z. Kastrati); alexander.gustafsson@lnu.se (A. Gustafsson)
🌐 https://lnu.se/en/staff/mohamadhomam.mawaldi/ (M. H. Mawaldi); https://lnu.se/en/staff/zenun.kastrati/ (Z. Kastrati); https://lnu.se/en/staff/alexander.gustafsson/ (A. Gustafsson)
iD 0009-0001-1862-671X (M. H. Mawaldi)

Recent advancements in deep learning (DL), particularly the development of the transformer architecture [4], have given rise to powerful large language models (LLMs) like ChatGPT [5] and led to a new era of technological innovation. Despite the challenges posed by LLMs like hallucinations [6], their impact on various industries is profound, such as healthcare [7], tourism [8], and the legal sector [9]. The legal field, particularly in intellectual property (IP), stands to gain substantially from the intrinsic NLP capabilities of LLMs. This is due to the sector's reliance on extensive and complex documents that require a deep comprehension of the specific meaning and implications of each term [10]. LLMs, equipped with word embeddings and attention mechanisms, could provide state-of-the-art performance compared to classical machine learning & DL approaches [11, 12].

Patent claims, the heart and defining element of patents, receive significant attention in the research that studies utilizing LLMs for patent generation [10, 13]. In contrast, products in the commercial sector aimed at end users adopt a more holistic approach, providing services that extend beyond mere LLM patent claim generation. Software companies like ClaimMaster Software LLC [14], PowerPatent Inc. [15], and Rowan TELS Corp. [16] provide services ranging from patent proofreading and drafting to the generation of office action responses enhanced by LLMs. Although some solutions may initially suggest local processing, the computationally intensive nature of LLMs inference typically necessitates routing all AI-driven tasks to cloud-based infrastructure. This reliance on cloud services raises serious concerns about data security and privacy. The possibility that major LLM providers like OpenAI, Google, and Microsoft might use interaction data to improve their models poses a threat to the confidentiality of patent applications during the crucial prosecution phase. Once the details of an invention become public knowledge, the novelty can be destroyed. To mitigate these risks, this study proposes deploying a secure, on-premise patent drafting solution backed by local LLMs, minimizing reliance on AI cloud services and ensuring greater control over sensitive data, supporting patent attorneys in patent drafting. This approach leverages state-of-the-art open-source LLMs and prompt engineering techniques to generate patent sections, such as the background, abstract, and detailed description, directly from human-crafted patent claims, without requiring prior fine-tuning of LLMs. This proposed solution will be implemented and deployed internally at AWA[1], an international intellectual property firm offering a comprehensive range of IP services.

2. Related Work

Early adoption of LLMs for patent-related tasks can be traced back to the release of GPT-2. In [17], researchers demonstrated GPT-2's ability to generate patent claims, observing its "unreasonably fast" adaptation, as the model learned to produce text resembling the structure and style of patent claims after fine-tuning on a dataset of half a million examples. The recommendations for future research from [17] were addressed in [18], where they fine-tuned GPT-2 on a considerably larger dataset of 11 million patents, focusing on generating patent titles, abstracts, and claims. However, both studies concluded that future research should concentrate on improving LLM generation, as it currently falls short of achieving human-level performance in these tasks.

Continuing to focus on patent claims, the study by [19] introduced a distinctive approach by employing a GPT-based model, PatentGPT-J, which was pretrained on patent text rather than fine-tuned, in contrast to the previously mentioned papers. The objective was to assess PatentGPT-J's effectiveness in aiding patent drafting, rather than generating complete texts. Specifically, the study aimed to evaluate the reduction in keystrokes for typists with the next-word suggestions provided by the model. Interestingly, larger parameter variants of the model did not outperform those with fewer parameters for this task. InstructPatentGPT [20], which was also based on PatentGPT-J, introduced a groundbreaking approach in the field that transcends traditional fine-tuning. It enhanced its model using Reinforcement Learning from Human Feedback (RLHF) to generate patent claims, expecting that this training method would elevate the quality of the produced claims. The model learns to refine the language of the claims it generates, adjusting aspects such as length and terminology by incorporating feedback related to the

[1] https://www.awa.com/

"granted" and "pre-grant" statuses of patent applications during the prosecution process. Consistent with previous studies, [20] noted that the claims produced still require significant improvements to align with the standards required by patent offices. A more recent study, [21], pre-trained and fine-tuned four models—LLaMA-2-7B and 13B, as well as Mistral-7B and 8x7B—specifically for the generation of biomedical patent claims. The research concludes that although the claims generated by the LLMs exhibited potential, human oversight is essential to ensure that these claims satisfy the critical patentability requirements of quality, novelty, and non-obviousness.

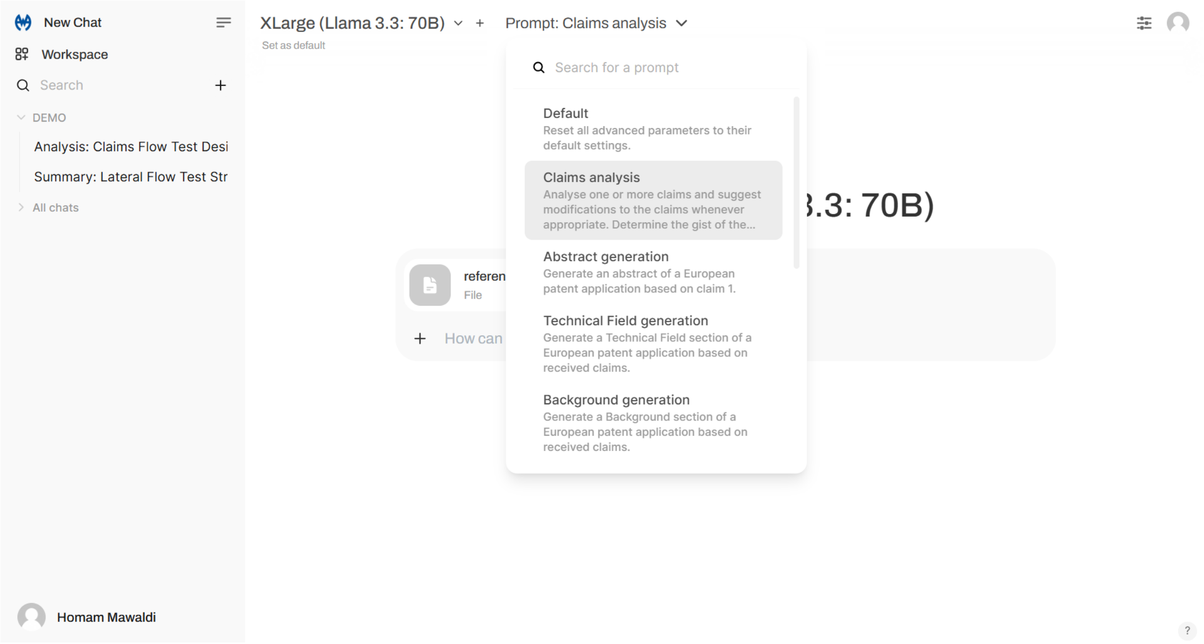
In addition to patent claims, patent specifications have received considerable attention, as demonstrated by the study conducted by [22]. In their research, the authors utilized claims and drawings as the foundation for generating the remainder of a patent application. They employed two pre-trained models: a decoder-only model, GPT-J, which contains 6 billion parameters, and an encoder-decoder model, T5, with 11 billion parameters. Both models were fine-tuned on a dataset of patents specifically related to computing arrangements based on biological models. The evaluation of these models revealed improvements in the specifications of the generated patents. However, the authors stressed the importance of including patent attorneys in carefully reviewing the generated specifications to ensure both quality and accuracy, as models hallucinate descriptions of claim features and drawing descriptions. The necessity for including a patent attorney in the process is emphasized in the work of [23], which raises concerns about privacy and the interpretability of LLMs' outputs. The conclusion drawn is that LLMs should serve to augment patent attorneys' expertise rather than replace them. The authors suggest a human-in-the-loop workflow as a method to mitigate hallucinations and maintain high-quality output for the designated tasks.

Fine-tuning LLMs is a resource-intensive process, requiring substantial GPU resources, a suitable fine-tuning dataset, and considerable time investment. This can hinder the rapid adaptation of LLMs to emerging tasks and newly released foundation models within organizations. To mitigate these challenges, techniques like Low-Rank Adaptation (LoRA) [24] have emerged as effective solutions. LoRA optimizes model fine-tuning by adjusting only a small subset of parameters, known as low-rank adapters. This method involves freezing the pre-trained weights and incorporating trainable rank decomposition matrices, significantly reducing the number of trainable parameters while maintaining or even improving performance on downstream tasks. Another line of research focused on prompt engineering, where techniques like chain-of-thought prompting improve complex reasoning without fine-tuning [25]. Prompting has also been shown to outperform domain-specific fine-tuned models. For example, GPT-4, guided by carefully designed prompts, surpassed Flan-PaLM 540B and Med-PaLM 2 on the MultiMedQA benchmark [26]. These results suggest that prompt engineering could serve as a powerful and efficient alternative to fine-tuning.

The quality of a patent application, rooted in the strength of its claims, leads to faster approval timelines [27]. The language in patent applications, particularly in claims, differs from everyday English and does not match the training data used for LLMs. This complexity positions patent attorneys as the most qualified for the task, consistent with prior research. This study utilizes prompt engineering to enhance LLM outputs in generating patent sections, such as summaries, abstracts, and detailed descriptions, directly from human-crafted claims while incorporating a human-in-the-loop method, as suggested by earlier studies. This approach avoids the lengthy process of fine-tuning. It combines these elements into a user-friendly, on-premise deployed solution, aiming to improve the workflow of patent attorneys at AWA during the drafting process, all while upholding the highest security standards.

3. Proposed Solution

Security and privacy were key priorities during the implementation of AWACopilot. To that end, LLM inference is performed on a dedicated server within the AWA internal network, isolated from external access and accessible only to authorized AWA devices. The solution was deployed using Docker [28] on the server, with all outbound internet connections restricted for added security. Furthermore, the solution frontend was implemented as a web application instead of a native application, which eliminates



New Chat

Workspace

Search

DEMO

Analysis: Claims Flow Test Desi

Summary: Lateral Flow Test Str

All chats

Homam Mawaldi

XLarge (Llama 3.3: 70B) +

Set as default

Prompt: Claims analysis

Search for a prompt

Default
Reset all advanced parameters to their default settings.

Claims analysis
Analyse one or more claims and suggest modifications to the claims whenever appropriate. Determine the gist of the...

Abstract generation
Generate an abstract of a European patent application based on claim 1.

Technical Field generation
Generate a Technical Field section of a European patent application based on received claims.

Background generation
Generate a Background section of a European patent application based on received claims.

.3: 70B)

referen
File

+ How can

?

Figure 1: AWACopilot website showcasing visible options for patent tasks

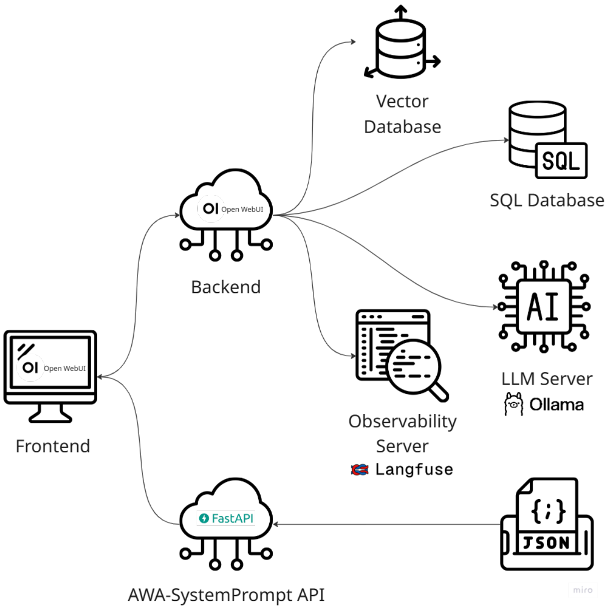


Figure 2: Communication between components in the proposed solution.

the need for user-side installation. This design choice facilitated rapid prototyping and iterative development, streamlining modification, observability, and updates. Access to the web application was restricted to authorized users, requiring individual credentials for interface access. Figure 1 presents a screenshot of the frontend of the proposed solution.

AWACopilot, as illustrated in Figure 2, processes user requests via a multi-stage architecture. User instructions, combined with the task obtained from the AWA-SystemPrompt API, are passed from the frontend to the backend. Crucially, attached documents submitted by the user are handled based on the user's selection: either they are used to retrieve relevant information through a Retrieval-Augmented Generation (RAG) process, through a vector database, or the entire document is directly passed to the LLM. The backend then forwards the assembled instructions and context to both the LLM Server and the observability server, concurrently notifying the frontend about the ongoing processing. As the LLM Server generates tokens, the frontend receives and displays them in real-time. Upon completion, the observability server records the full interaction in the database.

The solution is built upon the open-source platform Open-WebUI [29]. Open-WebUI, developed using Svelte[2] with JavaScript for the frontend and FastAPI [30] with Python for the backend, provides a foundation with an active community and a rich feature set. The backend provides support for user management, file handling, and RAG functionality [31]. The capabilities of Open-WebUI can be easily expanded through the use of functions on the frontend and pipelines on the backend. The following paragraphs will detail how these two features were utilized. Such functionalities made this platform ideal for agile development and rapid prototyping compared to building a platform from scratch.

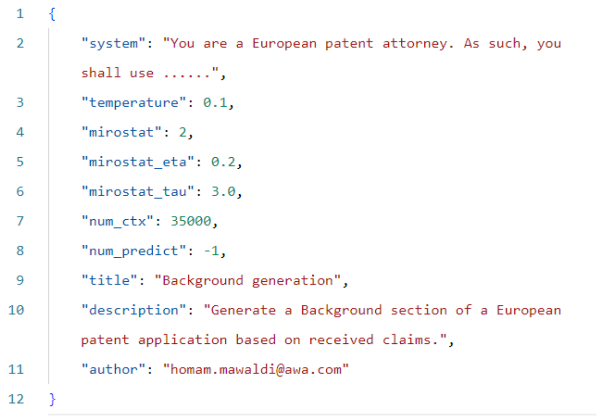
To allow users to assign tasks to LLMs using specific system prompts, a crucial requirement for AWACopilot was to decouple these prompts from the underlying LLMs. In contrast, Open-WebUI tightly integrates system prompts and inference parameters to a predefined LLM, leading to an inflexible system that fails to meet the needs of this study. This limitation prompted the development of AWA-SystemPrompt, which addresses the missing functionality in Open-WebUI. AWA-SystemPrompt is a REST API created with FastAPI and supported by a straightforward document database utilizing JSON storage. It provides system prompts specifically designed for patent-related tasks, like claim analysis, abstract generation, technical field generation, background generation, and detailed description generation. Since the Open-WebUI could not be extended to accommodate such features, a branch was created from the main repository. Subsequently, the frontend was updated to utilize this API to pass the instruction to the LLMs through the backend, offering users a dropdown menu to select from the available system prompts. Furthermore, users can create, edit, delete, and share prompts, extending beyond the default system prompts. Given their specialized knowledge and expertise in various scientific and technical fields, patent attorneys are ideally positioned to create effective prompts for patent-related tasks. Their deep understanding of both patent law and complex technical language enables them to craft precise instructions that yield the desired results for each specific case. These custom prompts can be made accessible to other AWACopilot users or kept private. This modular approach, implementing system prompt management as a separate API, minimized modifications to the core Open-WebUI codebase, thereby simplifying future updates and integration with new Open-WebUI releases. As illustrated in Figure 3, a system prompt is a JSON object that contains instructions for the LLM and parameters for sampling. These parameters govern various aspects of the LLM's behavior, including the context window (number of tokens the model considers) and the generation length (number of tokens the model produces). Moreover, parameters like temperature and Mirostat affect the randomness and coherence of the output, influencing the model's "creativity" and adherence to the specified task.

For the LLM server, Ollama [32] was chosen due to its ease of deployment and management. Ollama supports the serving of LLMs through a REST API, whether the models are sourced from Ollama.com or other repositories like Hugging Face[3]. In both scenarios, the selected model is downloaded, stored on the server, and served through the Ollama API, which can then relay LLM requests. Upon receiving a request, Ollama loads the model into memory, a process whose duration depends on the number of the model's weights. Once the model is loaded, Ollama performs inference and subsequently offloads the model either upon receiving a new request for a different model or after five minutes of inactivity. Alternative frameworks, including vLLM [33] and Max from Modular [4], were also assessed, especially for their superior inference performance compared to Ollama. However, their less efficient dynamic model loading and offloading capabilities based on API requests rendered them suboptimal for AWACopilot's needs. The strategy of local deployment influenced the selection of LLMs, emphasizing models suitable for on-premise execution. As a result, the focus was placed on open-source models with permissive licenses, including the Llama3 [34], Gemma [35], and Deepseek [36] families.

[2] https://svelte.dev/

[3] https://huggingface.co/

[4] https://www.modular.com/max



```json
{
    "system": "You are a European patent attorney. As such, you shall use ......",
    "temperature": 0.1,
    "mirostat": 2,
    "mirostat_eta": 0.2,
    "mirostat_tau": 3.0,
    "num_ctx": 35000,
    "num_predict": -1,
    "title": "Background generation",
    "description": "Generate a Background section of a European patent application based on received claims.",
    "author": "homam.mawaldi@awa.com"
}
```

Figure 3: Example of a system prompt containing LLM inference instructions and parameters served by AWA-SystemPrompt API

To ensure comprehensive monitoring, debugging, and user support, the open-source observability framework Langfuse [37] was deployed in a dedicated Docker container. Langfuse, which implements OpenTelemetry[5] and integrates with Open-Web UI via its backend pipeline feature through a REST API, captures detailed information about each LLM interaction. Serving as a model- and framework-agnostic LLM engineering platform, Langfuse enables debugging, analysis, and iteration on LLM applications, allowing it to capture detailed information about each LLM interaction. This data includes input tokens, output tokens, and the generated output itself, which facilitates the creation of statistics on LLM performance and offers valuable insights for debugging and improving system prompts. Additionally, through the functions feature of Open-WebUI, users could rate LLM responses on a scale from 0 to 10 and provide textual feedback. This feedback gets collected and linked to the recorded trace in Langfuse, enabling continuous improvement and iterative refinement of both LLMs and system prompts.

4. Discussion

Although this study has an exploratory nature and does not present results, several preliminary observations emerged from the exploratory process. One key insight is the crucial role of LLMs' context window size: patent attorneys often work with lengthy materials such as invention disclosures, patent claims, and prior art documents, making it essential for LLMs to process large inputs effectively. While a RAG approach can be effective, in certain situations, providing the LLM with the entire document within its context window enables a more thorough analysis, allowing the model to capture the intricate details and nuances necessary for complex tasks that demand a holistic understanding. Another observation is that even a well-performing AI tool might not achieve adoption if it does not integrate naturally into the workflows already established in patent drafting; simply offering a tool is not enough without facilitating habitual use. Additionally, LLMs sometimes fail to generate sufficiently detailed or complete outputs for long and complex tasks, suggesting that agentic or multi-step workflows may be necessary. Overall, these exploratory insights highlight that launching a product involves much more than technical performance — it requires a deep focus on usability, integration, and real-world practicality to truly serve end users.

5. Limitations

While this study demonstrates the deployment of a secure, on-premise solution for patent drafting assistance, several limitations should be taken into consideration. The infrastructure requirements for local deployment present a scalability challenge. Operating LLMs on-premise necessitates substantial

[5] https://opentelemetry.io/

and costly GPU resources, not only for initial model loading but also to reserve sufficient memory for the model's context window. This constraint may limit the number of model parameters that can be loaded, which can potentially lead to performance degradation. It may also reduce throughput when serving multiple concurrent users. As a result, this could create a perception of unreliability for the end user. Consequently, broader adoption and responsiveness may be hindered compared to cloud-based alternatives.

Furthermore, workflow integration poses a challenge. AWACopilot is accessible via a web application, which may introduce friction for attorneys accustomed to established drafting practices. Additionally, recent advancements in agentic workflows, leveraging reasoning and the collaborative capabilities of diverse LLMs, offer promising potential for enhanced patent task performance; however, their integration was not feasible within the scope and timeframe of this study. Future research could investigate the incorporation of such complex AI workflows. Finally, the evaluation of AWACopilot's effectiveness is currently based on early-stage feedback and limited-scale testing. A comprehensive assessment of its impact on attorney productivity and drafting quality in a real-world production environment remains an essential area for future investigation.

6. Conclusion and Future Work

This study explored deploying AWACopilot, a secure on-premise solution utilizing open-source LLMs to assist AWA patent attorneys in drafting patent applications sections from human-crafted claims. It addressed critical data security concerns during sensitive prosecution phases while demonstrating a modular and adaptable system design. Future work could focus on developing a quantifiable methodology to evaluate the effectiveness of different LLM-system prompt pairs in patent generation tasks. This includes investigating whether larger parameter models consistently yield better results and assessing the tool's impact on attorney workflows through expanded user studies, surveys, and feedback sessions.

Declaration on Generative AI

While preparing this study, the authors utilized API calls to gpt-4o-mini, gemini-2.0-flash, and o3-mini in order to: Paraphrase and reword, Improve writing style, and Grammar and spelling check. Following the use of these services, the authors reviewed and edited the content as necessary and take full responsibility for the publication's content.

References

[1] R. Rubilar-Torrealba, K. Chahuán-Jiménez, H. De La Fuente-Mella, Analysis of the Growth in the Number of Patents Granted and Its Effect over the Level of Growth of the Countries: An Econometric Estimation of the Mixed Model Approach, Sustainability 14 (2022) 2384. doi:10.3390/su14042384.

[2] EUIPO, EPO, Patents, Trade Marks and Startup Finance, Study, EUIPO, EPO, 2023.

[3] World Intellectual Property Organization., WIPO Patent Drafting Manual., second edition ed., World Intellectual Property Organization, Geneva, Switzerland, 2023. doi:10.34667/TIND.44657.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention Is All You Need, 2017. doi:10.48550/ARXIV.1706.03762.

[5] OpenAI, Introducing ChatGPT, https://openai.com/index/chatgpt/, 2022.

[6] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, T. Liu, A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions, ACM Trans. Inf. Syst. 43 (2025) 42:1–42:55. doi:10.1145/3703155.

[7] A. Fatima, M. A. Shafique, K. Alam, T. K. Fadlalla Ahmed, M. S. Mustafa, ChatGPT in medicine: A cross-disciplinary systematic review of ChatGPT's (artificial intelligence) role in research,

clinical practice, education, and patient interaction, Medicine 103 (2024) e39250. doi:10.1097/MD.0000000000039250.

[8] D. Gursoy, L. , Yu, H. and Song, ChatGPT and the hospitality and tourism industry: An overview of current trends and future research directions, Journal of Hospitality Marketing & Management 32 (2023) 579–592. doi:10.1080/19368623.2023.2211993.

[9] J. Lai, W. Gan, J. Wu, Z. Qi, P. S. Yu, Large language models in law: A survey, AI Open 5 (2024) 181–196. doi:10.1016/j.aiopen.2024.09.002.

[10] L. Jiang, S. Goetz, Natural Language Processing in Patents: A Survey, 2024. doi:10.48550/arXiv.2403.04105. arXiv:2403.04105.

[11] F. Ariai, G. Demartini, Natural Language Processing for the Legal Domain: A Survey of Tasks, Datasets, Models, and Challenges, 2024. doi:10.48550/ARXIV.2410.21306.

[12] C. M. Greco, A. Tagarelli, Bringing order into the realm of Transformer-based language models for artificial intelligence and law, Artificial Intelligence and Law 32 (2024) 863–1010. doi:10.1007/s10506-023-09374-7.

[13] S. Casola, A. Lavelli, Summarization, simplification, and generation: The case of patents, Expert Systems with Applications 205 (2022) 117627. doi:10.1016/j.eswa.2022.117627.

[14] ClaimMaster Software LLC, Patent claim master, https://www.patentclaimmaster.com, 2025.

[15] PowerPatent Inc., PowerPatent: Patent prosecution software, https://powerpatent.com, 2025.

[16] Rowan TELS Corp, Effective patent drafting with rowan patents, https://rowanpatents.com, 2025.

[17] J.-S. Lee, J. Hsiang, Patent claim generation by fine-tuning OpenAI GPT-2, World Patent Information 62 (2020) 101983. doi:10.1016/j.wpi.2020.101983.

[18] D. Christofidellis, A. B. Torres, A. Dave, M. Roveri, K. Schmidt, S. Swaminathan, H. Vandierendonck, D. Zubarev, M. Manica, PGT: A prompt based generative transformer for the patent domain, in: ICML 2022 Workshop on Knowledge Retrieval and Language Models, 2022, pp. 1–7.

[19] J.-S. Lee, Evaluating generative patent language models, World Patent Information 72 (2023) 102173. doi:10.1016/j.wpi.2023.102173.

[20] J.-S. Lee, InstructPatentGPT: Training patent language models to follow instructions with human feedback, Artificial Intelligence and Law (2024). doi:10.1007/s10506-024-09401-1.

[21] F.-C. Chen, C.-L. Pan, Evaluating application of large language models to biomedical patent claim generation, World Patent Information 80 (2025) 102339. doi:10.1016/j.wpi.2025.102339.

[22] J. Wang, S. K. R. Mudhiganti, M. Sharma, Patentformer: A Novel Method to Automate the Generation of Patent Applications, in: F. Dernoncourt, D. Preoţiuc-Pietro, A. Shimorina (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, Association for Computational Linguistics, Miami, Florida, US, 2024, pp. 1361–1380. doi:10.18653/v1/2024.emnlp-industry.101.

[23] L. V. Bui, Advancing patent law with generative AI: Human-in-the-loop systems for AI-assisted drafting, prior art search, and multimodal IP protection, World Patent Information 80 (2025) 102341. doi:10.1016/j.wpi.2025.102341.

[24] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-Rank Adaptation of Large Language Models, in: International Conference on Learning Representations, 2021, pp. 1–13.

[25] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, in: Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22, Curran Associates Inc., Red Hook, NY, USA, 2022, pp. 24824–24837.

[26] H. Nori, Y. T. Lee, S. Zhang, D. Carignan, R. Edgar, N. Fusi, N. King, J. Larson, Y. Li, W. Liu, R. Luo, S. M. McKinney, R. O. Ness, H. Poon, T. Qin, N. Usuyama, C. White, E. Horvitz, Can Generalist Foundation Models Outcompete Special-Purpose Tuning? Case Study in Medicine, 2023. doi:10.48550/arXiv.2311.16452. arXiv:2311.16452.

[27] D. Harhoff, S. Wagner, The Duration of Patent Examination at the European Patent Office, Management Science 55 (2009) 1969–1984. doi:10.1287/mnsc.1090.1069.

[28] D. Merkel, Docker: Lightweight Linux containers for consistent development and deployment,

Linux J. 2014 (2014) 2:2.

[29] T. J. Baek, Open-webui: User-friendly ai interface (supports ollama, openai api, and more), https://github.com/open-webui/open-webui, 2023. BSD-3-Clause License.

[30] S. Ramírez, Fastapi, 2018. URL: https://github.com/fastapi/fastapi, fastAPI framework, high performance, easy to learn, fast to code, ready for production.

[31] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive NLP tasks, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Curran Associates Inc., Red Hook, NY, USA, 2020, pp. 9459–9474.

[32] Ollama contributors, Ollama, https://github.com/ollama/ollama, 2025.

[33] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. Gonzalez, H. Zhang, I. Stoica, Efficient Memory Management for Large Language Model Serving with PagedAttention, in: Proceedings of the 29th Symposium on Operating Systems Principles, SOSP '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 611–626. doi:10.1145/3600006.3613165.

[34] A. Grattafiori, et al., The Llama 3 Herd of Models, 2024. doi:10.48550/ARXIV.2407.21783.

[35] A. Kamath, et al., Gemma 3 Technical Report, 2025. doi:10.48550/ARXIV.2503.19786.

[36] D. Guo, et al., DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning, 2025. doi:10.48550/ARXIV.2501.12948.

[37] Langfuse contributors, Langfuse: Open source llm engineering platform, https://github.com/langfuse/langfuse, 2025. MIT License.