Patent Response System Optimised for Faithfulness: Procedural Knowledge Embodiment with Knowledge Graph and Retrieval Augmented Generation

Jung-Mei Chu[1,2]**, Hao-Cheng Lo**[1,2]**, Jieh Hsiang**[1]**, Chun-Chieh Cho**[2]**,**
[1]National Taiwan University, [2]JCIPRNET
Correspondence: austenpsy@gmail.com

Abstract

A successful response to Office Action is crucial for an invention to obtain a patent. While previous attempts have applied generalised LLMs, such as GPT-4, in the response process, there remains significant room for improvement in generating faithful, unbiased, and practically valuable responses. To address this issue, we propose the Patent Response System Optimised for Faithfulness (PRO). PRO explicitly incorporates procedural knowledge used by patent agents during drafting arguments in response. This framework comprises several key components: (1) Our proposed PRLLM is a LLM tailored for patent responses, designed to have comprehensive patent domain-specific knowledge. (2) Our proposed PPNet encodes legal interpretations and relationships between technical components from judicial sources through a knowledge graph. (3) The augmented generation processes retrieve relevant information from both the patent text and PPNet to augment the PRLLM's input and generate faithful responses. Results show that PRO significantly reduces unfaithfulness across six error types compared to several settings. For instance, PRO outperforms GPT-4 by an average of 39% in terms of faithfulness. This demonstrates the effectiveness of our domain-specific approach in improving the quality of automated patent responses.

1 Introduction

Large Language Models (LLMs), such as GPT-4 (OpenAI, 2023) and LLaMa2 (Touvron et al., 2023), are deemed generalised and not domain-specific, posing challenges in the patent field. In the intellectual property field, patents filed with the United States Patent and Trademark Office (USPTO) are continuously evolving and growing, with new technologies and legal terms requiring complex analysis (USPTO, 2023). Recently, research has focused on developing or applying language models (LMs) and LLMs tailored for patent language to address tasks such as patent drafting (Lee and Hsiang, 2020), prior art search (Lo et al., 2024), and semantic analysis (Chu et al., 2024).

Although these efforts have been made, LLMs have not significantly improved the Office Action (OA; e.g., rejection) and response (e.g., argument or amendment) process. This process involves detailed communication and extensive exchanges of technical and legal knowledge between examiners and patent agents to ensure the inventions' novelty and non-obviousness. Chu et al. (2024) have started investigating the use of LMs/LLMs and recommender systems to automate patent responses. However, due to the concern of privacy, the distinctive nature of patent language, the uniqueness of each invention, and the intricacy of formulating responses, considerable improvements are still needed in patent response systems.

This leads us to our first research question: **can we develop a domain-specific patent response LLM (PRLLM)?** To investigate this, we constructed a dataset comprising patents and their corresponding OA-response histories over 10 years. This dataset also includes a wide range of types, domains, and tasks, ensuring comprehensive coverage. Incorporating previous data during training helps retain knowledge from earlier training phases, thus preventing the forgetting issue (Ibrahim et al., 2024). For the model, we selected LLaMa2 as the base model for continual pretraining among open-source LLMs. For supervised fine-tuning (SFT), we used paired OA-responses. The zero-shot results showed that while the model performs well in terms of formatting responses, identifying key legal and technical terms, it struggles with analysing examiners' rejections (e.g., novelty or non-obviousness analysis), even when additional information is provided (see section 2.1 and section 5).

This raises another question: **how can we enhance the faithfulness of PRLLM in developing**

Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024), pages 146–155
August 16, 2024 ©2024 Association for Computational Linguistics

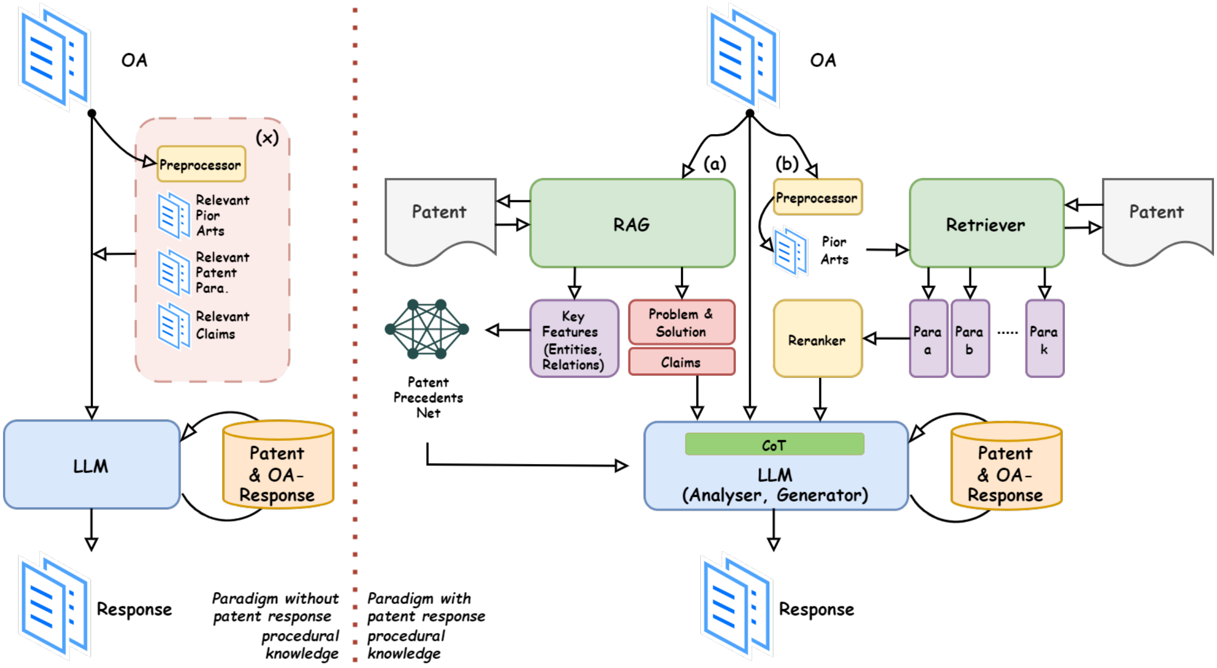

Figure 1: **Architecture Overview**. Left: Paradigm without patent response procedural knowledge, using only our PRLLM. Right: PRO framework.

arguments? Empirically, patent agents utilise a series of *procedural knowledge* during response analysis. Upon receiving an OA, they first identify the points of contention (e.g., rejections). From there, they follow *dual paths*. The first path involves finding the core inventive concept related to the point of contention, which could be reflected in the patent and/or independent claims. This includes identifying the patent's key features and its problem-solution. Agents then search for relevant past precedents to support their arguments (Garrod, 2010). The second path addresses rejections that the invention is similar to prior art. Agents analyse the relevant paragraphs in the patent that relate to the prior art, using them as the basis for their arguments. Combining these foundations, agents develop arguments and/or amendments to address the deficiency in the examiner's Broadest Reasonable Interpretation (BRI) and/or in the patent's claim. This type of procedural knowledge is not present in the previous LLMs, as it is implicit knowledge that agents use during the response process.

Hence, we propose a novel framework: **Patent Response System Optimized for Faithfulness (PRO)**. This framework aims to explicitly incorporate *procedural knowledge* into the model. Specifically, the framework includes a patent precedents KG (PPNet) that represents the external precedents patent agents might refer to during developing arguments. This KG characterises the relationships between invention technologies, not only in common or dictionary definitions but with legal interpretations that include judicial logic and specificity.

Additionally, the framework involves multiple Retrieval-Augmented Generation (RAG) processes, where retrievers use points of contention or prior art to retrieve relevant information in the patent, and the generator uses PPNet/PRLLM-retrieved results to produce key features, problem-solution statements, and the resulting response. Experimental results demonstrate that this framework significantly reduces unfaithfulness compared to baselines.

We make several key contributions:

- We pioneered the development of a domain-specific patent response LLM (PRLLM).
- We are the first to introduce PPNet, a KG of patent precedents. The KG serves as the foundation for retrieving relations between entities and is used in subsequent reasoning processes.
- We propose the framework PRO, which embodies the procedural knowledge used by patent agents in the response process. This framework combines PPNet and RAG with proposed PRLLM. This integration effectively enhances the faithfulness in PRLLM results.

2 Architecture Overview

Considering the domain-specific nature of patent responses, we first developed the patent response LLM. This model is designed to run locally for security reasons and is well-versed in patent language, various technical terms, relevant legal terminology, and the structure, format, and analysis required for responses. This LLM forms the core foundation of our entire technical architecture and can function as both a generator and a retriever within our framework (see section 3 for its training details).

2.1 Paradigm without Procedural Knowledge

As shown on the left side of fig. 1, the most intuitive way to use PRLLM is through zero-shot application. When a patent agent encounters an OA, they can directly use PRLLM to generate the response content. This represents the simplest form of application.

A slightly more complex approach (see fig. 1 (x)) involves breaking down the information in the OA and identifying relevant details to add to the model input. Specifically, this involves extracting the examiner's rejections, relevant prior art and patent paragraphs, and the key claims under dispute. Given the model's window size limitations, these extracted details are token-optimised before being provided as input to PRLLM, resulting in a more precise response compared to the zero-shot method.

Both of these methods are direct applications, which we refer to as the paradigm without procedural knowledge. While this paradigm is simple, it lacks the integration of procedural knowledge crucial to the patent response process, potentially limiting its effectiveness.

2.2 Paradigm with Procedural Knowledge (PRO)

As shown on the right side of fig. 1, our PRO framework explicitly incorporates the *procedural knowledge* used by patent agents into the system. It consists of *dual paths*: *PPNet path* and *prior art retrieval path*.

For *PPNet path* (see fig. 1 (a)), we first use regular expressions to extract points of contention and the corresponding independent claims from the OA. Using this information, we perform RAG to identify key features and problem solutions. Specifically, we retrieve relevant texts in the patent using cosine similarities of dense vector representations derived from the PRLLM. During the generation phase, our generator takes this textual information to output key features, including the relevant components (entities) and their relationships (relations), as well as the problem-solution of the patent.

We then use these extracted components and relationships to query our constructed PPNet. This KG helps to retrieve the legal implications of technical details within the patent. For example, if one queries *"what is a gate above?"* it might answer *"a gate is above a layer"* and *"above means neither 'directly above' nor simply 'at a higher place than'"*, providing precise legal interpretations.

For *prior art retrieval path* (see fig. 1 (b)), the objective is to utilise the examiner's cited prior art paragraphs (which challenge the novelty and non-obviousness of the invention) to retrieve relevant paragraphs of the current patent application. Since examiners typically specify the locations and content of these prior art paragraphs, we can extract this information using regular expressions. After extracting the relevant prior art content, we apply the retrieval method identical to the first path, using PRLLM to identify similar paragraphs in the current patent application. These retrieved paragraphs are then re-ranked based on their importance, with the examiner's most critical paragraph prioritised, followed by other similar passages.

This approach reflects one fact: While examiners often indicate the specific locations of contentious parts in the patent, our method not only relies on these key passages for argumentation but also uncovers additional details in the invention that the examiner may have overlooked. These overlooked details can be used to supplement and strengthen our response analysis.

Finally, the results from the two paths—the components and judicial rationales retrieved from PPNet, the problem-solution of the patent, and the key independent claims, along with the relevant passages from the invention—are combined with the relevant content in the OA to form the input for PRLLM.

Before this input is fed into the LLM, we perform a CoT process. This process is designed to determine the priority and functionality of each input and use reasoning prompting. Different inputs hold different levels of importance in constructing arguments and analyses. It is crucial for the LLM to understand the functionality and priority of these inputs to create a coherent and logical response. By structuring the inputs in this way and using reasoning prompts, PRLLM can generate responses with

Model	Params	Vocabs	LR	Context Length
PRLLM-13B	13B	32K	3.0×10^{-4}	16K
PRLLM-70B	70B	32K	2.0×10^{-5}	16K

Table 1: The information and attributes of PRLLM models.

higher faithfulness and accuracy.

3 PRLLM Training Details

We followed the approach outlined by Touvron et al. (2023) in training our PRLLM models. Using LLaMA2 as the base model, we trained models with parameters of 13 billion (13B) and 70 billion (70B), naming the series PRLLM-13B and PRLLM-70B respectively. The training process was divided into two main stages: continual pretraining and supervised fine-tuning (SFT).

3.1 Continual Pretraining

Data. To create an effective pretraining dataset, we ensured diversity and comprehensive coverage in our data. The patent domain encompasses extensive legal and engineering knowledge from various fields, necessitating a dataset that reflects this diversity.

First, our dataset includes patent documents and OA records, spanning from 2003 to 2022. This dataset comprises a total of 956,779 patents and 1,269,271 OA records from USPTO, accounting for 55.08% of the entire dataset. Second, we incorporated publicly available online resources, such as academic papers (12.64%), websites (11.24%), Wikis (9.28%), books (2.19%), exam databases and code repositories (2.07%), and news articles (2.02%). Lastly, the dataset includes some internal resources, such as judicial rulings (5.41%).

This comprehensive dataset design ensures that our Patent Response LLM has access to rich and diverse data during the pretraining phase. Leveraging data from various fields helps reduce potential biases in the model's patent response process. Ibrahim et al. (2024) have shown that incorporating data from different domains in the pretraining phase can maintain the generalization capabilities of LLM models.

Training. We initiated pretraining using an optimized autoregressive transformer. We employed the LLaMA2 13B and 70B versions. The training was conducted on an A100 GPU cluster, utilizing the AdamW optimizer combined with BFloat16 mixed precision to ensure training stability. Additionally, we implemented Cosine Learning Rate Scheduling for learning rate adjustments. Each training batch consisted of 4M tokens. To mitigate model performance regression, we extended the training context length from the original 4K to 16K (Xiong et al., 2023). Table 1 outlines the attributes and pretraining hyperparameters of the PRLLM models.

3.2 SFT

Data. During SFT, our data is divided into two parts. The first part, directly related to PRLLM, consists of paired OA-response datasets from 2023, totaling 10,000 instances. We denote this dataset as $\mathcal{D}_{\text{pr}}$. The second dataset is a general dataset ($\mathcal{D}_{\text{g}}$) comprising 20,000 instances, which were sampled from a variety of sources such as UltraChat (Ding et al., 2023), Databricks-dolly-15k (Conover et al., 2023), and the Guanaco Dataset (Dettmers et al., 2024). The final dataset ($\mathcal{D}$) used for fine-tuning is the union of these two datasets, $\mathcal{D} = \mathcal{D}_{\text{pr}} \cup \mathcal{D}_{\text{g}}$.

Training. We merged all instances and outputs from dataset $\mathcal{D}$. Each instance and its corresponding output were separated by a special token. This unified dataset was used to perform SFT on the two PRLLM models. Next, we omitted the loss calculation on tokens from user instructions and applied a weighted autoregressive objective (Wang et al., 2023). The loss function used in this training process is:

$$\mathcal{L}(\Theta) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[-\alpha \sum_{i \in O} \log p(x_i \mid \tilde{\mathbf{x}}; \Theta)] \quad (1)$$

where α is 1 if $\mathbf{x}$ is from D_{pr} and 0.15 if $\mathbf{x}$ is from D_{g}, O means output, Θ represents the model's parameters, and $\tilde{\mathbf{x}} = (x_0, x_1, \ldots, x_{i-1})$ represents the tokenized input sequence. In a similar vein, we utilised a cosine learning rate scheduler with learning rate of 2×10^{-5} and a batch size of 128. The models were fine-tuned over a total of 2 epochs.

4 PPNet: Construction & Evaluation

4.1 Building PPNet

Similar to constructing the Wikidata KG (Vrandečić and Krötzsch, 2014), we built PPNet for patent responses argument foundation. PPNet sources include judicial relationships of components and relevant judgment contents such as Markman Hearings (Creel, 2013; Garrod, 2010). The construction process involves several steps: First, we performed data cleaning and annotation on the collected materials. Next, we carried out knowledge extraction, which includes Named Entity Recognition (NER), attribute extraction, and relation extraction. These steps rely not only on existing NLP techniques but also on manual annotation or verification by patent agents, attorneys, and engineers. Through these procedures, we extracted key information from the judgments and stored it in the knowledge graph.

As a result, PPNet can be represented as a heterogeneous KG consisting of triplets *(head, relation, tail)*, denoted as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ is the set of entities (e.g., components), $\mathcal{R}$ is the set of relations (e.g., verbs), and $\mathcal{T}$ is the set of triplets. In total, PPNet comprises 4 million entities, 403 types of relations, and over 7 billion triplets.

4.2 PPNet QA Pipeline

To handle complex question-answering tasks in Knowledge Graph Question Answering (KGQA), we adopted a method inspired by Sen et al. (2023). Our implementation for KGQA on PPNet is as follows:

We use a sequence-to-sequence model to predict the distribution of relations that need to be traced in PPNet. That is, the decoder predicts a relation distribution in PPNet, performing this process for up to m-hops. Each hop generates a relationship distribution, indicating which relation might be traced in that step.

Specifically, before the QA process, let three sparse linear matrices be *head-to-triplet* $\mathbf{M}_h$, *relation-to-triplet* $\mathbf{M}_r$, and *tail-to-triplet* $\mathbf{M}_t$. We start with an initial query entity vector $e \in \mathbb{R}^{N_\mathcal{E}}$ and a relation vector $r \in \mathbb{R}^{N_\mathcal{R}}$. The entity from the query is represented as a one-hot vector in the entity space, which is mapped to a triplet vector using $\mathbf{M}_h$. For the relation, we use the relation vector predicted by the model and map it to a triplet vector through $\mathbf{M}_r$. Hence, the first hop can be expressed as:

$$\tau = \mathbf{M}_h e \odot \mathbf{M}_r r \tag{2}$$

where $\odot$ denotes element-wise multiplication. Then, using the tail-to-triplet matrix $\mathbf{M}_t$, the weighted triplet vector τ is mapped back to an entity vector:

$$e' = \mathbf{M}_t^T(\tau) \tag{3}$$

where e' represents the begin of the second hop. At each hop, only the top k weighted triplets are retained, and these triplets are converted into natural language representations.

4.3 Experiments and Results

Dataset. To test the PPNet QA pipeline in answering patent judgment-related questions, we constructed a dataset for the following experiments. This dataset was collaboratively built by patent agents, attorneys, and engineers. The questions involve previous precedents, focusing particularly on technical components and their associations with others. For instance, a question might be, *"What does a metal apparatus comprise?"* with a possible answer being *"copper"*.

The entire dataset consists of 4,730 questions (3,000 for the training set, 300 for the validation set, and 1,430 for the testing set). These questions are well-defined, with some involving multiple hops of reasoning to thoroughly test the capabilities and accuracy of the QA pipeline.

Experimental Setup. For the model, we selected several sequence-to-sequence models that have performed well in previous seminal work (Sen et al., 2023; Wu et al., 2023; Baek et al., 2023), including T0 models (Sanh et al., 2021), Flan-T5 models (Chung et al., 2024), and T5 models (Raffel et al., 2020). In our experimental setup, we set $m = 5$, meaning that the model can extract up to 5 triplets in PPNet. For metrics, we used Hit@1, Hit@3, and Hit@5 as evaluation metrics to measure the performance of the models.

Model	$k = 1$	$k = 3$	$k = 5$
T5-3B	81.12	86.10	86.80
T5-11B	**86.39**	**88.93**	**89.42**
Flan-T5-3B	78.29	79.25	80.57
Flan-T5-11B	81.37	84.36	85.38
T0-3B	82.40	86.05	88.76
T0-11B	82.33	86.25	87.60

Table 2: Experimental results of PPNet QA under different models at different Hit-k values

Results. As shown in table 2, the T5-11B model demonstrated superior performance in the KGQA task on PPNet with Hit@1 at 86.39%, Hit@3 at 88.93%, and Hit@5 at 89.42%, followed by the

Table 3: Results of Evaluation Metrics and Error Rates Across Different Settings for Assessing the Quality of Generated Responses. RAG refers to RAG in fig. 1 (a); Rtrvr refers to Retriver in fig. 1 (b).

Generator	Method	RAG/Rtrvr	RA	PA	IN	EN	IV	EV	RC	IE
LLaMa2-13B	Zero-shot	-	29.32	35.31	83.42	82.41	85.33	84.19	92.35	86.50
LLaMa2-70B	Zero-shot	-	30.04	39.22	80.08	84.78	81.60	87.32	89.24	88.75
PRLLM-13B	Zero-shot	-	56.07	66.59	64.57	64.05	68.67	64.79	74.15	65.99
PRLLM-70B	Zero-shot	-	55.12	68.14	59.63	65.69	65.08	71.63	73.61	72.00
LLaMa2-70B	CoT	-	58.59	41.04	64.89	69.17	68.04	71.17	72.52	71.93
PRLLM-13B	CoT	-	64.72	56.49	62.61	61.90	72.90	68.80	70.54	69.66
PRLLM-70B	CoT	-	79.39	71.42	55.87	62.34	58.72	66.99	68.89	68.34
LLaMa2-70B	CoT	LLaMa2-70B	66.12	60.40	57.49	62.15	68.44	66.82	68.93	67.03
PRLLM-13B	CoT	PRLLM-13B	80.55	67.93	24.49	24.28	32.16	31.46	38.57	38.25
PRLLM-70B (Mixed)	CoT	GPT-4	87.35	67.26	10.64	11.96	15.28	21.75	30.72	29.60
GPT-4	CoT	GPT-4	85.43	62.18	13.28	12.81	16.58	22.38	36.61	33.47
PRLLM-70B (**PRO**)	CoT	PRLLM-70B	89.18	67.21	**7.80**	**8.33**	**11.69**	**14.10**	**20.03**	**19.08**

T0-3B model and the T5-3B model. This indicates the effectiveness of not only large model size but also model architecture in capturing and retrieving relevant triplets from the knowledge graph.

5 Evaluation on Generation

5.1 Unfaithfulness Error Taxonomy

To evaluate the faithfulness of PRO, we defined a taxonomy of errors (see Table 3) based on Kim et al.'s (2024) typology protocol. Our taxonomy includes six types of errors: Intrinsic Entity Error (IN), Extrinsic Entity Error (EN), Intrinsic Event Error (IV), Extrinsic Event Error (EV), Reasoning Coherence Error (RC), and Irrelevant Evidence Error (IE) (see appendix A for details).

We made specific modifications to Kim et al.'s (2024): Noun-Phrase Errors were consolidated into Entity Errors because, in patents, modifiers can change the meaning significantly. Overgeneralization Errors were merged into Irrelevant Evidence Errors, as both involve information that is not relevant to the current point of contention. These adjustments ensure the error taxonomy is more applicable to the context of patent responses.

Additionally, in patent responses, inventors typically prefer not to have their claims restricted in scope. Therefore, amendments are less desirable compared to arguments. Hence, we introduced a domain-specific metric to measure faithfulness: Recall of Argument (RA) and Precision of Argument (PA). In this context, a generated response judged as an argument (rather than an amendment) is considered true, and vice versa.

5.2 Experimental Setup

To assess the quality of generated responses, we employed human evaluation, recruiting a group of experts in the patent field to evaluate the generated responses based on the six types of errors and whether the generated content was an argument or an amendment. A total of 4,153 generated responses to OAs from 2020-2022, which were unseen by PRLLM before, were evaluated with the ground-truths (GT) (see appendix B for details).

Our evaluation included several different settings. For the paradigm without procedural knowledge, native methods with only generators were used, including two setups: zero-shot and integrated external resources with reasoning (CoT). For the paradigm with procedural knowledge (PRO), this framework included multiple modules such as RAG in fig. 1 (a), retriever in fig. 1 (b), and generator.

In our experiments, we used different combinations of LLMs. For instance, we used GPT-4 for RAG and the retriever, and our PRLLM for the generator. However, Ding et al. (2024) indicate that using the same large language model for both the retriever and generator in a RAG system can be beneficial, as it ensures consistency in language understanding and generation and leverages shared internal representations and knowledge. Therefore, we focused more on using the same LLM across all three modules.

5.3 Results

According to table 3 the PRO framework using PRLLM-70B performed the best, achieving the lowest error rates across all settings. Compared to the closed-source state-of-the-art GPT-4, PRO showed significant improvements: IN had a 7.8% error rate (+41% improvement), EN had an 8.33% error rate (+35% improvement), IV had an 11.69% error rate (+29% improvement), EV had a 14.10% error rate (+37% improvement), RC had a 20.03%

error rate (+45% improvement), and IE had a 19.08% error rate (+43% improvement). On average, PRO outperformed GPT-4 by 39%, demonstrating that our domain-specific PRO framework with PRLLM is superior to generalised LLMs.

Several trends are also revealed: Zero-shot performance is inferior to CoT, and CoT is less effective than the PRO framework, indicating that the inclusion of procedural knowledge results in the most faithful responses. Additionally, extrinsic errors are generally common than intrinsic errors, particularly in larger models, suggesting larger models may introduce irrelevant external information. Lastly, our findings confirm that using mixed LLMs across different modules does not perform as well as using a consistent LLM throughout.

Regarding RA and PA, the results show that PRO consistently achieved the best performance across all settings, indicating high accuracy. Specifically, in CoT setup, especially within PRO, RA was greater than PA. This indicates that the system is more inclined to analyse and rebut the examiner's opinions rather than directly amending the claims to limit the scope of the invention. This behavior aligns with the practical tendencies in the patent industry, where arguments are often preferred over amendments to avoid narrowing the claim scope.

For qualitative results on generated arguments, we refer to table 4. Across three main settings, the formatting of the responses shows no significant issues. In the zero-shot setting, the arguments primarily restate the content of the OA and merely mention a claim without providing analysis, indicated by gray text, showing an IE error. In the CoT setting, the arguments start to take shape but often repeat the examiner's points and prematurely agree with them, which is not encouraged in patent responses. This is shown in red text, indicating an IV error. Additionally, parts of the argument analysis lack complete logical reasoning, shown in orange text, indicating an RC error. In the PRO setting, the generated arguments closely match the GT, both in format and completeness of reasoning. This indicates that the PRO framework effectively enhances the quality of patent responses, demonstrating its effectiveness in the domain.

6 Conclusion

We introduced a domain-specific patent response LLM (PRLLM) that can run locally to ensure privacy during patent responses, as well as a KG of patent precedents (PPNet). Our proposed PRO framework explicitise the procedural knowledge used by patent agents, combining PPNet and RAG with PRLLM, significantly enhancing faithfulness across six error types compared to PRLLM alone and outperforming the state-of-the-art generalised LLM, GPT-4. Future research can focus on prompt tuning in CoT, addressing other aspects of patent responses beyond novelty and non-obviousness, and considering the history trajectory of OAs to further improve response effectiveness.

Table 4: Example Generated Arguments under Different Settings with PRLLM-70B. Text in gray, red, and orange indicates errors, while text in blue indicates correct responses.

Setting	Response
Zero-shot	Claims 1-4 and 12-14 are rejected under 35 U.S.C. 102. ... Independent claim 12 recites an information processing apparatus comprising, in part: a processor configured to: receive a printing request for a document based on a spoken voice of a user; perform a control for outputting a response voice indicating a part including each element of a ...
CoT	Claims 1-4 and 12-14 are rejected under 35 U.S.C. 102. ... Applicant respectfully disagrees. *cited inventor* discloses a voice command for specifying print settings. However, *cited inventor* does not disclose "a second spoken voice of the user ... In contrast, *cited inventor* does not teach any document-related features, paragraph [0015]. Consequently, the selection of a print setting cannot be construed as the claimed feature of "receiving a second spoken voice of the user...
PRO	Claims 1-4 and 12-14 are rejected under 35 U.S.C. 102. ... Specifically, the claimed "designation of a specific part" pertains to the selection of a part from among multiple parts of a document. ... In contrast, *cited inventor* does not disclose any teachings related to documents, and the selected print setting itself is not being printed but is rather a configuration used to print an image. Consequently, the selection of a print setting cannot be construed as the claimed feature of "receiving a second spoken voice of the user, which includes a designation of a specific part among the plurality of parts [of the document] ...
GT	Claims 1-4 and 12-14 are rejected under 35 U.S.C. 102. ... The claimed “designation of specific part” is a selection of part out of the plurality of parts of the document. However, there is no teaching of document in *cited inventor* , and the selected print setting itself is not being printed. It is a configuration used to print a picture. In other words, the selection of print setting cannot be interpreted as the claimed “receive a second spoken voice of the user, which includes a designation of a specific part among the plurality of parts [of the document] ...

References

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. *arXiv preprint arXiv:2306.04136*.

Jung-Mei Chu, Hao-Cheng Lo, Jieh Hsiang, and Chun-Chieh Cho. 2024. From paris to le-paris: Toward patent response automation with recommender systems and collaborative large language models. *arXiv preprint arXiv:2402.00421*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Thomas L Creel. 2013. *Patent Claim Construction and Markman Hearings*. Practising Law Institute.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*.

Yujuan Ding, Wenqi Fan, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meets llms: Towards retrieval-augmented large language models. *arXiv preprint arXiv:2405.06211*.

David Garrod. 2010. *Glossary of Judicial Claim Constructions in the Electronics, Computer and Business Method Arts*. Public Patent Foundation, Benjamin N. Cardozo School of Law.

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. 2024. Simple and scalable strategies to continually pre-train large language models. *arXiv preprint arXiv:2403.08763*.

Kyungha Kim, Sangyun Lee, Kung-Hsiang Huang, Hou Pong Chan, Manling Li, and Heng Ji. 2024. Can llms produce faithful explanations for fact-checking? towards faithful explainable fact-checking via multi-agent debate. *arXiv preprint arXiv:2402.07401*.

Jieh-Sheng Lee and Jieh Hsiang. 2020. Patent claim generation by fine-tuning openai gpt-2. *World Patent Information*, 62:101983.

Hao-Cheng Lo, Jung-Mei Chu, Jieh Hsiang, and Chun-Chieh Cho. 2024. Large language model informed patent image retrieval. *arXiv preprint arXiv:2404.19360*.

OpenAI. 2023. Gpt-4 technical report. arxiv 2303.08774. Technical report.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Priyanka Sen, Sandeep Mavadia, and Amir Saffari. 2023. Knowledge graph-augmented language models for complex question answering. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pages 1–8.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

USPTO. 2023. Fy 2023 agency financial report. Technical report, United States Patent and Trademark Office, Alexandria, VA.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023. Openchat: Advancing open-source language models with mixed-quality data. *arXiv preprint arXiv:2309.11235*.

Yike Wu, Nan Hu, Guilin Qi, Sheng Bi, Jie Ren, Anhuan Xie, and Wei Song. 2023. Retrieve-rewrite-answer: A kg-to-text enhanced llms framework for knowledge graph question answering. *arXiv preprint arXiv:2309.11206*.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, et al. 2023. Effective long-context scaling of foundation models. *arXiv preprint arXiv:2309.16039*.

A Typology

To evaluate the faithfulness of our work, we employed six error types to examine the quality of the generated responses. These error types are adapted from Kim et al.'s (2024) typology protocol to fit the practical scenarios of patent responses. Here is a simple, short, and hypothetical source for exemplary purposes:

> **Patent Application:** The present invention pertains to a device, wherein said device comprises a wood layer positioned above two copper gates to enhance conductivity.
>
> **Office Action:** The claimed invention lacks novelty because a prior art reference also discloses a layer positioned above a gate to enhance conductivity.

Below are the definitions of each error type along with examples relevant to patent responses.

A.1 Intrinsic Entity Error (IN)

An Intrinsic Entity Error occurs when there is a misrepresentation of named entities, quantities, or other surface realizations from a given source. This type of error also includes the incorrect combination of modifiers meant for one entity with another entity.

> **Incorrect Argument:** The present invention comprises a wood layer positioned above *three wooden gates*, which is not identical to the prior art.
>
> **Correct Argument:** The present invention comprises a wood layer positioned above *two copper gates*, which is not identical to the prior art.

A.2 Extrinsic Entity Error (EN)

An Extrinsic Entity Error occurs when new entities are introduced that were not present in the given source, or when modifiers that are not presented in the source are incorrectly combined with entities.

> **Incorrect Argument:** The present invention includes a wood layer positioned above *a gold circuit*, which is not disclosed in the prior art.
>
> **Correct Argument:** The present invention comprises a wood layer positioned above *two copper gates*, which is not disclosed in the prior art.

A.3 Intrinsic Event Error (IV)

An Intrinsic Event Error occurs when events mentioned in the source are misrepresented, either through misunderstanding the event.

> **Incorrect Argument:** The present invention describes *the wood layer is placed beside the gates*.
>
> **Correct Argument:** The present invention describes *the wood layer is positioned above the gates*.

A.4 Extrinsic Event Error (EV)

An Extrinsic Event Error occurs when new events that are not present in the given source are introduced.

> **Incorrect Argument:** The present invention includes *a wood layer is used to store spiritual energy*, which is not disclosed in the prior art.
>
> **Correct Argument:** The present invention *a wood layer is positioned above two copper gates to enhance conductivity*, which is not disclosed in the prior art.

A.5 Reasoning Coherence Error (RC)

A Reasoning Coherence Error occurs when there are logical flaws in the flow of reasoning within the generated explanation, leading to a lack of coherence or weak support for the claim.

> **Incorrect Argument:** The present invention is not identical to the prior art.
>
> **Correct Argument:** The present invention is novel because *the wood layer is positioned above two copper gates, which enhances conductivity*, unlike the prior art that only discloses a single gate.

A.6 Irrelevant Evidence Error

An Irrelevant Evidence Error occurs when the explanation includes evidence that does not directly support the claim, or when it makes broad statements or conclusions that extend beyond the provided evidence.

> **Incorrect Argument:** The present invention *uses eco-friendly materials*, which is not relevant to the enhancement of conductivity discussed in the prior art.
>
> **Correct Argument:** That *the wood layer is positioned above two copper gates to enhance conductivity* is not disclosed in the prior art which discusses a single gate without mentioning such an arrangement.

B Human Evaluation Procedure

To evaluate the effectiveness of each setting described in section 5, we employed a human evaluation method. We recruited a total of 331 experts in the patent field, including patent applicants, engineers, agents, scholars, and attorneys. Of the participants, 30.21% were female, and the median education level was a master's degree. Each participant was randomly assigned a varying number of evaluation cases. Each case included the published and public versions of the patent, the corresponding OA, the actual response to the OA, and a response generated by our experimental setup. Additionally, each participant received experimental instructions and an informed consent form.

After reading the informed consent form and the experimental instructions, participants were required to evaluate the generated response based on the six predefined error types, specifically focusing on the examiner's rejections related to novelty (35 U.S.C. § 102) and non-obviousness (35 U.S.C. § 103). These error evaluations were multi-select.

Beyond the error evaluations, participants also had to determine whether the content of the response was more aligned with an amendment or an argument. They then judged the true response content similarly. Upon completing their tasks, participants received compensation in compliance with labor regulations.

In total, the participants effectively evaluated 4,153 OAs, encompassing approximately 11K points of contention related to novelty and non-obviousness. This evaluation process ensured a comprehensive assessment of the generated responses' quality.