AutoSpec: An Agentic Framework for Automatically Drafting Patent Specification

Anonymous ACL submission

Abstract

Patents play a critical role in driving technological innovation by granting inventors exclusive rights to their inventions. However the process of drafting a patent application is often expensive and time-consuming, making it a prime candidate for automation. Despite recent advancements in language models, several challenges hinder the development of robust automated patent drafting systems. First, the information within a patent application is highly confidential, which often prevents the use of closed-source LLMs for automating this task. Second, the process of drafting a patent application is difficult for even the most advanced language models due to their long context, technical writing style, and specialized domain knowledge. To address these challenges, we introduce AutoSpec, a secure, agentic framework for **Auto**matically drafting patent **Spec**ification. Our approach decomposes the drafting process into a sequence of manageable subtasks, each solvable by smaller, open-source language models enhanced with custom tools tailored for drafting patent specification. To assess our system, we design a novel evaluation protocol in collaboration with experienced patent attorneys. Our automatic and expert evaluations show that AutoSpec outperforms existing baselines on a patent drafting task.

1 Introduction

Drafting a patent application has long been a key component of intellectual property protection. Yet, the drafting process remains a difficult and laborious task. Inventors face many hurdles to patenting their inventions, including high monetary costs and significant time commitments (Wang et al., 2024). This discourages smaller entities and individual inventors from pursuing patents, stifling innovation and competition.

LLMs offer a promising way to alleviate these issues by automating the patent drafting process. Recent work has shown that LLMs can achieve impressive performance on many complex writing tasks, including ones within the legal domain (Katz et al., 2023; Ariai and Demartini, 2025). However, there remain several obstacles to the development and deployment of automatic patent drafting systems.

A major challenge in automating patent drafting is ensuring the security and confidentiality of sensitive invention details. Leakage of this information could compromise the patent's validity or result in an outright rejection of the application. This makes on-premises deployment of patent drafting systems highly desirable, restricting the use of more powerful, proprietary LLMs for solving this task.

These challenges are compounded by the fact that patent drafting remains difficult for even the most advanced language models (Jiang and Goetz, 2025). Patent specifications often span tens of thousand of words, which is beyond what current LLMs can output in a single generation. Patent applications also integrate highly specialized domain knowledge, using a combination of legal and technical language that is difficult for LLMs to replicate (Wang et al., 2024).

To address these issues we propose AutoSpec, an agentic method for **Auto**matically generating patent application **Spec**ification. Given the core details of an invention, AutoSpec produces a full specification by first creating a structured outline. Our outline generation method is constructed to emulate the way that patent attorneys decompose the workflow for drafting patent specification. This outline breaks down the drafting process into manageable subtasks, each of which are solvable by smaller, open-source LLMs in combination with custom-built tools we create specifically for drafting patent specification. These custom tools are designed according to expert input and use a combination of fine-tuning, prompting, and retrieval to effectively draft patent application disclosure.

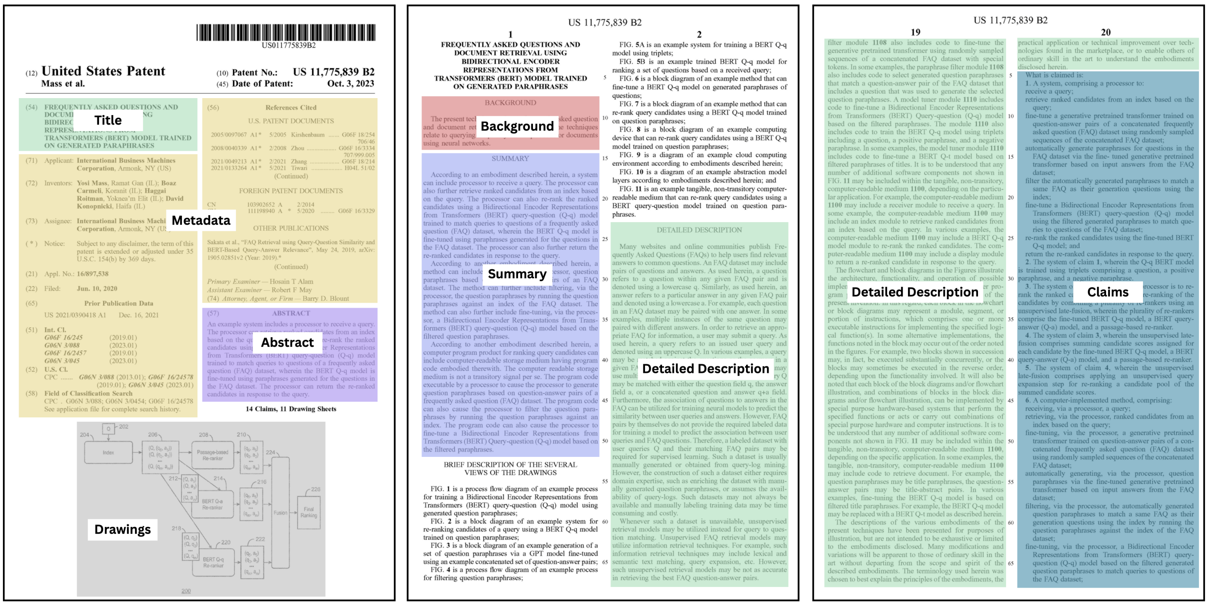

Figure 1: Three pages of a published patent application. The specification consists of the Abstract, Background, Summary, and Detailed Description.

To rigorously assess our system, we introduce a novel evaluation protocol for analyzing patent specifications, developed in collaboration with expert patent attorneys. This protocol is centered on an annotation scheme designed to highlight the critical aspects of high-quality patent disclosure and to standardize the evaluation of machine-generated patent applications. Leveraging this protocol, we evaluate our approach on a patent drafting task using both automated metrics and human expert assessments. Our results show that AutoSpec outperforms existing baselines. We release our evaluation data which consists of 75 machine-generated patent disclosures annotated according to our evaluation protocol. Our contributions are summarized as follows:

- We introduce AutoSpec, a novel agentic framework for drafting patent specification. AutoSpec is built around open-source LLMs, ensuring drafting remains secure and reliable.
- We design an evaluation protocol for evaluating patent disclosure, developed with expert input to capture the key elements of high-quality patent specification.
- We evaluate our framework on a patent drafting task. Our results show that AutoSpec outperforms existing baselines according to automatic and expert evaluations.

2 Background

Patent applications are legal documents that define an invention. They consist of a set of claims and a specification (also referred to as the disclosure). The claims serve to define the scope of an invention concisely and unambiguously. The specification is written based on the content in the claims, and typically includes an abstract, background, summary, and detailed description. Depending on the field of the invention, the patent specification may also contain drawings and drawing descriptions which are included in the detailed description. Figure 1 shows a complete example of a patent application.

Patent language is often very technical, using specialized terms, legal jargon, and sometimes new terms to describe novel concepts. Patents frequently create their own definitions for terms, which can differ significantly from how those words are used in normal language or even within the relevant technical field. These self-defined terms are typically quite artificial and are unlikely to appear in other documents. This makes it difficult for LLMs, which are typically trained on general internet text, to emulate the kind of language in patent applications (Jiang and Goetz, 2025).

Patent specifications are very long with a length of about 13.5k tokens on average (Suzgun et al., 2023). The detailed description comprises the bulk of this length with an average of 11.9k tokens. The specification is written primarily based on claim information which is typically around 1.3k tokens. This means the disclosure must elaborate heavily on the content in the claims and incorporate relevant external concepts that are key to explaining the invention. This combination of long generation and elaboration is also difficult for LLMs which are not extensively trained on long text sequences. For example, despite it's 128k context length LLaMA 3 expends only about 5.5% of it's computational budget training on text sequences longer than 8k tokens (Llama Team, 2024; Touvron et al., 2023).

Patents are granted on the basis of novelty, meaning that if any information about the invention is in the public domain the patent application will be rejected. This limits the use of proprietary LLMs due to privacy concerns (Li et al., 2025). Instead, on premises deployment with open-source models is desirable. However, these models tend to be less capable which further exacerbates difficulty of automatically drafting patent disclosure.

3 Related Work

Prior work in the field of patent generation has typically been centered around generating shorter sections of the specification such as the abstract or summary (Jiang et al., 2024; Zhou et al., 2024). Some additional works have proposed tasks such as next claim generation or creating individual figure descriptions (Aubakirova et al., 2023; Shukla et al., 2025; Lee and Hsiang, 2019; Jiang et al., 2025b). Other recent directions of research include paraphrasing disclosure and simplifying/revising claims (Casola et al., 2023; Jiang et al., 2025a).

The closest related method to ours is Patentformer (Wang et al., 2024). In this work, the authors fine-tune a GPT-J and T5 (Raffel et al., 2023) model on pairs of claims and their closest related paragraph in the specification. While this method does allow models to generate patent disclosure, it makes several simplifying assumptions that diverge from actual patent drafting practices, for instance, the notion that each paragraph in the specification maps to a single claim (Jiang and Goetz, 2025). Training on pairs of claims and their closest matching paragraph also disincentivizes the model from elaborating. Patent disclosure often reiterates the claims to some extent, which may encourage the model to simply reiterate claim content without including the external information needed for drafting the full disclosure.

Prior work on evaluating machine-generated patent disclosure has been limited. Most existing works use metrics such as perplexity or BLEU for evaluation (Papineni et al., 2002). While this can be somewhat effective for short texts, these measures have been shown to struggle evaluating longer sequences (Hu et al., 2024). The most comprehensive work in this area is PatentEval (Zuo et al., 2024) which proposes an error typology for generating new claims based on previous ones and for generating the abstract based on the claims. To our knowledge, there is no existing protocol for evaluating full patent specifications.

4 Method

In this section we outline AutoSpec, a novel agentic method for generating full patent specifications. Our method is designed to take in the claims of a patent application, along with optional OCR-extracted figure text, and generate the specification. AutoSpec consists of three main components: the **orchestrator**, **generator**, and **merger**. An overview of the AutoSpec workflow is illustrated in Figure 2.

4.1 Orchestrator

The orchestrator is designed to process the claims and OCR-extracted figure text of a patent in order to generate an outline for the full disclosure. It starts by building a template composed of standard components found in most patent applications. These include shorter sections such as the abstract and background, as well as the claims themselves. We refer to the items added in this stage as "template items."

The orchestrator then expands the initial template by adding items specific to the particular patent application. This is done by prompting an open-source LLM to extract key technical concepts from the claims, along with some brief information about each concept. Depending on the length and complexity of the claims, this extraction may be performed in a single pass or across multiple iterations. Each item added to the outline in this stage is marked as requiring a retrieval step. We refer to the items added in this stage as "technical items."

For each technical item, the orchestrator uses an internet search api to retrieve relevant information about the concept. This internet search api can be a proprietary tool as the individual technical concepts do not contain any sensitive information that would compromise the integrity of the invention.

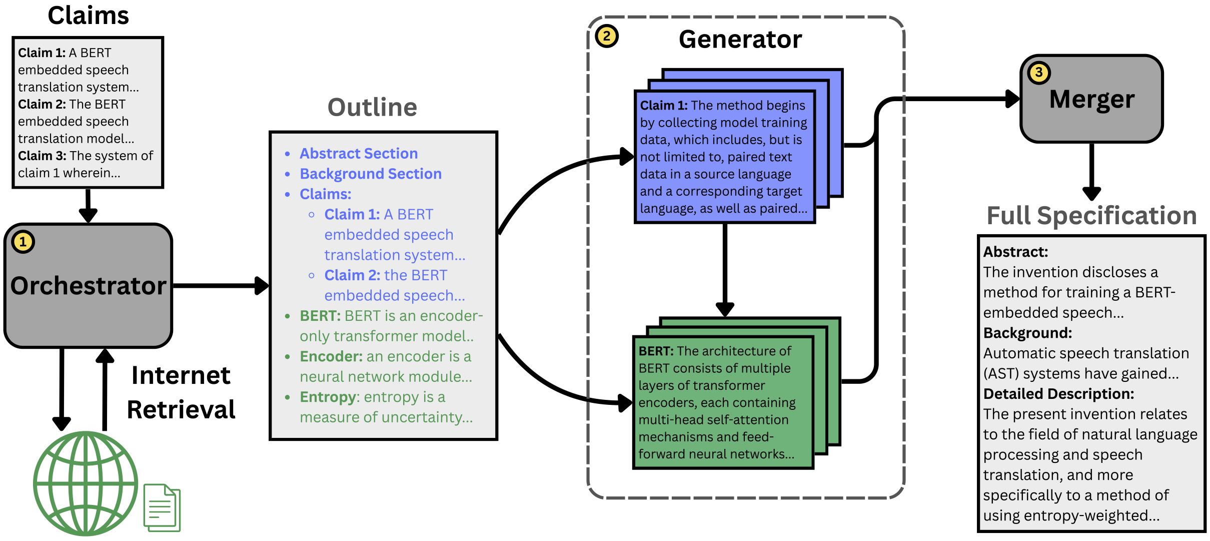
Claims
Claim 1: A BERT embedded speech translation system...
Claim 2: The BERT embedded speech translation model...
Claim 3: The system of claim 1 wherein...
1
Orchestrator
Internet Retrieval
Outline
Abstract Section
Background Section
Claims:
Claim 1: A BERT embedded speech translation system...
Claim 2: the BERT embedded speech...
BERT: BERT is an encoder-only transformer model..
Encoder: an encoder is a neural network module...
Entropy: entropy is a measure of uncertainty...
2
Generator
Claim 1: The method begins by collecting model training data, which includes, but is not limited to, paired text data in a source language and a corresponding target language, as well as paired...
BERT: The architecture of BERT consists of multiple layers of transformer encoders, each containing multi-head self-attention mechanisms and feed-forward neural networks...
3
Merger
Full Specification
Abstract:
The invention discloses a method for training a BERT-embedded speech...
Background:
Automatic speech translation (AST) systems have gained...
Detailed Description:
The present invention relates to the field of natural language processing and speech translation, and more specifically to a method of using entropy-weighted...

Figure 2: A diagram illustrating the workflow of AutoSpec. The orchestrator uses the claims, OCR-extracted figure text, and an internet search tool to generate an outline consisting of template items and technical items. The generator creates the specification for each outline item, then the merger combines them to form the full specification.

The retrieved documents, combined with the original claims, are then passed to a language model, which generates contextually relevant content that aligns with the invention as described in the claims. This output is appended to the corresponding technical item in the outline. This step is essential for ensuring the disclosure meaningfully expands upon the claims rather than merely restating them. We observe that open-source language models often struggle to elaborate effectively on claim content without the aid of external information.

This orchestration process reflects how patent attorneys typically approach drafting disclosures. Attorneys typically write patent applications in segments, some of which are more standardized and primarily involve restating or discussing the claims, while others require more detailed explanation and the inclusion of external information. This approach forms the basis for the two item categories in our structured outline: template items for standard content and technical items for concept-specific elaboration.

4.2 Generator

The generator is responsible for producing all of the text that appears in the final disclosure. It is built based on an open-source LLM that has been trained on patent specifications to better capture the language and style typical of patent applications. This domain-specific training is crucial for ensuring the generated text aligns with standard drafting conventions. The generator powers two custom tools, each designed to handle a different type of item in the outline.

The first tool is designed to generate the specification for each template item in the outline. Since these sections can be written using only the claims and figure information, the tool takes as input the claims, the relevant outline item, and a custom prompt. It then uses the generator to produce the corresponding portion of the disclosure.

The second tool is responsible for generating the disclosure sections corresponding to the technical items in the outline. It takes as input a custom prompt, the claims, the specific outline item, and the existing disclosure content produced by the first tool. Including the previously generated disclosure provides valuable context, enabling the model to produce more coherent and relevant text that better aligns with content of the specification.

The specification for each item in the outline is generated using one of the two tools. The disclosure for all template items must be completed first, as it serves as input for generating the content of the technical items. This ordering mirrors the workflow of human drafters, who often begin with standardized sections before elaborating on specific technical details.

4.3 Merger

The merger is designed to take all of the output given by the generator and combine them to create the final specification. The sections corresponding to each template item are produced independently and merged simply by concatenating the subsections in order. After merging, each paragraph is sequentially numbered. An LLM is then used to integrate the disclosure for the technical items. The model is prompted to provide reasoning about where to insert the paragraph, indicate the insertion position, and generate a revised version of the paragraph to ensure a smooth transition between sections.

5 Evaluation Protocol

To establish a consistent framework for evaluating patent disclosures, we developed a novel evaluation protocol in collaboration with experienced patent attorneys. This protocol is based on an annotation scheme that identifies key elements of high-quality patent specification. Disclosures are assessed across five categories, each rated on a scale from one to five. The category definitions and scoring guidelines are detailed below.

Language style evaluates how closely the language and word choice of the specification matches the style a human would use when writing a description of an invention. The disclosure should be dry and factual, avoiding excessive promotional or advocating language. It should not directly reference the claims (for example, by saying “as given in claim 1”) and should avoid "patent profanity," which are overly specific words like "crucial" or "critical" when describing the invention. These terms unnecessarily limit the scope of the invention and reduce its enforceability.

A score of one reflects pervasive issues, including excessive advocacy, use of patent profanity, and frequent claim references. A score of three indicates a mix of inappropriate and acceptable language, with substantial portions written in a suitable legal tone. A score of five signifies that the specification is almost entirely written in the proper style, using dry, factual language with minimal issues. Minimal advocating language is acceptable.

Elaboration assesses how well the specification expands on the content of the claims. Good disclosure should not simply repeat the claim language but should explain and elaborate on the key technical concepts contained in the claims to help the reader better understand the scope of invention.

A score of one indicates that the specification simply restates the claims with little or no additional detail. A score of three suggests that some elaboration is present, but much of the disclosure closely mirrors the claim content. A score of five reflects thorough elaboration on all key technical concepts needed to adequately understand the invention.

Diversity score evaluates the diversity of language and content in the disclosure. Good specification should not repeat the same content or unnecessarily extend its length by restating the same points. It should also avoid repeating long strings of text and should use some variation in language throughout.

A score of one indicates a high level of repetition, with substantial duplication of content or long strings of identical text. A score of three suggests moderate repetition, though significant portions of the specification show adequate variation. A score of five indicates that the specification has little to no unnecessary repetition, resembling the writing style of a human.

Factual accuracy evaluates how accurate the content in the disclosure is. All of the content should be factual without hallucinations. The disclosure should also not contain any references to nonexistent figures, claims, other sections not present in the patent specification.

A score of one indicates frequent inaccuracies, including false or misleading statements and references to nonexistent elements. A score of three reflects occasional issues, but the majority of the content is accurate and consistent with the claims. A score of five signifies that the disclosure is entirely accurate, with no hallucinated content or invalid references.

Coverage of claims evaluates whether or not there is any content missing in the disclosure. All of the claims should be addressed in the disclosure without any important information omitted.

A score of one indicates that the specification covers few, if any, of the claims, with significant omissions. A score of three means that only partial coverage is provided, approximately half of the claim content is addressed. A score of five indicates that the specification fully incorporates all information from the claims, with no important elements missing.

6 Experiments

We evaluate the effectiveness of our agent through both automated metrics and expert human assessments. The results demonstrate that AutoSpec outperforms multiple baseline approaches in generating patent specifications. Additionally, we per-

Model	PatentSBERTa Similarity↑	BERT for Patents Similarity↑	Patent Profanity↓	Diversity Difference↓
LLaMA 3.3	0.821 (0.078)	0.931 (0.042)	0.44 (0.70)	1.23 (1.22)
GPT-4o (Single-Gen)	0.834 (0.070)	0.925 (0.046)	0.92 (0.97)	1.90 (1.07)
GPT-4o (Multi-Gen)	*0.866 (0.064)*	*0.944 (0.037)*	33.70 (19.94)	1.47 (1.80)
Patentformer	0.821 (0.083)	0.941 (0.036)	**0.02 (0.14)**	3.87 (3.49)
AutoSpec (Template)	0.835 (0.076)	0.943 (0.039)	*0.21 (0.64)*	**0.11 (1.69)**
AutoSpec (Ours)	**0.879 (0.071)**	**0.950 (0.037)**	0.28 (0.75)	*1.23 (1.18)*

Table 1: Automatic evaluation results of AutoSpec against four baselines along with an alternative version of AutoSpec with only template items in the outline. The best scores for each category are **bold**, the second best scores are shown in *italics*, standard deviations are in parenthesis.

form an error analysis comparing AutoSpec with two baseline models, using feedback provided by our expert evaluators. To support further research, we release the expert evaluation dataset, which includes 75 machine-generated patent disclosures annotated according to the protocol described in Section 5.

6.1 Implementation

For our implementation of AutoSpec we utilize LLaMA 3.3 70b as the base LLM. We fine-tune this model using LoRA (Hu et al., 2021) on data consisting of claim-specification pairs. Our data is sourced from a subset of the HUPD dataset (Suzgun et al., 2023) supplemented by data scraped from Google patents. The HUPD dataset only includes patent applications up to 2018, so we collect this additional data in order to incorporate more recent patent specifications. We make this data publicly available for replication purposes.

The orchestrator, generator, and merger all use this trained model for their tasks. We find that LoRA fine-tuning allows the model to better replicate patent language while retaining it's general-purpose instruction-following capabilities. Additional details for our implementation can be found in Appendix A.

6.2 Baselines

We evaluate our method against four baseline approaches, described below. Further implementation details for each method are in Appendix A.

LLaMA 3.3 We use the LLaMA 3.3 70b parameter base model as our first baseline. This model has no fine-tuning or any of the additional components given in Section 4.

GPT-4o (Single-Gen) For this baseline, we prompt GPT-4o to generate the entire patent disclosure in a single pass. The input includes only the template items defined in Section 4, without any technical items.

GPT-4o (Multi-Gen) This baseline also uses GPT-4o, but generates the disclosure section by section. For each template item, the model is prompted using the current item along with previously generated content to maintain coherence across the full draft.

Patentformer The final baseline is based on the Patentformer method introduced by Wang et al. (2024). We fine-tune the LLaMA 3.3 70B model on their released dataset of claim-specification pairs. Disclosure is generated iteratively, with the model conditioned on the claims and the previously generated paragraph.

6.3 Automatic Evaluation

For our automatic evaluation we generated patent disclosures for 100 published patents selected by two patent experts in the field of biotechnology. The generated disclosures were assessed using the following metrics. Additional implementation details are provided in Appendix A.

Semantic Similarity We assess semantic similarity using two embedding models specifically trained for use on patent text: PatentSBERTa (Bekamiri et al., 2024) and BERT for Patents[1]. For each model, we compute embeddings for both the original and generated disclosures, and calculate cosine similarity to measure alignment.

Patent Profanity We approximate the language quality of each method by checking for the presence of patent profanity within the disclosure. This is done via keyword matching using a curated list of problematic terms and phrases provided by patent experts.

N-gram Diversity We use n-gram diversity from

[1] https://github.com/google/patents-public-data

Method	Language Style	Elaboration	Diversity	Factual Acc.	Coverage
GPT-4o (Multi-Gen)	3.24 (0.60)	**3.68 (0.63)***	3.08 (1.00)†	3.92 (0.57)†	3.84 (0.75)†
Patentformer	3.80 (1.08)†	2.20 (1.00)	2.28 (1.10)	2.84 (1.40)	1.96 (0.98)
AutoSpec	**3.96 (0.68)**†	3.24 (0.88)†	**3.60 (0.87)***	**4.04 (0.98)**†	**4.32 (0.99)***

Table 2: Expert evaluation results of GPT-4o (Multi-Gen), Patentformer, and AutoSpec. The best scores for each category are shown in **bold**, standard deviations are in parenthesis. Statistically significant improvements (independent two-sample t-test, $p < 0.05$) over the both baselines are marked with *, improvements over one baseline are marked with †.

Li et al., 2016 to estimate the linguistic variety within each disclosure. Patents naturally include some repetition, so we report the absolute difference in average n-gram diversity between the generated disclosure and the original specification to capture language diversity.

The results of our automatic evaluations are presented in Table 1. We also include results for a variant of AutoSpec that only uses the template items from the outline to generate the disclosure. AutoSpec achieves the highest scores on both semantic similarity metrics, with GPT-4o (Multi-Gen) ranking second. However, GPT-4o (Multi-Gen) performs poorly in terms of avoiding patent profanity, averaging over 33 flagged instances per disclosure. In contrast, Patentformer achieves the best performance on this metric, with an average of just 0.02 instances, followed by both AutoSpec agents. Notably, all top-performing methods utilize models fine-tuned on patent disclosures, underscoring the importance of domain-specific training for accurately replicating the style and structure of patent language.

The AutoSpec agent that uses only template items exhibits the smallest difference in language diversity compared to the gold specifications, followed by the full-outline AutoSpec agent. All other baseline methods, with the exception of Patentformer, produce disclosures with greater language diversity than the original specifications. Patentformer performs the worst on this metric, showing a diversity difference more than twice as large as the next closest method.

Llama 3.3 and GPT-4o (Single-Gen) underperform across all metrics. Both attempt to generate the entire specification in a single generation, which likely contributes to their reduced performance. This contrasts with the other methods, all of which incorporate some form of task decomposition in the drafting process. These findings show the importance of breaking the drafting task into smaller sub-tasks to improve output quality.

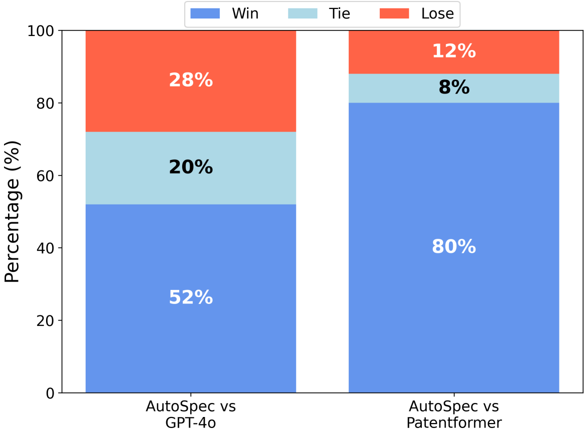

Figure 3: AutoSpec's win, loss, and tie rate vs GPT-4o and Patentformer according to expert rankings.

7 Expert Evaluation

For our expert evaluation, we compared the performance of AutoSpec against Patentformer and GPT-4o (Multi-Gen). We generated patent disclosures for 25 biotechnology patents, selected by two experienced patent professionals. Each disclosure was evaluated using the annotation scheme described in Section 5. In addition, the experts were asked to rank the disclosures based on their usefulness to a patent attorney as a first draft. To the best of our knowledge, this represents the first expert evaluation conducted on full, machine-generated patent specifications. We measured inter-annotator agreement using Kendall's Tau (Kendall, 1938) and obtained a score of 0.15, indicating a statistically significant correlation between expert ratings (see Appendix A for details).

The results of our expert evaluation are presented in Table 2. AutoSpec outperforms all baselines across every metric except for elaboration, where GPT-4o achieves the highest score. Patentformer performs relatively poorly overall, though it demonstrates strong performance in language style, comparable to AutoSpec and notably better than GPT-4o. AutoSpec's respective win rates versus GPT-4o and Patentformer are given in Figure 3. Against GPT-4o, AutoSpec achieves a win rate of 52% and a loss rate of 28%. Its performance against Patentformer is even stronger, with a win rate of 80% and a loss rate of only 12%.

7.1 Expert Comments and Error Analysis

In addition to providing annotations and rankings, our expert evaluators also provided comments on each disclosure. They observed that the GPT-4o method frequently employed patent profanity, a finding consistent with our automatic evaluation. Specifically, the model often explicitly referenced claims and regularly used terms like "crucial" and "critical" to describe the invention. Experts also noted that its tone was overly conversational and tended to advocate for the invention rather than presenting it in the dry, factual style typical of patent specifications. This frequent use of advocating language sometimes led to incomplete explanations of claim elements, which affected it's ability to adequately address all of the claims in the disclosure.

Despite these shortcomings, GPT-4o demonstrated a notable strength in its ability to elaborate on claim concepts. It managed to do this effectively without relying on external tools such as internet search or retrieval mechanisms. This capability is likely attributable to the model's scale, both in terms of its size and the scope of its training data, which appears sufficient to support robust technical elaboration directly from its internal knowledge. This is in contrast to open-source models which have a more difficult time elaborating on technical concepts without leveraging external tools.

Experts commented that Patentformer had a tendency to hallucinate figures and certain aspects of the claims. It was often repetitive, frequently restating claim language without deeper elaboration. One area where Patentformer excelled was in it's language style, likely due to the model's extensive fine-tuning on patent disclosure. These results further highlight the importance of leveraging fine-tuning for automatic patent drafting. Patent specification is a unique instance where dry, technical language is highly desirable. This runs counter to the typical use cases for LLMs which are trained to be conversational and engaging. Therefore it is difficult for LLMs to emulate this language through prompting alone.

While the AutoSpec agent performed the best in general, there were notable failure modes highlighted during the evaluation. Many of these were centered around it's elaboration, which was the only evaluation category where it did not score the highest. More details on these shortcomings are in the Limitations section.

8 Future Work

One promising direction of future work is to extend AutoSpec to draft other sections of patent applications such as the claims. Existing approaches to claim drafting typically generate claims either from prior claims or directly from the specification. However, this does not reflect the real-world drafting process, where patent attorneys often base claims on input provided by inventors. Incorporating this workflow into AutoSpec could lead to a more effective agent which can effectively generate both the claims and specification based on inventor-provided invention details.

Another potential direction is to develop more robust automatic metrics based on our evaluation protocol. In particular, assessing how closely LLM-generated annotations align with expert ratings could improve evaluation quality. This, in turn, could support the use of online training methods, such as reinforcement learning, to further refine patent drafting models.

9 Conclusion

Patent applications are key to protecting intellectual property and driving technological innovation. However, many smaller entities and individual inventors face obstacles to patenting their inventions due to the significant costs associated with drafting a patent application. To alleviate these issues we proposed AutoSpec, an agentic framework for automatically drafting patent specifications. AutoSpec's design is informed by expert input and mirrors the structured approach patent attorneys use to draft disclosures. To evaluate our framework we developed a novel, expert informed evaluation protocol for evaluating full patent disclosures. We evaluated our method using automatic and expert evaluations and found that our AutoSpec agent outperforms existing baselines on a patent drafting task. Additionally, we release a dataset of machine-generated patent disclosures annotated according to our evaluation protocol, providing a valuable resource for further research.

Limitations

Despite its strengths, the AutoSpec system has limitations. In our expert evaluations, we observed instances where AutoSpec took certain technical concepts in the invention out of context. For example, in one instance a claim set made reference to “scaffolding” which in the context of chemistry refers to the core structure of a molecular compound or a class of compounds. However, the system mistakenly included a section in the disclosure discussing scaffolding in the context of construction. To protect sensitive claim content, we exclude claim text from the internet search component, but this occasionally leads to the retrieval and inclusion of irrelevant or misleading information in the disclosure.

Another limitation is that AutoSpec is currently built around text-only language models. Prior research has demonstrated that extracting OCR text from patent drawings can enable accurate figure descriptions (Wang et al., 2024; Shukla et al., 2025). However, integrating multimodal models that can process both text and images would likely enhance the quality of the generated specifications, making this a promising direction for future development.

Both our framework and evaluation protocol were developed with input from patent attorneys who practice in the United States. Since patent standards and disclosure requirements vary across jurisdictions, AutoSpec may not generalize well to other countries’ legal frameworks. Further work is needed to adapt and evaluate the system for use in patent offices outside the United States.

References

Amine Abdaoui and Sourav Dutta. 2023. Attention over pre-trained sentence embeddings for long document classification. *Preprint*, arXiv:2307.09084.

Farid Ariai and Gianluca Demartini. 2025. Natural language processing for the legal domain: A survey of tasks, datasets, models, and challenges. *Preprint*, arXiv:2410.21306.

Dana Aubakirova, Kim Gerdes, and Lufei Liu. 2023. Patfig: Generating short and long captions for patent figures. *Preprint*, arXiv:2309.08379.

Hamid Bekamiri, Daniel S. Hain, and Roman Jurowetzki. 2024. Patentsberta: A deep nlp based hybrid model for patent distance and classification using augmented sbert. *Technological Forecasting and Social Change*, 206:123536.

Silvia Casola, Alberto Lavelli, and Horacio Saggion. 2023. Creating a silver standard for patent simplification. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’23, page 1045–1055. ACM.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Steffen Herbold. 2024. Semantic similarity prediction is better than other semantic similarity measures. *Preprint*, arXiv:2309.12697.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Yutong Hu, Quzhe Huang, Mingxu Tao, Chen Zhang, and Yansong Feng. 2024. Can perplexity reflect large language model’s ability in long text understanding? *Preprint*, arXiv:2405.06105.

Lekang Jiang and Stephan M. Goetz. 2025. Natural language processing in the patent domain: a survey. *Artificial Intelligence Review*, 58(7).

Lekang Jiang, Pascal A. Scherz, and Stefan Goetz. 2025a. 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.

Lekang Jiang, Caiqi Zhang, Pascal A. Scherz, and Stefan Goetz. 2025b. Can large language models generate high-quality patent claims? In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1272–1287, Albuquerque, New Mexico. Association for Computational Linguistics.

Lekang Jiang, Caiqi Zhang, Pascal A Scherz, and Stephan Goetz. 2024. Can large language models generate high-quality patent claims? *Preprint*, arXiv:2406.19465.

Daniel Martin Katz, Dirk Hartung, Lauritz Gerlach, Abhik Jana, and Michael J. Bommarito II. 2023. Natural language processing in the legal domain. *Preprint*, arXiv:2302.12039.

M. G. Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Jieh-Sheng Lee and Jieh Hsiang. 2019. Patent claim generation by fine-tuning openai gpt-2. *Preprint*, arXiv:1907.02052.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 110–119, San Diego, California. Association for Computational Linguistics.

Siyan Li, Vethavikashini Chithrra Raghuram, Omar Khattab, Julia Hirschberg, and Zhou Yu. 2025. PAPILLON: Privacy preservation from Internet-based and local language model ensembles. 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 3371–3390, Albuquerque, New Mexico. Association for Computational Linguistics.

AI @ Meta Llama Team. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Preprint*, arXiv:1908.10084.

Chantal Shaib, Joe Barrow, Jiuding Sun, Alexa F. Siu, Byron C. Wallace, and Ani Nenkova. 2025. Standardizing the measurement of text diversity: A tool and a comparative analysis of scores. *Preprint*, arXiv:2403.00553.

Shreya Shukla, Nakul Sharma, Manish Gupta, and Anand Mishra. 2025. Patentlmm: Large multimodal model for generating descriptions for patent figures. *Preprint*, arXiv:2501.15074.

Mirac Suzgun, Luke Melas-Kyriazi, Suproteem Sarkar, Scott D Kominers, and Stuart Shieber. 2023. The harvard uspto patent dataset: A large-scale, well-structured, and multi-purpose corpus of patent applications. In *Advances in Neural Information Processing Systems*, volume 36, pages 57908–57946. Curran Associates, Inc.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Juanyan Wang, Sai Krishna Reddy Mudhiganti, and Manali Sharma. 2024. Patentformer: A novel method to automate the generation of patent applications. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1361–1380, Miami, Florida, US. Association for Computational Linguistics.

Shu Zhou, Xin Wang, Zhengda Zhou, Haohan Yi, Xuhui Zheng, and Hao Wan. 2024. The master-slave encoder model for improving patent text summarization: A new approach to combining specifications and claims. *Preprint*, arXiv:2411.14072.

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.

A Additional Implementation Details

In this section we give further details on our implementation of AutoSpec, baselines, and evaluations. All of our source code, including prompts will be released upon acceptance. We will also be releasing the data we use to train AutoSpec as well as our expert-annotated evaluation data.

A.1 AutoSpec

For AutoSpec, we train a LLaMA 3.3 70b parameter base model for one epoch using LoRA finetuning on four NVIDIA RTX A6000 GPUs. We use a learning rate of 5e-06 and an effective batch size of 8 on a dataset containing 1,354 patents, 750 come from the publicly available HUPD dataset (Suzgun et al., 2023) and 574 were scraped from Google patents. We quantize the model to four bits using k-means quantization for use at inference time. We use the OpenAI web search tool for internet retrieval and retrieve the top one document that matches the search query.

A.2 Patentformer

We recreate the Patentformer method by training a LLaMA 3.3 70b parameter base model using the Patentformer dataset released by Wang et al., 2024. We use LoRA fine-tuning on four NVIDIA RTX A6000 GPUs for one epoch with a learning rate of 5e-05 and an effective batch size of 8. We quantize the model to 4 bits using k-means quantization for inference.

The Patentformer dataset consists of claims mapped to single paragraphs in the specification. Certain specification paragraphs also have OCR-extracted figure texts mapped to them. The dataset also includes the previous paragraph in the disclosure for context. The model is trained using this dataset to generate single specification paragraph from a single claim and the previously generated paragraph. The patentformer dataset includes tags to provide additional context to the model for generating specification, we remove these from the text during our final evaluations. See Wang et al., 2024 for complete details.

A.3 GPT-4o and LLaMA 3.3

To create the GPT-4o and LLaMA 3.3 baselines we use prompt engineering to create the final bots. We focused on prompting the bots to adopt a legal language style and to format their disclosure without any markdown. GPT-4o was particularly prone to generating specification with markdown headers and lists. Whereas patent disclosure should be formatted as a series of paragraphs. Prompt engineering alleviates this to some degree, however GPT-4o still occasionally generates markdown in it's disclosure even explicitly told not to. This is also the case for language style. Both GPT-4o and LLaMA 3.3 struggle to replicate the language style of patent applications despite extensive prompt engineering and in-context examples.

A.4 Semantic Similarity

To measure semantic similarity we use two different SentenceTransformers (Reimers and Gurevych, 2019) models that have been trained extensively on patent data. Both of these models have a maximum sequence length of 512 tokens, which is below the length of a typical patent disclosure. To measure the similarity between the full disclosures, we segment the document into smaller chunks, create embeddings for each chunk, then combine the embeddings using mean pooling (Abdaoui and Dutta, 2023). Prior work has shown that specially trained models tend to be more effective at measuring sematic similarity than n-gram based metrics (Herbold, 2024). Therefore we choose this method for measuring semantic similarity as opposed to other metrics such as BLEU or ROUGE.

A.5 Patent Profanity

To measure patent profanity we look for the following terms within the disclosure: “crucial”, “critical”, “prior art”, “necessary aspect”, “necessary component”. We also look for the term “claim” followed by an integer to assess where the model directly references the claims. These terms were provided to us by patent attorneys.

A.6 N-Gram Diversity

N-gram diversity is defined as the ratio of unique n-gram counts to all n-gram counts in a document (Shaib et al., 2025). We calculate the n-gram diversity for each disclosure using the following formula:

$$\mathrm{NGD}(D) = \sum_{n=1}^{10} \frac{\#\ \text{unique } n\text{-grams in } D\oplus}{\#\ n\text{-grams in } D\oplus}$$

A.7 Expert and Automatic Evaluations

For both our expert an automatic evaluations we relied on patent attorneys to select our evaluation sets. This was done for the expert evaluation to ensure that the attorneys had the expertise to assess the patents. We also did this for the automatic evaluation to ensure a quality selection of patents. Not all published patents are of equal quality, and one key feature of a good patent application is its ability to withstand litigation. The attorneys we collaborated with have a strong track record of doing this therefore we chose to have them select our automatic evaluation set as opposed to randomly selecting patents from an existing dataset. Our data collection protocol is IRB approved.

During the expert evaluation we presented the attorneys with three different disclosures without telling them which specification was generated by which method. They rated each disclosure according to our evaluation protocol then ranked them based on how useful they would be if given to them as a first draft of a disclosure. These rankings are used to determine our win-rate in Table 3.

To measure the inter-annotator agreement between our raters we used Kendall's Tau. Kendall's Tau is a measure of correlation between two sets of ordinal data, ranging from -1 to +1. A value of +1 indicates perfect agreement in rankings, -1 indicates perfect disagreement, and 0 indicates no association. This value can also be used for a statistical test with a null hypothesis of no correlation between the rankings.

Both patent attorneys annotated five overlapping patents during the evaluation which as used as the data for calculating inter-annotator agreement. We calculated Kendall's Tau on our data and found a value of 0.15 which indicates a slight, but statistically significant correlation between the ratings for our sample size. We also measured the weighted Cohen's kappa for the rating and found a value of 0.17 which also indicates slight correlation (Cohen, 1960).

A.8 System Ablations

We relied on small scale expert evaluations to test the different components of AutoSpec such as the retrieval tool, prompting methods, and inclusion of the different modules. This was done by generating two disclosures for each ablation and having one expert rate the outputs using our evaluation protocol from Section 5. We used this method to determine the final design for our system. We found this method to be more reliable than automatic evaluations.