PAP2PAT: Benchmarking Outline-Guided Long-Text Patent Generation with Patent-Paper Pairs

Valentin Knappich[1,2] **Anna Hätty**[1] **Simon Razniewski**[3] **Annemarie Friedrich**[2]

[1]Bosch Center for AI, Germany
[2]University of Augsburg, Germany
[3]ScaDS.AI & TU Dresden, Germany

{valentin.knappich,anna.haetty}@de.bosch.com, simon.razniewski@tu-dresden.de, annemarie.friedrich@uni-a.de

Abstract

Dealing with long and highly complex technical text is a challenge for Large Language Models (LLMs), which still have to unfold their potential in supporting expensive and time-intensive processes like patent drafting. Within patents, the description constitutes more than 90% of the document on average. Yet, its automatic generation remains understudied. When drafting patent applications, patent attorneys typically receive invention reports (IRs), which are usually confidential, hindering research on LLM-supported patent drafting. Often, pre-publication research papers serve as IRs. We leverage this duality to build PAP2PAT, an open and realistic benchmark for patent drafting consisting of 1.8k patent-paper pairs describing the same inventions. To address the complex long-document patent generation task, we propose chunk-based outline-guided generation using the research paper as technical specification of the invention. Our extensive evaluation using PAP2PAT and a human case study show that LLMs can effectively leverage information from the paper, but still struggle to provide the necessary level of detail. Fine-tuning leads to more patent-style language, but also to more hallucination. We release our data and code at https://github.com/boschresearch/Pap2Pat.

1 Introduction

Securing intellectual property is a long and costly process that requires both deep technical knowledge and expertise in patent law. This motivates the use of technology to boost patent attorney productivity. Natural language processing (NLP) already assists prior art search (Shalaby and Zadrozny, 2019; Stamatis, 2022; Pujari et al., 2021) and patent landscaping (Choi et al., 2022; Pujari et al., 2022). Research on Large Language Models (LLMs) in the patent domain has recently gained momentum (Shomee et al., 2024; Jiang and Goetz, 2024; Casola and Lavelli, 2022; Wang et al., 2024b), but patent drafting remains a largely manual task.

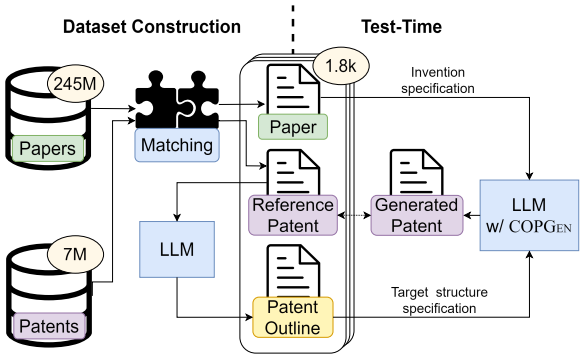

Figure 1: PAP2PAT dataset creation (left) and experimental setup (right).

A patent typically consists of claims, which define the invention and the legally relevant scope of protection, and a description, which provides technical details in sections like *Field of the Invention*, *Background*, *Summary*, and *Detailed Description*. Prior work has primarily focused on generating abstracts and claims (Hamborg et al., 2017; Lee, 2020; Christofidellis et al., 2022; Lee, 2023; Zuo et al., 2024; Lee, 2024; Bai et al., 2024b). The description makes up over 90% of the document, [1] implying that large productivity gains are expected from writing support for this section. Yet, generating them remains a significant challenge for LLMs due to their length, technical complexity, and specialized language (Wang et al., 2024a,c). Existing work on automatic patent generation suffers from a number of shortcomings, such as ill-posed task setups, lack of open benchmarks, and disregard for patent descriptions (Jiang and Goetz, 2024). In this paper, we tackle all of these issues: we propose a novel setup, evaluation metrics, and outline-guided models for description generation in a realistic setting using open and human-created data.

Inventors typically submit invention reports

[1]Measured on our dataset using the Llama-3 tokenizer: 0.7% abstract, 91.8% description, 7.5% claims.

Findings of the Association for Computational Linguistics: ACL 2025, pages 9524–9554
July 27 - August 1, 2025

(IRs), which patent attorneys formalize into patent applications. In many research labs, it is common to use a pre-publication paper as invention report, which leads to so-called *patent-paper pairs* (PPPs, Murray (2002)), an unrecognized treasure for AI research. To facilitate the study of LLMs on patent drafting, we create PAP2PAT, a new benchmark of 1.8k carefully identified PPPs and patent outlines. We develop and validate a method for reliably matching patents and papers describing the same invention (see Figure 1). For LLM-supported patent drafting, we envision a practical setting in which the attorney, given a paper, provides an outline for the patent. This outline acts as a flexible mechanism to control the document structure and content while keeping manual effort low.

A major challenge of the proposed task is document length: patent descriptions in PAP2PAT are on average 18k tokens long, some exceeding 180k tokens. While current LLMs increasingly support long context windows, they struggle to generate similarly long outputs (Liu et al., 2024; Bai et al., 2024a; Wu et al., 2025; Ye et al., 2025). As a remedy, we propose *chunk-based outline-guided patent generation* (COPGEN), which effectively generates long patent documents in chunks.

Evaluating generated patents poses significant challenges due to their length, their technical complexity, and the high cost of manual evaluation. No standard evaluation metrics are established in the literature, and prior work (Wang et al., 2024c,a) commonly resorts to standard text similarity metrics which do not work well on long documents (Lattimer et al., 2023; Que et al., 2024). We adapt a suite of metrics based on natural language inference (NLI) and authorship attribution to the specific case of evaluating factual correctness, coverage, and language style for long-form patent generation on PAP2PAT. Our main contributions are:

(1) We create and release PAP2PAT, a new benchmark for patent drafting based on open data that closely aligns with real-world settings.
(2) We derive a comprehensive suite of automatic evaluation metrics for evaluating generated patents.
(3) We propose a chunk-based outline-guided patent generation approach with effectively controllable output length. Our method increases coverage considerably while keeping factuality high.
(4) We conduct extensive evaluations finding that state-of-the-art LLMs can effectively use information from the papers, but still struggle to provide the necessary level of detail, and that fine-tuning leads to much higher stylistic similarity with patents, but substantially decreased factuality.
(5) Our human evaluation confirms these findings and indicates promising potential for productivity gains of patent attorneys.

2 Related Work

Patent-Paper Pairs (PPPs). Many research labs practice concurrent patenting and academic publishing. Murray and Stern (2005) find that almost 50% of their sampled academic papers from *Nature Biotechnology* have a corresponding US patent. In economics, PPPs have been used to study innovation dynamics, like whether patenting promotes or hinders the free flow of innovations (Murray and Stern, 2005; Magerman et al., 2011). In that context, several approaches to finding PPPs have been proposed: Murray (2002) and Murray and Stern (2005) identify pairs manually by analyzing their full texts and citation networks. Magerman et al. (2010) and Van Looy et al. (2011) explore several data mining techniques. We refine and extend their approach, and add criteria for author overlaps, date ranges, competing candidates, and licenses. Gans et al. (2017) propose a taxonomy of PPPs that includes both to 1-to-1 matches and m-to-n matches. To ensure that papers are a solid source of information about the invention, we design our matching procedure to find only 1-to-1 matches. We publish, to the best of our knowledge, the first PPPs dataset for NLP research.

Patent Generation. Most prior work on patent generation has focused on titles, abstracts, and claims. Christofidellis et al. (2022) train a multitask GPT2 model to generate these parts. Lee (2023) pre-trains a GPT-J-6B architecture on entire patents and evaluates it on claim generation. Zuo et al. (2024) use GPT-3.5-turbo and Llama-2 to generate abstracts from claims, and claims from previous claims. Wang et al. (2024a) experiment with the generation of individual description paragraphs based on patent claims and drawing descriptions. In practice, it is not likely that claims are already finalized at the time of writing of the description section. In their work concurrent to ours, Wang et al. (2024c) leverage an agent framework to generate complete patents based on purely automatically generated invention specifications. They share our core principle of using a divide-and-conquer strategy for long-document generation, but

Split	# pairs	# patent tokens	# paper tokens	# outline bullets		
				short	medium	long
train	1000	17.8k ± 15.1k	7.9k ± 4.5k	36.8 ± 29.2	73.5 ± 60.0	149.0 ± 122.0
val	242	18.2k ± 16.3k	8.0k ± 4.1k	37.4 ± 30.8	74.4 ± 62.9	150.6 ± 127.5
test	500	18.1k ± 13.2k	8.1k ± 3.9k	37.5 ± 24.7	74.9 ± 50.9	151.6 ± 103.7
nc-test	71	18.4k ± 13.9k	9.5k ± 4.6k	37.8 ± 22.7	76.0 ± 46.4	154.4 ± 94.8
all	1813	17.9k ± 14.7k	8.1k ± 4.3k	37.1 ± 28.0	74.1 ± 57.5	150.1 ± 117.0

Table 1: PAP2PAT statistics. Values are reported as mean ± std. Token counts correspond to the Llama-3 tokenizer.

evaluate only using surface-level metrics. Our more realistic setting relies on real-world invention specifications and provides deeper insights due to more sophisticated evaluation metrics.

Outline-guided Generation is a paradigm in which LLMs use an outline to produce longer, more structured and coherent text. Outlines are either generated in a planning stage (Drissi et al., 2018; Yao et al., 2019; Sun et al., 2022; Yang et al., 2022; Lee et al., 2024; Wang et al., 2024d; Shao et al., 2024) or provided as input (Fang et al., 2021; Spangher et al., 2022; Yang et al., 2023; Li et al., 2023b). They are created using extraction of key words (Yao et al., 2019), phrases (Fang et al., 2021), or sentences (Drissi et al., 2018; Sun et al., 2022; Yang et al., 2022; Li et al., 2023b), generated using LLMs (Yang et al., 2023) or defined interactively (Goldfarb-Tarrant et al., 2019). In our work, we posit that outlines should be provided by the patent attorney to satisfy the high demand for user control.

3 PAP2PAT Benchmark

We present the PAP2PAT dataset containing 1.8k PPPs, each annotated with three outlines for generation. We describe the steps taken to construct PAP2PAT, analyse the obtained corpus, and propose evaluation metrics for our new benchmark.

3.1 Dataset Construction

We here give an overview of the construction of PAP2PAT (for details, see Appendix A).

Scraping PPPs. The patent and paper in a PPP typically do not cite each other, so we cannot rely on front-page or in-text citations (Marx and Fuegi, 2020, 2022) to find PPPs. Therefore, prior work has developed heuristics to match patents and papers based on document metadata (Magerman et al., 2010; Van Looy et al., 2011). We use the USPTO dataset[2] containing 6.7M patent applications from 2005 to April 2024. For each patent, we query SemOpenAlex (Färber et al., 2023) using SPARQL and retrieve papers with overlapping authors lists and publication dates. We filter the results based on titles, abstracts, other candidate matches for the same patent, and paper licenses. Table 2 shows the remaining number of candidates after each filtering step. A major limiting factor for our benchmark size are the restrictive licenses of many scientific articles.[3]

Manual Validation. To verify the precision of our matching pipeline, the first author performs a manual validation. We randomly sample 60 PPPs, read both abstracts, skim the documents, compare the figures and get an overview of the authors' related work. We spend a total of 5 hours, i.e., 5 minutes per pair on average. In 55/60 (91.7%) pairs, the paper describes the invention as a core contribution. In three pairs, the best match for the patent would have been a prior paper by the same authors. In two pairs, the paper would have been best matched to a related but different patent by the same inventors. In these five imperfect matching cases, the papers still provide meaningful training and evaluation signals, as they are highly relevant to the invention and contextualized by the outline. Overall, the precision of our matching approach is high.

Outline Generation. Outlines consist of target headings and short bullet points summarizing the document structure and high-level content of every section (see Figure 2 and Appendix E). For PAP2PAT, we generate them automatically from the original patents using Llama-3 70B (Dubey et al., 2024). In practice, they are provided by the user. To ensure that the model adheres to the desired output format, we use SGLang (Zheng et al.,

[2] https://bulkdata.uspto.gov

[3] ACL with its CC-BY license being a positive exception.

Filter	# pairs
Authors + Date	930k
+ Term Overlap	100k
+ Distinctiveness	21k
+ Permissive License	1.8k

Table 2: Filter criteria and remaining number of candidates.

2023) for constrained decoding. We enforce a fixed number of bullet points n_{bullets} per section, where n_{bullets} is proportional to the length of the text in that section (n_{chars}) and defined as

$$n_{\text{bullets}} = \begin{cases} \max(1, \lfloor n_{\text{chars}}/l \rfloor) & \text{if } n_{\text{chars}} > 0 \\ 0 & \text{else} \end{cases}$$

where l is the number of characters that each bullet point summarizes on average. We create outlines with three levels of granularity: *long* (l=500, avg. 150 items), *medium* (l=1000, avg. 74 items) and *short* (l=2000, avg. 37 items), see Table 1. The average bullet point length is 5.4 $\pm$ 2.3 words. For each section, we additionally provide the number of characters in the original patent to signal the desired content lengths during generation.

3.2 Task Description and Data Splits

We propose the following generation task for PAP2PAT: Given a patent outline O and a research paper C containing a specification of the invention, models should output a patent P. The input data provides the desired output length in characters per section. We split our dataset randomly into *train* (n=1000), *test* (n=500) and *validation* (n=242), see Table 1. We additionally create a non-contaminated test set (*nc-test*) that contains all pairs with patents published in 2024 (n=71), i.e., after the pretraining cut-off date of all evaluated open-weight LLMs, addressing the concern that LLMs might have seen test data during pretraining (Ravaut et al., 2024).

3.3 Corpus Statistics and Analysis

Dataset statistics are presented in Table 1 (further statistics and plots in Appendix K). Papers typically include details and analyses regarding experiments; patents usually contain more information on applications and practical benefits. We analyze the lexical overlap between the respective documents and find that only 8.3% of the 4-grams are shared. This highlights the complexity of the task: the two documents describe the same invention from different perspectives, using different linguistic styles.

3.4 Proposed Evaluation Metrics

We adapt a suite of metrics to analyze the performance of patent generation models from multiple perspectives. The length and specialized domain of patent documents make automatic evaluation challenging. We thus propose to disentangle factual content overlap from stylistic similarity using established metrics that are well-suited for long documents.

3.4.1 Content-level metrics

To assess the factual consistency and coverage of the generated patent, we measure semantic overlap with the reference patent and the provided paper. The NLI-based metric SCALE (Lattimer et al., 2023) estimates factual consistency between two documents by computing pairwise entailment scores between *premise chunks* and *hypothesis sentences*. While premise chunks can be large, SCALE still requires a quadratic computation. To make this feasible on long documents, we sample 10 sentences from every hypothesis document. For each sentence, we retrieve the 5 most relevant premise chunks according to BM25 and then compute the maximum NLI score across these chunks. For system-level scores, we average the scores obtained for all sampled sentences. For PAP2PAT, we propose two variants of SCALE:

Factuality. *Ref* $\rightarrow$ *Gen* and *Ref+Pap* $\rightarrow$ *Gen* quantify to what extent the semantic content of the generated patent (*Gen*) is supported by the reference patent (*Ref*), and by the reference patent and the paper (*Ref+Pap*), respectively.

Coverage. To measure the degree to which the generated patent covers the information content of the reference patent, we use the generated patent as premise document and the reference patent as the hypothesis (*Gen* $\rightarrow$ *Ref*).

We confirm the applicability of these scores to our benchmark by computing a few trivial baselines (see Table 3): taking the reference patent itself as the hypothesis document results in an upper bound of around 89%; using the most similar patent from the train set according to BM25 similarity to the paper results in a score of less than 30%. Taken together with the low standard deviations across 5 executions, this demonstrates that the score is able

to differentiate and that the sample size is sufficient to make meaningful system-level comparisons.

3.4.2 Language-level metrics

Patents are written in special **style**, including the use of partially legally relevant phrases and constructions. To estimate to what extent the generated patents adhere to this style, we compare corpus-wide n-gram profiles, an established method for authorship attribution (Keselj et al., 2003; Zecevic, 2011; van Dam, 2013; Mikros and Perifanos, 2013). Here, we use the set of all reference patents from the same split as basis to extract the profile of a typical patent attorney P^n_{ref} and compare it to the profile extracted from the generated patents P^n_{gen}. Each profile contains the 1000 most frequent n-grams for $n \in \{1, 2, 3, 4\}$. To compute the similarity between P^n_{ref} and P^n_{gen}, we adopt the formula from Keselj et al. (2003):

$$sim_n = \sum_{g \in P^n_{\text{ref}} \cup P^n_{\text{gen}}} 1 - \left(\frac{P^n_{\text{ref}}(g) - P^n_{\text{gen}}(g)}{P^n_{\text{ref}}(g) + P^n_{\text{gen}}(g)} \right)^2$$

We additionally integrate a profile from StyloMetrix[4] (Okulska et al., 2023), which implements rule-based detection of 196 linguistic features, including e.g. verb tenses, modal verbs, POS tags, lexical items, figures of speech, and linguistic constructions such as fronting or similes. We compute the final score as the average of the four n-gram profile similarities and the StyloMetrix profile similarity, i.e., $Style = \frac{1}{5}((\sum_{i=1}^{4} sim_n) + sim_{stylo})$.

To measure the **repetitiveness**, we compute the *repetition rate* RR (Cettolo et al., 2014), i.e., the fraction of n-grams that appear more than once. We compute RR over sliding windows of 256 tokens and average the scores. To differentiate between repetitive language and unintended infinite repetitions, we additionally report the fraction of windows with an RR score greater than 80 (RR>80).

4 COPGEN: Chunk-based Outline-guided Patent Generation

Since LLMs cannot yet generate sufficiently long documents in a single call, we instead choose to generate patents in chunks. Figure 2 summarizes the approach. We chunk the outline, select the most relevant parts of the paper (*paper context*) depending on the chunks' outline bullet points, and prompt an LLM to generate the patent text for that chunk. The outlines of previous chunks are included for global document context. Finally, we concatenate the generated outputs and apply only a lightweight post-processing to remove duplicate headings at chunk boundaries. This framework enables generating a long document in parallel, with customized context per chunk and controllable length.

Token Allocation and Chunking. In the default setting, we allocate per chunk 2k tokens for the instruction, 3k tokens for the paper context, and 2k tokens for the output patent ($\mathcal{T}\{$*inst=2k; pap=3k; pat=2k*$\}$). We choose this setting because preliminary experiments show that on our task, LLMs only generate up to roughly 2-3k tokens regardless of the requested amount of content, and because this allows comparing with models that support up to 8k tokens. The chunking procedure segments the outline into chunks depending on the token allocation. In particular, it determines the number of outline bullet points per chunk using the reserved number of patent tokens and the average number of characters per outline bullet point[5]. It keeps sections intact whenever possible.

Length Control Mechanism. Bai et al. (2024a) find that LLMs' response lengths are constrained within a certain range and only moderately adjust to length requests in the prompt, making instructions unreliable for controlling length. However, as the requested length decreases, the LLM is more likely to meet it. Hence, in COPGEN, we decrease the number of allocated patent tokens per chunk, thereby increasing the number of chunks, until the desired total length is reached. In our experimental setting, we do this on a corpus-level, i.e., we search for a setting where the average length matches that of the references, leading to the calibrated token allocation $\mathcal{T}\{$*inst=2k; pap=3k; pat=400*$\}$.

Paper Context Selection. For each chunk i, the retriever selects relevant paragraphs from the paper to create the paper context c_i. We use BM25 (Robertson and Zaragoza, 2009) with the chunk's outline o_i as query. We always include the abstract of the paper and all headings, adding paper paragraphs successively in the order of their relevance ranking until the token limit is reached.

Patent Generation. The paper context c_i, the current outline o_i and prior outlines $\text{o}_{j<i}$ are combined to a prompt (see Appendix J). The LLM generates patent chunk p_i, using constrained decoding to ad-

[4] https://github.com/ZILiAT-NASK/StyloMetrix

[5] We translate between number of tokens and number of characters using average corpus statistics of patents in PAP2PAT.

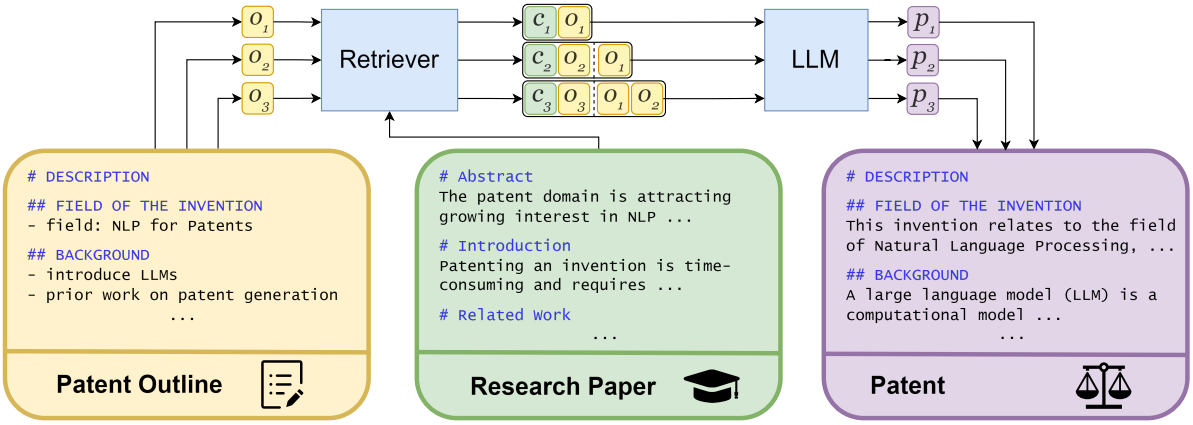

Figure 2: Overview of chunk-based outline-guided patent generation (COPGEN). We chunk the outline, retrieve the parts of the paper that are most relevant for that chunk, prompt the LLM, and concatenate the results. The desired output length and the number of allocated patent tokens per chunk determine the number of chunks.

here to the outline's headings.

5 Experiments

In this section, we describe our experiments using PAP2PAT to assess the capabilities and limitations of current LLMs with COPGEN.

5.1 Evaluation Metrics

We primarily focus on the evaluation metrics proposed in Section 3.4. For the sake of completeness, we also report ROUGE-L F1 (Lin, 2004) and BERTScore F1 (Zhang et al., 2020). For BERTScore, we circumvent the limit of 512 tokens with a sliding window, and use SciBERT (Beltagy et al., 2019), which has been shown to be effective in the patent domain (Pujari et al., 2021). We also report DiscoScore (Zhao et al., 2023), a state-of-the-art **coherence** metric rooted in Centering Theory (Grosz et al., 1995) and based on BERT (Devlin et al., 2019). We use the DS-SENT-NN variant since the authors report that it performs best on long texts. We extend the published implementation[6] with a sliding window to obtain BERT embeddings.

5.2 Models

Choice of LLMs. To make our work reproducible, we only leverage recently published open-weight LLMs with state-of-the-art results on other generation tasks. We include Llama-3 8B and 70B (Dubey et al., 2024), Mixtral-8x7B (Jiang et al., 2024), and Qwen2-72B (Yang et al., 2024).

[6] https://github.com/AIPHES/DiscoScore

Baselines. We include four baselines, upper bounds, and sanity checks for the evaluation setup. In particular, we consider the paper, the outline, the reference patent and the most similar patent from the train set (highest BM25 similarity to the paper) as the generated document. Mixtral-8x7B and Qwen2-72B support 32k tokens, thus we use them to determine the performance without COPGEN by prompting them with the complete paper and outline (**Single LLM-call**).

Fine-tuning. We fine-tune Llama-3 8B on the training split of PAP2PAT using LoRA (Hu et al., 2022). We adopt the hyperparameters proposed by Tribes et al. (2024), see Appendix G. Since the fine-tuned model frequently generates infinite repetitions, we design a post-processing procedure that detects and removes them (see Appendix C).

5.3 Main Findings

Table 3 shows our main evaluation results. Overall, we find that **COPGEN** strongly improves upon using a **Single LLM-call**, in which the generated patents are much too short (on average less than one fifth of the reference patents). COPGEN provides an effective remedy, creating much longer and length-controllable outputs with higher coverage (correlating with higher text similarity scores). In terms of language style, COPGEN also has a clear advantage. For RR, values around 12-14% seem to be ideal according to the scores of the reference and similar patents.

While the fine-tuned models produce more repetitions, both the Single LLM-call and zero-shot COPGEN have similar repetition rates and only the smaller and fine-tuned Llama models get stuck in

	Tokens	Text Sim ↑		Content-Level Metrics (SCALE) ↑			Language ↑		Repetitions	
				Coverage	Factuality					
		BS	R-L	*Gen→Ref*	*Ref→Gen*	*Ref+Pap→Gen*	Style	DS	RR	RR>80
Heuristic Baselines / Skylines										
Reference Patent	*18.1k (100%)*	*100*	*100*	*88.6 ± .15*	*88.5 ± .18*	*88.7 ± .18*	*100*	*100*	14.4	0.2
Similar Patent	30.1k (166.0%)	65.8	33.7	29.4 ± .37	27.6 ± .32	27.6 ± .36	75.3	98.3	12.3	0.1
Outline	1.4k (7.9%)	56.5	10.1	39.7 ± .72	61.6 ± .14	61.9 ± .15	24.4	85.6	20.1	0.2
Paper	8.1k (44.5%)	69.7	39.4	44.8 ± .61	46.5 ± .40	*89.0 ± .13*	47.2	98.4	8.4	0.0
Single LLM-call ($\mathcal{T}${*inst=2k; pap=∞; pat=∞*})										
Mixtral-8x7B	3.2k (17.7%)	66.1	23.1	38.1 ± .70	**66.9** ± .87	72.0 ± .71	***42.6***	96.9	17.9	0.6
Qwen2-72B	2.8k (15.6%)	***66.6***	21.3	***40.3*** ± .45	65.8 ± .47	72.4 ± .43	39.6	97.3	9.5	0.5
w/o Paper	3.5k (19.3%)	66.1	***23.2***	38.9 ± .29	65.9 ± .54	66.6 ± .38	37.1	96.6	9.8	0.6
w/o Outline	2.0k (11.1%)	64.2	16.9	34.9 ± .40	56.7 ± .31	**75.3** ± .23	39.8	***97.4***	9.3	0.1
COPGEN ($\mathcal{T}${*inst=2k; pap=3k; pat=2k*})										
Llama-3 8B	9.6k (53.1%)	68.7	41.4	40.3 ± .21	60.8 ± .49	65.7 ± .55	43.2	97.0	27.0	4.4
Llama-3 8B SFT	27.5k (151.5%)	***70.4***	42.8	42.0 ± .25	49.3 ± .39	52.1 ± .39	59.4	98.0	53.7	29.3
w/ rep. removal	17.3k (95.4%)	71.2	***49.6***	42.1 ± .48	49.6 ± .38	53.4 ± .59	**64.7**	**98.5**	38.4	8.5
Mixtral-8x7B	5.6k (31.0%)	69.1	35.5	41.8 ± .61	62.3 ± .31	68.5 ± .30	49.3	97.5	14.5	0.4
Llama-3 70B	6.1k (33.9%)	70.2	39.0	42.7 ± .60	***64.5*** ± .47	***68.6*** ± .58	49.5	97.4	17.5	0.2
Qwen2-72B	8.1k (44.8%)	70.2	41.3	***44.1*** ± .32	62.5 ± .44	67.9 ± .35	47.5	97.3	12.8	1.2
COPGEN ($\mathcal{T}${*inst=2k; pap=3k; pat=400*})										
Qwen2-72B	18.1k (100%)	**71.7**	**50.8**	**46.8** ± .31	59.7 ± .63	65.3 ± .44	47.8	97.1	10.0	0.5

Table 3: Experimental Results. *Italic values* represent upper bounds that used test data in the prediction. The best value per column is **bold**, the best per section ***bold italics***. Tokens are reported as absolute and relative to the reference. BS = BERTScore. R-L = ROUGE-L. DS = DiscoScore. Text Sim = Standard Text Similarity Metrics.

infinite repetitions. All models achieve very high coherence scores, indicating that all LLMs in the study do not suffer from generating incoherent text.

Impact of Outline and Paper as Input. Ablating the paper or the outline from the prompt in the Single LLM-call baseline leads to lower coverage. When ablating the outline, the output drops markedly in length and sticks closely to the paper (as shown by the high score for *Ref+Pap→Gen*), but without producing the content desired for the patent (*Ref→Gen* drops by almost 10pp.). COPGEN is able to leverage the outline and paper context effectively, as demonstrated by its much higher coverage.

Length Control and Coverage-Factuality Trade-off. Figure 3 shows that allocating fewer tokens per chunk for the patent, i.e., generating more chunks, leads to a near-linear increase in output length. Our length control mechanism is able to find an optimal setting in which the average output length corresponds to that of the reference patents (last row of Table 3), resulting in the overall best text similarity and coverage scores, while keeping factuality high (as opposed to the fine-tuned models that produce similar length). Coverage and factuality are intuitively antagonistic (see Figure 3), similar to precision and recall: as the generated text becomes longer, it becomes increasingly difficult to maintain factuality but easier to achieve higher coverage. This also explains why the Single LLM-call baselines achieve higher factuality scores than COPGEN runs that generate more than five times as much text.

5.4 Analysis of COPGEN Settings

We now study various settings of COPGEN.

Impact of Outline Granularity. In Figure 4, we observe a consistent improvement across coverage, factuality and BERTScore with more detailed outlines: users can directly improve output quality by providing more details. Notably, contrary to our experiments on length control, the improvements in coverage are achieved without longer outputs and without decreasing factuality.

Impact of Selection of Paper Context. We study how providing informative context taken from the paper influences results. In Figure 5, we show retriever ablation results including two baselines: *NoPaper* does not add any context from the paper, and *AbstractOnly* uses only the abstract. As an upper bound, we use *BM25Oracle,* where the BM25 query is the original patent text instead of the outline. We observe a monotonically increasing performance, demonstrating that associated papers

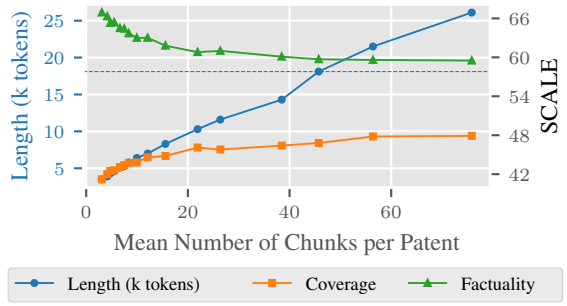

Figure 3: Controlling output length. Each point on the x-axis represents a run of Qwen2-72B with varying token allocation $\mathcal{T}\{inst{=}2k;\ pap{=}3k;\ pat{=}l_{pat}\}$ where l_{pat} ranges from 200 to 10,000. A lower l_{pat} results in more chunks. The dashed blue line represents the average reference patent length.

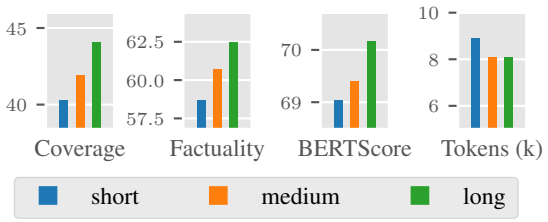

Figure 4: Outline granularity ablation of Qwen2-72B with default token allocation.

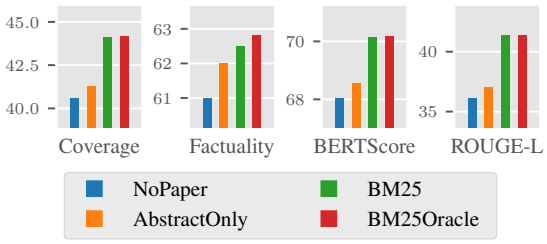

Figure 5: Retriever ablation. The plot depicts the results of Qwen2-72B using the long outline and default token allocation.

provide valuable information. There is only a small gap between *BM25* and *BM25Oracle*, suggesting that the outline is an effective BM25 query and that more elaborate retrieval methods could close the gap further.

Test Data Contamination. If patents are (partially) memorized during pre-training, one would expect a sudden drop in performance when evaluating on patents published after the pre-training cutoff date, i.e., on the non-contaminated test set. We see only minor differences in text similarity and SCALE metrics, though the style scores drop significantly (see Appendix D). This is likely due to domain distribution shifts (see Appendix K). Overall, the results do not indicate systematic issues with memorization.

Fine-tuning. Fine-tuning results in significantly longer outputs, which is consistent with the findings of Bai et al. (2024a). However, this increased length is partially due to infinite repetitions, an issue also observed in concurrent work on patent generation by Wang et al. (2024c). Prior work has identified repetitive training data as a key factor contributing to infinite repetitions (Li et al., 2023a). We hence hypothesize that the cause is patents' inherently repetitive style. Patents often present numerous variations of the invention, each introduced with similar phrasing and detailing different combinations of components. This also results in a 71% higher RR than for papers. Despite these repetitions, fine-tuning improves standard text similarity metrics like ROUGE-L and BERTScore. Wang et al. (2024c) find the same, along with much lower scores in their human evaluation, concluding that the repetitions lead to over-rewarding in n-gram-based metrics. We find contrary evidence that removing many repetitions from the generated patents in post-processing actually further improves ROUGE-L by 6.8pp. Our evaluation instead reveals that the improvement achieved by fine-tuning is likely mainly due to stylistic similarity: style scores increase by over 20pp. while factuality metrics decrease by over 10pp.

6 Human Evaluation

To gain further insight into the practical applicability beyond the automatic metrics, we conduct a human case study with two patent attorneys for the field of AI. We randomly select 10 samples from the NLP field out of the PAP2PAT test set and use the outputs of Qwen2-72B with COPGEN and default token allocation. For each of the samples, the attorneys are provided with the paper, the outline, the generated patent and the reference patent. They are then tasked to evaluate the quality of the generated patent based on the hypothetical scenario that they got the paper, wrote the outline and sent both to the LLM. We ask about strengths, weaknesses and potential for time savings. The attorneys evaluated 10 and 5 samples, respectively, yielding 15 total evaluations and spending about 30 minutes per sample on average.

Among the 15 evaluations, the attorneys saw substantial time savings in 8 cases, showing promising

potential to increase their productivity. However, they also identify two main limitations that could hinder such time savings, which are consistent with observations from our automatic evaluation, further validating our chosen metrics. These limitations are:

(1) **Style**: The model often fails to use non-limiting language, which is essential in the patent description to avoid unnecessary limitations. For instance, stating that *"the system includes a crucial component"* may be too limiting, whereas rephrasing it as *"the system may include a preferred component"* provides more flexibility to adapt claims in the grant procedure and does not narrow the claims' interpreted scope of protection. Furthermore, the model occasionally uses promotional phrases like *"have revolutionized"* that are common in NLP papers but inappropriate in patents.

(2) **Level of detail**: The patent description should describe the invention precisely enough to enable an ordinary person skilled in the art to reproduce it. This level of detail is often still not generated, which is also in line with the lower absolute scores for coverage compared to those for factuality.

7 Conclusion

In this work, we have presented the first study on the generation of complete patent descriptions in a realistic experimental setting, using papers as real-world invention specifications. We create the PAP2PAT benchmark, propose targeted evaluation metrics, develop COPGEN to enable the generation of long patent documents, and conduct a human evaluation. We find that LLMs can effectively generate patent drafts from research papers, but that there is still headroom for improving the level of detail and linguistic style. Promising directions for future research include developing fine-tuning methods that maintain factuality and avoid repetitions, as well as conducting user studies to identify efficient interaction patterns.

Limitations

This study focuses on open-weight models, leaving the exploration of closed-source commercial models as a potential avenue for future research. Expanding our experimental setting to include these models could provide additional insights.

In this work, we do not study the automatic generation of outlines, but consider them user input. Generating them in a planning stage from the paper or other input could help decrease manual effort further.

Due to the immense cost of patent attorneys, our expert case study analyses only a small sample of the dataset. Conducting large-scale human evaluations could generate further insights.

Acknowledgments

We would like to thank the patent attorneys Philipp Mangold and Charlotte Hellmann for evaluating the quality of generated patents, and for providing valuable insights into the peculiarities of patents.

Ethics Statement

The development of AI-powered patent generation tools may raise ethical concerns. We emphasize that our research is intended to support and augment the work of patent professionals, rather than to fully automate the patenting process or facilitate malicious activities such as patent trolling.

References

Yushi Bai, Jiajie Zhang, Xin Lv, Linzhi Zheng, Siqi Zhu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024a. LongWriter: Unleashing 10,000+ Word Generation from Long Context LLMs.

Zilong Bai et al. 2024b. PatentGPT: A Large Language Model for Intellectual Property.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Silvia Casola and Alberto Lavelli. 2022. Summarization, simplification, and generation: The case of patents. Expert Systems with Applications, 205:117627.

Mauro Cettolo, Nicola Bertoldi, and Marcello Federico. 2014. The repetition rate of text as a predictor of the effectiveness of machine translation adaptation. In Proceedings of the 11th Conference of the Association for Machine Translation in the Americas: MT Researchers Track, pages 166–179, Vancouver, Canada. Association for Machine Translation in the Americas.

Seokkyu Choi, Hyeonju Lee, Eunjeong Park, and Sungchul Choi. 2022. Deep learning for patent

landscaping using transformer and graph embedding. Technological Forecasting and Social Change, 175:121413.

Dimitrios Christofidellis, Antonio Berrios Torres, A. Dave, M. Roveri, Kristin Schmidt, Sarath Swaminathan, Hans Vandierendonck, D. Zubarev, and Matteo Manica. 2022. PGT: A prompt based generative transformer for the patent domain. In International Conference on Machine Learning.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mehdi Drissi, Olivia Watkins, and Jugal Kalita. 2018. Hierarchical text generation using an outline. In 15th International Conference on Natural Language Processing.

Abhimanyu Dubey et al. 2024. The Llama 3 Herd of Models.

Le Fang, Tao Zeng, Chao-Ning Liu, Liefeng Bo, Wen Dong, and Changyou Chen. 2021. Outline to Story: Fine-grained Controllable Story Generation from Cascaded Events. ArXiv.

Michael Färber, David Lamprecht, Johan Krause, Linn Aung, and Peter Haase. 2023. SemOpenAlex: The Scientific Landscape in 26 Billion RDF Triples. In The Semantic Web – ISWC 2023, pages 94–112, Cham. Springer Nature Switzerland.

Joshua S. Gans, Fiona E. Murray, and Scott Stern. 2017. Contracting over the disclosure of scientific knowledge: Intellectual property and academic publication. Research Policy, 46(4):820–835.

Seraphina Goldfarb-Tarrant, Haining Feng, and Nanyun Peng. 2019. Plan, write, and revise: An interactive system for open-domain story generation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), pages 89–97, Minneapolis, Minnesota. Association for Computational Linguistics.

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. Computational Linguistics, 21(2):203–225.

Felix Hamborg, Moustafa Elmaghraby, Corinna Breitinger, and Bela Gipp. 2017. Automated generation of timestamped patent abstracts at scale to outsmart patent-trolls. In Proceedings of the 2nd Joint Workshop on Bibliometric-Enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2017) : Tokyo : Japan, number 1888 in CEUR Workshop Proceedings, pages 101–106. SunSITE Central Europe.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In International Conference on Learning Representations.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of Experts.

Lekang Jiang and Stephan Goetz. 2024. Artificial Intelligence Exploring the Patent Field.

Vlado Keselj, Fuchun Peng, Nick Cercone, and Calvin Thomas. 2003. N-GRAM-BASED AUTHOR PROFILES FOR AUTHORSHIP ATTRIBUTION. In Proceedings of the Conference Pacific Association for Computational Linguistics.

Antoinette F. Konski and Linda X. Wu. 2015. Inventorship and Authorship. Cold Spring Harbor Perspectives in Medicine, 5(11):a020859.

Barrett Lattimer, Patrick CHen, Xinyuan Zhang, and Yi Yang. 2023. Fast and Accurate Factual Inconsistency Detection Over Long Documents. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 1691–1703, Singapore. Association for Computational Linguistics.

Jieh-Sheng Lee. 2020. Controlling patent text generation by structural metadata. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pages 3241–3244.

Jieh-Sheng Lee. 2023. Evaluating generative patent language models. World Patent Information, 72:102173.

Jieh-Sheng Lee. 2024. InstructPatentGPT: Training patent language models to follow instructions with human feedback. Artificial Intelligence and Law.

Yukyung Lee, Soonwon Ka, Bokyung Son, Pilsung Kang, and Jaewook Kang. 2024. Navigating the Path of Writing: Outline-guided Text Generation with Large Language Models.

Huayang Li, Tian Lan, Zihao Fu, Deng Cai, Lemao Liu, Nigel Collier, Taro Watanabe, and Yixuan Su. 2023a. Repetition in repetition out: Towards understanding neural text degeneration from the data perspective. In Thirty-seventh Conference on Neural Information Processing Systems.

Yunzhe Li, Qian Chen, Weixiang Yan, Wen Wang, Qinglin Zhang, and Hari Sundaram. 2023b. Advancing Precise Outline-Conditioned Text Generation with Task Duality and Explicit Outline Control. In Conference of the European Chapter of the Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Xiang Liu, Peijie Dong, Xuming Hu, and Xiaowen Chu. 2024. LongGenBench: Long-context Generation Benchmark. In Findings of the Association for Computational Linguistics: EMNLP 2024, pages 865–883, Miami, Florida, USA. Association for Computational Linguistics.

T Magerman, Bart van Looy, and K Debackere. 2011. In search of anti-commons: patent paper pairs in biotechnology. an analysis of citation flows. In European Network of Indicator Designers (ENID) STI Indicators Conference.

Tom Magerman, Bart Van Looy, and Koenraad Debackere. 2015. Does involvement in patenting jeopardize one's academic footprint? An analysis of patent-paper pairs in biotechnology. Research Policy, 44(9):1702–1713.

Tom Magerman, Bart Van Looy, and Xiaoyan Song. 2010. Exploring the feasibility and accuracy of Latent Semantic Analysis based text mining techniques to detect similarity between patent documents and scientific publications. Scientometrics, 82(2):289–306.

Matt Marx and Aaron Fuegi. 2020. Reliance on science: Worldwide front-page patent citations to scientific articles. Strategic Management Journal, 41(9):1572–1594.

Matt Marx and Aaron Fuegi. 2022. Reliance on science by inventors: Hybrid extraction of in-text patent-to-article citations. Journal of Economics & Management Strategy, 31(2):369–392.

George K. Mikros and Kostas Perifanos. 2013. Authorship attribution in greek tweets using author's multilevel n-gram profiles. In AAAI Spring Symposium: Analyzing Microtext.

Fiona Murray. 2002. Innovation as co-evolution of scientific and technological networks: Exploring tissue engineering. Research Policy, 31(8):1389–1403.

Fiona Murray and Scott Stern. 2005. Do Formal Intellectual Property Rights Hinder the Free Flow of Scientific Knowledge? An Empirical Test of the Anti-Commons Hypothesis. w11465, page w11465, Cambridge, MA. National Bureau of Economic Research.

Inez Okulska, Daria Stetsenko, Anna Kołos, Agnieszka Karlińska, Kinga Głąbińska, and Adam Nowakowski. 2023. StyloMetrix: An Open-Source Multilingual Tool for Representing Stylometric Vectors.

Subhash Pujari, Jannik Strötgen, Mark Giereth, Michael Gertz, and Annemarie Friedrich. 2022. Three Real-World Datasets and Neural Computational Models for Classification Tasks in Patent Landscaping. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 11498–11513, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Subhash Chandra Pujari, Annemarie Friedrich, and Jannik Strötgen. 2021. A Multi-task Approach to Neural Multi-label Hierarchical Patent Classification Using Transformers. In Advances in Information Retrieval, pages 513–528, Cham. Springer International Publishing.

Haoran Que, Feiyu Duan, Liqun He, Yutao Mou, Wangchunshu Zhou, Jiaheng Liu, Wenge Rong, Zekun Moore Wang, Jian Yang, Ge Zhang, Junran Peng, Zhaoxiang Zhang, Songyang Zhang, and Kai Chen. 2024. HelloBench: Evaluating Long Text Generation Capabilities of Large Language Models.

Mathieu Ravaut, Bosheng Ding, Fangkai Jiao, Hailin Chen, Xingxuan Li, Ruochen Zhao, Chengwei Qin, Caiming Xiong, and Shafiq Joty. 2024. How much are llms contaminated? a comprehensive survey and the llmsanitize library. arXiv preprint arXiv:2404.00699.

Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends® in Information Retrieval, 3(4):333–389.

Walid Shalaby and Wlodek Zadrozny. 2019. Patent retrieval: A literature review. Knowledge and Information Systems, 61(2):631–660.

Yijia Shao, Yucheng Jiang, Theodore Kanell, Peter Xu, Omar Khattab, and Monica Lam. 2024. Assisting in Writing Wikipedia-like Articles From Scratch with Large Language Models. 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 6252–6278, Mexico City, Mexico. Association for Computational Linguistics.

Homaira Huda Shomee, Zhu Wang, Sathya N. Ravi, and Sourav Medya. 2024. A Comprehensive Survey on AI-based Methods for Patents.

Alexander Spangher, Yao Ming, Xinyu Hua, and Nanyun Peng. 2022. Sequentially controlled text generation. In Findings of the Association for Computational Linguistics: EMNLP 2022, pages 6848–6866, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Vasileios Stamatis. 2022. End to End Neural Retrieval for Patent Prior Art Search. In Advances in Information Retrieval, volume 13186, pages 537–544, Cham. Springer International Publishing.

Xiaofei Sun, Zijun Sun, Yuxian Meng, Jiwei Li, and Chun Fan. 2022. Summarize, Outline, and Elaborate: Long-Text Generation via Hierarchical Supervision from Extractive Summaries. In Proceedings of the 29th International Conference on Computational Linguistics, pages 6392–6402, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Christophe Tribes, Sacha Benarroch-Lelong, Peng Lu, and Ivan Kobyzev. 2024. Hyperparameter Optimization for Large Language Model Instruction-Tuning.

Michiel van Dam. 2013. A basic character n-gram approach to authorship verification notebook for pan at clef 2013. In Conference and Labs of the Evaluation Forum.

Bart Van Looy, Bart Baesens, Tom Magerman, and Koenraad Debackere. 2011. Assessment of Latent Semantic Analysis (LSA) Text Mining Algorithms for Large Scale Mapping of Patent and Scientific Publication Documents. SSRN Electronic Journal.

Juanyan Wang, Sai Krishna Reddy Mudhiganti, and Manali Sharma. 2024a. 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.

Qiyao Wang, Jianguo Huang, Shule Lu, Yuan Lin, Kan Xu, Liang Yang, and Hongfei Lin. 2024b. IPEval: A Bilingual Intellectual Property Agency Consultation Evaluation Benchmark for Large Language Models.

Qiyao Wang, Shiwen Ni, Huaren Liu, Shule Lu, Guhong Chen, Xi Feng, Chi Wei, Qiang Qu, Hamid Alinejad-Rokny, Yuan Lin, and Min Yang. 2024c. AutoPatent: A Multi-Agent Framework for Automatic Patent Generation.

Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Min zhang, Qingsong Wen, Wei Ye, Shikun Zhang, and Yue Zhang. 2024d. Autosurvey: Large language models can automatically write surveys. In Advances in Neural Information Processing Systems, volume 37, pages 115119–115145. Curran Associates, Inc.

Yuhao Wu, Ming Shan Hee, Zhiqing Hu, and Roy Ka-Wei Lee. 2025. LongGenBench: Benchmarking Long-Form Generation in Long Context LLMs.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 Technical Report.

Kevin Yang, Dan Klein, Nanyun Peng, and Yuandong Tian. 2023. DOC: Improving Long Story Coherence With Detailed Outline Control. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3378–3465, Toronto, Canada. Association for Computational Linguistics.

Kevin Yang, Yuandong Tian, Nanyun Peng, and Dan Klein. 2022. Re3: Generating longer stories with recursive reprompting and revision. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 4393–4479, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-Write: Towards Better Automatic Storytelling. Proceedings of the AAAI Conference on Artificial Intelligence, 33(01):7378–7385.

Xi Ye, Fangcong Yin, Yinghui He, Joie Zhang, Howard Yen, Tianyu Gao, Greg Durrett, and Danqi Chen. 2025. LongProc: Benchmarking Long-Context Language Models on Long Procedural Generation.

Andelka Zecevic. 2011. N-gram based text classification according to authorship. In Recent Advances in Natural Language Processing.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In International Conference on Learning Representations.

Wei Zhao, Michael Strube, and Steffen Eger. 2023. DiscoScore: Evaluating text generation with BERT and discourse coherence. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 3865–3883, Dubrovnik, Croatia. Association for Computational Linguistics.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Jeff Huang, Chuyue Sun, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. 2023. Efficiently Programming Large Language Models using SGLang.

You Zuo, Kim Gerdes, Eric Villemonte de La Clergerie, and Benoît Sagot. 2024. PatentEval: Understanding

Errors in Patent Generation. In NAACL2024 - 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Mexico City, Mexico.

A PAP2PAT Dataset

We present the PAP2PAT dataset containing 1.8k PPPs from a variety of domains, each annotated with multiple outlines. It serves two main purposes. First, it is a challenging benchmark for LLMs that requires long-text generation capabilities and deep understanding of the technical domain as well as patent law. Second, it facilitates the development of AI-powered tools for patent drafting, where the patent attorney only performs post-editing rather than writing from scratch, potentially incurring massive cost savings. In the following, we describe the steps taken for constructing the PAP2PAT dataset: scraping and filtering PPPs, parsing the full-text documents and generating the patent outlines.

A.1 Scraping Patent-Paper Pairs

The patent and paper in a PPP typically do not cite each other, so we cannot rely on front-page or in-text citations (Marx and Fuegi, 2020, 2022) to find PPPs. The matching must rely on document metadata and content alone. We start out with the USPTO dataset[7] containing 6.7M patent applications from 2005 to April 2024. For each patent, we query SemOpenAlex (Färber et al., 2023) using SPARQL and retrieve papers with overlapping authors lists and publication dates. Next, we filter the results based on their titles, abstracts, other candidate matches for the same patent, and paper licenses, as elaborated below. Table 2 shows the remaining number of candidates after each filtering step. As can be seen, the major limiting factor for our benchmark size is the issue of scientific articles being published under restrictive licenses. An example match is shown in Table 4. We perform a systematic manual post-hoc evaluation of the heuristics to validate their precision (see Section A.2).

Author Overlap. The patent and the paper of a PPP are by definition authored by overlapping sets of individuals. The overlap of author lists have therefore been identified as an effective (yet not sufficient) criteria for matching PPPs (Magerman et al., 2010). The requirements for paper authorship are typically much lower than those for patent inventorship (Konski and Wu, 2015). In many cases, only the main author(s) and the senior author(s) are listed as inventors. We accordingly employ an asymmetric score that only measures the fraction of inventors $i \in I$ that are also authors $a \in A$, not vice versa:

$$\text{sim}_{\text{author}} = \frac{|I \cap A|}{|I|} \geq 0.8$$

This score's effectiveness increases with the number of inventors, so we only consider patents with at least two inventors. Implementing $\text{sim}_{\text{author}}$ requires some form of author name disambiguation to avoid false negatives (different spellings, e.g., with, without, or with abbreviated middle name) and false positives (e.g., very common names like John Smith). There are no author identifiers shared between the patent and paper datasets, so our disambiguation uses the surface names only. To account for false negatives, we use the aliases stored in SemOpenAlex and consider an inventor to be an author if their name matches exactly with one of the aliases. False positives are marginalized by author combinations and subsequent filters: it is highly unlikely that there exist two groups of people with the same set of names working on the same topic at the same time.

Date Range. We require the paper's publication date to be within one year before and two years after the patent application date. The former corresponds to the USPTO's grace period,[8] which allows inventors to file patent applications up to one year after they disclosed the invention to the public. The two-year period after the application date was selected because qualitative analyses identified it as the point of diminishing returns, beyond which the incidence of true positives notably decreases, while the rate of false positives significantly increases.

Term Overlap. We compare titles and abstracts of patents and papers using term overlap metrics. We first obtain a set of terms T using removal of stopwords and punctuation[9] and stemming.[10] The score is then computed as the number of shared terms, normalized by the minimum or maximum number of terms, following Magerman et al. (2015). We additionally weight each term using its IDF

[7] https://bulkdata.uspto.gov

[8] https://www.uspto.gov/web/offices/pac/mpep/s2153.html

[9] based on spaCy's `en_core_web_sm`

[10] based on NLTK's PorterStemmer

Field		Patent	Paper Candidate 1 (✓)	Paper Candidate 2 (✗)
Authors	Content	Ge Wang, Wenxiang Cong	Wenxiang Cong, Yan Xi, Bruno De Man, Ge Wang	Wenxiang Cong, Yan Xi, Peter Fitzgerald, Bruno De Man, Ge Wang
	$\text{sim}_{\text{author}}$		1.0	1.0
Title	Content	Monochromatic CT Image Reconstruction from Current-Integrating Data via Machine Learning	Monochromatic image reconstruction via machine learning	Virtual Monoenergetic CT Imaging via Deep Learning
	sim_{term}		1.0 / 0.63	0.39 / 0.32
Abstract	Content	A machine-learning-based monochromatic CT image reconstruction method is described ...	X-ray computed tomography (CT) is a nondestructive imaging ...	The physical process of X-ray CT imaging is described ...
	sim_{term}		0.71 / 0.19	0.50 / 0.23
Date		2021-08-30	2020-11-01 (✓)	2021-04-14 (✓)

Table 4: Example for the matching of a PPP. The scores correspond to the respective metrics, the term metrics are shown as $\min$-normalized / $\max$-normalized. Both papers have author lists that contain all the inventors, were published inside the date range, and have titles and abstracts with sufficiently high absolute term similarity to the patent. Since the term similarity scores of paper 1 are higher than those of paper 2 by the specified margin (see Distinctiveness filtering step), paper 1 is correctly matched to the patent.

value across all titles and abstracts of patents and papers.

$$\text{sim}_{\text{term}}(pat, pap) = \frac{\sum_{t \in T(pat) \cap T(pap)} \text{idf}(t)}{\text{agg}\Big(\sum_{t \in T(pat)} \text{idf}(t), \sum_{t \in T(pap)} \text{idf}(t)\Big)}$$

where pat and pap are either the titles or the abstracts of the documents, and $\text{agg} \in \{\min, \max\}$. Thus, we have 4 scores in total, for which we set the thresholds to be 0.15 with $\min$ normalization and 0.1 with $\max$ normalization. We choose the values based on interactive experimentation, but perform a post-hoc validation of the matching precision.

Distinctiveness. In the remaining candidate pairs, there are still many cases where one patent is matched to multiple papers or vice versa. We disambiguate these cases by comparing the term overlap metrics among these ambiguous candidate groups. We only keep a pair if 3 out of 4 term metrics are higher than those of any other candidate in the group by a margin of 0.15 and 0.1, for $\min$ and $\max$, respectively.

License. We filter the matches for licenses that allow redistribution and commercial use, i.e., CC-BY,CC0,and public domain. We use the license information provided by SemOpenAlex and by the ArXiv API.

A.2 Manual Validation

To verify the precision of our matching pipeline, we perform a manual validation, conducted by the first author of this paper. We randomly sample 60 PPPs, read both abstracts, skim the documents, compare the figures and get an overview of the authors' related work. We spend roughly 5 minutes per pair on average. We find that in 55/60 (91.7%) pairs, the paper indeed describes the invention as one of the core contributions. In three pairs, the best match for the patent would have been a prior paper by the same authors. In two pairs, the paper would have been best matched to a related but different patent by the same inventors. This result validates the precision of our matching approach. In the five imperfect matching cases, the papers still provide meaningful training and evaluation signals, as they are still closely related to the invention and contextualized by the outline.

A.3 Document Parsing

We parse all patents and papers into a nested JSON schema where each section has a title, paragraphs and subsections field. We make considerable efforts to obtain clean data: we perform LLM-based section hierarchy reconstruction for patents, font-based section hierarchy reconstruction for paper PDFs and formula conversion for patents and papers. We provide more details in Appendix B.

A.4 Dataset Characteristics

We split our dataset randomly into *train* (n=1000), *test* (n=500) and *validation* (n=242). We additionally create a non-contaminated test set (*nc-test*) that contains all pairs with a patent published in 2024 (n=71), i.e., after the pretraining cut-off date of all evaluated open-weight LLMs. Thus, we address

the concern that LLMs might have seen test data during pretraining (Ravaut et al., 2024). Table 1 shows dataset statistics across the splits. Appendix K shows further statistics and plots, including the distribution over domains and the number of pairs over time.

In general, both patents and papers contain information not present in the other. The paper typically includes more experimental details and insights drawn from the experiments. The patent usually contains more information on the applications and practical benefits of the invention. We analyze the lexical overlaps between the documents and find that only 2.1% of the 4-grams are shared. This underlines the complexity of the task: the two documents describe the same invention from a different perspective using different language. Nevertheless, we find that it is common for attorneys to copy content from the paper to the patent (or vice versa). For instance, many patents and papers share a portion of the figures, as well as verbatim copied text.

B Document Parsing Pipeline

In total, we use three different data sources for the full texts. For the patents, we use the USPTO bulk downloads[11]. For the papers, we use PubMed if available and PDF otherwise. The goal is to extract a clean representation of the full text into a common JSON format. In this format, every section has a title, a list of paragraphs and a list of subsections. We write parsers for the XML formats from USPTO, PubMed and the PDF parser Grobid[12]. In addition, we perform several cleaning steps:

1. **PDF Hierarchy Reconstruction.** The JSON format is hierarchical by nature, for instance to enable better chunking of patents and section-based retrieval from papers. However, Grobid does not detect section levels if the sections are not numbered. To reconstruct the levels in these cases, we implement a solution that searches for the headings in the PDF file, extracts their font properties, orders them by size, boldness and capitalization, and infers the levels from that.

2. **Patent Hierarchy Reconstruction.** In USPTO's patent XML files, there is a level attribute associated with every heading, but we find that it is rarely correct; most headings are placed on the same level. To reconstruct the levels, we pass the list of headings to an LLM and instruct it to infer the levels based on their names (e.g., "Example 1" and "Example 2" and likely children of "EXAMPLES").

3. **Formula Conversion.** Many patents and papers include formulas that can be an important part of the document. However, in USPTO's and PubMed's XML formats, formulas are represented in MathML syntax, which is extremely hard to read and arguably hard to generate. To that end, we convert all MathML formulas to latex using pandoc[13].

4. **Metadata Section Filtering.** Patents usually contain a number of metadata sections in the full text, such as information regarding funding or cross-references to related patents. To filter out these sections, we collect a list of such heading names and remove a section if its heading has a Levenshtein distance less than 3 to any one of the blacklisted headings.

[11] https://bulkdata.uspto.gov/

[12] https://github.com/kermitt2/grobid

[13] https://github.com/jgm/pandoc

C Repetition Removal

Algorithm 1 Repetition Detection

```
function MATCHES(l1, l2)
    n_match ← 0
    for i ← 0 to len(l1) − 1 do
        if l1[i] = l2[i] then
            n_match ← n_match + 1
        end if
    end for
    return n_match / len(l1) > 0.9
end function

function DETECT_REPETITIONS(words, min_length, max_cycle_length)
    n ← len(words)
    for k = 1 to max_cycle_length do
        if matches(words[-k:], words[-2 * k : -k]) then          ▷ Check for repetition of length k
            i ← 2                                                ▷ Number of times the pattern is repeated
            while matches(words[-k:], words[-(i + 1) * k : -i * k]) do   ▷ Search for additional occurrences
                i ← i + 1
            end while
            remove_indices ← [n - (i - 1) * k, ..., n]
            total_length ← i * k
            if total_length ≥ min_length then
                words ← words[: n - (i - 1) * k]                 ▷ Keep only first occurrence of pattern
                remove_rest ← detect_repetitions(words, min_length, max_cycle_length)   ▷ Recursive call for remaining text
                return remove_indices + remove_rest              ▷ Return indices to remove
            end if
        end if
    end for
    return []
end function
```

We design a procedure to remove infinite repetitions from generated outputs to study their effect on evaluation metrics. In that context, we characterize an infinite repetition as a sequence of tokens that appears multiple times until the end of the generation. To find such repetitions, we apply the following recursive algorithm in Algorithm 1 to each chunk and remove the returned indices. To account for reptitions where the model alters the patterns slightly (e.g., incrementing a number) in each iteration, we consider two word sequences equal if 90% of their positions are equal. By default, we use min_length = 50 and max_cycle_length = 300.

D Contamination Results

	Tokens	Text Sim ↑		Content-Level Metrics (SCALE) ↑			Language ↑		Repetitions	
				Coverage	Factuality					
		BS	R-L	*Gen* → *Ref*	*Ref* → *Gen*	*Ref* + *Pap* → *Gen*	Style	DiscoScore	RR	RR>80
Heuristic Baselines / Skylines										
Reference Patent	*18.4k (100.0%)*	*100.0*	*100.0*	*88.6 ± .57*	*88.5 ± .23*	*88.7 ± .21*	*99.8*	*100.0*	15.2	0.1
Similar Patent	26.1k (141.8%)	66.0	35.3	31.4 ± .93	28.0 ± .64	28.3 ± .68	53.9	98.1	12.5	0.1
Outline	1.4k (7.6%)	56.7	10.2	42.1 ± .94	63.4 ± .88	63.8 ± .86	25.9	85.1	19.4	0.0
Paper	9.5k (51.7%)	69.6	42.2	46.7 ± 1.01	46.5 ± .87	*88.8 ± .41*	37.2	98.0	8.2	0.0
Single LLM-call ($\mathcal{T}\{$*inst=2k; pap=∞; pat=∞*$\}$)										
Mixtral-8x7B	3.0k (16.3%)	***66.5***	21.7	40.7 ± 1.16	**67.2** ± 1.19	73.9 ± 1.16	***36.5***	96.8	15.9	0.1
Qwen2-72B	2.9k (15.5%)	66.4	21.2	***42.0*** ± 1.21	65.0 ± .76	71.9 ± .23	34.5	***96.9***	**8.9**	**0.0**
w/o Paper	3.4k (18.2%)	65.9	***21.8***	41.4 ± 1.25	66.3 ± 1.04	67.0 ± .99	34.1	96.1	9.0	0.2
w/o Outline	2.0k (11.0%)	63.7	15.8	36.7 ± 1.27	56.5 ± .67	**75.7** ± 1.13	30.3	96.8	***8.2***	**0.0**
COPGEN ($\mathcal{T}\{$*inst=2k; pap=3k; pat=2k*$\}$)										
Llama-3 8B	9.6k (52.1%)	68.8	41.4	42.3 ± 1.11	60.3 ± 1.60	65.6 ± 1.23	36.8	96.6	25.4	2.7
Llama-3 8B SFT	27.0k (146.8%)	71.2	44.0	44.0 ± 1.47	49.4 ± 1.42	52.1 ± 1.36	45.0	97.8	53.7	27.6
w/ rep. removal	18.2k (99.1%)	***72.1***	**51.1**	44.7 ± 1.03	52.0 ± 1.13	55.4 ± 1.15	**50.7**	**98.2**	39.6	8.3
Mixtral-8x7B	6.3k (34.4%)	68.8	34.4	45.1 ± 2.00	64.4 ± 1.40	***70.6*** ± 1.24	39.5	96.9	14.4	0.1
Llama-3 70B	6.1k (33.4%)	70.5	39.5	45.9 ± 1.15	64.7 ± 1.06	68.7 ± .87	44.6	97.0	17.2	0.1
Qwen2-72B	8.2k (44.7%)	70.4	40.1	***46.9*** ± 1.26	***65.3*** ± .83	69.8 ± .42	44.3	97.0	***10.9***	**0.0**
COPGEN ($\mathcal{T}\{$*inst=2k; pap=3k; pat=400*$\}$)										
Qwen2-72B	17.2k (93.2%)	***71.5***	***50.5***	**49.9** ± 1.70	59.2 ± 1.00	64.9 ± .43	41.9	96.6	8.8	0.2

Table 5: Experimental Results on the non-contaminated test set. *Italic values* represent upper bounds that used test data in the prediction. The best value per column is **bold**, the best per section ***bold italics***. Tokens are reported as absolute and relative to the reference. BS = BERTScore. R-L = ROUGE-L. Text Sim = Standard Text Similarity Metrics.

E Patent Outline Example

```
# DESCRIPTION

## CROSS-REFERENCE TO RELATED APPLICATIONS

- reference prior applications

## BACKGROUND

- limitations of current text recognition methods

## SUMMARY

- outline method and system for character recognition

## DETAILED DESCRIPTION

- introduce character recognition difficulties
- describe lateral approach to character recognition
- define views and bounding box
- explain binarization and noise removal
- describe oblique/skew detection and removal
- outline segmentation process
- explain lateral-view-based analysis and characteristic points selection
- describe generation of feature vector
- outline classification and recognition with Artificial Neural Network
- describe training and knowledge base of Artificial Neural Network
- summarize system and method block diagram

### Handling Compound Characters

- introduce compound characters
- motivate lateral view based approach
- discuss limitations of conventional character recognition algorithms
- clarify scope and interpretation of patent claims
```

Listing 1: Example patent outline (short variant) for the pair W6364285-US20140112582. The outline corresponds to more than 5 pages. This example was randomly selected.

F Patent Structure

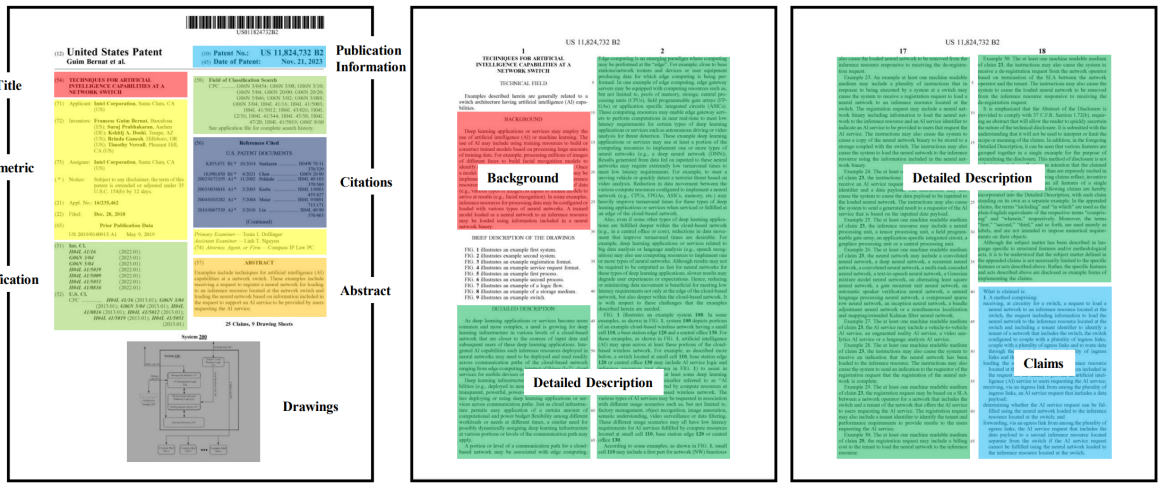

Figure 6: Illustration of the structure of a patent from Jiang and Goetz (2024). Note that multiple pages from the detailed description are omitted. The description includes all sections except the front matter and claims. In our experiments, we exclude sections containing only metadata, such as statements regarding funding.

G Hyperparameters

Generation		Training	
Parameter	Value	Parameter	Value
max sequence length	8192	max sequence length	8192
temperature	0.6	learning rate	0.00031622
		scheduler	cosine
		warmup ratio	0.1
		epochs	3
		batch size	32
		lora alpha	60
		lora dropout	0.05
		lora r	128

Table 6: Hyperparameters during generation and training. Training parameters are adopted from Tribes et al. (2024). We run the experiments on Nvidia H100 80GB GPUs. We use a single H100 for inference and training of the 8B model and 4xH100 for the inference of the larger models using tensor parallel. We estimate the total number of GPU-hours to be 720.

H SPARQL Query

```
PREFIX fabio: <http://purl.org/spar/fabio/>
PREFIX dct: <http://purl.org/dc/terms/>
PREFIX soa: <https://semopenalex.org/ontology/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX datacite: <http://purl.org/spar/datacite/>

SELECT DISTINCT
  ?paper ?authors ?author_names ?author_overlap ?title ?abstract ?date ?doi
  (GROUP_CONCAT(?location; SEPARATOR="\n") as ?locations)
  (GROUP_CONCAT(?url; SEPARATOR="\n") as ?urls)
  (GROUP_CONCAT(?pdf_url; SEPARATOR="\n") as ?pdf_urls)
  (GROUP_CONCAT(?license; SEPARATOR="\n") as ?licenses)

WHERE {
    SELECT DISTINCT
        ?paper
        (GROUP_CONCAT(DISTINCT ?author; SEPARATOR="\n") as ?authors)
        (GROUP_CONCAT(DISTINCT ?author_name; SEPARATOR="\n") as ?author_names)
        (SAMPLE(?author_overlap) as ?author_overlap)
        (SAMPLE(?title) as ?title)
        (SAMPLE(?abstract) as ?abstract)
        (SAMPLE(?date) as ?date)
        (SAMPLE(?doi) as ?doi)
        ?location
        ?url
        ?pdf_url
        ?license

    WHERE {
        # Information required for Matching
        ?paper a soa:Work ;
            dct:creator ?author ;
            dct:title ?title ;
            dct:abstract ?abstract ;
            dct:date ?date ;
            datacite:doi ?doi .

        ?author foaf:name ?author_name .

        # Optional Information for Downloading
        OPTIONAL {
            ?paper soa:hasLocation ?location .
            OPTIONAL { ?location fabio:hasURL ?url_ . }
            OPTIONAL { ?location soa:pdfUrl ?pdf_url_ . }
            OPTIONAL { ?location dct:license ?license_ . }
            BIND(COALESCE(?url_, "<EMPTY_PLACEHOLDER>") as ?url)
            BIND(COALESCE(?pdf_url_, "<EMPTY_PLACEHOLDER>") as ?pdf_url)
            BIND(COALESCE(?license_, "<EMPTY_PLACEHOLDER>") as ?license)
        }

        # Count number of matching authors
        {
            SELECT ?paper ?author_overlap
            WHERE {
                {
                    SELECT ?paper (COUNT(DISTINCT ?author) AS ?matching_authors)
                    WHERE {
                        ?paper dct:creator ?author .
                        ?author ?p ?name .
                        FILTER (?p IN (foaf:name, soa:alternativeName))
                        FILTER (?name IN (<AUTHOR_LIST>))
                    }
                    GROUP BY ?paper
                    ORDER BY DESC(?matching_authors)
                    LIMIT 500 # having too many results in the subquery will make query time out
                }

                ?paper dct:date ?date .

                BIND (xsd:float(?matching_authors) / xsd:float(<NUM_AUTHORS>) as ?author_overlap)
                FILTER (?author_overlap >= <AUTHOR_OVERLAP_THRESHOLD>)
                FILTER (?date > "<DATE_EARLIEST>"^^xsd:dateTime)
                FILTER (?date < "<DATE_LATEST>"^^xsd:dateTime)
            }
        }
    }
    GROUP BY ?paper ?location ?url ?pdf_url ?license
    HAVING (COUNT(?author) > 1)
}
GROUP BY ?paper ?authors ?author_names ?author_overlap ?title ?abstract ?date ?doi
```

Listing 2: SPARQL query template used to retrieve papers for a given patent from SemOpenAlex. Template variables `<var>` are filled based on the query patent.

I Summary Generation Prompt

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>

You are a highly skilled patent attorney with decades of experience in drafting high-quality patent applications. You answer every question in the most concise way possible, without adding unnecessary explanations.<|eot_id|><|start_header_id|>user<|end_header_id|>

### INSTRUCTION

For the sections of a patent application shown below, write a bullet list that summarizes the discourse structure of the document.

### OUTPUT FORMAT

The output needs to be in markdown syntax. Keep the headings as they are and add bullet lists summarizing the structure of each section. Do NOT write nested lists.

### GUIDELINES

Here are important guidelines you need to follow:

- **{n_words} words per bullet**: Every bullet point should summarize roughly {n_words} words in just a couple of words on a very high level.
- **Structure, not content**: The bullet points should not contain all the content. You should not write a summary of the content, but a summary of the structure! For instance, you should write 'motivate neural networks' rather than writing what the motivation for a neural network is.
- **Start with verbal phrases**: If applicable, start the bullet points with phrases like 'define', 'motivate', 'summarize', 'limitations of', 'application of' or 'embodiment'. You are not restricted to this set of phrases, just use them as inspiration. Avoid overusing the phrase 'describe'.
- **Specificity**: Avoid overly generic bullet points like 'define method' at all cost!
- **Conciseness**: Keep every bullet point as concise as possible! Do NOT write more than 5 words per bullet!
- **{n_bullets} bullet points in total**: You have a fixed budget of bullet points for the whole text. Make sure to write exactly {n_bullets} bullet points in total. This is with respect to all the text you are shown.
- **Coverage**: Since you cannot write more than {n_bullets} bullet points in total, make sure you don't make the list too fine-grained in the beginning. All text must be covered! Use numbers '(i/n)' after the dash as progress indicators with respect to the current section.

### EXAMPLE

Here is an example of the output format:

'''md
# HEADING 1 (0 bullet points)

## HEADING 1.1 (2 bullet points)

- (1/2) introduce neural networks
- (2/2) advantage over svm

## HEADING 1.2 (3 bullet points)

- (1/3) derivation of backpropagation
- (2/3) software design of automatic differentiation
- (3/3) example applications
'''


### Inputs

Here is the patent application you need to summarize:


'''md
{context}
'''<|eot_id|>
```

Listing 3: Prompt used to generate bullet point summaries with Llama-3 70B

J Patent Generation Prompt

```
### ROLE

You are a highly skilled patent attorney with decades of experience in drafting high-quality patent applications.
You assist scientists in transforming their scientific discoveries into lucrative patents.

### TASK DESCRIPTION

Your task is to draft a patent application.

### INPUTS

As input, you will be provided a research paper and a patent outline, each serving a distinct purpose.

1. Research Paper:

The research paper describes a novel invention to be patented.
The scientist has selected the most relevant excerpts from the paper.
Your task is to extract the invention from the paper and write a patent application for it.

2. Patent Outline:

The patent outline summarizes the desired discourse structure of the patent document.
It is in markdown format and contains a number of bullet points per section.
Use this outline as a rough guidance during drafting.
Note that the number of bullet points is also a strong indicator of the desired length! If bullet points are provided, each one corresponds to about 71 words or 1 paragraphs on average.
You should cover all content mentioned in the outline but you are not restricted to it! Feel free to add any further information that you feel would improve the patent application.

3. Prior Patent Outline:

Unless you are asked to generate the beginning of a patent, the user will also provide you with the outline of all prior content.
Use it as global context where you currently stand in the process and do not repeat yourself.

### GUIDELINES

There are a couple of guidelines you need to follow strictly:

- You might be asked to draft only parts of a patent document. Do not draft the whole patent but only those sections requested by the user.
- Copy the headings from the outline exactly. You must include only the headings provided in the outline!
- You must always write complete sentences and avoid keywords, bullet lists and enumerations!
- You must use proper language and maintain a very high level of detail, as you would expect to find in a good patent!
- The patent must act as a standalone document, therefore do not refer to the research paper in the patent!
```

Listing 4: System prompt used for outline-guided paper-to-patent generation

````
Here are the most relevant parts of the research paper describing the invention:

```md
# Abstract

Background Heart failure patients with reduced ejection fraction (HFREF) are heterogenous, and our ability to identify
      patients likely to respond to therapy is limited. We present a method of identifying disease subtypes using high-
      dimensional clinical phenotyping and latent class analysis that may be useful in personalizing prognosis and
      treatment in HFREF. Methods A total of 1121 patients with nonischemic HFREF from the β-blocker Evaluation of Survival
       Trial were categorized according to 27 clinical features. Latent class analysis was used to generate two latent
      class models, LCM A and B, to identify HFREF subtypes. LCM A consisted of features associated with HF pathogenesis,
      whereas LCM B consisted of markers of HF progression and severity. The Seattle Heart Failure Model (SHFM) Score was
      also calculated for all patients. Mortality, improvement in left ventricular ejection fraction (LVEF) defined as an
      increase in LVEF ≥5% and a final LVEF of 35% after 12 months, and effect of bucindolol on both outcomes were compared
       across HFREF subtypes. Performance of models that included a combination of LCM subtypes and SHFM scores towards
      predicting mortality and LVEF response was estimated and subsequently validated using leave-one-out cross-validation
      and data from the Multicenter Oral Carvedilol Heart Failure Assessment Trial. Results A total of 6 subtypes were
      identified using LCM A and 5 subtypes using LCM B. Several subtypes resembled familiar clinical phenotypes. Prognosis
      , improvement in LVEF, and the effect of bucindolol treatment differed significantly between subtypes. Prediction
      improved with addition of both latent class models to SHFM for both 1-year mortality and LVEF response outcomes.
      Conclusions The combination of high-dimensional phenotyping and latent class analysis identifies subtypes of HFREF
      with implications for prognosis and response to specific therapies that may provide insight into mechanisms of
      disease. These subtypes may facilitate development of personalized treatment plans.


# Introduction

...

We hypothesize that subtypes of nonischemic HFREF exist that may be differentiated by constellations of clinical features
      that reflect underlying pathophysiology. These subtypes may have variable clinical courses and responses to treatment
      , and identification of these subtypes may provide insight into mechanisms of HFREF and facilitate personalized
      prediction of outcomes and treatment response. Traditional outcomes-driven analyses are limited in the number of
      clinical features that can be evaluated due to the number of potential interactions between features contributing to
      the development and progression of HFREF. Latent class analysis is one statistical method of identifying groups of
      individuals within a population that share similar patterns of categorical variables such as symptoms or comorbid
      conditions, and it has been used in a number of medical disciplines including heart failure for exploration,
      characterization, and validation of diseases subtypes as well as for risk stratification and prediction of treatment
      response. [3]–[9] Latent class analysis has also been used to establish diagnostic standards for complex disease
      syndromes, and use of latent class analysis has been proposed as a method of dealing with large numbers of complex
      interactions and multiple comparisons in determining likelihood of response to interventions. [10]–[12] Briefly,
      latent class analysis hypothesizes the existence of unobserved classes within a population that explain patterns of
      association between variables and uses maximum-likelihood estimation to divide the population into subgroups by
      calculating a probability of subgroup membership for each symptom or comorbidity. An individual's subgroup membership
       may therefore depend on the presence or absence of many different characteristics in a given model. When the
      population in question has a shared disease, the results are data-driven definitions of disease subtypes where each
      subtype is characterized by a distinct combination of clinical features. Many clinical variables can thereby be
      incorporated into an analytic model while preserving statistical power for outcomes analysis by identifying the most
      prevalent combinations of variables upon which to focus. We propose using complex phenotype descriptions of patients
      in combination with latent class analysis to identify subtypes of nonischemic HFREF that may have different prognoses
       and likelihoods of treatment response.

...


# Methods


## Trial Design

The design of BEST has been described previously. [14], [15] A list of all recruitment sites is found in the Appendix S1.
      All patients had New York Heart Association (NYHA) class III or IV HFREF (LVEF ≤35%) and were randomized in a double-
      blind fashion to either bucindolol or placebo. Patients were considered ischemic if they had ≥70% obstruction in a
      major epicardial coronary artery by angiography or evidence of prior myocardial infarction and excluded from this
      analysis. [16] The primary endpoint was cumulative all-cause mortality. Secondary endpoints were all-cause mortality
      at one year and LVEF response defined as improvement in LVEF ≥5% with a final LVEF of ≥35% as measured using multi-
      gated acquisition scan (MUGA). The design of MOCHA has also been described previously. [13] All patients had an LVEF
      ≤35%, were mostly NYHA class II or III and had stable HF symptoms for 1 month prior to enrollment. They were
      randomized to placebo, low (6.25 mg bid), medium (12.5 mg bid), or high-dose (25 mg bid) carvedilol. Death and LVEF
      improvement as measured by MUGA were secondary endpoints in the original MOCHA analysis. Mortality data was only
      available up to one year of follow-up in MOCHA.
```
````

Listing 5: User prompt used for outline-guided paper-to-patent generation (Part 1)

```
## Identification And Definition Of Latent Classes

Patients were scored according to 27 clinical features (Tables 1 and 2). Criteria were encoded and applied in a MySQL server environment (Oracle Corporation, Redwood Shores, CA). [17] Patient clinical profiles were analyzed collectively using latent class analysis [18] applied to two sets of clinical variables we designated as Latent Class Models (LCM) A and B (Tables 1 and 2). LCM A and B differed only in the clinical variables included in each model. LCM A included variables that describe a patient's non-cardiac characteristics that can contribute to the pathogenesis of HFREF including age, gender, race, body mass index, and presence of comorbidities such as diabetes, atrial fibrillation, or valvular disease. [19]-[23] LCM B included variables that describe cardiac function, progression, and severity of HFREF including right- and left-ventricular function, hemodynamic parameters such as heart rate and blood pressure, end-organ function such as estimated creatinine clearance, and signs of venous congestion such as jugular venous distension and alanine aminotransferase levels. [24]-[33] In total, 3 variables were included in both models: body mass index, creatinine clearance, and hematocrit. All 3 variables have been implicated in the pathogenesis of HFREF and can also be markers of severity of HFREF. [34], [35] They were included in both models to illustrate that the variable implications of clinical features in different contexts may be represented using this approach. [34], [36]-[40] Two sets of related variables were also included: age of HF onset (LCM A) vs. chronologic age (LCM B) and presence of hypertension (LCM A) vs. presence of hypotension (LCM B). Age of HF onset, a static value, may be relevant to the HFREF etiology, while chronologic age may be related to HF progression. Similarly, presence of hypertension (LCM A) may be related to HF etiology while hypotension (LCM B) may be a marker of advanced HF.

...


## Association Between Latent Class Models And Outcomes

...


## Validation Of Multivariate Models

...


# Results


## Patient Characteristics

...


## Latent Class Model A (Table 1)

LCM A subtypes were characterized by distinct collections of clinical features that frequently resembled known HFREF syndromes. Subtype A1 was characterized by advanced age of onset, non-Caucasian race, male gender, HTN, mild-moderate renal insufficiency, and elevated rates of atrial fibrillation (24.5%). Subtype A2 was characterized by middle age of onset, female gender, moderate renal insufficiency, anemia, high body mass index, and very high rates of diabetes mellitus (74.6%), hypertension (95.0%), hyperlipidemia (93.8%), and hypertriglyceridemia (91.1%). Subtype A3 was characterized by middle age of onset, female gender, Caucasian race, hyperlipidemia, hypertriglyceridemia, anemia, and the presence of left bundle branch block (LBBB). Subtype A4 was characterized by young age of onset, non-Caucasian race, obesity, anemia, and lower rates of traditional cardiac risk factors such as hyperlipidemia, hypertriglyceridemia, and diabetes mellitus. Subtype A5 was characterized by advanced age of onset, Caucasian race, atrial fibrillation (86.2%), mitral valve disease (48.3%), aortic valve disease (21.8%), history of pacemaker placement (42.5%), and a significantly higher rate of prior sudden cardiac death (16.1%). This subtype had the smallest number of subjects (7.8%), whereas subtype A6 was the largest with 28.3% of subjects. Subtype A6 was characterized by middle age of onset, Caucasian race, male gender (100%), high body mass index, hypertension, hyperlipidemia, and hypertriglyceridemia with less associated diabetes mellitus (32.8%) than was seen in Subtype A2.


## Latent Class Model B (Table 2)

...


## Association With Outcomes

...


## Differences In Treatment Effects Between Latent Classes

...


## Combined Models

...


## Model Comparisons

...


## Validation

...
```

Listing 6: User prompt used for outline-guided paper-to-patent generation (Part 2)

```
# Discussion

Using the combination of high-dimensional clinical phenotyping and latent class analysis, we have identified a number of HFREF subtypes with distinct clinical profiles that demonstrate significant variation in prognosis as measured by all-cause mortality and response to bucindolol as measured by reduction in mortality and increased likelihood of LVEF response (Figure 4). Several of the LCM A subtypes resemble previously described nonischemic HFREF phenotypes, while LCM B subtypes model HF progression and severity. The latent class models, particularly LCM A, remained significantly associated with certain outcomes after combining them with the SHFM, suggesting that the information in the latent class models is different from the information in the SHFM Score. Taken together, these results suggest that our approach to HFREF subtype identification may be useful for identifying patients with potentially 'reversible' HFREF as well as those more likely to benefit from bucindolol.


## Insight Into Mechanisms Of Disease And Treatment Response

...


## Identification Of Hfref Subtypes Using Latent Class Analysis

This analysis demonstrates the potential utility of combining high-dimensional clinical phenotyping and latent class analysis for identifying relevant subtypes of HFREF. It is impossible to determine multivariate odds ratios for all of the variables included in the latent class models presented here using a traditional regression model, as the number of possible interactions (26,542,080 and 432,000,000 for LCM A and LCM B, respectively) prevents calculation using realistic sample sizes. Latent class analysis provides a quantitative mechanism of reducing the number of comparisons by aggregating individuals with similar clinical profiles. Our approach produces data-driven definitions of HFREF subtypes that integrate a large number of clinical features but are not dependent on any one feature for classification. Consequently, a feature like age may not have the same implications among all individuals. For example, subtype A4 is associated with worse outcomes than subtypes A2 or A6 despite younger age and lower burden of comorbid diseases. Clinical features may therefore be associated with a conditional probability for different outcomes depending on their context, capturing relevant interactions between comorbid conditions without direct calculation of all possible interactions. The added value of LCM A and B membership to SHFM for predicting survival despite sharing several common variables suggests that LCM A and B subtype may provide additional prognostic information to the SHFM Score. Finally, the variability in clinical outcomes observed between subtypes suggests that this approach could be useful in identifying patients with higher likelihood of HFREF reversibility in the absence of an obvious reversible etiology or conversely for identifying high risk patients for accelerated advanced HFREF therapy.


## Implementation And Sharing

...


## Limitations

...

Another important limitation is the generalizability of these latent class definitions. Utilization of the coefficients derived in this analysis to determine LCM subtype for other patients assumes that the patient population is the same as the nonischemic patients enrolled in BEST. This assumption may be particularly problematic for LCM B, which includes LVEF in its definition. Like all clinical trials, the inclusion and exclusion criteria of the BEST study are a critical source of selection bias and limit the generalizability of any predictive models developed from BEST to patients that do not meet those entry criteria. [14] This is especially relevant for data-driven latent class models like those presented here, as subtype definitions are by definition dependent on the original study population, and patient subtypes not present in the derivation population might be misidentified. It must also be remembered that latent classes only represent patterns of the variables included in the models, and that those latent classes may not necessarily exist as recognizable patient types in an independent population, [6] due in part to other variables that may be important in a disease process. The utility of these models must therefore be validated further in other patient populations, and the definitions of subtypes will need to be revised over time as more diverse patient populations are incorporated.

...


## Conclusion

High-dimensional phenotyping combined with latent class analysis provide a method of identifying subtypes of nonischemic HFREF patients who may have shared pathophysiology with implications for prognosis and response to bucindolol therapy. Significant reduction in all-cause mortality and increase in likelihood of LVEF response was associated with bucindolol treatment in specific groups identified using these classification methods. Identification of patients' HFREF subtype may provide a means of personalizing clinical prognosis and estimating likelihood of responding to medical treatment.


```
```

Listing 7: User prompt used for outline-guided paper-to-patent generation (Part 3)
```

````
Here is the outline of the desired patent application. Per bullet point, write roughly 1 paragraphs or 71 words.


First, here is the outline of what you have already written:

```md
# DESCRIPTION

## FIELD OF THE INVENTION

- define field of invention

## BACKGROUND OF THE INVENTION

- describe heart failure
- limitations of current therapy

## SUMMARY OF THE INVENTION

- motivate invention
- introduce HDCP and LCA
- describe subtypes and stages of non-ischemic HF
- correlate subtypes and stages with clinical course and β-blocker response
- describe retrospective analysis of BEST data
- compare results with SHFM predictions
- assess utility of SHFM in predicting response to β-blocker
- list exemplary β-blockers
- describe calculated SHFM Score
- evaluate β-blocker treatment, HF subtype, HF stage, and SHFM
- identify 6 HF subtypes and 5 HF stages
- associate HF subtype, HF stage, and SHFM with mortality and EF improvement
- perform multivariate analysis
- improve predictive performance with HF subtype and HF stage information
- describe method for predicting response to β-blocker therapy
- determine HF subtype and HF stage
- use HDCP to identify HF subtypes and HF stages
- describe method for treating non-ischemic HF patient
- determine treatment procedure for non-ischemic HF patient
- describe apparatus for determining HF subtype, HF stage, or combination thereof
```

Now, continue drafting and add the following points:

```md
# DESCRIPTION

## DETAILED DESCRIPTION OF THE INVENTION

- introduce non-ischemic HF patient prediction methods
- motivate classification of patients into subtypes
- describe clinical features influencing HF subtype and stage
- list clinical conditions used to determine HF subtype and stage
- outline steps to determine HF subtype
- explain calculation of HF subtype probability
- describe determination of HF stage
- outline apparatuses for determining HF subtype and stage
- describe input device for entering clinical condition information
- explain database for storing coefficients
- outline output device for displaying HF subtype and stage
- discuss updating coefficients with additional data
- describe high-throughput phenotyping method
- outline CPU and microprocessor for calculating HF subtype and stage
```
Limit your response to the sections mentioned in the summary: "DETAILED DESCRIPTION OF THE INVENTION". Remember what you have already written and do not repeat yourself.
````

Listing 8: User prompt used for outline-guided paper-to-patent generation (Part 4)

K Dataset Statistics

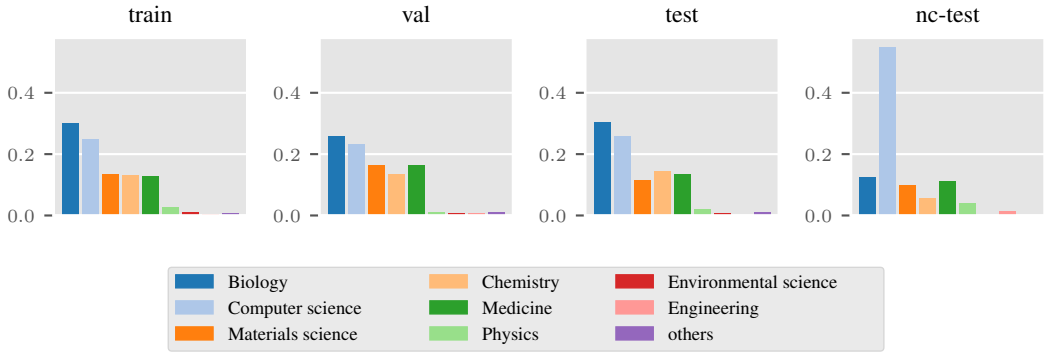

Figure 7: Distribution of domains across dataset splits. Domains are extracted from OpenAlex.

Differences between Domains. We analyze the performance across domains (see Appendix K for domain distributions) and show the results in Table 7. We include the two most represented domains in the dataset: computer science (CS) and biology (Bio). Reference patents from the biology domain are much longer than computer science patents. We find that generated Bio patents achieve better factuality but lower coverage across models, despite relative lengths being very similar. Stylistic similarity is also higher for Bio patents. Furthermore, model ranking differ between the domains: while Llama-3 70B performs best on Bio patents, Qwen2-72B is highly competitive on CS patents. However, further analysis is needed due to limited sample sizes.

	Style	Length	Coverage	Factuality
Biology (n=152)				
Llama-3 8B	41.7	12.2k / 23.0k	39.3 ± 1.4	61.0 ± 0.5
Llama-3 70B	48.9	7.7k / 23.0k	42.0 ± 0.9	66.2 ± 1.2
Mixtral-8x7B	47.7	7.4k / 23.0k	41.3 ± 1.1	63.7 ± 0.6
Qwen2-72B	48.3	10.0k / 23.0k	42.9 ± 1.1	63.2 ± 1.0
Computer Science (n=130)				
Llama-3 8B	37.5	7.0k / 13.1k	42.5 ± 1.2	61.0 ± 0.6
Llama-3 70B	44.5	4.6k / 13.1k	44.2 ± 0.8	63.4 ± 0.7
Mixtral-8x7B	44.7	4.0k / 13.1k	43.5 ± 0.9	61.2 ± 0.5
Qwen2-72B	45.4	5.3k / 13.1k	46.7 ± 0.8	61.6 ± 1.0

Table 7: Domain comparison. We report the style score, the length of the generated and reference patents (Length, generated/reference), coverage *Gen* $\rightarrow$ *Ref*, and factuality *Ref* $\rightarrow$ *Gen*.

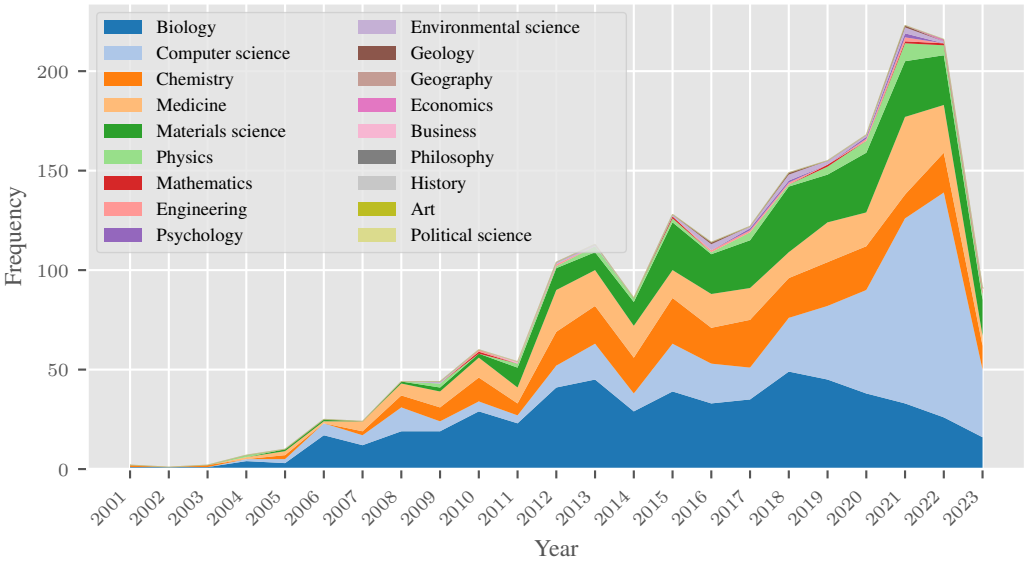

Figure 8: Distribution of domains over time. Domains are extracted from OpenAlex.

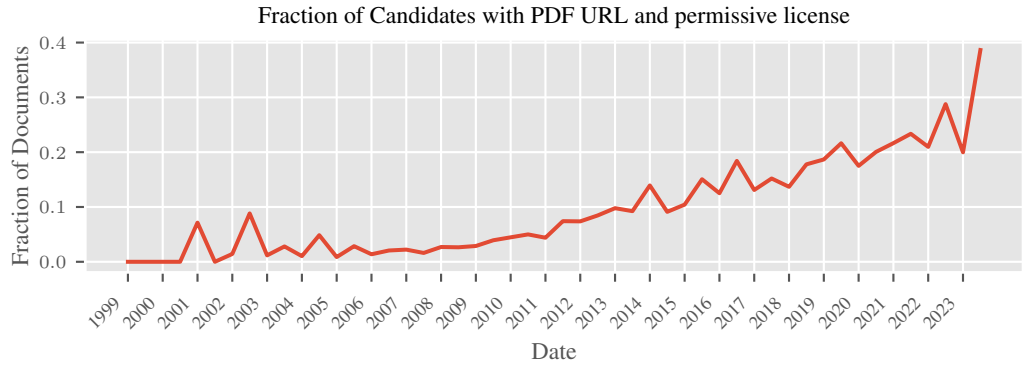

Figure 9: Fraction of permissive licenses over time.

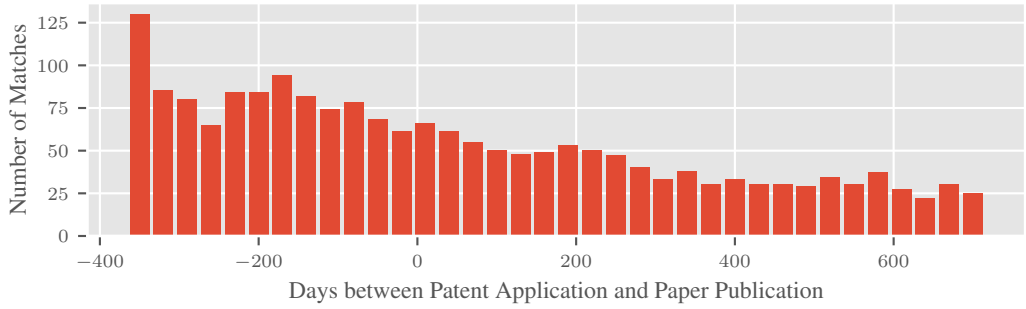

Figure 10: Date offsets between patents and papers. Negative offset means paper was published first.