SAUL: End-to-End Retrieval Augmented Patent Argumentation

Gabriel Au and Luis Brena and William Jurayj and Alexander Martin
Johns Hopkins University
{gau1, lbrenap1 , wjurayj1, amart233}@jhu.edu

Abstract

Patents are a corner stone for protecting the intellectual property of individuals and corporations. However, the process of writing patent proposals, applying for patents, granting patents, and litigating patent infringement are incredibly time intensive task. Searching through the prior art (existing patents) is a time intensive task which often is done by legal professionals, whose time comes at a high cost. Additionally protecting patents from being infringed upon by new technologies and new patent applications requires an extensive understanding of legal statutes and the patent approval process. In this work, we look to lower the barrier for both patent holders, appliers, granters, and litigators by introducing an end-to-end **S**ystem for **Pa**tent **U**nderstanding and **L**itigation (SAUL). SAUL consists of three distinct aspects: patent representation in structured forms, patent retrieval for finding related patents, and patent argumentation prediction for understanding the enforceability of patents. We report promising results in both retrieval and argumentation when using structured representations of patents. Code can be found at https://github.com/wjurayj/patent_saul.

1 Introduction

For a patent to be approved in the United States, it must meet four primary conditions: **(i)** the invention must be able to be used – the invention must work and cannot just be theory, **(ii)** the patent must include a clear description of how to make and use the invention **(iii)** the invention must be novel – something not done before **(iv)** the invention must be not obvious – as related to a change to something already invented[1].

When arguing for patent infringement or creating the claims of a patent, it is crucial to consider various factors that may arise during the patent claim process. Claims are evaluated according to established criteria (Filarksi and Shafer, 2010) and through judicial proceedings, such as Markman hearings withing the United States District Court (Surden, 2011), which serve to analyze the initial interpretations of the claims in question. Identifying the clauses that substantiate the validity of a patent and ascertain potential infringement is a bottleneck for the both patent writers, holders, and granters. For inventors, being able to find potential patents and understand what is similar can help identify areas where differentiation is important. For holders and granters, being able to know when their intellectual property (I.P.) is violated (holders) or prevent patents that overlap with the same I.P. (granters) is an important, but time consuming part of both patent applications and litigation.

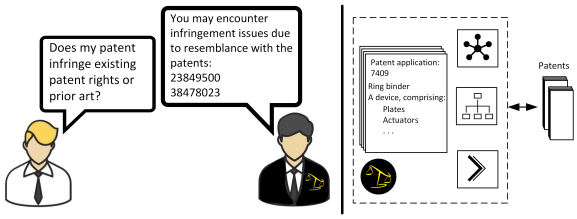

Figure 1: SAUL provides assistance in the patent application, granting, or litigation process by transforming patents into structured representations that are used to find similar prior arts and provide arguments for infringement, lowering the barrier in patent processes.

To better improve these bottlenecks for both the litigation and applications, we suggest that the involved parties "better call SAUL." SAUL (**S**ystem for **Pa**tent **U**nderstanding and **L**itigation) is our proposed end-to-end system for the patent application or patent litigation process. Specifically, we focus on two of the four main components for a patent: **(ii) the description of how to make and use the invention**, and **(iii) the novelty of an invention** . To better understand the description of the invention, we propose the decomposition of

[1] Constraints taken from `https://www.uspto.gov/patents/basics/essentials`

patent documents into structured representations, as templates or graphs, that can be used for both retrieving similar patents and argumentation of the enforceability of a patent. We find that the structured representations used by SAUL allow for less context and more focused inference during both retrieval and argumentation prediction. We see an increase in performance in both tasks when a structured representation instead of the full document. SAUL shows promising results for both retrieval and argumentation paving the way for future work in patent understanding. In summary our contributions are as follows:

1. We introduce the first dataset for patent enforceability argumentation, and demonstrate the utility of structured representations for outcome prediction.
2. We introduce SAUL: an end-to-end system for patent retrieval and litigation.
3. We introduce novel structured representations of patents to simplify their representation for both human readability and underlying modeling, increasing performance in retrieval and argumentation.

2 Related Work

Structured Representations are a common way to abstract long text, like documents, into a usable form. These representations have been extensively studied in information extraction literature (Baker et al., 1998; Doddington et al., 2004; Strassel and Mitchell, 2003; Sundheim, 1992). These works have seen large adoption in document-level information extracting for events and their arguments (Chen et al., 2023; Du and Cardie, 2021; Vashishtha et al., 2023). One specific work to note is Du and Cardie (2021), which extracts template information in a more natural form using question answering to target slot fillers in templates. We adopt a similar approach for our pure template representations.

However, templates are less understandable to untrained annotators (Gantt et al., 2024). To improve this limitation, some work has viewed the task of structured extraction from the lens of summarization (Gantt et al., 2024; Walden et al., 2024). Here, information in documents that would be extracted by a system is written in natural language making extractions more human readable. We draw inspiration from this when creating summaries of patent information using LLMs.

Another intuitive method for structured representations is graphs. Graphs are extensively researched in clustering, matching, and search (Fleshman et al., 2024; Fürstenau and Lapata, 2012; de Marneffe et al., 2007; Mohler et al., 2011; Okita, 2013). We take inspiration from the field of protein biophysics, where graphs with nodes and edges that contain physiochemical information can be matched to find exceedingly similar protein structures or binding sites (Weskamp et al., 2007; Malod-Dognin and Pržulj, 2014). We ask if dense embeddings can be used in the place of physiochemical information for a more nuanced search of similar patents.

Information Retrieval Recent methods in retrieval have adopted LLMs in their encoding methods (Ma et al., 2023; Weller et al., 2024). However, these tasks generally focus on natural language representations retrieving text documents from sentence queries. Recently Li et al. (2024), have implemented retrieval on structured json formatted representations. Other work, like Weller et al. (2024), has adopted information need into queries.

Neuro-symbolic reasoning In addition to the pervasive problem of hallucinations, or false generations, language models often will produce correct answers through faulty judgments (Huang et al., 2024; McCoy et al., 2019). While some task domains permit this type of epistemic failure, a high-stakes and litigious environment like legal reasoning requires outcomes to be reached through the correct process. A popular structure to ensure this robust support structure is an entailment tree, where each node's claim is compositionally entailed by its children such that the root node is supported by the set of atomic facts at the tree's leaves (Dalvi et al., 2021). The atomic facts at the leaves may be drawn from a knowledge graph, or otherwise retrieved from an unstructured knowledge base (Neves Ribeiro et al., 2022). This strategy simulates deductive reasoning, and mirrors approaches used in mathematical proofs (Bostrom et al., 2022). Neurosymbolic algorithms have showed strong results for searching over such trees to produce explainable reasoning traces for technical domains, but these typically require intermediate annotations of individual entailment steps (Weir et al., 2024). Towards enabling this type of proof tree search, we provide candidate entailment trees for the arguments we mine from 20 court cases, mirroring the Horn-clause strtucture used by Weir and Durme (2022).

Legal Argumentation In the interest of understanding the formal entailment structures present in reasoning, myriad argument mining techniques have been developed (Zhang et al., 2022). Notable efforts in the legal space include flat span categorizations on ECHR cases (Habernal et al., 2023) and structured efforts in CJEU fiscal aid cases which annotate both spans of texts and relations between these spans, such as support and attack (Santin et al., 2023). These structured representations have been used in non-legal argumentation as well, such as reasoning about beliefs (Saha et al., 2021) and in synthesizing persuasive arguments (Hua et al., 2019). Still, it retains particular importance in legal reasoning and argumentation because many questions of law allow for multiple possible interpretations, requiring courts to scrutinize justifications for claims in addition to the correctness of these claims (Hou et al., 2024).

3 Task Definitions

Retrieval Patent retrieval is the task of ranking a collection of patents based on their relevance to a given query. We represent a collection of patents D as structured representations $D_g = \{P_1^g, \ldots, P_N^g\}$, document representations $D_d = \{P_1^d, \ldots, P_N^d\}$, or pairs of structured representations $D_o = \{P_1^o, \ldots, P_N^o\}$, where $P_i^o = \{P_i^g, P_i^d\}$. A retrieval model takes a query Q, represented by natural language $Q = \{q_1, \ldots, q_m\}$ or a patent representation P, and returns a ranking of the patents that aligns with the true relevance judgments for the query.

Argumentation Given a set of patents $D = \{P_0, \ldots, P_n\}$ we would like to anticipate how a court case centered around this patent's enforceability would go. We treat a case as containing 4 key segments: a set of facts $F = \{f_0, \ldots, f_n\}$, a set of rules $R = \{r_0, \ldots, r_n\}$, the top-level claims made by the plaintiff and defendant $C = \{c_P, c_D\}$, expressing their desired outcome which they request the court to deliver, and finally the court's decision of the case's outcome $O \in C$. We define our task as building a predictive model for D based on c_P, c_D, and different combinations of external information, comparing the utility of the extracted facts F with the compressed versions of the involved patents such as D_d and D_g. We additionally produce proof-tree artifacts for future work in legal argument mining (Weir and Durme, 2022; Weir et al., 2024).

4 Patent Dataset

We first introduce a dataset of patents for this study, which includes 173636 unique patents published between 1988 and 2023 [2]. These patents have been formatted into structured files and are organized randomly, with filenames serving as identifiers.

Identifying related patents that may potentially infringe on one another is a complex task. To incorporate implicit supervision, we examine cases that reference a patent's claims and its enforceability. We select 20 patents across 20 cases as the query patents and use the other mentioned patents in the case as relevant patents for relevancy judgments in retrieval or establishing ground truth for legal argumentation [3]. This provides a set of patents between 1 (itself) and 9 relevant patents for each query.

Retrieval Data For relevancy judgments, we treat all patents mentioned in a case as "relevant" to one another and all other cases in the dataset as unrelated (providing a large number of distractor documents). In addition to the patent based queries, we introduce natural language queries that target specific information about a patent, following Weller et al. (2024). For natural language queries the only relevant patent is the one which the query is based on and all others are irrelevant. Each query patent has one human written query.

Argumentation In order to develop methods for understanding patent litigation, we focus on the subset of US patents which have been mentioned in court cases. We introduce a pipeline to collect cases involving patents from the web site casetext.com which discuss the terms "patent claims" and "enforceability". We use GPT-4o1 preview (Davis, 2024) to decompose these cases into the following components:

- Facts: the facts of the case, usually outlined the beginning.
- Rules: the inference rules of the case, including the formatted citations to the statutes or precedents that they are drawn from.
- Plaintiff and Defendant argument trees: the hierarchical structure that both sides use to support their top-level claim.

[2] The dataset can be found here: https://huggingface.co/datasets/lbrenap1/saul

[3] See specific patents used in Appendix A

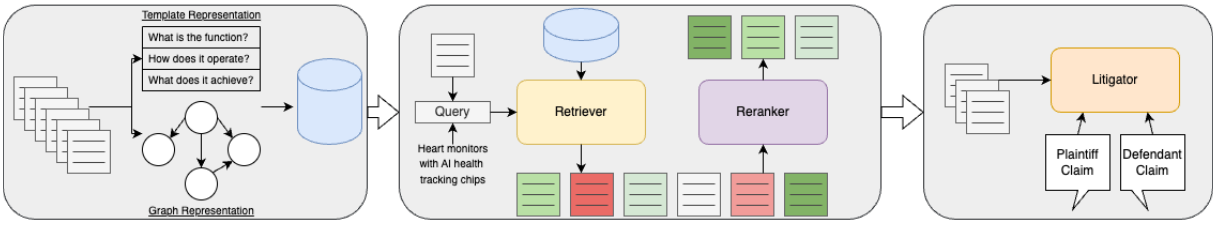

Figure 2: Overview of SAUL. In the first module, we convert long patent documents into smaller representations as graphs, templates, or summaries. In the second module, the retrieval module takes a patent representation or natural language query and retrieves documents from the index. Those documents are then reranked to provide a final list of relevant patents to the query. Lastly, the third module takes patents along with claims made defendent and prosecutors and predicts the enforcability of a patent along with the reasoning.

- Court's reasoning: the entailment tree that the court uses to justify their decision.

Every non-leaf node in each of the three argument trees has a set of children which collectively entail their parent. Exactly one of these children is an inference rule, while the remainder should be facts or other intermediate tree nodes. We hope that this data structure will enable future research into argument search, since this Horn clause construction echoes the logical structures used in proof tree search (Weir and Durme, 2022).

5 End-to-End Patent Understanding

We now introduce SAUL an end-to-end approach patent understanding. SAUL has three main aspects: (i) structured and natural langauge patent representations, (ii) patent retrieval for finding related patents based on natural language queries or patent documents, and (iii) patent argumentation for understanding the enforceability of patents in the context of legal statutes.

5.1 Patent Representation

We propose a few candidate representation structures for patents, and compare their performance on relevant patent retrieval and argument outcome prediction tasks. The relative brevity of these representations makes them efficient to store and reason over, while their structure helps the extraction process capture relevant information and streamlines reasoning processes.

Summary We represent patents using two different summaries. The first summary is created by the applicant of the patent. This summary serves as the abstract section of the patent document. The other summary representation is generated by a LLM. We adopt a similar summarization strategy to Gantt et al. (2024), where the LLM is directed to focus on information that may be extracted from a patent, such as the use, manufacturing process, and overview of the invention. Unlike Gantt et al. (2024), this summary is not conditioned on pre-extracted information due to the large variation in information contained in patents across domains. We generate these summaries using Phi-3.5-MoE (Abdin et al., 2024).

Given a patent $P^g = (p_1, p_2, \ldots, p_n)$, our objective is to produce an output sequence $S_\phi = (s_1, s_2, \ldots, s_m)$ (where $m < n$) that maximizes the conditional probability $P(S|P^g)$:

$$S = \arg\max_{S_\phi} P(S_\phi|P^g) \tag{1}$$

To maintain focus and avoid broadening the semantic range of the summary, we limit the output tokens to 1000 and set the temperature to 0.6.

QA We developed an interrogation methodology based on the doctrine of equivalents to analyze patent claims. By asking questions about function, purpose, mode of operation, and alternatives, we aimed to capture the key characteristics of the patent. This framework provides a comparative perspective across different patents.

Our goal is to create a clear semantic representation of each aspect of the doctrine, improving interpretative accuracy. Using the Phi-3.5-MoE model, we input the entire patent text and generated responses limited to 300 tokens, employing a temperature setting of 0.4 with a repetition penalty.

This method follows the triple identity test from legal precedents, assessing whether the invention performs the same function in the same way to achieve the same result. We believe it would be effective because it is based on human evaluation frameworks.

Graph Representation For the graph representation, we used Phi to generate code for inserting nodes and edges based on spans in patent claims. For each sentence (or semi-colon delineated segment) of a patent claim, Phi is asked to (1) identify relevant components based on the sentence and a list of past components, (2) describe each new component based on the sentence, and (3) infer relationships between the relevant components based on the sentence. Output from the model is parsed using Python to insert nodes and edges, discarding invalid entries.

Surprisingly, Phi did not require finetuning to produce reasonable results. By only varying system prompting and few-shot examples, the model generated relatively error-free and descriptive graphs for several test patents. However, scaling up to a larger set of patents led to more errors, which is a potential area for improvement.

The graph structure offer a simple framework for representing technologies, with nodes containing text detailing the composition of an invention, and edges having description of functional relationships between nodes. We choose to investigate graph structures as a potential method of analogous reasoning between different patent claims and a way to generate nuanced summaries of patent technologies for retrieval.

5.2 Patent Retrieval

For patent retrieval, we introduce an LLM based approach for retrieval and reranking following Ma et al. (2023). Specifically, we use LLaMA 2 (Touvron et al., 2023) as the underlying language model.

Our task requires a unique data: structured representations of patents, which most likely has not been seen during LLM pretraining. Additionally, retrievers like RepLlama and RankLlama (Ma et al., 2023) are only trained for passage retrieval with natural language queries (typically a question). However, our current experiment is limited in the amount of data we have. We do not have enough patents to finetune a LM and evaluate extensively. Instead we pretrain on similar strucutred data using event templates structured as QA pairs, following Du and Cardie (2021), with the templates and documents coming from FAMuS (Vashishtha et al., 2023). We provide further details to this setup in Appendix B.

Embedding During finetuning we append an end-of-sequence token to the input of the query and document so that LLaMA pools the dense representation in the final token. Thus, the vecto embedding of a query or document is computed as:

$$V = \text{Decoder}(t_1 \dots t_m \text{EOS})$$

We then calculate the similarity betewen a query and document in terms of the dot product similarity of the correspond dense representation V as $\text{S}(Q, D) = \langle V_Q, V_D \rangle >$. The model is then trained using an InfoNCE loss calulated:

$$\mathcal{L}(Q, D^+, \{D_N\} = -\log p(D = D^+|Q) = \\ -\log \frac{\exp(\text{S}(Q, D^+))}{\exp(\text{S}(Q, D^+)) + \sum_{D^-} \exp(\text{S}(Q, D^-))}$$

During inference in the retrieval module of SAUL, the documents are pre-encoded from any representation to create an index of documents. The queries are then encoded real-time and we perform search on the index in a (simple) flat index setting to demonstrate model effectiveness.

Reranking During finetuning of our reranker, we deviate from Ma et al. (2023) towards Nogueira et al. (2020). We implement our LLaMA reranker through language modeling, where the learns to focus on only the tokens YES and NO. This allows the reranking to be simply trained as through a language modelling objective depending on the query/document pair. Specifically, we format the input to the language model as <Document> [SEP] <Query>. We first encode the document because we want to ensure its representation is independent of the query. Then our language modeling objective is

$$\mathcal{L} = -\log p(x|D, Q)$$

where x is the single token decoded during the language modeling and corresponds to the YES/NO label for the document.

5.3 Legal Argumentation

We evaluate two open-source language models on the task of legal outcome prediction, which requires the inference about which side of a legal dispute is more likely to win (Valvoda and Cotterell, 2024). For each case, we prompt two types of Llama models to choose which litigant (plaintiff or defendant) is more likely to win the case. In the control case, the model must choose between only the names and the top-level claims made by the plaintiff and defendant, with no context or justifications. We evaluate two patent-centric cases, in which this instruction is pre-pended with the set of patents that are mentioned in the case. In the **Questions** case, these patents are represented as the question-answer pairs, while the **Summary** case represents these patents as the provided abstract. In both cases, the adjudicating language model can also see the names and holders of each patent. Finally, we show a reasonable upper bound of performance by replacing these patent representations with the full list of the extracted facts from the case.

6 Results

6.1 Retrieval

We evaluate two variations of our model: retrieval only (SAUL-R) and retrieval with reranking (SAUL-RR), both with LLaMA2-7B, against OkapiBM25 (BM25) (Robertson and Zaragoza, 2009). We evaluate three different retrieval settings: patent-to-patent (Doc2Doc) using the full patent as the query and document, template-to-template (T2T) using the QA template representations as the query and documents, and summary-to-summary (Sum2Sum) using summaries as the patents as the query and documents. For evaluation on all methods, we report recall at 1 (R@1), 5 (R@5) and 10 (R@10), NDCG and NDCG@10, mean average percision (mAP), and the probability that a relevant document is ranked before a non relevant one (AUC). For all metrics, higher is better.

In Table 1, we observe interesting results from our methods. The first thing we observe is the limitation of using documents as the retrieval. When performing inference, the inference takes longer than 24hrs or causes out of memory with increased batch sizes, demonstrating the unsuitability for large contexts in retrieval. We leave smarter inference for future work. Additionally, we notice similar performance among the document and structured/reduced representations. However, the increased inference time of the smaller representations with similar performance is preferred. Lastly, we notice a trend in structured extractive techniques with our generated extractive summaries and templates performing better than the human written patent summaries. This is most likely due to the more focused nature of our representations focusing on what goes into a patent rather than the overall patent and protections.

Natural Language Queries A natural extension of our retrieval method is the use of natural language queries instead of documents and structured representations. When brainstorming products in industry or potentially patentable products, being able to ensure that an ideas is not protected intellectual property is important to do before committing large amounts of hours or money to producing the invention. Weller et al. (2024) introduced the use of instruction based natural language queries to target information needs in queries. We use Promptriever (Weller et al., 2024) based on LLaMA2-7B for this exploration of queries that target specific information about the function, operation, and results of a patent. We hand write queries for these topics like "find me patents in medicine technology that detect and quantify rare mutations in cell-free DNA from bodily fluids." We elaborate on the query creation and experimental setup in Appendix D.

In Table 2, we report the results of this ablation experiment. We report the retrieval models ability on natural language forms of our representations. In this setting, we see that the method which uses template representations performs better than the generated summary. We attribute this performance slightly to the annotation protocol, since the queries were written with the answers to the questions in mind. However, further investigation of the differences would be interesting.

6.2 Graph Representations

We provide an ablation study to evaluate our graph representation on retrieving patents. This method attempts to determine claim similarities by calculating a pairwise novelty score between two claim graphs. First, nodes from both graphs are embedded in a 384-dimensional space with a small encoder model, bge-small (Xiao et al., 2023). We then use HDBScan, a hierarchical clustering algorithm (Campello et al., 2015), to group nodes into sets of supernodes, irrespective of the graph from which the node came from. Nodes within supernodes are referred to as "sub-nodes."

The general goal of constructing supernodes is to establish an analogous comparison between two claims. Supernodes represent semantic groupings in the two claims. If supernodes contain sub-nodes from different claims, then there exists a point of compositional similarity. Beyond compositional similarity, the edges between overlapping supernodes will then correspond to a functional similarity in the claims in addition to having similar components.

After this step, a general novelty score is calculated from the percentage of sub-nodes in over-

Setting	Model	R@1	R@5	R@10	NDCG	NDCG@10	mAP	AUC
Doc2Doc	BM25	27.47	74.37	80.95	0.8629	0.8629	80.95	1.0
	SAUL-RR	-	-	-	-	-	-	-
T2T	BM25	27.47	74.37	80.95	0.8629	0.8629	80.95	1.0
	SAUL-RR	27.47	69.77	76.35	0.8405	0.8155	75.19	97.27
GS2GS	BM25	27.47	74.37	80.95	0.8605	0.8605	80.52	99.79
	SAUL-RR	27.47	73.79	80.95	0.8595	0.8595	80.56	99.60
PS2PS	BM25	22.87	50.76	60.63	0.7122	0.6238	57.16	78.26
	SAUL-RR	17.89	38.48	45.06	0.6064	0.4492	46.18	67.92

Table 1: Retrieval Results

Rep	R@1	R@5	NDCG	AUC
Sum	22.91	50.43	0.6013	75.77
Temp	25.18	69.79	0.8011	84.35

Table 2: Results of Promptriever on different patent representations.**Sum:** summaries, **Temp:** QA template representation

	R@1	R@5	NDCG	mAP
SAUL-R	27.47	69.71	0.8502	75.38
Graph	27.47	66.42	0.7781	68.66

Table 3: Comparison of a graph-based clustering and SAUL-R on the graph representation.

lapping supernodes between the two graphs. This score represents the general novely to the components in the claim, denoted as the "cluster novelty score",

$$N_{\text{Clus}} = \frac{\text{Total Nodes} - \text{Overlapping Nodes}}{\text{Total Nodes}} \quad (2)$$

Each supernode's description arises from aggregating all sub-node descriptions and edges between sub-nodes in the same supernode set. Conversely, superedges between supernodes have descriptions aggregated from all edges between sub-nodes in different supernode sets. Cosine distances from the same dense embeddings denote the similarity between each analogous supernode (contains sub-nodes from both claim graphs) and their respective superedges. Thus, a novelty score based on supernodes (compositional novelty) and edge (functional novelty) is,

$$N_{\text{Comp/Fun}} = 1 - \text{Cosine Similarity} \quad (3)$$

To compare one patent against another, the three metrics (cluster, compositional, and functional novelty) are calculated between each claim. Using the minimum novelty scores for each claim, we determine a claim-level novelty, as calculated below.

$$N_{\text{Claim}} = N_{\text{clus, min}} \frac{N_{\text{comp, min}} + N_{\text{fun, min}}}{2} \quad (4)$$

We use minimum novelty scores instead of a mean, as a claim will only infringe strongly on one other claim if two patents are identical. Broader patent-level novelty is the mean of the claim-level novelty scores.

In Table 3, we can see that the retrieval performance of the graph representations is similar to the performance of SAUL-R when performing retrieval using the graph representation. We additionally show a heatmap detailing patent-level total novelty, clustering novelty, compositional novelty, and functional novelty are shown in Figure 3. In this heatmap we see that patents from the same cases are often similar to each other and that most other patents are irrelevant unless from a similar domain (like technology). We note that experimentation with this style of clustering may be useful for methods like Fleshman et al. (2024), however, we leave this exploration to future work.

6.3 Legal Argumentation

In the control case, where the model is required to simply choose blindly between contradictory claims, both 8b and 70b parameter models perform slightly worse than chance. Access to both patent summaries and question-answer sets for all patents involved (alongside metadata like their titles and patent holders) improves from this baseline. Notably, although the smaller model outperforms using the summaries, the larger model beats all other

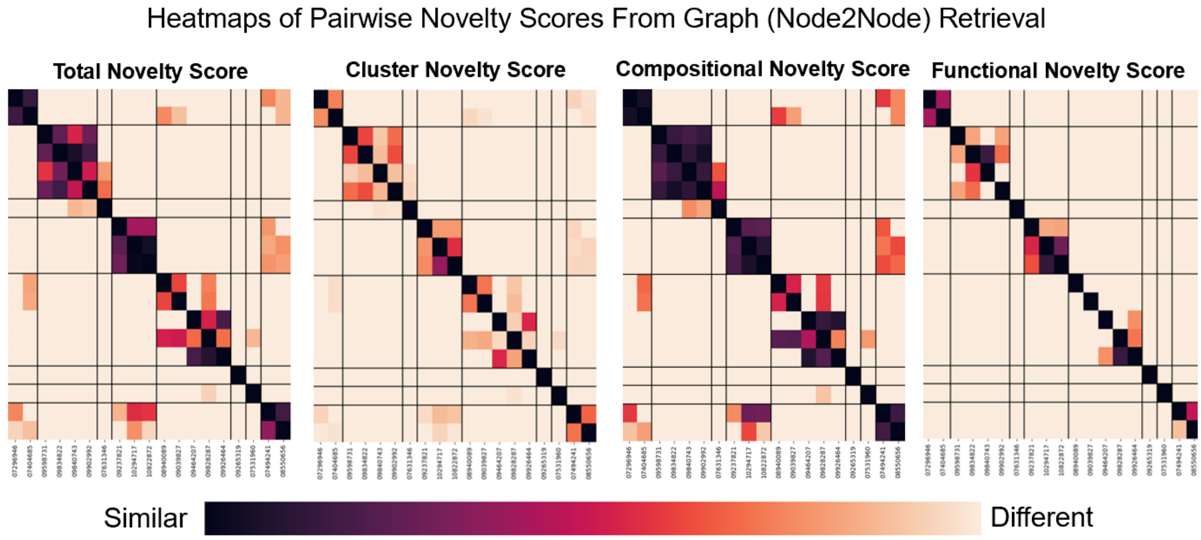

Figure 3: Heatmap showing pairwise novelty scores from Node2Node graph retrieval. Patent numbers are shown on the bottom, with lines denoting patents in the same case group.

patent-centric combinations when given access to the question-answer representations.

We also conduct some qualitative analysis of these argument components. Of the 15 rules and 36 facts extracted from these 20 cases that mention any version of "enforceability", 11 rules and 18 facts place responsibility for the enforceability of the patent on the related concept of "inequitable conduct". Inequitable conduct is not a feature of a patent itself, but instead describes the intention of a patent applicant to deceive the USPTO (McGowan, 2010). Therefore, the applicability of these rules will not be determined by the contents of the patent itself. We posit that this feature of patent litigation places a hard upper bound on how well cae outcome prediction can be performed using information from patents alone. Another possible reason the facts-centric prompt outperforms could be that information from the final decision leaks into the way that facts are phrased, since the entire case is written by a judge after she has determined the outcome of the case.

7 Conclusion

In this work, we introduced SAUL, a novel approach for end-to-end patent understanding. SAUL uses fine-grained extractive representations of patents to represent patents for both more succinct and smaller context representations. Additionally, the dataset and method are the first in the field for patent retrieval and litigation providing a strong base for future work. We find that our representations provide for efficient retrieval with similar performance to full documents. Furthermore, we show that large language models are most empowered to predict case outcomes using our structured patent representations, delivering substantial improvements over baselines.

Model	Control	Questions	Summary	Facts
8B	0.35	0.45	0.55	0.9
70B	0.40	0.65	0.50	0.9

Table 4: Results of legal argumentation. Control case asks the language model to choose between sides based only on top-level requests of the court. Remaining columns indicate whether the language model has access to Question/Answer or Summary format of relevant patents, or access to the case's extracted facts.

Limitations

Due to time and resource constraints, legal expertise was not procured for any annotations or evaluations presented. While we aim to provide robust self-supervision in its place, we recommend that practitioners consult legal experts before deploying our system into a real-world setting.

Retrieval Our retrieval setting is limited by the the representation of the documents. We use a dense single vector representation of patents. While efficient, requiring only one dot product operation in retrieval, the method severely limits the expressivity of the representation in retrieval.

Multi-vector approaches like tokenwise interaction (Khattab and Zaharia, 2020) or nugget (Qin and Van Durme, 2023) may provide better representations for retrieval. Graph retrieval is also limited by graph construction time. Although Phi is a small model, it takes a considerable amount of time to generate each graph, which makes the method better suited for small patents sets which require more detailed comparison.

Argumentation Due to resource constraints, we only decompose 20 cases discussing patent enforceability. Furthermore, our decompositions have not been reviewed by legal professionals. A common theme in these cases is the influence of outside factors in deciding a case, such as intent to deceive or procedural matters–this places a hard upper bound on the potential performance of our patent-centric approach to outcome prediction.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada. Association for Computational Linguistics.

Kaj Bostrom, Zayne Sprague, Swarat Chaudhuri, and Greg Durrett. 2022. Natural language deduction through search over statement compositions. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4871–4883, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ricardo J. G. B. Campello, Davoud Moulavi, Arthur Zimek, and Jörg Sander. 2015. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Trans. Knowl. Discov. Data*, 10(1).

Yunmo Chen, William Gantt, Weiwei Gu, Tongfei Chen, Aaron White, and Benjamin Van Durme. 2023. Iterative document-level information extraction via imitation learning. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1858–1874, Dubrovnik, Croatia. Association for Computational Linguistics.

Bhavana Dalvi, Peter Alexander Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021. Explaining answers with entailment trees. In *Conference on Empirical Methods in Natural Language Processing*.

Ernest Davis. 2024. Testing gpt-4-o1-preview on math and science problems: A follow-up study. *Preprint*, arXiv:2410.22340.

Marie-Catherine de Marneffe, Trond Grenager, Bill MacCartney, Daniel Matthew Cer, Daniel Ramage, Chloé Kiddon, and Christopher D. Manning. 2007. Aligning semantic graphs for textual inference and machine reading. In *American Association for Artificial Intelligence*.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Xinya Du and Claire Cardie. 2021. Event extraction by answering (almost) natural questions. *Preprint*, arXiv:2004.13625.

Tom Filarksi and Heather N. Shafer. 2010. *Patent Litigation Strategies Handbook*, chapter 28. Bloomberg BNA, Arlington, VA.

William Fleshman, Aleem Khan, Marc Marone, and Benjamin Van Durme. 2024. Adapterswap: Continuous training of llms with data removal and access-control guarantees. *Preprint*, arXiv:2404.08417.

Hagen Fürstenau and Mirella Lapata. 2012. Semi-supervised semantic role labeling via structural alignment. *Computational Linguistics*, 38:135–171.

William Gantt, Alexander Martin, Pavlo Kuchmiichuk, and Aaron Steven White. 2024. Event-keyed summarization. *Preprint*, arXiv:2402.06973.

Ivan Habernal, Daniel Faber, Nicola Recchia, Sebastian Bretthauer, Iryna Gurevych, Indra Spiecker genannt Döhmann, and Christoph Burchard. 2023. Mining legal arguments in court decisions. *Artificial Intelligence and Law*, 32(3):1–38.

Abe Hou, William Jurayj, Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme. 2024. Gaps or hallucinations? scrutinizing machine-generated legal analysis for fine-grained text evaluations. In *Proceedings of the Natural Legal Language Processing Workshop 2024*, pages 280–302, Miami, FL, USA. Association for Computational Linguistics.

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

Xinyu Hua, Zhe Hu, and Lu Wang. 2019. Argument generation with retrieval, planning, and realization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2661–2672, Florence, Italy. Association for Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2024. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. *Preprint*, arXiv:2004.12832.

Millicent Li, Tongfei Chen, Benjamin Van Durme, and Patrick Xia. 2024. Multi-field adaptive retrieval. *Preprint*, arXiv:2410.20056.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2023. Fine-tuning llama for multi-stage text retrieval. *Preprint*, arXiv:2310.08319.

Noël Malod-Dognin and Nataša Pržulj. 2014. Gr-align: fast and flexible alignment of protein 3d structures using graphlet degree similarity. *Bioinformatics*, 30(9):1259–1265.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Annual Meeting of the Association for Computational Linguistics*.

David McGowan. 2010. Inequitable conduct. *"Loyola of Los Angeles Law Review"*.

Michael Mohler, Razvan C. Bunescu, and Rada Mihalcea. 2011. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Annual Meeting of the Association for Computational Linguistics*.

Danilo Neves Ribeiro, Shen Wang, Xiaofei Ma, Rui Dong, Xiaokai Wei, Henghui Zhu, Xinchi Chen, Peng Xu, Zhiheng Huang, Andrew Arnold, and Dan Roth. 2022. Entailment tree explanations via iterative retrieval-generation reasoner. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 465–475, Seattle, United States. Association for Computational Linguistics.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. *Preprint*, arXiv:2003.06713.

Tsuyoshi Okita. 2013. Local graph matching with active learning for recognizing inference in text at ntcir-10. In *NTCIR Conference on Evaluation of Information Access Technologies*.

Guanghui Qin and Benjamin Van Durme. 2023. Nugget: neural agglomerative embeddings of text. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3:333–389.

Swarnadeep Saha, Prateek Yadav, Lisa Bauer, and Mohit Bansal. 2021. ExplaGraphs: An explanation graph generation task for structured commonsense reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7716–7740, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Piera Santin, Giulia Grundler, Andrea Galassi, Federico Galli, Francesca Lagioia, Elena Palmieri, Federico Ruggeri, Giovanni Sartor, and Paolo Torroni. 2023. Argumentation Structure Prediction in CJEU Decisions on Fiscal State Aid. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, pages 247–256, Braga Portugal. ACM.

Stephanie Strassel and Alexis Mitchell. 2003. Multilingual resources for entity extraction. In *Proceedings of the ACL 2003 Workshop on Multilingual and Mixed-language Named Entity Recognition*, pages 49–56, Sapporo, Japan. Association for Computational Linguistics.

Beth M. Sundheim. 1992. Overview of the fourth Message Understanding Evaluation and Conference. In *Fourth Message Understanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992*.

Harry Surden. 2011. Efficient uncertainty in patent interpretation. *Wash. Lee L. Rev*, 68.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Josef Valvoda and Ryan Cotterell. 2024. Towards Explainability in Legal Outcome Prediction Models. *arXiv preprint*. ArXiv:2403.16852 [cs].

Siddharth Vashishtha, Alexander Martin, William Gantt, Benjamin Van Durme, and Aaron Steven White. 2023. Famus: Frames across multiple sources. *Preprint*, arXiv:2311.05601.

William Walden, Pavlo Kuchmiichuk, Alexander Martin, Chihsheng Jin, Angela Cao, Claire Sun, Curisia Allen, and Aaron Steven White. 2024. Cross-document event-keyed summarization. *Preprint*, arXiv:2410.14795.

Nathaniel Weir and Benjamin Van Durme. 2022. Nellie: A neuro-symbolic inference engine for grounded, compositional, and explainable reasoning. In *International Joint Conference on Artificial Intelligence*.

Nathaniel Weir, Kate Sanders, Orion Weller, Shreya Sharma, Dongwei Jiang, Zhengping Jiang, Bhavana Dalvi Mishra, Oyvind Tafjord, Peter Jansen, Peter Clark, and Benjamin Van Durme. 2024. Enhancing systematic decompositional natural language inference using informal logic. *Preprint*, arXiv:2402.14798.

Orion Weller, Benjamin Van Durme, Dawn Lawrie, Ashwin Paranjape, Yuhao Zhang, and Jack Hessel. 2024. Promptriever: Instruction-trained retrievers can be prompted like language models. *Preprint*, arXiv:2409.11136.

Nils Weskamp, Eyke Hullermeier, Daniel Kuhn, and Gerhard Klebe. 2007. Multiple graph alignment for the structural analysis of protein active sites. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(2):310–320.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Gechuan Zhang, Paul Nulty, and David Lillis. 2022. A decade of legal argumentation mining: Datasets and approaches. In *International Conference on Applications of Natural Language to Data Bases*.

A Patents Used

1. World Wide Stationery Mfg. Co. v. U.S. Ring Binder
   - U.S. Patent No. 7,296,946
   - U.S. Patent No. 7,404,685
2. ORTHO-McNEIL PHARMACEUTICAL, INC. v. KALI LABORATORIES
   - U.S. Reissue Patent No. RE39,221
   - U.S. Patent No. 5,336,691
3. ICM Controls Corp. v. Honeywell Int'l, Inc.
   - U.S. Patent No. 5,889,645
   - U.S. Patent No. 5,337,206
   - U.S. Patent No. 6,222,719
4. Guardant Health, Inc. v. Found. Med., Inc.
   - U.S Patent No. 9,598,731
   - U.S Patent No. 9,834,822
   - U.S Patent No. 9,840,743
   - U.S Patent No. 9,902,992
5. Amgen, Inc. v. Ariad Pharmaceuticals, Inc.
   - U.S. Patent No. 6,410,516
6. Roche Palo Alto LLC v. Apotex, Inc.
   - U.S. Patent No. 5,110,493
7. Int'l Bus. Machs. Corp. v. Priceline Grp. Inc.
   - U.S. Patent No. 7,631,346
   - U.S. Patent No. 5,961,601
   - U.S. Patent No. 5,796,967
   - U.S. Patent No. 7,072,849
   - U.S. Patent No. 5,347,632
   - U.S. Patent No. 4,688,167
   - U.S. Patent No. 5,758,072

- U.S. Patent No. 5,594,910
- U.S. Patent No. 4,805,134

8. Vehicle IP, LLC v. General Motors Corp.
    - U.S. Patent No. 6,535,743
    - U.S. Patent No. 5,543,789
9. Magnetek, Inc. v. Kirkland Ellis
    - U.S Patent No. 5,432,409
10. GeigTech E. Bay v. Lutron Electronics Co.
    - U.S. Patent No. 9,237,821
    - U.S. Patent No. 10,294,717
    - U.S. Patent No. 10,822,872
    - U.S. Patent No. 1,476,160
11. Knauf Insulation, LLC v. Johns Manville Corp.
    - U.S. Patent No. 8,114,210
    - U.S. Patent No. 8,940,089
    - U.S. Patent No. D631,670
    - U.S. Patent No. 9,039,827
    - U.S. Patent No. 9,464,207
    - U.S. Patent No. 9,469,747
    - U.S. Patent No. 9,828,287
    - U.S. Patent No. 9,926,464
12. Cambridge Mobile Telematics, Inc. v. Sfara, Inc.
    - U.S. Patent No. 8,989,952
    - U.S. Patent No. 9,333,946
13. Monsanto Co. v. Bayer Bioscience
    - U.S. Patent No. 5,545,565
    - U.S. Patent No. 5,767,372
    - U.S. Patent No. 6,107,546
    - U.S. Patent No. 5,254,799
14. Peerless Indus., Inc. v. Crimson Av LLC
    - U.S. Patent No. 7,823,850
15. Sloan Valve Co. v. Zurn Industries, Inc.
    - U.S. Patent No. 7,607,635
    - U.S. Patent Application Publication No. 2006/015729
    - U.S. Patent No. 7,481,413
    - U.S. Patent No. 1, 912,937
16. Divine Purse Co. v. Shepax, Inc.
    - U.S. Letters Patent No. 9,265,319 B1
    - U.S. Patent No. 9,265,319 B1
17. Everlight Elecs. Co. v. Nichia Corp.
    - U.S. Patent No. 6,653,215
    - U.S. patent No, 5,998,925
    - U.S. Patent no. 7,531,960
    - Japan's Patent Office, patent number H10-36835
18. E.I. DU PONT DE NEMOURS COMPANY v. MACDERMID, INC.
    - U.S. Patent No. 6,171,758 B1
    - U.S. Patent No. 6,773,859 B2
19. Kenall Mfg. Co. v. Cooper Lighting, LLC
    - U.S. Patent No. 6,984,055
    - U.S. Patent No. 7,494,241
    - U.S. Patent No. 8,550,656
    - U.S. Patent No. RE45,563
    - U.S. Patent No. RE45,591
    - U.S. Application No. 10/156,423
20. CFMT, Inc. v. Steag Microtech Inc.
    - U.S. Patent No. 4,911,761
    - U.S. Patent No. 4,778,532
    - U.S. Patent No. 4,816,081
    - U.S. Patent No. 4,722,752
    - U.S. Patent No. 4,736,760
    - U.S. Patent No. 4,643,774

B Event Retrieval Pretraining

To pretrain the model, we adapt a similar strategy to Ma et al. (2023). Specifically, we use LoRA (Hu et al., 2021) on a LLaMA-2-7B model creating two different adapters for embedding and reranking. We then finetune on the FAMuS dataset (Vashishtha et al., 2023) using templates as the structured representations.

Relevancy Judgements The FAMuS dataset consists of event templates annotated from spans of reports (short passages) and source documents. In the annotation, they have 5 event templates for each template which have annotations in a document. There is a 1-1 correspondance between a document and event template and there is only one event annotated per document. We use the source documents (as they are longer and approximately match the length of patent documents). During retrieval (embedding) training, we use the template-document as the only relevant pair. During reranking, we give a relevancy score of 2 (highest) to the exact document match and give a partial relevancy score (1) to all documents with the same event type. All other documents are scored as not relevant (0). While this is not an ideal reranking setting, we believe it was the best approach to allow for partial relevancy in similar topics during reranking. However, during training we ignore these rankings and use all scores 1 and 2 as relevant for the list.

Template Format Following Du and Cardie (2021), we format our templates as question-answer pairs. We use a simple question of "what is the **slot name**?" which follows directly from Du and Cardie (2021). Then the answers are the spans of text annotated during the FAMuS dataset collection. An example template would be:

> **Event:** Activity_pause **[SEP]** What is the Activity? The public inquiry into the controversial Mottram - Tintwistle bypass **[SEP]** What is the Agent? John Watson **[SEP]** What is the Place? Stalybridge Civic Hall **[SEP]** What is the Time? when the Highways Agency admitted it had got its figures wrong **[EOS]**.

C Natural Language Queries

To create natural language queries, one of the authors read patents along side the answers to the QA templates answers for function (What is the essential function of the product or process?), operation (How does the product or process operate?), and results (What result does this product or process achieve?). Based on the answers to these questions and the general patent, the annotator creates a query that targets information of one or more from 4 categories: (i) the topic/area of the patent, (ii) the function of the invention, (iii) the operation of the invention, and (iv) the result of the invention. We show an example in Table 5.

D Graph generation prompts

Below are the prompts

D.1 Component Extractor

<|system|> Your job is to identify relevant components among the item list and the claim segment. For example,

Input: "components": ["shuffling apparatus", "card holders", "shuffle button", "cards"], "claim": "a conveying mechanism adapted to convey cards between each card holder and at least two other card holders, the conveying mechanism controllable to select a destination card holder of the at least two card holders"

Output: "relevant": ["card holders", "cards"] <|end|>

D.2 Function finder

<|system|> Your job is to find the function of an invention component. For example,

Input: "components" : ["card holder", "cards"], "claim": "at least three card holders, each adapted to hold at least one deck of cards"

Output: "card holder": "one of at least three holders for shuffled or unshuffled decks", "cards": "playing cards for games that need to be shuffled" <|end|>

D.3 Relation Finder

<|system|> Your job is to determine relationships between components in patent claims based on a dictionary of components.

For example:
Input: "light bulb" : "composed of a filament that illuminates surroundings", "ceiling fan" : "cools surroundings", "switch": "sends an electrical signal to light bulb and gearbox", "gearbox": "drives a the fan speed", "filament": "glows by electrical stimulation"

Output: ("switch", "causes electricity to flow to and turns on", "light bulb") ("gearbox","controls the rotation speed of", "ceiling fan") ("filament", "is part of", "light bulb") <|end|>

Title: "Shade bracket with concealed wiring" **Abstract:** "A bracket is configured to be coupled to a support surface ... such that the bracket extends away from the ... to power a motor of the roller window shade assembly ... when the bracket is coupled ... and supports the roller window shade assembly."
Query 1: What is the essential function of the product or process? **Answer:** The essential function is to securely mount a roller window shade assembly to a support surface, providing a hidden pathway for the electrical wiring that powers the shade's motor.
Query 2: How does the product or process operate? **Answer:** It operates by securely mounting ... passage within the bracket allows electrical wiring to pass ... powering it while remaining hidden.
Query 3: What results does this product or process achieve? **Answer:** Secure mounting, hidden electrical wiring, and limiting rotation for stability.
Query 4: What ingredients, materials, or processes are alternatives? **Answer:** Alternatives include metals (e.g., aluminum), plastics, PVC, and adjustable arm designs for projection limitation.

Table 5: Example of queries and their abbreviated answers generated by Phi-3.5-MoE