CLAIM: A Tool for Analyzing Legal Documents

Georgios Koutsopoulos[1,*], Noran Kniby[1], Pauline Wiman[1], Ingrida Karina-Berzina[2] and Janis Stirna[1]

[1]Department of Computer and Systems Sciences, Stockholm University, Borgarfjordsgatan 12, Kista, 16407, Stockholm, Sweden
[2]Department of Law, Stockholm University, Universitetsvägen 10 C, 10691, Stockholm, Sweden

Abstract

In alignment with the ongoing digitalization in all aspects of science and society, the legal domain is experiencing a significant growth in the volume of produced documents. Manual approaches for analysis of legal corpora, like Intellectual Property case decisions, are deemed resource-intensive, in terms of time and effort. As a response, we present the CLAIM tool, which has been developed combining Model-Driven Development with a Large Language Model. As a Design Science Research project, the tool's development has been based on the needs of its stakeholders, which have been elicited in participatory modeling workshops. The prototype version presented in this paper includes two main phases of analysis; information extraction and semantic information retrieval. Its functionality is briefly demonstrated by analyzing trademark cancellation cases which have been filed in bad faith.

Keywords

Conceptual modeling, Intellectual Property, Legal analysis, Model-Driven Development, Large Language Model, ChatGPT

1. Introduction

Over the last decades, society is undergoing a digitalization that results in a significant growth of produced documents [1]. The legal domain is no exception; the volume of legal documents is consistently on the rise. Until the 1950s, research and practice in the law relied entirely on manual extraction of the substantive information included in legal documents. The term legal documents refers to cases, statutes, regulations, contracts, and any other type of text document that is used by professionals of the legal domain [2]. In practice, even during analyses facilitated by computers in recent years, a human expert was necessary for reading the document, identifying, and representing the essential parts of the text in a format usable by computers [2]. Technical constraints also have limited the very nature of legal research that is feasible for legal researchers and practitioners, most of whom lack a programming background. For this reason, empirical research analyzing large sets of legal documents has been far overshadowed by traditional doctrinal research analyzing a small set of keystone cases, thus, saving time and effort [3].

A specific area within Law that is affected by this situation is the area of Intellectual Property (IP) law [4]. A large volume of patent, trademark, and copyright judicial decisions is available to practitioners and researchers. Yet, even if analyzing large corpora of IP decisions has the potential to reveal deep insight, for example, similarities and overarching trends in case outcomes in a way not possible by reading a subset of cases, the impracticality of manual handling hinders the procedure. In this way, valuable insight may remain hidden from researchers and practitioners. This situation presents not only a challenge, but also an opportunity for improvements, which can be based on various approaches.

In: Janis Grabis, Yves Wautelet, Emanuele Laurenzi, Hans-Friedrich Witschel, Peter Haase, Marco Montali, Cristina Cabanillas, Andrea Marrella, Manuel Resinas, Karolin Winter. Joint Proceedings of HybridAIMS and CAI Workshops. Co-located with CAiSE 2025.

*Corresponding author.
georgios@dsv.su.se (G. Koutsopoulos); nkniby@proton.me (N. Kniby); pauline.wiman@dsv.su.se (P. Wiman); ingrida.karina-berzina@juridicum.su.se (I. Karina-Berzina); js@dsv.su.se (J. Stirna)
0000-0003-2511-9086 (G. Koutsopoulos); 0009-0000-8050-1507 (N. Kniby); 0009-0005-5412-7386 (P. Wiman); 0009-0009-5258-3512 (I. Karina-Berzina); 0000-0002-3669-832X (J. Stirna)

Regarding the required time, Large Language Models (LLMs), and especially the state-of-the-art versions, have demonstrated a high level of efficiency on performing tasks without the need for task-specific training [5]. Regarding some legal tasks, their efficiency and accuracy can reach a degree that is not only comparable, but often superior to human experts [6]. Therefore, utilizing LLMs and Artificial Intelligence (AI) technology in general, is an approach that can provide automation in the analysis procedure. In this way, precious time and effort can be saved, and in certain cases, this can be a critical factor that determines the feasibility of a legal analysis. Regarding the required effort, it can be affected by reducing the complexity of the task. Conceptual modeling (CM), and its specialization, Enterprise Modeling (EM) [7], can assist by providing a solid information structure that points towards the right directions, either methodologically, or as a blueprint for developing a supporting system.

The aim of this paper is to introduce the CLAIM tool (Case-based Legal Analysis of Intellectual property Matters), a domain-specific prototype solution for IP decision analysis that combines the two approaches, LLMs and EM. The tool is described and demonstrated in a specific analysis of “bad faith” [4] IP decisions.

The rest of the paper is structured as follows. Section 2 provides a brief overview of the theoretical background that is related to this study, Section 3 describes the involved methodological decisions, Section 4 presents the CLAIM tool. Sections 5 and 6 discuss the study and provide concluding remarks, respectively.

2. Background

This section provides a brief background on the topics that are relevant to this study.

2.1. Intellectual Property Law and Legal Document Analysis

IP law includes a wide spectrum of legal protections for patents, trademarks, copyrights, trade secrets and any other creations of the human mind [8]. Courts of different levels around the world produce volumes of decisions. Despite the differences in the contextual factors, like national and regional variations in legislations, the decisions are characterized by a consistent narrative pattern, consisting of a factual overview, legal reasoning, and a conclusion [4, 8]. This does not mean that the employed terminology is equally consistent. This volume of decisions was always archived, yet, during recent years, has become digitally available, a fact which enables the potential for empirical research on IP law trends [2].

The decisions are often grouped in sets, forming specific corpora that can be used during analyses. However, these corpora are often diverse and heterogeneous, using domain-specific language, case citations and statutory references that may differ by jurisdiction [9]. In addition, the outcomes of a case may be existing in a case document in unclear ways, due to reliance on complex legal argumentation. In such cases, manual analysis is deemed necessary, yet, this activity is not only highly resource-intensive but also prone to errors [10]. As a response to these challenges, modern approaches to legal analyses opt towards computational methods, in order to automate tasks like information extractions and document classification [3].

2.2. Technological Perspectives on Legal Analysis

2.2.1. Model-Driven Development

Model-Driven Development (MDD) [11] is an approach for software engineering, focused on the use of high-level abstractions, commonly known as conceptual models. CM, and its specialization, EM [7], provide the artifacts that are used during the software lifecycle. Instead of producing detailed low-level programming code, developers can define the functionalities of a system that is being developed using the models. These models are platform independent, yet, they can be implemented in any specific platform. MDD aims to reduce complexity, improve maintainability, and accelerate software development cycles

[12]. Regarding the applicability of MDD in legal analytics, where case management and contextual variations require adaptability and quick responses, MDD can potentially provide valuable advantages. The complexity of the domain can be countered by producing high-level conceptualizations that can be applied to different cases and contexts. Participatory modeling [13] can be used as a bridge between software developers, modeling experts, and legal domain experts to provide solutions that can facilitate the development of tools for legal analysis. This can be achieved by identifying and conceptualizing recursive patterns in legal forms, decisions, and judicial opinions, as a means “to routinize and scale very cheap and very high quality solutions to the myriad of legal needs” [14].

2.2.2. Large Language Models

LLMs, such ChatGPT or other similar transformer-based models, have demonstrated significant capabilities across a wide spectrum of natural language processing tasks [5]. A large set of vast and diverse corpora has been used during the training of these models. This fact results in the models having the ability to interpret legal texts, even if they contain context-specific and inconsistent terminology, and extract semantic insight from them [9], a feature which is considered highly valuable. This is applicable in cases where extensive corpora need to be processed for annotation and/or classification [6]. The identification of specific factors that drive the classification is essential for the task. However, confirming and explaining that the results produced by an LLM are bias-free and transparent can be challenging, and this fact can, in return, raise concerns and hinder the entire effort. This is reasonable, if we take into consideration that any mistakes in the legal domain may have significant impact and consequences. For this reason, integrating LLMs with an approach like MDD, has the potential to improve the trustworthiness of the results, since the reliance on LLMs is reduced.

3. Methodology

CLAIM has been developed as a Design Science Research [15] project, following the framework suggested in [16]. The framework suggests artifact development in five steps: (i) Problem explication, (ii) Requirements elicitation, (iii) Artifact development, (iv) Demonstration, and (v) Evaluation.

The explication of the problem was based on its stakeholders, that is, legal researchers and practitioners. In the case of CLAIM, the Intellectual Property Law research group of the Department of Law (Juridicum) of Stockholm University is the direct stakeholder involved in the project.

To elicit requirements, four 3-hour workshops were conducted with the participation of the development team and a representative of Juridicum. The aim was to elicit requirements in the form of conceptual models that could be used as input for the development. In practice, the workshops involved data collection using the think-aloud protocol [17] to gather insight on the context and conditions of IP case decisions, and to clarify the terminology used in the domain. Two of the workshops were conducted as participatory modeling [13] sessions. This means that the collected data was used to develop conceptual models, to be used as input during the artifact’s development. In particular, the Unified Modeling Language (UML) [18] was used to create a Class diagram representing the domain.

Regarding the development of the artifact, the two approaches mentioned earlier, LLMs and EM, as part of MDD, have been combined. CLAIM has been built as an application using Node.js [19] for the backend combined with HTML/CSS/Javascript protocols. MariaDB [20] has been used for the database. These have been combined with OpenAI’s ChatGPT-4o via their APIs. Node.js has been the basis on which the architecture and logic of the application has been built, while ChatGPT has been used for its semantic analysis capabilities. In particular, from a Semantics-driven Engineering perspective, ChatGPT, in the CLAIM project, contributes to the (i) knowledge capturing role, and (ii) end user-facing role [21]. Finally, for the demonstration of the CLAIM tool, a case study has been employed. The case is the analysis of European Union Intellectual Property Office (EUIPO) case decisions related to trademarks cancelled because they were filed in bad faith. More details follow on the next sections. Regarding the evaluation of the tool, hitherto, there is no completed formal assessment to report, yet, an evaluation by stakeholders from Juridicum is planned to take place.

4. The Tool

In this section, the architecture and the main features of the CLAIM tool are presented.

4.1. Architecture

The domain of Intellectual Property claims has been modeled according to the information elicited during the workshops. The produced information structure is shown in the UML model of Fig. 1 and explained as follows. Every IP Case (referring to a cancellation action initiated against a registered trademark) is about one or more Trademarks, and is always filed as a Document. One or more assessment Factors may be derived from one or more Documents, and these Factors may then be used in one or more Assessments of Case Documents. This part of the model establishes the foundation of a case-based analysis, that is, older cases help the assessment of new ones. The case-based nature of the tool has been included in its name, to make this fact clear.

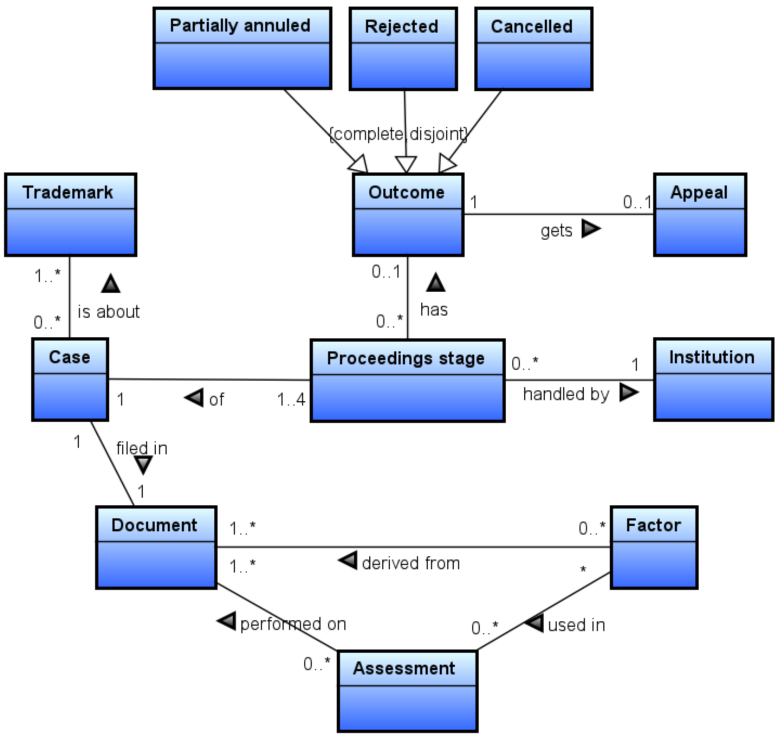

Figure 1: The domain model for the IP case procedures.

In the rest of the model, the Case is associated to at least one Proceedings stage, with a maximum of four possible stages per Case. Every Proceedings stage is handled by exactly one Institution, and may have up to one Outcome, that is the decision. Every Outcome may get up to one Appeal. Outcome is modeled as a generalization, with its specialization classes being Cancelled (referring to the trademark registration), Rejected (referring to the case), and Partially annulled (also referring to the trademark). These specializations are all the possible outcomes, and cannot be overlapping.

The domain model has been implemented as the database in MariaDB, allowing the capturing of all the essential elements for documenting all different types of cases, the information that is relevant to each case, along with the associations among the information objects. An essential element in CLAIM is the connection between the domain model and the data extraction. For each object derived from model concepts, the LLM is instructed to locate and identify the associated concepts and their instances in the text. The set of objects is stored in the database as an instance of the domain model.

Besides these core relationships, the tool also captures a case's evolving status before a final decision is reached. In practice, a case is initially filed, goes through an evaluation phase, and is being processed, potentially at different judicial levels; Cancellation division, Court of Appeals, General Court, or the Court of Justice of the EU, depending on the number of appeals so far. Once a decision is reached, the case may continue with appeals, up to three times, or be finalized if no appeal occurs. Combining these dynamic states with the domain model ensures that the complex procedural aspects of IP cases, like a chain of appeals and multiple possible outcomes are reflected in CLAIM's data structure.

By capturing both the domain's concepts and its procedural aspects, CLAIM ensures comprehensive case documentation, retrieving relevant factors, and tracking legal proceedings over several institutional levels.

4.2. Processing Phases

CLAIM has two main phases of processing, which are discussed in this section.

4.2.1. Corpus Processing: Information Extraction

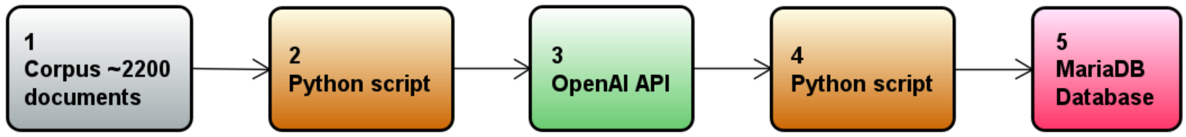

Figure 2: Corpus processing in CLAIM.

Fig. 2 shows the corpus procedure, a description of which follows.

1. All documents in the corpus are converted to txt files.
2. The OpenAI API is connected to the Python script. Each file gets analyzed based on the prompt given in the script.
3. The results from the analysis are inserted into the MariaDB database using the Python script.

In this phase, the factors objects that have been derived from the Domain model are used as a basis for the extraction of results with semantic consistency, along with the rest of the information that has been included in the model.

4.2.2. Prompt Processing: Semantic Legal Information Retrieval

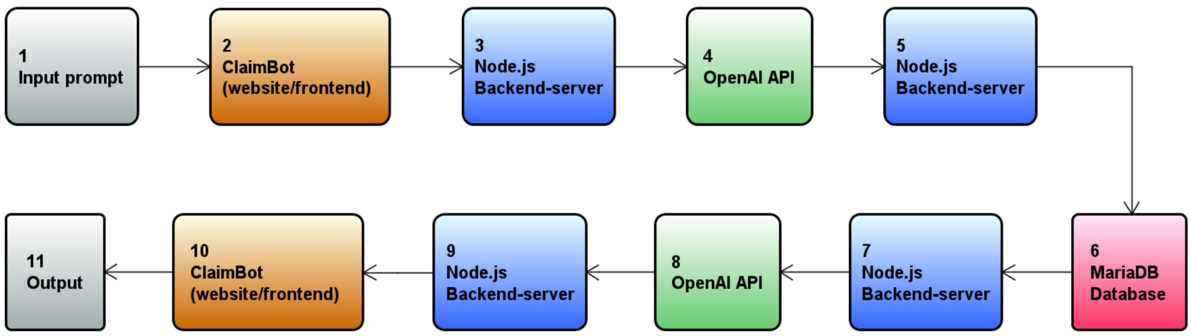

Figure 3: Prompt processing in CLAIM.

The prompt procedure is depicted in Fig. 3, and its description follows.

1. A prompt is provided by the user.

2. The frontend receives the prompt and sends it to the backend-server.
3. The prompt is saved in the backend-server.
4. The backend-server makes an OpenAI API-call with the prompt (and asks for it to translate it from natural language to SQL).
5. The backend-server receives the SQL query from OpenAI and saves it.
6. The query is sent to the MariaDB database.
7. The backend-server receives the results from the database.
8. The backend-server sends the results with instructions to the OpenAI API.
9. The backend-server receives the output and sends it to the frontend.
10. The frontend displays the output results.
11. The user can see the results as charts in a report.

In order to avoid re-inventing the wheel, we have opted for using the integrated capabilities of ChatGPT for semantic legal information retrieval. Our current retrieval process uses a mapping from user queries to SQL statements, which are generated by the LLM. Unlike Retrieval-Augmented Generation (RAG) [22] where entire documents are used, we rely on the domain factor objects which have been previously extracted and stored in the database. Essentially, the current version of the tool is focused on performing content analysis [23], in other words, the final results are usually quantified data derived from counting frequencies in qualitative data.

4.3. User Interface

The tool's main User Interface (UI) is shown in Fig. 4. We opted for a minimally designed simple interface that consists of an input field, along with two default prompts placed right above the input field. They include suggested questions and/or tasks that aim to drive the user's interaction with the tool. For example, a default prompt may be asking about the proceedings stage where cancellations most commonly occur. Such prompts not only promote the CLAIM tool's capabilities, but may also potentially inspire users.

4.4. A Brief Demonstration

For a brief demonstration, we rely on a specific corpus that consists of the EUIPO decisions under Article 59(1)(b) or Article 52(1)(b). The issue concerns IP decisions on cases where trademark cancellations have been the issue, and in particular, cases that have been filed in bad faith. Bad faith is a term referring to improper and/or dishonest intents while creating, using, or enforcing IP rights, for example intent to deceive, or exploit another party's good will, or even fraudulent representations [24]. The access to the corpus has been provided to us via EUIPO, upon request.

The case-based nature of CLAIM required the identification of factors for assessing the IP case decisions. The Factor objects that have been identified and stored in the MariaDB database formed an extensive list, so we are only providing here a few examples:

- ”Identity or similarity of the signs”
- ”Dishonest intention”
- ”Free-riding on reputation”
- ”Pre-existing relationship abused”

Being semantics-driven, the information retrieval does not rely on the identification of the exact key phrases, on the contrary, any semantically consistent argument is valid. An example question that has been posed on the specific corpus, and analyzed using the existing factors is “What percentage of cancellation actions filed ultimately results in the cancellation of a mark?”. CLAIM identifies the proceedings stages, according to the Domain model and retrieves the required information and performs the simple calculation. In this way, it produces and reports the response to the question, being 12.5%, as shown in Fig. 4.

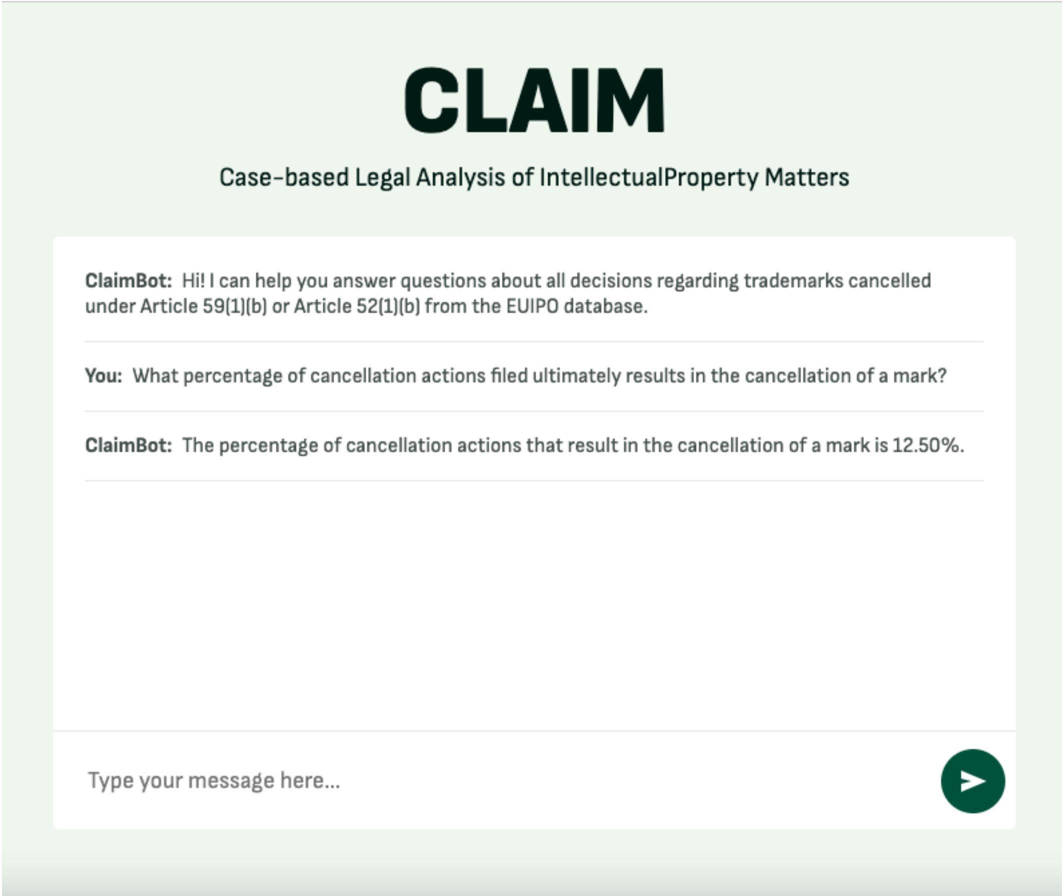

Figure 4: An example prompt and response in CLAIM.

5. Discussion

Combining MDD and LLM functionalities has the potential to address practical challenges in the analysis of legal documents. To our knowledge, there is no approach that combines MDD with LLMs for legal analysis in the IP domain. In particular, the LLM-driven information extraction in an IP domain model can be considered as novelty. Despite the fact that CLAIM is an early prototype with a lot of room for improvements, the combination of AI with conceptual modeling has shown its potentials. These potentials are maximized when complex domains are concerned, like the legal domain, and more specifically, the IP law domain. Addressing complexity with conceptual modeling and resource-intensiveness with AI, provides the means to tackle a challenging issue. We aspire that this study can motivate additional efforts, not only in the Law, but in any other equally challenging domain.

However, since CLAIM is an early prototype, a series of future iterations of development are expected. There is a variety of possible steps for future versions of the tool. Initially, a step which is already planned concerns the refinement of the reported results. Pattern recognition will be used to identify potential inconsistencies between factors in cases, or between factors and outcomes. In such instances, we plan to adjust the tool for "flagging" these possibly anomalous results for manual analysis by a human user. This will minimize the risks of borderline cases being analyzed incorrectly, due to conflicting or omitted factors. For example, if a legal document includes several factors that would commonly lead to rejecting a case, yet, the trademark is cancelled, or vice versa, the tool should flag the case for potential mistakes and ask for a manual check from the user.

Exploring the use of RAG and embedding-based queries is also under consideration. Our current model-driven factor-based relational schema has been chosen for clarity and ease, since factors are easier to interpret by our stakeholders. However, we acknowledge the potentials for enhancing text retrieval of the above-mentioned approaches.

An additional future step concerns the automatic construction of corpora from specific databases, according to the users' expressed requests. For example, applying filters in a database, downloading files, and grouping them can be a future functionality of the tool.

Finally, a systematic evaluation is expected to follow soon, as a means to capture a structured list of potential improvements, including their feasibility and priority, so that a solid plan for future development iterations is elicited. This concerns the quality of the results. From a task improvement perspective, the stakeholders have already expressed the opinion that the performed task would not be feasible manually. In other words, CLAIM, even in its current version, is considered a valuable asset.

6. Conclusions

This paper introduced CLAIM, a prototype legal analysis tool that has been developed by combining MDD and LLM functionalities. The combination of conceptual modeling, in particular, UML domain modeling with ChatGPT's semantic analysis capabilities, reduces the required time and effort for interpreting complex and extensive legal documents. Even as an early prototype, it has shown great promise in expanding the sophistication of analysis that is possible using large corpora of legal documents. Future work aims to refine the anomaly detection features, automate corpus creation, and assess the results empirically in diverse legal contexts.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] M. Hilbert, P. López, The World's Technological Capacity to Store, Communicate, and Compute Information, Science 332 (2011) 60–65. URL: https://www.science.org/doi/10.1126/science.1200970. doi:10.1126/science.1200970.

[2] K. D. Ashley, Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age, 1 ed., Cambridge University Press, 2017. URL: https://www.cambridge.org/core/product/identifier/9781316761380/type/book. doi:10.1017/9781316761380.

[3] M. A. Hall, R. F. Wright, Systematic Content Analysis of Judicial Opinions, California Law Rev. 96 (2008) 63–122.

[4] W. Cornish, D. Llewelyn, T. Aplin, Intellectual property: patents, copyright, trade marks and allied rights, Classics series, ninth edition ed., Sweet & Maxwell, Thomson Reuters, London, 2019.

[5] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language Models are Few-Shot Learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[6] M. Gray, J. Savelka, W. Oliver, K. Ashley, Can GPT Alleviate the Burden of Annotation?, in: G. Sileno, J. Spanakis, G. Van Dijck (Eds.), Frontiers in Artificial Intelligence and Applications, IOS Press, 2023. URL: https://ebooks.iospress.nl/doi/10.3233/FAIA230961. doi:10.3233/FAIA230961.

[7] K. Sandkuhl, J. Stirna, A. Persson, M. Wißotzki, Enterprise Modeling: Tackling Business Challenges with the 4EM Method, The Enterprise Engineering Series, Springer, Berlin, Heidelberg, 2014. URL: http://link.springer.com/10.1007/978-3-662-43725-4. doi:10.1007/978-3-662-43725-4.

[8] L. Bently, B. Sherman, D. Gangjee, P. Johnson, Intellectual property law, Law trove, sixth edition ed., Oxford University Press, Oxford, 2022. doi:10.1093/he/9780198869917.001.0001.

[9] M.-F. Moens, Information extraction: algorithms and prospects in a retrieval context, number 21 in Information retrieval series, Springer, Dordrecht [Great Britain], 2006.
[10] H. Surden, Machine Learning and Law, Washington Law Review 89 (2014). URL: https://papers.ssrn.com/abstract=2417415.
[11] R. France, B. Rumpe, Model-driven Development of Complex Software: A Research Roadmap, in: Future of Software Engineering (FOSE '07), IEEE, Minneapolis, MN, USA, 2007, pp. 37–54. URL: http://ieeexplore.ieee.org/document/4221611/. doi:10.1109/FOSE.2007.14.
[12] D. Schmidt, Guest Editor's Introduction: Model-Driven Engineering, Computer 39 (2006) 25–31. URL: http://ieeexplore.ieee.org/document/1597083/. doi:10.1109/MC.2006.58.
[13] J. Stirna, A. Persson, Enterprise Modeling: Facilitating the Process and the People, Springer International Publishing, Cham, 2018. URL: http://link.springer.com/10.1007/978-3-319-94857-7. doi:10.1007/978-3-319-94857-7.
[14] W. Henderson, A Blueprint for Change, Pepperdine Law Review 40 (2013). URL: https://digitalcommons.pepperdine.edu/plr/vol40/iss2/7.
[15] A. Hevner, S. Chatterjee, Design Research in Information Systems, volume 22 of *Integrated Series in Information Systems*, Springer US, Boston, MA, 2010. URL: http://link.springer.com/10.1007/978-1-4419-5653-8. doi:10.1007/978-1-4419-5653-8.
[16] P. Johannesson, E. Perjons, An Introduction to Design Science, Springer International Publishing, Cham, 2014. URL: http://link.springer.com/10.1007/978-3-319-10632-8. doi:10.1007/978-3-319-10632-8.
[17] E. Charters, The Use of Think-aloud Methods in Qualitative Research An Introduction to Think-aloud Methods, Brock Education Journal 12 (2003). URL: https://journals.library.brocku.ca/brocked/index.php/home/article/view/38. doi:10.26522/brocked.v12i2.38, section: Articles.
[18] Object Management Group (OMG), OMG® Unified Modeling Language®, 2017. URL: https://www.omg.org/spec/UML/2.5.1/PDF.
[19] Node.js — Run JavaScript Everywhere, ????. URL: https://nodejs.org/en.
[20] MariaDB Foundation - MariaDB.org, ????. URL: https://mariadb.org/.
[21] R. Buchmann, J. Eder, H.-G. Fill, U. Frank, D. Karagiannis, E. Laurenzi, J. Mylopoulos, D. Plexousakis, M. Y. Santos, Large language models: Expectations for semantics-driven systems engineering, Data & Knowledge Engineering 152 (2024) 102324. URL: https://linkinghub.elsevier.com/retrieve/pii/S0169023X2400048X. doi:10.1016/j.datak.2024.102324.
[22] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, in: H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 9459–9474. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.
[23] K. A. Neuendorf, The Content Analysis Guidebook, SAGE Publications, Inc, 2455 Teller Road, Thousand Oaks California 91320, 2017. URL: https://methods.sagepub.com/book/the-content-analysis-guidebook-2e. doi:10.4135/9781071802878.
[24] European Parliament, Council of the European Union, Regulation (EU) 2017/1001 of the European Parliament and of the Council of 14 June 2017 on the European Union trade mark (codification) (Text with EEA relevance), 2017. URL: http://data.europa.eu/eli/reg/2017/1001/oj/eng, legislative Body: EP, CONSIL.