*Legal Knowledge and Information Systems*
*E. Francesconi et al. (Eds.)*


*doi:10.3233/FAIA220449*

# Semantic Querying of Knowledge Rich Legal Digital Libraries Using Prism[1]

Hasan JAMIL [a,2]
[a] *Department of Computer Science, University of Idaho, USA*
ORCiD ID: Hasan Jamil https://orcid.org/0000-0002-3124-3780

**Abstract.** Contemporary legal digital libraries such as Lexis Nexis and WestLaw allow users to search case laws using sophisticated search tools. At its core, various forms of keyword search and indexing are used to find documents of interest. While newer search engines leveraging semantic technologies such as knowledgebases, natural language processing, and knowledge graphs are becoming available, legal databases are yet to take advantage of them fully. In this paper, we introduce an experimental legal document search engine, called *Prism*, that is capable of supporting legal argument based search to support legal claims.

**Keywords.** Document Engineering, Story Understanding, Premise Graph, Knowledge Graphs, Graph Matching, Natural Language Processing.

## 1. Introduction

Search is a basic function supported in all digital archives of information. While search techniques have evolved in structured and unstructured databases, it is still an ongoing research issue in digital libraries and document databases [10], in stark contrast with other types of digital libraries such as music [6], mathematics [14], judicial [4], etc. in terms of techniques and applications. Among the search techniques currently in use, some form of keyword search [17] or text mining based association search [1] are prevalent. In recent research, however, an emerging trend of searching digital libraries using knowledge graphs (KG) is gaining popularity [8] with the goal to improve semantic matching [9].

It is not uncommon in digital library search for users to land on useful documents almost by accident [13]. This is because most of the search engines do not allow queries that make sense semantically. For example, the following legal query

> $Q_1$: *"List case laws where parents retained jurisdiction in Virginia despite the opposing parent having the home state jurisdiction in another state under UCCJEA."*

is unlikely to return any case laws that meet the exact legal criteria expressed in the query. Most likely a keyword search will return all Michigan cases under UCCJEA mentioning

[1]This publication was partially made possible by an Institutional Development Award (IDeA) from the National Institute of General Medical Sciences of the National Institutes of Health under Grant #P20GM103408.
[2]Corresponding Author: Hasan Jamil, Department of Computer Science, University of Idaho, USA. Email: jamil@uidaho.edu.

home state and nothing much. To appreciate the complexities inherent in this query, it is important that we understand the structure of legal briefs, and the USA UCCJEA law.

### *1.1. Structure of Case Laws*

Roughly, there are two types of laws – black letter law, or the articles of the constitution and case laws, or the adjudicated proceedings of the cases in the courts of law. Case laws are specific litigation in which black letter laws and other case laws, called legal precedents, are applied and the legal merits of opposing arguments are decided by the courts. In the USA, we have a three tier court system – 1) trial court where litigation first starts, merit is decided based on facts, and logic of the arguments by direct application of the laws; 2) appeals court where constitutionality of decisions made by trial courts is challenged and decided relative to the case at hand; and finally 3) the supreme court which adjudicates any misinterpretation of the laws by the lower courts.

With respect to digital representation of judicial documents, counsels of the parties involved in a litigation submit legal briefs, courts rule on the briefs, and produce another document called ruling or judgment. These rulings become legal precedents and enter the database as case laws. A ruling has roughly four parts: 1) court, party, counsel details, and case details such as case number, dates and jurisdiction. 2) a preamble that states the overall description of the litigation and applicable case laws. 3) facts that lay out the ”truth” about the case as seen by each party which can be established by evidence. 4) legal argument why or why not the facts lead to legal conclusions supported by case laws. Finally, 5) relief sought or final opinion of the courts after deliberations and argument.

### *1.2. Article UCCJEA*

UCCJEA stands for Uniform Child Custody Jurisdiction and Enforcement Act (we refer the readers to [3] for a detailed exposition to UCCJEA provisions in Virginia), which is an article of USA federal constitution dealing with jurisdiction of litigating parents or custodians of minor children residing in distinct court jurisdictions, often in multiple states. Some versions of the UCCJEA act has been adopted by all 50 states.

#### *1.2.1. Essence of UCCJEA*

Two of the main purposes of the article were to (i) stop competing jurisdictions from abusing their power and forcing families to needlessly waste resources in multiple states by usurping the jurisdiction from a state having a legitimate claim on the jurisdiction, and (ii) prevent parents from seeking a more convenient forum across state boundaries to make it difficult for the other parent to seek relief from a rightful jurisdiction or to frustrate them. The article does so in many ways but mainly by (i) removing the use of “best interest of the child” argument from the clauses of UCCJA (the predecessor of UCCJEA), (ii) prioritizing the jurisdictional bases in a tie proof hierarchy so that a state/court at a higher strata can assume jurisdiction, (iii) allowing a state at a higher strata exercise jurisdiction without any regard for a court in a competing jurisdiction at a lower strata in the absence of an exclusive continuing jurisdiction by another court, (this sounds confusing) (iv) requiring the courts to evaluate their jurisdictional authority anew (even if the court has exclusive continuing jurisdiction) every single time a new cause is brought before them by recognizing the fact that jurisdiction of any court is not

permanent and may change primarily due to the mobility of the child, and (v) giving the home state the absolute priority and preemptive jurisdiction over all courts again in the absence of exclusive continuing jurisdiction by another court.

### *1.3. Use Case*

Abebi and Pierre had a child named Fiia when they divorced in a Mississippi court at which time they agreed to joint custody of Fiia. Subsequently, both Abebi and Pierre moved to Virginia and Michigan respectively, and Fiia moved to Michigan according to the terms of the Mississippi court order which granted each parent a two year primary rotational custodianship. However, upon moving to Virginia, Abebi filed for sole primary custody in Virginia and a jurisdictional litigation ensued involving three different states. The primary question being debated was which court has jurisdiction over Fiia, and where this custody matter will be decided.

On pleading with Virginia by Abebi, the court assumed jurisdiction despite objection that Michigan was Fiia's home state and Virginia did not have jurisdiction to make an initial determination. Furthermore, Mississippi still had exclusive continuing jurisdiction and did not decline to exercise jurisdiction. Michigan, on the other hand, deferred to Virginia stating "since" Virginia is exercising jurisdiction, it (MI) cannot despite having home state jurisdiction, and despite the UCCJEA stating that Michigan is not required to extend full faith and credit to Virginia court because they did not have the jurisdiction in the first place.

With the intention of appealing the two decisions in Virginia as well as Michigan courts, Pierre is looking for case laws that show precedent supporting Virginia's stance, and then researching if that erroneous decision was reversed by superior courts in Virginia . In fact, there are plenty of case laws that refute the Virginia and Michigan position in the case of Abebi v Pierre in favor of Pierre that existing legal search engines cannot find or link multiple case laws to offer a more complete picture.

For example, the Janet Miller-Jenkins v. Lisa Miller-Jenkins, Virginia (2006)[3] case is almost exactly identical to Abebi v. Pierre and supports Pierre's position, which was denied by both Virginia and Michigan, Janet and Lisa lived together in Virginia in the 1990's where Lisa gave birth to their daughter IMJ in April 2002. Soon after in August 2002 they moved to Vermont and entered into a civil union. Unfortunately, in September 2003, the parties ended their relationship and Lisa moved to Virginia with IMJ while Janet remained in Vermont.

Lisa asked a court in Vermont to dissolve their union in November 2003 and sought legal and physical custody of IMJ. In June 2004, Vermont issued a temporary order awarding Lisa primary custody. On July 1, 2004 Lisa filed for sole custody with sole parental rights in Virginia upon Virginia's affirmation of Marriage Affirmation Act. Upon learning the Virginia action, the Vermont court on July 19, 2004, exercised its exclusive continuing jurisdiction, stating that it will not defer to Virginia and ordered that its previous custody order be followed. When the Virginia courts proceeded with the litigation in Virginia, the court of Vermont forcefully ignored all Virginia orders holding that Virginia lacked subject matter jurisdiction and retained its (Vermont's) right to exercise jurisdiction. The Vermont Supreme court held that the state acted according to its established law, had jurisdiction to do so and that Parental Kidnapping Prevention Act (PKPA)

[3] http://www.courts.state.va.us/opinions/opncavwp/2654044.pdf

afforded preemptive jurisdiction to Vermont and denied full faith and credit to Virginia orders that contradicted those entered by the Vermont court. On appeal in Virginia, the Virginia court of appeals affirmed and upheld Vermont's position.

Both Vermont and Virginia appeals court's positions have been affirmed in several other similar courts such as in Rogers v. Rogers, Alaska (1995)[4], Swalef v. Anderson, Virginia (2007)[5], Key v. Key, Virginia (2004)[6], and in numerous other cases. In particular, in Markle v. Markle, Michigan (2007)[7], the Michigan court of appeals denied to extend full faith and credit to Texas court's custody order citing Texas court's lack of subject matter jurisdiction. In Johnson v. Johnson, Michigan (2005)[8], the Michigan court of appeals reversed the Michigan trial court's order that denied Michigan jurisdiction in favor of Idaho without determining that Michigan was an inconvenient forum by simply determining that Michigan lacked jurisdiction under a scenario similar to that of Abebi v. Pierre case even though Michigan was the home state.

The critical point is that none of these cases were systematically discovered using existing search engines in law libraries; rather, they were accidentally discovered [13] on the internet by Pierre. The research issue we are addressing is designing a search engine to help find cases that support or refute the position of a plaintiff or defendant given the case facts as the user sees it. We call the reasoning a user uses to establish a claim a *premise graph*. The task the search engine assumes is to find the cases that at least partially match the edges in the premise graph, and possibly all the edges to render a conclusion. We lay out our experimental model in the sections to follow.

## 2. Document Understanding using *Prism*

Abebi v. Pierre illustrates a complex system of information structure that most likely will not lend itself to traditional query engines such as keyword search, layered indexing, and other techniques discussed earlier, to produce the documents these litigants seek. More novel approaches based on knowledge graphs [5, 7, 11] or knowledge driven querying of digital documents [2, 12] were not shown to be effective in the type of search we are interested in. We therefore propose a document authoring and engineering model to enrich legal documents with meta-information at creation time so that improved semantic search becomes possible. Our goal is to make the enrichment steps as user transparent as possible.

A careful examination of the UCCJEA black letter law suggests a premise-conclusion relationship in the form a logic structure $\alpha \leftarrow \beta_1, \wedge, \ldots, \wedge\beta_m$, where $\beta_i$s are the conjuncts in the antecedent and $\alpha$ is the consequent of a logical implication. For example, for the following facts,

resident(pierre,fiia,michigan).
jurisdiction(Cust,Subject,State,homestate) ← resident(Cust,Subject,State),
¬ jurisdiction(Cust,Subject,State,exclusivecontinuing).
jurisdiction(Cust,Subject,State, homestate) ← resident(Cust,Subject,State),

[4] http://touchngo.com/sp/html/sp-4293.htm
[5] http://www.courts.state.va.us/opinions/opncavwp/2510061.pdf
[6] http://www.courts.state.va.us/opinions/opncavwp/1079041.pdf
[7] https://www.michbar.org/opinions/appeals/2007/081407/36789.pdf
[8] http://www.michbar.org/opinions/appeals/2005/030105/26467.pdf

jurisdiction(Cust,Subject,State,exclusivecontinuing),
declined(Cust,Subject,State,exclusivecontinuing).
jurisdiction(Cust,Subject,State, convenientforum) ←
resident(Cust,Subject,State), deferred(Cust,Subject,State,ExState)
jurisdiction(Cust,Subject,ExState,exclusivecontinuing).

the above rules codifying Home State jurisdiction under UCCJEA will determine that Pierre, as a custodian of Fiia, has home state jurisdiction in Michigan. However, if we add this fact to the database,

jurisdiction(pierre,fiia,mississippi,exclusivecontinuing).

Pierre will not gain home state jurisdiction in Michigan. This rule base then can act as a recommendation system to suggest Pierre to seek a convenient forum determination, or home state deferral by the state of Mississippi.

Some of the facts claimed in the legal briefs or pleadings are subject to dispute, and a ruling is necessary. For example, in Miller-Jenkins v. Miller-Jenkins, Virginia (2006), as well as in Abebi v. Pierre, Michigan (2009), both Lisa and Abebi claimed home state jurisdiction. In Lisa's case, home state was obvious since IMJ lived with Lisa in Virgina for more than six months. Lisa could not exercise the home state jurisdiction because Vermont was exercising its exclusive continuing jurisdiction, which takes precedent under UCCJEA. However, for Abebi, Fiia lived in Virginia for two weeks, after moving from Mississippi, and then lived with Pierre for more than four months at the time Abebi filed for custody. In such cases, both parties need to state why they believe their respective states have jurisdiction. A judge then decides the correct status based on case laws, which is clearly spelled out in the UCCJEA article. We can capture the premises for residency as the following set of rules.

resident(Cust,Subject,State) ← livedin(Cust,Subject,State,From,To),
duration(Days,From,To), filed(Date), Date=To, Days>183.
resident(Cust1,Subject,State1) ← livedin(Cust1,Subject,State1,From1,To1),
livedin(Cust2,Subject,State2,From2,To2), priorto(To2,From1),
duration(Days1,From1,To1), duration(Days2,From2,To2), filed(Date),
Date=To1, Days1>Days2.

The rules above say that on the date of filing the case, a custodian gains residency in a state if the child lived in that state six months or more continuously until the date of filing, or if the child lived in that state the most compared to the state she lived immediately prior. Note that both cannot simultaneously hold true. Now given the following facts, Pierre is certain to gain residency, i.e., home state residency.

livedin(abebi,fiia,virginia,1/1/2007,1/14/2007).
livedin(pierre,fiia,michigan,1/15/2007,5/20/2007).

The duration predicate above can be implemented as a computable function that will return the difference between two dates in number of days, and priorto as a Boolean function that returns true or false given two dates To and From if To is prior to From.

The technical issue now is, how do we arrive at these logical conclusions from a search of the available digital documents? One way to accomplish this is to design a text understanding system in ways similar to [7, 15] that is capable of deriving fact predicates, e.g., livedin or resident, from the case laws, and applying these rules to decide if a

document is relevant and meets the query conditions. In this approach, no additional manipulation of the documents will be necessary except the knowledge extraction engine. However, we can expect the search cost to be high because all documents will need to be understood and mined first for discovering the predicates. An alternative is to create these knowledge at the time of document authoring. We adopt the latter approach because it is efficient, even though slightly demanding and intrusive for users authoring the documents. We, however, contend that our document engineering approach is efficient for both creating documents, and processing queries.

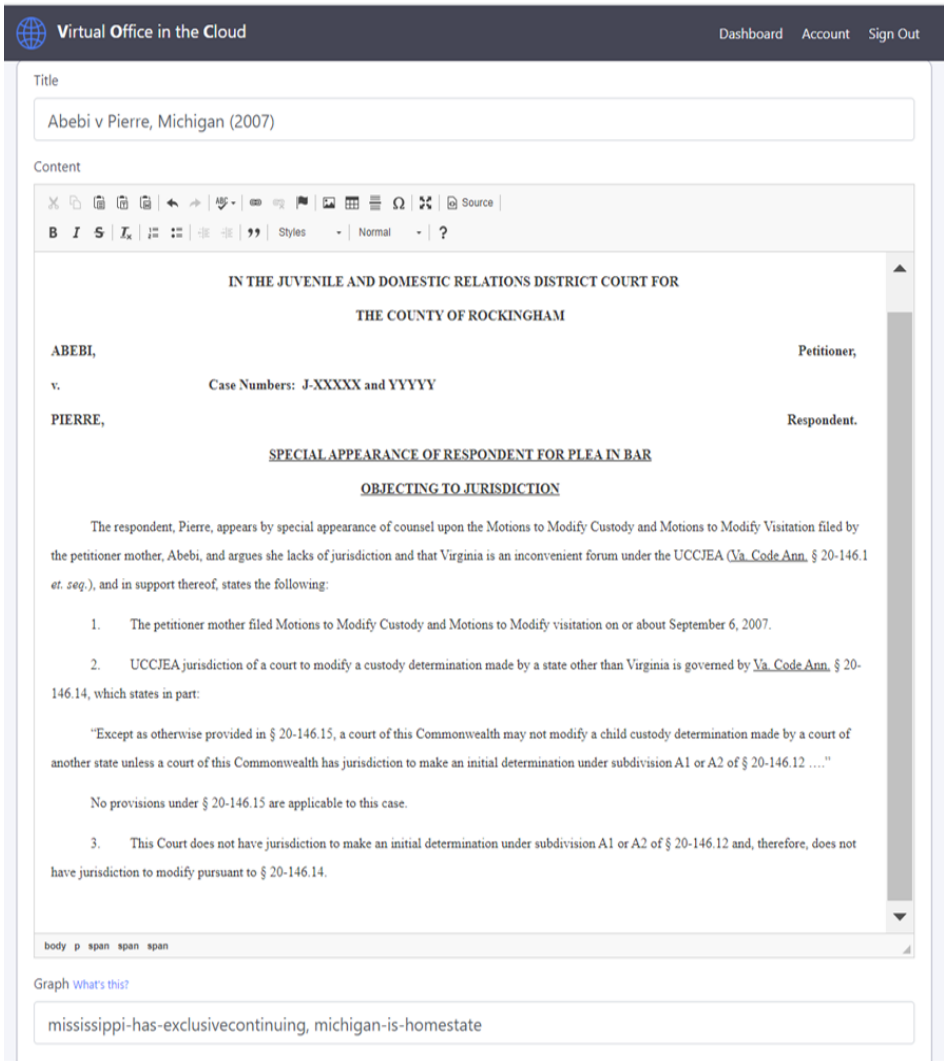



**Figure 1.** Prism user interface for document engineering with premise graph embedding.

The main idea is to design an HTML WYSIWYG legal document editor that will transparently embed a premise graph into the document as a searchable meta-data, which will not be rendered, yet the authors of the document will be able to view and edit it.

To help authors embed the graph, we design a type-ahead searchable legal terms such as *resident, exclusive continued jurisdiction, convenient forum*, etc. from which authors are able to pick node descriptions for a premise graph along with the required parameters. For example, they will be able to construct a node "Mississippi" has "exclusive continuing jurisdiction" over "Fiia" as a triple ⟨Fiia, Mississippi, Exclusive Continuing⟩ that we call *c-term*, or complex term. Subsequently, with a click of a mouse, this c-term can be added to an edge as a node, and stored as the document meta-data. Figure 1 shows the editor in use by the attorney of Pierre filing the objection to Abebi's attempt to retain jurisdiction in Virginia.

### 2.1. AND/OR Graphs

The major reasons question answering systems or legal search engines such as Lexis Nexis or WestLaw fail to respond to queries such as $Q_1$ is because they require causal reasoning or causality determination [16] which none of these contemporary digital libraries support. Since such causalities are application specific and orthogonal to document authoring, we believe they need to be addressed separately. Current approaches to such discoveries tend to be based on learning models, are quite involved and computationally expensive, in systems that support something of similar nature. In Prism, we seek to find a cheaper and more direct solution using the concept of directed AND/OR graphs that was exploited in past research [18].

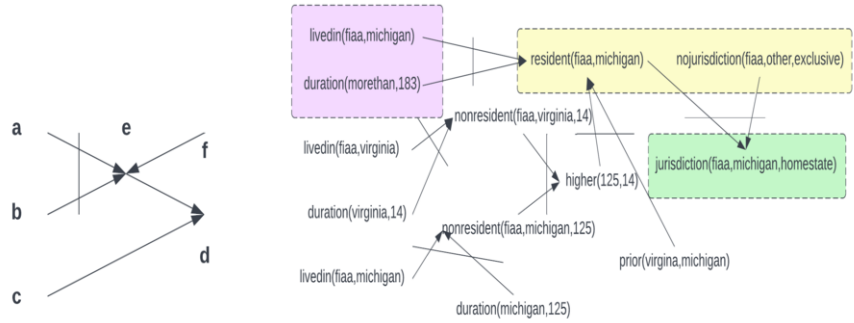
The process we have adopted to capture the premise graphs in Prism exploits the AND/OR graph representation. For example, the jurisdiction/4 rule[9] can be represented in the form of a modified AND/OR graph as shown in figure 2(b). In this modified AND/OR graph, the nodes are pre-processed and made grounded, and unlike the logic rules discussed in section 2, there are no variables. In other words, the rules are instantiated with ground facts. Users select these facts from a fixed set of terms which come with predefined slots to be filled in. For example, when the term resident is selected, the interface asks for two values, one, the name of the child, another, the state, and once supplied, generates the c-term. In order to support more complex premise graphs, Prism also allows expressing premise graphs in the form of RDF-like triples, node1-edgename-node2 type of edges, as shown in figure 1 with the document rendered and in 2(c) as the HTML document representation. Note that the premise graph is not visible to readers, yet remains visible to the author during editing.

Users are also able to visualize the premise graph before they save the document. Prism allows validation functions to check if the premise graph is legally valid, and semantically accurate. It reports mistakes using color coding of the edges. All semantically and legally accurate edges are shown in green, and the others in red. Edges being edited or not validated are shown in black.

### 2.2. Semantic Search using Premise Graphs

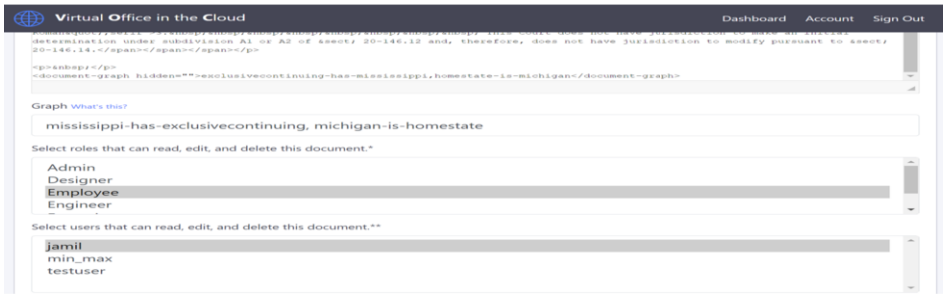
Usually, a counsel will try to find case laws that support even part of their claims. In other words, it is usually difficult to find case laws that have the exact circumstances that will warrant identical outcomes in the court of law. Given that a large percentage of cases are decided on erroneous premises and often get redressed in appeals or in supreme

[9] jurisdiction/4 means the predicate jurisdiction has four arguments.




(a) Causal net in the form of AND/OR graphs.

(b) jurisdiction/4 (home state) rules captured using AND/OR graph.




(c) Premise graphs are embedded in the document using a non-rendering mode.

**Figure 2.** Representing Premise graphs in Prism.

courts, it is not unlikely to find contradictory case laws. Therefore, case laws need to be interlinked so that the whole decision process is clear . Consequently, counsels can piece together their legal claims by citing cases that support parts of their arguments in the premise graphs, with the hope of finding such support for every part of their premise graph.

We, therefore, support a maximal constrained subgraph isomorphic matching search of the premise graphs of case laws in a systematic way. To understand the process, let us consider another case, Michelle v. Maxwell, 2002 (Nebraska) over the custody of Elli. In her case, let us assume that the case law contains the purple, yellow and green shaded parts of the premise graph shown in figure 2(b) with the following details: livedin(elli,nebraska), resident(elli,nebraska) and nojurisdiction(elli,other,exclusive) replacing the corresponding nodes in the premise graph. A search by Pierre's attorney with the entire premise graph in figure 2(b) which he intends to prove as his whole case, will match with Elli's case law since it supports "maximally" his argument that Michigan has jurisdiction over Fiia. This is because circumstances are identical to Fiia's with the only substitution being of Nebraska for Michigan, an isomorph.

On the other hand, if Prism can find another case law that supports the other branch of Pierre's premise graph, namely the non-shaded branches, Prism will include that case law as well, which only strengthens his argument even though one support is logically

sufficient. In reality, Prism will list all such matches. The important issue to note here is that Prism will also find partial matches. For example, consider a case in which Prism could only find support (matching) for the purple branch, and nothing else. In that event, Prism will list this case as a possible partial match if and only if it could not find any more case laws to support the yellow or the green shaded portions (i.e., it did not find Elli's case). This is called the maximal constrained subgraph isomorphic search – i.e., Prism always searches for maximum possible matches. Technically, Prism breaks down every AND and OR into individual subgraphs to match isomorphically, then constructs the maximal matches from the parts within the same document, and discards a match the moment a relatively more maximal match is found.

## 3. Implementation of Prism as a Virtual Office Environment in the Cloud

We have implemented Prism as a virtual legal office environment, called *VOiC*, in which document privacy and controlled sharing are top priorities. We have used Flask for its well-known support for web applications, using its two core components Werkzeug for web server functions and Jinja for HTML templating. Flask is extendable by virtue of its support for many extensions, and it also works with the majority of third-party Python libraries, which we have used as well. In addition to several other extensions of Flask, we have used the flask_ckeditor extension. CKEditor is an embeddable rich text editor with full support for HTML editing. This extension allows a core feature of Prism-HTML editing and embedding of graphs directly into documents. Bootstrap 4 open source front-end framework was used for creating platform-agnostic and responsive websites using its wide range of CSS styling options. For data management, SQLAlchemy was used to seamlessly convert data from a SQLite relational database into Python objects.

VOiC provides a comprehensive document management system – a Virtual Office in the Cloud. It consists of four main processes: storing, sharing, searching, and rendering. Storage includes the holding of users, roles, and documents in a relation database. Sharing relates to access control, and allowing users to access pertinent documents through their username and role. Searching serves users with a tool for data discovery with documents searchable by their search graph, title, and content. Rendering forms the front-end portion of VOiC. Together, these elements form a robust solution for document management in a virtual setting.

VOiC has a powerful role-based access control for document sharing. It also supports two search options – keyword or substring search, and graph search. On submitting the keywords, a SQLAlchemy query then executes and retrieves all documents in which the search query keywords are a substring in the title or content. The graph search uses the maximal isomorphic search as described earlier, and is thus a more powerful search. However, the graph search is substantially slower than text search.

## 4. Conclusion

Both VOiC and Prism are ongoing research projects to support experimentation on a new approach to document authoring, sharing and searching, and collecting enough usage data to understand the usefulness of this new digital office environment. We feel that the

approach and the technology can also be used across other scientific domains including ecology, computational biology, and network science to search for scholarly documents to discover interacting entities, such as cause-effect relationship in nature, gene regulatory networks, and so on. However, more research will be necessary to understand how the effectiveness of our system in other scientific disciplines.

## References


[1] Y. Asiri. Short text mining for classifying educational objectives and outcomes. *Comput. Syst. Sci. Eng.*, 41(1):35–50, 2022.

[2] T. Aso, T. Amagasa, and H. Kitagawa. A method for searching documents using knowledge bases. In *iiWAS2021, Linz, Austria, 29 November 2021 - 1 December 2021*, pages 250–258. ACM, 2021.

[3] V. Constitution. Uniform child custody jurisdiction and enforcement act. https://law.lis.virginia.gov/vacode/title20/chapter7.1/. Accessed: 8/18/2022.

[4] M. de Lourdes da Silveira, B. A. Ribeiro-Neto, R. de Freitas Vale, and R. T. Assumpção. Vertical searching in juridical digital libraries. In F. Sebastiani, editor, *ECIR 2003, Pisa, Italy, April 14-16, 2003,*, volume 2633 of *LNCS*, pages 491–501. Springer, 2003.

[5] J. S. Dhani, R. Bhatt, B. Ganesan, P. Sirohi, and V. Bhatnagar. Similar cases recommendation using legal knowledge graphs. *CoRR*, abs/2107.04771, 2021.

[6] B. Duggan and B. O'Shea. Tunepal: Searching a digital library of traditional music scores. *OCLC Syst. Serv.*, 27(4):284–297, 2011.

[7] E. Filtz. Building and processing a knowledge-graph for legal data. In *ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part II*, volume 10250 of *LNCS*, pages 184–194, 2017.

[8] G. Heidari, A. Ramadan, M. Stocker, and S. Auer. Leveraging a federation of knowledge graphs to improve faceted search in digital libraries. In *TPDL 2021, Virtual Event, September 13-17, 2021*, volume 12866 of *LNCS*, pages 141–152. Springer, 2021.

[9] T. T. Huynh, N. V. Do, T. N. Pham, and N. T. Tran. A semantic document retrieval system with semantic search technique based on knowledge base and graph representation. In *SoMeT 2018, Granada, Spain, 26-28 September 2018*, volume 303 of *Frontiers in Artificial Intelligence and Applications*, pages 870–882. IOS Press, 2018.

[10] P. G. Ipeirotis. Searching digital libraries. In L. Liu and M. T. Özsu, editors, *Encyclopedia of Database Systems, Second Edition*. Springer, 2018.

[11] A. C. Junior, F. Orlandi, D. Graux, M. Hossari, D. O'Sullivan, C. Hartz, and C. Dirschl. Knowledge graph-based legal search over german court cases. In *ESWC 2020 Satellite Events, Heraklion, Crete, Greece, May 31 - June 4, 2020*, volume 12124 of *LNCS*, pages 293–297. Springer, 2020.

[12] I. Kollia, K. Rapantzikos, G. B. Stamou, and A. Stafylopatis. Semantic query answering in digital libraries. In *SETN 2012, Lamia, Greece, May 28-31, 2012*, volume 7297 of *LNCS*, pages 17–24, 2012.

[13] S. W. Kumpulainen and H. Kautonen. Accidentally successful searching: Users' perceptions of a digital library. In *CHIIR 2017, Oslo, Norway, March 7-11, 2017*, pages 257–260. ACM, 2017.

[14] A. Oviedo, N. Kasioumis, and K. Aberer. $5e^{\{x+y\}}$: Searching over mathematical content in digital libraries. In *ACM/IEEE-CE Joint Conference on Digital Libraries, Knoxville, TN, USA, June 21-25, 2015*, pages 283–284. ACM, 2015.

[15] F. Sovrano, M. Palmirani, and F. Vitali. Legal knowledge extraction for knowledge graph based question-answering. In *JURIX 2020, Brno, Czech Republic, December 9-11, 2020*, volume 334 of *Frontiers in Artificial Intelligence and Applications*, pages 143–153. IOS Press, 2020.

[16] P. Thagard. Causal inference in legal decision making: Explanatory coherence vs. bayesian networks. *Appl. Artif. Intell.*, 18(3-4):231–249, 2004.

[17] V. Yellepeddi, P. Manimegalai, and S. B. Suvanam. Accurate approach towards efficiency of searching agents in digital libraries using keywords. *J. Medical Syst.*, 43(6):164:1–164:6, 2019.

[18] H. Yu, Q. Zhou, and M. Liu. A dynamic composite web services selection method with qos-aware based on AND/OR graph. *Int. J. Comput. Intell. Syst.*, 7(4):660–675, 2014.