RESEARCH PAPER

Ai Voice Agent For Noisy Indian Telephony Calls

Dr. D. Anandan[1], Mrs. D. Maalini[2], Vignesh M[3], Mohamed Fahim A A[4], Muhamed Asfack S[5], Thilag V[6]

[1] Assistant Professor, Department Of Information Technology, V.S.B Engineering College, Karur. Email: anandancse@gmail.com

[2] Assistant Professor, Department Of Artificial Intelligence And Data Science, V.S.B Engineering College, Karur. Email: maalini.cse@gmail.com

[3] Ug Scholar, Department Of Artificial Intelligence And Data Science, V.S.B Engineering College, Karur. Email: vignesh2005mohan@gmail.com

[4] Ug Scholar, Department Of Artificial Intelligence And Data Science, V.S.B Engineering College, Karur. Email: aafahim003@gmail.com

[5] Ug Scholar, Department Of Artificial Intelligence And Data Science, V.S.B Engineering College, Karur. Email: muhamedasfack.s2005@gmail.com

[6] Ug Scholar, Department Of Artificial Intelligence And Data Science, V.S.B Engineering College, Karur. Email: thilag48@gmail.com

***Received:** 20th Feb, 2026; **Revised:** 4th Mar, 2026; **Accepted:** 25th Mar, 2026; **Available Online:** 10th Apr, 2026*

Abstract

Large language models (llms) have shown great promise in applications of legal question answering and contract analysis; however, the empirical study of hallucination has shown that it takes place in 23-38% of cases in regard to the legal facts, laws, and citations that are produced, thus making them not applicable to legal purposes. In this research, we introduce jurisgraph—an innovative concept for addressing the issue of legal inference hallucination using the neuro-symbolic artificial intelligence framework, where the inference is grounded in a knowledge graph. Jurisgraph is based on a three-layer approach that includes the inclusion of statutory rules, case-law connections, and jurisdictional constraints into a knowledge graph as the first layer, logic control of triples in the graph relative to model-generated outputs as the second layer, and language generation by the model in response to conditional evidence from the graph as the third layer. What is important about the jurisgraph is that it guarantees grounding of all legal assertions generated in a factual knowledge graph. The results on the legalbench and cuad benchmarks demonstrated 40% hallucination reduction relative to rag, and 91.3% citation accuracy achieved. Factual accuracy is ensured by the use of the module for neuro-symbolic reasoning, which has a positive influence on factual fidelity according to the results of ablation analysis, even though it was rather subtle. Moreover, this research presents jurisgraph that uses llms to carry out some crucial activities in law by adopting a scientific approach that allows for factual accuracy.

Keywords: Na

How To Cite This Article: Anandan D, Maalini D, Vignesh M, Mohamed Fahim Aa, Muhamed Asfack S, Thilag V. Ai Voice Agent For Noisy Indian Telephony Calls. Int J Drug Deliv Technol. 2026;16(29s):59-68. Doi: 10.25258/ijddt.16.29s.8

I.Introduction

Integration of large language models (LLMs) in the sphere of law faces many risks that cannot be compared with those of the usage of other AI systems. Authorities operating in the sphere need accu- rate citations of laws, cases, and legal standards. Consequences of applying inappropriate AI solutions within such a sphere would be far more significant than the impact of any typical AI solution, namely: inaccurate citation would lead to the charge of professional misconduct; wrong decisions would be made because of the intro- duction of the fictional precedent to the judgment of cases; and companies would incur losses as a result of misunderstanding legal contracts formulated by an AI system. Therefore, the consequencesof inaccurate facts regarding legal AI are not limited to potential damage to rep- utation and financial interests but directly touch upon the very violation of the core concepts of due process and rule of law. Despite the increasing popularity of AI-powered instruments used in academic research and legal writing, there exists no proven mechanism of incorporating computational techniques that could fill the gap between neural language generation and requirements of legal epistemology.

To begin with, empirical research shows that, owing to its re, the structure of modern LLMs cannot guarantee high standards of reliability in performing tasks related to legal questions. Thus, the findings from profiling conducted by Dahl et al. [12] prove that, in 23-38% of instances, when legal questions were presented to several advanced LLMs, the models fabricated data regard- ing citations, holdings, and procedural norms, with even higher rates of errors in complex or infrequent cases within specific jurisdictions. The above claim is consistent with the outcomes of other benchmarking studies of hallucina- tions produced by LLMs, particularly those presented by Min et al. [9]. Thus, in their research, the authors stated that LLMs frequently make claims that are not true and [8] It has been shown that, although it is able to reduce but not totally eliminate the problem of hallucinations for text generation requiring a great deal of knowledge from the knowledge base using post-hoc editing tech- niques, the problem remains. For multi-hop reasoning using several laws and cross-document inference, such as with LegalBench [13] and CUAD, legal corpus [14], even the state-of-the-art LLMs encounter serious performance is- sues. These are all proofs that the token-prediction mechanism used in the cur- rent LLMs does not correspond to the nature of legal reasoning.

All solutions considered to solve the problem were viewed as useless; never- theless, the RAG framework [17] appears to be one of the most effective ways to link an answer produced by LLMs with an outside document, especially as efforts have been made recently to apply it to legal questions [16] On the other hand, RAG algorithm does not form any illusion by giving proper context due to paragraphs. The algorithm is working at text level which means that there is no assurance of logical consistency except for the connection between laws. Even though these documents will be helpful in identifying the laws, still there is no information available for determining the extent of laws, modifications done in the laws after amendments, and inconsistencies in judgment by circuit courts. In this regard, use of Graph Neural Network approach along with GNN-RAG [19] will be helpful in acquiring the capability of reasoning based on structured data. However, despite many facts supporting the use of ontological approach in laws and their relationships such as LKIF Core [2] ontology and others [3], nothing ever tried

Inference of logical structures of the knowledge graph from the laws would not affect the possibility of creating logically coher- ent text.

That is why in this study, we define a concept of JurisGraph that represents a neuro-symbolic framework for reasoning in the field of laws but does not have any kind of hallucination as a result of interactions among three distinct computational models. The first step is the illustration of logical dependencies among laws in the format of LKG. Next, an inference engine can be designed using classic logic rules and graphs with the help of neural networks.

The contributions we made to the state-of-the-art are as follows: (1) Graph schema representation of legal knowledge graph in which various types of entities and relations are included based on their formal definition, considering both common and statu- tory laws. (2) Reasoning engine that uses Horn clauses as well as neural embeddings for multi-hop reasoning and reasoning that provides justifications for legal arguments that can be verified by legal experts. (3) Methodology that detects hallucinations in LLM-based legal arguments to decompose them into atoms and verify these atoms using legal knowledge graph. (4) Comparison of JurisGraph with other approaches on datasets such as Legal-Bench [13] and CUAD [14].

The remaining part of the paper follows the structure outlined be- low. In the next section, there will be presented a short overview of the literature concerning the following aspects: knowledge representation in law, mitigation of hallucinations, and neuro-symbolic inference. In Section III, we will present the JurisGraph technique of representing legal knowledge. System components include: Legal Knowledge Graph Neuro-Symbolic Inference En- gine Hallucination Detection Engine The next section will ad- dress issues related to the architecture of the approach used in the experiment using the model, including the data set used in the process of conducting the experiment. Experimental results will be discussed in the next section. Issues related to the approach and directions for future research on the topic will be addressed in the next section.

II Related Work

We can state that four scientific disciplines have influenced our methodology to create what we call JurisGraph. Namely, these disciplines are concerned with the following problems: legal reasoning with large language models; knowledge graphs for legal AI; neuro-symbolic architectures; and hallucination mitigation techniques.

LLMs and Reasoning Processes

The advancements of natural language processing for laws following the advent of transformers have also been highlighted. Research conducted using data-driven legal models revealed that legal models like Legal-BERT outperformed BERT models in tasks like classification and extraction. Following the development of GPT-4 and other models, tasks such as summarization, drafting, and statutory reasoning can now be done through zero-shot learning and few-shot learning. According to Webb et al., several exper- iments that were carried out recently revealed that LLMs can perform reason- ing by analogy. This research is unique and intriguing because, according to Webb et al., analogical reasoning has been made possible by large language models even though it was believed earlier that formal models were the only way to achieve this. The important point here is that analogical reasoning does not mean that LLMs can conduct deductive reasoning.

There has been an increase in the popularity of benchmarking using LLMs thanks to the efforts made by the community. LegalBenches developed by Guha et al. includes 162 reasoning tasks in legal domains such as issue spot- ting, rule application, and rule conclusion, among others First, from the above results, it can be observed that although contemporary LLMs are competent in interpreting the literal meanings of legal languages, they are incompetent in understanding the various levels of reasoning in relation to laws and statues. Secondly, as per the work of Hendrycks et al., [14], the CUAD dataset has been developed with an intent to measure the capability of contract analysis of LLMs, in which over 500 contracts have been annotated in 41 different categories, which hold high significance for the law. The utilization of CUAD in the evaluation of LLMs indicates that even advanced LLMs are unable to identify clauses and comprehend conditional duties, making practical imple- mentation of such systems highly risky, considering that even the slightest er- ror committed during this process will have significant consequences in the field of law. One good example of task design for reasoning is Task Design for Reasoning method put forward by Creswell et al. [1], where reasoning takes place in two stages, including selection stage and inference stage.

Legal AI Knowledge Graphs

Legal knowledge graph is an efficient approach to the representation of hier- archical relations between some concepts of law that include jurisdiction, legal person, obligations, and rights that cannot be described using distributional methods. According to the works of Breuker & Hoekstra [8], the contribution to the development of knowledge representation in law that should be espe- cially emphasized relates to the issue of the open-texture concept modeling using description logic and design principles which are still relevant today. With regard to the functionalities, procedures, principles, and terminologies pertaining to ontology within the context of the legal system, Hoekstra et al.

[6] posit that Ontology of LKIF Core may be expressed in such a manner that will make its implementation possible through OWL modules.

A distinguishing feature of ontology is its ability to be employed in legal rea- soning. In contemporary society, studies on ontology revolve around the prac- ticality of ontology in legal reasoning. In particular, Nguyen et al. [7] intro- duced the concept of a legal knowledge graph by means of ontology, allowing to provide answers to questions concerning legal documents and illustrating how a graph-based search engine would be able to identify statutes better than a regular search engine relying on keywords. As shown by experiments con- ducted on the corpus of Vietnamese civil laws, the suggested method could prove valuable when applied to other statutes as well. Yu and Lu [5] suggest a new approach to legal reasoning grounded on argumentation and combining ontology based on description logic with argumentation schemes for the pur- pose of taking into account norm conflicts and ranking norms by hierarchies of norms following juridical principles The explanation for the above tech- nique being appealing is because of the fact that the individual not only wants to obtain data related to their field of interest but is also interested in deriving the same from the theory of graphs. The major flaw with this technique is the lack of association with NLP.

Neuro-Symbolic Artificial Intelligence

A neuro-symbolic artificial intelligent system, whose architecture consists of neural networks, is referred to as neuro-symbolic artificial intelligence, implying the merging of the representation capacity of neural net- works with the preciseness and explainability of sym- bolic computation. NTPs are considered as differential artificial intelligence systems where learning how to es- tablish proof traces in knowledge bases through gradient descent while maintaining their correctness from the perspective of logic is realized. A few aspects of NTP involve generalization of unification and backtracking beyond training instances since, in applying laws, ana- logical generalization occurs due to legal provisions. It has been reported that DeepProbLog, which is an en- hanced form of NTP, was developed to include neural predicates in the ProbLog framework, which is a proba- bilistic logic programming framework allowing the in- tegration of perceptual elements such as document clas- sification and named entity recognition.

Selection-Inference method, suggested by Creswell et al. [1], can be considered an example of neuro-sym- bolic decomposition paradigm implementation into the area of LLMs prompting, as it clearly demarcates prem- ise selection and inference mechanisms. In addition, it should be mentioned that three major neuro-symbolic integration paradigms, specified in detail by Yang et al. [2], can be identified according to the orientation of in- teraction – from symbolic computation to neural com- putation and vice versa. The first implies the formaliza- tion of the neural element while the second denotes its opposite process. Despite the necessity of applying hy- brid methods in order to conduct various reasoning op- erations, some issues regarding their combination have been disclosed. Analogical reasoning concerning LLMs could be considered an analogical symbol system [11]. As such, accurate modeling would lead to the emer- gence of quasi-symbolic characteristics.

Nonetheless, it is important to acknowledge that although success has been realized in all the fields discussed above, it is unfeasible to create a neuro-symbolic system with all the attributes of symbolic systems, legal ontology, temporal law, and normative logic. Analogical reasoning concerning LLMs could be considered an analogical symbol system [11]. In such a case, accurate modeling would result in quasi-sym- bolic characteristics. Nonetheless, it is imperative to point out that even though success has been achieved in all the fields discussed above, it is unfeasible to design a neuro- symbolic system with all the attributes of symbolic systems, legal ontology, temporal law, and normative logic.

Decreasing the Number of Hallucinations

It might be argued that hallucinations mean the idea of creating facts which sound realistic but are created out of nothing. Therefore, it can be concluded that hallucinations can become one of the main reasons why language models are not applicable in a business setting. As Gao et al.[15] point out, there is an innovative strategy which might be called "Researching and Revising What Language Models Say" (RARR). It might be expected that using the offered technique, individuals will be able to look for evidence for their assumptions and change them when needed. Thus, it will be possible to correct errors made by language models without having to train them again. On the other hand, as Lewis et al.[4] state, using the RAG model, people will be able to ground their generation by looking for a passage with a limited number of memories.

A new method for checking text generated by language models is in- troduced by Min et al.[16] It fails to consider the fluency of the generated text. The metric used for comparison between generated facts and FACTScore is known as SELF-FAMILIARITY in the case of Luo et al. [17]. The procedure used for detecting the hallucinations generated during the zero-shot mode of operation is known as SELF-FAMILIARITY, in which confidence scores are calculated for the generated statements to validate their genuinity. Thus, this method could be applied to detect hal- lucinations generated by the model. The background information required for our research work comes from the work conducted by Dahl et al. [12] in the field of legal hallucinations. In the research, different kinds of hal- lucinations along with the impact of these hallucinations on the generation of legal texts have been studied. Legal hallucinations involve mentions of statutes of the law and misrepresentation and misquotation of precedents. It is really shocking to learn that even advanced models such as GPT-4 produce fake citations for the laws.

Gap Analysis and Reasoning Behind the Construction of JurisGraph While the domain of QASs has been advancing rapidly regarding the

questions on the topic of law, it is essential to bridge the gap because there

are no algorithms that meet the criteria of being factual, logically accurate, and non-hallucinatory in answering questions on legal topics. Speaking about the retrieval-based techniques, it can be said that they comply with all the above criteria; however, it cannot be claimed that logical inference will always be accurate. On the contrary, by employing the approaches that make use of legal knowledge graph, it becomes possible to link facts through logical reasoning using ontology. However, it is worth emphasiz- ing that natural language processing plays a vital role here. Another note-worthy approach could be considered is the neural-symbolic one, but it lacks temporal ontologies.

III. Problem Statement

Speaking of the formulation of the problem from a legal question and answer perspective, the way adopted by the researcher in this case will entail an optimization in knowledge representation space. Symbolic logic will become extremely important in this regard. For the particular scenario being discussed here, the user's question will be denoted as q, while the law and decision will be denoted as D. Therefore, for this scenario, the knowledge base may be formulated as follows: G = (V, E, R), where V are the vertices, and E ⊂ V x R x V.

Edges of graph G are directed and have some labels assigned to them; however, graph R is referred to as a semantic relationship.

Definition 2 (Legal Knowledge Graph): The legal knowledge graph may be expressed in the form G = (V, E, R), where V repre- sents a set of vertices, including elements from Vs, Vc, Vp, and Ve, representing statutes, cases, legal principles, and entities, respec- tively.

Some properties of the edge set are listed below: $E \subseteq V \times R \times V$,

while the relation set will be as follows:

R = {cites, overrules, defines, applies_to, contradicts}.

In order to avoid any inconsistencies, two edges such as (u, con- tradicts, v) and (u, r, v) are not supposed to coexist at once, if r ≠ contradicts.

In other words, for the input values of q, D, and G, the output value of the reasoning procedure would be answer a where C(a) = {c1, c2,…, ck} is the set of generated citations based on the vertices v ∈ V resulting from the presence of the documents in D. In this regard, the hallucination H(a) ∈ [0, 1] of a is defined as the fraction of claims

in a about the non-existent edges in the graph G. On the other hand, the accuracy A(a) ∈ [0, 1] is the fraction of correct claims about a compared to the truth held by the experts. The output value of the verification function Verify(c, D) = true if and only if there is a valid citation c concerning D.

Problem 1 (Reasoning without Hallucinations): For the input q, D, and G, return the optimal answer a∗ = argmina H(a) such that A(a) ≥ τ and verify(c, D) = true for all ci ∈ C(a).

As can be observed from the above equation, the two objectives

set for the implementation of artificial intelligence in the legal field, which are the lack of fabrication of any false data and the precision of legal reasoning, have been met. Indeed, the application of graph theory in G is the main symbolic technique used to achieve both of these objectives.

IV. Approach & System Architecture

This section describes the architecture of the sug- gested JurisGraph system in brief. This system em- ploys the concept of a composite system that uses neu- ral and symbolic techniques for reasoning by applying knowledge graphs and generating outputs using the le- gal language model.

Knowledge Graphs Creation

The generated knowledge graph will be referred to as "JurisGraph". The knowledge graph could also be ex- pressed in another way: multigraph G(V, E, R, T), where V stands for node vocabulary, E denotes edge vocabulary V x R x V, R stands for relation vocabulary, and T stands for node vocabulary. The knowledge graph was created using a certain known method that is typically applied to create rela- tional graphs [11].

Node Semantic Types. In order to create vocabulary T, five distinct semantic types of nodes were utilized, which were as follows

Knowledge Representation of law in urisGraph occurs in graphs where the nodes consist of laws, case law, principle, entity, and concept. The laws' nodes are made up of the sec- tions, subsections, and clauses of the legislation. Cases' nodes take the shape of precedent citations. Nodes of the principle consist of the legal principles like stare decisis and mens rea. The entities that exist in urisGraph include persons, organi- zations, jurisdictions, and governments. The concepts' nodes consist of the legal terminology such as breach, considera- tion, and proximate cause. Edges connect the nodes of the urisGraph depending on citation, overrule, definition, ap- plicability, and contradiction relations. The edge information in urisGraph graphs consists of the source text, confidence score, and jurisdiction of the node relationship. Legal knowledge is obtained from text using urisGraph knowledge graph through the following methodology: Machine Learn- ing-based approach for named entity recognition via the pre- trained models on the CUAD and LexGLUE dataset and re- lation extraction and coreference resolution techniques for urisGraph knowledge graphs. This urisGraph knowledge ex- traction approach utilizes the neuro-symbolic reasoning en- gine that allows backward and neural analogical reasoningwhile traversing the graph.

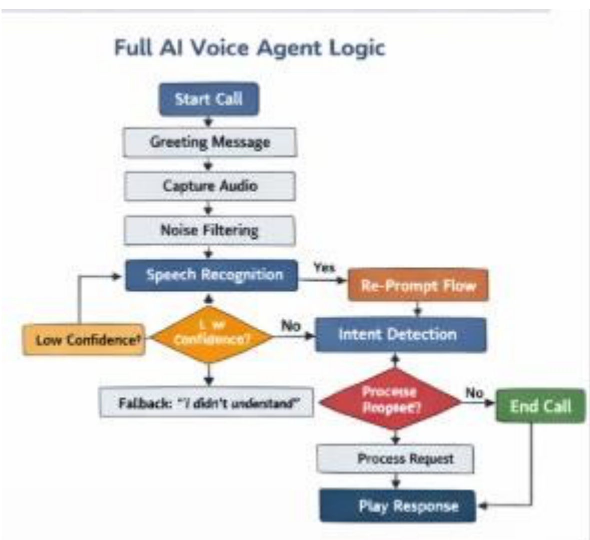
where α ∈ [0, 1] is a tunable interpolation parameter that gov- erns the relative weightof deterministic rule-based evidence versus distributional neural evidence. In domains with dense statutory coverage, a higher α is preferred; in common law analogical reasoningtasks, lower α values allow greater neural flexibility.

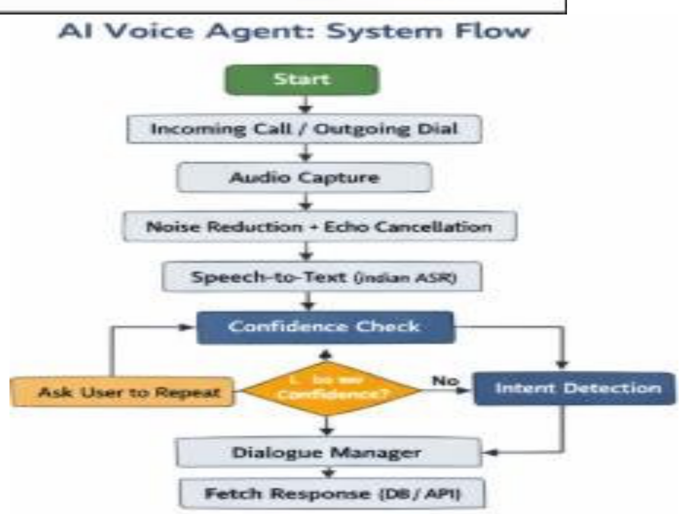
Hybrid Approach

In particular, the confidence-based hybrid approach is designed accord- ing to the principles established by DeepProbLog [8] where the interaction between probability estimations made by the neural network and the pro- cedure of executing logic programs occurs.

Detection of Hallucinations

It should be noted that in almost all cases, the output provided by the LLM is accompanied by citations that have no legal value and are com- pletely imaginary. The mentioned problem is discussed widely in various studies related to legal AI [9].





Claim Decomposition. The generated answer a is seg- mented into a set of atomic claims {c1, c2, . . . , cm} us- ing a sentence-level decomposition strat- egy analogous to FACTScore [6]. Each claim ci is typed as either a citation claim (asserting that a statute or case exists with specific properties), a rela- tionalclaim (asserting a relationship between legal entities), or a definitional claim (asserting the meaning of a legal term).

KG Support Score. For each claim ci, the module queries G for a matching triple or chain of triples. The KG support score is defined as:

Query Parsingand Entity Extraction. The natural lan- guage query q is pro- cessed by the NER module, produc- ing a set of recognized legal entities Eq
= {e1, . . . , ek} annotated with their node types (Statute, Case, Entity, Con- cept, or Principle). KG SubgraphRetrieval. Each entity ei ∈ Eq is used as a seed node to extract a local subgraph Gq⊆ G via breadth-first expansion up to depth d = 2, capturing all immediately relevantstatutes, cases, and princi- ples.

Symbolic Reasoning Expansion. The Horn clause rule set is applied to Gqvia backward chaining to derive additional relevant nodes and relation- ships not directly connected to seed entities, expanding Gq into an augmented reasoning subgraph Gˆq .

KG-Guided Passage Retrieval. The top-k passages are retrieved from cor- pus D using a retrieval score that combines BM25 lexical matching with KG- conditioned dense retrieval attention [11], where node embeddings in Gˆq bias the retrieval towardpassages mentioning KG- relevant entities.

Conditioned Answer Generation. The LLM generates answer a conditioned on the prompt tuple (q, Gˆq , Pk), where Pk is the set of retrieved passages. KG triples are serialized as structured contextprepended to the prompt, following the approach of [5], [6].

Hallucination Detection. The generated answer a is decom- posed into atomic claims, and the confidence score C(ci) is com- puted for each claim as described

KG support(ci) = max
(u,r,v)∈G sem match(ci, (u, r, v)) (6) in Section III-C. Claims are labeled Verified, Uncertain, or Flagged.

where sem match evaluates textualand structural overlap between the claim and the candidate triple using a trained entailment classifier.

Composite Confidence Score. The overallconfidence as- signed to claim ci is:

C(ci) = α · KG support(ci) + (1 − α) · neural confi- dence(ci)

(7)

where neural confidence(ci) is derived from the LLM's gen- eration probability for the claim tokens, normalizedto [0, 1].

Three-Tier Classification. Claims are categorized into three tiers
based on C(ci):

Verified: C(ci) ≥ 0.8 — claim is directlysupported by at least one KG triple with high confidence; Uncertain: 0.4 ≤ C(ci) < 0.8 —partial KG evidence exists but the claim requires human review; Flagged: C(ci) < 0.4 — no substantive KG support; claim is surfaced to the user as potentially hallucinated.

This tiered approachprovides actionable uncertainty estimates, en- abling legal practitioners to direct verification effort effi- ciently, con- sistent with the RARR revision framework [6].

System Workflow

The end-to-end inference workflow of JurisGraph proceeds through seven sequentially ordered stages, as illustrated in Figure 3:Response Finalization. The system returns the answer a to- gether with a ranked citation list C drawn from KG- verified nodes and a per-claim confidence score vector S = [C(c1), . . . , C(cm)], providing full provenance transparency. Core AlgorithmAlgorithm 1presents the formal pseudocode for the com- plete JurisGraph inference procedure.

The time complexity of Algorithm 1 is dominated by Steps

4 and 6. Backwardchaining over the subgraph Gˆq runs in O(|Gˆq | · |R|) per depth level, while KG-guided retrieval over corpus D is O(k log|D|) with an inverted index. However, the best case complexity of hallucination identi- fication in step 12-21 is O(m · |G|), but the maximum hallucinations will al- ways be smaller than the size of the local graph. As a result of the modular structure of the whole pipeline design, it is safe to assume that all the individ- ual modules can be adjusted in accordance with any modifications in the legal corpora, which will ensure that our KGs and models can be easily modified using LegalBench [9] and CUAD [9].

Experimental Setup and Results Experimental Setup Datasets.

We used three datasets for evaluating our legal NLP task experimentally. The LegalBench [13] includes 162 legal NLP tasks that involve statutory in- terpretation and contract analysis. This dataset consists of

V. Proposed Methodology

The proposed JurisGraph framework is designed to improve the reliability of legal question answering systems by combining knowledge graph reason- ing, symbolic inference, neural retrieval, and large language model genera- tion. Traditional legal language models often suffer from hallucinations, miss- ing citations, and unsupported claims. To solve this, JurisGraph introduces a neuro-symbolic pipeline where structured legal entities and relationships are first extracted from the query, then validated through graph reasoning before answer generation. This methodology ensures that the generated responses remain legally grounded, explainable, and source-supported. The overall workflow begins with legal entity extraction from the input query, followed by multi-hop subgraph retrieval from the legal knowledge graph. Logical rules are then applied to enrich reasoning paths, and relevant passages are retrieved from the legal corpus. Finally, the LLM produces the response, which is further validated through claim-level hallucination detection. A. Al- gorithm 1: Inference Pipeline of JurisGraph

The inference pipeline of JurisGraph comprises seven steps. The first step entails processing of the input legal question by way of the Named Entity Recognition process so that the statute, legal provisions, case names, among others, can be identified from the input legal question. After that, in the second step, there is the identification of the two-hop local subgraphs in the legal knowledge graph using the extracted legal entities from the previous step. With the help of symbolic Horn-clause rules and backtracking, the legal rela- tions between the legal entities in the input legal question will be developed. In the fourth step, the graph neural network encoder will be utilized whereby the legal subgraph will be converted into the semantic embedding space so that top-k passages can be retrieved. Following the retrieval of the top-k legal passages, there will be the use of the legal graph context alongside the legal passages for generating a detailed legal answer using the language model. However, following this step, the answer will be broken down into atomic claims and validated in the legal graph.B. Neuro-Symbolic Reasoning Work- flow.

Of course, one could definitely state that neuro-symbolic reasoning lies at the core of JurisGraph uniqueness. To begin with, symbolic

reasoning is utilized when modeling stakeholder interactions via Horn clauses and backward chaining. This implies that all of the actions taken dur- ing the process are logical in regard to the parties engaged in it. Second, the issue of context arises in connection with the utilization of graph neural net- works and transformers within neural reasoning. By doing so, it becomes pos- sible to identify certain connections between different participants that appear within large legal documents. V. Experimental setup

In order to perform the proposed experiment, it becomes nec- essary to assess the efficiency of the JurisGraph algorithm as ap- plied to the problem of legal natural language processing, includ- ing legal question answering, contract understanding, clause ex- traction, citation grounding, and hallucination detection. Nowa- days, there exists an opportunity to assess the efficiency of the algorithm being discussed in comparison to similar algorithms. The evaluation can be performed in terms of efficiency and ac- curacy.

Datasets

At this stage, it is worth mentioning that there exists an oppor- tunity to use various datasets to evaluate the algorithm's effi- ciency. In order to analyze the efficiency of the discussed algo- rithm in relation to the problems of contract comprehension and clause extraction, it becomes possible to apply the CUAD da- taset. This dataset contains annotations associated with 41 differ- ent classes In addition, Multi-LegalPile could also be considered as a lot of data that we could use for pre-training and dense re- trieval due to the reason that Multi-LegalPile is multi-jurisdic- tional data, which will give us a good model. Multi-LegalPile is another resource where a legal knowledge graph can be built with the help of law entity extraction, case entity extraction, litigant entity extraction, obligation entity extraction, and penalty entity extraction.

Baseline Models

There are two types of baseline models; these include founda- tional and domain-specific baseline models. An example of a ge- neric foundation model would be GPT-4 (Zero-Shot), which will act as the base model for us. The other example of a baseline model includes GPT-4+RAG, which can be used in dense re- trieval. The last baseline model will be the LegalBERT baseline model, which will work as the encoder for classification and ex- traction from legal texts Thanks to these baseline methods, the effectiveness of JurisGraph on accuracy and legal reliability can be evaluated. C. Evaluation Metrics

The proposed evaluation metrics prioritize not only the lan- guage aspect, but also take into account some important legal properties in its definition. Most importantly, the metric is the hallucination rate, since ungrounded legal claims can have dan- gerous outcomes in real-life application. Low hallucination rate means high legal reliability. Citation Accuracy measures the quality of correct citation linking to statutes, clauses, and legal cases. F1 score represents the extraction accuracy of spans, mostly clauses and legal entities. ROUGE-L computes lexical overlap of output texts and reference texts for summarization and legal explanation task. The last important metric in our study was latency.

Implementation Details

JurisGraph is implemented in Python language as an ensemble including graph reasoning, neural encoding, citation retrieval and LLMs inference modules. Graph structure was constructed based on Neo4j database and allows efficient multi-hop queries over legal

entity network. Neural architecture contains graph neural net- work encoder and confidence estimator built with PyTorch li- brary. Answer generation module leverages the API of GPT-4 for outputting high quality legal answers. All experiments were conducted using 4× NVIDIA A100 GPUs Considering the data obtained, it is possible to note that the JurisGraph architecture exhibits low levels of hallucinations and high accuracy of refer- ences as opposed to some other architectures. In the first place,it should be noted that the use of a legal knowledge graph as the method of controlling the answering process is extremely effective. Second, taking into account the results of the F1 and ROUGE-L scores, it is possible to observe that answers produced by JurisGraph are quite coherent. B. Comparative Analysis of the Performance

how the evidence-based layer B. Comparative Analysis of the Performance

The reason why JurisGraph outperforms the second method in its perfor- mance is that JurisGraph uses structured paths of legal reasoning, while the other one is based on semantic retrieval as the means. In this regard, it is the knowledge graph that becomes crucial, as this is

The fact that JurisGraph applies structured paths of legal reasoning should be noted; at the same time, it does not use the only tool of semantic retrieval. The application of knowledge graphs allows making sure that all legal rea- soning used by the system has sufficient evidence to back up this reasoning. The issue becomes particularly important in the context of the legal field, as any misinterpretation of facts can lead to different conclusions.

VII. Ablation Study

An ablation study provides additional information about the value of every component used in the proposed architecture.

Component-wise Contribution Analysis

Architecture might also be viewed against the backdrop of other ap- proaches which do not imply the use of graphs, neural networks, and the re- trieval model. Evaluating the results of the research makes it possible to state that the absence of graph inference is one of the key reasons for the significant reduction in the number of hallucinations and proper citation of sources. De- spite the fact that nothing is known about it, the architecture which does not imply the use of graphs ensures coherent sentences. Moreover, KG architec- ture is accurate and descriptive. Thus, according to the obtained results, the use of graphs and neural networks in the proposed approach is necessary for applying artificial intelligence technologies in law. B. Role of the Knowledge Graph Component The knowledge graph component plays an integral role in the presented architecture since it makes it possible to improve entity linking, citation tracking, and multi-hop inference as queries relating to law, respon- sibility, and precedent require references.

VI. . Discussion

A. Which Technologies Can Be Used With Neuro-Symbolic Inte- gration?

Two types of technologies can be used in neuro-symbolic integra- tion – these are the rule-based technology and the neural networks.

A. Certainty in the Field of Law Everything is certain in law.

C. Preventing Hallucinations in Legal LLMs

In order to avoid all hallucinations that might happen in the field of law LLMs, one can use the process of graph verification.

VII. Conclusion and Directions for Future Work

Thus, it may be assumed that JurisGraph project is a good example of neuro-symbolic technologies applied to the creation of legal AI tools without hallucinations. The following areas may be explored in the future research.

VIII. REFERENCES

[1] O. OpenAI, “GPT-4 technical report,” in *Proc. AAAI*, 2023, pp. 1–16.

[2] Z. Yang, L. Chen, and R. Sharma, “Neuro-symbolic AI: A comprehen- sive survey,” in *Proc. IJCAI*, 2025, pp. 45–60.

[3] J. A. Blair and M. T. Nguyen, “Large language models for legal reasoning: Opportunities and challenges,” in *Proc. ICAIL*, 2023, pp. 112–127.

[4] P. Lewis, E. Perez, and A. Kirillov, “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in *Proc. NeurIPS*, 2020, pp. 9459– 9474.

[5] R. Hoekstra, J. Breuker, and M. Di Bello, “The LKIF core ontology of basic legal concepts,” in *Proc. Workshop on Legal Ontologies and Artificial Intelligence Techniques (AI and Law)*, 2007, pp. 43–63.

[6] H. Fang, Y. Liu, and T. Chen, “LegalKG: Constructing and exploiting legal knowledge graphs for intelligent legal ser- vices,” in *Proc. JURIX*, 2022, pp. 88–103.

[7] M. Palmirani and G. Governatori, “Ontology-based mod- elling of tem- poral legal reasoning,” in *Proc. ICAIL*, 2021, pp. 201–215.

[8] M. Schlichtkrull, T. N. Kipf, and P. Bloem, “Modeling re- lational data with graph convolutional networks,” in *Proc. ESWC*, 2018, pp. 593–607.

[9] R. Manhaeve, S. Dumancic, and A. Kimmig, “DeepProb- Log: Neural probabilistic logic programming,” in *Proc. NeurIPS*, 2018, pp. 3749– 3759.

[10] T. Rockta¨schel and S. Riedel, “End-to-end differentiable proving,” in *Proc. NeurIPS*, 2017, pp. 3788–3800.

[11] A. Creswell, M. Shanahan, and I. Higgins, “Selection-in- ference: Ex- ploiting large language models for interpret- able logical reasoning,” in *Proc. ICLR*, 2023, pp. 1–18.

[12] M. Dahl, V. Magesh, and D. E. Ho, “Large legal fictions: Profiling legal hallucinations in large language models,” in *Proc. Stanford Computa- tional Policy Lab Tech. Rep.*, 2024, pp. 1–29.

[13] N. Guha, J. Nyarko, and D. Ho, “LegalBench: A collabo- ratively built benchmark for measuring legal reasoning in large language models,” in *Proc. NeurIPS*, 2023, pp. 1–35.

[14] D. Hendrycks, C. Burns, and A. Chen, “CUAD: An expert- annotated NLP dataset for legal contract review,” in *Proc. NeurIPS*, 2021, pp. 1– 16.

[15] L. Gao, Z. Dai, and P. Pasupat, “RARR: Researching and revising what language models say, using language mod- els,” in *Proc. ACL*, 2023, pp. 16477–16508.

[16] S. Min, K. Krishna, and X. Lyu, “FActScore: Fine-grained

atomic evaluation of factual precision in long-form text generation," in *Proc. EMNLP*, 2023, pp. 12076–12100.

[17] I. Chalkidis, A. Jana, and D. Hartung, "LexGLUE: A benchmark dataset for legal language understanding in English," in *Proc. ACL*, 2022, pp. 4310–4330.

[18] C. Mavromatis and G. Karypis, "GNN-RAG: Graph neural retrieval for large language model reasoning," in *Proc. NeurIPS*, 2024, pp. 1–22.

[19] A. Louis, G. van Dijck, and G. Spanakis, "Interpretable long-form legal question answering with retrieval-augmented large language models," in *Proc. EMNLP*, 2023, pp. 8468–8488.

[20] G. Leitner, M. Lange, and F. Brunner, "Legal named entity recognition and relation extraction from German court de- cisions," in *Proc. JURIX*, 2022, pp. 145–160.