Discourse based argumentation analysis for LLM verification

Boris Galitsky

Knowledge Trail San Jose CA US

Abstract

Large Language Models (LLMs) frequently produce fluent but unverifiable reasoning, resulting in potential hallucinations and faulty inferences. This study proposes an argumentation-based verification framework ValidArgLLM in which the reasoning expressed by an LLM is transformed into a defeasible logic program (DLP) representing world knowledge and a given problem description—such as a patient health complaint. The DLP is executed within a symbolic reasoning engine, and the resulting inferences are compared to the LLM's natural-language conclusions. The strength of arguments is computed based on discourse structure of text expressing arguments. Divergence between symbolic and neural reasoning outcomes indicates possible hallucination or inconsistency in the model's internal logic.

Keywords: Hallucinations, faulty inferences, argumentation-based verification, Defeasible Logic Program (DeLP), discourse analysis

1. Introduction

Large language models (LLMs) have achieved impressive results across diverse natural language processing tasks, inspiring interest in their use for domains that require structured reasoning. However, integrating LLMs into settings that demand context-sensitive, multi-step decision-making remains challenging. These models often excel at generating fluent and informed text but struggle to reason systematically, weigh competing possibilities, or revise conclusions when new information appears (Ferrag et al 2025). Their reasoning is associative rather than strategic, limiting their ability to handle complex, evolving problems.

Another major limitation is interpretability. Unlike human experts who reason through explicit, traceable arguments, LLMs reach conclusions through opaque statistical processes. This opacity makes it difficult to understand or justify their outputs, reducing trust and accountability in applications that require verifiable logic. Furthermore, LLMs are prone to reasoning hallucinations—producing statements that sound coherent but conflict with facts or internal logic. Because they lack explicit mechanisms for defeasibility or conflict resolution, such inconsistencies can undermine reliability. To address these gaps, LLMs must be complemented by external reasoning and verification layers capable of enforcing logical consistency and explaining why conclusions hold.

One promising approach is to pair an LLM with a symbolic or logical reasoning engine—for example, a Prolog-style rule base, a constraint solver, or a formal ontology of medical conditions (Galitsky 2025). The LLM produces a candidate answer, while the reasoning system independently checks whether that answer follows from known facts and rules. This creates a dual-track pipeline: the generative model proposes, the logical module disposes.

Such a framework can flag contradictions (e.g., a diagnosis incompatible with the patient's lab values), highlight unsupported steps in a rationale, or even suggest corrected outputs when classical reasoning yields a different conclusion. Over time, it can also feed back into training, teaching the LLM to prefer responses that survive external verification

To mitigate these risks, we introduce a *neuro-symbolic verification framework* ValidArgLLM that externalizes and tests an LLM's reasoning through *argumentation analysis*. The key idea is to translate the model's implicit causal and conditional reasoning into a formal system—defeasible logic programming—and verify its conclusions via a logical solver.

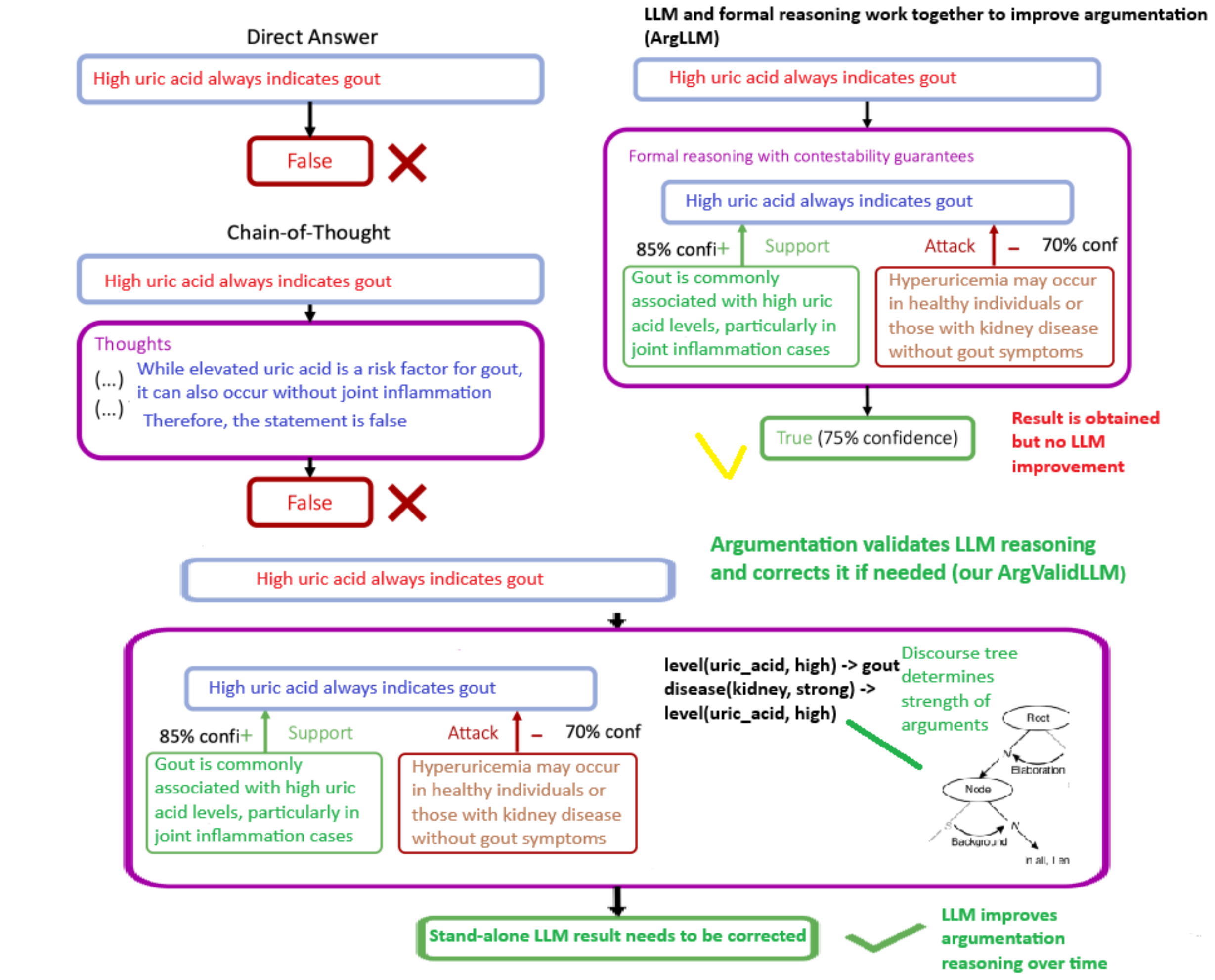

Direct Answer
High uric acid always indicates gout
False
Chain-of-Thought
High uric acid always indicates gout
Thoughts
(...) While elevated uric acid is a risk factor for gout, it can also occur without joint inflammation
(...) Therefore, the statement is false
False
LLM and formal reasoning work together to improve argumentation (ArgLLM)
High uric acid always indicates gout
Formal reasoning with contestability guarantees
High uric acid always indicates gout
85% confi+
Support
Attack
–
70% conf
Gout is commonly associated with high uric acid levels, particularly in joint inflammation cases
Hyperuricemia may occur in healthy individuals or those with kidney disease without gout symptoms
True (75% confidence)
Result is obtained but no LLM improvement
Argumentation validates LLM reasoning and corrects it if needed (our ArgValidLLM)
High uric acid always indicates gout
High uric acid always indicates gout
85% confi+
Support
Attack
–
70% conf
Gout is commonly associated with high uric acid levels, particularly in joint inflammation cases
Hyperuricemia may occur in healthy individuals or those with kidney disease without gout symptoms
level(uric_acid, high) -> gout
disease(kidney, strong) -> level(uric_acid, high)
Discourse tree determines strength of arguments
Root
N
Elaboration
Node
S
N
Background
In all, I en
Stand-alone LLM result needs to be corrected
LLM improves argumentation reasoning over time

Fig. 1 Four truth assessment approaches and two LLM-argumentation based architectures

Despite significant advancements in task performance, current approaches to argumentative reasoning continue to face critical limitations in terms of justifiability and interpretability. While models can often produce seemingly coherent explanations, it remains ambiguous how final decisions are actually reached or which intermediate steps genuinely contributed to the outcome. Recent studies have revealed that Chain-of-Thought (CoT) reasoning, though designed to enhance transparency, is itself prone to hallucinations, inconsistencies, and spurious reasoning paths that do not accurately represent the underlying inference mechanisms of large language models (Arcuschin et al., 2025; Barez et al., 2025). These findings call into question the validity of CoT traces as reliable post hoc justifications of model behavior rather than mere rhetorical artifacts. Moreover, in multi-agent or multi-LLM debate settings, the reasoning process becomes even more opaque: the discussions between models are typically unstructured, non-formalized, and difficult to audit, making it nearly impossible to systematically reconstruct the logic that led to a collective conclusion. This lack of traceability undermines confidence in the epistemic soundness of model-driven argumentation and highlights the need for structured frameworks capable of capturing, verifying, and explaining the reasoning trajectories that lead to final decisions.

Fig. 1 compares four levels of reasoning used by language models, showing how they evolve from simple answers to structured, self-correcting discourse-based reasoning. At the top, the *direct answer* model provides an immediate response without explanation. It may be correct or incorrect, but there is no visibility into why the model chose that answer. Because no reasoning steps are revealed, errors cannot be traced or corrected.

The next level, *chain-of-thought reasoning*, adds a sequence of intermediate steps that make the process more interpretable. However, these steps remain unverified. The reasoning might sound plausible while still being factually wrong, since the model does not test or challenge its own statements.

The *argumentative model* introduces a more structured approach. Instead of producing one line of reasoning, it generates multiple arguments, distinguishing between supporting and attacking ones. This enables a form of contestability: each conclusion is backed by explicit evidence and can be challenged by counterarguments. Still, this stage only formalizes reasoning—it does not validate or improve it. The model outputs argument structures but lacks a mechanism to revise its own conclusions.

The *discourse-based argument validation model* ValidArgLLM at the bottom represents the most advanced form. It integrates discourse structure to evaluate how strongly each argument contributes to the overall reasoning. By analyzing rhetorical relations such as elaboration, justification, or background, it can weigh the importance of each argument and decide which should dominate the conclusion. This allows the system to detect when the original model's answer is inconsistent with the discourse-level balance of evidence and automatically correct it. Over time, this validation loop enables the model to refine its reasoning and become more consistent and interpretable. This architecture adds discourse awareness, verifiable structure, and self-correction. It moves beyond merely producing or scoring arguments—it understands how arguments interact and uses that understanding to ensure that reasoning outcomes are both coherent and defensible.

Rhetorical Relation	Nucleus (main point)	Satellite (less important context)	Relative Argument Strength (Nucleus : Satellite)
Cause	The patient developed a high fever.	Because the patient had recently returned from a malaria-endemic area.	0.8 : 0.2
Effect / Result	The antibiotic successfully cleared the infection.	As a result, the patient's white blood cell count decreased.	0.7 : 0.3
Condition	The patient must take insulin twice daily.	If her blood glucose remains above 180 mg/dL.	0.6 : 0.4
Contrast	The patient reports severe knee pain.	Whereas the ankle pain has largely subsided.	0.55 : 0.45
Elaboration	The patient experienced a heart attack.	Specifically, a non-ST elevation myocardial infarction affecting the inferior wall.	0.65 : 0.35
Concession	The patient's cholesterol level improved.	Although his diet compliance was inconsistent.	0.75 : 0.25
Background	The surgeon performed an emergency appendectomy.	The patient had arrived at the hospital only two hours earlier.	0.85 : 0.15
Enablement / Purpose	The nurse administered a sedative.	To facilitate the insertion of a central venous catheter.	0.7 : 0.3
Evidence / Justification	The patient is diagnosed with pneumonia.	Supported by chest X-ray showing bilateral infiltrates.	0.6 : 0.4
Evaluation	The treatment outcome is considered successful.	According to the hospital's quality benchmarks.	0.65 : 0.35

Table 1: nucleus and satellites and their importance in attacking and supporting arguments

2. Discourse and defeasibility

Each rhetorical relation has a *nucleus* (the "main" proposition) and a *satellite* (supporting or contextual material). The satellite always carries less essential information than the nucleus (Table 1). One can see that nucleus contains main diagnostic/treatment fact (higher base probability) and satellite caries contextual/supporting info with lower significance. These values are obtained in the course of improvement of validation performance, described in Evaluation section.

Hence the rules are:

1) strict rules (must hold, nucleus) Head <- Body.
2) defeasible rules (should hold, satellite) Head <~ Body.

Facts are either strict or defeasible:

fact.   % strict fact   vs *fact* <~ .   % defeasible fact

We now show how a *defeasible logic program* in ValidArgLLM would be built from a real *nucleus vs. satellite* pair: Nucleus (main claim):

"The patient must immediately start a course of antibiotics for bacterial pneumonia."

Satellite (supporting context):

"Because the chest X-ray shows an infiltrate consistent with pneumonia."

Here the **nucleus** expresses a mandatory action (antibiotics).

The **satellite** is evidence/justification (X-ray finding) — useful, but not itself the main prescription.

In a defeasible logic program:

```
% Strict rule from the nucleus: MUST do this if bacterial pneumonia diagnosed
start_antibiotics(Patient) <- diagnosis(Patient, bacterial_pneumonia).
% Defeasible rule from the satellite: SHOULD suspect pneumonia if X-ray shows infiltrate
diagnosis(Patient, bacterial_pneumonia) <~ chest_xray_infiltrate(Patient).
% Facts:
chest_xray_infiltrate(john) <~ .   % defeasible evidence
```

The nucleus states the obligatory/primary outcome, while the satellite is only supportive, so its rule is defeasible/overridable.

The nucleus carries the main point of the discourse. In DLP you can treat it as a *strict rule or a strict fact*, because it is asserted to hold independently of the supporting material.

diagnosis(Patient, bacterial_pneumonia).

Even if we later remove the satellite (the X-ray), the nucleus stands on its own.

The satellite carries supporting or contextual information. In DLP you model it as a *defeasible rule* whose conclusion only fires under the context of the nucleus (or some nucleus-derived condition). This makes it conditional:

```
% satellite only applies if nucleus (diagnosis) is already true
chest_xray_infiltrate(Patient) <~ diagnosis(Patient, bacterial_pneumonia).
```

More generally:

```
satellite_fact(X) <~ nucleus_fact(X).
```

So in ValidArgLLM the satellite is not automatically accepted; it's accepted *defeasibly* and only when its nucleus context holds. In a defeasible logic program derived from discourse, nucleus statements are converted into strict rules or strict facts that hold independently. Satellite statements are converted into defeasible rules whose applicability is conditional on the corresponding nucleus being true; they express "should" or "likely" information that can be overridden or becomes vacuous if the nucleus is absent.

3. Enabling DeLP with argument strength computation

Given a natural-language case description, the LLM within ValidArgLLM constructs a set of *defeasible rules* encoding its inferred world knowledge:

r1: gout :- asymmetric_joint_inflammation, uric_acid_high.
r2: immune_arthritis :- symmetric_joint_inflammation, fever.
r3: asymmetric_joint_inflammation :- not symmetric_joint_inflammation.
r4: prefer immune_arthritis over gout if fever.
Here, each rule expresses a conditional belief that may be overridden by stronger evidence.

A *defeasible logic program* is a set of facts, strict rules, Π, of the form *A:-B*, and a set of defeasible rules Δ of the form, *A*- < *B*, whose intended meaning is "if *B* is the case, then usually *A* is also the case." A DeLP for knowledge sources includes facts which are extracted from search results and strict and defeasible clauses where the head and body form commonsense reasoning rules (Garcia and Simari, 2004).

Let *DT*=(*N,R*) be a *discourse tree*, where *N* is the set of elementary discourse units (EDUs), and $R \subseteq N \times N$ is the set of rhetorical relations between nucleus and satellite spans. Each relation r_i=(nucleus,satellite,relation) ∈R has a rhetorical type (e.g., *Cause, Evidence, Elaboration, Concession*) and an associated relative argument strength coefficient $\propto_{relation} \in \mathbb{I}$, representing how much more influential the nucleus is compared to its satellite.

For every defeasible rule A -< $B_1, B_2, \dots, B_k \in \Delta$ we associate a discourse strength weight w(A)∈ $\mathbb{I}$ computed as $w(A) = \frac{1}{|Sat(A)|} \sum_{(nuc,sat,relation \in R_A} \propto_{relation},$

where R_A is the set of discourse relations in which participates, and *Sat*(*A*) are its supporting satellites. Thus, rules derived from nucleus EDUs obtain higher *w*(*A*) values (closer to 1), while rules from satellite EDUs obtain lower *w*(*A*)proportionally to their rhetorical importance.

Let *P*=(Π, Δ) be a DeLP program and *L* a ground literal. A defeasible derivation of *L* from *P* consists of a finite sequence $L_1, L_2, \dots, L_n$ of ground literals, such that each literal L_i is in the sequence because:

1. L_1 is a fact in Π, or
2. there exists a rule R_i in *P* (strict or defeasible) with head L_i and body $B_1, B_2, \dots, B_k$ and every literal of the body is an element L_i of the sequence appearing before L_j ($j<i$).

Let *h* be a literal, and P=(Π, Δ) a DeLP program. We say that <*A*, *h*> is an argument for *h*, if *A* is a set of defeasible rules of Δ, such that:

1. there exists a defeasible derivation for *h* from (Π ∪ *A*);
2. the set (Π ∪ *A*) is noncontradictory; and
3. *A* is minimal: there is no proper subset A_0 of *A* such that A_0 satisfies conditions (1) and (2).

Hence an argument <*A*, *h*> is a minimal noncontradictory set of defeasible rules, obtained from a defeasible derivation for a given literal *h* associated with a program P.

The generated DLP is executed within a defeasible reasoning environment which is the SWI-Prolog extension. Each rule is treated as an argument, and conflicts among arguments give rise to a dialectical structure that determines which conclusions are warranted under a chosen semantics (e.g., grounded or preferred extension).

The logical conclusions derived by the solver are compared with the LLM's original verbal output. If both converge (e.g., both conclude *immune arthritis*), the LLM's reasoning is considered *grounded*. If they diverge (e.g., solver: *immune arthritis*, LLM: *gout*), the discrepancy signals a reasoning hallucination—a claim unsupported by the formal reconstruction of its own reasoning.

We represent a quantitative bipolar argumentation framework (QBAF) (Baroni et al 2019) denoted as a quadruple $\langle A, R^-, R^+, \tau \rangle$ via DeLP with discourse features indicating an argument strength. This framework includes a finite set of arguments A, disjoint binary relations of attack $R^- \subseteq A \times A$ and support R+ ⊆ A×A, and argument strength function $\tau : A \rightarrow \mathbb{I}$.

Gradual semantics recursively compute an argument's dialectical strength by combining its base score with the aggregated strengths of its attackers and supporters. Given a gradual semantics, such as DF-QuAD (Rago et al. 2016), denoted σ, each argument $\alpha \in A$ obtains a strength $\sigma(\alpha) \in \mathbb{I}$. ValidArgLLM combines discourse-based argument strength with gradual argument semantics in the following way:

1. LLM produces an argumentation tree B whose root is x. Every other node is an argument generated by G. Every node, has a single attacker and one supporter argument pointing to it.
2. Intrinsic argument strength attribution E(B) → Q assigns a base score Q to every node via some evaluator model E.
3. Argumentative strength calculation Σ(Q) → Q(x) applies a gradual semantics σ to add the argument strength to discourse strength, starting from the root claim.
4. Claim verification prediction g(Q(x)) → $\mathbb{I}$ predicts the final result: the claim is true when Q(x)≥0.5 or false otherwise.

The strength aggregation function is defined as $\sigma: \mathbb{I}^* \rightarrow \mathbb{I}$, where for a permutation of arguments $S=(v_1, \ldots, v_n) \in \mathbb{I}^*$: if $n = 0 : \sigma(S) = 0$; if $n = 1 : \sigma(S) = v_1$; if $n = 2 : \sigma(S) = f(v_1,v_2)$
if $n > 2 : \sigma(S) = f(s(v_1, \ldots, v_{n-1}), v_n)$ with the base function $f : \mathbb{I} \times \mathbb{I} \rightarrow \mathbb{I}$ defined, for $v_1, v_2 \in \mathbb{I}$, as:

$$f(v_1,v_2) = v_1 + (1-v_1) \cdot v_2 = v_1 + v_2 - v_1 \cdot v_2$$

Thus, the base function expresses sequences of strengths of attackers or supporters by S, proportionally increasing the attacking or supporting arguments' strength towards 1 (Rago et al 2016)

4. Implementation

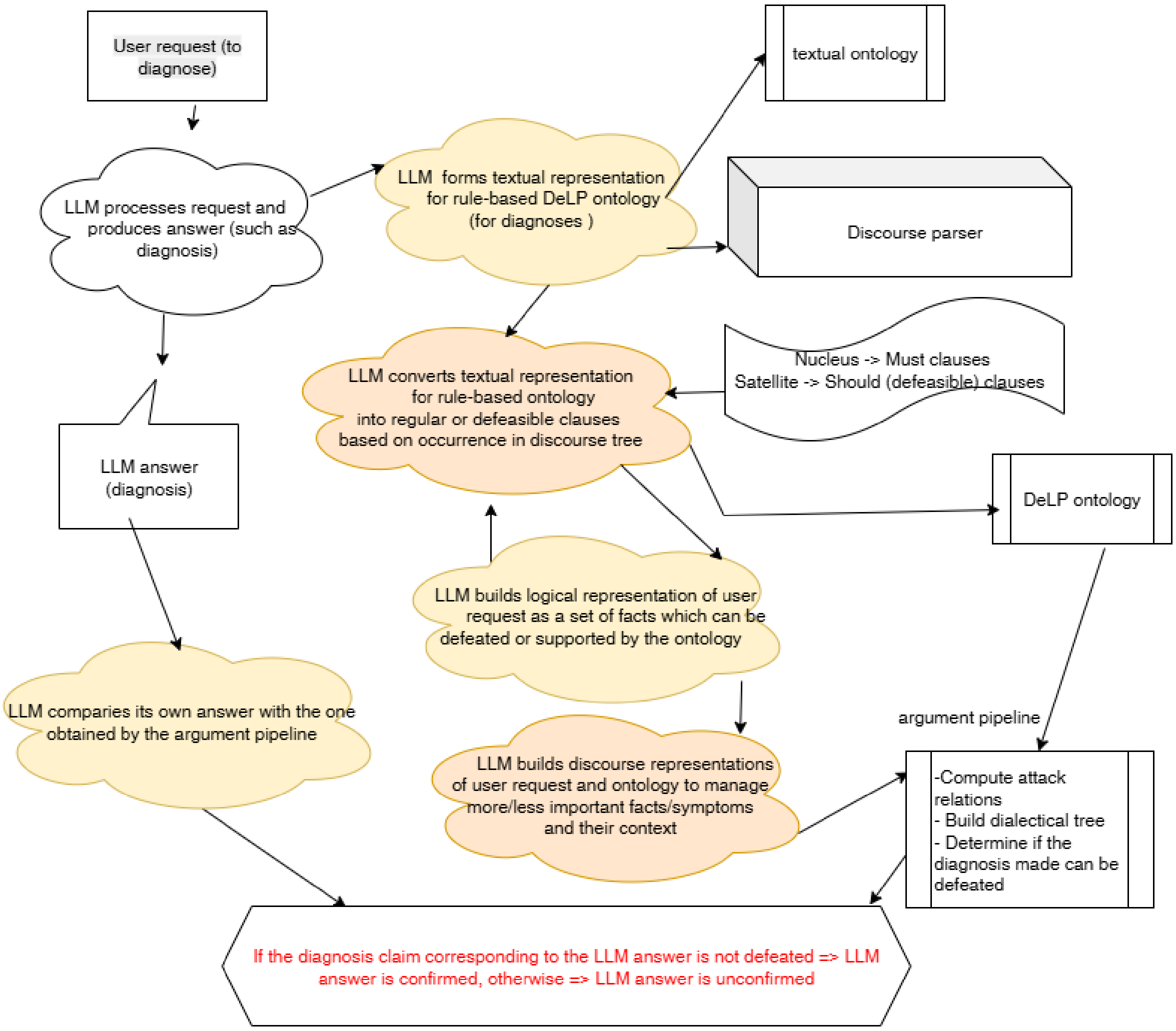

User request (to diagnose)
textual ontology
LLM processes request and produces answer (such as diagnosis)
LLM forms textual representation for rule-based DeLP ontology (for diagnoses)
Discourse parser
LLM converts textual representation for rule-based ontology into regular or defeasible clauses based on occurrence in discourse tree
Nucleus -> Must clauses
Satellite -> Should (defeasible) clauses
LLM answer (diagnosis)
DeLP ontology
LLM builds logical representation of user request as a set of facts which can be defeated or supported by the ontology
LLM comparies its own answer with the one obtained by the argument pipeline
argument pipeline
LLM builds discourse representations of user request and ontology to manage more/less important facts/symptoms and their context
-Compute attack relations
- Build dialectical tree
- Determine if the diagnosis made can be defeated
If the diagnosis claim corresponding to the LLM answer is not defeated => LLM answer is confirmed, otherwise => LLM answer is unconfirmed

Fig. :2 ValidArgLLM system architecture

Fig. 2 illustrates a hybrid neuro-symbolic diagnostic reasoning pipeline where an LLM and a Defeasible Logic Programming (DeLP) reasoning engine work together to verify or refute an LLM-generated medical diagnosis. The goal of the architecture is to ensure that an LLM's diagnostic answer (for example, "The patient has gout") is not only linguistically plausible but also logically justified and consistent with a structured medical ontology. If the logical reasoning pipeline cannot defeat the diagnosis claim, it is confirmed; otherwise, the LLM answer is marked unconfirmed.

Workflow is as follows:

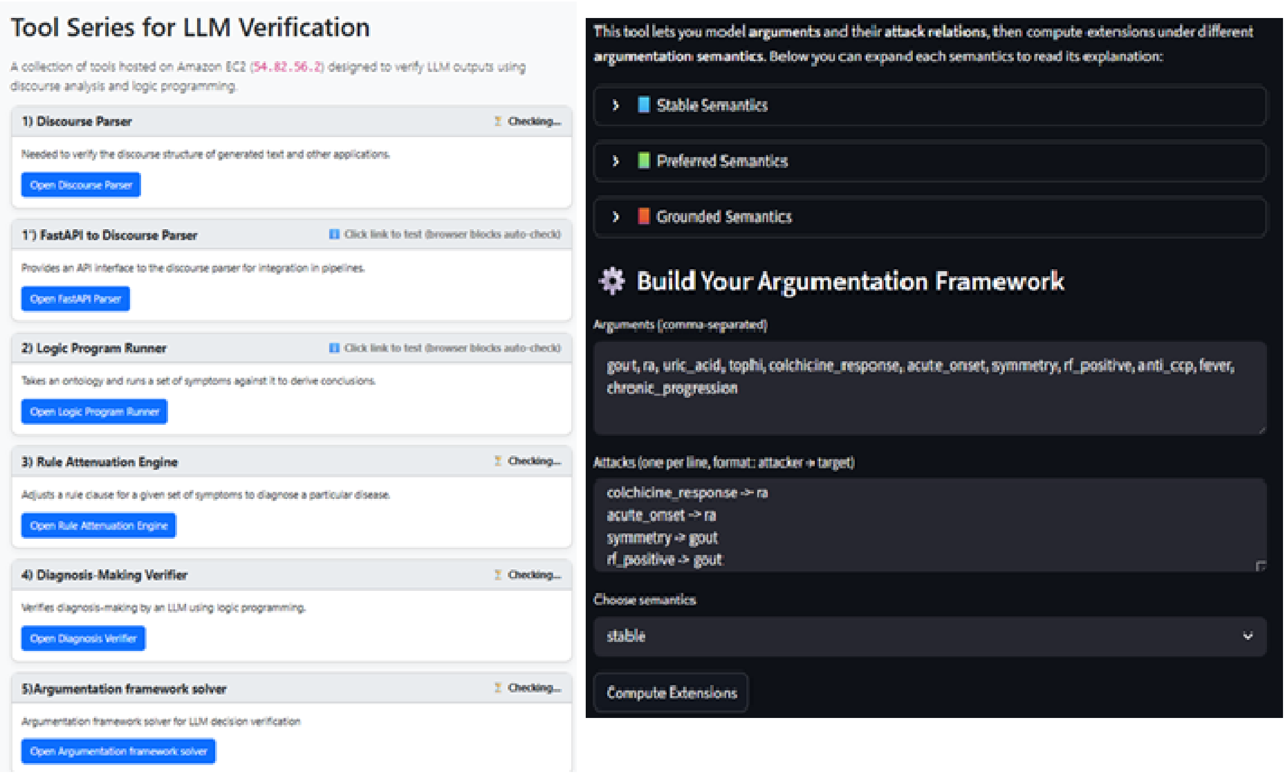
1. User Input. The process begins with a user request, such as asking the model to provide a diagnosis.
2. LLM Generates Initial Answer. The LLM processes the request and outputs an initial diagnosis or conclusion (e.g., "The disease is gout").
3. Ontology and Discourse Setup. A textual ontology of medical knowledge (rules, relationships, symptoms, conditions) and a discourse parser are available. The discourse parser identifies rhetorical relations in the text — for instance, nucleus (main facts) and satellite (contextual or defeasible facts). Nucleus → "Must" clauses (non-defeasible rules) and Satellite → "Should" clauses (defeasible rules)
4. LLM forms ontology representation, transforming textual information into a rule-based ontology in the DeLP format — essentially translating natural-language reasoning into structured logical rules.
5. Conversion to defeasible logic. The LLM converts these rules into regular (strict) or defeasible (soft) clauses depending on their role in the discourse (main vs. secondary information).
6. Building logical representation of the user request. The system formalizes the user's question and the LLM's proposed answer as a set of logical facts that can either be defeated or supported by the ontology.
7. Discourse representation integration. The LLM builds discourse representations of both the user request and the ontology, capturing which arguments are more or less important (nucleus/satellite weighting) and how they relate contextually.
8. Argumentation Pipeline. The argumentation module computes attack relations among rules (contradictions or counter-arguments), dialectical trees, representing possible argumentative dialogues between supporting and opposing claims, and defeasibility outcomes, determining whether a claim survives all counter-arguments
9. Comparison and validation. The LLM compares its original diagnosis with the verified diagnosis obtained through the logical argumentation process.
10. Decision. If the logical reasoning shows that the diagnosis claim is not defeated, it is confirmed as valid. If the claim is defeated by stronger counter-arguments from the ontology, it is marked unconfirmed.

This architecture combines:

- Neural generation (LLM) → producing hypotheses and natural-language reasoning.
- Symbolic verification (DeLP) → testing those hypotheses for logical soundness.
- Discourse analysis → weighting arguments by rhetorical importance.

Together, these components produce a contestable and interpretable diagnostic system, where the model's answer can be justified or overturned through structured reasoning.

The demo of argument-based LLM verifier is available at Tool Series for LLM Verification (Fig. 3 and Fig. 4).



Tool Series for LLM Verification

A collection of tools hosted on Amazon EC2 (54.82.56.2) designed to verify LLM outputs using discourse analysis and logic programming.

1) Discourse Parser ⏳ Checking...

Needed to verify the discourse structure of generated text and other applications.

Open Discourse Parser

1') FastAPI to Discourse Parser ℹ Click link to test (browser blocks auto-check)

Provides an API interface to the discourse parser for integration in pipelines.

Open FastAPI Parser

2) Logic Program Runner ℹ Click link to test (browser blocks auto-check)

Takes an ontology and runs a set of symptoms against it to derive conclusions.

Open Logic Program Runner

3) Rule Attenuation Engine ⏳ Checking...

Adjusts a rule clause for a given set of symptoms to diagnose a particular disease.

Open Rule Attenuation Engine

4) Diagnosis-Making Verifier ⏳ Checking...

Verifies diagnosis-making by an LLM using logic programming.

Open Diagnosis Verifier

5)Argumentation framework solver ⏳ Checking...

Argumentation framework solver for LLM decision verification

Open Argumentation framework solver

This tool lets you model **arguments** and their **attack relations**, then compute extensions under different **argumentation semantics**. Below you can expand each semantics to read its explanation:

› 🟦 Stable Semantics

› 🟩 Preferred Semantics

› 🟥 Grounded Semantics

⚙ Build Your Argumentation Framework

Arguments (comma-separated)

gout, ra, uric_acid, tophi, colchicine_response, acute_onset, symmetry, rf_positive, anti_ccp, fever, chronic_progression

Attacks (one per line, format: attacker -> target)

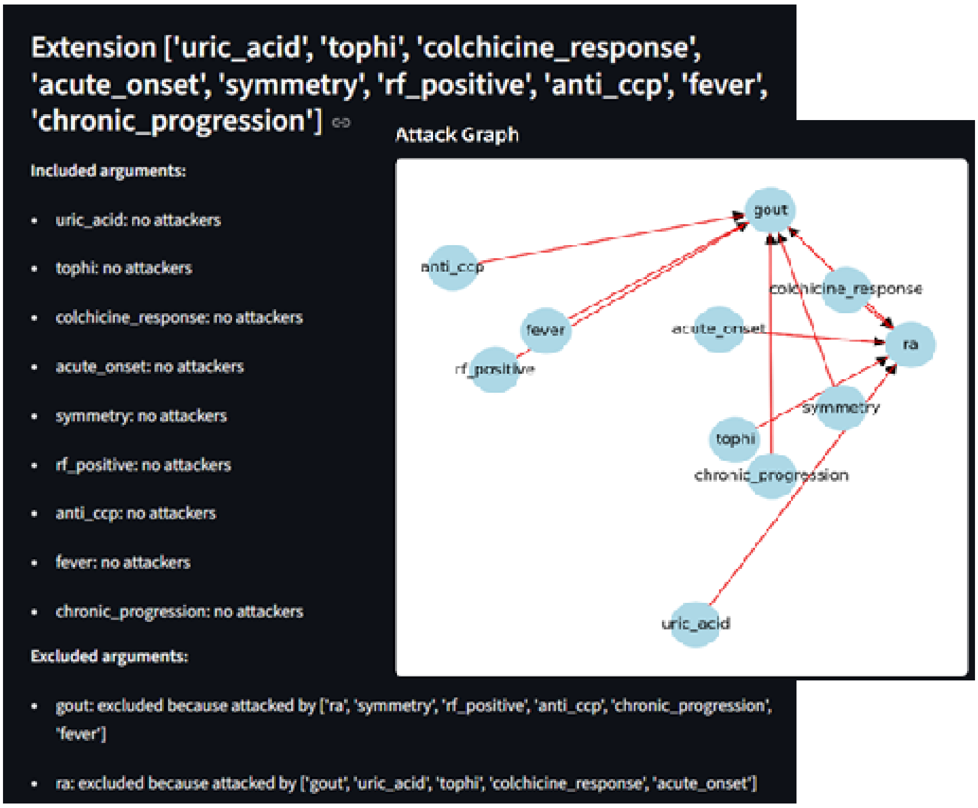
colchicine_response -> ra
acute_onset -> ra
symmetry -> gout
rf_positive -> gout

Choose semantics

stable

Compute Extensions

Fig. 3. A Tool series for logic-based LLM verification (on the left) and argumentation tool (on the right)



Extension ['uric_acid', 'tophi', 'colchicine_response', 'acute_onset', 'symmetry', 'rf_positive', 'anti_ccp', 'fever', 'chronic_progression']

Included arguments:

- uric_acid: no attackers
- tophi: no attackers
- colchicine_response: no attackers
- acute_onset: no attackers
- symmetry: no attackers
- rf_positive: no attackers
- anti_ccp: no attackers
- fever: no attackers
- chronic_progression: no attackers

Excluded arguments:

- gout: excluded because attacked by ['ra', 'symmetry', 'rf_positive', 'anti_ccp', 'chronic_progression', 'fever']
- ra: excluded because attacked by ['gout', 'uric_acid', 'tophi', 'colchicine_response', 'acute_onset']

Fig. 4 Argument pipeline interface

5. Evaluation

We evaluate on three claim-verification datasets that we derive from existing QA/NLI resources: TruthfulHalluc (from TruthfulQA; Lin et al., 2021), MedHalluc (from MedQA; Jin et al., 2020 and PubMedQA; Jin et al., 2019), and eSNLI_Halluc (from eSNLI; Camburu et al., 2018). For each source, we convert items into question–answer (QA) style pairs and then inject controlled inconsistencies by appending randomly sampled, semantically incompatible attributes (facts, circumstances, symptoms). These perturbations create positive “hallucination” cases; unmodified items serve as negatives.

Our focus is hallucination detection for model answers using argumentation analysis as a validator. The validator assesses whether an answer’s central claim is defeated by the argument-validation system. We define a hallucination as a claim whose defeat probability exceeds 0.5. This cautious threshold is motivated by safety-critical domains (health, legal, finance), where we prefer to reject answers that are defeated with substantial probability.

Dataset size and prevalence are as follows. Each hallucination dataset contains 1,000 QA pairs with a 50% hallucination rate (balanced positives/negatives). In the original source datasets the natural hallucination rate is <1%; our perturbation procedure raises prevalence to enable meaningful detection metrics and comparability with prior LLM-argumentation studies.

We report F1 for hallucination prediction in Table 2. It lists: (i) a GPT-5 baseline; (ii) our claim-verification tool that uses argument validation; and (iii) a discourse-aware variant where argument strength additionally incorporates discourse cues beyond the default computation. This final column shows the incremental gains from discourse-informed argument strength.

	GPT-5	GPT-5 +ValidArgLLM	GPT-5 + ValidArgLLM + discourse-based argument strength assessment
TruthfulHalluc	0.53	0.72	0.78
MedHalluc	0.61	0.77	0.80
eSLNI_Halluc	0.49	0.68	0.67

Table 2: Evaluation results of hallucination detection

Our MedHalluc results are broadly comparable to prior work: ArgMed-Agents with GPT-4 reports 0.91 predictive accuracy (Hong et al., 2024); ArgLLM with GPT-4o reports 0.80 (Friedman et al., 2015); and an ensemble of ArgLLMs achieves 0.73 (Ng et al., 2025). That said, these systems estimate claim truthfulness, whereas our study predicts hallucination via whether a claim is defeated by the argument-validation module, so the targets differ and the numbers are not strictly comparable.

6. Related work

The approach of Bezou-Vrakatseli (2023) leverages argument schemes—structured templates that capture common patterns of reasoning—and their associated critical questions, which probe the assumptions, exceptions, and contextual factors underlying those schemes. By using these as a framework for classifying and analyzing arguments, the method provides a semantically richer alternative to surface-level textual analysis. In the context of LLM verification, this enables evaluators to assess not just whether an LLM produces grammatically or factually correct responses, but whether it constructs logically sound, ethically nuanced arguments that align with established norms of rational discourse. The critical questions act as a diagnostic tool, revealing whether the model truly understands the reasoning behind ethical positions or is merely mimicking plausible-sounding rhetoric.

This verification strategy directly supports the project’s broader goal of fostering ethical debate between humans and AI systems. By evaluating LLMs through the lens of argumentation theory, researchers can determine how well these models engage in principled reasoning about moral dilemmas, identify potential biases or logical fallacies, and measure their capacity to both construct and critique ethical arguments. Ultimately, this contributes to "ethics for AI"—ensuring AI systems behave responsibly—and "AI for ethics"—using AI as a tool to help humans reflect on and refine their own ethical reasoning. Such a dual focus positions LLMs not just as information providers, but as collaborative partners in navigating complex moral questions.

Earlier approaches to argumentative reasoning, such as ArgLLMs (Freedman et al., 2025), determined argument quality scores using the confidence of the argument-generating model itself. These systems treated the model's internal probability estimates as proxies for argument plausibility, thereby grounding evaluation in model-intrinsic uncertainty rather than discourse-level coherence. Subsequent research adopted reward-model-inspired scoring (Lambert et al., 2025), introducing an external evaluator LLM to assign quality scores. In this configuration, an independent model was prompted with an argument and its parent node and asked to rate its relevance, factuality, and ambiguity on a normalized [0,1] scale. Two main setups were explored:

1. *Estimated Arguments*, where supporting arguments were scored while the root claim remained fixed at 0.5; and
2. *Estimated All*, where both root claims and subordinate arguments were assessed via discrete truth and certainty labels mapped to continuous scores.

Also, (Ng et al 2025) showed that MArgE can significantly outperform single LLMs, including three open source models (4B to 8B parameters), and existing ArgLLMs, as well as prior methods for unstructured multi-LLM debates.

These techniques diversified the evaluation signal and reduced model-specific bias but remained primarily semantic—focused on the *content* of individual arguments rather than their *rhetorical role* in a discourse structure. A

In contrast, our work introduces a discourse-based scoring framework that evaluates arguments in the context of their rhetorical relations and structural significance within the discourse tree. Instead of relying solely on model-elicited or reward-style judgments, scores are inferred from the hierarchical organization of argumentative elements—linking claims, evidence, and counterarguments through nucleus–satellite dependencies and coherence relations. This approach enables the system to weight contributions based on discourse salience rather than raw textual confidence, thereby capturing how strongly each component supports or undermines the overall claim.

While reward-model scoring focuses on *factual adequacy* and *semantic quality*, our discourse-based approach emphasizes relational justifiability, integrating structural reasoning to yield a more explainable and linguistically grounded measure of argument strength.

7. Conclusions

Argument analysis tool belongs to the series of LLM verification tools including logic programming, answer set programming, and rule attenuation (Fig …) All these tools rely on discourse analysis to determine rule structure and weights to build a logic program representation. We observed that LLMs can be reliably verified by discourse-based argumentation analysis.

Declaration on Generative AI

During the preparation of this work, the author(s) used GPT-5 for grammar and spelling check. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] Bezou-Vrakatseli E (2023) Evaluation of LLM Reasoning via Argument Schemes. Online Handbook of Argumentation for AI, Vol.4 p 1

[2] Louis A and Nenkova A. 2012. A Coherence Model Based on Syntactic Patterns. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1157–1168, Jeju Island, Korea. Association for Computational Linguistics.

[3] Freedman, G.; Dejl, A.; Gorur, D.; Yin, X.; Rago, A.; and Toni, F. 2025. Argumentative Large Language Models for Explainable and Contestable Claim Verification. Proceedings of the AAAI Conference on Artificial Intelligence, 39(14): 14930–14939.

[4] Rago, A.; Toni, F.; Aurisicchio, M.; and Baroni, P. 2016. Discontinuity-Free Decision Support with Quantitative Ar gumentation Debates. In Principles of Knowledge Represen tation and Reasoning: Proceedings of the Fifteenth Interna tional Conference, KR 2016, 63–73.

[5] Ng MP and Junqi Jiang and Gabriel Freedman and Antonio Rago and Francesca Toni. MArgE: Meshing Argumentative Evidence from Multiple Large Language Models for Justifiable Claim Verification, arxiv 2508.02584

[6] Arcuschin, I.; Janiak, J.; Krzyzanowski, R.; Rajamanoharan, S.; Nanda, N.; and Conmy, A. 2025. Chain-of-thought rea soning in the wild is not always faithful. ICLR 2025 Reasoning and Planning for LLMs Workshop.

[7] Barez, F.; Wu, T.-Y.; Arcuschin, I.; Lan, M.; Wang, V.; Siegel, N.; Collignon, N.; Neo, C.; Lee, I.; Paren, A.; et al. 2025. Chain-of-thought is not explainability. Preprint, arXiv.

[8] Lambert, N.; Pyatkin, V.; Morrison, J.; Miranda, L. J. V.; Lin, B. Y.; Chandu, K. R.; Dziri, N.; Kumar, S.; Zick, T.; Choi, Y.; Smith, N. A.; and Hajishirzi, H. 2025. Reward Bench: Evaluating Reward Models for Language Modeling. In Findings of the Association for Computational Linguis tics: NAACL 2025, 1755–1797.

[9] Kaminski, Roland & Wanko, Philipp. (2017). A Tutorial on Hybrid Answer Set Solving with clingo. In: Reasoning Web. Semantic Interoperability on the Web (pp.167-203)

[10] Garcia, A., Simari, G., 2004. Defeasible logic programming: an argumentative approach. Theory Pract. Log. Program. 4, 95–138.

[11] Ferrag MA, Norbert Tihanyi, Merouane Debbah, Reasoning beyond limits: Advances and open problems for LLMs, ICT Express, 2025.

[12] Galitsky B (2025) Enabling large language model with plug-and-play symbolic reasoning components. In Health Apps of Neuro-symbolic AI, Elsevier pp 59-80.

[13] Lin S, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. CoRR, abs/2109.07958, 2021.

[14] Jin, B. Dhingra, Z. Liu, W. W. Cohen, and X. Lu, “Pubmedqa: A dataset for biomedical research question answering,” 2019.

[15] Jin D, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? A large-scale open domain question answering dataset from medical exams. CoRR, abs/2009.13081, 2020.

[16] Hong S, Liang Xiao, Xin Zhang, Jianxia Chen (2024) ArgMed-Agents: Explainable Clinical Decision Reasoning with LLM Disscusion via Argumentation Schemes

[17] Camburu O-M, Tim Rocktäschel, Thomas Lukasiewicz, Phil Blunsom (2018) e-SNLI: Natural Language Inference with Natural Language Explanations. Advances in Neural Information Processing Systems 31 (NeurIPS 2018)