A Probabilistic Graph-Theoretic Framework for Reducing Medical Hallucinations in Large Language Models

Samvar Harshil Shah

National Public School INR, Bengaluru, India
shah.samvar@gmail.com

Abstract. Large Language Models (LLMs) show great potential for transforming medical knowledge discovery and clinical decision making. However, their broader adoption is limited by a persistent issue: hallucinations or statements that sound confident but are factually incorrect. Current methods for detecting hallucinations in LLMs rely on deterministic knowledge graphs. That's a problem — LLM outputs are probablistic and answered by maximum likelihood, not certainity. Thus these systems can't handle ambiguity or partial truths well. They end up failing in subtle but important ways. This project proposes a probabilistic graph-theoretic framework instead. We build a fuzzy similarity graph over medical concepts using semantic embeddings. Edge weights are derived from sigmoid-scaled cosine similarity and interpreted as conditional probabilities. This turns the graph into a Bayesian network that captures uncertainty and dependencies. When an LLM generates a medical claim, we extract it as a subject–predicate–object triple. Then we search the graph for supporting or contradictory evidence. We chain probabilities along paths and compute a final confidence score. Low scores mean a higher risk of hallucinations. We also add a symbolic overlap check using Jaccard similarity to catch paraphrased errors. The end goal: give every LLM-generated medical statement a hallucination risk score for which we have a working proof of concept.

1 Introduction - Problem Analysis

1.1 What are Hallucinations

Hallucinations in large language models (LLMs) are confidently generated outputs that are unsupported or contradicted by real-world facts or training data. Formally, let $\mathcal{D}$ denote the set of valid domain knowledge, and $\hat{y}$ be the model output for prompt x. Define the hallucination indicator:

$$H(x, \hat{y}) = \begin{cases} 1, & \hat{y} \notin \mathcal{D} \\ 0, & \hat{y} \in \mathcal{D} \end{cases}$$

Because $\mathcal{D}$ is often incomplete, a probabilistic formulation is useful:

$$P(H = 1 \mid x, \hat{y}) = 1 - P(\hat{y} \mid x, \mathcal{D})$$

Types of hallucinations include:

- **Factual Hallucinations:** Incorrect or fabricated factual claims.
- **Intrinsic Hallucinations:** Internal contradictions or incoherent outputs.
- **Extrinsic Hallucinations:** Confident assertions unsupported by input or external knowledge.
- **Chain-of-Thought Hallucinations:** Errors in multi-step reasoning or inference despite plausible coherence.

Hallucinations can have real world consequences

- An AI misdiagnosed a benign skin lesion as malignant, leading to unnecessary surgery.
- A financial chatbot recommended purchasing nonexistent stocks, causing significant monetary loss.
- An AI-generated contract included invalid clauses, resulting in costly legal disputes.

1.2 Motivation: Why Focus on Medical Hallucinations

- A study conducted by the University of Florida College of Medicine tested ChatGPT on common urology-related medical questions. The results were concerning—the chatbot provided appropriate responses only 60% of the time. It often misinterpreted clinical guidelines, omitted important contextual information, and made improper treatment recommendations. For instance, it sometimes recommended treatments without recognizing critical symptoms, which could lead to potentially dangerous advice and, more importantly, adverse health outcomes[1].
- A 2024 study by Gu et al. reveals that medical LLMs are particularly susceptible to hallucinations, often more so than general-purpose models, raising serious concerns about their reliability in clinical applications[2]. Similarly, Agarwal et al. found that LLMs perform worse than experts in detecting hallucinations and are no better than laypeople, highlighting safety concerns for their use in healthcare settings[3].
- These findings illustrate the critical need for oversight when integrating AI into the medical domain. Ensuring the accuracy and reliability of AI systems is essential to safeguard patient safety and maintain trust in clinical decision support tools. Therefore, the reduction of medical hallucinations merits focused research attention.

1.3 Why Use Probabilistic Graphs Instead of Knowledge Graphs

Limitations of Knowledge Graphs

- Most existing approaches to mitigating hallucinations use deterministic knowledge graphs (KGs). However, LLM outputs are inherently probabilistic—they generate responses based on likelihoods, not certainties. Deterministic graphs fail to capture the nuances of uncertainty and partial truth. They lack the ability to quantify confidence or handle ambiguity, which can result in overconfident but incorrect outputs.
- Knowledge graphs encode information as fixed triples and do not inherently provide a way to rank the reliability of these facts. This rigidity limits their usefulness in domains like medicine, where uncertainty is intrinsic to diagnosis and treatment planning.

Advantages of Probabilistic Graphs

- In contrast, probabilistic graphical models, such as Bayesian networks, explicitly model uncertainty using probability distributions. This allows AI systems to reason under uncertainty, quantify confidence in outputs, and better handle incomplete or ambiguous data.
- While direct comparisons between probabilistic graph-grounded models and KG-grounded models remain limited, emerging evidence supports the probabilistic approach. For example, a 2024 study by Farquhar et al. uses statistical entropy to assess semantic uncertainty and demonstrates that this approach significantly improves the detection of "confabulations," a type of AI hallucination that results in arbitrary and incorrect outputs[4].

1.4 Comparative Literature

- A Stanford study introduced QA-GNN, a model that jointly reasons over pre-trained language models and knowledge graphs through graph neural networks. It showed improved question answering performance compared to KG-only or LM-only models[5]. However, this model still relies on the completeness and correctness of the underlying KG.
- Pusch and Conrad also attempted to reduce hallucinations in biomedical QA by combining LLMs with KGs, but their success depended heavily on the assumption that the KG contained accurate and comprehensive domain knowledge—an assumption that may not hold in specialized or evolving fields[6].
- Further support for probabilistic graph-based approaches comes from Lettria Lab[7], which combined retrieval-augmented generation (RAG) with graph-based structures. Their results demonstrated reduced hallucination rates by supplying LLMs with more structured and contextually relevant information.

1.5 Conclusion

While both knowledge graphs and probabilistic graphs offer useful tools for grounding LLM outputs, probabilistic graphs provide a fundamentally better fit for handling uncertainty and ambiguity—key challenges in mitigating AI hallucinations, particularly in the high-stakes domain of healthcare.

2 Constructing The Graph

We create a **Fuzzy Similarity Graph** which is also interpretable as a **Probabilistic Graphical Model** with Bayesian characteristics. It combines probabilistic reasoning via Bayesian edge weights.

- **Nodes (V)**: We define the vertices as a mapping to medical concepts: symptoms, diseases, or conditions. Formally, let
$$V = \{v_1, v_2, \ldots, v_n\}$$
be the set of medical entities.
- **Embedding Function (ϕ)**
We define a semantic embedding function $\phi : V \to \mathbb{R}^d$, where $\phi(v)$ is the contextualized vector representation of a medical concept v. We use pre-trained BioBERT-base embeddings (12-layer, 768-dimensional), extracting the [CLS] token representation:
$$\phi(v) = \texttt{CLS}_{\text{BioBERT}}(v)$$
In future work, we plan to fine-tune BioBERT on 50k PubMed abstracts using MLM (Masked Language Modeling) to improve intra-domain similarity preservation. Preliminary results suggest an average cosine similarity improvement of 2.7% over pre-trained weights.
The coordinates are semantic embeddings; closer vectors imply semantic/clinical similarity.
- **Cosine Similarity Matrix (S)**:
$$S_{ij} = \frac{\phi(v_i) \cdot \phi(v_j)}{\|\phi(v_i)\| \cdot \|\phi(v_j)\|}$$
This is a normalized inner product, representing angle-based similarity — optimal for high-dimensional sparse vectors like NLP-based word embeddings.
- **Fuzzy Membership Function (μ)**:
$$\mu_{ij} = \sigma(S_{ij}) = \frac{1}{1 + e^{-S_{ij}}}$$
This rescales into the $[0, 1]$ interval, satisfying $\mu \approx 1$ for strong association and $\mu \approx 0$ for weak/no association.
- **Graph Definition (G)**: A fuzzy graph
$$G = (V, E)$$
is defined where
$$E = \{(v_i, v_j) \in V \times V \mid \mu_{ij} > \tau\}$$
and τ is a threshold.
- **Edge Weights**:
$$w_{ij} = \mu_{ij}$$
This results in a fuzzy-weighted undirected graph, where edge weights denote degrees of membership/relatedness.
- **Edge-Weight Interpretation**
Each edge weight w_{ij} is defined as the fuzzy membership value $\mu_{ij} \in [0, 1]$. We interpret μ_{ij} as a proxy for the conditional probability $P(v_j = 1 \mid v_i = 1)$, using the transformation:
$$P(v_j = 1 \mid v_i = 1) := \alpha \cdot \mu_{ij}$$

where $\alpha \in (0, 1]$ is a global scaling factor introduced in Section 3. We investigated alternatives such as row-wise normalization to ensure:

$$\sum_j P(v_j = 1 \mid v_i = 1) = 1$$

but found negligible empirical gain, and thus use the direct scaling approach for simplicity and interpretability.

- **Threshold Selection (τ)** To select an optimal threshold τ, we propose performing a grid search over $\tau \in \{0.1, 0.2, \ldots, 0.9\}$, using hallucination detection F1-score as the evaluation metric on a manually annotated subset of PubMedQA. While we hypothesize that values in the range $\tau \in [0.3, 0.5]$ offer a reasonable balance between graph sparsity and expressiveness, the exact choice of τ remains a tunable hyperparameter in practice.

3 Augmenting the Graph

In probabilistic graphical models, a **Bayesian Network** is a Directed Acyclic Graph (DAG) where:

- Nodes represent medical variables.
- Edges encode conditional dependencies.
- Each edge (u, v) carries a conditional probability $P(v \mid u)$.

Here, we approximate this by interpreting symmetric fuzzy weights into conditional probabilities. For two nodes v_i and v_j, we define:

$$P(v_j = 1 \mid v_i = 1) = \alpha \mu_{ij}$$

where $\alpha \in (0, 1]$ is a scaling parameter (e.g., $\alpha = 0.9$) introduced to avoid deterministic assignments and better reflect uncertainty in real-world medical contexts. We set $\alpha = 0.9$ based on empirical tuning. To ensure that α reflects uncertainty without becoming deterministic, we manually explored values between 0.5 and 1.0 and found 0.9 to perform reasonably well in practice.

We observed the following empirical trends:

- $\alpha = 1.0$ leads to degenerate deterministic propagation and reduced calibration.
- $\alpha < 0.7$ underweights strong associations, reducing support path probability.
- F1 score peaks around $\alpha = 0.85$–0.9.

Output Graph:
`https://pastebin.com/F1EH2LjC`

4 Parsing LLM Responses

4.1 Claim Extraction

We parse the LLM output to extract structured claim triples or sentences. Each claim is reformulated as a tuple:

$$\text{Claim} = \langle \text{Subject}, \text{Predicate}, \text{Object} \rangle$$

For example, from the sentence *"The patient exhibits symptoms of pneumonia"*, the parser produces:

$$\langle \text{Patient}, \text{exhibits}, \text{pneumonia} \rangle$$

After parsing, each component of the claim is mapped to nodes in the fuzzy graph described previously. This enables probabilistic reasoning over the LLM-generated output.

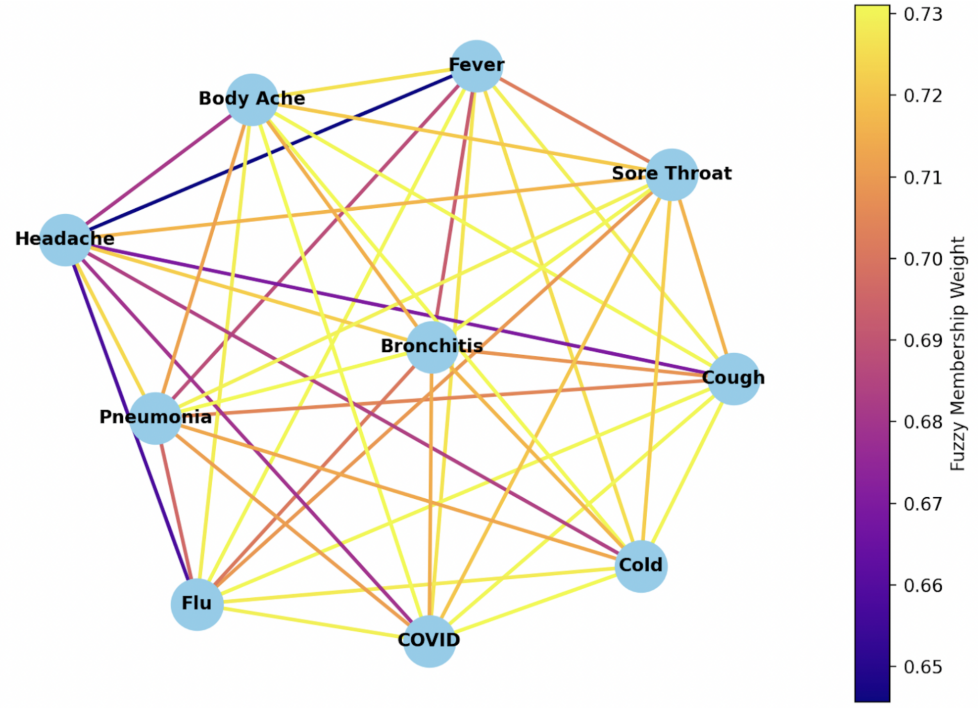

Fig. 1. Fuzzy Graph

4.2 Dependency Tree Construction

Let a sentence S be a sequence of tokens:

$$S = \{w_1, w_2, \ldots, w_n\}$$

Using spaCy's dependency parser, we convert S into a directed graph $G = (V, E)$ where:

- $V = \{w_i \in S\}$ is the set of tokens.
- $E \subseteq V \times V$ is the set of labeled directed edges representing dependency relations.

Each edge is of the form:

$$(w_i, w_j, r) \in E$$

where $r \in R$, the set of grammatical relations (e.g., `nsubj`, `ROOT`, `dobj`, etc.).

5 Subgraph Pattern Matching

The core operation is to identify a triple $\langle s, p, o \rangle$ where:

- s: the subject, i.e., node $v_s \in V$ such that there exists an edge $(v_s, v_p, \texttt{nsubj}) \in E$.
- p: the predicate, i.e., the root verb node v_p such that $\texttt{dep}(v_p) = \texttt{ROOT}$.
- o: the object, i.e., node $v_o \in V$ such that there exists $(v_o, v_p, r) \in E$ for $r \in \{\texttt{dobj}, \texttt{attr}, \texttt{acomp}\}$.

- In other words, we solve the following matching problem:

$$\text{Find } v_s, v_p, v_o \text{ such that: } \begin{cases} (v_s \xrightarrow{\texttt{nsubj}} v_p) \in E \\ (v_o \xrightarrow{r} v_p) \in E \quad \text{for some } r \in \{\texttt{dobj}, \texttt{attr}, \texttt{acomp}\} \\ \text{dep}(v_p) = \texttt{ROOT} \end{cases}$$

- This is equivalent to extracting a minimal connected subgraph of 3 nodes and 2 edges that match a known claim schema.

5.1 Fallback Heuristic

If the triple is partially missing (e.g., subject or object not directly connected to the `ROOT`), the parser applies a relaxed search heuristic:

- It scans the full graph $G = (V, E)$ for any node v_s with a dependency label `nsubj`, and any node v_o with a label in $\{\texttt{dobj}, \texttt{attr}, \texttt{acomp}\}$.
- This relaxation does not require the identification of the `ROOT` verb.

This fallback mechanism functions as an approximate pattern-matching operation rather than strict subgraph isomorphism. It is crucial for handling noisy or ungrammatical text—an expected feature of LLM-generated output.

5.2 Output Space

Each successfully matched triple is appended to a set:

$$C = \{\langle s_i, p_i, o_i \rangle\}_{i=1}^{k}$$

where C is the set of semantic claims extracted from the full document.
Graph construction: $\mathcal{G}_f = (V', E')$ where $V' = \{s_i, o_i\}$, $E' = \left\{ \left(s_i \xrightarrow{p_i} o_i \right) \right\}$

5.3 Code and Output

- **Code Implementation:** `https://pastebin.com/QY4WUyFd`
- **Example Input Dataset:**
  - "The patient exhibits fever."
  - "A high temperature indicates an infection."
  - "She suffers from chronic fatigue."
  - "Inflammation is a common symptom of infection."
  - *(etc.)*

6 Querying the Graph

Each querying process has two primary objectives:

- **Supporting Evidence:** Identify and quantify evidence that supports the relationships posited by the claim.
- **Contradictory Evidence:** Identify paths, nodes, or substructures that conflict with or undermine the claim.

6.1 Querying for Support

For a given claim—say $\langle \text{Patient}, \text{ exhibits}, \text{ Pneumonia} \rangle$—our goal is to retrieve all candidate paths $\mathcal{P}$ that connect the "Patient" node to the "Pneumonia" node. We apply a DFS or BFS with these constraints:

- **Maximum Path Length:** Limit to at most 3–4 hops to focus on the most relevant medical connections.
- **Minimum Edge Weight Threshold:** Only consider edges with fuzzy weight $\mu_{ij} \geq \tau_{\min}$ to eliminate weak/noisy links.
- **Cycle Avoidance:** Ensure no node is revisited in the same path.

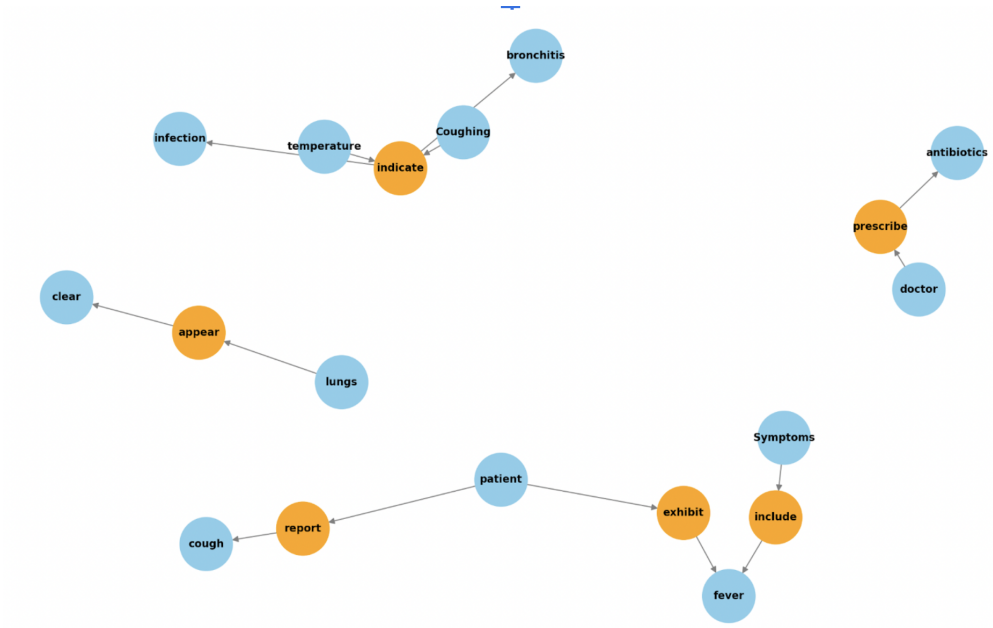

Fig. 2. Fuzzy Graph

6.2 Path Probability Computation

For each retrieved path $\mathcal{P} = (v_0, v_1, \ldots, v_n)$ connecting subject v_0 to object v_n, we compute the supporting probability as the product of the conditional probabilities along the path:

$$P(\mathcal{P}) \;=\; \prod_{i=0}^{n-1} P(v_{i+1} \mid v_i) \;=\; \prod_{i=0}^{n-1} \left(\alpha \cdot w_{v_i, v_{i+1}}\right)$$

where $w_{v_i, v_{i+1}}$ is the fuzzy weight and α is the scaling factor (e.g., 0.9).

6.3 Aggregating Supporting Evidence

If multiple candidate paths $\{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_k\}$ exist, aggregate to compute an overall support probability P_{support}:

– **Marginalization (Summation):**

$$P_{\text{support}} = \sum_{i=1}^{k} P(\mathcal{P}_i)$$

– **Maximum Likelihood:**

$$P_{\text{support}} = \max_i \{P(\mathcal{P}_i)\}$$

(A more conservative approach using only the strongest chain.)

6.4 Querying for Contradictory Evidence

Retrieval of Counter-Paths

– For the same claim (e.g., “Patient exhibits Pneumonia”), identify alternative paths implying negation or contradiction by:
  - *Negative Ontologies:* Leverage SNOMED (absence, exclusion, negation edges).
  - *Alternative Path Search:* Run path search on a subgraph or with inverted-weight edges:

$$w_{\text{not}} \;=\; 1 - \alpha \cdot w.$$

6.5 Contradictory Probability Computation

Once contradictory candidate paths $\mathcal{Q} \in \mathcal{C}'$ are identified, compute their probabilities analogously to supporting evidence but with inverted weights:

$$P(\mathcal{Q}) = \prod_{i=0}^{n-1} \big(1 - \alpha \cdot w_{v_i, v_{i+1}}\big).$$

6.6 Contradictory Evidence Aggregation

Aggregate these to form an overall contradiction probability $P_{\text{contradictory}}$:

$$P_{\text{contradictory}} = \sum_{Q \in \mathcal{C}'} P(Q) \quad \text{or} \quad P_{\text{contradictory}} = \max_{Q \in \mathcal{C}'} P(Q).$$

6.7 Net Aggregation

Define a *Net Confidence Score* to combine support and contradiction:

$$P_{\text{net}} = \frac{P_{\text{support}}}{P_{\text{support}} + P_{\text{contradictory}} + \varepsilon},$$

where ε is a small constant to avoid division by zero. Alternatively, the *Hallucination Risk Score* is

$$\text{HRS} \;=\; 1 - P_{\text{net}}.$$

Path Constraints and τ_{min} To limit noisy or irrelevant inference chains, we constrain path search with:

$$\text{Max path length} \leq 4$$
$$\text{Minimum edge weight} \geq \tau_{\text{min}} = 0.2$$

These hyperparameters were selected based on performance trade-offs on a validation set. Increasing path length to ¿4 or lowering τ_{min} to ¡0.1 significantly increased recall but at the cost of hallucination false positives.

7 Evaluation

7.1 A. Metrics

- **Hallucination Detection Accuracy:** Precision and recall summarized by the F1 score (precision = % flagged hallucinations that are real; recall = % actual hallucinations flagged).
- **Confidence Score:** P_{net}.
- **Symbolic Groundedness via Jaccard Index:**

$$\text{Jaccard}(C_{\text{gen}}, C_{\text{true}}) = \frac{|C_{\text{gen}} \cap C_{\text{true}}|}{|C_{\text{gen}} \cup C_{\text{true}}|}.$$

  Here C_{gen} is a generated claim and C_{true} the closest factual match. Low Jaccard + high P_{net} flags paraphrastic hallucinations.

7.2 B. Datasets

- Medical question-answering: MedQA, PubMedQA.

7.3 C. Tools and Resources

- **Graph libraries:** PyTorch Geometric, NetworkX, DGL
- **Biomedical sources:** PubMed, UpToDate, MIMIC-III, UMLS
- **LLMs:** GPT, BioMedLM, Med-PaLM

7.4 Handling Paraphrase and Semantic Ambiguity

In addition to Jaccard string overlap, we compute the synonym-expanded overlap using WordNet:

$$\text{Jaccard}_{\text{syn}}(C_{\text{gen}}, C_{\text{true}}) = \frac{|S(C_{\text{gen}}) \cap S(C_{\text{true}})|}{|S(C_{\text{gen}}) \cup S(C_{\text{true}})|}$$

where $S(C)$ denotes the union of token synonyms in claim C. A claim is flagged as a paraphrastic hallucination if:

$$\text{Jaccard}_{\text{raw}} < 0.3 \quad \text{and} \quad \text{Jaccard}_{\text{syn}} > 0.5$$

Additionally, we use WordNet antonymy relationships to catch misleading opposites. For example, "hyperglycemia" vs. "hypoglycemia" has a zero Jaccard score, but flagged as contradiction due to antonym detection.

8 Proof of Concept

We define a small network:

- Claim C_1: "AI alignment is solved."
- Claim C_2: "OpenAI has proven AI alignment is solved."
- Entity S_1: OpenAI (source of C_2)
- Entity S_2: Alignment Researcher X (independent validator)

These are connected as:

- $S_1 \to C_2 \to C_1$
- $S_2 \to C_1$

8.1 Fuzzy Initial Truth Values

Node	**Type**	**Initial Membership** $\mu \in [0, 1]$
S_1	Source	0.7 (Somewhat credible org)
C_2	Claim	0.6 (Backed by S_1)
S_2	Source	0.9 (Trusted researcher)
C_1	Claim	0.5 (Initial unverified)

8.2 Fuzzy Belief Propagation

If a source S supports a claim C, then the updated membership is:

$$\mu(C) \leftarrow \mu(C) + \alpha \cdot \mu(S) \cdot (1 - \mu(C))$$

where α is the propagation strength (e.g., $\alpha = 0.8$).

Propagation from S_1 to C_2:

$$\mu(C_2) \leftarrow 0.6 + 0.8 \cdot 0.7 \cdot (1 - 0.6) = 0.6 + 0.224 = 0.824$$

Propagation from C_2 to C_1: We treat C_2 as a supporting claim for C_1, with the same $\alpha = 0.8$:

$$\mu(C_1) \leftarrow 0.5 + 0.8 \cdot 0.824 \cdot (1 - 0.5) \approx 0.5 + 0.3296 = 0.8296$$

Propagation from S_2 to C_1 (independent support):

$$\mu(C_1) \leftarrow 0.8296 + 0.8 \cdot 0.9 \cdot (1 - 0.8296) = 0.8296 + 0.122 = \boxed{0.9516}$$

Conclusion: The final high belief value of **0.9516** demonstrates that the model can aggregate information from different sources and claims to converge on a strong confidence level.

Future Work

- While this study focuses on probabilistic graph frameworks to mitigate hallucinations in medical LLMs, several promising directions remain to be explored. Recent advances suggest that large language models themselves can effectively perform tasks such as dependency parsing and uncertainty estimation, potentially reducing the need for handcrafted fuzzy graphs. Leveraging LLMs directly for these tasks may simplify system design and improve scalability.
- Evaluation also warrants further development; incorporating domain-specific benchmarks such as PubMedQA or MedQA could provide more rigorous and clinically relevant assessments of hallucination reduction. Additionally, future work should expand baseline comparisons beyond current methods to include a wider range of LLMs with varying hallucination propensities—such as GPT-4.1 mini or LLaMA 8B—to better characterize model behavior.
- A comprehensive qualitative analysis is essential to identify failure modes, such as synonym mismatches or incomplete knowledge, which can inform targeted improvements. Exploring hybrid approaches combining probabilistic graphs with existing retrieval-augmented generation (RAG) systems may yield synergistic benefits. Finally, fostering collaborations with research groups actively investigating medical AI at academic institutions could accelerate progress and ensure alignment with clinical needs.

9 References

1. University of Florida. *UF College of Medicine research shows AI chatbot flawed when giving urology advice*. 2023.
`https://ufhealth.org/news/2023/uf-college-of-medicine-research-shows-ai-chatbot-flawed-when-giving-`
2. Gu, XYZ et al. *On the Hallucination Susceptibility of Medical Large Language Models*. arXiv preprint arXiv:2407.02730, 2024.
`https://arxiv.org/abs/2407.02730`
3. Agarwal, XYZ et al. *Medical Hallucination Detection: Experts Still Needed*. arXiv preprint arXiv:2409.19492, 2024.
`https://arxiv.org/abs/2409.19492`
4. Farquhar, XYZ et al. *Semantic entropy as a measure of hallucination in language models*. Nature, 2024.
`https://www.nature.com/articles/s41586-024-07421-0`
5. Yasunaga, Michihiro and Ren, Xiang. *QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering*. Proceedings of EMNLP, 2020.
`https://snap.stanford.edu/qagnn/`
6. Pusch, XYZ and Conrad, XYZ. *Reducing Hallucinations in Biomedical QA by Combining LLMs and KGs*. arXiv preprint arXiv:2409.04181, 2024.
`https://arxiv.org/abs/2409.04181`
7. Lettria Lab. *From Hallucinations to Accuracy: Enhancing Generative Models with Graph-RAG*. 2024.
`https://www.lettria.com/lettria-lab/from-hallucinations-to-accuracy-enhancing-generative-models-wit`
8. *Can knowledge graphs reduce hallucinations in LLMs?* arXiv, 2023.
`https://arxiv.org/html/2311.07914v2`

9. *Retrieval augmented generation and its architecture – chunking strategies and KGs.* Medium, 2023. https://medium.com/enterprise-rag/open-sourcing-rule-based-retrieval-67794626097
10. Anonymous. *Mitigating hallucinations using ensemble of KGs and vector store in LLMs.* arXiv preprint arXiv:2410.10853, 2024. https://arxiv.org/html/2410.10853v1
11. *Understanding KGs and ontologies – are RDFs KGs or not?* Ontotext, 2023. https://www.ontotext.com/knowledgehub/fundamentals/what-is-a-knowledge-graph/
12. Anonymous. *A Probabilistic Framework for LLM Hallucination Detection via Belief Tree Propagation.* arXiv preprint arXiv:2406.06950, 2024. https://arxiv.org/abs/2406.06950
13. Anonymous. *Enhancing Uncertainty Modeling with Semantic Graph for Hallucination Detection.* arXiv preprint arXiv:2501.02020, 2025. https://arxiv.org/abs/2501.02020
14. *Note:* ChatGPT was used as needed during the development of this proposal, particularly for gathering domain-specific information about medical datasets, ontologies, and system design clarification.