A Bayesian Inspired Knowledge Graph Framework for Uncertain Data: Design and Comparative Analysis

Amitai Goldmeer
Temple University
Philadelphia, PA
amitai.goldmeer@temple.edu

Abstract

Knowledge Graphs are widely used to represent structured information, but often assume binary truth values for entities and relationships, which limits applicability for domains where uncertain data is widespread. This paper presents a Bayesian Knowledge Graph (BKG) framework as a Bayesian inspired approach which alters the traditional subject-predicate-object triple format with confidence values and applies Bayesian style belief updating to define relationship strength overtime. Utilizing alpha/beta distributed priors, the system updates and propagates node & edge confidence, edge beliefs, and node reliability as new evidence enters the system. This framework for handling uncertain data is compared against the Non-Axiomatic Reasoning System (NARS) as an alternative system for handling uncertain data, due to its role as an inspiration for this project, and its status as a more mature framework for handling uncertain data in practice.

1 Introduction

Knowledge Graph systems are traditionally systems with a binary understanding of truth values. If an entity or relationship exists inside the graph structure, then it exists, else it does not exist. While this type of data structure has its usage in traditional knowledge spaces & in the AI space, it leaves room for improvement or change. The failure of traditional KG systems to handle uncertain data leaves much to be desired, especially as the connection and visualization of data in spaces that use uncertain data could be highly beneficial.

The goal of the Bayesian Knowledge Graph (BKG) project was to attempt the construction of a knowledge graph system that could support uncertain data and utilize the confidence levels of all data within the system to better estimate the certainty of that data via Bayesian statistics. Unlike a traditional Bayesian Network, it does not utilize conditional probability tables, enforce acyclic graphs, or compute global probability distribution, instead performing localized belief aggregation which relies on the topology of the graph for propagation.

2 Related Works

2.1 Knowledge Graphs

Knowledge Graphs (KGs) represent structured information as subject-predicate-object triples where nodes are entities and directed edges hold semantic relationships between those entities[1]. By allowing for semantic connection of data, it separates itself from Relational Databases (Ex: MySQL) though interconnectedness of all data instead of being largely separated by tables, only coming together in the case of joins or subtables. Instead, any node has the potential to be directly linked to any other node, and relationships or causality can be determined from underlying properties. Often, 'ontologies' are seen as part of a knowledge graph as well. An ontology can be seen as a blueprint for how a knowledge graph is built, creating a standard set of usable labels (edge types), classes (node types) and metadata properties.

KGs have applications in a number of settings, historically in search engines and answering systems. In search engines, they have been used to enhance search results by providing structured information about entities and improving search relevance by providing semantic understanding of related terms (query expansion). More recently, they have been used in enterprise use-cases to formalize complex structures and specialized product-domains. Additionally, they have been integrated into AI structures to provide grounding and semantic understanding for LLMs alongside vector databases.

2.2 Uncertainty in Knowledge Graphs

Representing uncertainty in Knowledge Graphs is an ongoing field of research with numerous approaches on how to handle construction, data conflict, and knowledge alignment [2]. One proposal was *UKGE*[3], a KG embedding model meant to preserve structural and confidence data via an embedding space, as opposed to previous projects which only operated with binary relation classification. More recently, a proposal for a framework that generated prediction intervals over uncertain Knowledge Graph embeddings (like the ones generated by UKGE) was published, which would attempt to address the gap between confidence estimates created by embeddings and legitimate uncertainty quantification [4]. The BKG described in this paper takes an approach to the same goal from a different angle, performing localized belief updates instead of attempting to learn embeddings.

2.3 Bayesian Networks and Belief Propagation

The theoretical basis of this project pulls from Bayesian inference and graphical models. *Probabilistic Reasoning in Intelligent Systems*[5] establishes the foundations for belief propagation and how uncertainty can be spread across a network. Traditional Bayesian Networks operate as directed acyclic graphs (DAGs), which the BKG system diverges from. The local propagation design draws from literature surrounding belief propagation [6], which extends the initial algorithms for graphs that contain cycles. While value convergence is not guaranteed in graphs with cycles, the usage of depth-limited propagation acts as an effective heuristic. The BKG implements this depth-bounded propagation instead of full 'message-passing', creating a simple and compatible form of belief propagation that works well on all types of graph structure. The alpha/beta updating at the core of the project is grounded in the beta-binomial conjugate relationship. The *Compendium of Conjugate Priors*[7] and MIT course materials on probability and statistics [8] establish why these beta distributions are appropriate for representing binary confidence. The beta is the conjugate prior for the evidence, meaning that sequential updates would reduce to parameter addition. Each incoming observation with confidence c adds c to α and (1-c) to β, where the resulting posterior is also beta distributed. This creates a system that is consistent across evidence ordering and avoids the need for global distribution, which would be antithetical to the project goals and computationally expensive.

2.4 Non-Axiomatic Reasoning System

The Non-Axiomatic Reasoning System (NARS) is a reasoning system built on top of Non-Axiomatic Logic (NAL). It's designed to operate under conditions with 'insufficient knowledge', representing beliefs as frequency/confidence pairs. The original NARS system has seen a number of revisions since its earliest launches [9], but its fundamentals have remained unchanged. In the intervening time, NARS has been used in various projects and has acted as a comparative point for other projects and ideas in the greater field. In 2024, it was used in a comparative analysis of how deduction, induction, and abduction formulas work between Probabilistic Logic Networks (PLN

- a different reasoning framework) and NARS. This type of cross-framework analysis, evaluating how different systems handle the same data, was an inspiration for the comparative analysis portion of this project [10].

2.5 Probabilistic and Uncertain Knowledge Representation

Beyond NARS and Bayesian Networks, other frameworks address uncertain knowledge representation that were taken into consideration during the development of this project. Markov Logic Networks combine first-order logic with probabilistic graphs in a manner not dissimilar from the BKG project [11]. Additionally, an implementation of a probabilistic KG system[12] that utilized data and rules mining to generate probabilities was found, which was one of the most concrete related examples that was found in relation to the BKG.

The frameworks and prior works mentioned above establish the theoretical foundation and context for the design and implementation of the Bayesian Knowledge Graph. The following section details the system's technical design.

3 Technical Specifications

3.1 Knowledge Graph

The BKG project utilizes the KG format as a basis for its data structure, with the intention of building a system that can handle uncertain data in a way that is consistent with the principles of KGs. Particularly, the topology of the graph is utilized for belief propagation, and the interconnectedness of data is utilized for belief updating. Node and edge metadata are used for the storage and updating of confidence and reliability values, which are core to the belief updating process.

While no dataset in this project makes use of a formal ontology, one could be used without detriment. It is not built into the belief aggregation or propagation, but has the potential to make tracking relationships easier and allow for deeper insights into data connections.

The KG software *Neo4j* was used for the second iteration of the project, which allows for high-quality visualization, granular searching, and easy embedding of metadata. The first iteration of the project utilized the *PyVis* package, which is built on top of NetworkX and Vis.js, which was highly customizable at the cost of less powerful search and a worse user experience.

3.2 Bayesian Weighting

Traditional Bayesian Networks function as probabilistic graphical models in which nodes represent predicates (random variables), and edges encode conditional dependencies between them. Each predicate is associated with a prior probability that reflects an initial belief before observing data, and these priors are systematically updated through evidence using conditional probability tables (CPTs). Inference is performed globally across the network via Bayes' Rule, allowing beliefs about all connected predicates to be updated in a consistent, multiplicative manner based on observed data. For this reason, Bayesian Networks are strictly directed acyclic graphs (DAGs) to avoid infinite propagation.

The principal idea was to imitate a Bayesian Network and apply it in a graphical format. By combining the way that Bayesian networks are able to make statistical determinations on forms of uncertain data with the interconnectedness of graphed knowledge, the proposed product would allow for intelligent decision-making. Ideally, relationships (edges) would re-weight themselves as new data was added based on the added knowledge into the system. While Bayesian Networks compute probabilities over Bayes' Rule across an entire graph (global probabilistic inference), this project utilizes local linear belief updating, does not use conditional probability tables, allows graphical cycles, and learns predicate priors as opposed to assuming them.

Under these definitions, the BKG is best described as a Bayesian inspired belief updating system. The system operates at the core level of Bayesian belief updating, via the usage of the beta distribution (α, β) as a conjugate Bayesian prior. New observations update beliefs using the distributions, which are Bayesian-consistent. Uncertainty (the core principle on display) is represented via the beta distribution.

3.3 Non-Axiomatic Reasoning System (NARS)

The Non-Axiomatic Reasoning System (NARS) is a reasoning framework designed to operate using data with a level of uncertainty and limited knowledge. It represents knowledge as weighted beliefs with frequency and confidence instead of fixed truth. NARS continuously processes incoming information, revises existing beliefs when contradictions or new information are processed, and allocates attention to relevant tasks.

In this project, NARS serves as both an inspiration and a conceptual comparison point for building more robust reasoning systems. The similarity between the two systems lies in how uncertain data is handled and how new knowledge is integrated with the current knowledge base. Both systems move away from binary truth values and instead utilize degrees of belief through confidence. However, while NARS uses non-axiomatic logic and revision rules, the Bayesian Knowledge Graph utilizes Bayesian principles, grounding belief updates in probabilistic inference and propagation. OpenNARS 3.1.2 was used for this project [13][14].

4 System Design & Iterations

4.1 NARS-like Bayesian Knowledge Graph

The first iteration of the project was an attempt to create a system that was inspired by NARS but utilized Bayesian principles for belief updating. It utilized an α/β distribution, where α represented the count of positive evidence, and β represented the count of negative evidence (in this case, the negative evidence against known positive evidence). Given a starting 'prior strength' (set to 2 in this iteration), $\alpha = \beta = 2$ created an initial belief of $\frac{2}{2+2} = 0.5 = 50\%$ confidence. A prior strength of 2 instead of 1 required a higher amount of evidence to shift beliefs. When a new relationship was observed, given confidence c, a weighted confidence was calculated and incremented as $\alpha = \alpha + c$, and β was incremented to $\beta = \beta + (1 - c)$.

A concept of 'node reliability' was introduced as part of the weighted confidence calculation, each of which also had α, β values to determine how trustworthy a node was based on the quality of relationships it was attached to. This created a feedback loop of reliability, wherein consistent addition of evidence would increase the reliability of nodes and the confidence of edges.

This is referred to as 'NARS-like' due to the direct way in which data reliability and confidence are affected. In NARS, confidence & frequency are determined by repeated instances of an object and its confidence being added to the system. While this is very useful for NARS as a reasoning system, it is less helpful in this system. It would require consistent affirmation of the same knowledge to affect the confidence of uncertain data, which is not guaranteed or likely to happen, depending on the circumstances of the evidence. The goal was to allow all incoming data to affect other data, both positively and negatively, for belief updating based on surrounding data. This goal may have been achievable using this method in other formats, but as a knowledge graph type structure, there are rarely duplicate triples at the frequency needed to influence the system.

4.2 Bayesian Knowledge Graph V1

The second iteration operated in a similar structural manner to the first, but with numerous changes to the belief system to better fit the knowledge graph format, and to better utilize the graph structure for belief updating. This system is inspired by NARS-style evidence accumulation, but does not implement the semantics, calculus, or inference rules of that system. Instead of working with global updating (Bayesian Networks) or reinforced belief updating (NARS), the BKG system updates confidence locally and in-line based on graphical connections as a type of evidence aggregation.

It uses beta distributions for edge-level beliefs, node-level reliability, and predicate-level priors. Nodes and predicates are initialized with $\alpha = \beta = 0.5$ to represent an uninformative prior, and edges are initialized from a predicate prior (confidence) when seen. Node reliability is calculated as such when a triple (Subject, Predicate, Object, Confidence) is ingested:

- Subject Reliability = α Subject / (α Subject + β Subject)
- Object Reliability = α Object / (α Object + β Object)
- Node Weight = (Subject Reliability + Object Reliability) / 2

Edges are initialized if they do not exist, and inherit the *learned statistic* of the predicate type. For example, if the edge 'worksAt' is a newly initialized type, it starts with (0.5,0.5) at a confidence of 50%. In the same example, the 3rd 'worksAt' edge might start with (3.2, 1.8) at a confidence of 65%.

From there, evidence strength is calculated by multiplying the calculated node weight by the evidence scale (in the system currently, set to 3). This takes into account how reliable the nodes are with an amplification factor, and determines how strong a piece of evidence should be taken at. Given an edge with confidence 'c', and evidence strength 's' it's α is calculated as $\alpha = \alpha + c * s$ and $\beta = \beta + (1 - c) * s$

$$(\alpha_e, \beta_e) \longleftarrow (\alpha_e + c * s, \beta_e + (1 - c) * s)$$

Equation for Bayesian weighting (belief updating) rule for edges in the BKG system.

Once edge belief has been updated, node reliability can be updated as well. This is done using the raw confidence, not the version scaled by evidence strength. α and β are recalculated by adding confidence and 1-confidence, and reliability is recalculated by $\frac{\alpha}{\alpha+\beta}$. Low confidence edges reduce node reliability, and high confidence edges increase node reliability. Lastly, the predicate prior is updated. This is done in the same manner as the node reliability is, using raw confidence.

This design creates a positive feedback loop in which high-confidence edges increase node reliability, and high-reliability nodes amplify future evidence. Predicate-specific learning ensures that each relationship type learns and stabilizes independently, balancing a large re-weighting against individual reliability metrics. Utilizing an evidence scale/strength allows users to determine a workable scale of how much evidence should be valued with each instance based on the circumstances of usage and the amount of evidence.

4.3 Bayesian Knowledge Graph V2

The current iteration of the project is a refinement of the previous iteration, with a focus on improving the belief updating and propagation methods. An additional focus of updating the interface used for the graph was added, as the previous interface that was used could not handle the potential scale of the project, and was not adept at searching or visualizing large graphs.

Initially, in a more customizable graphing software, the decision was made to trade off customization for better searchability, embedding of metadata, and the ability to have multiple runs easily built on top of each other. For these reasons, the interface now runs through Neo4j, though it could feasibly run through any similar graph software. Neo4j allows for granular searching through the Cypher language, metadata that can be well embedded into nodes and edges, and easily visualized and searched for independently of each other, and better visibility of graph interconnectedness at scale.

Primarily, handling belief propagation outside of a single update needed to be addressed. In NARS, this is handled through repeated evidence accumulation, which then gives weight to inference and assumption. This was attempted already in development and was both impractical and incorrect. A first attempt for propagation in the V1 iteration was implemented, but left room for error, either through not properly propagating in some areas or creating the potential for infinite propagation in others.

Solving this in a way that was consistent and explainable required trial and error, but was solidified into the following: There is a maximum number of nodes a spawned propagation can affect before it is stopped. Within that chain, propagation decay is implemented, so that the effects are lessened the further away from the initially affected node. Propagation effects are determined by an evidence scale, where evidence strength (determined by the node weight and a maximum scale factor) is multiplied by the decay factor (γ), and then adjusted as α and β accordingly. Additionally, the evidence_scale hyperparameter has been removed in favor of the evidence_strength variable. The two are functionally the same in usage, but are now calculated dynamically based on max_scale and node weight instead of being an independent hyperparameter.

$$\alpha' = \alpha + \gamma^d \cdot \frac{\text{max\_scale} \cdot w}{1 + w} \cdot c$$
$$\beta' = \beta + \gamma^d \cdot \frac{\text{max\_scale} \cdot w}{1 + w} \cdot (1 - c)$$

Equations for belief propagation rules for nodes BKG system

The four hyperparameters the system now utilizes are as follows:

- prior_strength (default = 0.5): the strength of the initial prior for nodes and predicates, which determines how much evidence is required to shift beliefs from the initial 50% confidence.
- max_scale (default = 6.0): the maximum strength of evidence that can be applied to a relationship, which is then scaled by node reliability to determine actual evidence strength.
- γ (decay factor) (default = 0.7): the rate at which evidence strength decays during propagation, which is applied exponentially based on the distance from the initial node.

- max_depth (default = 5): the maximum number of nodes that can be affected by a single propagation event, which serves as a hard limit to prevent infinite propagation and control the scope of belief updates.

4.3.1 Evidence Propagation Example

This example demonstrates how belief propagation works in the BKG system. Consider a pre-existing knowledge graph with nodes A, B, and C. The edge $B \rightarrow C$ exists in the network structure but has never been directly observed; it therefore holds only the predicate prior with confidence 0.5. When a new observation arrives for edge $A \rightarrow B$ with confidence 0.8, the system first updates $A \rightarrow B$ and then propagates this belief to neighboring edges. The propagation mechanism causes $B \rightarrow C$ to be updated based on the increased reliability of node B, demonstrating how evidence can influence related beliefs without direct observation.

Nodes: A, B, C

Edges: $A \rightarrow B$ with predicate p_1 $B \rightarrow C$ with predicate p_2

Initial Confidence: $A \rightarrow B : 0.8$ $B \rightarrow C : 0.6$

Parameters: prior_strength = 0.5 max_scale = 6.0 γ (decay factor)= 0.7 max_depth = 1

Goal: Propagate belief from $A \rightarrow B$ to $B \rightarrow C$.

Step 1: Observation for $A \rightarrow B$

Node Reliability: $R_A = \frac{0.5}{0.5+0.5} = 0.5$ $R_B = \frac{0.5}{0.5+0.5} = 0.5$

Node Weight: $w = \frac{0.5+0.5}{2} = 0.5$

Scale: scale $= \frac{6.0 \cdot 0.5}{1+0.5} = 2.0$

Edge Update ($A \rightarrow B$)**:**

- decay $= 0.7^0 = 1.0$
- evidence $= 2.0 \cdot 1.0 = 2.0$
- $\alpha = 0.5 + 0.8 \cdot 2.0 = 2.1$
- $\beta = 0.5 + (1 - 0.8) \cdot 2.0 = 0.9$
- confidence $= \frac{2.1}{2.1+0.9} = 0.7$

Node Updates: $A : (0.5, 0.5) \rightarrow (2.1, 0.9)$ $B : (0.5, 0.5) \rightarrow (2.1, 0.9)$

Note: The edge $B \rightarrow C$ already exists in the network with only the predicate prior (no direct observations). It is updated through belief propagation triggered by the new evidence on $A \rightarrow B$.

Inferred Confidence: The confidence from $A \rightarrow B$ (0.7) is used to propagate the belief to $B \rightarrow C$.

Node Reliability: $R_B = \frac{2.1}{2.1+0.9} = 0.7$ $R_C = \frac{0.5}{0.5+0.5} = 0.5$

Node Weight: $w = \frac{0.7+0.5}{2} = 0.6$

Scale: scale $= \frac{6.0 \cdot 0.6}{1+0.6} = 2.25$

Edge Update ($B \to C$):

- decay $= 0.7^1 = 0.7$ (propagation depth = 1)
- evidence $= 2.25 \cdot 0.7 = 1.575$
- $\alpha = 0.5 + 0.7 \cdot 1.575 = 1.6025$
- $\beta = 0.5 + (1 - 0.7) \cdot 1.575 = 0.9725$
- confidence $= \frac{1.6025}{1.6025+0.9725} \approx 0.622$

Node Updates: $B : (2.1, 0.9) \to (3.2025, 1.3725) \qquad C : (0.5, 0.5) \to (1.6025, 0.9725)$

Final Results

Node Reliability: $A : \frac{2.1}{2.1+0.9} = 0.7 \qquad B : \frac{3.2025}{3.2025+1.3725} = 0.7 \qquad C : \frac{1.6025}{1.6025+0.9725} \approx 0.622$

Edge Beliefs:

- $A \to B : (2.1, 0.9)$ confidence $= 0.7$
- $B \to C : (1.6025, 0.9725)$ confidence ≈ 0.622

5 Experiments & Results

5.1 Data

All data used for comparison between the BKG and NARS is branched off a central dataset of medical data. The subjects/objects of this dataset include a number of common ailments, symptoms, and treatments. The predicates note these items as symptoms, causes, or treatments. Confidence of each relationship was randomly generated to be between 0.3 and 0.95. The dataset's medical diagnoses may not be fully accurate, however, semantic correctness was not the focus, and the dataset was used to demonstrate structural uncertainty behavior in a domain that may be realistic. A secondary dataset titled "CN15k" [15] was publicly sourced for demonstration of the potential scope of the Bayesian Knowledge Graph, and is described as a 'common sense knowledge graph'.

5.2 BKG on Large Data

The objective of this section is to evaluate the scalability and stability of the BKG framework when applied to a large dataset. The primary hypothesis is that the BKG's belief updating mechanism would maintain stable confidence values at scale without runaway confidence increase or decrease, and that the system would be able to process large amounts of data in reasonable timeframes.

The CN15k dataset was chosen for this experiment due to its size (16,000 triples) and its characterization as a 'common sense' knowledge graph, which is a relevant use case for uncertain data. The structure of the dataset was that all values were numerically encoded with separate key-value files to associate them with string objects. This structure provided a practical test for the BKG's data processing capabilities, as it required an additional step of re-associating the numerical values to their corresponding string representations before loading into the graph, and required approximately 60 minutes to fully process on a personal computer.

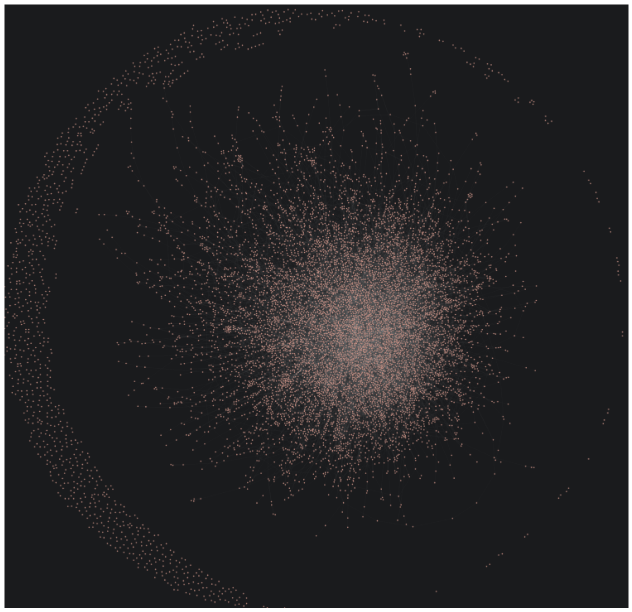

Figure 1: CN15k Full Graph Network

For each predicate, the following metrics were computed: average original confidence (the confidence assigned in the dataset), average final confidence (the confidence after processing through the BKG), and average confidence change (the difference between original and final confidence). The maximum shift of confidence across all predicates was also calculated to assess the stability of the system. The maximum shift of confidence in relations (the entire dataset only had 35 predicate labels) ranged from only $+1.6\%$ to -13.6%. This mirrors behavior that was seen in smaller datasets as well, that the system appears to have an easier time reducing decreasing confidence than it does increasing it. Additionally, data does seem to stabilize well at scale. Even with propagation capped, a potential concern was still a runaway increase or decrease to the extremes, which was seemingly avoided in this example.

The results from this experiment support the hypothesis that the BKG framework can maintain stable confidence values at scale. Confidence values seemed to decrease more than they increased, suggesting that the system was conservative in belief amplification, which is preferable to a tendency for runaway confidence increase. Additionally, the absence of extreme values in either direction indicates that propagation limits and scaling factors are an effective method to prevent runaway feedback loops.

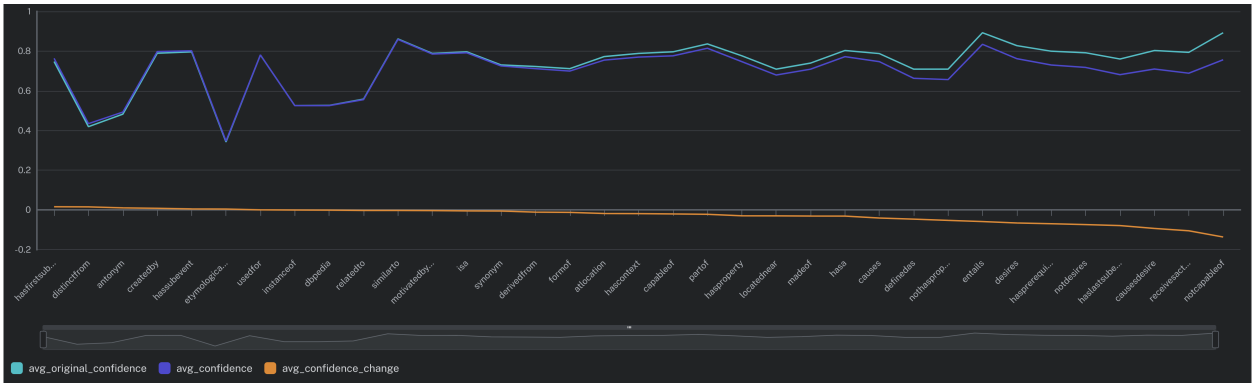

Figure 2: Average Confidence Changes CN15k

5.3 BKG vs NARS

This section summarizes experimental findings from a comparison of the Bayesian Knowledge Graph framework against NARS (OpenNARS v3.1.2). Findings are drawn from BKG output in both graphical and tabular formats, and from NARS inference logs. All experiments in this section used a version of the Medical dataset referenced in the data section.

Parameter Updates: Within OpenNARS, multiple parameters were updated to ensure better responsiveness from the system. Due to the high volume of data immediately loaded, the attention mechanism within NARS struggled to answer direct questions posed by the user. Default question priority was adjusted from 0.85 to 0.99 and default question durability was adjusted from 0.9 to 0.99. Concept bag size was adjusted from 1000 to 3000, task buffer size was adjusted from 10 to 100, and global buffer size was adjusted from 20 to 100. Concept forgetting cycle was adjusted from 10 to 30, new task forgetting cycle was adjusted from 1 to 5, max matched term link was adjusted from 10 to 25, and max reasoned term link was adjusted from 3 to 7. All of these changes were done incrementally to optimize NARS's attention towards user queries, and to ensure higher term and task retention.

5.3.1 NARS-Specific Issues

Initial attempts for experiments in NARS led to a number of unexpected results. These were partially caused by experimental design, including data simplicity and high data volume being loaded all at once, but are also representative of issues posed by NARS's architecture. In order to mimic belief aggregation in NARS, data was deliberately repeated in .nal files so that confidence and frequency could properly aggregate (which the BKG did not have to replicate). Everything in this section was found using the full MedData.nal and MedData_bridged.nal files.

Confidence Calibration: In order to match the data that the BKG intakes data, part of the chosen methodology for loading data into NARS was to have repeated instances of the same observation as part of the loaded file. This was done to ensure that relationships were not overly 'brittle' and would not immediately shift drastically with new data. However, an unintended side effect became that all retrieved beliefs returned maximum frequency due to over-reinforcement, regardless of the source data's original confidence.

```
OUT: <Fever --> (/,is_symptom_of,_,Influenza)>. %1.00;0.90% {2 : 1}
OUT: <Fatigue --> (/,is_symptom_of,_,Influenza)>. %1.00;0.90% {3 : 3}
OUT: <Headache --> (/,is_symptom_of,_,Migraine)>. %1.00;0.90% {8 : 11}
```

Figure 3: Source confidence values were 0.78, 0.83, and 0.89 respectively. All collapsed to 1.00.

Relation Collapse: An additional issue that became prevalent quickly was the semantic collapse of relations. Given the small nature of the dataset compared to the volume of input, NARS began to treat predicate terms as independent concept nodes instead of relationships, with equivalent weight to independent disease or symptom terms. This caused predicates to appear as subject/objects in inheritance statements and to form equivalence relations in specific instances. This has been termed as *relation collapse* as relationship labels are collapsing into objects themselves.

```
OUT: <is_symptom_of <-> (*,Wheezing,Common_Cold)>. %0.27;0.17%
OUT: <(*,(/,is_symptom_of,_,Migraine),Food_Poisoning) --> is_symptom_of>. %0.50;0.00%
OUT: <is_symptom_of --> (*,(/,is_symptom_of,_,Migraine),Food_Poisoning)>. %0.50;0.00%
```

Figure 4: Predicate is_symptom_of collapses into an independent concept, which is treated as a term

Sudden ('Thin Air') Inferences: Over time, NARS began to generate equivalence relations between diseases that had no direct connection in the source data. Investigation suggested that these inferences were driven by shared predicates over shared symptoms. While NARS is designed to make these types of non-stated inferences by analogy and inference rules, this behavior was done for incorrect reasons in this circumstance, and could lead to incorrect decision-making.

```
OUT: <Anemia <-> Migraine>. %1.00;0.45% {4905 : 65;87}
Statement 65: <(*,Migraine,Triptans) --> treated_with>
Statement 87: <(*,Anemia,Iron_Supplements) --> treated_with>
```

Figure 5: Sudden equivalence between unrelated medical subjects.

'Bridging' Solution: All three of the above problems were at least partially related to NARS, eventually using relationship labels to generate inferences that were not taxonomically correct. While this is a more extreme example, given the lack of variety in used predicate labels, it is something that is always possible to occur. To this end, a simple set of 'bridging rules' was added explicitly to the start of all .nal files that were meant to define relationship taxonomy.

Each of the three relationship labels got a forward and backward rule, where each rule was designed to force NARS to understand the relationship in a certain way during its statement decomposition. The forward rule reads that *"If any x and y have the product relation z, then x inherits the relational image of z with respect to y."*. Conversely, the backward rule reads that *"If x inherits the relational image of z with respect to y, then x and y have the product relation z."*. In both cases, it means that any time NARS attempts to derive a statement including a relationship with that label as an object, it re-establishes the difference between objects and relationships. An example is as follows:

```
<<(*,?x,?y) --> is_symptom_of> ==> <?x --> (/, is_symptom_of,_,?y)>>.
<<(?x --> (/, is_symptom_of,_,?y))> ==> <(*,?x,?y) --> is_symptom_of>>.
```

Figure 6: Bridging Rule example for predicate label "is_sympton_of"

```
Line 1: <(*,Fever,Influenza) --> is_symptom_of>.
Line 2: <Fever --> (/,is_symptom_of,_,Influenza)>.
```

Figure 7: Forward Rule: What NARS sees (line 1) to what it derives via the rule (line 2) Backward Rule: What NARS derives (line 2) to what the rule re-establishes (line 1)

The addition of bridging rules nearly completely solved the first two listed issues (Confidence Calibration & Relation Collapse). These rules helped to preserve graded values during loading, avoiding the collapse to confidence = 1.0 regardless of listed and desired confidence values. They were most successful during relation collapse, as the explicit typing of rules and expected decomposition almost entirely avoided NARS attempting to turn relationship labels into independent subject/object nodes. While it still happens occasionally, confidence is significantly reduced, and the frequency of occurrence is substantially reduced as well. Sudden equivalence was still something that occurred even after adding in bridging rules, the confidence levels of these incorrect inferences were notably reduced.

```
OUT: <Poor_Sleep --> (/,is_symptom_of,_,Chronic_Fatigue_Syndrome)>. %0.56;0.90%
OUT: <Itchy_Eyes --> (/,is_symptom_of,_,Seasonal_Allergies)>. %0.32;0.90%
OUT: <Loss_of_Smell --> (/,is_symptom_of,_,COVID_19)>. %0.93;0.90%
```

Figure 8: Confidence Calibration after implementation of Bridging Rules

```
OUT: <Lyme_Disease <-> Rheumatoid_Arthritis>. %0.63;0.44%
OUT: <Influenza <-> Pneumonia>. %0.63;0.43%
OUT: <Mononucleosis <-> Strep_Throat>. %0.55;0.43%
OUT: <Common_Cold <-> Seasonal_Allergies>. %0.60;0.43%
```

Figure 9: Relation Collapse example post Bridging Rules. Notably low frequency and confidence compared to previous example

```
OUT: <Anemia <-> Migraine>. %1.00;0.45% {4905 : 65;87}
Statement 65: <(*,Migraine,Triptans) --> treated_with>
Statement 87: <(*,Anemia,Iron_Supplements) --> treated_with>
```

Figure 10: Cross-Disease Sudden Equivalence post Bridging Rules.

In all three instances, these are issues that the Bayesian Knowledge Graph does not suffer from. The BKG framework stores confidence as alpha/beta ratios and applies scaling through node reliability and predicate priors. Confidence is derived from input data, and does not require any type of correcting steps to preserve that graded or derived confidence (confidence calibration). In terms of separation of nodes and relationships, predicates and their typed edge labels are stored inside the property graph. These predicates are separated entirely from nodes and do not participate in node-level inference, aggregation, or propagation. Predicate priors are tracked and calculated independently as accumulators and have no representation within graph nodes.

By design, the type of relation collapse that happened inside NARS cannot happen within the BKG framework. In the same vein, the causes of cross-node inference cannot happen without topological paths. In the example first listed, where 'Migraine' and 'Anemia' created a correlation, it could not happen in the BKG, as there was no shared edge path. It is a conservative design where propagation exclusively moves along asserted edges. While this can be a limiting factor as NARS will make connections that the BKG never will, including many valid ones that follow correctly with its inference rules, a more conservative approach in the BKG potentially ensures higher data purity.

5.3.2 Contradiction Handling

This section focuses on the system's ability to handle contradictions in data, which is a necessary aspect of reasoning with uncertain data. The hypothesis is that both systems will reduce confidence in affected relationships with the introduction of contradictory data. Due to the incompatible architectures of the two systems, it was unclear how the degree of confidence reduction would compare and whether the same relationships would be affected in both systems.

A dataset consisting of 25 triples was constructed using the previously described medical domain. Two versions of this dataset were used, one with the initial 25 triples and one with an additional 6 contradictory triples that directly contradicted previously established evidence. These contradictory statements were repeats of given data with deliberately low confidence (0.1 - 0.3) to suggest that the results were false or untrustworthy. Both systems were tested on the original dataset and then on the dataset with contradictions, and confidence shifts were observed. The primary metric for evaluation was the confidence shift in the 6 relationships that were contradicted.

Relationship	**BKG Base**	**BKG Post**	**BKG Δ**	**NARS Base**	**NARS Post**	**NARS Δ**
Anemia *caused_by* Iron Def.	0.818	0.644	-0.174	0.87	0.59	-0.280
Influenza *treated_with* Oseltamivir	0.800	0.576	-0.224	0.93	0.62	-0.310
Fever *is_symptom_of* Influenza	0.687	0.463	-0.224	0.78	0.54	-0.240
COVID-19 *caused_by* Viral Inf.	0.853	0.694	-0.159	0.95	0.63	-0.320
Migraine *treated_with* Triptans	0.865	0.699	-0.166	0.92	0.66	-0.260
Pneumonia *treated_with* Antibiotics	0.844	0.638	-0.206	0.91	0.66	-0.250

Table 1: Confidence shift under partial contradiction: BKG vs. NARS

Under the effects of contradiction, both systems did revise confidence levels in similar measures. The post-contradiction confidences ranged from 0.46 - 0.7 for the BKG, and 0.54 - 0.66 for NARS. While the ranges overlap, order is not preserved, with the confidence ordering of the affected relationships differing between the two systems. Within that range, NARS did revise more aggressively than BKG. NARS deltas have a range from -0.24 to -0.32, while BKG deltas range from -0.16 to -0.22. Both of these are accepted and expected outcomes. 19% of the statements in the loaded data ($\frac{6}{31}$) were contradictory statements designed to bring down confidence, so confidence decreases generally within the range of 20%, which is considered to be a reasonable degradation. Much lower and either system may be considered too change resistant, much higher and it may have been seen as too susceptible to change.

NARS certainty increased in the related objects after the contradiction, from 0.9 to 0.95. This is an expected behavior as it reflects that more evidence has been processed. However, it can be considered a reminder that NARS is susceptible to faulty data in its own way, in that, regardless of the validity, it adds certainty to the associated relationship. This could lead to belief hardening around an incorrect premise. The BKG handled this differently, where evidence increases $\alpha + \beta$ (denominator of edge belief), which will tighten the beta distribution. Depending on evidence consistency with current confidence and node reliability, this could serve to increase or decrease uncertainty. In this case, edge uncertainty increased with the contradictory data compared to without it.

The results from both systems support the hypothesis, as both systems revised confidence values in response to contradictory data. The higher confidence reduction from NARS suggests that its revision rules are more sensitive to conflicting evidence, while the BKG's more moderate updates show a conservative approach in line with the reliance on accumulated evidence and node reliability.

5.3.3 Data Order Sensitivity

This experiment evaluates if the order of data ingestion affects final confidence values in either system. The initial hypothesis is that the BKG should be affected by data ordering to a greater degree than NARS due to the fact that confidence is derived from the α/β ratio, and α and β are updated with each new piece of evidence. While NARS does undergo belief revision, which affects confidence, the belief revision rules are expected to be more robust to data ordering.

The same 25-triple medical dataset used in the previous experiment was reorganized into three orderings: baseline (original), reversed, and randomized. Each ordering was processed through both systems, and final confidence values for the six representative edges were recorded and compared across orderings. The representative edges were chosen as they were the ones that were directly affected by the contradictory statements in the previous experiment, and thus would be expected to show the most significant changes in confidence if ordering had an effect.

In NARS, the order of data did not affect confidence values, instead, all belief values were identical across all orderings. There are two posited reasons for this. Firstly, the repeated data loading method means that each triple enters the system with enough individually reinforced evidence that the belief is ’grounded’ before the next triple is processed. This was deliberate, as otherwise NARS would have had too much drift in all categories, but it creates reinforcement here. Secondly, NARS’s revision rules are both associating and commutative. With the same set of evidence, the beliefs should converge, which is exactly what happened.

The BKG framework did show some sensitivity to data ordering, though the effects were moderate and within reasonable limits. Final confidence values ranged from 0.013 to 0.065 across the three runs. This behavior is explainable by the node reliability feedback loop. Nodes that appear earlier accumulate reliability faster, which then amplifies the strength later applied to shared edges and predicates. This is especially on display in the ’random’ experiment, where grouped items were loaded in separately, creating both notably higher and lower confidence values than the other two runs.

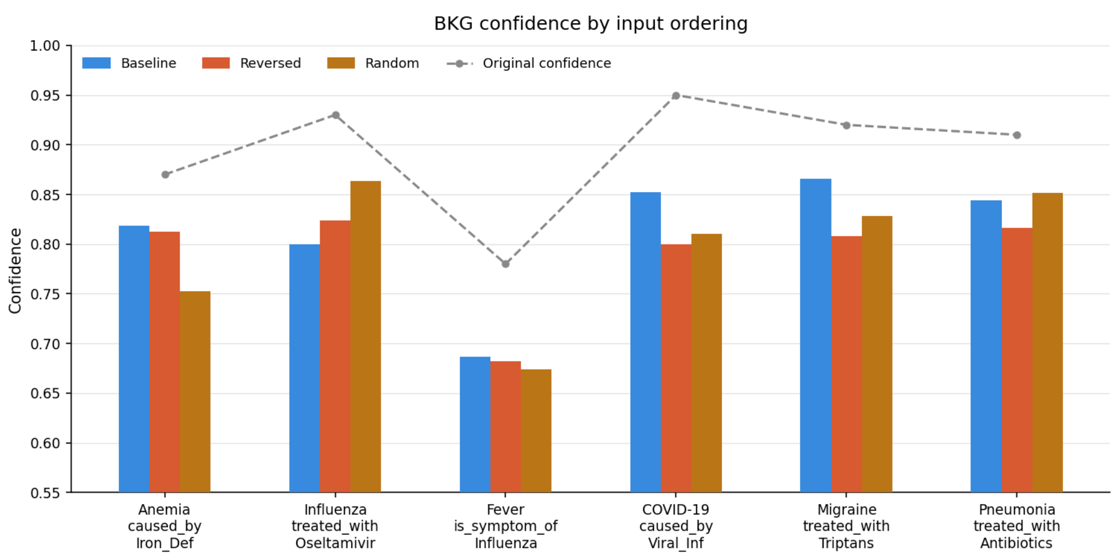

Figure 11: Confidence values for representative edges across orderings

While this could be a potential issue, the experiment also suggests that this effect is bounded. No edge shifted by more than 0.065, and no confidence values changed sign, or moved outside a reasonable range. It still presents a limitation in the system that could be deliberately exploited, or could lead to faulty or mismatched conclusions if data is not loaded in a consistent manner. Overall, the results from this experiment support the hypothesis that the BKG is more sensitive to data ordering than NARS, but also suggest that the effects of this sensitivity are moderate and bounded.

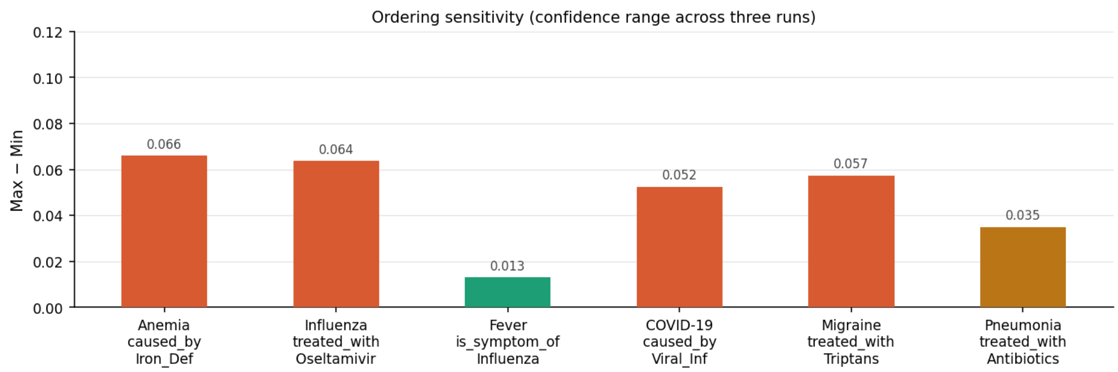

Figure 12: Ordering sensitivity (confidence range across three runs)

6 Discussion

The experiments conducted in this paper are designed as a test of a single architectural structure: that a knowledge graph system utilizing a topology-based belief propagation is an effective method to handle uncertain data. The BKG's design choices deliberately trade complex inferential power for a more conservative and potentially reliable approach for handling uncertain data, and the results of those experiments reflect that.

The large-scale CN15k experiment demonstrated that the design properties of the BKG can hold up at a scale more representative of real-world usage. With 16,000 triples and 35 predicate labels, confidence adjustments remained bounded (+1.6% to -13.6%) without runaway feedback loops in either direction. The system's tendency to reduce confidence more often than to increase it reflects the evidence-scaling mechanism, where node weights are bounded and then decayed with propagation depth. This conservative approach is preferable in uncertain domains where overconfidence in low-quality data can lead to more potential issues than underconfidence in high-quality data.

The NARS-specific issues documented, including confidence calibration, relation collapse, and sudden inference, were findings that highlighted both potential issues with the structural architecture of NARS and how the explicit separation of concepts and relationships in the BKG can avoid these issues. NARS's high level of inference power is a strength of the system, but it can also lead to issues with data purity and reliability if not carefully managed. The BKG's architectural choices deliberately avoid these issues by separating predicates from objects, with propagation moving exclusively through asserted edges. Where these trade-offs matter is domain-specific, as the expected rate of inputted data and importance of data purity vs inferential power will differ across use-cases.

The contradiction handling experiment supported a point about how the two systems represent knowledge. Both systems revised confidence according to the volume of contradictory data. NARS revised more aggressively (20-30% confidence reduction) while the BKG did so more conservatively (15-22% confidence reduction). Potentially more significant was the certainty differences under contradiction. NARS's certainty increased from 0.9 to 0.95, as a byproduct of additional evidence processing. This is a consistent behavior for the system, but it means that NARS can 'harden' around beliefs actively being contradicted. The BKG increased edge uncertainty under the same contradictions, as it widened the beta distribution with additional evidence. In use cases where separating high-confidence from high-certainty matters - domains where "X is confident but uncertain" is meaningfully separate from "X is confident and certain" - the BKG's approach may be more desirable, as it allows for confidence to be reduced without hardening certainty around a potentially incorrect belief. In domains where contradictory data is common, consequential, or both, this could be a significant advantage.

The ordering sensitivity experiment revealed an important limitation of the current design. The BKG showed bounded sensitivity to data order, with confidence values shifting by up to 6.5% across the three orderings, caused by the node reliability feedback loop (where nodes accumulating reliability earlier amplified future evidence). In contrast, NARS showed to be order-insensitive, producing identical results across all orderings due to its revision rules and the evidence preloading used in this study. The significance of this sensitivity depends on the use-case. In settings where data is expected to arrive consistently and in order of occurrence, the ordering effect is unlikely to create a major issue. However, in settings where data ordering may arrive in unpredictable ways or without a temporal order, it shows a vulnerability. Future work on this limitation may explore mitigation methods such as normalization for node reliability accumulation.

Taken as a whole, these experiments position the BKG as a viable framework for structured, uncertain data where the relationships between entities are of equal importance to the entities themselves. The comparison to NARS does not suggest system superiority due to major architectural and use-case differences, but instead that they occupy different spaces in the field of uncertain data handling. The BKG's contribution to the space is showing that local topological Bayesian belief updating can produce stable, interpretable, and proportionate confidence adjustments to uncertain data without the architectural overhead of an entire reasoning engine. While the BKG does not have the same inferential power as NARS, it offers a more conservative and potentially more reliable approach to handling uncertain data in a knowledge graph setting, especially in domains where data purity and relationship structure are of high importance.

7 Future Work

Several options for future work present themselves from this project. At a high level, these include propagation scaling, a formal ontology layer, a broader comparative study, and a more in-depth treatment of evidence.

The current implementation has the hyperparameter for propagation depth capped at 5. This was used even in the large dataset experiment, as it is a computationally expensive process to run the Bayesian Knowledge Graph over such a large dataset. Parallelizing propagation across different subgraphs could be a potential solution to this, as well as implementing more efficient methods for belief updating. Regardless of the methodology, it would be valuable to explore the effects of deeper propagation on confidence values and to evaluate whether the current capping is sufficient for most use cases.

A formal ontology layer could be added to the BKG to provide a more structured framework for types of nodes and edges. The current system treats all predicates as accumulators without semantic constraints, and utilizing an ontology would allow for different predicate types to have different prior structures. This would increase domain awareness, deepen semantic understanding, and allow for a more nuanced usage of the knowledge graph structure that the framework provides.

The comparison against NARS in this paper was limited in scope. A broader comparative study including other frameworks for uncertain knowledge representation (Probabilistic Soft Logic, Markov Logic Networks, other uncertain knowledge graph frameworks) would help to better place the BKG in the field of other existing systems, and refine what trade-offs it has compared to other methods. Other comparative experiments would also be of potential use, including one that was planned but not executed for this paper regarding the effects of sparse vs dense evidence chains.

Finally, a more in-depth treatment of evidence could be explored. The current implementation uses a simple scaling mechanism based on node reliability and predicate priors, but there may be more sophisticated methods for weighting evidence that could improve the system's performance. For example, incorporating temporal information about when evidence was observed, or using machine learning techniques to learn optimal weighting strategies from data, could enhance the BKG's ability to handle uncertain data effectively.

References

[1] A. Hogan et al., "Knowledge Graphs," *ACM Computing Surveys*, vol. 54, no. 4, pp. 1–37, May 2022, arXiv:2003.02320 [cs]. DOI: 10.1145/3447772 [Online]. Available: http://arxiv.org/abs/2003.02320

[2] L. Jarnac, Y. Chabot, and M. Couceiro, *Uncertainty Management in the Construction of Knowledge Graphs: A Survey*, en, arXiv:2405.16929 [cs], Jul. 2024. DOI: 10.48550/arXiv.2405.16929 [Online]. Available: http://arxiv.org/abs/2405.16929

[3] X. Chen, M. Chen, W. Shi, Y. Sun, and C. Zaniolo, *Embedding Uncertain Knowledge Graphs*, en, arXiv:1811.10667 [cs], Feb. 2019. DOI: 10.48550/arXiv.1811.10667 [Online]. Available: http://arxiv.org/abs/1811.10667

[4] Y. Zhu et al., *Certainty in Uncertainty: Reasoning over Uncertain Knowledge Graphs with Statistical Guarantees*, en, arXiv:2510.24754 [stat], Oct. 2025. DOI: 10.48550/arXiv.2510.24754 [Online]. Available: http://arxiv.org/abs/2510.24754

[5] *Probabilistic Reasoning in Intelligent Systems*, en, ISBN: 9780080514895. [Online]. Available: https://www.sciencedirect.com/book/monograph/9780080514895/probabilistic-reasoning-in-intelligent-systems

[6] E. P. Xing, R. Das, Z. Liu, and D. Gupta, "CMU Belief Propagation," en, [Online]. Available: https://www.cs.cmu.edu/~epxing/Class/10708-14/scribe_notes/scribe_note_lecture13.pdf

[7] D. Fink, "A Compendium of Conjugate Priors," en, [Online]. Available: https://www.researchgate.net/publication/238622435_A_Compendium_of_Conjugate_Priors

[8] *Introduction to Probability and Statistics (MIT)*, en. [Online]. Available: https://ocw.mit.edu/courses/18-05-introduction-to-probability-and-statistics-spring-2022/pages/classes-reading-and-in-class-materials/

[9] P. Wang, "Non-Axiomatic Reasoning System (Version 4.1)," en, [Online]. Available: https://cdn.aaai.org/AAAI/2000/AAAI00-204.pdf

[10] B. Goertzel, *PLN and NARS Often Yield Similar strength $\times$ confidence Given Highly Uncertain Term Probabilities*, en, arXiv:2412.19524 [cs], Dec. 2024. DOI: 10.48550/arXiv.2412.19524 [Online]. Available: http://arxiv.org/abs/2412.19524

[11] M. Richardson and P. Domingos, "Markov logic networks," en, *Machine Learning*, vol. 62, no. 1-2, pp. 107–136, Feb. 2006. DOI: 10.1007/s10994-006-5833-1 [Online]. Available: http://link.springer.com/10.1007/s10994-006-5833-1

[12] allthingssecurity, *Allthingssecurity/probabilistic_kg*, original-date: 2025-08-31T07:00:00Z, Aug. 2025. [Online]. Available: https://github.com/allthingssecurity/probabilistic_kg

[13] *Opennars/opennars_v3.1.2*, original-date: 2021-10-12T20:51:00Z, May 2025. [Online]. Available: https://github.com/opennars/opennars_v3.1.2

[14] T. A. Team, *The conceptual design of opennars 3.1.2*, Feb. 2022. [Online]. Available: https://cis.temple.edu/tagit/publications/PAGI-TR-13.pdf

[15] *PSL CN15k*, en. [Online]. Available: https://www.kaggle.com/datasets/hughiephan/psl-cn15k