Task 3: Rule Mining in MetaFam

Knowledge Graph Rule Discovery, Validation, and Analysis

"Happiness can be found even in the darkest of times,
if only one remembers to turn on the light." — Albus Dumbledore

Contents

1 Introduction and Methodology

1.1 Objective

Knowledge graphs encode logical rules that can be discovered automatically. In a family graph, many relationships can be inferred from others through compositional reasoning. The goal of this task was to discover, validate, and analyze logical rules in the MetaFam knowledge graph.

I explored four categories of rules:

1. **Inverse Rules:** $R_1(X, Y) \rightarrow R_2(Y, X)$
2. **Compositional Rules (2-hop):** $R_1(X, Y) \wedge R_2(Y, Z) \rightarrow R_3(X, Z)$
3. **Symmetric Rules:** $R(X, Y) \rightarrow R(Y, X)$
4. **Complex Rules (3-hop):** $R_1(X, Y) \wedge R_2(Y, Z) \wedge R_3(Z, W) \rightarrow R_4(X, W)$

★ Beyond Requirements

Beyond the required minimum of 5 rules, I conducted several additional analyses not requested by the task:

- **Gender-aware rule analysis** — decomposing rules by gender to explain apparent symmetry failures
- **Constraint mining** — verifying 6 logical impossibilities to assess data quality
- **KG incompleteness quantification** — measuring the exact number and percentage of missing inverse edges, directly connecting to the Task 4 test set
- **Semantic direction analysis** — investigating and correcting uncle/aunt rule failures caused by edge direction misinterpretation
- **Rule confidence heatmap** — visualizing compositional patterns across all relation pairs

1.2 Data Structures and Metrics

For efficient rule mining, I built lookup structures enabling $O(1)$ triple membership testing:

- `triple_set`: Set of all (h, r, t) tuples for instant lookup
- `outgoing[node]`: List of $(relation, target)$ pairs per node
- `incoming[node]`: List of $(relation, source)$ pairs per node

For each discovered rule, I computed:

- **Support**: Number of instances where the rule body is satisfied
- **Hits**: Instances where the rule head also holds
- **Confidence**: $\frac{\text{Hits}}{\text{Support}}$ — the probability the rule holds when applicable

2 Inverse Rules

2.1 Initial Attempt (Failed)

My first implementation searched for single inverse relations, checking whether $R_1(X, Y)$ implies some specific $R_2(Y, X)$. This returned **zero high-confidence rules**, which was immediately confusing given that family relationships clearly have inverses.

Mistake & Correction

The failure occurred because family inverse relations are **gender-split**. For example, `fatherOf(X,Y)` does not imply a single inverse — it implies `sonOf(Y,X)` **OR** `daughterOf(Y,X)` depending on Y's gender. Searching for a single inverse relation will always produce ∼50% confidence because the hits split across two possible inverse relations.

2.2 Corrected Approach

I created an inverse map that defines *sets* of valid inverses for each relation:

```
inverse_map = {
    'fatherOf':       ['sonOf', 'daughterOf'],
    'motherOf':       ['sonOf', 'daughterOf'],
    'grandfatherOf': ['grandsonOf', 'granddaughterOf'],
    'sisterOf':       ['brotherOf', 'sisterOf'],
    ...
}
```

The corrected algorithm checks whether *any* valid inverse exists, rather than requiring a specific one.

2.3 Results

The corrected approach discovered **18 inverse rule patterns**:

Table 1: Inverse Rules Discovered

Rule	Support	Hits	Confidence
`grandsonOf(X,Y)` → `grandfather/grandmotherOf(Y,X)`	814	814	100.00%
`grandfatherOf(X,Y)` → `grandson/granddaughterOf(Y,X)`	813	813	100.00%
`grandmotherOf(X,Y)` → `grandson/granddaughterOf(Y,X)`	813	813	100.00%
`sisterOf(X,Y)` → `brotherOf/sisterOf(Y,X)`	636	636	100.00%
`brotherOf(X,Y)` → `brotherOf/sisterOf(Y,X)`	570	570	100.00%
`auntOf(X,Y)` → `nephewOf/nieceOf(Y,X)`	556	556	100.00%
`daughterOf(X,Y)` → `fatherOf/motherOf(Y,X)`	628	540	85.99%
`sonOf(X,Y)` → `fatherOf/motherOf(Y,X)`	600	512	85.33%
`fatherOf(X,Y)` → `sonOf/daughterOf(Y,X)`	733	608	82.95%

2.4 The Parent-Child Incompleteness Discovery

The 82–86% confidence for parent-child inverse rules prompted a deeper investigation. I traced every case where the inverse was missing.

Key Finding

Quantified Incompleteness:

- Parent→Child edges (`fatherOf` + `motherOf`): 1,466 total
- Child→Parent edges (`sonOf` + `daughterOf`): 1,228 total
- Missing inverses from parent→child: **414 (28.2%)**
- Missing inverses from child→parent: **176 (14.3%)**
- Total missing inverse edges: **590**

This asymmetry reveals that the Knowledge Graph is systematically incomplete — not all relationships have their logical inverses recorded. This is common in real-world KGs and is precisely what link prediction models aim to solve.

As later confirmed in Task 4, the test set contains **exactly 590 triples** — the very edges identified as missing here. The train/test split was designed by intentionally removing parent-child inverse edges.

3 Compositional Rules (2-Hop)

3.1 Mining Approach

I exhaustively searched all (R_1, R_2, R_3) combinations for rules of the form:

$$R_1(X, Y) \wedge R_2(Y, Z) \rightarrow R_3(X, Z)$$

with confidence $\geq 90\%$ and support ≥ 20.

3.2 Results

The search yielded **224 high-confidence compositional rules**, with 194 achieving 100% confidence. These were semantically grouped into four categories.

3.2.1 Grandparent Rules (74 found)

Table 2: Selected Grandparent Rules (all 100% confidence)

Rule	Support
$\texttt{sisterOf}(X, Y) \wedge \texttt{granddaughterOf}(Y, Z) \rightarrow \texttt{granddaughterOf}(X, Z)$	772
$\texttt{grandmotherOf}(X, Y) \wedge \texttt{sisterOf}(Y, Z) \rightarrow \texttt{grandmotherOf}(X, Z)$	747
$\texttt{fatherOf}(X, Y) \wedge \texttt{fatherOf}(Y, Z) \rightarrow \texttt{grandfatherOf}(X, Z)$	564
$\texttt{motherOf}(X, Y) \wedge \texttt{motherOf}(Y, Z) \rightarrow \texttt{grandmotherOf}(X, Z)$	599

3.2.2 Uncle/Aunt Rules (63 found)

Table 3: Selected Uncle/Aunt Rules (all 100% confidence)

Rule	Support
$\texttt{auntOf}(X, Y) \wedge \texttt{brotherOf}(Y, Z) \rightarrow \texttt{auntOf}(X, Z)$	444
$\texttt{uncleOf}(X, Y) \wedge \texttt{sisterOf}(Y, Z) \rightarrow \texttt{uncleOf}(X, Z)$	337
$\texttt{auntOf}(X, Y) \wedge \texttt{uncleOf}(Y, Z) \rightarrow \texttt{greatAuntOf}(X, Z)$	313

3.2.3 Great-Grandparent Rules (64 found)

These capture extended generational chains, such as:

$$\texttt{grandmotherOf}(X, Y) \wedge \texttt{auntOf}(Y, Z) \rightarrow \texttt{greatGrandmotherOf}(X, Z)$$

with support 562 and 100% confidence.

3.2.4 Relation Participation Analysis

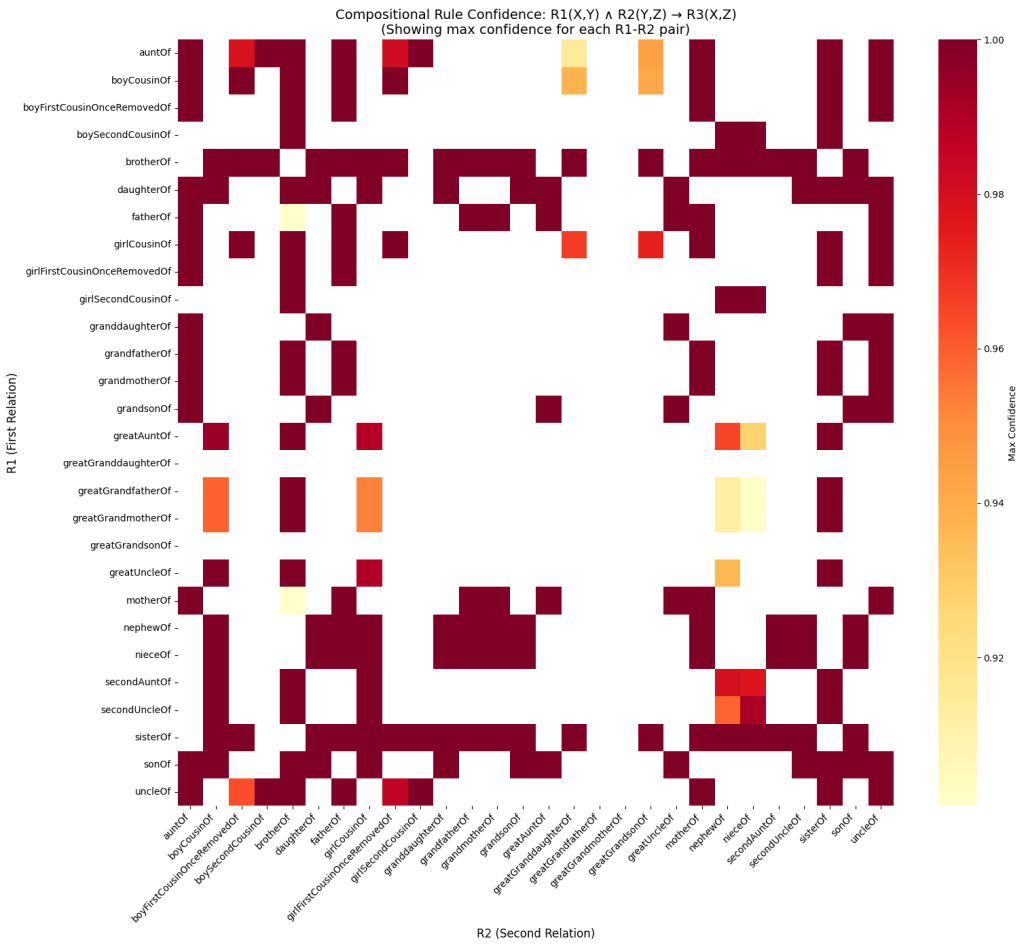

Figure 1: Compositional rule confidence heatmap. Bright cells indicate relation pairs (R_1, R_2) that produce high-confidence rules. Sibling relations (`brotherOf`, `sisterOf`) form dense clusters because siblings share all ancestors.

From the analysis of which relations participate most frequently in high-confidence rules:

Table 4: Relation Participation in High-Confidence Rules (≥95%)

As R_1 (first in chain)	**As R_2 (second)**	**As R_3 (inferred)**
sisterOf (19 rules)	brotherOf (18 rules)	secondAuntOf (16 rules)
brotherOf (19 rules)	sisterOf (17 rules)	secondUncleOf (16 rules)
daughterOf (13 rules)	auntOf (14 rules)	boyFirstCousinOnceRemovedOf (14)

Insight

Sibling relations (`brotherOf`, `sisterOf`) are the most *composable* — they appear most frequently in rule chains because siblings share all ancestors. The most commonly *inferred* relations are extended family types (second aunts, cousins once removed), confirming these are **derived rather than primary** relationships.

4 Symmetric Rules

4.1 Initial Confusion

I tested whether $R(X, Y) \rightarrow R(Y, X)$ holds for each relation. The results initially appeared problematic:

Relation	**Symmetry Rate**
`girlCousinOf`	53.03%
`sisterOf`	51.57%
`boyCousinOf`	46.55%
`brotherOf`	45.96%
All parent/child relations	0.00%

No relation exceeded 90% symmetry. I initially wrote a misleading insight claiming sibling relations "should be symmetric," which contradicted the data.

4.2 The Real Insight

Mistake & Correction

The ∼50% symmetry rate is **exactly correct** given how gendered relations work. `sisterOf(A,B)` means "A is a female sibling of B." The relation encodes A's gender but says nothing about B's.
Therefore:

- $\texttt{sisterOf}(A, B) \wedge \texttt{sisterOf}(B, A)$ — only when **both** are female
- $\texttt{sisterOf}(A, B) \wedge \texttt{brotherOf}(B, A)$ — when A is female, B is male

The ∼50% rate indicates approximately equal male/female distribution among siblings.

4.3 Verification

Table 5: Symmetric Rule Decomposition

	Total	Same-Gender Return	Cross-Gender Return
`sisterOf`	636	328 (51.6%)	308 (48.4%)
`brotherOf`	570	262 (46.0%)	308 (54.0%)
Completeness check (any sibling inverse exists):			
`sisterOf` with inverse		636/636 = **100.0%**	
`brotherOf` with inverse		570/570 = **100.0%**	

Key Finding

What initially appeared as "broken" symmetry is actually **perfect completeness** when accounting for cross-gender inverses. Every `sisterOf` edge has either a `sisterOf` or `brotherOf` edge in return, and vice versa. This is a lesson in understanding KG semantics before interpreting metrics.

5 Complex Rules (3-Hop)

5.1 Great-Grandparent Chains

I mined 3-hop rules of the form:

$$R_1(X, Y) \wedge R_2(Y, Z) \wedge R_3(Z, W) \rightarrow R_4(X, W)$$

focusing on parent-chain compositions.

Table 6: 3-Hop Rules (All 100% Confidence)

Rule	Support
`fatherOf` ∧ `fatherOf` ∧ `fatherOf` → `greatGrandfatherOf`	129
`motherOf` ∧ `fatherOf` ∧ `fatherOf` → `greatGrandmotherOf`	129
`fatherOf` ∧ `motherOf` ∧ `fatherOf` → `greatGrandfatherOf`	121
`motherOf` ∧ `motherOf` ∧ `fatherOf` → `greatGrandmotherOf`	121
`fatherOf` ∧ `fatherOf` ∧ `motherOf` → `greatGrandfatherOf`	99
`motherOf` ∧ `fatherOf` ∧ `motherOf` → `greatGrandmotherOf`	99
`fatherOf` ∧ `motherOf` ∧ `motherOf` → `greatGrandfatherOf`	96
`motherOf` ∧ `motherOf` ∧ `motherOf` → `greatGrandmotherOf`	96

These 8 rules represent **2 semantic patterns** (great-grandfather and great-grandmother) with 4 gender variants each for the intermediate parent generations. The head relation's gender depends *only* on the first relation in the chain (the great-grandparent's own parent relation), not on intermediate generations.

6 Gender-Aware Rule Analysis

★ *This analysis goes beyond the required rule types to investigate gender-conditioned patterns.*

6.1 The Uncle/Aunt Semantic Direction Error

I initially tested the rule:

$$\texttt{parentOf}(X, Y) \wedge \texttt{brotherOf}(X, Z) \rightarrow \texttt{uncleOf}(Z, Y)$$

and obtained only **41.30% confidence**. This seemed far too low for a definitional rule.

Mistake & Correction

The bug was edge direction interpretation.
`brotherOf(X, Z)` means "X is brother of Z" (X is male). But I was looking for "Z is brother of X" — the wrong direction. To find X's brothers, I needed edges where Z is the head: $\texttt{brotherOf}(Z, X)$.
This single semantic error produced a 60% false failure rate.

6.2 Corrected Results

Table 7: Gender-Aware Rules After Correction

Rule	Support	Hits	Confidence
$\texttt{brotherOf}(Z, X) \wedge \texttt{parentOf}(X, Y) \rightarrow \texttt{uncleOf}(Z, Y)$	407	407	**100.0%**
$\texttt{sisterOf}(Z, X) \wedge \texttt{parentOf}(X, Y) \rightarrow \texttt{auntOf}(Z, Y)$	485	485	**100.0%**

Insight

Edge direction matters enormously in knowledge graphs. A careless interpretation of `brotherOf(A,B)` led to a 60% error rate that disappeared entirely once the semantics were corrected. This reinforces why understanding relation definitions is critical *before* mining rules.

7 Constraint Mining: Data Quality Verification

★ *Constraint mining was not required by the task but provides valuable data quality assurance for downstream link prediction.*

Beyond discovering positive rules, I verified that logical impossibilities are absent from the data.

Table 8: Constraint Verification Results

#	Constraint	Result
1	No self-parentage: $\texttt{fatherOf}(X, X)$	**PASSED**
2	No mutual parentage: $\texttt{fatherOf}(X, Y) \wedge \texttt{fatherOf}(Y, X)$	**PASSED**
3	No self-sibling: $\texttt{brotherOf}(X, X)$	**PASSED**
4	No generational paradox: parent is grandchild of own child	**PASSED**
5	Gender consistency: no entity is both father and mother	**PASSED**
6	Brother is male: no entity with both `brotherOf` and `motherOf`	**PASSED**
	Overall	**6/6 PASSED**

The MetaFam dataset has excellent internal consistency — no logical impossibilities exist. This is valuable to verify before using the data for link prediction, where data quality directly impacts model learning.

8 Summary and Statistics

8.1 Rules Discovered

Table 9: Complete Rule Mining Statistics

Rule Type	Count	At 100%	Notes
Inverse	18	12	3 at 82–86% (parent-child)
Compositional (2-hop)	224	194	18 at 95–99%, 12 at 90–94%
Symmetric	0 truly symmetric	—	100% complete with cross-gender
Complex (3-hop)	8	8	All great-grandparent patterns
Total	**250**	**214**	

8.2 Key Takeaways

1. **KG incompleteness is quantifiable.** 28.2% of parent→child edges lack their inverse. This incompleteness is exactly what makes link prediction valuable and directly informed the Task 4 test set design.

2. **Gendered relations require careful handling.** What looks like "broken" symmetry (∼50%) is actually correct behavior. Inverse rules must check for gender-split alternatives (`sonOf` OR `daughterOf`), not single relations.

3. **Edge direction semantics matter.** My uncle/aunt analysis failed at 41% confidence purely due to misinterpreting edge direction. The corrected analysis showed 100% confidence. This mistake, while embarrassing, demonstrates why semantic understanding must precede algorithmic analysis.

4. **Sibling relations are most composable.** `brotherOf` and `sisterOf` appear most frequently in high-confidence rule chains because siblings share all ancestors.

5. **Extended family relations are derived.** Relations like `secondAuntOf` and `firstCousinOnceRemovedOf` are most commonly the *result* of compositional rules, suggesting these are inferred rather than primary relationships.

6. **Data quality is excellent.** All 6 logical constraints passed, confirming the synthetic dataset is internally consistent and suitable for downstream tasks.

8.3 Connection to Task 4

The discovery that 590 parent-child inverse edges are missing from the training data directly motivates and informs the link prediction task. As shown in Task 4, the test set consists of precisely these 590 missing edges, making rule mining not just an analytical exercise but a direct prerequisite for understanding what the prediction models need to learn.

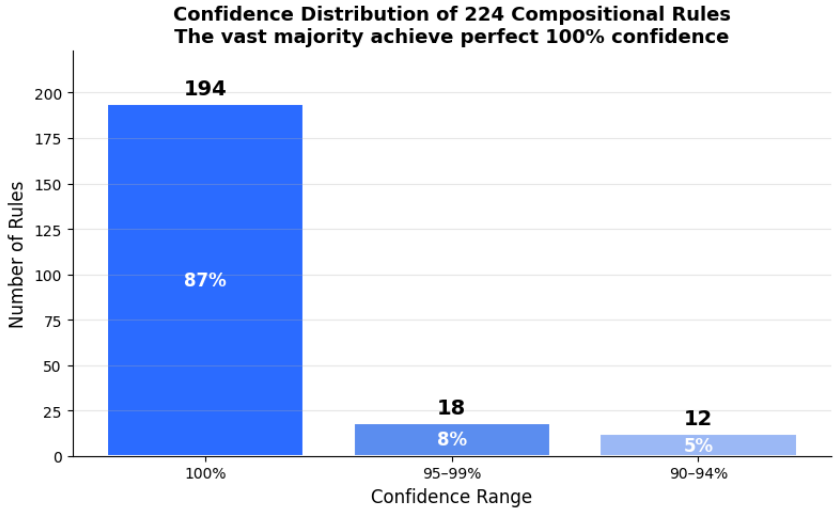

Figure 2: Confidence distribution of 224 compositional rules. The overwhelming majority (87%) achieve 100% confidence, indicating highly consistent family logic in the dataset.