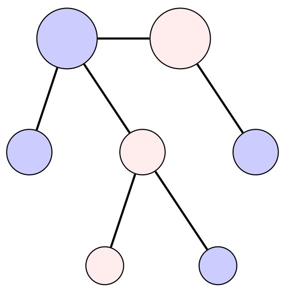
MetaFam Knowledge Graph

Task 1: Dataset Exploration

"Makes a diff'rence, havin' a decent family" — Rubeus Hagrid



Precog Research Task

Knowledge Graph Analysis on Family Networks

February 2026

Contents

1 Introduction

1.1 Background

Knowledge Graphs (KGs) provide a structured representation of real-world entities and their relationships, stored as **(head, relation, tail)** triples. For example, the triple `(Alice, motherOf, Bob)` represents that Alice is the mother of Bob, giving textual and semantic meaning to graph structures.

MetaFam is a synthetic family knowledge graph that models familial relationships between individuals. This report presents a comprehensive exploration of the MetaFam dataset, analyzing its structure, computing relevant graph metrics, and extracting qualitative insights about family network organization.

1.2 Objectives

The primary objectives of this exploration task are:

1. **Load and understand** the dataset structure
2. **Compute relevant statistics** appropriate for family knowledge graphs
3. **Create meaningful visualizations** to support findings
4. **Identify important individuals** using graph centrality metrics
5. **Understand hierarchical structure** through generational analysis
6. **Extract qualitative insights** about family network patterns

1.3 Theoretical Foundation

1.3.1 Graph Representation

The MetaFam dataset is represented as both:

- **Directed Graph (DiGraph)**: Preserves semantic direction of relationships (e.g., `fatherOf` implies parent→child direction). Essential for genealogical analysis.
- **Undirected Graph**: Treats relationships as bidirectional connections. Useful for community detection and clustering analysis.

1.3.2 Why Graph Analysis for Genealogy?

- **Centrality metrics** identify important individuals (patriarchs, matriarchs)
- **Connected components** reveal separate family lineages
- **Generational depth** maps the family tree hierarchy
- **Clustering coefficient** measures the closure of familial relationships

2 Dataset Overview

2.1 Basic Statistics

The MetaFam knowledge graph contains the following basic statistics:

Table 1: MetaFam Dataset Summary Statistics

Metric	Value	Description
Total Nodes (People)	1,316	Unique individuals in the graph
Total Edges (Relations)	13,821	Family relationship triples
Unique Relationship Types	28	Distinct family relation categories
Average Degree	10.50	Mean connections per person

2.2 Relationship Types

The 28 unique relationship types in MetaFam can be categorized into six main groups:

Table 2: Relationship Type Categories

Category	Count	Relations
Parent-Child	12	fatherOf, motherOf, sonOf, daughterOf, grandfatherOf, grandmotherOf, grandsonOf, granddaughterOf, greatGrandfatherOf, greatGrandmotherOf, greatGrandsonOf, greatGranddaughterOf
Sibling	2	brotherOf, sisterOf
Aunt/Uncle	8	auntOf, uncleOf, nieceOf, nephewOf, greatAuntOf, greatUncleOf, secondAuntOf, secondUncleOf
Cousin	6	boyCousinOf, girlCousinOf, boySecondCousinOf, girlSecondCousinOf, boyFirstCousinOnceRemovedOf, girlFirstCousinOnceRemovedOf
Spouse	0	(Not present in dataset)

Key Observation: The absence of spouse relations (`husbandOf`, `wifeOf`) in the dataset is notable. This suggests the graph focuses on blood relations rather than marriage connections, which has implications for link prediction tasks.

3 Structural Analysis

3.1 Global Network Metrics

3.1.1 Graph Density

$$\text{Density} = \frac{|E|}{|V| \times (|V| - 1)} = \frac{13,821}{1,316 \times 1,315} = 0.007987 \tag{1}$$

The graph density of **0.008** indicates a **very sparse network**, which is characteristic of social and family networks. This sparsity arises because each individual connects only to a limited set of relatives, not to everyone in the extended family network.

Interpretation: In a family network, even with 1,316 people, each person only has direct relationships with approximately 10-11 others on average, resulting in the observed sparse connectivity pattern.

3.1.2 Clustering Coefficient

$$\text{Average Clustering Coefficient} = 0.7346 \tag{2}$$

The **high clustering coefficient (0.73)** indicates strong transitivity in family relationships. This means:

- If person A is related to B, and B is related to C, there's a high probability A is also related to C
- Family structures naturally form tight-knit clusters
- Siblings share parents, cousins share grandparents, etc.

3.2 Connectivity Analysis

Table 3: Connected Component Statistics

Metric	**Value**
Weakly Connected Components	50
Largest Component Size	27 nodes
Smallest Component Size	26 nodes
Average Component Size	26.3 nodes

Key Insights:

1. **Forest Structure:** The graph is a **forest** consisting of 50 separate family trees, not a single connected component.
2. **Uniform Family Sizes:** Component sizes are remarkably uniform (26-27 nodes), suggesting synthetically generated families of similar structure.
3. **No Bridges Between Families:** The absence of spouse relations means families remain isolated—in real genealogy, marriages would create bridges between family clusters.

3.3 Degree Distribution

Table 4: Degree Statistics

Metric	Min	Max	Mean
In-Degree	0	23	10.50
Out-Degree	1	22	10.50

Asymmetry Analysis:

In directed family knowledge graphs with relations like `fatherOf`:

- **High Out-Degree**: Indicates ancestors (they have many `parentOf`, `grandparentOf` relations pointing out)
- **High In-Degree**: Indicates individuals with many relations pointing TO them (heavily referenced)

4 Generational Analysis

4.1 Generation Computation Algorithm

Generational depth is computed using a BFS-based algorithm:

1. Extract parental subgraph (only `fatherOf`, `motherOf` edges)
2. Identify root nodes (in-degree = 0 in parental subgraph)
3. Run BFS from roots, assigning generation levels
4. Handle disconnected components separately

4.2 Generation Distribution

Table 5: Generation Distribution in MetaFam

Generation	Count	Percentage
Generation 0 (Great-grandparents)	519	39.4%
Generation 1 (Grandparents)	572	43.5%
Generation 2 (Parents)	216	16.4%
Generation 3 (Children)	9	0.7%
Total	**1,316**	**100%**

Key Insights:

1. **4 Generations**: The family graph spans 4 generations (0-3), representing great-grandparents through great-grandchildren.

2. **Inverted Pyramid**: Newer generations (Gen 2, 3) have significantly fewer members, reflecting typical family tree structures where older generations have more accumulated members over time.
3. **Mean Generation = 0.78**: The average person is in an early generation, consistent with the pyramid structure.
4. **Few Youngest Members**: Only 9 individuals (0.7%) are in Generation 3, the most recent generation.

4.3 Generational Relevance for Link Prediction

Generation information is crucial for predicting relationships:

- `fatherOf` relations only occur from Gen n to Gen $n+1$
- `grandparentOf` relations span two generations
- `siblingOf` relations occur within the same generation
- `cousinOf` relations exist between individuals of similar generations in different branches

5 Gender Classification

5.1 Rule-Based Inference

Gender is inferred deterministically based on relationship semantics where a node appears as the **HEAD** (source) of a relation:

Male-indicating relations:

```
fatherOf, brotherOf, sonOf, uncleOf, grandfatherOf,
nephewOf, boyCousinOf, grandsonOf, greatUncleOf, ...
```

Female-indicating relations:

```
motherOf, sisterOf, daughterOf, auntOf, grandmotherOf,
nieceOf, girlCousinOf, granddaughterOf, greatAuntOf, ...
```

5.2 Gender Distribution

Table 6: Gender Distribution

Gender	Count	Percentage
Female	670	50.9%
Male	646	49.1%
Unknown	0	0%
Unmapped (conflicts)	0	0%
Total	**1,316**	**100%**

Key Insights:

1. **Balanced Distribution**: Near-perfect gender balance (51% female, 49% male)
2. **Complete Classification**: 100% of nodes successfully classified (no unknowns or conflicts)
3. **Data Quality**: Zero unmapped nodes indicates no data inconsistencies (no individual assigned conflicting gender-based relations)
4. **Link Prediction Utility**: Gender is a strong constraint for predicting gender-specific relations (`brotherOf` vs `sisterOf`)

6 Centrality Analysis

6.1 PageRank Centrality

PageRank measures node importance based on the quality of incoming connections:

$$PR(v) = \frac{1-d}{N} + d \sum_{u \to v} \frac{PR(u)}{out(u)} \tag{3}$$

where $d = 0.85$ (damping factor) and N = number of nodes.

Table 7: PageRank Statistics

Metric	Value
Minimum	0.000114
Maximum	0.001857
Mean	0.000760

6.1.1 Top Individuals by PageRank

Table 8: Top 10 Individuals by PageRank ("Important Ancestors")

Rank	Node	PageRank	Generation
1	gabriel241	0.001857	2
2	lea1165	0.001841	2
3	raphael29	0.001809	2
4	christian712	0.001682	2
5	tobias713	0.001682	2
6	emilia428	0.001676	2
7	simon172	0.001644	2
8	victoria279	0.001631	2
9	benjamin952	0.001603	1
10	helena1135	0.001571	2

Observation: Most high-PageRank individuals are in Generation 2, indicating they are "important ancestors" with many descendants pointing to them through various relationships.

6.2 Betweenness Centrality

Betweenness centrality measures how often a node lies on shortest paths:

$$B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{4}$$

Table 9: Top 5 "Bridge" Individuals by Betweenness

Rank	Node	Betweenness
1	lea1165	0.0001
2	valentin638	0.0001
3	gabriel241	0.0001
4	nora536	0.0001
5	stefan1192	0.0001

Key Insight: Betweenness centrality values are very low (max = 0.0001), indicating **few bridge individuals** in the network. This is consistent with the forest structure—without inter-family marriages, there are no natural bridges connecting different family clusters.

6.3 Degree-Based Importance

6.3.1 Top by Out-Degree (Ancestors)

Table 10: Top 5 Individuals by Out-Degree

Rank	Node	Out-Degree	Generation
1	oskar133	22	2
2	larissa136	22	2
3	fabian140	22	2
4	laura143	22	2
5	dominik1036	22	2

High out-degree indicates many outgoing relations (`parentOf`, `grandparentOf`), identifying **prolific ancestors**.

7 Qualitative Insights

7.1 Structural Characteristics

1. **Sparse Graph**: Family knowledge graphs are inherently sparse (density < 0.01) because individuals only connect to immediate and extended family, not the entire population.

2. **No Self-Loops**: The graph contains no self-loops (a person cannot be their own relative), which is domain-appropriate.

3. **Forest Structure**: The graph is a forest of 50 disjoint family trees. In real genealogy, marriages would create bridges, but this synthetic dataset lacks spouse relations.

4. **Directed Edges**: Edge direction encodes semantic meaning (parent→child vs child→parent).

7.2 Implications for Downstream Tasks

7.2.1 Community Detection (Task 2)

- Each connected component is already a natural "community" (family unit)
- Within families, sub-communities may correspond to nuclear family units
- Generation boundaries may define community structure

7.2.2 Rule Mining (Task 3)

- Transitive rules: $(X, \text{fatherOf}, Y) \wedge (Y, \text{fatherOf}, Z) \rightarrow (X, \text{grandfatherOf}, Z)$
- Inverse rules: $(X, \text{fatherOf}, Y) \rightarrow (Y, \text{sonOf}, X)$
- Gender constraints: $(X, \text{motherOf}, Y) \rightarrow \text{gender}(X) = \text{Female}$

7.2.3 Link Prediction (Task 4)

Node attributes are valuable features:

- **Generation**: Constrains relation types (parent-child spans 1 gen, grandparent spans 2)
- **Gender**: Determines gender-specific relations (`brotherOf` vs `sisterOf`)
- **Community**: Relationships more likely within the same family cluster
- **PageRank**: High-PageRank nodes more likely to have additional links
- **Degree**: Highly connected individuals have more potential for additional links

8 Node Attributes for Export

The following attributes are computed and stored for each node, exported to GEXF format for Gephi visualization:

Table 11: Node Attributes Stored in Graph

Attribute	Type	Description
`in_degree`	int	Number of incoming edges
`out_degree`	int	Number of outgoing edges
`pagerank`	float	PageRank centrality score (0-1)
`generation`	int	Generational depth (0 = oldest, -1 = unassigned)
`gender`	str	Male / Female / Unknown / Unmapped

Note: Betweenness centrality is computed for analysis but **not stored** as a node attribute to reduce export file size.

8.1 Gephi Visualization Recommendations

1. **Color nodes by**: `gender` attribute (blue = male, pink = female)
2. **Size nodes by**: `pagerank` or `out_degree`
3. **Vertical layout**: Use `generation` for Y-axis positioning
4. **Layout algorithm**: ForceAtlas2 works well for family tree visualization

9 Summary

9.1 Key Findings

Table 12: Summary of Key Findings

Aspect	Finding
Scale	1,316 people, 13,821 relations, 28 relation types
Structure	Forest of 50 families, 26-27 members each
Sparsity	Density = 0.008 (typical for social networks)
Clustering	High transitivity (0.73) due to family structure
Depth	4 generations, pyramid distribution
Gender	Balanced (51% F, 49% M), fully classified
Central Figures	Identified via PageRank, mostly Gen 2 ancestors

9.2 Output Files

- `outputs/gephi/metafam_task1_refined.gexf` — Enriched graph for Gephi
- `outputs/plots/relationship_distribution.png` — Bar chart of relation frequencies
- `outputs/plots/degree_distribution.png` — In/out degree histograms
- `outputs/plots/generation_distribution.png` — Generation histogram
- `outputs/plots/gender_distribution.png` — Gender bar chart

A Complete Relationship List

1. auntOf
2. boyCousinOf
3. boyFirstCousinOnceRemovedOf
4. boySecondCousinOf
5. brotherOf
6. daughterOf
7. fatherOf
8. girlCousinOf
9. girlFirstCousinOnceRemovedOf
10. girlSecondCousinOf
11. granddaughterOf
12. grandfatherOf
13. grandmotherOf
14. grandsonOf

15. greatAuntOf
16. greatGranddaughterOf
17. greatGrandfatherOf
18. greatGrandmotherOf
19. greatGrandsonOf
20. greatUncleOf
21. motherOf
22. nephewOf
23. nieceOf
24. secondAuntOf
25. secondUncleOf
26. sisterOf
27. sonOf
28. uncleOf

B Theoretical Formulas Reference

B.1 Graph Density

For a directed graph $G = (V, E)$:

$$D = \frac{|E|}{|V| \cdot (|V| - 1)} \tag{5}$$

B.2 Clustering Coefficient

For node v with neighbors $N(v)$:

$$C(v) = \frac{|\{e_{jk} : v_j, v_k \in N(v), e_{jk} \in E\}|}{|N(v)| \cdot (|N(v)| - 1)} \tag{6}$$

B.3 PageRank

Iterative formula with damping factor $d = 0.85$:

$$PR(v) = \frac{1 - d}{N} + d \sum_{u \in B(v)} \frac{PR(u)}{L(u)} \tag{7}$$

where $B(v)$ is the set of nodes linking to v, and $L(u)$ is the out-degree of u.

B.4 Betweenness Centrality

$$B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{8}$$

where σ_{st} is the number of shortest paths from s to t, and $\sigma_{st}(v)$ is the number passing through v.