NLP Final Project

Yamit Ehrlich - 313600751

https://github.com/yamitehr/geneval

GENEVAL (Genealogical Evaluation): Evaluating Multi-Hop Reasoning Limitations in Large Language Models

Abstract. I present GENEVAL, a benchmark of 501 questions across 7 reasoning tasks over the British Royal Family genealogy. Evaluating Gemini 2.5 Flash and Pro, I found that while both models achieve ~90% on simple queries, multi-hop reasoning* degrades significantly, converging to ~40% at 6 hops regardless of model size.

*Multi-hop reasoning: traversing multiple relationship links to answer a query (e.g., "Who is the mother of the father of X?").

1. Introduction. LLMs are increasingly used to explore historical data and are often treated as authoritative truth, and the British Royal Family, widely discussed online and extensively documented across centuries, provides a rich domain to test whether such perceived factual reliability holds under reasoning.
I found that LLMs achieve 90%+ accuracy on simple genealogical queries, demonstrating they possess the necessary knowledge. However, performance degrades sharply when traversing multiple relationship hops - indicating that the limitation lies in reasoning rather than knowledge. No existing benchmark tests this specifically, so I created **GENEVAL**: 501 questions over 314 British Royal Family members from Wikidata [3].

Recent work supports my findings: Yang et al. [1] found multi-hop evidence is "substantial for the first hop but only moderate for subsequent hops," and Wang et al. [2] showed that scaling up model size does not improve multi-step reasoning.

2. GENEVAL Dataset:

1. **Data Collection**: I queried Wikidata [3] using SPARQL to extract 314 British Royal Family members into a CSV with structured metadata: birth/death dates, parents, spouses, reign periods, and royal house (see **Appendix A** for details).
2. **Question Generation**: The raw data is preprocessed to compute derived relationships (e.g., siblings from shared parents, lifespans from dates). Task generators then programmatically create questions by sampling people and relationships, with ground-truth answers computed directly from the structured data. Each question also carries metadata (royal house, time period, gender, etc.) derived from the people it references, enabling analysis across multiple dimensions (see **Appendix B** for details and example).
3. **GENEVAL Benchmark**: 501 questions, 7 task types, equal difficulty distribution (33/33/33).

Task Type	Count	Difficulty Criteria	Example
Multi-Hop Reasoning	99 (20%)	Easy: 1-2 hops; Medium: 3-4 hops; Hard: 5-6 hops	Who is the mother of the father of the father of Henry VIII?
Temporal Reasoning	72 (14%)	Easy: 2-person; Medium: 3-person; Hard: 4-5 person ranking	Who was born first: Elizabeth I or Mary I?
Negative Reasoning	72 (14%)	Easy: 2-3 options; Medium: 4-5 options; Hard: nested negations	Which of these was NOT a child of George III?
Sibling Inference	72 (14%)	Easy: direct sibling; Medium: half-sibling; Hard: sibling count	Are Edward VI and Elizabeth I full or half siblings?
Constraint Satisfaction	63 (13%)	Easy: 2 constraints; Medium: 3; Hard: 4+ constraints	Name a king who ruled before 1400 and lived past 60
Adversarial Ambiguity	63 (13%)	Easy: unique name; Medium: 2 same-name; Hard: 3+ same-name	Which Edward died first: the father or the son?
Comparative Lifespan	60 (12%)	Easy: simple comparison; Medium: reign overlap; Hard: multi-person	Who lived longer, Henry VII or Henry VIII?

3. Methodology.

- **Models:** Gemini 2.5 Flash and Gemini 2.5 Pro

- **Evaluation:** Accuracy using LLM-as-judge [4] for answer equivalence
- **Secondary metric:** Accuracy by Hop Count (compositional degradation curve)

4. Results.

E/M/H = Easy/Medium/Hard accuracy breakdown. Our analysis tool also segments by royal house, time period, century, people involved, monarch status, gender composition, relationship type, dynasty, and hop count (see **Appendix C** for details and example).

Task Type	Flash	Flash E/M/H	Pro	Pro E/M/H
Adversarial Ambiguity	100%	100/100/100	98%	100/100/93
Temporal Reasoning	97%	100/100/90	99%	100/100/95
Sibling Inference	97%	100/92/100	90%	100/81/94
Comparative Lifespan	95%	100/92/100	**78%**	100/78/58
Negative Reasoning	89%	100/100/**66**	82%	96/95/**54**
Constraint Satisfaction	**81%**	76/71/95	**73%**	66/76/**76**
Multi-Hop Reasoning	**71%**	**90/81/39**	**66%**	**81/63/51**
Overall	**89%**	**94/92/81**	**83%**	**89/85/75**

Key findings: 1. **Hard Multi-Hop is the weakest category** for both models (39% Flash, 51% Pro), confirming multi-hop reasoning as a fundamental limitation 2. **Hard Negative Reasoning also reveals weakness** (66% Flash, 54% Pro), suggesting nested negations are challenging 3. **Both models achieve near-perfect accuracy** on Adversarial Ambiguity and Temporal Reasoning (97-100%) 4. **Flash outperforms Pro by 6% overall** (89% vs 83%)

Multi-Hop Degradation Analysis.

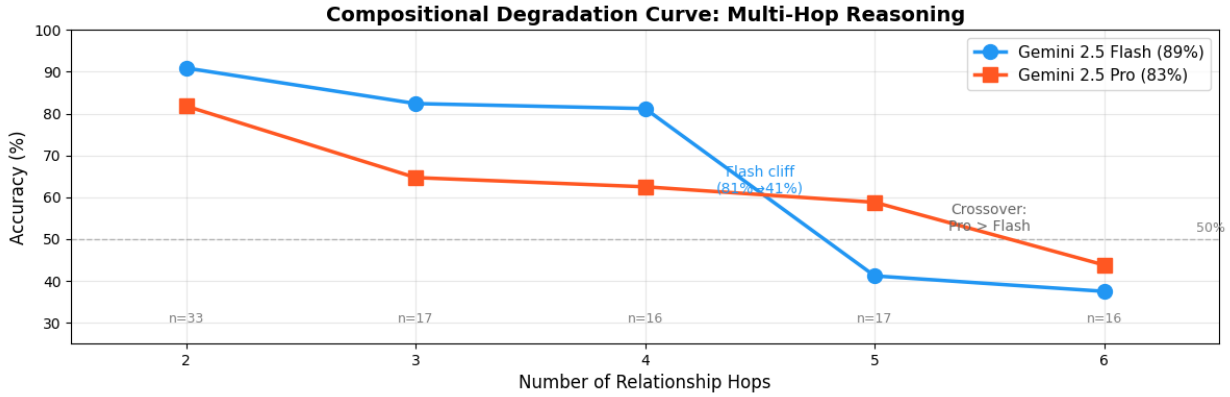

Figure 1: Compositional degradation curves showing Flash's sharp cliff at 5 hops and Pro's gradual decline starting at 3 hops.

Key Finding: Both models degrade as hop count increases, converging to ~40% accuracy at 6 hops - a fundamental ceiling for multi-hop genealogical reasoning. Flash maintains >80% through 4 hops before a sharp drop, while Pro degrades more gradually starting at 3 hops.6.

5. Contributions.

1. **Novel benchmark:** GENEVAL, the first genealogy-specific multi-hop reasoning dataset (501 questions, created from scratch) 2. **Fundamental limitation identified:** Both models converge to ~40% accuracy at 6 hops, revealing a ceiling in multi-hop genealogical reasoning 3. **Degradation patterns:** Performance degrades as hop count increases, with both models dropping below 50% at 5+ hops 4. **Practical implication:** For reliable genealogical queries, limit relationship chains to 2-3 hops.

References

[1] Yang, S., et al. (2024). *Do Large Language Models Latently Perform Multi-Hop Reasoning?* ACL 2024.

[2] Wang, P., et al. (2024). *Do Large Language Models Have Compositional Ability? An Investigation into Limitations and Scalability.* arXiv:2407.15720.

[3] Wikidata. https://www.wikidata.org/

[4] Zheng, L., et al. (2023). *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena.* NeurIPS 2023.

Appendixes

Appendix A: Dataset Schema and Preprocessing

This appendix details the structure of the underlying dataset and the preprocessing steps applied before question generation.

A.1 Raw Data Source

The genealogical data is stored in **british_royal_family_FINAL.csv**. This file was created by querying the Wikidata SPARQL endpoint [3] to extract publicly available information on 314 members of the British Royal Family and their direct relations.

A.2 Data Schema

The CSV file contains the following 13 columns:

Column	Description
first_name	The first name of the individual.
last_name	The last name of the individual.
date_of_birth	The birth date (YYYY-MM-DD).
date_of_death	The death date (YYYY-MM-DD), if applicable.
mother	The canonical name of the individual's mother.
father	The canonical name of the individual's father.
reign_start	The date the individual's reign began, if a monarch.
reign_end	The date the individual's reign ended, if a monarch.
house	The royal house the individual belonged to (e.g., House of Windsor).
spouse	The canonical name of the individual's spouse.
canonical_name	The standardized full name used for unique identification.
wikidata_uri	The unique identifier for the entity in Wikidata.
gender	The gender of the individual.

A.3 Data Preprocessing

The raw CSV data is processed by the RoyalFamilyDataLoader class found in tasks/data_loader.py. This class is responsible for:

1. **Loading**: Reading the CSV file into an in-memory dictionary, keyed by the canonical_name.

2. **Canonicalization**: Ensuring that all references to individuals (e.g., in mother, father, and spouse columns) use the consistent canonical_name, preventing ambiguity.
3. **Graph-like Structure**: The data is loaded into a map where each person is a key. This structure represents the family tree as a graph, allowing relationships to be traversed efficiently by looking up connected individuals.

Appendix B: Questions Generation Algorythm

This section describes the framework used to programmatically generate evaluation questions.

B.1 Framework Overview

The generation logic is built on a modular design. A base class, TaskGenerator (task_base.py), provides core functionalities like creating question dictionaries. Each of the 7 task types inherits from this base class and implements its own specific generate_questions logic. This design allows for the easy addition of new reasoning tasks.

B.2 Detailed Example: Multi-Hop Reasoning

The multi-hop reasoning task (tasks/task_1_multihop.py) is a primary example of the generation process. The goal is to create a question that requires chaining multiple relationships together.
The algorithm works as follows:

1. **Determine Hop Count**: Based on the desired difficulty (Easy, Medium, or Hard), the algorithm selects a number of "hops" or relationship steps (e.g., 2 for Easy, 5-6 for Hard).
2. **Generate a Random Path**: It creates a random sequence of relationship types. For a 3-hop question, a potential path could be ['father', 'mother', 'spouse']. For harder questions, 'child' relationships are included to increase complexity.
3. **Find a Valid Path**: The algorithm picks a random person from the dataset (e.g., *George V*). It then attempts to "walk" the generated path starting from that person:
    - Find the 'father' of George V → *Edward VII*.
    - From Edward VII, find the 'mother' → *Queen Victoria*.
    - From Queen Victoria, find the 'spouse' → *Albert, Prince Consort*.
4. **Validate and Finalize**: If the path can be completed without hitting a dead end (e.g., a person with no recorded spouse), the person at the end of the path (*Albert, Prince Consort*) is designated as the ground-truth **answer**. If the path fails, the algorithm discards it and starts over.
5. **Formulate the Question**: The natural language question is constructed by reversing the path: "Who is the **spouse** of the **mother** of the **father** of George V?"

This programmatic approach ensures that all questions are valid, have a known correct answer derived from the source data, and can be generated at scale with controllable difficulty.

B.3 Generated Question Format Example

The generated questions, stored in data/examples.jsonl, follow a standardized JSON structure. Each entry in the .jsonl file represents a single question and its associated metadata:


```
{
  "question": "Who is the father of the mother of Edward III of England (1312–1377)?",
  "answer": "Henry I of England (1068–1135)",
  "task_type": "multi_hop",
  "task_name": "Multi-Hop Relationship Traversal",
  "difficulty": "medium",
  "metadata": {
    "hops": 3,
    "start_person": "Edward III of England (1312–1377)",
    "path": ["mother", "father"],
    "requires_compositional_reasoning": true
  }
}
```

Yamit Ehrlich - 313600751

https://github.com/yamitehr/geneval

Appendix C: Example of a Detailed Evaluation Results
This appendix provides the complete, detailed breakdown of evaluation results for the gemini-2.5-pro model across all 501 questions. The table below, generated by the modified analyze_results.py script, segments the model's accuracy by numerous factors:

- **Task Type** - Performance on each of the 7 reasoning tasks
- **Difficulty** - Easy, medium, hard breakdown
- **Royal House** - Tudor, Stuart, Windsor, etc.
- **Time Period** - Medieval, Early Modern, Modern
- **Century** - 9th through 21st century
- **People Involved** - Number of entities in each question
- **Monarch Involvement** - Questions involving monarchs vs. non-monarchs
- **Gender Composition** - All male, all female, or mixed
- **Relationship Type** - Parent-child, sibling, spouse, etc.
- **Dynasty** - Single house vs. cross-house questions

Segmentation	Category	Accuracy	Correct/Total
Task Type	Adversarial Ambiguity	98.4%	62/63
	Comparative Lifespan Reasoning	78.3%	47/60
	Constraint Satisfaction	73.0%	46/63
	Multi-Hop Relationship Traversal	65.7%	65/99
	Negative Reasoning	81.9%	59/72
	Sibling Inference	90.3%	65/72
	Temporal Reasoning	98.6%	71/72
Difficulty	Easy	89.0%	113/127
	Medium	85.4%	182/213
	Hard	74.5%	120/161
Royal House	Commonwealth	100.0%	1/1
	House of Denmark	100.0%	3/3
	House of Hanover	87.5%	14/16
	House of Lancaster	71.4%	10/14
	House of Normandy	78.7%	59/75
	House of Plantagenet	86.9%	186/214
	House of Saxe-Coburg and Gotha	100.0%	3/3
	House of Stuart	79.5%	132/166
	House of Tudor	88.5%	85/96
	House of Wessex	80.5%	70/87
	House of Windsor	91.4%	32/35
	House of York	70.0%	28/40
Time Period	Medieval	83.6%	317/379
	Early Modern	81.5%	154/189
	Modern	85.7%	60/70
Century (top 5)	9th Century	89.5%	17/19
	10th Century	85.4%	35/41
	11th Century	73.9%	51/69
	12th Century	82.9%	63/76
	13th Century	87.2%	82/94
People Involved	1 person	83.1%	212/255
	2 people	87.0%	87/100
	3 people	81.9%	68/83

NLP Final Project

Yamit Ehrlich - 313600751

https://github.com/yamitehr/geneval

	4 people	81.8%	27/33
	5 people	91.7%	11/12
	6 people	0.0%	0/2
Monarch	All monarchs	75.7%	103/136
	Has monarch	85.6%	83/97
	No monarchs	85.4%	229/268
Gender	All males	83.9%	209/249
	All females	86.2%	94/109
	Mixed	80.3%	102/127
Relationship	Constraint	73.0%	46/63
	Disambiguation	98.4%	61/62
	Multi-Generational	65.2%	58/89
	Negative	82.5%	33/40
	Parent-Child	83.0%	39/47
	Sibling	91.2%	52/57
	Spouse	70.0%	7/10
	Temporal	89.5%	119/133
Dynasty	Single house	83.2%	263/316
	Cross-house	82.2%	152/185
Hop Count	2 hops	81.8%	27/33
	3 hops	64.7%	11/17
	4 hops	62.5%	10/16
	5 hops	58.8%	10/17
	6 hops	43.8%	7/16
Overall Accuracy: 82.8%			