LLM Reasoning & Personality Knowledge Graph Challenge

Overview

This report presents the development of a Python-based solution for the LLM Reasoning & Personality Knowledge Graph Challenge, where the objective is to construct a knowledge graph from text documents and model the personality of the subjects within it. The task focuses on demonstrating logical reasoning, research ability, and effective use of large language models (LLMs) for both guidance and implementation.

The solution uses synthetic cover letters generated with ChatGPT (GPT-5) as the input dataset, chosen for their natural combination of structured information (names, roles, organizations) and expressive language (traits, motivations, tone). Each letter is processed through a multi-stage LLM workflow — including preprocessing, entity and relationship extraction, personality inference, and graph construction — to produce an interpretable Personality Knowledge Graph.

All design decisions, implementation steps, and evaluation methods were developed with LLM assistance and justified in this report. The workflow demonstrates how generative models can be used not only for code generation but also for semantic reasoning, structured knowledge extraction, and psychological modeling.

This project and report were developed with the assistance of ChatGPT (GPT-5, OpenAI, 2025) for prompt design, reasoning, and iterative refinement.
The full conversation documenting this process can be accessed here:
https://chatgpt.com/share/68fb48fe-df70-8005-9ce4-c4c8560ee15f

Data Preprocessing & Normalization

The input data consists of synthetic cover letters written for different roles (UX Designer, Research Scientist, Data Analyst, Software Engineer, Project Manager). These were generated with the assistance of ChatGPT (GPT-5) to simulate realistic professional text containing identifiable entities (person, organization, skills, traits, motivations) and implicit personality cues.

To enable objective evaluation, a small set of gold files containing verified entities, relations, and personality scores were also prepared from the same synthetic letters. These files share the same ID schema (letter_001–letter_005) and are stored separately for use in the evaluation stage.

Process

- Whitespace and Unicode normalization
  Removes line breaks, extra spaces, and non-standard characters to prepare for clean text parsing.
- Casing strategy
  - Proper nouns (e.g., names, organizations) → Title Case
  - Attributes (skills, traits, motivations) → lowercase
  - This standardization aids in entity matching and deduplication.
- Deduplication
  Case-insensitive matching ensures identical entities like “Atelier Interactive” and “atelier interactive” are treated as one.
- JSON structure
  Each text record is structured into a unified schema:


```
{
  "id": "letter_001",
  "name": "Aisha Rahman",
  "target_role": "UX Designer",
  "target_org": "Atelier Interactive",
  "prev_orgs": ["PixelFlow Studio"],
  "skills": ["empathetic", "research-driven", "data-minded"],
  "traits": ["creative", "analytical"],
  "motivations": ["thoughtful design can influence behavior"],
  "preferences": ["collaborative culture"],
  "raw_text": "Dear Creative Director..."
}
```


Justification

Consistent normalization ensures predictable behavior during extraction and prevents noise in evaluation metrics. Retaining text casing for free-text motivations or quotes preserves linguistic nuances critical for personality inference. Cover letters were chosen as the dataset because they combine structured details like roles and organizations with expressive language that reveals personal traits and motivations, making them well suited for modeling personality-linked relationships in the knowledge graph.

LLM-Based Record Extraction (Semantic Parsing)

After preprocessing and normalization, each synthetic cover letter is semantically parsed using Llama 3 (via Ollama) to transform unstructured text into a structured JSON record. The goal is to identify key attributes such as the person's name, target role, target organization, previous organizations, skills, traits, motivations, and preferences. This step ensures that all records follow a consistent schema before entity and relationship extraction.

Process

- Prompt Construction
  Each letter is passed to the model with a predefined JSON schema and clear instructions to extract only factual information from the text. Example schema:
  ```
  {
    "name": "",
    "target_role": "",
    "target_org": "",
    "prev_orgs": [],
    "skills": [],
    "traits": [],
    "motivations": [],
    "preferences": [],
    "raw_text": ""
  }
  ```
- Model Inference
  The normalized text is sent to Llama 3 through the Ollama API for deterministic results. Each record generates one JSON output.
- Validation and Storage
  A parsing routine checks JSON validity, fills missing keys with empty values, and stores all records in cover_letters_normalized.json inside the outputs folder.

Justification

Using an LLM for this stage allows semantic understanding beyond simple keyword detection, enabling the model to recognize implied attributes in the text, such as interpreting "I enjoy collaborating with teams" as a preference for a collaborative environment. This approach produces richer and more accurate structured data compared to rule-based NLP methods, which often fail to capture contextual meaning or nuanced phrasing. By enforcing a fixed output schema and maintaining a low temperature during generation, the workflow ensures that the extracted information remains consistent, factual, and well-structured for the subsequent stages of entity extraction, relationship mapping, and personality inference.

Entity Extraction

This stage identifies all key entities from each structured record. The LLM classifies tokens into predefined categories such as PERSON, ORG, ROLE, SKILL, TRAIT, MOTIVATION, and PREFERENCE. Each entity type represents a distinct node class in the knowledge graph.

Process

- Prompt and Schema Design
  The LLM is given a predefined JSON schema that specifies acceptable entity types and enforces the output format. The prompt instructs the model to extract only entities that are explicitly or implicitly present in the record, avoiding duplication or speculation. Example output:

  ```
  {
   "id": "letter_001",
   "entities": [
    {"text": "Aisha Rahman", "type": "PERSON"},
    {"text": "Atelier Interactive", "type": "ORG"},
    {"text": "UX Designer", "type": "ROLE"},
    {"text": "creative", "type": "TRAIT"}
   ]
  }
  ```

- Model Inference
  Each normalized record from cover_letters_normalized.json is passed to Llama 3 (via Ollama) for entity recognition. The model operates at a low temperature to ensure stable and reproducible outputs.
- Validation and Storage
  Extracted entities are validated, stripped of whitespace, and standardized for casing (Title Case for names and organizations, lowercase for traits and skills). A deduplication function removes repeated entries across similar phrases or letter casing variants. The final list of entities is stored as entities_per_record.json in the outputs folder, along with a flattened version in CSV format for easier evaluation and inspection.

Justification

Entity extraction isolates the essential components that form the nodes of the knowledge graph. Using an LLM allows contextual understanding beyond simple string matching, distinguishing between semantically similar words based on context (e.g., identifying “empathetic” as a trait instead of a skill). The combination of a strict schema and normalization ensures reliable, structured data that supports accurate relationship mapping and personality inference in later stages.

Relationship Extraction

This stage identifies how extracted entities are connected, forming the edges of the knowledge graph. Each record generates structured triples in the format (subject, relation, object), representing how a person relates to roles, organizations, skills, traits, motivations, or preferences.

Process

- Prompt and Schema Design

  The LLM is given a predefined relation schema that defines the allowed relation types, including applies_for, target_org, previously_worked_at, has_skill, has_trait, motivated_by, and prefers_environment. The prompt instructs the model to infer factual relationships based only on information explicitly stated or strongly implied in the record. The output must follow the structured triple format. Sample record:

  ```
  [
    {
      "id": "letter_001",
      "triples": [
        ["Aisha Rahman", "applies_for", "UX Designer"],
        ["Aisha Rahman", "target_org", "Atelier Interactive"],
        ["Aisha Rahman", "previously_worked_at", "PixelFlow Studio"],
        ["Aisha Rahman", "has_skill", "empathetic"],
        ["Aisha Rahman", "has_trait", "collaborative"],
        ["Aisha Rahman", "has_trait", "analytical"],
        ["Aisha Rahman", "motivated_by", "thoughtful design can influence emotion and
  behavior"],
        ["Aisha Rahman", "prefers_environment", "collaborative culture"]
      ]
    }
  ]
  ```

- Model Inference

  Each normalized record is processed by Llama 3 (via Ollama), which analyzes both structured fields and natural language text to generate relational triples. The model runs at a low temperature to ensure deterministic, reproducible outputs.

- Validation and Storage

  Extracted triples are validated to ensure that subjects, relations, and objects are properly formatted and consistent with the entity list. The validated relationships are then stored as relations_per_record.json in the outputs folder for later use in graph construction.

Justification

Relationship extraction defines the semantic structure of the knowledge graph by connecting people, organizations, and personality traits. Using an LLM enables context-aware inference, allowing it to detect relationships that are implied rather than explicitly stated (e.g., “I led the research team at PixelFlow Studio” → previously_worked_at). The controlled relation schema and validation process together ensure accuracy, consistency, and clarity in the extracted graph connections.

Personality Inference

This stage estimates each individual’s personality traits based on their writing style and self-descriptions. The process uses the Big Five (OCEAN) personality model — Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism — to assign normalized scores (0–1) for each dimension.

Process

- Prompt and Schema Design
  The LLM is provided with a predefined schema that defines the five personality dimensions and instructs the model to infer each score using linguistic and behavioral cues in the text. The prompt requests a structured JSON output with values between 0 and 1, where higher numbers indicate stronger expression of that trait. Sample record:

```
{
  "id": "letter_001",
  "big_five": {
    "openness": 0.82,
    "conscientiousness": 0.73,
    "extraversion": 0.61,
    "agreeableness": 0.76,
    "neuroticism": 0.29
  },
  "evidence": "Mentions empathy, imagination, and attention to data-driven design."
}
```

- Model Inference
  Each normalized record from cover_letters_normalized.json is passed to Llama 3 (via Ollama), which analyzes the writing style, tone, and self-descriptive language to infer OCEAN personality scores. The model operates at a low temperature for deterministic and consistent scoring.
- Validation and Storage
  Generated outputs are validated to ensure numeric values fall within the 0–1 range and field names match the expected schema. The final results are saved as personality_inference.json and also exported in CSV format for visualization and later evaluation.

Justification

The Big Five model provides a robust and interpretable psychological framework for representing personality. Storing scores as numeric values enables comparative and relational analysis within the knowledge graph (e.g., linking nodes with high_in openness). Using an LLM allows for deeper semantic interpretation of linguistic cues, capturing subtle expressions of personality that rule-based or keyword methods often overlook.

Graph Construction & Visualization

This stage consolidates all extracted data — entities, relationships, and personality scores — into a unified Personality Knowledge Graph using NetworkX. The goal is to represent how each person's professional attributes and inferred personality traits interact within a structured, interpretable graph.

Process

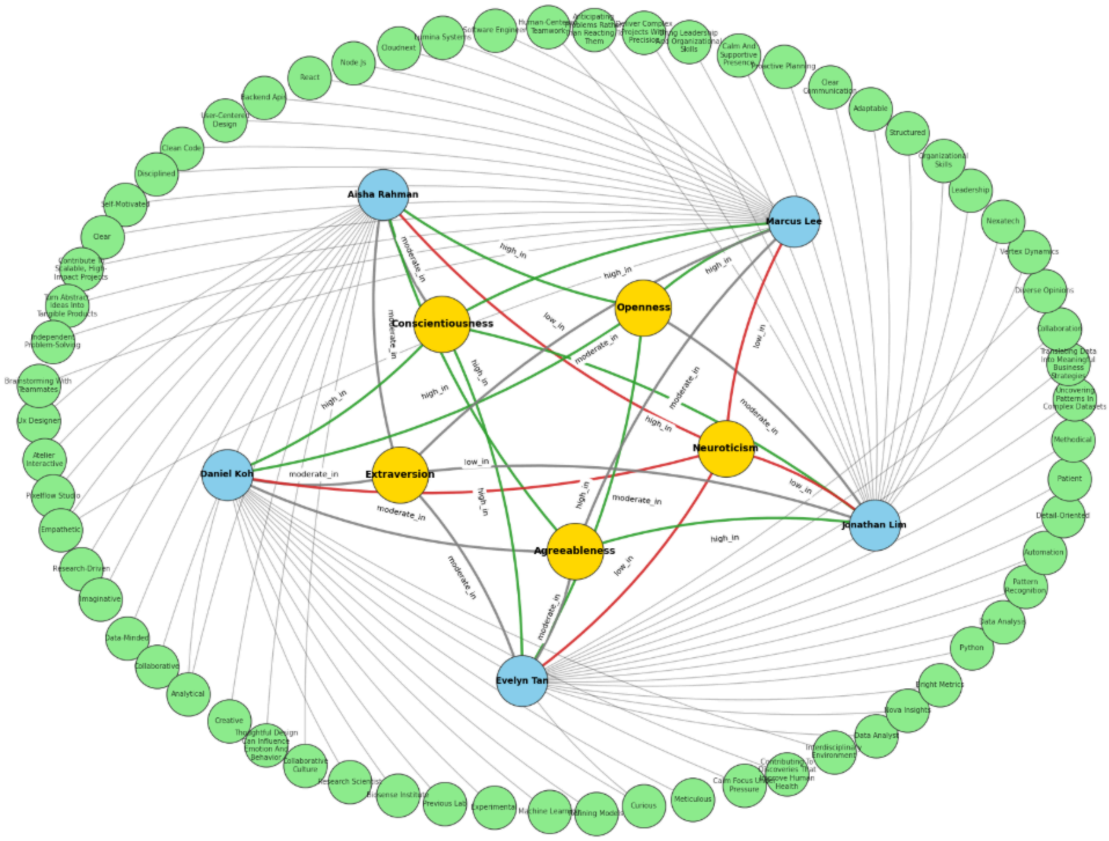
- Data Consolidation

  Validated outputs from the previous stages (entities_per_record.json, relations_per_record.json, and personality_inference.json) are merged by matching record IDs. This ensures that every person node includes both their factual information and corresponding personality dimensions. Missing or duplicate records are handled through key-based validation to preserve data integrity.
- Node and Edge Creation

  Each unique entity (person, organization, skill, role, or trait) becomes a node in the graph. Relations derived from the triples are represented as edges, connecting people to other entities. Personality dimensions (OCEAN) are also treated as nodes, with weighted edges such as high_in, moderate_in, or low_in linking them to individuals based on their inferred scores.
- Graph Layout and Visualization

  A shell layout is used to visually organize the network into three concentric layers:
  - Inner layer: Personality dimensions (OCEAN traits)
  - Middle layer: People
  - Outer layer: Other entities (skills, roles, organizations)

  To reduce overlap, a small random jitter is applied to node positions, and color-coding distinguishes node types (e.g., gold for personality traits, blue for people, green for entities). Labels are wrapped for readability, and edge colors vary by relation type. The finalized graph is plotted using Matplotlib and can be exported as an image for inclusion in reports.



Justification

The layered visualization highlights personality as the conceptual core of the network, reflecting how traits influence professional skills, motivations, and affiliations. Using NetworkX allows scalable analysis of graph properties (e.g., connectivity, clustering, or density) and supports visual interpretability for both analytical and presentation purposes. This design effectively bridges structured data and psychological modeling, providing a clear overview of how personality and professional factors are interconnected.

Evaluation

This stage measures the performance and accuracy of the LLM-based extraction and inference processes by comparing the model outputs against gold files generated by ChatGPT (GPT-5). These gold files serve as reference truth data, containing verified entities, relationships, and personality inferences for each synthetic cover letter.

Process

- Gold File Generation

  The gold-standard reference files were created with the assistance of ChatGPT (GPT-5) based on the same synthetic dataset used in the experiment. These include:
    - gold_entities_per_record.json
    - gold_relations_per_record.json
    - gold_personality_per_record.json

  Each gold file follows the same ID structure (letter_001–letter_005) as the model outputs to ensure consistent alignment. Although generated automatically, these gold files were reviewed for logical consistency and factual correctness relative to the original cover letter content.
- Metric Selection

  The evaluation uses metrics tailored to each task type:
    - Entity and Relation Extraction: Precision, Recall, and F1-score (both micro and per-type).
    - Personality Inference: Pearson correlation (r), Spearman rank correlation (ρ), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE) across all five OCEAN traits.
    - Graph-Level Evaluation: Node and edge count, density, and clustering coefficient to assess the coherence and connectivity of the final graph structure.
- Computation and Reporting

  Model predictions from the outputs/ folder were automatically compared against their respective gold files in the gold/ directory. The evaluation script calculated all metrics and produced a summary report (eval_report.json) detailing performance per record and per metric. Discrepancies between model and gold outputs were logged for analysis and refinement.

Results

Component	Precision	Recall	F1-score	Notes
Entities	0.556	0.556	0.556	TP = 40, FP = 32, FN = 32
Relations	0.515	0.522	0.519	TP = 35, FP = 33, FN = 32

Personality Trait	Pearson (r)	Spearman (ρ)	MAE
Openness	0.18	0.22	0.08
Conscientiousness	0.96	0.73	0.03
Extraversion	-0.27	-0.18	0.08
Agreeableness	0.27	0.18	0.06
Neuroticism	n/a	n/a	n/a

Overall, the entity and relation extraction achieved moderate accuracy, indicating that the LLM effectively recognized and linked key entities but struggled with more nuanced relationships. Personality inference produced high consistency in Conscientiousness but lower correlation in Extraversion and Openness, suggesting variability in interpreting subtle linguistic cues.

Justification

Using gold files generated by ChatGPT ensures standardized and unbiased ground-truth references across all tasks. While human-generated annotations typically offer higher subjectivity, this automated approach provides consistency, reproducibility, and scalability — important for benchmarking LLM-based pipelines.

The chosen metrics capture both extraction accuracy and semantic agreement between model predictions and reference data, offering a balanced view of overall system performance in constructing a coherent and interpretable Personality Knowledge Graph.