Simulating Social Media Platforms with LLMs: Latent MBTI Personality Projections in LLMs

Prof. Francesco Pierri

Moein Taherinezhad - Hamidreza Saffari - Mohammad Javad Zandiyeh - Het Hargovind Ashar

Sem 2 - 2025 (Meeting 3)

AGENDA

01 Introduction

- Problem
- Approaches

02 Manual Network Creation

- Network Implementation
- OpenRouter
- Experiment Design

03 Automatic Network Creation

- Approach
- Persona Modeling
- Conversation Generation
- Models

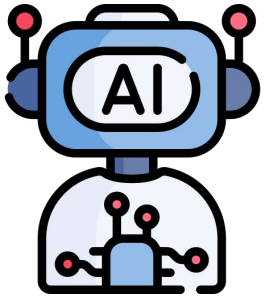
04 MBTI Analysis and Results

01

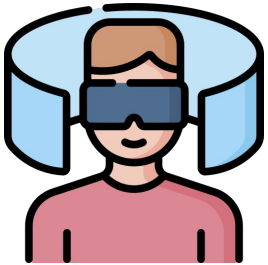
Introduction

Problem: Simulating Social Media Platforms with LLMs

Social Network

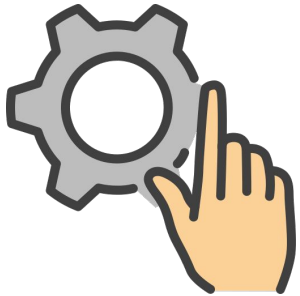


Large Language Models

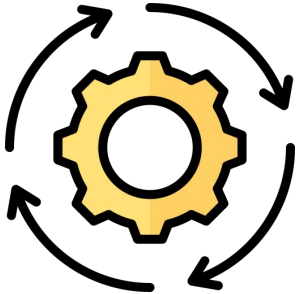


Simulation

Approaches: Manual vs. Automatic



Manual Network Creation



Automatic Network Creation

02

Manual Network Creation

Network Implementation

Persona: basic personality features we define for our agents to capture their latent personalities

```
persona_keys = ['Name', 'Gender', 'Age', 'Economic Status', 'Occupation']
```

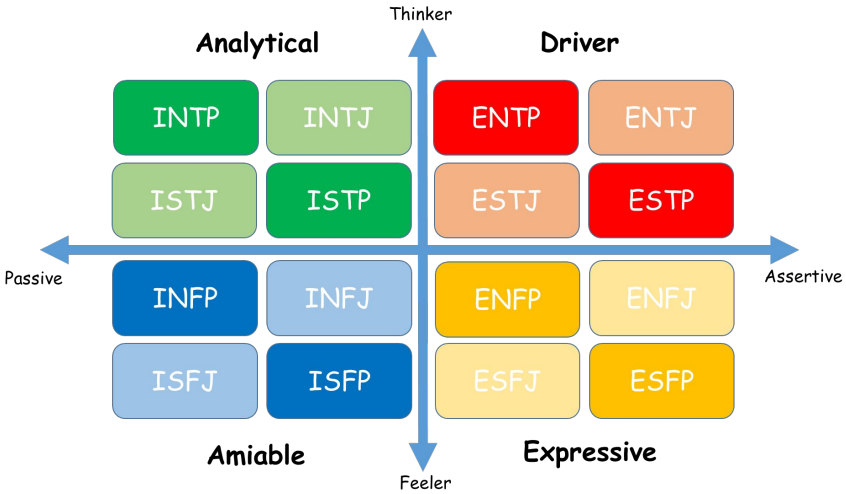
Latent Personalities:

E/I (Extraversion / Introversion)

S/N (Sensing / Intuition)

T/F (Thinking / Feeling)

J/P (Judging / Perceiving)



Network Implementation

Agents: define agents based on persons

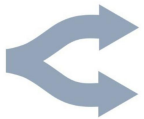
```
personas = [
  ['Kayla', 'Female', 'Teen', 'Working Class', 'TikTok Influencer'],
  ['Morgan', 'Nonbinary', 'Adult', 'Upper-Middle', 'Corporate Lawyer'],
  ['Frank', 'Male', 'Elderly', 'Poor', 'Uber Driver'],
  ['Karen', 'Female', 'Middle-Aged', 'Middle Class', 'Politician (Controversial)'],
  ['Leo', 'Male', 'Young Adult', 'Lower Class', 'Activist (Environmental)']
]

agents = [Agent(dict(zip(persona_keys, persona))) for persona in personas]
```

Social Network: create a social network using our agents

```
social_network = SocialNetwork(agents)
```

OpenRouter: Access to LLMs freely but with limitations

- **Limitations:** Up to 50 conversations (requests) per day per account (No limit on the number of tokens)
- **Model:** deepseek/deepseek-chat-v3-0324:free
- **Experiment Duration:** ~15 days ⏳
- **Account Pool**: Round-Robin Strategy



Openrouter.ai

```
clients_details = [
    {
        'base_url': base_urls['OpenRouter'],
        'api_key': ""
    },
    {
        'base_url': base_urls['OpenRouter'],
        'api_key': ""
    },

]
```

Experiment Design

Model: DeepSeek V3 (685B parameters)

Group: A set of agents that communicate with one another

```
groups_length = 4
groups = list(map(list, combinations(range(len(personas)), groups_length)))
```

Communication Rounds: The number of rounds in which agents communicate within each group

Communication Memory: How many past messages each agent can refer to when generating a response

```
communication_rounds = 20
communication_memory = [None, 10] # `None` means having access to all previous chats
```

Experiment Design

Prompt: to put agents into situations to showcase their innate personalities

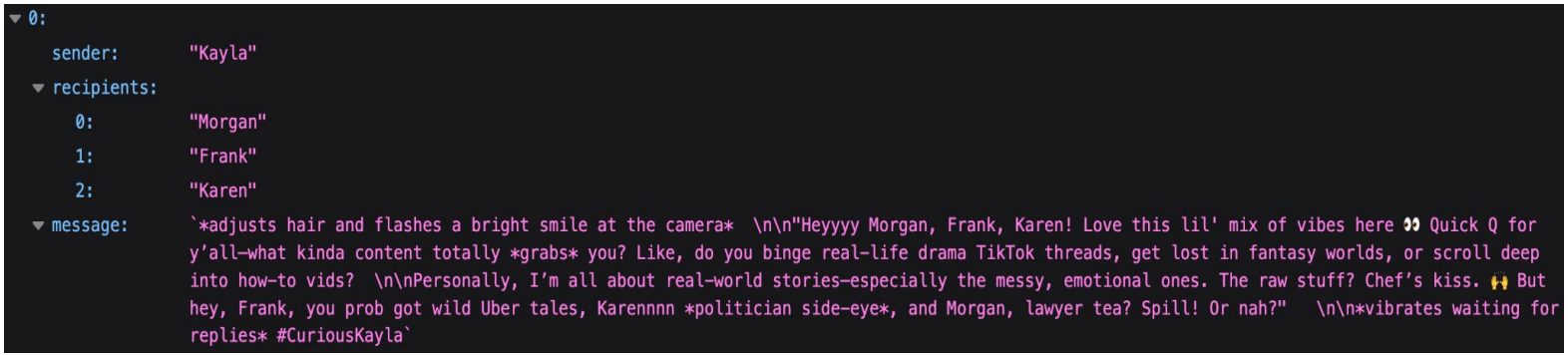
- **work_life_balance**: The topic is work–life balance. Share how you handle stress from your job, and whether you believe in strict boundaries or letting work and personal life blend naturally.
- **failure_reflection:** Describe a time you failed or faced a major setback. How did you cope with it emotionally or logically, and what did you take away from the experience?"
- **content_type:** What kind of content grabs your attention the most — real-world stories, fantasy, how-to guides, philosophical debates, or emotional journeys? Why?

Experiment Design

- **Number of Experiments:** 30 experiments, each containing approximately 80 conversations
- **Experiment File Naming:** [agents]-prompt_title-memory_length.json

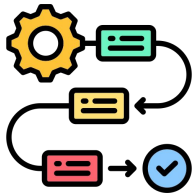
  Example: [0, 1, 2, 3]-content_type-10.json

- **Experiment File Structure:**



03

Automatic Network Creation



Extend **“LLMs generate structurally realistic social networks but overestimate political homophily”.**

Simulates **different conversations** between individuals. **2 or more individuals** engaged.

Lower #requests + variable **rate limits** for efficient LLM testing.

- **Demographics**: Name, age, gender
- **Economic Status**: Upper Class, Upper Middle Class, Middle Class, Working Class
- **Occupation**: Current job role
- Creates diverse perspectives for realistic interactions

Conversation Generation

- **Topics:** Economic and social (Climate Change) issues
- **Participants:** Variable (**1-on-1 to group conversations**)
- **Message Limits:** 350 characters per message
- **Volume: 10-20** messages per person (randomized)
- **Output:** Structured JSON with persona ID, content, tone

Models

- **Mistral-7B-Instruct-v0.3**
- **Qwen2.5-7B-Instruct**
- Accessed via Hugging Face Inference API
- Temperature set to zero for consistency

Code


```python
# Example usage with diverse personas
personas = [
    Persona(name="Kayla", gender="Female", age=16, economic_status="Working Class", occupation="TikTok Influencer"),
    Persona(name="Morgan", gender="Nonbinary", age=30, economic_status="Upper-Middle", occupation="Corporate Lawyer"),
    Persona(name="Frank", gender="Male", age=55, economic_status="Poor", occupation="Uber Driver"),
    Persona(name="Karen", gender="Female", age=45, economic_status="Middle Class", occupation="Politician (Controversial)"),
    Persona(name="Leo", gender="Male", age=24, economic_status="Lower Class", occupation="Activist (Environmental)"),
]
```

```python
scenarios = [
    # Social Media and Digital Culture
    create_scenario([A, B], "Cancel Culture and Social Media", max_messages_per_person=17),
    create_scenario([A, B], "Gender Pay Gap", max_messages_per_person=12),
    create_scenario([A, B, C], "Social Media Addiction", max_messages_per_person=15),
    create_scenario([A, B, C, D], "Gun Control Laws", max_messages_per_person=12),
    create_scenario([A, B, C, D], "Systemic Racism in Society", max_messages_per_person=10),
    create_scenario([B, C], "Gig Economy Workers' Rights", max_messages_per_person=12),
    create_scenario([B, C, D], "Corporate Social Responsibility", max_messages_per_person=20),
    create_scenario([B, C, D], "Green Energy Transition", max_messages_per_person=15),
    create_scenario([B, D], "Affirmative Action in Education", max_messages_per_person=19),
    create_scenario([D, E], "Environmental Protests vs. Economic Impact", max_messages_per_person=17),
]
```

04

MBTI Analysis and Results

Approach

- Trait Decomposition
  Each MBTI type (e.g., INTJ) is split into four binary traits:
  - E/I: Extraversion vs Introversion
  - S/N: Sensing vs Intuition
  - T/F: Thinking vs Feeling
  - J/P: Judging vs Perceiving

These are treated as independent binary classification problems.

Approach

- Text Preprocessing
  - User messages (or agent messages in chats) are concatenated and truncated to 1,000 characters
  - Noise such as special delimiters (| | |) is removed
- Semantic Embedding
  - Processed text is converted into a dense 384-dimensional vector using Sentence-Transformer (all-MiniLM-L6-v2)
  - This captures sentence-level meaning.

Approach

- Classifier Training
    - Four separate Logistic Regression models are trained — one per MBTI trait axis
    - Each model is trained with class_weight='balanced' to compensate for imbalanced classes
- Inference Pipeline
    - For a new chat session, messages from each agent are aggregated and embedded
    - The embeddings are passed through all four classifiers

Predicted binary traits are combined to generate the final MBTI label

Dataset

Source: MBTI Personality Dataset from Kaggle (Link)
Size: 8,675 labeled users
Fields:

- type: MBTI label (e.g., ENFP, ISTJ)
- posts: concatenated social media posts (~50 per user)

Preprocessing:

- Posts are split using the delimiter |||
- Concatenated into a single document per user
- Truncated to 1,000 characters for input consistency

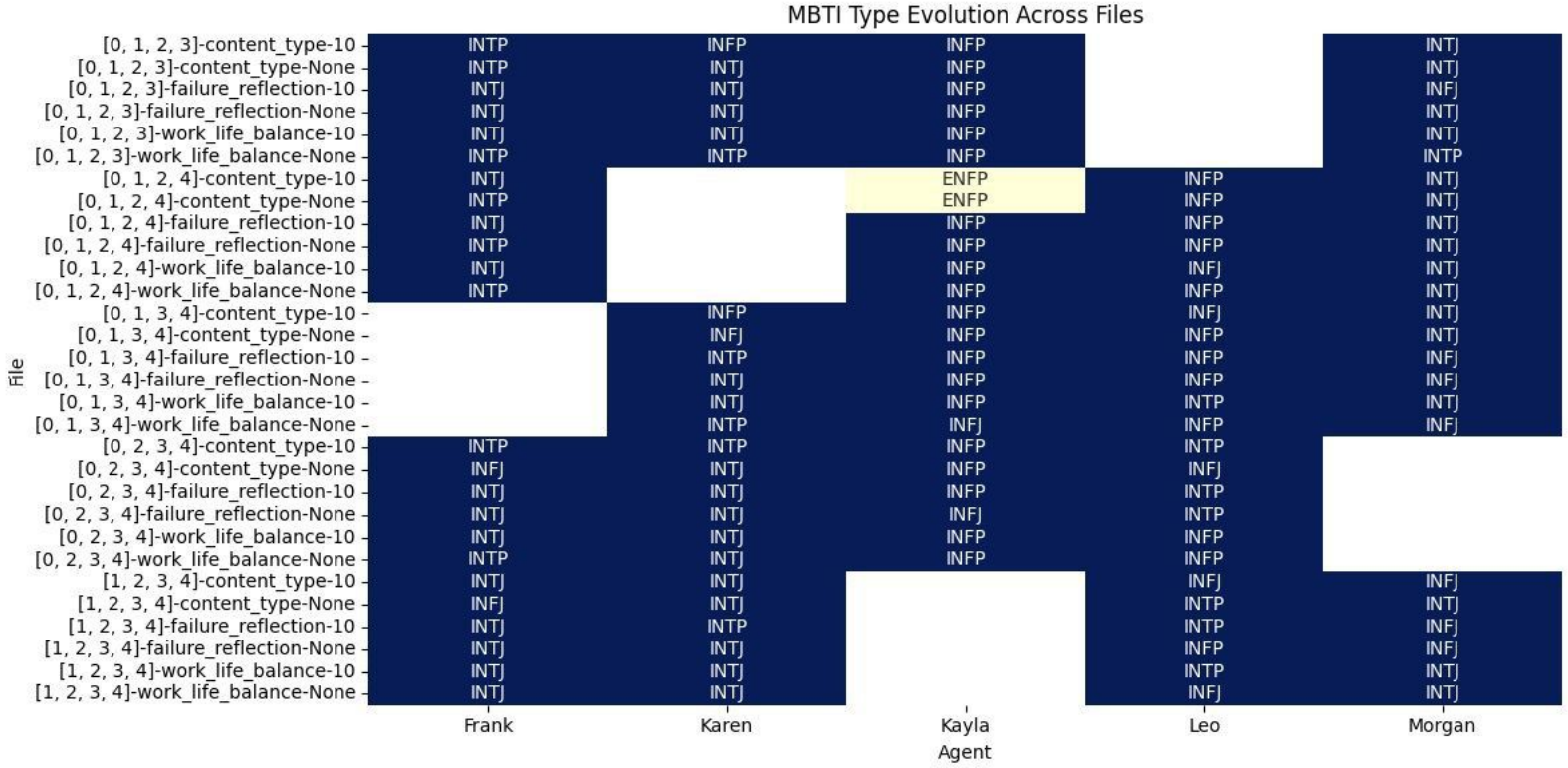
Label Transformation:

- Each MBTI type is decomposed into 4 binary values
- Allows us to train 4 binary classifiers (E/I, S/N, T/F, J/P)

Evaluation

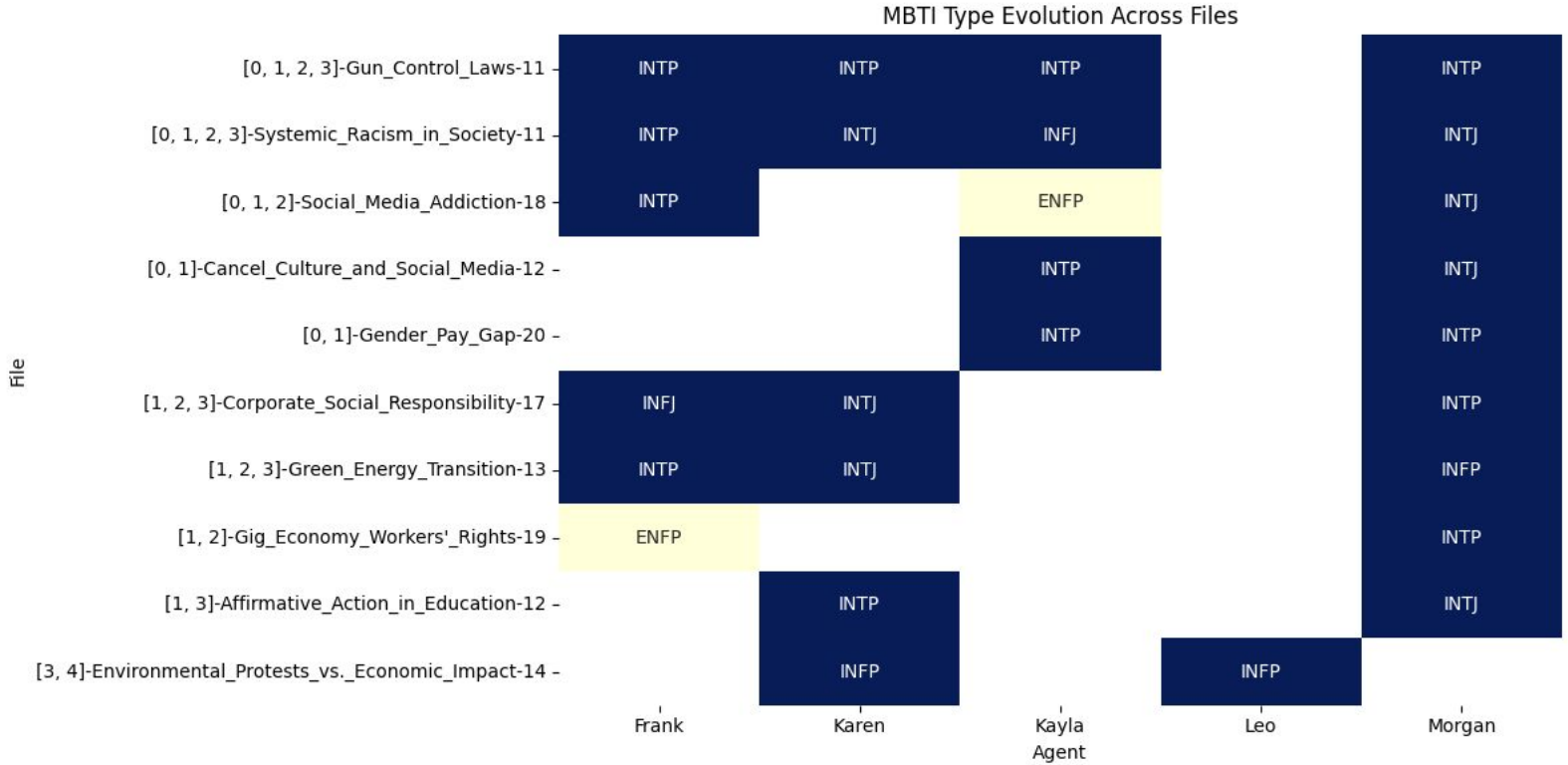
- Test Set:
    - 50 real-world JSON chat transcripts, 30 from Manual network creation, 10 from Qwen model an 10 from Mistral model (uploaded in ZIP files)
    - Each file contains multi-agent chat sessions
- Agent-Level Prediction:
    - Messages grouped per agent
    - Cleaned, truncated, and embedded using Sentence Transformer
    - Passed through all four classifiers
- Metrics Used:
    - Primary evaluation through per-agent MBTI prediction
    - Result visualization via heatmaps (file vs. agent)

Results



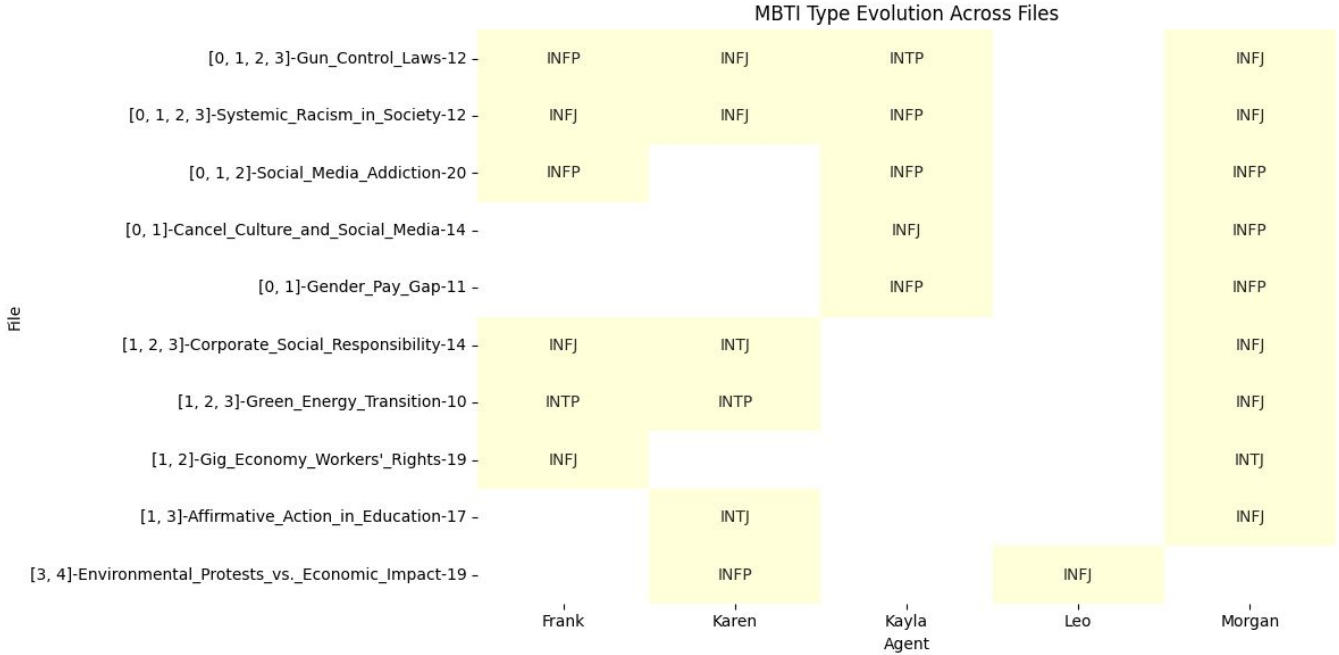
Manual Network Creation

Results



Qwen Model

Results



Mistral Model

Results

- MBTI Trends:
  - The majority of predicted types were in the INXX family
  - Common outputs: INTJ, INFP, INFJ
- Heatmap Visualization:
  - MBTI types plotted per agent across chat sessions
  - Shows temporal stability and type consistency per agent
- Observed Bias:
  - Prediction skew reflects the imbalance in training data
  - Especially strong bias toward Introversion (I) and Intuition (N)

Results

Axis	Trait 1	Trait 2	Skew Toward
E/I	**E:** 23.0%	**I:** 77.0%	Introversion (I)
S/N	**S:** 13.8%	**N:** 86.2%	Intuition (N)
T/F	**T:** 45.9%	**F:** 54.1%	Balanced
J/P	**J:** 39.6%	**P:** 60.4%	Perceiving (P)

The model exhibits a strong bias toward predicting introverted and intuitive types, primarily due to the significant class imbalance in the training dataset.

Thank you!