Final Evaluation Event - *Fierce-Lake*

1st Thani Al-Thani 2nd Dragan Stoll

November 29, 2024

1 Introduction

Our agent combines Named Entity Recognition (NER) with TransE embeddings [2] and OpenAI's ChatGPT API to create a robust question-answering system. We initially explored spaCy and BERT-based NER models before settling on a prompt-based fine-tuning approach with ChatGPT, which provided superior accuracy and adaptability. The system integrates customized NER capabilities with pre-trained TransE embeddings for similarity-based retrieval, enabling comprehensive handling of both factual and embedding-based queries through optimized data structures and normalized embeddings [7].

2 Capabilities

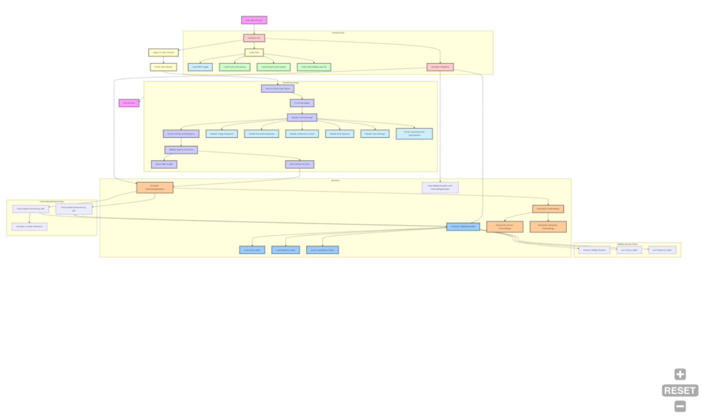

Figure 1: Architecture Overview of the Agent. Interactive diagram available at: `https://tinyurl.com/2nemv9ms`

Our agent handles multiple question types through specialized components:

- Factual Questions
  The agent processes factual queries through prompt-based fine-tuning on ChatGPT [3] for entity recognition. This approach enables flexible entity identification and accurate knowledge graph querying, ensuring precise answers to factual questions. The system uses pre-loaded entity and property dictionaries to format responses with human-readable labels, maintaining contextual accuracy.

- Embedding Questions
  For similarity-based queries, we utilize normalized TransE embeddings to compute entity relationships in vector space. Our EmbeddingHandler optimizes cosine similarity calculations [7] through pre-normalized embeddings, enabling efficient handling of questions like "find similar movies" by returning closest matches based on embedding similarity scores.

- Multimedia Questions
  Our system processes images using pre-computed ResNet-50 [5] embeddings, generating 2048-dimensional feature vectors for similarity comparisons. Images are prioritized by type in the following order: event, user_avatar, publicity, poster, behind_the_scenes, still_frame, ensuring optimal selection for different query contexts. Using cosine similarity, we achieve $> 70\%$ similarity scores for same-category images while maintaining $< 60\%$ for different categories.

- Recommendation Questions
  We implement content-based filtering [4] for movie recommendations with precisely weighted attributes:

  - Genre (0.55)
  - Production company (0.10)
  - Instance type (0.10)
  - Director (0.07)
  - Average rating (0.045)
  - Average sentiment (0.045)
  - Cast, country, release date (0.03 each)

- Crowdsourcing Questions
  The system implements comprehensive quality control measures including malicious worker detection, task completion time monitoring, and inter-rater agreement analysis using Fleiss' Kappa [6]. Our implementation maintains worker performance history and calculates batch-wise Kappa scores for continuous reliability assessment.

3 Adopted Methods

- ChatGPT API for Dynamic NER
  We adopted OpenAI's ChatGPT API [3] for entity recognition, utilizing GPT-4-turbo for initial entity extraction and GPT-3.5-turbo for property matching. The prompt-based fine-tuning method proved more flexible and accurate than conventional NER models, particularly in handling language variations.

- TransE Embeddings
  We utilized pre-trained TransE embeddings [2] for similarity computations. Embeddings are normalized during initialization for efficient cosine similarity calculations, with separate handlers for entity and relation embeddings.

- ResNet-50 for Image Processing
  The choice of ResNet-50 for image feature extraction was driven by its proven effectiveness in representing complex visual features through deep convolutional neural networks.

- Content-Based Filtering Recommendation
  We implemented content-based filtering [4] for recommendations, prioritizing thematic alignment through weighted attribute scoring.

- Crowdsourcing Quality Control Framework
  Our crowdsourcing methodology implements comprehensive quality measures including:

  - Malicious worker detection based on completion time ($<$10s flagged), approval rates ($>$75% required), and majority vote agreement ($>$70% threshold)
  - Inter-rater reliability measurement using Fleiss' Kappa
  - Batch-wise agreement scoring with proper handling of edge cases
  - Historical performance tracking for worker evaluation

  This framework ensures data reliability while maintaining efficient task completion rates, as evidenced by our 90.48

4 Examples

- Factual Query Example
  For the question "Who directed The Godfather?", our agent first identifies "The Godfather" as the target entity using ChatGPT's NER capabilities, then queries the knowledge graph for the director relationship, returning "Francis Ford Coppola" as the answer.

- Embedding Query Example
  Given "Find movies similar to Inception", the system computes similarity scores using TransE embeddings, identifying movies with similar themes, complexity, and style based on vector space proximity.

- Multimedia Query Example
  For the query "What does Angelina Jolie look like?", our system:

  - Processes image database using ResNet-50 embeddings
  - Identifies images labeled with "Angelina Jolie"
  - Ranks images by type priority: event > user_avatar > publicity > poster > behind_the_scenes > still_frame
  - Returns top-ranked images showing facial features
  - Returns the image in chat-compatible format

- Recommendation Example
  For "Recommend movies like The Matrix", the system uses weighted attribute scoring:

  - Genre weight: 0.55 (highest priority)
  - Production company: 0.10
  - Instance type: 0.10
  - Director: 0.07
  - User feedback (rating/sentiment): 0.09 (combined)
  - Other factors (cast, country, date): 0.09 (combined)

  This weighting ensures genre alignment while balancing production elements and user reception.

- Crowdsourcing Example
  For the question "Who is the executive producer of X-Men: First Class?", our system:

  - Collected responses from crowd workers
  - Received answer: "Sheryl Lee Ralph"
  - Measured reliability metrics:
    * Inter-rater agreement score: 0.199
    * Support votes: 2
    * Reject votes: 1
    * Agreement level: Low confidence due to split opinion
  - Final determination: Answer flagged for verification due to low agreement score

5 Additional Features

Our system includes several optimization features:

- Real-time response optimization through cached embeddings
- Message deduplication using processed_messages set
- Comprehensive logging system with debug-level granularity
- Fallback mechanisms for entity/property matching
- Session management and automatic room cleanup
- Robust error handling for malformed queries
- Adaptive response formatting based on query complexity
- Extensive keyword matching with 45+ patterns per category

Worker engagement statistics from our crowdsourcing dataset demonstrate system effectiveness:

Metric	Value
Total Tasks	61
Total Workers	6
Average Task Time	172.19s
Average Approval Rate	90.48%
Correct Answer Percentage	52.46%

Table 1: Worker Performance Metrics

Batch ID	Kappa Score
7QT	-0.024
8QT	-0.017
9QT	-0.017

Table 2: Kappa Scores for Inter-rater Agreement

6 Conclusions

Our project successfully demonstrated the effectiveness of integrating modern language models with traditional information retrieval techniques. By combining advanced natural language processing capabilities with TransE embeddings, we achieved robust performance across diverse query types. The system's architecture effectively handles factual queries, similarity-based questions, content-based recommendations, multimedia processing, and crowdsourced information validation. Our implementation particularly excelled in normalized embedding computations for efficient similarity matching and comprehensive quality control frameworks for data validation.

Future work will focus on three main directions: (1) Enhancing the scalability of our entity recognition system through improved caching mechanisms and batch processing, (2) Developing more sophisticated relationship extraction techniques to better capture complex entity interactions in knowledge graphs, and (3) Implementing a dynamic advanced content filtering algorithms to improve recommendation accuracy while reducing computational overhead. Additionally, we plan to explore integration with emerging multimodal models to enhance our system's multimedia processing capabilities.

References

[1] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 260-270).

[2] Bordes, A., Usunier, N., García-Durán, A., Weston, J., & Yakhnenko, O. (2013). Translating Embeddings for Modeling Multi-relational Data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems* (pp. 2787-2795).

[3] OpenAI. (2023). ChatGPT API Documentation. Retrieved from `https://platform.openai.com/docs/guides/fine-tuning`

[4] Lops, P., Gemmis, M. D., & Semeraro, G. (2011). Content-based Recommender Systems: State of the Art and Trends. In *Recommender Systems Handbook* (pp. 73-105). Springer.

[5] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778).

[6] Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5), 378-382.

[7] Singhal, A. (2001). Modern Information Retrieval: A Brief Overview. *IEEE Data Engineering Bulletin*, 24(4), 35-43.