Final Evaluation - GreenCracklingFire

Igli Kristo (*igli.kristo@uzh.ch*)
Tim Fries (*timaurin.fries@uzh.ch*)

August 18, 2026

1 Introduction

This report describes our question answering agent, which interprets natural language questions about films and retrieves information from a Wikidata-based RDF graph. The system supports multiple types of queries: (1) factual questions answered via SPARQL, (2) embedding-based questions resolved through TransE vector reasoning, (3) recommendation queries based on a hybrid method combining graph-derived similarity and collaborative filtering, and (4) multimedia questions requiring the bot to retrieve and display images such as actor photos or movie posters. In addition to these core capabilities, the agent integrates robust entity and relation extraction, a question-type classifier that selects the appropriate answering strategy, and a fallback mechanism using large language models for entity disambiguation. All components—including extraction modules, SPARQL processing, embedding inference, recommendation logic, multimedia retrieval, LLM fallback, and conversation management are orchestrated by a central BotAgent that coordinates the entire reasoning pipeline. The LLM fallback component, was implemented using the Ollama inference engine. It is used for entity disambiguation, classification support, and graceful degradation when traditional extraction methods fail.

2 Capabilities - Multimedia

The agent extracts named entities using a combination of regex patterns and spaCy. Each extracted candidate is mapped to a Wikidata entity through string matching and label lookup. Relation extraction is performed using keyword dictionaries, with an embedding-based fallback for ambiguous formulations.

The main extension of this evaluation event is support for multimedia questions requiring direct image output. The agent detects such queries using a dedicated classifier that recognizes keywords like "show me", "picture of", "look like", or "poster".

Once a multimedia request is identified, the EntityExtractor resolves the referenced film or actor. To link this entity to an image, the agent retrieves its IMDB ID via property P345 from the RDF graph. The system then loads

a curated images.json index that maps IMDB identifiers to relative image file paths.

If a matching IMDB ID is found, the agent returns the image using the platform-specific format

`image:<relative-path>`

which causes the evaluation environment to render the image inline. This mechanism ensures that questions such as "Can you show me the poster of Forrest Gump?" or "Does Denzel Washington look like this?" produce actual images rather than plain hyperlinks or text descriptions.

3 Adopted Methods

The system relies on spaCy for entity recognition, rdflib for RDF access and SPARQL execution, and NumPy for vector operations. TransE embeddings are used for approximate reasoning when explicit facts are missing. Cosine similarity serves as the primary comparison metric across embeddings and feature vectors. The multimedia component uses IMDB identifiers, a structured image index, and local file access to retrieve and render images. For robustness, the agent uses an LLM fallback based on the Ollama framework. The fallback model is invoked when entity extraction fails, when a question cannot be classified reliably, or when both SPARQL and embedding-based methods return no answer.

4 Examples

For multimedia questions, the bot identifies the entity, resolves its IMDB ID, finds the corresponding image entry in `images.json`, and responds with the correct inline rendering token. For example, the question "Show me the poster of Forrest Gump" results in a reply of the form:

`image:0077/4bZr216gYuanA1urxFC561FmsIp`

which the platform displays directly as the movie poster.

5 Conclusions

The agent combines factual querying, embedding-based inference, and newly added multimedia retrieval capabilities. The enhanced extraction pipeline and the integration of IMDB-linked image indexing allow the system to correctly answer not only knowledge-oriented questions but also visual ones. This modular design ensures robustness and extensibility for future evaluation events.