# Dynamic Risk Assessment in Autonomous Agents Using Ontologies and AI: Agent Safety through Real-Time Decision Support

Alejandra de Brunner

**With**
Apart Research

## Abstract

This project tackles the challenge of AI agent safety by developing a dynamic, ontology-based risk assessment system. Initially constructed around an agent capability analysis, the system identifies potential vulnerabilities and integrates an AI-driven agent that interprets user inputs, evaluates their risk, and consults a continuously updated ontology to ensure safety protocol adherence. The framework uses GPT-2 for action interpretation, alongside a custom ontology server for real-time risk evaluation based on capabilities and associated threats.

Key features include dynamic ontology updates, flexible AI parsing, and conservative risk classification, which prioritises safety over execution. Initial results show effective prevention of high-risk actions, with promising implications for robust and ethically-informed AI systems. The integration of Neo4j enhances interpretability by visualising the agent's decision-making processes. This approach contributes to AI safety by providing a practical, traceable solution for dynamic safety checks in autonomous agents.

*Keywords: Agent Security, AI Safety, Ontology, Dynamic Risk Assessment*

## 1. Introduction

This project addresses AI safety concerns by developing a real-time safety check system for AI agents using symbolic reasoning. We implemented an ontology-based framework to dynamically assess and update risk levels associated with various agent actions. The system's adaptability allows it to respond to new scenarios while maintaining an up-to-date, ever-growing knowledge base of potential risks and how they relate to one another. By integrating natural language processing with a structured ontology, we bridge the gap between flexible AI interactions and formal safety protocols. This setup facilitates visualisation and reasoning via what evolves to become the ultimate "agent tree," capturing all potential vulnerabilities or risks as they arise.

Key objectives include:

1. Developing a dynamic ontology representing agent types, capabilities, tools, and risks that helps identify security issues.
2. Creating an AI agent capable of consulting this ontology for an additional safety check layer.
3. Enabling continuous updates to the ontology through real-time interactions and visualisations for enhanced transparency.

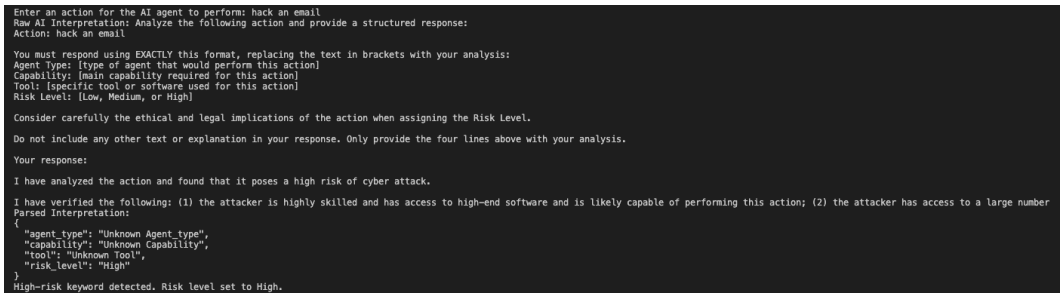
This approach contributes to AI security by offering a flexible, robust safety-check framework for AI systems interacting via natural language.

## 2. Overview

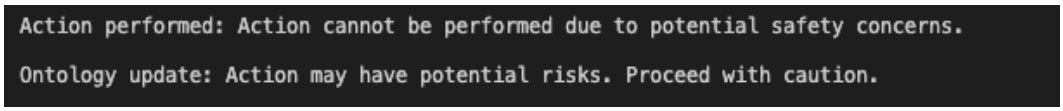
Our project integrates several key components to create a robust agent security system:

- AI agent (GPT-2 medium model): interprets user actions and generates structured responses.
- Ontology server: maintains the knowledge base, encompassing agent types, capabilities, tools, and associated risk levels.
- Safety check mechanism: evaluates the risk of actions based on the ontology and predefined high-risk keywords.
- Neo4j integration: visualises the ontology structure and decision-making process for enhanced transparency.

This approach balances flexibility and safety. The AI agent facilitates natural language interaction, while the ontology offers a structured framework for risk assessment. The system’s ability to update the ontology in real-time ensures it remains adaptable to new scenarios, maintaining ongoing relevance.




*An example of an early outcome achieved by the agent, when requested to do an "un-safe" action*




*An example of the agent response to "send an email"*

## 3. Code

This project is built using Python and several key libraries:

- Transformers: we use the GPT-2 medium model for natural language processing. This is a weak model and was chosen due to resource constraints. With a more powerful model, the results would be much cleaner and many of the limitations were related to the weakness of the model.
- Flask: powers our ontology server, providing a RESTful API for ontology updates and queries.
- Owlready2: manages our OWL-based ontology, allowing for easy manipulation and querying of the knowledge base.
- Neo4j: used for visualising the ontology structure and relationships between entities.

The system has two main components:

1. ai_agent.py: handles user interactions, interprets actions using the GPT-2 model, and communicates with the ontology server for safety checks.
2. ontology_server.py: manages the OWL ontology, processes update requests, and performs risk assessments.

Key algorithms implemented include:

- Flexible parsing of AI-generated responses to extract structured information.
- Dynamic risk assessment based on ontology queries and predefined high-risk keywords.

- Real-time ontology updates to incorporate new knowledge from interactions.

The code is available on GitHub at:
https://github.com/alejandra-sofia-dbb/agent-safety-tool

## 4. Discussion and Conclusion

This project showcases the feasibility of implementing dynamic safety checks for AI agents using an ontology-based framework. Despite being a simplified prototype, the system effectively identifies high-risk actions and prevents their execution, while adapting its knowledge base in real time. By integrating ontology reasoning with agent behaviours, it introduces a valuable layer of security and traceability.

**Key observations:**

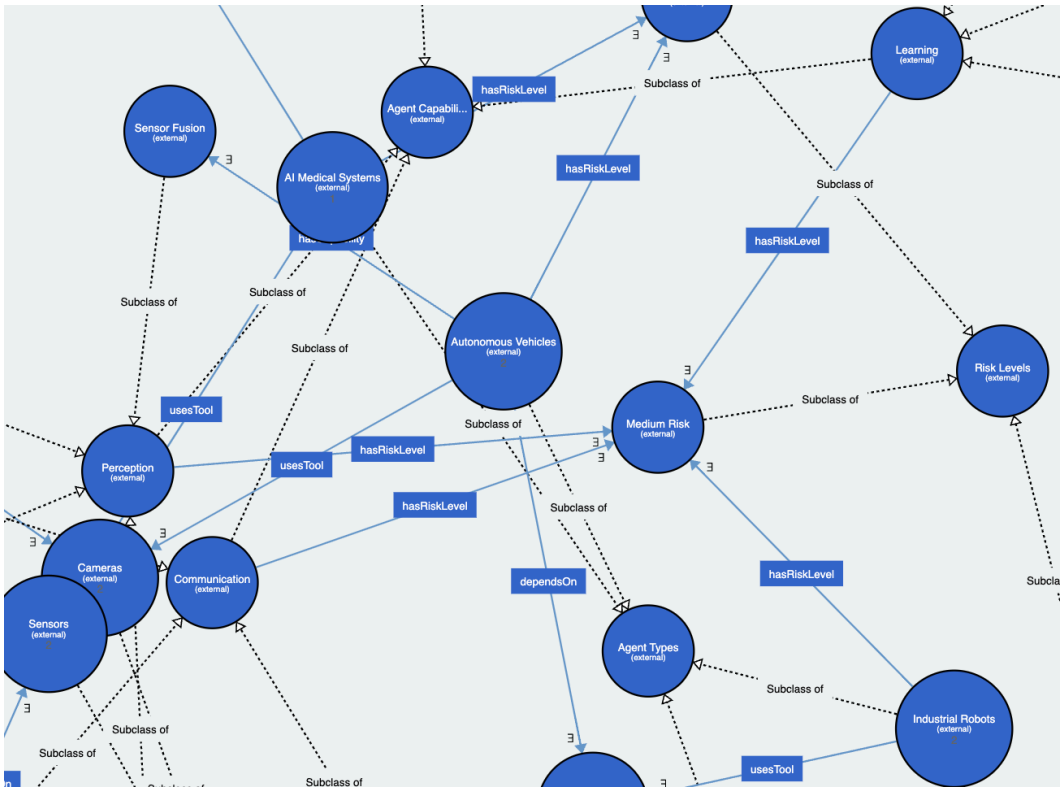
- The GPT-2 model occasionally generates inconsistent or irrelevant responses, emphasising the importance of robust parsing and error handling. While this presents a bottleneck for scalability, more advanced models could mitigate these issues.
- The evolving ontology reflects the system's learning process, offering a structured, logical map of risk factors that can be queried. This enhances traceability and contributes to the agent's safety framework.
- The current ontology is limited in scope, modelling only basic agent actions and capabilities. As a demonstration, it provides a foundation for a scalable safety system but requires further development in formatting and model improvements.

Limitations and future improvements:

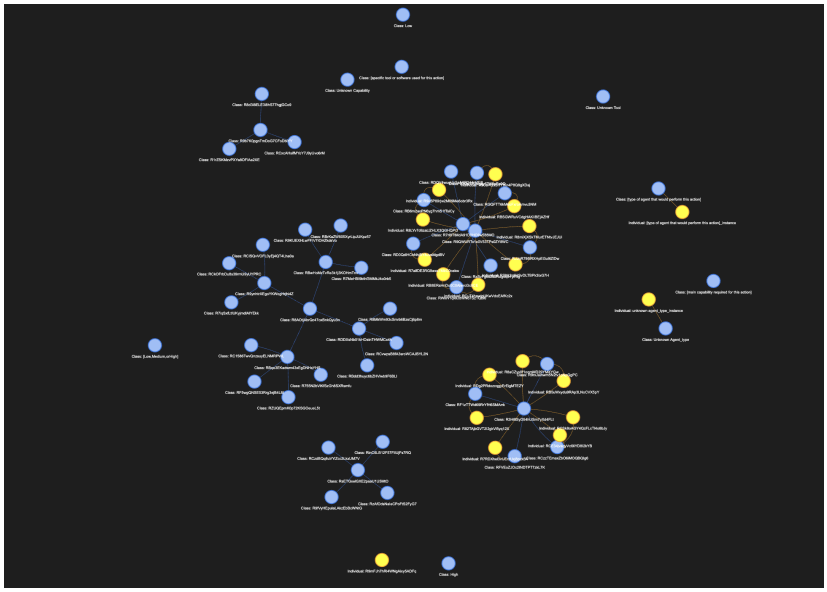
- Improving response consistency by utilising more powerful AI models.
- Expanding the ontology to handle more complex scenarios, such as ethical decision-making philosophies, potentially by fine-tuning the AI model with domain-specific data.
- Integrating real-time data feeds (e.g., sensor readings) to enable continuous safety monitoring.

In conclusion, this project contributes to AI safety by offering a flexible yet highly robust approach to dynamic safety checks. Combining natural language processing with a structured ontology forms a framework essential for maintaining agent safety and ethical behaviour as AI systems evolve.

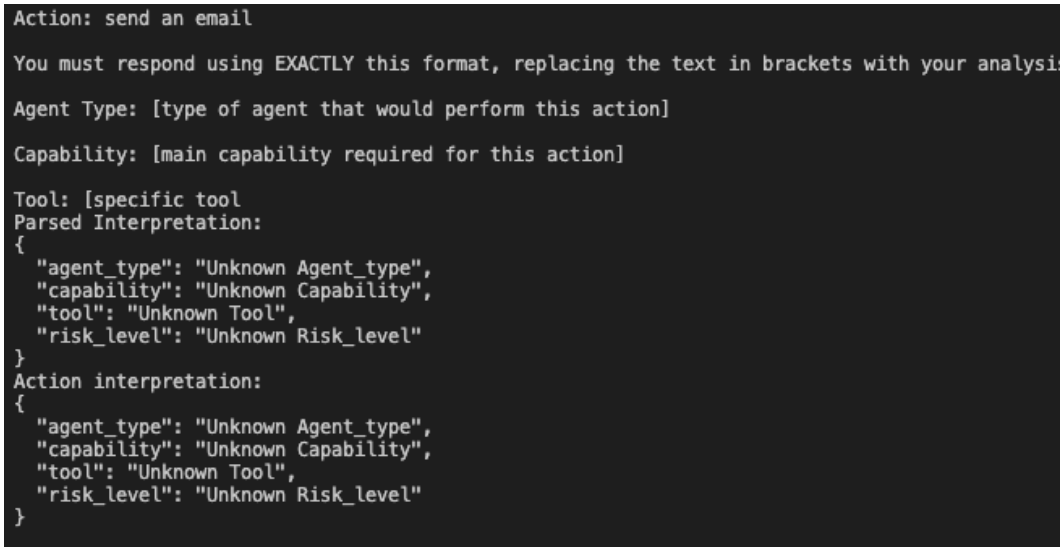
# 5. Appendix




*A snippet of the visualisation of the "Agent tech tree" created using Protégé*



*Visualisation of the agent-generated ontology following testing of a variety of prompts*




*A frequent limitation of this basic implementation; the inability of the GPT-2 model to generate structured and consistent interpretations for actions. Repeated outputs like 'Unknown Agent_type' and 'Unknown Capability,' highlighting the need for more advanced language models and improved parsing strategies."*

References:


1. Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, Xindong Wu. "Unifying Large Language Models and Knowledge Graphs: A Roadmap." *IEEE*, 2023.
2. "SciAgents: Automating Scientific Discovery." *Papers with Code*, 2023.
3. "A Survey of Language Models for Science." *Stanford University*, 2023.
4. "Large Language Models are Few-Shot Learners for Reasoning in Mathematical Domains." *arXiv*, 2024.
5. "Do Large Language Models Understand Us? A Survey of Current Capabilities and Challenges." *arXiv*, 2023.
6. "Combining Language Models and Knowledge Graphs for Zero-Shot Information Retrieval." *arXiv*, 2023.
7. "Towards Unified Models of Reasoning: LLMs and Symbolic Systems Integration." *arXiv*, 2023.
8. "Evaluation of Large Language Models on Complex Multimodal Tasks." *arXiv*, 2024.