Research Paper: PROMETHEUS 1.0 - An Experimental Framework for AGI

Abstract

PROMETHEUS 1.0 is an experimental Artificial General Intelligence (AGI) framework designed with modular components to simulate human-like cognitive processes. The architecture spans multiple cognitive domains including memory, learning, perception, reasoning, planning, interaction, and meta-cognition. This framework is untested, untrained, and undebugged, representing an exploratory prototype for AGI development rather than a production-ready model.

1. Introduction

The development of AGI remains an elusive goal in the field of artificial intelligence. PROMETHEUS 1.0 seeks to implement core cognitive functions based on contemporary theories like Global Workspace Theory (GWT) and meta-learning strategies. Despite its conceptual ambition, this framework has not undergone benchmarking, training, or debugging.

The framework's design emphasizes modularity and flexibility, allowing future researchers to experiment with different cognitive architectures. It aims to replicate core aspects of human-like intelligence, including abstract reasoning, memory consolidation, social cognition, and causal reasoning.

2. Architecture Overview

The system is divided into several domains:

- **Core**: Global Controller, Adaptive Attention, Self-Assessment.
- **Interaction**: Interaction Manager, Social Cognition, Theory of Mind.
- **Knowledge**: Knowledge Graph, Ontology Builder, Cross-Domain Transfer.
- **Learning**: Meta-Learning, Dynamic Architecture Evolution.
- **Memory**: Episodic, Semantic, Unsupervised Memory.
- **Meta**: World Model, Meta-Narrative, Causal Experimenter.
- **Perception**: Multimodal Encoding, Grounded Language Interface.
- **Reasoning**: Abstract Reasoning, Neural-Symbolic Bridge.
- **Planning**: Universal Planning, Task Scheduling.
- **Cognition**: Cognitive Control, Goal Autonomy.
- **Reinforcement Learning**: RL-based agents for dynamic environments.

Each domain contains specialized modules that collaborate through a global workspace mechanism to achieve adaptive, context-aware behavior.

3. Module Descriptions

3.1 Core Components

- **Global Controller**: Implements GWT to mediate information flow across modules. It monitors task progression, memory access, and cognitive resources.
- **Adaptive Attention**: Dynamically adjusts focus between sensory streams based on task demands and novelty detection.
- **Self-Assessment Module**: Periodically evaluates task performance using historical performance metrics and triggers adaptive changes when needed.

3.2 Interaction and Social Cognition

- **Interaction Manager**: Handles communication between agents and external systems, supporting multi-agent interactions.
- **Theory of Mind Module**: Simulates beliefs, intentions, and emotional states of other agents to facilitate social understanding.
- **Social Cognition Module**: Interprets social cues and generates appropriate responses during interactions.

3.3 Knowledge Representation

- **Knowledge Graph**: Stores interconnected concepts with dynamic relationships, supporting causal and semantic queries.
- **Ontology Builder**: Uses pretrained transformers to abstract observations into structured ontological entities.
- **Cross-Domain Transfer**: Identifies transferable patterns across domains using neural embeddings and clustering techniques.

3.4 Learning Mechanisms

- **Advanced Meta-Learner**: Implements MAML-like meta-learning to adapt quickly to new tasks with minimal data.
- **Dynamic Architecture Evolver**: Modifies model architecture to adapt to new tasks using evolutionary algorithms.
- **Learning Strategy Optimizer**: Monitors learning performance and dynamically adjusts hyperparameters for improved efficiency.

3.5 Memory Systems

- **Episodic Memory**: Stores event-based experiences with temporal context.
- **Semantic Memory**: Stores fact-based knowledge derived from interactions and learning processes.
- **Unsupervised Memory**: Uses an autoencoder to compress memory representations while retaining useful information.
- **Memory Manager**: Coordinates interactions between different memory types and handles memory consolidation.

3.6 Meta-Cognition

- **Advanced World Model**: Simulates potential outcomes based on learned dynamics, facilitating counterfactual reasoning.
- **Meta-Narrative Module**: Generates human-readable explanations to justify decisions and reflect on failures.
- **Causal Experimenter**: Conducts Pearl-style causal interventions to refine causal models through interventions and observations.

3.7 Perception

- **Multimodal Encoder**: Combines sensor data from different modalities to create unified representations.
- **Grounded Language Interface**: Aligns textual inputs with sensory data, supporting language-driven task execution.
- **Affective Reasoning Engine**: Analyzes sensor data for affective cues, enabling emotionally-aware interactions.

3.8 Reasoning

- **Abstract Reasoning Engine**: Solves complex abstract tasks by detecting underlying patterns and relationships.
- **Neural-Symbolic Bridge**: Bridges deep learning embeddings with symbolic knowledge using differentiable logic layers.
- **Adaptive Causal Inference**: Dynamically refines causal relationships based on new observations.

3.9 Planning

- **Universal Planner**: Creates hierarchical plans using symbolic and RL strategies, considering task dependencies.
- **Task Scheduler**: Dynamically adjusts task execution order based on priority, urgency, and environmental conditions.
- **Hierarchical World Builder**: Constructs multi-scale world models for abstract and concrete task planning.

3.10 Cognitive Modules

- **Cognitive Control**: Manages goals and resources, optimizing the allocation of computational resources.
- **Goal Autonomy Engine**: Proposes intrinsic goals based on novelty detection and curiosity-driven exploration.
- **Intrinsic Curiosity Module**: Quantifies novelty by comparing expected and actual outcomes.
- **Meta-Curiosity Module**: Assesses novelty across different domains to prioritize exploratory behavior.

3.11 Reinforcement Learning

- **Generic RL Agent**: Uses Stable-Baselines3 to train task-specific skills in simulated environments.
- **RL Module**: Interfaces with planning and memory modules to integrate reinforcement signals into decision-making processes.

4. System Dynamics and Workflow

The execution workflow of PROMETHEUS 1.0 involves several stages:

1. **Perception and Input Processing**: Sensor data is processed by the sensory input manager and multimodal encoder.
2. **Knowledge Integration**: Processed inputs are linked to existing concepts in the knowledge graph.
3. **Task Inference and Goal Generation**: Cognitive modules propose and prioritize goals.
4. **Planning and Strategy Formulation**: The universal planner generates a task plan.
5. **Execution and Monitoring**: Tasks are executed with ongoing performance monitoring.
6. **Learning and Adaptation**: Learning modules update models based on performance feedback.

5. Limitations and Future Work

This framework remains untested and untrained. Debugging, optimization, and systematic benchmarking using established AGI benchmarks like the ARC dataset are crucial next steps. The interplay between neural and symbolic components also requires refinement. Additionally, the current implementation lacks robust error handling and optimization mechanisms for real-time applications.

Potential areas for future research include:

- Implementing scalable distributed computing support.
- Enhancing the ontology builder with more advanced NLP models.
- Developing task-independent reinforcement learning agents.
- Benchmarking performance across diverse domains to evaluate transfer learning capabilities.

6. Conclusion

PROMETHEUS 1.0 presents a modular approach to AGI development, integrating diverse cognitive components within a unified architecture. Future work should focus on validation, performance optimization, and exploration of novel cognitive mechanisms. While the current state is purely conceptual, the design choices provide a foundation for iterative development towards human-like artificial intelligence.

By @EliasofIX on GitHub and X