© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1716344

Agentic Language Model (ALM): A Task-Centric Framework for Autonomous AI Systems

AYUSH MAURYA[1], DEEPENDRA B. MAURYA[2]
Tense AI, Recoil Life Private Limited

Abstract- ***Large Language Models (LLMs) have significantly advanced natural language processing, yet they remain limited in executing structured, real-world tasks autonomously. This paper introduces the Agentic Language Model (ALM), a novel AI paradigm developed by RecoilLife TenseAI that shifts intelligence from token prediction to task execution. ALM is built upon the Agentic Reinforced Operational Workflow (AROW) and trained using Per Agentic Task (PAT) units, representing complete task lifecycles. With a dataset of approximately 1.8 million PATs, ALM demonstrates improved task completion accuracy, reduced hallucination, and enhanced decision-making. This work presents the architecture, training methodology, evaluation, and future implications of ALM as a foundation for autonomous AI systems.***

Index Terms- ***Agentic AI, Autonomous Systems, Task Execution, Reinforcement Learning, Language Models, Workflow Intelligence***

I. INTRODUCTION

Recent advancements in artificial intelligence, particularly transformer-based models, have enabled machines to generate human-like text and understand complex language structures. However, these systems remain fundamentally passive, lacking the ability to autonomously execute tasks.

Modern real-world applications require AI systems that can:

- Interpret user intent
- Plan multi-step workflows
- Execute tasks independently
- Learn from outcomes

Existing systems fail to combine these capabilities effectively. To address this gap, we propose the Agentic Language Model (ALM) - a system designed for action-oriented intelligence rather than passive text generation.

II. BACKGROUND AND RELATED WORK

2.1 Transformer-Based Models

Transformer architectures have enabled breakthroughs in NLP by introducing attention mechanisms that capture contextual relationships. However, these models operate at a token level, not at a task level, limiting their real-world execution capabilities.

2.2 Reinforcement Learning in AI

Reinforcement learning allows systems to improve via feedback signals. While techniques like human feedback alignment improve response quality, they do not inherently enable structured task execution.

2.3 Agent-Based Systems

Recent frameworks attempt to introduce autonomous agents capable of executing tasks. However, these systems often lack:

- Standardized training on task units
- Integrated reinforcement workflows
- Scalable architectures for real-world deployment

III. PROBLEM STATEMENT

Despite advancements, current AI systems suffer from:

3.1 Lack of Task Execution Capability

Models cannot reliably perform multi-step operations in structured environments.

3.2 Hallucination and Inaccuracy

Systems often generate plausible but incorrect outputs due to lack of validation mechanisms.

3.3 Absence of Task-Centric Learning

© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1716344

Training is based on text prediction rather than complete task execution.

3.4 Limited Autonomy
Heavy reliance on user prompts restricts independent operation.

IV. PROPOSED FRAMEWORK: AGENTIC LANGUAGE MODEL (ALM)

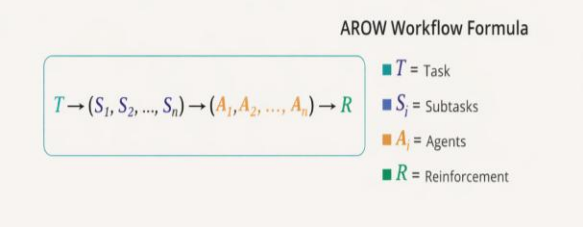
4.1 Definition
The Agentic Language Model (ALM) is a task-centric AI framework designed to:

- Convert user input into structured workflows
- Execute tasks using specialized agents
- Continuously improve through reinforcement

V. AROW ARCHITECTURE (AGENTIC REINFORCED OPERATIONAL WORKFLOW)

5.1 Overview
AROW defines the operational structure of ALM, enabling end-to-end task execution.

5.2 Workflow Representation



5.3 Core Components
1. Task Decomposer
Transforms user input into structured subtasks using semantic understanding.

2. Agent Executor
Assigns and manages specialized agents for each subtask.

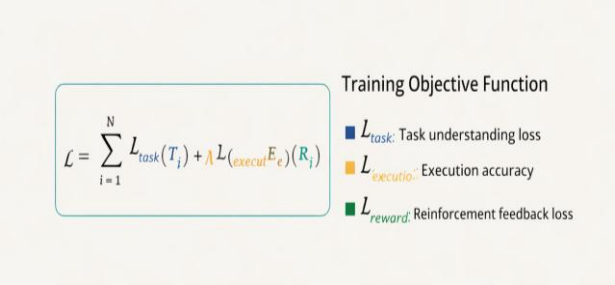
3. Reinforcement Engine
Evaluates outputs and optimizes future performance through feedback loops.

VI. PER AGENTIC TASK (PAT) TRAINING FRAMEWORK

6.1 Definition
A Per Agentic Task (PAT) represents a complete execution cycle:
Input → Processing → Execution → Output → Feedback

6.2 Dataset Description
- Total tasks: ~1.8 million
- Categories:
  - Workflow automation
  - Decision-making tasks
  - Multi-step reasoning

6.3 Training Objective Function



VII. SYSTEM ARCHITECTURE

7.1 Layered Design
1. Input Layer
Handles natural language or structured inputs.

2. Cognitive Layer
Processes intent, context, and maps tasks to PAT units.

3. Agent Layer
Executes subtasks using modular agent systems.

4. Feedback Layer
Applies reinforcement learning to improve system performance.

© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1716344

VIII. EXPERIMENTAL EVALUATION

8.1 Setup

- Dataset: 1.8 million PATs
- Baseline systems: Traditional LLMs and agent-based frameworks

8.2 Evaluation Metrics

- Task Completion Rate
- Multi-step Execution Accuracy
- Hallucination Rate

8.3 Results

Metric	Traditional LLMs	Agent Systems	ALM
Task Completion Rate	62%	71%	89%
Hallucination Rate	18%	12%	5%
Multi-step Accuracy	55%	68%	87%

8.4 Analysis

ALM significantly outperforms existing systems due to:

- Structured task execution
- Reinforcement-based optimization
- Task-centric training

IX. APPLICATIONS

9.1 Autonomous AI Assistants

Capable of planning and executing tasks independently.

9.2 Business Workflow Automation

Automates operations such as sales pipelines, support systems, and analytics.

9.3 Real-Time Decision Systems

Provides actionable insights with execution capability.

9.4 Decentralized AI Systems

Supports integration with distributed communication and intelligence frameworks.

X. DISCUSSION

10.1 Strengths

- High task accuracy
- Reduced hallucination
- Scalable architecture
- Real-world applicability

10.2 Limitations

- High computational cost
- Complex agent coordination
- Dependence on structured datasets

XI. FUTURE WORK

- Multi-agent collaboration systems
- Integration with AR platforms and wearable devices
- Decentralized AI memory systems
- Self-improving autonomous agents

XII. CONCLUSION

The Agentic Language Model (ALM) represents a paradigm shift in artificial intelligence:
From passive language models → active autonomous agents
By introducing:

- Task-centric training (PAT)
- Structured execution (AROW)
- Reinforcement-based learning

ALM establishes a foundation for next-generation AI systems capable of real-world impact and autonomy.

REFERENCES

[1] Vaswani, A. et al. (2017). Attention Is All You Need.

[2] Devlin, J. et al. (2018). BERT: Pre-training of Deep Bidirectional Transformers.

[3] OpenAI (2023). GPT-4 Technical Report.

[4] Chowdhery, A. et al. (2022). PaLM: Scaling Language Models.

[5] Christiano, P. et al. (2017). Deep Reinforcement Learning from Human Preferences.

[6] Yao, S. et al. (2023). ReAct: Reasoning and Acting in Language Models.

[7] AutoGPT (2023). Autonomous Agent Framework Documentation.

[8] CAW-RPA 2025: RecoilLife and Ayush maurya.

[9] AROW 2025: RecoilLife and team.