Faculty Of Engineering and Natural Sciences
Department of Electrical and Electronic Engineering
May 4, 2025

Integrating Large Language Models for Robot Task Planning

Almira Demirkıran 200202150

E-mail: almira.demirkiran@std.antalya.edu.tr

Mehmet Tosun 190202009

E-mail: mehmet.tosun@std.antalya.edu.tr

Abdallah Rasthy 200201125

E-mail: ali.rashty@std.antalya.edu.tr

Rustam Akhmedov 210201113

E-mail: rustam.akhmedov@std.antalya.edu.tr

Contents

Introduction

Large Language Models (LLMs) have revolutionized natural language processing (NLP) and are increasingly at the center of robot task planning, enabling autonomous systems to interpret sophisticated instructions, perform contextual reasoning, and integrate into robotic control frameworks. Their ability to bridge linguistic understanding with physical action renders them groundbreaking tools in human-robot collaboration and environmental adaptation. This paper provides a comprehensive overview of the structural foundation of LLMs, numerical sentence decomposition, training techniques, integration with robotic systems, neural network-based algorithms, and their assimilation with LIDAR data for robust task planning.

1. Large Language Model (LLM) Architecture

Large Language Models heavily rely most on the Transformer architecture, a deep neural network framework that makes extensive use of attention mechanisms to model linguistic structures with high accuracy.

1.1. Transformer Architecture

First introduced by Vaswani et al. (2017), the Transformer architecture overcomes RNNs' sequential computation limits through parallelized computing. Its most significant features are:

• **Encoder-Decoder Framework:** Encoder projects input sequences to contextualized vector representations, and decoder generates output sequences. LLMs have either encoder-only (e.g., BERT) or decoder-only (e.g., GPT) architecture in the majority.

• **Multi-Head Attention:** Captures token relations in several subspaces, enabling the model to attend to several contextual dependencies simultaneously.

• **Positional Encoding:** In order to overcome token order, Transformers incorporate sinusoidal or learned positional embeddings, adding sequence information to token representations.

• **Feed-Forward Networks:** Employed at every layer, the networks introduce non-linear mappings that enhance the ability of the model to generalize through complex linguistic structures.

1.2. Outside Token and Embedding

Even though tokenization and embedding form the foundation of LLMs, they possess a great deal more than these processes:

• **Contextual Embeddings:** Unlike static embeddings (e.g., Word2Vec), LLMs provide context-sensitive representations of tokens, which vary by context, capturing polysemy and syntactic nuance.

•**Attention Weights:** The attention mechanism captures inter-token dependencies, enabling the modeling of long-range relationships critical to understanding complex instructions.

•**Hierarchical Transformations:** Sequential Transformer layers progressively refine token representations, creating increasingly abstract and semantically more informative embeddings that encode linguistic hierarchies.

2. Sentence Decomposition to Numerical Representation

Sentence decomposition to numerical representation is a cornerstone of the utility of LLMs for robot task planning, enabling transformation of linguistic input into executable robot commands.

2.1. Tokenization and Initial Representation

Sentences are segmented into tokens using subword-based models (e.g., Byte-Pair Encoding or WordPiece) that make a vocabulary-size vs. morphological-coverage trade-off. Tokens are projected to dense vectors using an embedding layer, adding positional encodings to preserve sequence information.

2.2. Contextual Processing

The Transformer converts token embeddings through its attention mechanisms, computing weighted relations between tokens. Multi-head attention ensures that the representation of each token learns from both its syntactic and semantic context, enabling high-level comprehension of dependencies. Output embeddings represent the relational structure of the sentence.

2.3. Matrix Representation

The processed sentence is represented as an (n \times d) matrix, having (n) tokens and (d) dimensionality per embedding. The matrix holds the semantic meaning of the sentence and serves as a numerical interface to robotic systems. In some architectures, a sentence-level representation (e.g., a [CLS] token or pooled embedding) is derived for downstream processing.

2.4. Integration with Robot Systems

Representations are converted to robot use via additional operations such as mapping to structured representations exploitable by motion planning or task decomposition algorithms. Representations are taken as inputs to control structures in order to enable linguistically derived tasks to be performed.

3. Model Training

Training LLMs is computationally expensive with large-scale data and sophisticated optimization methods, divided into pre-training and fine-tuning phases.

3.1. Pre-training

Pre-training equips LLMs with general linguistic knowledge using huge, unlabeled text datasets:

•Objective Functions:

•Masked Language Modeling (MLM): Randomly masked tokens are predicted from context, encouraging bidirectional understanding (e.g., BERT).

•Causal Language Modeling (CLM): Models predict the sequence's next token, enabling generative capabilities (e.g., GPT).

• Next Sentence Prediction (NSP): Models are trained to detect sequential relationships among sentences.

• Data and Scale: Pre-training is done using corpora containing billions of tokens (e.g., Common Crawl, Wikipedia), processed on high-performance computing clusters for weeks (e.g., GPUs or TPUs).

• Challenges: The computational expense, driven by billions of parameters, creates scalability and environmental concerns, necessitating efficient training paradigms.

3.2. Fine-tuning

Fine-tuning converts pre-trained models to specific tasks, e.g., robot task planning:

• Task-Specific Data: Models are trained on robotic commands, human-robot interaction dialogue, or task planning domains.

• Optimization: Fine-tuning employs low learning rates to preserve pre-trained knowledge and acquire task-specific patterns, following transfer learning guidelines.

• Challenges: Limited task-specific data can lead to overfitting, while generalization over diverse robotic settings is a severe challenge.

3.3. Training Challenges

• Data Scarcity: Annotated high-quality datasets for robotic tasks are not common, and this limits model performance.

• Computational Overhead: Even large model fine-tuning requires heavy resources, limiting accessibility.

• **Generalization:** Finding a balance between task adaptation and robustness across diverse contexts remains a persistent challenge.

4. LLM and Model Matching

Incorporating LLMs into robot systems requires linguistic representations to be matched against control algorithms, a union of semantic understanding and physical execution.

4.1. Output Integration

LLM outputs, typically vector representations or constructed instructions, serve as inputs to robotic control structures. These are operated on by motion planning, task decomposition, or control algorithms, translating linguistic intent into actionable commands.

4.2. Hybrid Approaches

LLMs are paired with diverse robotic paradigms:

•**Deep Reinforcement Learning (DRL):** LLMs provide abstract planning for tasks, whereas DRL acquires low-level control policies in dynamic settings.

•**Stuartification (DRL):** LLMs transform instructions into symbolic planning languages (e.g., PDDL), enabling combination with traditional planners.

•**Control Theory:** LLM outputs are mapped into state-space models or proportional-integral-derivative (PID) controllers and executed directly.

4.3. Challenges

•**Semantic-Physical Gap:** Linguistic descriptions being mapped into physical actions are made difficult due to uncertainty of the environment and partial observability.

•**Real-Time Processing:** Computational requirements of Large LLMs could be in conflict with low-latency requirements of robot applications.

•**Robustness:** Stable performance on many tasks and environments calls for robust model design.

5. Neural Network-Based Algorithms

LLMs are a form of neural network-based algorithms, combined with other structures to enhance robot task planning.

5.1. Transformer-Based Models

Transformers form the basis of LLMs, employing attention mechanisms to represent contextual dependencies. In robotics, they facilitate parsing of commands and task planning by generating semantically informative representations.

5.2. Convolutional Neural Networks (CNNs)

CNNs excel at spatial information processing, such as visual input or LIDAR point clouds. Combined with LLMs, environmental context is introduced, enhancing situational awareness.

5.3. Recurrent Neural Networks (RNNs)

RNNs, less popular due to Transformer prevalence, are suited for sequential instruction processing or time series data and are conducive to tasks with temporal dependencies.

5.4. Reinforcement Learning

Reinforcement learning enables robots to learn from interactions with the world. LLMs make the process easier by defining reward functions or high-level policies, structuring the learning environment.

6. LIDAR and LLM Output Integration

LIDAR (Light Detection and Ranging) provides high-resolution 3D environmental data, which, when integrated with LLM outputs, enhances task planning robustness.

6.1. LIDAR Data Processing

LIDAR generates point clouds, processed through noise filtering, segmentation, and object recognition. These data are transformed to vector representations or 3D grids, appropriate for neural network models.

6.2. Fusion with LLMs

LLM outputs, as linguistic commands, are integrated with environmental data from LIDAR to produce contextually driven task plans. This integration correlates spatial constraints with semantic intent, typically through multi-modal neural networks or intermediate mapping layers.

6.3. Technological Issues

•**Alignment of data:** Temporal and spatial synchronization between linguistic and LIDAR data involves sophisticated preprocessing techniques.

•**Dimension mismatch:** The high dimensionality of the LIDAR data makes it difficult for combination with LLM representations and requires either dimensionality reduction or feature extraction.

•**Real-Time Constraints:** Real-time applications are subject to low-latency constraints and need hardware acceleration and efficient architectures.

7. Conclusion and Future Directions

Large Language Models hold revolutionary potential for robot task planning by uniting linguistic understanding and physical action. Their Transformer models, contextual representation, training protocols, neural network combinations, and complementarity with LIDAR data form a robust paradigm for autonomous agents. Computational expense, data scarcity, real-time processing needs, and the semantic-physical gap are still challenges.

Future directions may involve:

• **Data-Efficient Learning:** Constructing generalizing models from minimal task-specific data.

• **Multi-Modal Fusion:** Enhancing the integration of linguistic, visual, and sensor data towards worldwide perception.

• **Lightweight Models:** Creating computationally light LLMs for real-time robotic tasks.

References

1. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is All You Need. Advances in Neural Information Processing Systems, 30.

2. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

3. Brown, T. B., Mann, B., Ryder, N., et al. (2020). Language Models are Few-Shot Learners. Advances in Neural Information Processing Systems, 33.

4. Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction. MIT Press.