Bridging Natural Language and MAVLink: Dataset Generation and SLM Fine-Tuning for UAV Command Execution

Anonymous ACL submission

Abstract

Accurately interpreting natural language commands is crucial for deploying autonomous unmanned aerial vehicles (UAVs) in industrial environments. This study introduces the UAVIntent dataset by systematically using 16 Myers-Briggs Type Indicator (MBTI) personality types and drone operator roles for synthesizing the dataset with One-Shot Chain-of-Thought (CoT) based dataset pipeline. The dataset consists of 122 distinct command types derived from MAVLink documentation, totaling 19,088 data points.

We conducted extensive experiments on this dataset, evaluating different approaches for converting natural language instructions into MAVLink-based commands and extraction of parameters by fine-tuning multiple small language models (SLMs) and a retrieval-augmented generation (RAG) framework leveraging Phi-3. Among SLMs, DistilBERT achieves the highest command classification accuracy (99.22%), outperforming BART-Base (97.65%), BART-Large (98.83%) and RAG + Phi-3 (97.42%). For parameter extraction, RAG + Phi-3 attains the highest exact match accuracy (90.74%) and slot-wise accuracy (95.47%), but at a significantly higher computational cost. DistilBERT , while less accurate (82.34% exact match, 92.35% slot-wise), offers a more time-efficient alternative for real-time UAV command processing.

1 Introduction

Unmanned Aerial Vehicles (UAVs), commonly referred to as drones, are being increasingly utilized across domains such as defense, surveillance, disaster response, and autonomous logistics (Javaid et al., 2024). The ability to interpret natural language commands accurately and efficiently is critical for enabling autonomous UAV operations in both structured and unstructured environments (Sikorski et al., 2024). Traditional UAV control pipelines rely on rigid, predefined scripting interfaces, limiting adaptability and requiring manual reconfiguration for different mission scenarios (Javaid et al., 2024).

Recent advances in Natural Language Processing (NLP) have made it increasingly feasible to design UAV systems that respond to flexible human input rather than rigid control scripts (Tellex et al., 2011; Brown et al., 2020). Leveraging pretrained language models helps reduce redundant engineering efforts while enabling dynamic and semantically rich command execution. However, translating open-ended language into structured, MAVLink-compatible instructions remains a key challenge—particularly under constraints of real-time execution and precision control (Wei et al., 2022; Lewis et al., 2020).

Command Text:
"Switch relay instance 0 on and off 5 times, with each cycle lasting 3 seconds."

Intent: CycleRelay

Slots:

Name	Value	Description
instance	0	Relay instance number
count	5	Number of cycles
time	3	Cycle time in seconds

Figure 1: Example of a structured data point from the MAVLingo dataset, illustrating the mapping between a natural language command and its corresponding structured representation. The "intent" field identifies the high-level UAV command (e.g., `CycleRelay`), while the "slots" capture essential parameters (e.g., instance ID, repetition count, duration).

To address these limitations, we introduce a dataset for UAV command understanding that enables intent classification and slot filling for structured MAVLink command generation. Covering all 122 MAVLink command types with over 19,000 synthetic examples, the dataset includes role-based and personality-aware language variations. We benchmark four SLMs and LLMs using both fine-tuning and prompting, offering one of the first systematic comparisons for joint intent-slot prediction, and analyzing trade-offs in accuracy, computational cost, and deployment efficiency.

The remainder of this paper is organized as follows: In Section 2, we review prior work on command interpretation and language modeling across domains. Section 3 describes dataset generation process, annotation strategy, dataset statistics, and error analysis. Section 4 outlines the experimental setup, including model configurations and evaluation metrics. Section 5 presents a comprehensive analysis of model performance on classification and extraction tasks, inference efficiency, and real-world applicability. Finally, Section 6 concludes the paper with key findings and future research directions.

2 Related Work

The task of translating natural language into structured, executable commands has been a longstanding research challenge in fields like semantic parsing and robotic instruction following. Early works like GeoQuery (Zelle and Mooney, 1996) and ATIS (Dahl et al., 1994) laid the groundwork for structured query generation. More recent efforts have targeted physical domains. For instance, (Tellex et al., 2011) mapped language to navigation goals in robots, while (Misra et al., 2018) proposed a neural model for parsing natural instructions with visual observations for instruction execution. Similarly, (Matuszek et al., 2012) worked on grounded language understanding for robot perception and action.

However, the focus of these efforts has been limited to general robotics or indoor navigation. Works directly mapping to aviation control languages like MAVLink remain sparse, with most studies relying on fixed templates or hardcoded rules. Now we will discuses the datasets and approaches used in different domains.

Datasets for grounded command learning have expanded from simple action domains to rich, multi-step tasks involving visual and contextual grounding. Dataset TEACh (Blukis et al., 2021), and capture navigation in complex environments. Synthetic datasets such as SCAN (Lake and Baroni, 2018) have been useful for compositional generalization. The approach used by (Chen et al., 2019) for intent and slots classification over the snips (Coucke et al., 2018) and ATIS (Dahl et al., 1994) dataset.

Despite this progress, datasets combining operator context, psychological profiles, or role diversity remain largely unexplored. While few works incorporate user roles in dialog systems, none leverage MBTI-style (Myers, 2003) personality variation for command expression diversity in UAV settings for synthetic data generation. Our work addresses this gap by systematically generating commands using operator-role and personality-informed templates aligned with MAVLink specifications (mav).

Slot filling and intent detection are foundational tasks in task-oriented dialogue and command systems (Tur and De Mori, 2011). Joint modeling approaches such as (Chen et al., 2019), SlotRefine (Qin et al., 2020), and BERT-CRF () have been shown to improve performance by sharing representations between tasks.

However, most of these models are benchmarked on standard NLP datasets like ATIS (Dahl et al., 1994) or SNIPS (Coucke et al., 2018) and not adapted to domains requiring precise numerical and spatial parameter extraction, such as UAV control. The need for real-valued slot prediction with strict accuracy constraints is largely absent in existing literature. We address this with a MAVLink-grounded dataset, and compare SLM-based fine-tuning with LLM-based multi-stage prompting for UAV-specific slot filling. Now we will discuss the latest Prompt Engineering based approaches for solving the NLP tasks.

The rise of prompting-based techniques for LLMs has revitalized interest in zero-shot and few-shot learning (Brown et al., 2020). Chain-of-thought (CoT) prompting (Wei et al., 2022) and tool-augmented LLMs (Chen et al., 2023) have improved structured reasoning. RAG architectures (Lewis et al., 2020) and retrieval-augmented decoding (Izacard and Grave, 2021) allow LLMs to better reason over external structured documents such as APIs and manuals.

While these models have achieved success in question answering and code generation, their effectiveness in interpreting and executing low-level UAV control instructions remains underexplored. Our work introduces CommandPrompt Cascade, a novel multi-stage prompting pipeline combining

Phi-3.5 and GPT-4o, and benchmarks it against fine-tuned SLMs on command intent classification and slot extraction.

To the best of our knowledge, no prior work has provided a comprehensive dataset that combines UAV-specific command structures, role-based operator variation, and psychological typing (MBTI) (Myers, 2003) for command classification from text. Additionally, there has been little exploration of the trade-offs between small model fine-tuning and multi-stage prompting with LLMs for real-time command interpretation. So there are various novel contribution of current research in understanding the effectiveness of prompting and there comparison with SLM for the UAV Command classification from text in industrial settings.

A Novel, Personality-Aware Dataset for UAV Command Understanding We present an extended version of the MAVLingo dataset, consisting of 19,088 instances across 122 distinct MAVLink command types, systematically generated using MBTI (Myers, 2003) personality types and drone operator roles to simulate diverse linguistic expressions in natural UAV command scenarios.

Synthetic Data Generation Pipeline Aligned with MAVLink Standards We design a robust, structured pipeline for synthetic data generation, incorporating MAVLink command schemas, paraphrased natural language templates, and command slot annotations to bridge unstructured input and executable MAVLink messages.

Comparative Evaluation of SLM Fine-Tuning vs. LLM Prompting We conduct a detailed evaluation of SLM-based fine-tuning approaches (DistilBERT, BART-base, BART-large) against LLM prompting and RAG-based methods, analyzing performance on both intent classification and slot filling.

Empirical Trade-off Analysis: Accuracy vs. Efficiency We highlight the accuracy-efficiency trade-offs between fine-tuned lightweight models and LLM pipelines, showing that DistilBERT achieves 99.22% intent accuracy with minimal inference cost, while RAG + Phi-3.5 achieves 95.47% slot-wise accuracy at higher compute overhead.

3 Dataset

Our research aims to develop a command classification dataset that enhances the operational efficiency of drone users by accurately mapping user instructions to MAVLink commands. A well-structured and high-quality dataset is essential for improving the precision of command classification models, ensuring reliable communication between users and autonomous drone systems.

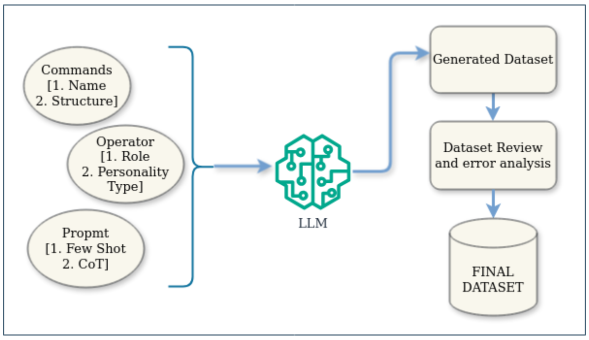

Figure 2: Overview of the UAVIntent dataset generation pipeline. The process begins by combining command definitions (including command names and MAVLink-compatible structures), drone operator characteristics (role and MBTI-based personality type), and carefully designed prompts (Few-Shot and Chain-of-Thought). These inputs are fed into GPT 4(o) (OpenAI et al., 2024) to generate natural language command instances.

3.1 Dataset Creation and Annotation

A systematic review of the MAVLink protocol (MAVLink Contributors, 2024) yielded 157 commands, of which 122 were retained (35 deprecated). To enhance linguistic and behavioral diversity, we incorporate operator roles and personality types of MBTI (Myers, 2003) during data generation.

ChatGPT-4o was prompted using engineered templates combining command details, role-personality mappings, and output format constraints. Prompts produced natural language instructions with Chain-of-Thought reasoning, returning structured JSON including intent, slots, and descriptions. Each entry was annotated with slot names, values, and intent. Figure 2 shows the pipeline for the generation of the dataset.

Dataset Generation Phases:

- **Phase 1: Initial Testing (n=38)** — Identified issues with missing parameters and formatting; 42% failure rate.

- **Phase 2: Template Refinement** — Enforced parameter completeness, explicit enums, and format clarification.

- **Phase 3: Bulk Generation** — Applied refined prompts to systematically generate data across command-role-MBTI combinations.

- **Phase 4: Human Validation** — Reviewed by three annotators and one expert. Faulty entries were regenerated.

Error Analysis

Two recurring issues were identified in the generated JSON outputs:
(1) missing values in the "commands" field (affecting 4% of samples), and
(2) blank "Mav_cmd_name" entries due to misclassification errors (impacting 40%). These issues were traced to limitations in prompt design and were resolved through improved template structuring and post-processing logic, ensuring both semantic accuracy and syntactic validity in the final outputs.

3.2 Dataset Statistics

The dataset comprises a total of 19,088 samples, generated from 122 unique MAVLink command classes. For each command, we aimed to create approximately 160 samples (derived from 16 MBTI personality types × 10 samples per type) to ensure linguistic diversity and behavioral variability. However, due to occasional generation inconsistencies—such as incomplete slot filling or incorrect formatting—some instances (e.g., only 9 instead of 10 per personality type) were either reduced or discarded when errors could not be automatically corrected. This pragmatic filtering ensured the final dataset retained only high-quality, structurally valid examples suitable for training and evaluation.

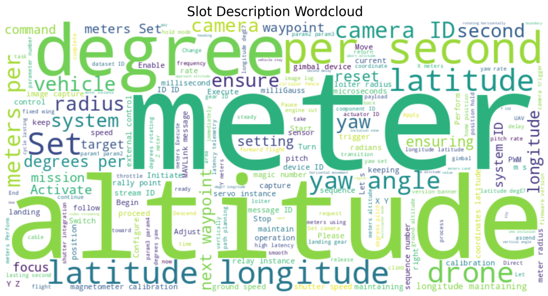

Figure 3: Word cloud of slot descriptions extracted from the MAVLingo dataset. Prominent terms such as *altitude*, *meter*, *longitude*, and *degree* highlight frequently used parameters in UAV command instructions. This visualization reflects the common structure and semantics of drone control vocabulary.

The dataset covers **281 unique slot types**, with 0–7 slots per command (avg. 4.3). Common slots include `altitude`, `yaw`, `latitude`, and `delay`, spanning geospatial, temporal, and categorical dimensions. Slot descriptions are 85.2% textual and 14.8% numerical. Figure 3 represents a word cloud of the slots present in the dataset.

4 Experiment and Methodology

This section presents the experimental setup, methodologies, evaluation metrics, and results for the task of command classification and parameter extraction. Three different approaches were employed:

1. Fine-tuning a BART-based and BERT-based Sequence Labeling Model (SLM)
2. Retrieval-Augmented Generation (RAG) with Large Language Models (LLMs)
3. Prompt-Based Few-Shot Learning with LLMs

Each method follows a two-step process: (i) identifying the command class from the given instruction and (ii) extracting relevant parameters, followed by slot-filling to generate a structured JSON output.

4.1 Experimental Setup

For fine-tuning, **facebook/bart-base, facebook/bart-large, and distilbert-base-uncased** were trained on the dataset using a cross-entropy loss function. The training objective was to classify **(**command types (CommandName) and **extract slot parameters** with high accuracy. Each model was optimized to effectively capture the relationships between input commands and their structured representations while maintaining robust generalization across different command variations.

The **RAG-based approach** integrated an external retrieval mechanism with an LLM. This method first retrieved relevant command descriptions before generating structured responses.

In the **prompt-based approach**, structured prompts containing command descriptions, operator characteristics, and few-shot examples were provided to the LLM. The model was instructed to generate JSON-formatted outputs directly.

4.2 Dataset

Our dataset consists of textual input commands and structured output containing slot values and a class label, CommandName. The dataset is stored in CSV format with two columns: Input (containing natural language instructions) and Output (containing slot-value pairs and CommandName) in JSON format. The dataset follows the format :

- **Input:** "Enable and reset the triggering system for all connected cameras while avoiding pause actions."
- **Output:** {"enable" : 1, "reset" : 1, "pause" : -1, "target_camera_id": 0, "CommandName": 30}

The dataset comprises a total of 19,088 data points, which are divided into training and testing sets using an 80:20 split. The training set consists of 15,255 samples, while the test set contains 3,833 samples. Additionally, the dataset includes 122 unique command labels (CommandName), representing different categories of commands.

4.3 Evaluation Metrics

The performance of all models was assessed using the following evaluation metrics:

- **Command Classification Metrics:** Accuracy, precision, recall, and F1-score were used to measure the effectiveness of predicting the correct CommandName. These metrics evaluate the model's ability to correctly classify commands while maintaining a balance between precision and recall.
- **Slot Filling Metrics:**
  Exact Match (EM): Measures the percentage of predictions that exactly match the ground truth JSON structure, ensuring the complete correctness of extracted slots.
  Slot-wise Accuracy: Assesses the correctness of individual slot predictions, providing a more detailed evaluation of the model's ability to extract parameters accurately.

4.4 Hardware and Hyperparameters

- **Hardware:** Experiments were performed on a system equipped with an NVIDIA RTX A5500 GPU.
- **Batch Size:** 4
- **Learning Rate:** 3e-5
- **Number of Epochs:** 15
- **Weight Decay:** 0.01 (used to mitigate overfitting)

5 Results and Analysis

We evaluate two paradigms for converting natural language UAV commands into structured MAVLink instructions: (i) fine-tuned Small Language Models (SLMs), and (ii) prompting-based Large Language Models (LLMs) using a RAG framework. The performance is assessed using command classification accuracy, slot extraction metrics, and inference efficiency.

5.1 Classification and Extraction Performance

Table 1 and Table 2 summarize the performance of all models on the intent classification and slot extraction tasks, respectively. Among the SLMs, DistilBERT achieved the highest classification accuracy (99.22%), outperforming BART-Large (98.83%), BART-Base (97.65%), and RAG + Phi-3 (97.42%).

While RAG + Phi-3 achieved the best slot-level extraction performance, with 90.74% exact match accuracy and 95.47% slot-wise accuracy (Table 2), this improvement came at the cost of significantly higher inference time, as discussed in the following section.

Model	**Accuracy (%)**	**Precision**	**Recall**	**F1**
DistilBERT	**99.22**	98.53	98.38	98.44
BART-Large	98.83	**98.98**	**98.80**	**98.80**
BART-Base	97.65	97.61	96.76	96.88
RAG + Phi-3	97.42	95.85	95.07	95.34
Prompt + Phi-3	49.18	-	-	-
Prompt + GPT-4o	59.43	-	-	-

Table 1: Intent classification performance: Performance of various models on UAV command intent classification. Metrics include accuracy, precision, recall, and F1-score. DistilBERT achieves the highest overall accuracy, while prompting-based LLMs underperform in high-class cardinality settings.

Model	**EM Accuracy (%)**	**Slot-wise Accuracy (%)**
DistilBERT	82.34	92.35
BART-Large	70.18	82.21
BART-Base	69.89	81.26
RAG + Phi-3	**90.74**	**95.47**

Table 2: Slot extraction performance across models. RAG + Phi-3 achieves the highest exact match (EM) and slot-wise accuracy, while DistilBERT provides a strong balance between performance and efficiency.

While RAG + Phi-3 excelled in slot-level extraction with 90.74% exact match accuracy and 95.47% slot-wise accuracy, it came at the cost of significantly higher inference time, as discussed below.

5.2 Inference Time and Latency Behavior

Figure 4 illustrates the normalized inference time distributions. DistilBERT exhibits a sharp, narrow peak indicating highly consistent and low-latency behavior. In contrast, prompting-based methods such as Phi-3 show greater variance and long-tail latency patterns. This highlights the practicality of SLMs for real-time UAV deployments where bounded inference time is critical. Table 3 shows the inference time analysis.

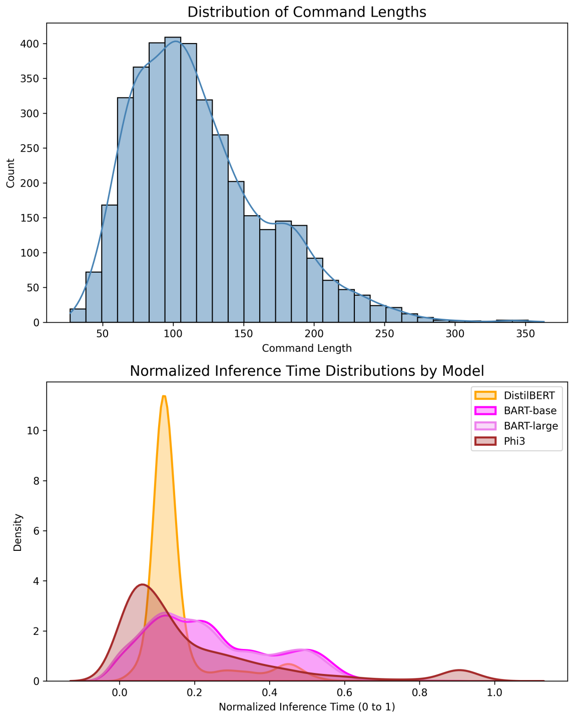

Figure 4: Distribution of command lengths and corresponding inference time by model.

5.3 Impact of Class Cardinality on Prompting Performance

We further analyze prompting behavior in large label space settings. While prompting methods such as few-shot CoT with Phi-3 or GPT-4o achieve near-human accuracy (up to 95%) on binary tasks like IMDb sentiment classification, their performance significantly degrades with high class cardinality. On our MAVLingo dataset, comprising 122 unique intent classes, prompting accuracy dropped to 40.1%.

This degradation stems from: (i) overlapping semantics between similar command types, (ii) in-context learning limitations in retaining a large number of classes, and (iii) lack of fine-tuned inductive bias in general-purpose LLMs. These findings emphasize that prompt-based LLMs are not robust in complex industrial settings requiring fine-grained classification.

Model	Inference Time (mean± std) in seconds
DistilBERT	0.49 ± 0.02
BART-Large	0.16 ± 0.08
BART-Base	0.10 ± 0.05
RAG + Phi-3	5.78 ± 5.61

Table 3: Inference Time Analysis: Comparison of average inference times (in seconds) and their standard deviations across different language models. DistilBERT and BART variants exhibit low and stable latency, while RAG combined with Phi-3 shows significantly higher and more variable inference time due to retrieval and generation overhead.

5.4 End-to-End Analysis: Suitability for UAV Applications

In practical UAV applications, both accuracy and latency are essential. While RAG + Phi-3 provides high slot-filling accuracy, its inference time is unpredictable, making it unsuitable for time-sensitive tasks. In contrast, DistilBERT offers a balanced trade-off—strong classification accuracy, acceptable slot performance, and highly predictable runtime characteristics.

5.5 Conclusion of Findings

Our results indicate that while prompting-based LLMs like Phi-3 show promise in structured slot extraction, they fall short in multi-class intent classification tasks and are hindered by high latency variance. Fine-tuned SLMs, especially DistilBERT, demonstrate robustness, scalability, and real-time efficiency—making them well-suited for industrial drone command understanding and execution.

5.6 Limitation

We used GPT-4o for synthetic data generation aligned with MAVLink schema. While the source context and prompt formats for inference differ, we acknowledge a partial overlap in model usage. However, we ensured separation between training-time generation and test-time evaluation to avoid memorization artifacts.

6 Conclusions and future works

In this work, we presented a drone dataset, designed for structured UAV command understanding through intent classification and slot extraction.

By integrating role-based and MBTI-informed language variations, the dataset captures diverse real-world expressions of MAVLink commands across 122 categories. We benchmarked both fine-tuned Small Language Models (SLMs) and Large Language Model (LLM)-based prompting approaches, offering a detailed analysis of their performance and efficiency trade-offs. Our results demonstrate that SLMs like DistilBERT provide competitive accuracy with lower computational overhead, making them suitable for real-time UAV applications, while LLMs offer higher extraction precision at greater cost. We hope this dataset and evaluation benchmark will serve as a foundation for future research in language-grounded drone control.

References

Introduction · MAVLink developer guide.

Valts Blukis, Ziad Al-Halah, Cynthia Ross, et al. 2021. Teach: Task-driven embodied agents that chat. In *EMNLP*.

Tom Brown, Benjamin Mann, Nick Ryder, et al. 2020. Language models are few-shot learners. In *NeurIPS*.

Nuo Chen, Hongguang Li, Baoyuan Wang, and Jia Li. 2023. From good to great: Improving math reasoning with tool-augmented interleaf prompting. *arXiv preprint arXiv:2401.05384*.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. In *Interspeech*.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, and Alexandre Caulier ... 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *Preprint*, arXiv:1805.10190.

Deborah A Dahl, Madeleine Bates, Michael Brown, et al. 1994. Expanding the scope of the atis task: The atis-3 corpus. In *Proceedings of the Workshop on Human Language Technology*, pages 43–48.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Shumaila Javaid, Nasir Saeed, and Bin He. 2024. Large language models for uavs: Current state and pathways to the future. *Preprint*, arXiv:2405.01745.

Brenden M Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *ICLR*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *NeurIPS*.

Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer, and Dieter Fox. 2012. A joint model of language and perception for grounded attribute learning. In *ICML*.

MAVLink Contributors. 2024. Introduction · mavlink developer guide. `https://mavlink.io/en/`. Accessed: 2025-04-29.

Dipendra Misra, Abhinav Shrivastava, and Abhinav Gupta. 2018. Mapping instructions and visual observations to actions with reinforcement learning. In *EMNLP*.

Isabel Briggs Myers. 2003. *MBTI manual: A guide to the development and use of the Myers-Briggs Type Indicator*. Cpp.

OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, and J. Altenschmidt ... 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2020. Slotrefine: A fast non-autoregressive model for joint intent detection and slot filling. In *ACL*.

Pascal Sikorski, Leendert Schrader, Kaleb Yu, Lucy Billadeau, Jinka Meenakshi, Naveena Mutharasan, Flavio Esposito, Hadi AliAkbarpour, and Madi Babaiasl. 2024. Deployment of large language models to control mobile robots at the edge. *Preprint*, arXiv:2405.17670.

Stefanie Tellex, Ross A Knepper, Adrian Li, Daniela Rus, and Nicholas Roy. 2011. Understanding natural language commands for robotic navigation and mobile manipulation. In *AAAI*.

Gokhan Tur and Renato De Mori. 2011. Spoken language understanding: Systems for extracting semantic information from speech. *John Wiley & Sons*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, et al. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

John M Zelle and Raymond J Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.