Intent Recognition in Speech-to-Text Processing in the Context of Natural Interaction with Cognitive Assistive Systems

Behnam Ensan[1], **Magnus Jung**[2], **Matthias Busch**[3], **Andreas Wendemuth**[1]
[1]Cognitive Systems, Otto-von-Guericke-University, Magdeburg, Germany
[2]Neuro-Information Technology, Otto-von-Guericke-University, Magdeburg, Germany
[3]Mobile Dialog Systems, Otto-von-Guericke-University, Magdeburg, Germany
{behnam.ensan, magnus.jung, matthias.busch, andreas.wendemuth}@ovgu.de

Abstract

This study investigates efficient speech-to-intent recognition for human–robot interaction in elderly-care environments in German, targeting deployment on resource-constrained platforms such as the Jetson AGX Orin. To benchmark performance, we created a domain-specific German dataset with two sub-datasets (PaSID and PaSynTex) that simulate specific nursing home communication scenarios. Two alternative speech-to-intent pipelines were developed and evaluated: a two-stage system combining automatic speech recognition (ASR) with a large language model (LLM), and an end-to-end large audio–language model (LALM) architecture. The performance of Whisper-based ASR systems was evaluated across a wide variety of LLMs and several LALMs, comparing intent-classification accuracy, latency, and resource efficiency. The results indicate that optimized ASR + LLM configurations, particularly Whisper Turbo coupled with Phi-3.5-mini or Qwen 2.5-7B, outperform unified LALM approaches while maintaining substantially lower memory and inference costs. Also, the analysis shows that, the unified LALM models outperform the two-step integration of ASR + LLM in the same configuration, but at the cost of higher resource utilization, likely due to limited optimization for edge deployment. Overall, the findings provide initial evidence that modular ASR + LLM pipelines provide a more practical solution for real-time, on-device intent recognition in assistive robotics in German, offering an effective trade-off between performance and deployability on resource-constrained platforms.

Keywords: speech-to-intent recognition, intent recognition, spoken language understanding (SLU), large language models (LLMs), large audio-language models (LALMs), edge AI deployment, assistive robotics, elderly care

1. Introduction

Large language models (LLMs), such as GPT (Open AI, 2023), have shown strong performance in natural language understanding (NLU), dialogue management, and generation (NLG). By fine-tuning or prompting these models for specific tasks, researchers can achieve end-to-end dialogue pipelines that reduce the need for domain-specific rules and heuristics (Madotto et al., 2020). Unlike traditional systems that separate intent classification and dialog management, using LLM-based frameworks we could unify these components, allowing more fluid, context-aware interactions (Minaee et al., 2024; Zhou et al., 2023). In numerous domains, from manufacturing and logistics to healthcare, the integration of robots into everyday tasks promises efficiency gains, cost savings, and improved service quality (Chiang and Trimi, 2020; Karabegović et al., 2015). However, these benefits cannot be realized without addressing the critical need for robots to understand humans and respond intuitively to complex cues.

A major challenge is ther recognition of intent from spoken input in real-time, especially under the computational constraints of embedded platforms like Jetson (NVIDIA Corporation, 2025b). Although modern LLMs and large audio-language models (LALMs) offer high accuracy, they are typically optimized for cloud-based inference, limiting their direct application in edge scenarios. This work addresses this gap by designing and evaluating optimized speech-to-intent pipelines suitable for real-time patient–robot interaction in elderly-care environments in Germany and deployable on edge hardware. This involves creating domain-specific datasets that simulate nursing home interactions in German and benchmarking automatic speech recognition (ASR) systems. The study also evaluates LLMs for intent detection with a focus on accuracy, latency, and resource efficiency. Integrated ASR+LLM pipelines are developed and optimized for edge deployment, and their performance is compared with recent end-to-end LALMs to identify the most effective and efficient solutions for real-time, on-device intent recognition.

In summary, this paper addresses the challenge of real-time intent recognition from spoken input in elderly-care robotics, with a focus on resource-efficient edge deployment. By developing and evaluating both modular ASR+LLM pipelines and end-to-end LALM pipelines, we examine the trade-offs among accuracy, latency, and resource demands. Through the creation of two domain-specific German datasets, we aim to provide an initial benchmark for speech-to-intent processing in this setting. The remainder of this paper presents the datasets, methodology, and model architectures, followed by an empirical evaluation of their performance and a

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 10102–10113
11–16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

discussion of their suitability for real-world assistive applications.

2. Background

Human-Robot Interaction (HRI) is becoming more relevant in healthcare, where robots support staff and improve patient care (Abdi et al., 2018). In nursing homes, socially assistive robots have shown promise in emotional support, cognitive stimulation, and reducing staff workload (Worth, 2024). In this context, dialogue systems play a central role in enabling effective communication between humans and robots.

2.1. Dialogue System in HRI

Modern dialogue systems have evolved from rule-based to neural end-to-end architectures, now powered by LLMs. These systems integrate key components such as ASR, NLU, and NLG, allowing more fluid and human-like conversations (Yi et al., 2024; Chung et al., 2023). Building on this foundation, adapter-based multi-modal LLMs introduce a modular approach to integrating various modalities (Li et al., 2024). Instead of training a single, monolithic model from scratch, adapters act as lightweight transformation layers, allowing a pre-trained LLM to incorporate additional input types, such as speech signals. This approach preserves the strengths of the base LLM while enabling it to process and align diverse modalities efficiently. In this context, **Ultravox** (Fixie AI, 2025) stands out as an innovative open-source adapter-based LALM that improves dialogue systems by directly processing speech input without the need for intermediate text conversion.

Unlike adapter-based models, which rely on separate speech encoders and fusion adapters, end-to-end LALMs jointly train a single model on both speech and text data. For example, **Qwen-Audio** (Chu et al., 2023), developed by Alibaba Cloud, is an end-to-end trained large audio-language model, designed to process a variety of audio inputs.

2.2. Intent Recognition

Intent recognition is the process of inferring an agent's goals by analyzing their behavior (Smith et al., 2022). This capability is crucial for the development of dialogue systems. The rise of pre-trained language models, such as BERT (Devlin et al., 2019), has improved intent classification by capturing complex semantic relationships within the language (Chen et al., 1902). Fine-tuning these models on domain-specific data has led to notable improvements in performance. For instance, the TOD-BERT model, pre-trained in task-oriented dialogues, has demonstrated improved performance in intent recognition tasks (Wu et al., 2020).

Recent studies have explored the application of LLMs for intent recognition. For example, the paper IntentGPT (Rodriguez et al., 2024) introduces a method that effectively prompts LLMs, such as GPT, to discover new intentions with minimal labeled data.

Furthermore, the study "User Intent Recognition and Satisfaction with Large Language Models" (Bodonhelyi et al., 2024) analyzes the quality of intent recognition and user satisfaction with answers from intent-based prompt reformulations of GPT-3.5 Turbo and GPT-4 Turbo models.

2.3. NVIDIA Jetson for HRI Applications

The NVIDIA Jetson platform (NVIDIA Corporation, 2025b) is a leading edge AI solution for resource-intensive tasks. Benchmark studies (NVIDIA Corporation, 2025, 2024) demonstrate its efficiency in running optimized ASR and LLM models using quantization and inference acceleration frameworks like TensorRT.

An empirical benchmark conducted by NVIDIA Jetson AI Lab (NVIDIA Corporation, 2024), utilizing MLC/TVM framework and 4-bit quantization, provided insights into the throughput performance of Jetson hardware. As shown in Table 1, the AGX Orin achieves between 40–150 tokens per second (TPS) for models ranging from approximately 1 billion to 8 billion parameters. Even running larger variants (such as 33B or 70B models) remains technically feasible, although at lower throughput levels (approximately 10 TPS and 5 TPS, respectively).

3. Multimodal Dataset for Evaluating Intent Detection in Elderly Care

To support the evaluation of intent detection in elderly-care scenarios, we created a domain-specific German dataset focused on patient–robot interaction. The dataset comprises two sub-datasets designed for different evaluation tasks and captures both spoken and written forms of communication. In addition, the data include three forms of expression with varying levels of linguistic complexity and implicitness. The following datasets were used in the analysis:

- **PaSID** (Patient Speech for Intent Detection): a dataset comprising one hour of German speech and 348 samples based on predefined scenarios simulating interactions in a nursing-home setting. Each sample is annotated with the corresponding intent and slot labels.
- **PaSynTex** (Patient Synthetic Text Expressions): a dataset of 300 synthetic German sentences, ranging from simple to implicit formulations, for intent classification and slot extraction.

The PaSID dataset was collected from ten student participants between 26 and 35 years of age.

Model	Jetson Orin Nano (TPS)	Jetson AGX Orin (TPS)
Phi-2 (2.7B) (Javaheripi et al., 2023)	24	74
Gemma-2B (Team et al., 2024)	27	75
Llama2-7B (Touvron et al., 2023b)	16	47
Llama3-8B (Grattafiori et al., 2024)	15	40
Llama2-13B (Touvron et al., 2023b)	-	25
Llama1-33B (Touvron et al., 2023a)	-	10
Llama2-70B (Touvron et al., 2023b)	-	5

Table 1: LLM Text-generation throughput (4-bit Quantization with MLC/TVM) (NVIDIA Corporation, 2024)

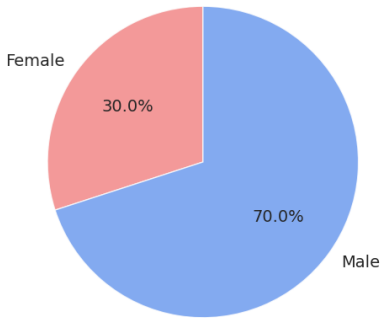

Figure 1: Gender distribution of participants

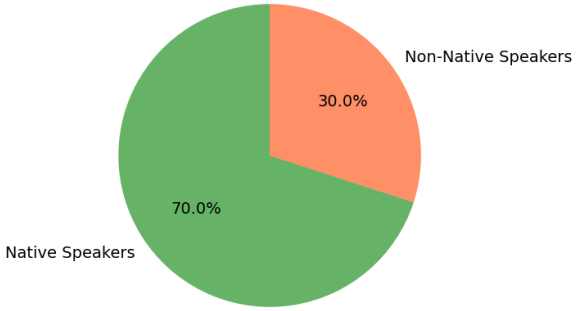

Figure 2: Language-background distribution of participants

Figures 1 and 2 illustrate the demographic and language-background distributions of the participants.

The PaSynTex dataset, consisting of 300 samples and four primary intents, was generated using ChatGPT-4o and manually reviewed. Both datasets were designed to approximate daily patient–robot interactions in elderly-care scenarios and to test the robustness of ASR systems, LLMs, and LALMs across different levels of linguistic complexity. Table 3 outlines the structure of the PaSID dataset.

To support the collection of structured, high-quality speech data, the **Patient–Robot Interaction**[1] survey was developed and implemented using the online platform **SoSci Survey** (SoSci Survey GmbH, 2023). During the audio-recording sessions, participants were instructed to use high-quality microphones or headsets to maximize audio clarity and minimize background noise. They were also given the opportunity to review and, if necessary, re-record their audio, which further improved the quality and consistency of the collected data.

The data-collection process was structured into three phases to capture a range of natural and controlled speech variations and to examine their effects on ASR performance. In the first phase, participants produced freely formulated speech, providing relatively natural and spontaneous input. In the second phase, they read pre-formulated texts describing specific scenarios in an elderly-care setting. In the final phase, speech was recorded in different modalities (e.g., speaking quickly in an emergency-like situation or slowly to ensure that each word was clearly understood). These samples were based on pre-formulated texts read aloud by the participants and were used to study the impact of different speaking styles on ASR performance.

We classified the dataset intents into four main categories: Assistance Request, Medication Reminder, Information Query, and Emergency Alert.

4. Methodology

This study evaluates two dialogue system pipelines for intent recognition from spoken input in patient–robot interaction. The first pipeline integrates ASR with an LLM for comparative performance analysis. The second pipeline uses LALMs as a unified end-to-end solution and is evaluated comparatively in terms of intent-detection performance, inference latency, and resource efficiency on edge hardware.

4.1. Modular ASR and LLM-Based Pipeline

A two-step pipeline (illustrated in Figure 3) is implemented in which speech input is first transcribed using selected ASR models (e.g., Whisper). The transcribed text is then processed by LLMs in the natural language analysis system (NLAS) component. Guided by prompt engineering, the LLMs classify user intent.

[1]The full survey is available at: https://befragungen.ovgu.de/patient-robot-interaction/.

Speech	Speech Text Reference (German)	English Translation	Intent	Slots
(Audio)	"Könnten Sie meine Brille suchen?"	"Could you look for my glasses?"	Assistance Request	object: Brille
(Audio)	"Erinnern Sie mich bitte daran, um 9 Uhr mein Insulin zu nehmen."	"Please remind me to take my insulin at 9 o'clock."	Medication Reminder	time: 9 Uhr, medication: Insulin
(Audio)	"Was steht heute auf dem Speiseplan zum Mittagessen?"	"What's on the menu for lunch today?"	Information Query	
(Audio)	"Mir ist sehr schwindelig."	"I feel very dizzy."	Emergency Alert	

Table 2: Example entries from the PaSID dataset

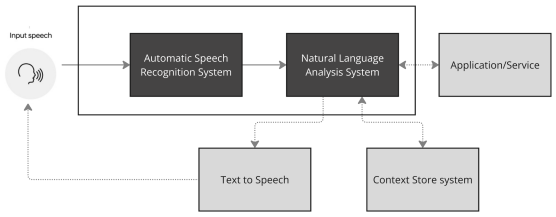

Figure 3: Dialogue system based on modular ASR and LLM integration

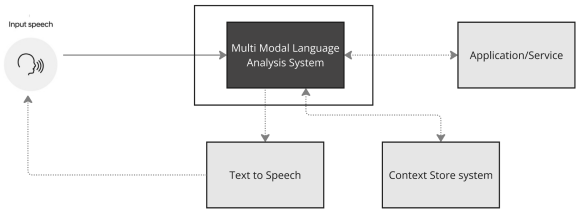

Figure 4: Dialogue system based on LALM

4.2. Large Audio-Language Model Pipeline

The multi-modal language analysis system component, illustrated in Figure 4, is designed to process spoken input directly for downstream tasks such as intent detection, thereby bypassing the explicit text-transcription step used in traditional ASR+LLM pipelines. Two LALM architectures are considered in this study:

- **Adapter-based models**: These models use a modular architecture that combines a frozen or fine-tuned ASR component with an LLM through lightweight adapter modules (e.g., Ultravox (Fixie AI, 2025)). The adapters act as transformation layers that align and fuse audio embeddings with text-based representations, enabling the LLM to process spoken-language input more effectively. This architecture allows flexible integration of pre-trained models while minimizing computational overhead and retraining requirements.

- **End-to-end trained models**: In contrast, these models are trained jointly on audio and text modalities using unified tokenization schemes and multimodal transformer encoders (e.g., Qwen-Audio (Chu et al., 2023)). They directly process raw audio input and perform intent classification and text generation within a single unified model. This approach removes intermediate processing steps and may offer advantages for real-time applications through lower processing latency and improved alignment between input and output modalities.

By comparing these two architectures, we examine the trade-offs among modular flexibility, end-to-end efficiency, and suitability for edge deployment.

4.3. Prompt Design

In this study, a structured zero-shot prompt template was used to guide LLMs and LALMs in classifying utterances into four intent categories: *Assistance_Request, Medication_Reminder, Information_Query, Emergency_Alert*. Using the same prompt across models supports a consistent evaluation setup and a fair comparison of their intent-classification performance.

Prompt

```
You are an intent classification
assistant.
You should only output one of
the following intents: As-
sistance_Request, Medica-
tion_Reminder, Information_Query,
Emergency_Alert.


Here is a short description of
each intent:
- Assistance_Request: The user
is seeking help with a non-
emergency task or activity.
- Medication_Reminder: The user
needs a reminder to take medica-
tion.
- Information_Query: The user
is asking for information or de-
tails.
- Emergency_Alert: The user is
experiencing distress or an ur-
gent situation requiring immedi-
ate attention.
Intent:
```

4.4. Edge-Device-Oriented Models

To reduce memory requirements and improve inference speed, all selected models were quantized to **4-bit precision**. Model sizes were systematically benchmarked and compared with recent edge-performance results.

The following key metrics were used to evaluate edge suitability:

- **Inference latency** (average processing time per input)
- **Tokens per second (TPS)** for generation-based models
- **Peak GPU memory usage** during inference
- **Intent classification accuracy**

Together, these metrics provide a practical basis for assessing whether a model is not only accurate, but also lightweight and fast enough for real-time deployment in embedded assistive systems.

4.5. Evaluation Metrics

The following evaluation metrics are used to compare the modular ASR+LLM pipeline and the unified LALM-based system.

For the ASR component, transcription quality is evaluated using *Word Error Rate (WER)* and BERT sentence similarity score (He et al., 2021). WER is defined as:

$$\text{WER} = \frac{S + D + I}{N_w} \tag{1}$$

where S is the number of substitutions, D the number of deletions, I the number of insertions, and N_w the total number of words in the reference transcription. To complement surface-level transcription accuracy, semantic similarity between the ASR output and the reference transcription is also measured using the BERT sentence similarity score. In addition, average inference time is recorded to assess ASR latency.

For the LLM and LALM components, intent classification is evaluated using *accuracy* and *F1-score*. Accuracy is defined as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Predictions}} \tag{2}$$

The F1-score is computed as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

where TP, FP, and FN denote the numbers of true positives, false positives, and false negatives, respectively.

To assess computational efficiency, inference latency (the inverse of the Real-Time Factor (RTF)), and throughput in *tokens per second (TPS)* are measured. TPS is defined as:

$$\text{TPS} = \frac{\text{Number of Output Tokens}}{\text{Inference Time (s)}} \tag{6}$$

In addition, maximum GPU memory usage during inference is recorded to assess the resource requirements of each model.

5. Experimental Setup

This section describes the experimental setup, model selection, and datasets used to evaluate speech-to-intent pipelines for patient–robot interaction in German.

5.1. Experimental Hardware

The experiments were conducted on an NVIDIA A100 80GB Tensor Core GPU (NVIDIA Corporation, 2025a). Although this hardware enables large-scale evaluation, the proposed pipelines are designed for deployment on resource-constrained edge devices such as the NVIDIA Jetson platform. To account for this difference, the analysis considers performance trade-offs between server-grade and edge hardware.

5.2. Model Selection

ASR: For ASR, a range of OpenAI Whisper models, from Tiny to Large-v3 (OpenAI, 2022), was evaluated because of their strong performance on ASR benchmarks and the availability of optimized open-source implementations and deployment frameworks, such as Faster-Whisper and WhisperTRT. Whisper models have shown strong performance in German speech transcription on Common Voice 15 (OpenAI, 2024) and Common Voice German (Meta AI, 2023). The Faster-Whisper implementation (SYSTRAN, 2025) was used for optimized inference. WhisperTRT (NVIDIA Corporation, 2025) is also a promising solution for edge deployment, but it lacks multilingual support.

LLMs: We evaluated a broad range of instruction-tuned models, spanning 0.5B to 90B parameters, on the task of intent recognition. Model selection was guided by multilingual support, efficiency on edge devices, and performance on established benchmarks (MMLU, GPQA, and MMLU-Pro) (Hendrycks, 2020; Wang et al., 2024; Rein et al., 2023). Benchmark information was collected from multiple sources, primarily official model reports and the ArtificialAnalysis.ai platform (Artificial Analysis, 2025), to make the comparisons as consistent as possible. Some variability may nev-

ertheless arise from differences in prompt formulation (Klu.ai, 2025) and option ordering in multiple-choice evaluations (Pezeshkpour and Hruschka, 2023; Zheng et al., 2024). However, these differences are minimal and do not affect the overall ranking of the models in the benchmark.

This study uses the **MLC** (Machine Learning Compilation) optimization framework (Machine Learning Compilation, 2023) for implementation and performance analysis, helping align the evaluation setup with NVIDIA Jetson benchmark practices (NVIDIA Corporation, 2024).

LALMs: The availability of LALMs adapted to German remains limited, although some existing models show potential for effectiveness in this context. End-to-end trained and adapter-based models LALMs, such as Qwen2-Audio and Ultravox (Chu et al., 2023; Fixie AI, 2025), were evaluated for direct speech-to-intent classification.

6. Results and Discussion

The results are presented for the ASR, NLAS, and end-to-end LALM components of the evaluated pipelines.

6.1. Evaluation of Automatic Speech Recognition System

The first part of the analysis focuses on evaluating various ASR models using the PaSID data. The models are assessed on multiple criteria.

As shown in Figure 5, larger ASR models generally achieve lower WER values. In particular, Whisper Turbo maintains low WER with more than 30% lower memory usage than larger models, making it a promising option for edge deployment.

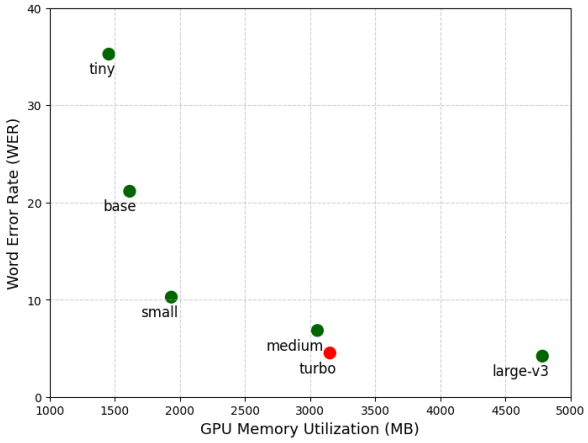

Figure 5: WER vs. peak GPU memory usage while inference for Faster-Whisper models on the PaSID dataset (A100 GPU)

Complementing this, semantic similarity results shown in Figure 6, measured using DeBERTa embeddings (He et al., 2021), indicate that both Whisper-turbo and Whisper-large-v3 achieve near-perfect semantic accuracy despite differences in model size and complexity.

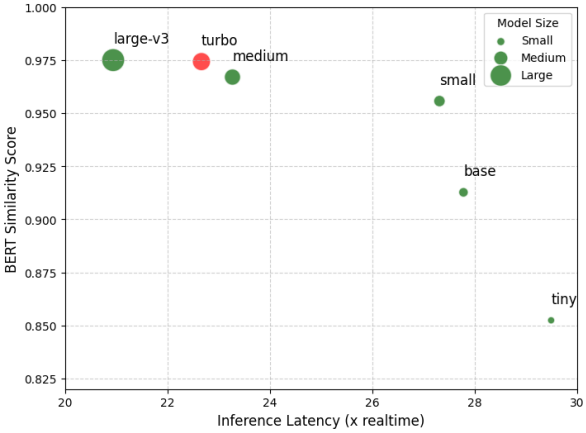

Figure 6: BERT similarity vs. latency (x realtime) for Faster-Whisper models on the PaSID dataset (A100 GPU)

6.2. Evaluation of Natural Language Analysis System

A variety of LLMs were evaluated on the developed PaSynTex dataset. Using 4-bit quantization and five iterations for robustness, model performance was measured by intent classification accuracy, end-to-end latency, and hardware efficiency.

Figure 7 shows a general correlation between model size and accuracy, with diminishing returns for models with 14B parameters or more. Phi-3.5-mini (Microsoft Azure-AI Services, 2024) again stands out, achieving top performance at a low resource cost.

A comparison of text-generation throughput between the experimental A100 system and the Jetson AGX Orin (NVIDIA Corporation, 2024), reveals a performance gap of 3 to 4 times, presented in

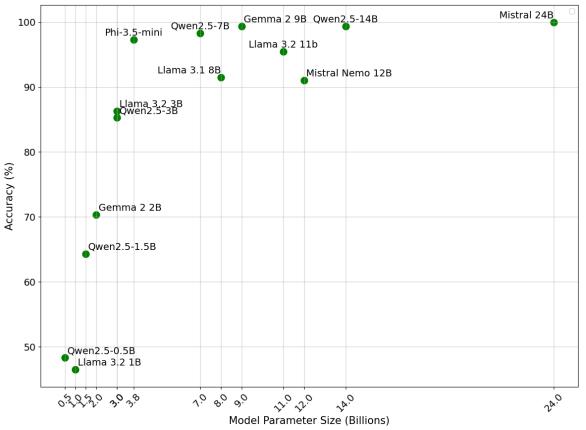

Figure 7: LLM accuracy vs. parameter size on the PaSynTex dataset

Figure 8. For example, Gemma 2B generates 282 tokens/sec on A100 versus 75 on Orin; Llama 2 7B drops from 154 to 47 tokens/sec. Although Jetson AGX Orin can technically run large models such as Llama 70B, the generation of 40 tokens takes over 8 seconds, making real-time performance impractical without further optimization.

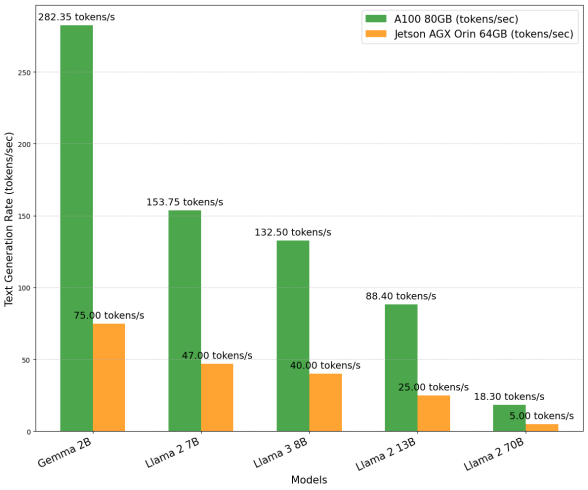

Figure 8: Text-generation throughput on NVIDIA A100 80GB vs. Jetson AGX Orin 64GB (4-bit quantization, MLC)

For assistive healthcare applications, language models must maintain high accuracy across all intent classes, especially for critical categories like *Emergency Alert* and *Medication Reminder*. Figure 9 illustrates that models such as Qwen2.5-14B and Gemma 2 9B consistently achieve near-perfect F1-scores across all classes. Phi-3.5-mini and Qwen2.5-7B also perform reliably while requiring fewer resources.

Although the top models excel in detecting emergencies, smaller models (e.g., Qwen2.5-3B, Llama 3.2 3B) show a reduced accuracy, which could be risky in real-world scenarios. Most confusion occurs between overlapping intent classes, such as *Information Query* and *Medication Reminder*, as detailed in Table 6.2.

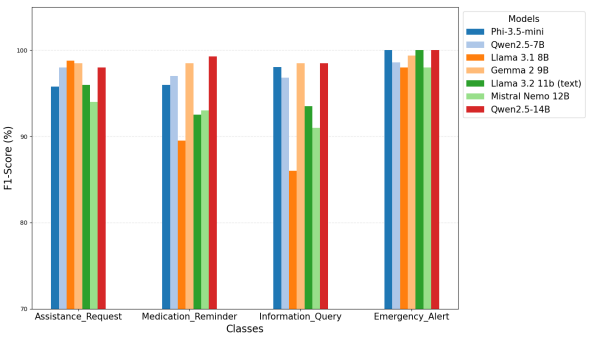

Figure 9: F1-score by intent class for LLMs on the PaSynTex dataset (4-bit quantization, 5 runs)

6.3. Evaluation of the End-to-End Pipelines

Figure 10 compares inference latency and intent-detection accuracy for end-to-end pipelines on the PaSID dataset. Turbo-Qwen2.5-14B and Llama 3.3 70B achieve the highest accuracy, but with high latencies (984 ms and 2178 ms, respectively), which limits their suitability for edge deployment. Models such as Turbo-Phi-3.5-mini and Turbo-Qwen2.5-7B provide a more favorable trade-off, with approximately 92-94% accuracy and moderate latency. Among LALMs, Ultravox-Turbo-Llama3.1-8B and Ultravox-Turbo-Mistral-Nemo-12B achieve strong results (90.9% and 88.5%, respectively) at the cost of longer inference times.

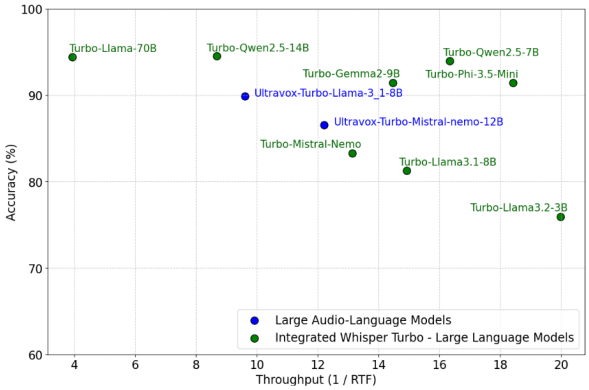

Figure 10: Intent-detection accuracy vs. inference latency (x realtime) for end-to-end pipelines on the PaSID dataset (A100 GPU)

Figure 11 compares peak GPU memory usage across the evaluated end-to-end pipelines. In general, LALMs require approximately 60% more memory than comparable two-step systems because of their unified architecture. In contrast, integrated ASR+LLM pipelines benefit from quantized and optimized implementations (4-bit LLMs and Faster-Whisper), which substantially reduce memory demands. Although LALMs achieve competitive accuracy, their limited quantization support currently reduces their efficiency for edge deployment.

Based on the analysis of the end-to-end pipelines, none of the evaluated systems achieved 100% accuracy on the PaSID dataset. Three main factors appear to contribute to the remaining errors.

First, ASR transcription errors sometimes led to a complete change in the intended meaning. For example, the utterance “Ich muss mich schnell übergeben” (“I need to vomit quickly,” indicating an emergency) was transcribed as “Ich muss das Spiel ergeben” (“I have to surrender the game”), which led to an incorrect classification as *Information Query* instead of *Emergency Alert*. This example highlights the importance of transcription quality for downstream intent recognition. Second, partici-

Sentence (German)	Sentence (English)	True Intent	Predicted Intent
Können Sie meine Brille suchen?	Can you help me find my glasses?	Assistance Request	Information Query
Bitte stellen Sie sicher, dass ich meine Augentropfen morgens und abends benutze.	Please make sure I use my eye drops in the morning and evening.	Medication Reminder	Information Query
Was steht auf meinem Medikamentenplan?	What is on my medication schedule?	Information Query	Medication Reminder

Table 3: Common intent confusions on the PaSynTex dataset

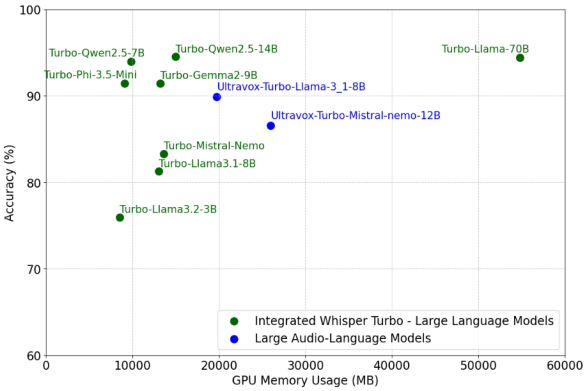

Figure 11: Intent-detection accuracy vs. peak GPU memory usage for end-to-end pipelines on the PaSID dataset (A100 GPU)

pants introduced errors who did not express their intention correctly. For instance, the sentence“Weißt du, wo meine Decke ist?” (“Do you know where my blanket is?”) was classified as an "Information Query" rather than the intended "Assistance Request", which was meant for the participant to express. Third, some utterances contained wording that shifted the perceived intent. For example, the sentence “Ich bin ausgerutscht. Könnten Sie mir bitte helfen, wieder aufzustehen?” (“I slipped. Could you please help me get up again?”) was in some cases classified as *Assistance Request* rather than the intended *Emergency Alert*. This likely reflects the indirect and polite formulation of distress, which may have led the system to interpret the utterance as a general request for help rather than as an urgent situation.

7. Conclusion

This study investigated intent detection in the context of speech to text processing for a defined elderly care scenario. It compares two pipelines: a two-step ASR+LLM system and an end-to-end Large Audio-Language Model (LALM) approach, with deployment targeted at resource-constrained hardware like the Jetson AGX Orin.

Datasets: Two custom datasets were created: PaSID for ASR and end-to-end pipeline evaluation, and PaSynTex for LLM evaluation, both targeting intent detection in realistic elderly-care scenarios. While these datasets provide an initial benchmark for this application setting, further validation on larger and more diverse datasets is needed to confirm the generalizability of the findings.

ASR Benchmark: Among the evaluated ASR models, **Whisper Large-v3**, and **Whisper Turbo** delivered top accuracy. **Whisper Turbo** offered the best balance of accuracy and resource efficiency within the tested setup.

LLM Benchmark: A diverse range of LLMs was assessed for intent classification. Smaller models like **Phi3.5-mini** and **Qwen2.5-7B** performed remarkably well, outperforming other models within the same size range and even some larger models. Slightly better performance was observed with mid-sized models like **Gemma 9B** and **Qwen2.5 14B**, although this came at the cost of increased latency.

2-Step Pipeline: ASR+LLM combinations using **Whisper Turbo** with **Phi3.5-mini** or **Qwen2.5-7B** achieved strong results while maintaining comparatively low resource usage. Within the scope of this study, these configurations seem better suited for edge deployment. Larger model combinations provided small performance gains, but these were accompanied by higher latency and GPU demands.

End-to-End LALMs: While LALMs like **Ultravox-Turbo-Llama3.1-8B** delivered competitive accuracy (up to 90%), they suggested 60% more GPU usage compared to the same configuration in the 2 step ASR+LLM integration systems, suggesting the latter as more efficient for edge scenarios.

8. Acknowledgement

This work was carried out within the project “Resilient human-robot collaboration in a mixed-skill environment (ENABLING)”, cofinanced by the European Regional Development Fund (ERDF), ZS/2023/12/182056 and the German Research Foundation (DFG) (SEMIAC under grant number No. 502483052).

9. Bibliographical References

Jordan Abdi, Ahmed Al-Hindawi, Tiffany Ng, and Marcela P Vizcaychipi. 2018. Scoping review on the use of socially assistive robot technology in elderly care. *BMJ open*, 8(2):e018815.

Artificial Analysis. 2025. Artificial analysis: Ai model & api providers analysis. Accessed : 2025/03/18.

Anna Bodonhelyi, Efe Bozkir, Shuo Yang, Enkelejda Kasneci, and Gjergji Kasneci. 2024. User intent recognition and satisfaction with large language models: A user study with chatgpt. *arXiv preprint arXiv:2402.02136*.

Q Chen, Z Zhuo, and W Wang. 1902. Bert for joint intent classification and slot filling. arxiv 2019. *arXiv preprint arXiv:1902.10909*.

Ai-Hsuan Chiang and Silvana Trimi. 2020. Impacts of service robots on service quality. *Service Business*, 14(3):439–459.

Willy Chung, Samuel Cahyawijaya, Bryan Wilie, Holy Lovenia, and Pascale Fung. 2023. Instructtods: Large language models for end-to-end task-oriented dialogue systems. *arXiv preprint arXiv:2310.08885*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

I Karabegović, E Karabegović, M Mahmić, and EJAIPE Husak. 2015. The application of service robots for logistics in manufacturing processes. *Advances in Production Engineering & Management*, 10(4).

Klu.ai. 2025. Mmlu benchmark (massive multitask language understanding). Accessed : 2025/03/18.

Hong Li, Zhiquan Tan, Xingyu Li, and Weiran Huang. 2024. Atlas: Adapter-based multi-modal continual learning with a two-stage learning strategy. *arXiv preprint arXiv:2410.10923*.

Machine Learning Compilation. 2023. Machine learning compilation (mlc) blog. `https://blog.mlc.ai/`. Accessed : 2025/03/18.

Andrea Madotto, Zihan Liu, Zhaojiang Lin, and Pascale Fung. 2020. Language models as few-shot learner for task-oriented dialogue systems. *arXiv preprint arXiv:2008.06239*.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey.

NVIDIA Corporation. 2024. Ai inferencing benchmarks for jetson orin nano super and jetson agx orin. Accessed : 2025/03/18.

NVIDIA Corporation. 2025a. *NVIDIA A100 Tensor Core GPU*. Accessed : 2025/03/18.

NVIDIA Corporation. 2025b. Nvidia jetson orin. Accessed : 2025/03/18.

Open AI. 2023. *Open AI Large language model*. Accessed : 2025/03/18.

Pouya Pezeshkpour and Estevam Hruschka. 2023. Large language models sensitivity to the order of options in multiple-choice questions.

Qwen Team. 2024. Qwen2-audio: Chat with your voice! Accessed : 2025/03/18.

Juan A Rodriguez, Nicholas Botzer, David Vazquez, Christopher Pal, Marco Pedersoli, and Issam Laradji. 2024. Intentgpt: Few-shot intent discovery with large language models. *arXiv preprint arXiv:2411.10670*.

Gary B Smith, Vaishak Belle, and Ronald PA Petrick. 2022. Intention recognition with problog. *Frontiers in Artificial Intelligence*, 5:806262.

Tammy Worth. 2024. Are robots the solution to the crisis in older-person care? *Nature*.

Chien-Sheng Wu, Steven Hoi, Richard Socher, and Caiming Xiong. 2020. Tod-bert: Pre-trained natural language understanding for task-oriented dialogue. *arXiv preprint arXiv:2004.06871*.

Zihao Yi, Jiarui Ouyang, Yuwen Liu, Tianhao Liao, Zhe Xu, and Ying Shen. 2024. A survey on recent advances in llm-based multi-turn dialogue systems. *arXiv preprint arXiv:2402.18013*.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large language models are not robust multiple choice selectors.

Xinyu Zhou, Delong Chen, and Yudong Chen. 2023. Towards joint modeling of dialogue response and speech synthesis based on large language model. *arXiv preprint arXiv:2309.11000*.

10. Language Resource References

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. 2023. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*.

Fixie AI. 2025. Ultravox: Build ai voice agents that communicate like we do. Accessed : 2025/03/18.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Dan Hendrycks. 2020. Mmlu: Measuring massive multitask language understanding. Accessed : 2025/03/18.

Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, et al. 2023. Phi-2: The surprising power of small language models. *Microsoft Research Blog*, 1(3):3.

Meta AI. 2023. Speech recognition on common voice german. Accessed : 2025/03/18.

Microsoft Azure-AI Services. 2024. Discover the new multi-lingual, high-quality phi-3.5 slms. Accessed : 2025/03/18.

NVIDIA Corporation. 2025. Whispertrt: Optimizing openai whisper with nvidia tensorrt. Accessed : 2025/03/18.

OpenAI. 2022. Openai whisper github repository. Accessed : 2025/03/18.

OpenAI. 2024. Whisper benchmark on common voice. Accessed : 2025/03/18.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*. Accessed : 2025/03/18.

SoSci Survey GmbH. 2023. *SoSci Survey Benutzerhandbuch (Data Collection Platform)*. Accessed : 2025/02/15.

SYSTRAN. 2025. Faster-whisper transcription with ctranslate2. Accessed : 2025/03/18.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.

11. Appendices

11.1. Details of Speech Data Collection

As described above, the data collection process was divided into three phases. Participants received detailed instructions to clarify the expected responses and to encourage natural utterance formulation. Figure 12 shows the introduction to the first phase of the speech-data collection process, while Figure 13 presents an example scenario from this phase.

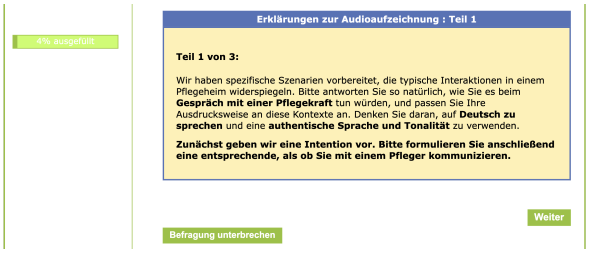

Figure 12: Instructions for participant speech-data collection in Phase 1

11.2. Additional ASR Model Analysis

Figure 14 shows WER as a function of ASR model size (in millions of parameters). For clarity, Whisper Large and Large-v2 are omitted because they share the same parameter size.

Wav2Vec2-Tevr (fine-tuned for German speech recognition) is used as a reference model due to its strong performance on the Common Voice German dataset (Meta AI, 2023). The Figure 14 shows that Whisper Large-v3 Turbo achieves performance comparable to Whisper Large-v3 and Wav2Vec2-Tevr on the PaSID dataset.

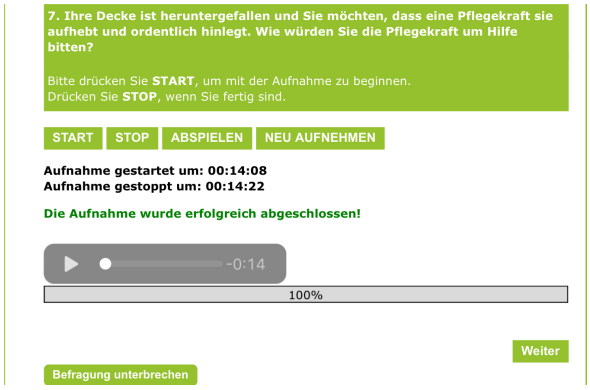

Figure 13: Example scenario from Phase 1 of the speech-data collection process

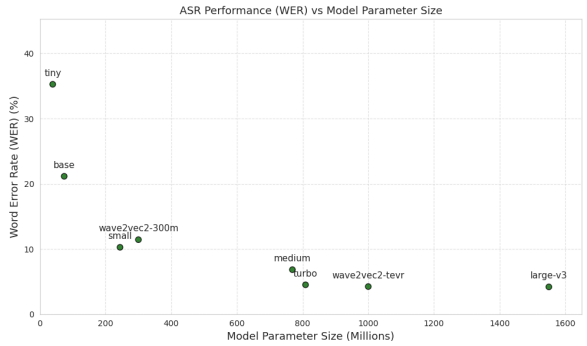

Figure 14: ASR performance (WER) vs. parameter size

11.3. Benchmark Rankings of Evaluated LLMs

Table 11.3 compares the rank positions of the evaluated LLMs across MMLU, MMLU-Pro, GPQA, and the zero-shot intent-classification experiment.

Model	MMLU	MMLU-Pro	GPQA	Experiment
Qwen2.5-0.5B	20	15	14	20
Llama 3.2 1B	19	14	17	19
Qwen2.5-1.5B	17	13	15	17
Gemma 2 2B	18	-	-	18
Llama 3.2 3B	16	10	16	15
Qwen2.5-3B	13	11	11	16
Phi-3.5-mini	6	6	-	9
Gemma 7B	15	12	-	14
Qwen2.5-7B	8	8	6	8
Ministral 8B	14	-	10	13
Llama 3.1 8B	11	7	12	11
Gemma 2 9B	10	9	9	6
Llama 3.2 11B	9	-	13	10
Mistral Nemo 12B	12	-	7	12
Qwen2.5-14B	5	3	8	6
Mistral 24B	4	5	3	1
Gemma 2 27B	7	4	5	1
Qwen2.5-32B	3	1	2	1
Llama 3.3 70B	1	2	1	1
Llama 3.2 90B	1	-	4	1

Table 4: Rank positions of models across external benchmarks and the experiment

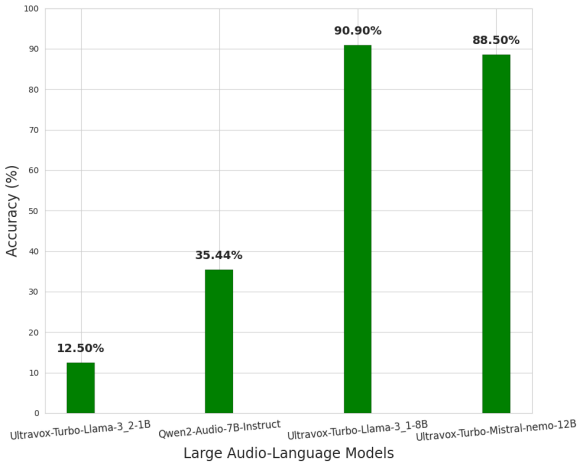

Figure 15: Intent Detection Performance of Large Audio Language Models

Larger models generally rank higher on MMLU, MMLU-Pro, and GPQA, which is consistent with their stronger general knowledge and reasoning capacity. In contrast, the intent-classification task used in this study is narrower in scope, resulting in smaller performance differences between medium-sized and large models. As a consequence, compact models can remain competitive in this domain-specific setting.

These results should be interpreted with caution, as some models have missing benchmark values and the task-specific dataset is less discriminative than the general benchmarks.

11.4. Additional Results for Large Audio–Language Models

Figure 15 depicts the intent detection accuracy of the large audio-language models (LALM) in the PaSID dataset.

The comparatively lower performance of Qwen2-Audio-7B-Instruct on the German intent-detection task may be explained by both its training-data composition and its architectural design. Although the model supports more than eight languages, including German, its training data primarily centered on Chinese and English, with more limited coverage of German. This imbalance may have led to a less robust understanding of German linguistic patterns. (Qwen Team, 2024).

11.5. Dataset

This section provides example entries from the dataset, illustrating the utterances, intent labels, and slot annotations (Table 11.5). In this study the performance of the systems were evaluated only on intent classification.

Sentence	Label	Slots
Könnten Sie mir helfen, meine Schuhe anzuziehen?	Assistance_Request	task: Schuhe anziehen; object: Schuhe
Ich brauche Unterstützung beim Gehen zum Speisesaal.	Assistance_Request	task: Gehen; object: None
Bitte helfen Sie mir, die Fernbedienung zu finden.	Assistance_Request	task: finden; object: Fernbedienung
Können Sie mir beim Anziehen meines Pullovers helfen?	Assistance_Request	task: Anziehen; object: Pullover
Ich kann meine Wasserflasche nicht finden, können Sie mir helfen?	Assistance_Request	task: finden; object: Wasserflasche
Könnten Sie mir bitte beim Aufstehen helfen?	Assistance_Request	task: Aufstehen; object: None
Ich brauche Hilfe, um meinen Rollator zu finden.	Assistance_Request	task: finden; object: Rollator
Bitte helfen Sie mir, mich hinzusetzen.	Assistance_Request	task: hinsetzen; object: None
Können Sie meine Brille suchen?	Assistance_Request	task: suchen; object: Brille
Helfen Sie mir bitte, ins Bett zu gehen.	Assistance_Request	task: ins Bett gehen; object: None
Erinnern Sie mich bitte an meine Blutdrucktabletten um 8 Uhr morgens.	Medication_Reminder	medication: Blutdrucktabletten; time: um 8 Uhr morgens
Ich muss vor dem Frühstück meine Schilddrüsentabletten nehmen.	Medication_Reminder	medication: Schilddrüsentabletten; time: vor dem Frühstück
Bitte erinnern Sie mich an meine Schmerzmittel um 15 Uhr.	Medication_Reminder	medication: Schmerzmittel; time: um 15 Uhr
Ich darf meine Vitaminpillen nicht vergessen, können Sie mich erinnern?	Medication_Reminder	medication: Vitaminpillen; time: None
Bitte geben Sie mir meine Schlafmittel vor dem Zubettgehen.	Medication_Reminder	medication: Schlafmittel; time: vor dem Zubettgehen
Erinnern Sie mich bitte daran, um 9 Uhr mein Insulin zu nehmen.	Medication_Reminder	medication: Insulin; time: um 9 Uhr
Ich muss nach dem Essen meine Herztabletten einnehmen.	Medication_Reminder	medication: Herztabletten; time: nach dem Essen
Bitte geben Sie mir vor dem Schlafengehen meine Medikamente.	Medication_Reminder	medication: Medikamente; time: vor dem Schlafengehen
Vergessen Sie nicht, dass ich um 14 Uhr meine Antibiotika brauche.	Medication_Reminder	medication: Antibiotika; time: um 14 Uhr
Können Sie mich alle vier Stunden an meine Augentropfen erinnern?	Medication_Reminder	medication: Augentropfen; time: alle vier Stunden; repetition: True
Wann kommt mein Physiotherapeut heute?	Information_Query	topic: Physiotherapie-Termin
Gibt es heute Besuchszeiten für Angehörige?	Information_Query	topic: Besuchszeiten
Was steht heute auf dem Speiseplan zum Mittagessen?	Information_Query	topic: Speiseplan
Kann ich heute an der Gymnastikstunde teilnehmen?	Information_Query	topic: Gymnastikstunde
Wie spät ist es jetzt?	Information_Query	topic: Uhrzeit
Wann ist mein nächster Arzttermin?	Information_Query	topic: nächster Arzttermin
Wie wird heute das Wetter?	Information_Query	topic: Wetter
Gibt es Neuigkeiten von meiner Familie?	Information_Query	topic: Familiennachrichten
Was gibt es heute zum Abendessen?	Information_Query	topic: Abendessen
Ich bin ausgerutscht und kann nicht aufstehen.	Emergency_Alert	type: Sturz
Mein Kopf tut sehr weh, bitte rufen Sie Hilfe!	Emergency_Alert	type: starke Kopfschmerzen
Ich habe plötzlich Schmerzen in meinem Bein!	Emergency_Alert	type: Beinschmerzen
Ich fühle mich benommen und kann kaum stehen.	Emergency_Alert	type: Benommenheit
Ich bekomme kaum Luft, bitte helfen Sie!	Emergency_Alert	type: Atemnot
Mir ist sehr schwindelig.	Emergency_Alert	type: Schwindel
Mein Herz schlägt sehr schnell, ich fühle mich nicht gut.	Emergency_Alert	type: Herzrasen
Ich habe starke Schmerzen im Brustbereich!	Emergency_Alert	type: Schmerzen im Brustbereich
Könnten Sie mir helfen, meine Schuhe zu binden?	Assistance_Request	task: Schuhe binden; object: Schuhe
Bitte reichen Sie mir die Fernbedienung.	Assistance_Request	task: reichen; object: Fernbedienung
Ich brauche Hilfe, um aus dem Bett zu kommen.	Assistance_Request	task: aufstehen; object: None
Können Sie das Licht für mich ausschalten?	Assistance_Request	task: Licht ausschalten; object: Licht
Bitte helfen Sie mir, meine Jacke auszuziehen.	Assistance_Request	task: Jacke ausziehen; object: Jacke
Bitte erinnern Sie mich an meine Medikamente um 8 Uhr.	Medication_Reminder	medication: Medikamente; time: 8 Uhr
Vergessen Sie nicht, mich an meine Tabletten vor dem Schlafengehen zu erinnern.	Medication_Reminder	medication: Tabletten; time: vor dem Schlafengehen
Ich muss meine Schmerztabletten nach dem Essen nehmen. Bitte erinnern Sie mich.	Medication_Reminder	medication: Schmerztabletten; time: nach dem Essen
Bitte sagen Sie mir Bescheid, wenn es Zeit für meine Tropfen ist.	Medication_Reminder	medication: Tropfen; time: None
Denken Sie daran, dass ich meine Herztabletten um 10 Uhr nehmen muss.	Medication_Reminder	medication: Herztabletten; time: 10 Uhr
Welches Datum haben wir heute?	Information_Query	topic: aktuelles Datum
Ich würde gerne wissen, was es heute zum Mittagessen gibt.	Information_Query	topic: Mittagessen
Wann ist mein nächster Besuch beim Arzt?	Information_Query	topic: nächster Arztbesuch
Können Sie mir sagen, wie das Wetter heute ist?	Information_Query	topic: Wetter
Ich möchte wissen, wann mein Physiotherapeut kommt.	Information_Query	topic: Physiotherapie-Termin
Mir ist sehr schwindelig. Ich brauche sofort Hilfe!	Emergency_Alert	type: Schwindel
Ich habe Atemnot. Bitte holen Sie einen Arzt!	Emergency_Alert	type: Atemnot

Table 5: Example entries from the PaSynTex dataset