Large Language Model Data Generation for Enhanced Intent Recognition in German Speech

Theresa Pekarek Rosin and **Burak Can Kaplan** and **Stefan Wermter**
University of Hamburg - Knowledge Technology
Vogt-Koelln-Strasse 30, 22527 Hamburg - Germany
www.knowledge-technology.info
Correspondence: theresa.pekarek-rosin@uni-hamburg.de

Abstract

Intent recognition (IR) for speech commands is essential for artificial intelligence (AI) assistant systems; however, most existing approaches are limited to short commands and are predominantly developed for English. This paper addresses these limitations by focusing on IR from speech by elderly German speakers. We propose a novel approach that combines an adapted Whisper ASR model, fine-tuned on elderly German speech (SVC-de), with Transformer-based language models trained on synthetic text datasets generated by three well-known large language models (LLMs): LeoLM, Llama3, and ChatGPT. To evaluate the robustness of our approach, we generate synthetic speech with a text-to-speech model and conduct extensive cross-dataset testing. Our results show that synthetic LLM-generated data significantly boosts classification performance and robustness to different speaking styles and unseen vocabulary. Notably, we find that LeoLM, a smaller, domain-specific 13B LLM, surpasses the much larger ChatGPT (175B) in dataset quality for German intent recognition. Our approach demonstrates that generative AI can effectively bridge data gaps in low-resource domains. We provide detailed documentation of our data generation and training process to ensure transparency and reproducibility.

1 Introduction

Speech command recognition is essential for natural interaction with artificial agents in everyday life, especially for elderly or handicapped users (Fronemann et al., 2021). Commercial solutions have seen vast improvements and increased usage over the last few years. However, their cloud-based models not only struggle with speech from user groups that would benefit from a voice assistant the most (Moro-Velazquez et al., 2019; Ngueajio and Washington, 2022), but they also introduce privacy issues because the processing of speech input usually does not happen locally.

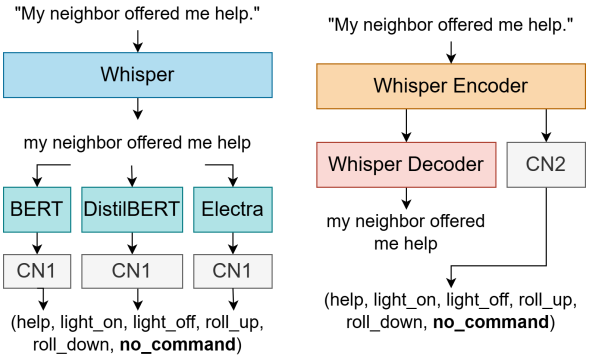

Figure 1: An overview of our model setup (left) and a traditional baseline (right). The speech is transcribed with a Whisper ASR model, the transcript is then classified with a transformer-based LM (BERT, DistilBERT, Electra) trained on a generated text dataset and a one-layer classification network (CN1). For the baseline, we use the output of the Whisper encoder to classify the intent with a two-layer classification network (CN2).

Additionally, traditional speech command recognition is usually restricted to short phrases and words (Warden, 2018), which requires the user to adapt to the system and change their way of speaking. That is usually not as intuitive for older people who might not have the same experience with recent technologies as younger users (Pekarek Rosin et al., 2025). Therefore, we argue that to create a more natural interaction with speech-based assistant systems, we need to move away from one-word command recognition to unconstrained utterance-based intent recognition.

However, the lack of large-scale datasets for languages besides English makes it difficult to implement such a change by retraining open-source speech models for intent recognition on different domains (e.g., elderly voices). Collecting new speech data on a larger scale for any speech recognition or classification task is not only an incredible effort for researchers and participants but also ultimately inefficient, since the process would need to

Proceedings of the 21st Conference on Natural Language Processing (KONVENS), Volume 1: Long and Short Papers, pages 349–359, 2025

be repeated every time new functionalities become available. This issue is amplified by the amount of training data that Transformer-based, state-of-the-art models require to be trained in their entirety.

We suggest two potential solutions to these issues: 1) through the use of layer-specific fine-tuning (Shor et al., 2019; Pekarek Rosin and Wermter, 2023) pretrained foundation models can be adapted to other domains with small amounts of speech data while preserving existing knowledge, 2) through the generation of additional domain-focused data with large language models (LLMs), we can increase the generalization abilities of the model to different linguistic patterns.

We combine a state-of-the-art automatic speech recognition (ASR) model, Whisper (Radford et al., 2023), with three transformer-based pretrained language models, BERT (Devlin et al., 2019), DistilBERT (Sanh et al., 2019), and Electra (Clark et al., 2020) for German speech intent recognition. We utilize three LLMs to generate additional training data for these language models, and we synthesize speech samples based on this data with a text-to-speech model to evaluate our approach.

2 Related Work

Speech command recognition typically involves accurately identifying words or short phrases as commands linked to functionalities in an underlying system. This classification can be performed on either the audio feature representations or the transcript produced by the ASR model. In contrast, intent recognition requires a larger context, such as complete sentences, to discern user intent. For English-language tasks, the Speech Command Dataset (Warden, 2018) is a commonly used resource for training and evaluating models in speech command recognition or keyword spotting.

Diverse strategies have been proposed for speech command recognition, with a recent approach by Sadovsky et al. (2023) exploring the use of spiking neural networks for this task, and achieving a maximum accuracy of 72% on the Speech Command Dataset. Other baselines mentioned in their paper reach 79% accuracy using Gated Recurrent Units.

Berg et al. (2021) propose the Keyword Transformer model. They train their model on the Speech Command Dataset for approximately 100 episodes and achieve between 97.49% and 98.56% accuracy using knowledge distillation with only minimal improvements over the baseline.

In low-resource settings, Kumar Nayak et al. (2023) explore speech command recognition in the Kui language using a small dataset of 7,090 utterances covering just seven words. Their convolutional neural network (CNN) baseline was the only model to exceed 90% accuracy after 300 training episodes, highlighting the challenges of limited data. Similarly, Hernández et al. (2024) investigate intent recognition in Spanish and Nahuatl by training models on a manually collected dataset of 383 natural language navigation commands. Their results show that transformer-based models outperform traditional baselines, especially for longer and more complex utterances.

Despite these advances, most approaches often rely on extensive training, large-scale datasets, or costly real-world data collection. Notably, the lack of recent work examining this research topic for German speech suggests that the absence of datasets deters research in that direction.

LLMs have emerged as powerful tools for synthetic dataset generation (Kaplan et al., 2025), enabling the creation of task-specific training data and offering a solution to data scarcity. Additionally, their capabilities in common-sense reasoning (Li et al., 2022) and their strong performance in natural language understanding make them well-suited for tasks such as intent recognition. Transformer-based language models have shown value in both data augmentation for intent classification (Kumar et al., 2020) and in intent classification itself (Chen et al., 2019).

Additionally, the use of more powerful LLMs for intent recognition has increased significantly (Dighe et al., 2024; Dzeparoska et al., 2024; Wang et al., 2024), and LLMs have been widely studied for their ability to assist people through multimodal applications with speech (Wagner et al., 2024; Padmanabha et al., 2024). However, the majority of the work has been conducted in English, leaving considerable room for exploration in other languages. While some recent work explores German-language applications of LLMs (Irrgang et al., 2024; Volk et al., 2024), they have primarily focused on other tasks. The release of German LLMs, such as LeoLM [1], has demonstrated the potential of LLMs for the German language and motivates our work, which explores the use of LLM-generated datasets to create robust intent recognition systems for German speakers.

[1] https://laion.ai/blog/leo-lm/

3 Methodology

3.1 Senior Voice Commands Dataset

The German Senior Voice Commands (SVC-de) is a dataset collected by Pekarek Rosin and Wermter (2023) for the development of speech-based interaction with a home assistant system for German senior citizens. The dataset contains recordings from 30 native German speakers (21 female, 9 male) between the ages of 50 and 99, of 52 sentence-based speech commands. Per speaker, approximately 6-7 minutes of audio from two different microphones is available, which leads to a total of 3 hours and 9 minutes of speech data. The recorded sentences can be separated into 6 classes: "help", "light_on", "light_off", "roll_up", "roll_down", and "no_command". The class "no_command" essentially serves to catch common false positives in command classification to decrease wrong triggers, while keeping the interaction natural without the involvement of wake-words or restriction of the user's speech.

3.2 Intent Recognition Dataset

For our intent recognition task, we generated sentences for the same six classes, with three different large language models (LLMs): LeoLM-13b[1], Llama3-8b[2], and ChatGPT by OpenAI. Llama3 is a well-known recent option for local LLMs, while LeoLM was developed specifically for the German language. We included ChatGPT as an assumed upper baseline, since it outperforms most LLMs on a large number of tasks, even though its lack of transparency makes results hard to reproduce. We generated approximately 2500 samples for each LLM.

3.2.1 Prompt Engineering

We tailored specific prompts for each label to maintain consistency and aimed to keep these prompts as similar as possible. Figure 2 illustrates a general example of the prompt structure. Every prompt, except for the "no_command" label, is comprised of four sentences, each targeting different sub-tasks. The initial sentence typically clearly outlines the scenario and initiates the task generation. The second sentence aims to enhance diversity by suggesting the LLM should incorporate various situations. The third sentence helps with the structuring of the outputs, making it easier to parse them for data collection. The parsing script is employed post-generation and utilizes the shown keywords to extract the necessary data. Moreover, those keywords provide additional clues and structure to the LLM by clearly indicating the start and end points of the sentences. The final sentence, inspired by Amin et al. (2023), is designed specifically to minimize unwanted outputs that usually happen due to the nature of the pre-training of LLMs and their initial underlying prompts. After parsing, each dataset is checked manually to remove sentences with grammatical issues, nonsense content, or ones unrelated to the command they were labeled as.

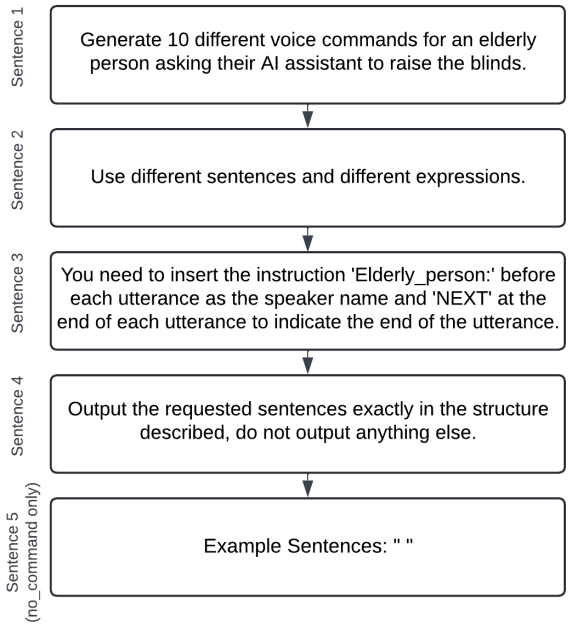

Figure 2: An example for the structure of our prompts. The original prompts are in German and can be found in Appendix A in Table 5.

The "no_command" label required more prompt engineering than the others, since the concept of false positive sentences does not seem to be easily graspable for LLMs. It often caused the LLM to hallucinate or generate data better suited to other labels. However, we managed to address this issue by employing a few-shot prompting approach with additional example sentences to enrich the existing prompt structure. To further enhance the diversity of the responses, we split the "help" label into two distinct prompts. In addition to the standard prompt, we also requested short calls for help (1-2 words) to account for emergencies, where the user might not be able to speak in full sentences. Additionally, for the "no_command" label, we tailored three different prompts that correspond to false pos-

[2] https://huggingface.co/meta-llama/Llama-3.1-8B

seed	top_p	top_k	RP	typ. p	temp.
0	1	10000	1	0.995	0.7

Table 1: The parameters used for the local LLMs. RP: Repetition Penalty

itive sentences for "help", "light_on" or "light_off", and "roll_up" or "roll_down", respectively. A few examples of false positive sentences would be: "My assistant turns on the lights as soon as it gets dark.", for control of the lights, "Every morning I open the blinds.", for control of the blinds, and "I should call my doctor later.", for help.

3.2.2 LLM Parameters

We utilized the local LLMs, LeoLM and Llama3, by integrating the TextgenWebUI Chat API [3] and Ollama[4], which offer similar functionalities as OpenAI's Chat API, but also allow access to the full range of LLM parameters. Table 1 contains all the parameters we used for the dataset generation for the local LLMs. The seeds were initialized randomly with 0 and then selected from the range $(0, 2^{35})$. They remained fixed during the generation to ensure the generated data would be diverse and reproducible. In our specific setup, the temperature setting was not as critical due to the stabilized seed; thus, we maintained it at the default value of 0.7. We selected high values for top_p, top_k, and $typical\_p$ to enrich the context diversity and expand the array of potential utterances. Additionally, we reduced the repetition penalty (RP) to prevent constriction of the LLM with the tokens it previously generated.

3.2.3 Speech Synthesis

Since our model is based on speech input, we utilize XTTS-v2[5], which is a multilingual text-to-speech model that generates high-quality speech, in its German language configuration. XTTS-v2 is an extension of the XTTS model by Casanova et al. (2024), which builds on the Tortoise model (Betker, 2023) to enable multilingual training, faster inference, and voice cloning.

We randomly select four speakers (two male, two female) between 70 and 80 years of age from the Common Voice DE 10.0 dataset (Ardila et al., 2020), and create short audio files as a reference for the generation of synthetic speech. We generate audio from these reference files for each LLM-generated text dataset and use the resulting synthetic speech for the complete evaluation of our approach.

3.3 Architecture

The task of intent recognition requires reliable speech recognition, not only to enable the use of language models to perform classification on the transcript but also to increase the transparency of the entire model in case of misclassification. Considering these conditions, we use the state-of-the-art Whisper model developed by Radford et al. (2023) in our setup. Whisper follows an encoder-decoder structure, which allows us to neatly utilize the same model for both the baseline and our approach. Specifically, we use a pretrained Whisper-small model that has already been fine-tuned for the German language[6] and adapt it to the domain of elderly German speech with the SVC-de dataset, following the approach by Pekarek Rosin and Wermter (2023). We continually train only the encoder part of the architecture with Experience Replay (Rolnick et al., 2019) on 10% of the Common Voice DE 10.0 dataset, to avoid overfitting the model on the limited vocabulary of the SVC-de dataset.

In our approach, we combine the domain-adapted Whisper model with a Transformer language model (LM) trained on synthetic text data to perform the task of intent recognition on the transcript provided by the Whisper model (Figure 1). We utilize three pretrained Transformer LMs: BERT (Devlin et al., 2019), DistilBERT (Sanh et al., 2019), and Electra (Clark et al., 2020). BERT was one of the first transformer-based LMs that allowed the adaptation to specific tasks without retraining the entire model, due to the pretrained bi-directional representations in each layer. This property makes BERT and other models like it uniquely suited for low-resource tasks. Additionally, these models have proven reliable on a variety of tasks, and pretrained German versions are readily available in the Hugging Face model repository[7].
DistilBERT reduces BERT's size by 40% through knowledge distillation, and Electra replaces BERT's masked language modeling task for pre-training with a more sample-efficient one. Since

[3] https://github.com/oobabooga/text-generation-webui
[4] https://ollama.com/
[5] https://huggingface.co/coqui/XTTS-v2
[6] https://huggingface.co/bofenghuang/whisper-small-cv11-german
[7] https://huggingface.co/

	train/test	ChatGPT	LeoLM	Llama3	Combined
	ChatGPT	95.59 ± 0.90	79.43 ± 1.79	71.00 ± 1.40	81.83 ± 1.09
BERT	LeoLM	82.91 ± 2.06	94.35 ± 0.76	76.75 ± 1.45	$\mathbf{84.26 \pm 0.40}$
	Llama3	80.00 ± 2.68	76.31 ± 1.00	92.01 ± 1.23	82.41 ± 0.44
	Combined	98.16 ± 0.92	97.43 ± 0.55	95.40 ± 0.76	96.98 ± 0.24
	train/test	ChatGPT	LeoLM	Llama3	**Combined**
	ChatGPT	92.99 ± 1.08	78.29 ± 2.17	71.75 ± 1.58	80.25 ± 0.69
DistilBERT	LeoLM	81.19 ± 1.44	93.71 ± 0.63	70.32 ± 1.42	$\mathbf{82.58 \pm 0.61}$
	Llama3	76.76 ± 2.76	74.59 ± 1.38	89.25 ± 1.40	79.67 ± 0.40
	Combined	97.70 ± 0.37	96.89 ± 0.55	94.40 ± 0.61	96.51 ± 0.36
	train/test	ChatGPT	LeoLM	Llama3	**Combined**
	ChatGPT	84.80 ± 1.79	68.89 ± 2.33	67.71 ± 1.81	72.86 ± 1.03
Electra	LeoLM	72.05 ± 2.32	87.24 ± 1.07	66.31 ± 1.72	$\mathbf{75.06 \pm 0.83}$
	Llama3	66.43 ± 1.14	71.04 ± 1.09	81.94 ± 2.92	72.35 ± 1.23
	Combined	94.88 ± 1.01	94.35 ± 0.81	93.85 ± 0.67	94.33 ± 0.36

Table 2: Results for intent recognition from text with all transformer models on all synthetic text datasets. Results are averaged over 5 runs, each model was trained for 5 epochs. We show the mean accuracy (%) on the test dataset, averaged over 5 checkpoints, and the standard deviation.

the application context of our approach is real-time interaction with the user, and we want to create a model that could be used locally without access to GPU resources, we utilize these comparatively smaller Transformer models instead of LLMs.

3.4 Experimental Setup

In our experimental setup, we take a pretrained BERT (bert-base-german-cased), DistilBERT (distilbert-base-german-cased), and Electra (german-nlp-group/electra-base-german-uncased) model from the Hugging Face model repository[7]. We equip each model with a fully connected output layer for classification and keep the pretrained model frozen. We train the models with a learning rate of 3e-4 and a dropout of 0.1 over 5 epochs, on the ChatGPT-, LeoLM-, and Llama3-generated text datasets, with a split of 70-20-10 for training, validation, and testing. We selected these hyperparameters empirically, since the models converged after approximately 4 epochs.

Afterward, we perform a cross-evaluation between datasets and also evaluate each model on a combination of all generated datasets, to examine whether there are noticeable differences in quality within the LLM-generated data. This also allows us to assess whether there are LLM-specific patterns in the data and if one of the datasets allows a higher level of generalization to unseen sentences.

We then combine the domain-adapted Whisper-small model with the trained LMs and evaluate our architecture on the generated speech datasets. The goal is to compare its performance with the accuracy achieved by the text transformers. We also examine the word error rate (WER) and character error rate (CER) of Whisper on the generated speech datasets and compare them with its performance on SVC-de, to assess the quality of the synthetic speech.

As a baseline, we use the output of the encoder part of the Whisper model to train a two-layer classification network for intent recognition on the SVC-de dataset only. This baseline represents traditional approaches and their limitations in low-resource domains.

4 Results

We train and evaluate BERT, DistilBERT, and Electra on each synthetic dataset to examine how well they generalize to different semantic structures and syntax. As seen in Table 2, the transformer models trained with data generated by LeoLM show the best generalization abilities across all datasets. While BERT and DistilBERT perform similarly,

Dataset	WER (%) ↓	CER (%) ↓
ChatGPT	12.14	6.16
LeoLM	15.01	8.41
Llama3	9.56	4.94
SVC-de	5.65	5.42

Table 3: The word error rate (WER) and the character error rate (CER) of the domain-adapted Whisper-small on the real-world SVC-de dataset and the synthetic speech datasets.

Electra's accuracy is on average 10% lower. We also trained the LMs on the SVC-de transcripts and found that the results of the evaluation matched the baseline results in Table 4, with low accuracies (35-40%) across all synthetic datasets.

The confusion matrices in Figure 3 show that models trained on the LeoLM data generally show a more stable performance, even for unseen samples generated by other models. All models exhibit some level of confusion with the differentiation of "light_on" and "roll_up" from their counterparts and "no_command". However, BERT outperforms both Electra and DistilBERT. Electra seems to have more issues overall with the separation of different classes, which is also reflected in the previously discussed lower accuracies. A notable, positive result is the fact that all models across all datasets seem to be able to distinguish real calls for help reliably from the false positives introduced in the data for "no_command".

We continue the evaluation of our approach and the baseline with synthetic speech, as discussed in Section 3.2. As can be seen in Table 3, the quality of the generated speech is within an acceptable range, considering the domain-adapted ASR model was not trained on the synthetic speech. We observe word error rates (WERs) of 12.14% for ChatGPT, 15.01% for LeoLM, and 9.56% for Llama3. Meanwhile, the real speech from the SVC-de dataset achieves a WER as low as 5.65%, which is in line with the results by Pekarek Rosin and Wermter (2023).

The results in Table 4 show that pre-training the LMs on the data generated by the LLMs increases the overall models' robustness against unseen sentences. This is supported by the significantly lower performance of the baseline on the synthetic data, with around a 50% difference in accuracy. The LeoLM data proves to be especially useful for pre-training, with Whisper+DistilBERT(LeoLM) achieving the highest accuracy with 83.01% on SVC-de compared to the baseline result of 95.05%. It even outperforms the models trained on a combination of the synthetic text data. The same patterns that can be observed in the evaluation of the text data (Table 2) are present here as well, with a performance drop for Electra-based models. Overall, we observe that adding a language model trained on a supplementary synthetic text dataset vastly improves generalization to new data, compared to the baseline results.

5 Discussion

In our approach, we equip a Whisper ASR model with a Transformer LM for intent recognition, to allow LLM-generated text data to supplement the small-scale speech dataset available for the task (SVC-de). Our results show that training the LM on the synthetic text data increases the overall model's ability to generalize across various semantic structures and syntax, and previously unseen sentences.

LeoLM, a local LLM specifically created for the German language, outperforms both ChatGPT and Llama3 in terms of generalization to unseen data (Table 2) and demonstrates a more stable performance overall (Figure 3). This shows the potential of language-specific fine-tuning for local LLMs, which offer more transparency of their parameters and therefore higher control of the output and reproducibility. Notably, models trained only on the Llama3 data do not trail behind the ones trained on LeoLM or ChatGPT, even though Llama3 is a smaller model. For the BERT models in particular, performance differences are minimal, and BERT(Llama3) even outperforms BERT(ChatGPT) on the dataset combination.

The confusion matrices in Figure 3 show that the distinction between "help" and the false positives contained in "no_command" seems to be straightforward for all models, which is great for real-life applications, since calls for help should ideally not be misclassified at all. All models seem to struggle to varying degrees with differentiating between "light_on" or "light_off" and "roll_up" or "roll_down", which is to be expected due to the high similarity of the generated sentences. The Electra models seem to have more issues with this differentiation overall, especially the model trained on the ChatGPT data. In a real home assistant system, this would ideally be alleviated by introducing

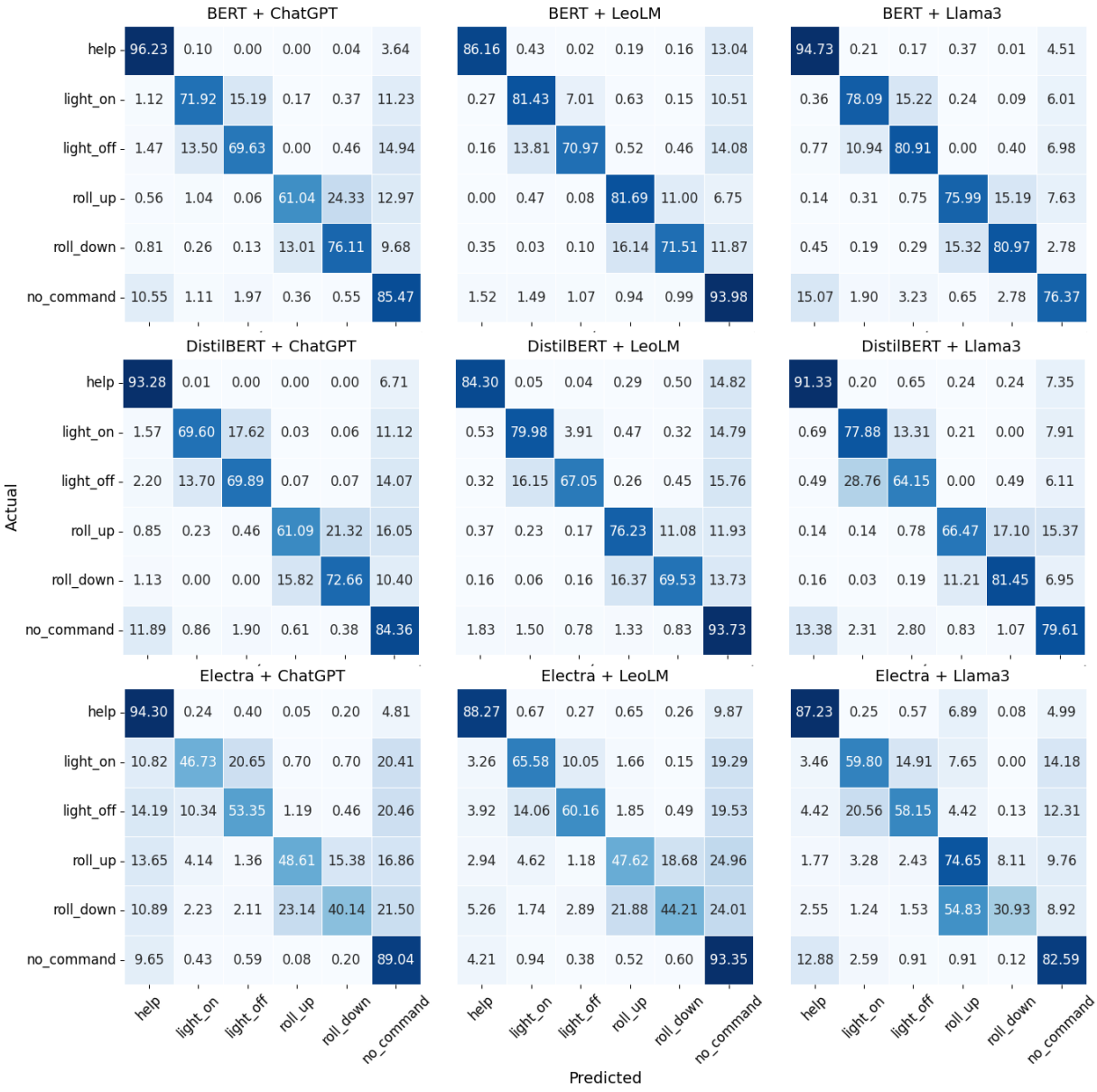

Figure 3: Confusion matrices for all LM+dataset variations, evaluated on the combination of all generated datasets.

context information to help reduce uncertainty, e.g., by checking the state of the lights or blinds.

The word error rates (WERs) of the synthetic speech datasets (Table 3) are comparable to the ones measured by Pekarek Rosin and Wermter (2023) for Whisper-small-de on the Common Voice DE test split (11.2%). This indicates that the quality of the synthetic speech approximates real speech. As can be seen in Table 4, the evaluation results on the synthetic speech datasets and SVC-de show that BERT and DistilBERT can keep the classification performance high for unseen speech data. The performance for the speech-based evaluation drops slightly across all models compared to the text-only evaluations, which is expected since ASR models usually create noisy transcriptions (Table 3), and some information might be lost. We also observe that supplementing with data from a single LLM (LeoLM) seems to be more beneficial than combining the data from all LLMs for training.

The low performance of the baseline Whisper-Encoder+CN on the synthetic datasets highlights how fine-tuning on small-scale speech data can significantly affect pretrained model generalizability and emphasizes the advantage of our approach. Since the WER of the synthetic speech is comparable to real speech, the limited vocabulary of the real dataset likely explains the baseline's low accuracy. Additionally, while SVC-de was used for domain adaptation of the Whisper encoder,

	train/test	ChatGPT-s	LeoLM-s	Llama3-s	SVC-de
Whisper+BERT	ChatGPT	91.49 ± 0.95	77.51 ± 1.44	71.35 ± 1.54	65.91 ± 3.86
	LeoLM	78.28 ± 0.46	89.95 ± 0.43	72.88 ± 1.06	$\mathbf{73.51 \pm 2.65}$
	Llama3	76.04 ± 1.11	71.99 ± 0.88	91.09 ± 0.70	63.44 ± 6.63
	Combined	93.63 ± 0.44	92.32 ± 0.26	94.23 ± 0.37	74.37 ± 2.87
Whisper+DistilBERT	ChatGPT	87.50 ± 1.60	73.93 ± 0.83	68.11 ± 2.23	75.58 ± 3.80
	LeoLM	74.29 ± 1.37	86.69 ± 1.42	67.43 ± 1.80	$\mathbf{83.01 \pm 1.54}$
	Llama3	75.04 ± 1.06	71.93 ± 0.83	87.04 ± 1.07	71.47 ± 1.89
	Combined	94.35 ± 0.11	92.13 ± 0.21	92.12 ± 0.42	72.28 ± 1.35
Whisper+Electra	ChatGPT	74.16 ± 1.34	65.50 ± 0.42	64.48 ± 0.84	61.67 ± 0.96
	LeoLM	65.94 ± 0.63	80.97 ± 0.36	63.35 ± 0.74	$\mathbf{71.42 \pm 2.27}$
	Llama3	67.16 ± 1.01	69.05 ± 1.64	81.76 ± 1.73	51.52 ± 4.53
	Combined	87.32 ± 0.65	86.48 ± 0.67	90.10 ± 0.32	68.61 ± 1.09
Whisper-Encoder+CN	SVC-de	34.72 ± 0.61	38.40 ± 0.65	36.36 ± 0.75	95.05 ± 0.39

Table 4: Results of the evaluation on the synthetic speech datasets (-s), and on the SVC-de dataset of all combinations of Whisper+LM models, trained on the text datasets generated by ChatGPT, LeoLM, and Llama3. Whisper-Encoder+CN is the baseline trained only on the real speech dataset, SVC-de. We show the mean accuracy (%), averaged over 5 checkpoints, and the standard deviation.

none of the Whisper+LM variations were explicitly trained on it for classification. Still, all models achieved above-average accuracy on SVC-de, with Whisper+DistilBERT(LeoLM) outperforming all other model variations and approaching the baseline trained on SVC-de (Whisper-Encoder+CN).

Finally, all our models require only 5 epochs to be trained sufficiently, which can be done in a fraction of the time needed for other approaches (Section 2), since each training run on the generated text data takes only 1-2 minutes.

6 Conclusion

In this paper, we present a novel approach for intent recognition (IR) in the domain of elderly German speakers using a home assistant system. We leverage a pretrained Whisper model and adapt it to the domain through layer-specific fine-tuning and continual learning on the SVC-de dataset. To address the limitations of the dataset, we generate supplementary classification datasets with three large language models (LLMs): LeoLM, Llama3, and ChatGPT. These datasets are then used to train transformer-based language models (LMs) for IR.

Our results show that a pretrained ASR model combined with an LM trained on synthetic text data displays increased robustness to diverse linguistic patterns and unseen vocabulary. Evaluating our models on high-quality synthetic speech shows that they outperform the baseline trained only on real-world data (SVC-de), indicating improved robustness to different speakers. This adaptability is critical for reducing user strain in real-world applications. We find that training Transformer LMs on synthetic text data is more efficient than continued ASR fine-tuning in terms of resources and generalizability of the model. Additionally, our models are reliably able to differentiate calls for help from false alarms, which is essential for a home-assistant system for elderly speakers.

This work is an exploration of LLMs and generative AI for the generation of German speech and language processing datasets. We show that (1) a domain-specific, smaller LLM like LeoLM (13B) can surpass larger models like ChatGPT (175B) in dataset quality, while offering transparency and reproducibility; (2) supplementing an ASR model with a language model trained on synthetic text data can enhance model performance and robustness; and (3) our method offers a fast and efficient way to adapt existing speech systems for IR tasks. As such, our approach offers a practical and scalable solution for deploying reliable home-assistant systems in real-world speaker environments.

Ethical Considerations

No experiments with human participants or additional recordings were conducted during this research. We chose to use local LLMs as well as ChatGPT for the dataset generation to explore their performance, since local LLMs allow for greater control of their parameters. For the sake of transparency, we have shared every parameter (Table 1) and prompt (Appendix A) we used. The generated data can be fully reproduced using the same LLMs, prompts, and parameters, provided that the same seed is used.

Limitations

In this research, we have encountered a couple of limitations associated with the use of smaller LLMs. It was observed that the LLM may truncate its outputs. To address this issue, we implemented a check during the parsing phase to verify the completeness of each utterance in terms of starting and ending keywords. If incomplete, the parser excludes that utterance, ensuring that the parsed data does not contain missing values. We also observed instances where recurrent utterances appear in more than one output. This issue is resolved by eliminating duplicate utterances during parsing and selecting the more diverse ones during manual filtering. For the ASR model, we chose one model to simplify the experimental setup, but ideally, we would have examined the performance of different versions of Whisper as well. The approach we follow for domain adaptation does an extensive evaluation with the same dataset, so we did not repeat a similar evaluation.

Acknowledgments

The authors gratefully acknowledge funding from Horizon Europe under the MSCA grant agreement No 101072488 (TRAIL), the German BMWK (SIDIMO), and the Study Abroad Graduate Scholarship by the Ministry of National Education of Türkiye.

References

Mostafa M. Amin, Erik Cambria, and Björn W. Schuller. 2023. Will affective computing emerge from foundation models and general artificial intelligence? a first evaluation of ChatGPT. *IEEE Intelligent Systems*, 38(02):15–23.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, Marseille, France. European Language Resources Association.

Axel Berg, Mark O'Connor, and Miguel Tairum Cruz. 2021. Keyword transformer: A self-attention model for keyword spotting. In *Proceedings of Interspeech 2021*, pages 4249–4253. ISCA.

James Betker. 2023. Better speech synthesis through scaling. *ArXiv*.

Edresson Casanova, Kelly Davis, Eren Gölge, Görkem Göknar, Iulian Gulea, Logan Hart, Aya Aljafari, Joshua Meyer, Reuben Morais, Samuel Olayemi, and Julian Weber. 2024. Xtts: a massively multilingual zero-shot text-to-speech model. In *Interspeech 2024*, pages 4978–4982.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. BERT for joint intent classification and slot filling. *arXiv*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *Proceedings of ICLR 2020*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT 2019*, page 4171–4186.

Pranay Dighe, Yi Su, Shangshang Zheng, Yunshu Liu, Vineet Garg, Xiaochuan Niu, and Ahmed Tewfik. 2024. Leveraging large language models for exploiting ASR uncertainty. In *Proceedings of 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12231–12235.

Kristina Dzeparoska, Ali Tizghadam, and Alberto Leon-Garcia. 2024. Intent assurance using LLMs guided by intent drift. *arXiv*.

Nora Fronemann, Kathrin Pollmann, and Wulf Loh. 2021. Should my robot know what's best for me? Human–robot interaction between user experience and ethical design. *AI & SOCIETY*, 37:517 – 533.

Amadeo Hernández, Rosa María Ortega-Mendoza, Esaú Villatoro-Tello, César Joel Camacho-Bello, and Obed Pérez-Cortés. 2024. Natural language understanding for navigation of service robots in low-resource domains and languages: Scenarios in spanish and nahuatl. *Mathematics*, 12.

Verena Irrgang, Veronika Solopova, Steffen Zeiler, Robert M. Nickel, and Dorothea Kolossa. 2024. Features and detectability of German texts generated with large language models. In *Proceedings of the 20th Conference on Natural Language Processing*

(KONVENS 2024), pages 264–280, Vienna, Austria. Association for Computational Linguistics.

Burak Can Kaplan, Hugo Cesar De Castro Carneiro, and Stefan Wermter. 2025. Can large language models generate effective datasets for emotion recognition in conversations? *ArXiv*.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pages 18–26, Suzhou, China. Association for Computational Linguistics.

Subrat Kumar Nayak, Ajit Kumar Nayak, Smitaprava Mishra, and Prithviraj Mohanty. 2023. Deep learning approaches for speech command recognition in a low resource kui language. *International Journal of Intelligent Systems and Applications in Engineering*, 11(2):377–386.

Xiang Lorraine Li, Adhiguna Kuncoro, Jordan Hoffmann, Cyprien de Masson d'Autume, Phil Blunsom, and Aida Nematzadeh. 2022. A systematic investigation of commonsense knowledge in large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11838–11855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Laureano Moro-Velazquez, Jaejin Cho, Shinji Watanabe, Mark A. Hasegawa-Johnson, Odette Scharenborg, Heejin Kim, and Najim Dehak. 2019. Study of the performance of automatic speech recognition systems in speakers with parkinson's disease. In *Proceedings of INTERSPEECH 2019*, pages 3875–3879, Graz, Austria. ISCA.

Mikel K. Ngueajio and Gloria Washington. 2022. Hey ASR system! why aren't you more inclusive? automatic speech recognition systems' bias and proposed bias mitigation techniques. a literature review. In *Proceedings of 24th International Conference on Human-Computer Interaction (HCI 2022)*, page 421–440. Springer-Verlag.

Akhil Padmanabha, Jessie Yuan, Janavi Gupta, Zulekha Karachiwalla, Carmel Majidi, Henny Admoni, and Zackory Erickson. 2024. Voicepilot: Harnessing LLMs as speech interfaces for physically assistive robots. *arXiv*.

Theresa Pekarek Rosin, Vanessa Hassouna, Xiaowen Sun, Luca Krohm, Henri-Leon Kordt, Michael Beetz, and Stefan Wermter. 2025. A framework for adapting human-robot interaction to diverse user groups. In *Social Robotics*, pages 24–38, Singapore. Springer Nature Singapore.

Theresa Pekarek Rosin and Stefan Wermter. 2023. Replay to remember: Continual layer-specific fine-tuning for german speech recognition. In *Artificial Neural Networks and Machine Learning–ICANN 2023*, pages 489–500.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. 2019. Experience replay for continual learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS)*, pages 348–358, Vancouver, BC, Canada. Curran Associates, Inc.

Erik Sadovsky, Maros Jakubec, and Roman Jarina. 2023. Speech command recognition based on convolutional spiking neural networks. In *Proceedings of the 33rd International Conference Radioelektronika (RADIOELEKTRONIKA)*, pages 1–5.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing (EMC^2) - NeurIPS 2019*.

Joel Shor, Dotan Emanuel, Oran Lang, Omry Tuval, Michael Brenner, Julie Cattiau, Fernando Vieira, Maeve McNally, Taylor Charbonneau, Melissa Nollstadt, Avinatan Hassidim, and Yossi Matias. 2019. Personalizing ASR for dysarthric and accented speech with limited data. In *Proceedings of INTERSPEECH 2019*, pages 784–788, Graz, Austria. ISCA.

Martin Volk, Dominic P. Fischer, Patricia Scheurer, Raphael Schwitter, and Phillip B. Ströbel. 2024. LLM-based translation across 500 years. the case for early New High German. In *Proceedings of the 20th Conference on Natural Language Processing (KONVENS 2024)*, pages 368–375, Vienna, Austria. Association for Computational Linguistics.

Dominik Wagner, Alexander Churchill, Siddharth Sigtia, Panayiotis Georgiou, Matt Mirsamadi, Aarshee Mishra, and Erik Marchi. 2024. A multimodal approach to device-directed speech detection with large language models. In *Proceedings of 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10451–10455.

Pei Wang, Keqing He, Yejie Wang, Xiaoshuai Song, Yutao Mou, Jingang Wang, Yunsen Xian, Xunliang Cai, and Weiran Xu. 2024. Beyond the known: Investigating LLMs performance on out-of-domain intent detection. *arXiv*.

P. Warden. 2018. Speech commands: A dataset for limited-vocabulary speech recognition. *ArXiv*.

A Prompts

Label	Prompt
help	Generieren Sie 10 verschiedene Sprachbefehle, mit denen ein älterer Mensch seinen KI-Assistenten in gefährlichen Situationen um Hilfe bitten kann, ohne jedes Mal explizit um Hilfe zu bitten. Verwenden Sie verschiedene Sätze und unterschiedliche Gesundheitssituationen. Sie müssen vor jeder Äußerung die Anweisung 'Ältere_Person:' als Sprechername einfügen und 'NÄCHSTES' am Ende jeder Äußerung, um das Ende der Äußerung zu kennzeichnen. Geben Sie die angeforderten Sätze genau in der beschriebenen Struktur aus, geben Sie nichts anderes aus.
	Generieren Sie 10 verschiedene sehr kurze Sprachbefehle, mit denen eine ältere Person ihren KI-Assistenten in gefährlichen Situationen um Hilfe bitten kann, ohne jedes Mal explizit um Hilfe zu bitten. Verwenden Sie verschiedene, kurze Sätze, die aus ein bis zwei Wörtern bestehen und verschiedene Gesundheitssituationen beschreiben. Fügen Sie vor jeder Äußerung die Anweisung 'Ältere_Person:' als Sprechername ein und am Ende jeder Äußerung 'NÄCHSTES', um das Ende der Äußerung zu markieren. Geben Sie die geforderten Sätze genau in der beschriebenen Struktur aus, geben Sie nichts anderes aus.
roll	Generieren Sie 10 verschiedene Sprachbefehle für eine ältere Person, die ihren KI-Assistenten bittet, die Rollläden hochzufahren. Verwenden Sie verschiedene Sätze und unterschiedliche Ausdrücke. Sie müssen vor jeder Äußerung die Anweisung 'Ältere_Person:' als Sprechername einfügen und 'NÄCHSTES' am Ende jeder Äußerung, um das Ende der Äußerung zu kennzeichnen. Geben Sie die angeforderten Sätze genau in der beschriebenen Struktur aus, geben Sie nichts anderes aus.
	Generieren Sie 10 verschiedene Sprachbefehle für eine ältere Person, die ihren KI-Assistenten bittet, die Rollläden herunterzufahren. Verwenden Sie verschiedene Sätze und unterschiedliche Ausdrücke. Sie müssen vor jeder Äußerung die Anweisung 'Ältere_Person:' als Sprechername einfügen und 'NÄCHSTES' am Ende jeder Äußerung, um das Ende der Äußerung zu kennzeichnen. Geben Sie die angeforderten Sätze genau in der beschriebenen Struktur aus, geben Sie nichts anderes aus.
lights	Generieren Sie 10 verschiedene Sprachbefehle für eine ältere Person, die ihren KI-Assistenten bittet, das Licht einzuschalten. Verwenden Sie verschiedene Sätze und unterschiedliche Ausdrücke. Sie müssen vor jeder Äußerung die Anweisung 'Ältere_Person:' als Sprechername einfügen und 'NÄCHSTES' am Ende jeder Äußerung, um das Ende der Äußerung zu kennzeichnen. Geben Sie die angeforderten Sätze genau in der beschriebenen Struktur aus, geben Sie nichts anderes aus.
	Generieren Sie 10 verschiedene Sprachbefehle für eine ältere Person, die ihren KI-Assistenten bittet, das Licht auszuschalten. Verwenden Sie verschiedene Sätze und unterschiedliche Ausdrücke. Sie müssen vor jeder Äußerung die Anweisung 'Ältere_Person:' als Sprechername einfügen und 'NÄCHSTES' am Ende jeder Äußerung, um das Ende der Äußerung zu kennzeichnen. Geben Sie die angeforderten Sätze genau in der beschriebenen Struktur aus, geben Sie nichts anderes aus.
no_command	Generieren Sie 10 Sätze von einer älteren Person, die von einer Spracherkennung fälschlicherweise als 'Bitte um Hilfe' klassifiziert werden können, aber in Wirklichkeit als 'kein Befehl' für einen KI-Assistenten verwendet werden. Der Assistent benutzt dafür eine Keyword Detection. Verwenden Sie verschiedene Sätze und verschiedene Ausdrücke. Sie müssen vor jeder Äußerung als Sprechernamen die Anweisung 'Ältere_Person:' und am Ende jeder Äußerung 'NÄCHSTES' einfügen, um das Ende der Äußerung anzuzeigen. Ein paar Beispiele: Ältere_Person: Kannst du mir bitte helfen, mein Handy zu finden? NÄCHSTES Ältere_Person: Mein Sohn hat mir gestern mit dem Garten geholfen. NÄCHSTES Ältere_Person: Manchmal muss ich um Hilfe bitten. NÄCHSTES Ältere_Person: Diese neuen Geräte sind ohne Hilfe gar nicht zu bedienen. NÄCHSTES Ältere_Person: Früher konnte ich alles alleine, ohne um Hilfe zu bitten. NÄCHSTES
	Generieren Sie 10 Sätze von einer älteren Person, die von einer Spracherkennung fälschlicherweise als 'Rollläden hoch- oder runterfahren' klassifiziert werden können, aber in Wirklichkeit als 'kein Befehl' für einen KI-Assistenten verwendet werden. Der Assistent benutzt dafür eine Keyword Detection. Verwenden Sie verschiedene Sätze und verschiedene Ausdrücke. Sie müssen vor jeder Äußerung als Sprechernamen die Anweisung 'Ältere_Person:' und am Ende jeder Äußerung 'NÄCHSTES' einfügen, um das Ende der Äußerung anzuzeigen. Ein paar Beispiele: Ältere_Person: Mein Assistent fährt die Rollläden jeden Abend pünktlich um 18:00 herunter. NÄCHSTES Ältere_Person: Im Sommer habe ich die Jalousien gerne den ganzen Tag unten. NÄCHSTES Ältere_Person: Es ist sehr praktisch, dass mein Sprachassistent die Rollläden steuern kann. NÄCHSTES Ältere_Person: Meine Rollläden sind beim letzten Sturm kaputtgegangen. NÄCHSTES Ältere_Person: Sobald meine Jalousien oben sind, kann ich meinen Tag beginnen. NÄCHSTES
	Generieren Sie 10 Sätze von einer älteren Person, die von einer Spracherkennung fälschlicherweise als 'Licht ein- oder ausschalten' klassifiziert werden können, aber in Wirklichkeit als 'kein Befehl' für einen KI-Assistenten verwendet werden. Der Assistent benutzt dafür eine Keyword Detection. Verwenden Sie verschiedene Sätze und verschiedene Ausdrücke. Sie müssen vor jeder Äußerung als Sprechernamen die Anweisung 'Ältere_Person:' und am Ende jeder Äußerung 'NÄCHSTES' einfügen, um das Ende der Äußerung anzuzeigen. Ein paar Beispiele: Ältere_Person: Mein Assistent schaltet mir jeden morgen die Lichter an. NÄCHSTES Ältere_Person: Gestern hatten wir schon sehr früh kein Licht mehr im Raum. NÄCHSTES Ältere_Person: Die Tatsache, dass mein Sprachassistent das Licht an- und ausschalten kann ist sehr praktisch. NÄCHSTES Ältere_Person: Da ist mir ein Licht aufgegangen. NÄCHSTES Ältere_Person: Manchmal ist es hier ziemlich dunkel ohne Licht. NÄCHSTES

Table 5: All the prompts used for the different classes in the datasets.