

SOLID: Self-seeding and Multi-intent Self-instructing LLMs for Generating Intent-aware Information-Seeking Dialogs

Arian Askari^{1*}, Roxana Petcu^{2*}, Chuan Meng², Mohammad Aliannejadi²
Amin Abolghasemi¹, Evangelos Kanoulas², Suzan Verberne¹

¹Leiden University {a.askari, m.a.abolghasemi, s.verberne}@liacs.leidenuniv.nl

²University of Amsterdam {r.m.petcu, c.meng, m.aliannejadi, e.kanoulas}@uva.nl

Abstract

Intent prediction in information-seeking dialogs is challenging and requires a substantial amount of data with human-labeled intents for effective model training. While Large Language Models (LLMs) have demonstrated effectiveness in generating synthetic data, existing methods typically rely on human feedback and are tailored to structured, task-oriented intents. In this paper, we leverage LLMs for zero-shot generation of large-scale, open-domain, intent-aware information-seeking dialogs to serve as training data for intent prediction models. We introduce SOLID, a method that generates dialogs turn by turn using novel *self-seeding* and *multi-intent self-instructing* strategies. Additionally, we propose SOLID-RL, a finetuned version that generates an entire dialog in one step using data created with SOLID. SOLID and SOLID-RL are each used to generate over 300k intent-aware dialogs, significantly surpassing the size of existing datasets. Experiments show that intent prediction models trained on sampled dialogs generated by SOLID and SOLID-RL outperform those trained solely on human-generated dialogs. Our findings demonstrate the potential of LLMs to expand training datasets, as they provide valuable resources for conversational agents across multiple tasks. Our self-seeding and self-instructing approaches are adaptable to various conversational data types and languages with minimal modifications.

1 Introduction

Understanding user intent in information-seeking dialogs is essential for conversational agents to fulfil the user’s information need. Specifically, identifying the underlying intent behind each conversational turn enables the dialog system to respond appropriately. For instance, if a user exhibits signs of confusion regarding the previous dialog turn, the system can reiterate its response with a more

* Equal contribution (shared co-authorship)

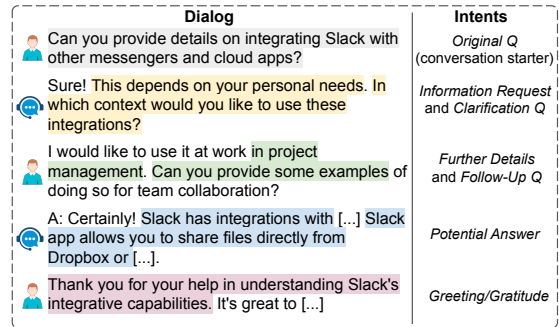


Figure 1: Example dialog with sequence of intents.

clear phrasing. A thorough definition of intents can be found in Appendix A.1. The turn-level intent identification task is called Intent Prediction¹ (IP) (Zamani et al., 2023). Leveraging predicted intents has been shown to benefit various Information Retrieval (IR) and Natural Language Processing (NLP) tasks, e.g., response selection (Yang et al., 2020; Kumar et al., 2019), answer selection and user satisfaction estimation (Deng et al., 2023, 2022). Figure 1 illustrates an example of intents in an information-seeking conversation extracted from the MSDialog-intent dataset (Qu et al., 2018a). These are the intents referenced throughout this study. The full taxonomy of MSDialog intents can be found in Table 7 in Appendix A.7. Notably, intents are defined at the utterance level, with each utterance being associated with one or potentially multiple intents. Using MSDialog intents ensures compatibility with existing datasets, however, our methodology is inherently flexible, as it can accommodate finer-grained taxonomies or additional annotations.

Motivation. Intent Prediction (IP) is a challenging task (Qu et al., 2019b). Recent work (Zhao et al., 2023a) has shown that prompting ChatGPT to handle IP tasks in a few-shot manner reaches limited performance, with ChatGPT performing worse

¹Intent prediction and intent classification are used interchangeably in this work.

than supervised IP methods. Supervised methods, however, require a large number of human-annotated information-seeking dialogs to achieve satisfactory performance, and often fail to generalize across different datasets. For example, Qu et al. (2019a) show that their IP model trained on MSDialog-intent achieves a precision of 69% compared to only 47% on the Ubuntu dialog corpus. This large gap in effectiveness might be due to the small size of existing datasets, which limits the ability to generalize. Furthermore, manually annotating intents is resource-intensive, leading to limited training data for IP models (Qu et al., 2019a; Penha et al., 2019a). Using LLMs directly for intent classification is not a reliable alternative, since it has been shown that LLMs have limited effectiveness in accurately labeling intents (Zhao et al., 2023b; Hu et al., 2023; Zhao et al., 2023a).

We employ LLMs for large-scale synthetic data augmentation, where each dialog utterance has one or more associated intents. Our proposed method generates each dialog by following a turn-by-turn sequence of real-world intents extracted from MSDialog-intent (Qu et al., 2018a). Leveraging these intents from existing human-labeled dialogs ensures that our synthetic data aligns and possesses the same level of diversity as real-world data. While MSDialog-intent is built on question answering (QA) interactions between information seekers and product specialists on Microsoft products, it is presented as a dialog dataset. Each dialog is entity-centric, i.e., it is constructed around a specific entity as the object of discussion.

Challenges. We identify two main challenges in using LLMs for generating intent-aware information-seeking dialogs: (i) Information-seeking dialogs are typically centred around a topic or entity that serves as the foundation of the conversation. In dialog generation, this information is known as the seed of the dialog, providing the starting point and context of the conversation (Kulkarni et al., 2024). An example of a dialog seed is shown in Figure 2. While the common practice is to leverage seed information from external resources (e.g., Wikipedia) (Ding et al., 2023), recent work has shown that feeding such seeds to an LLM often results in low-quality generations, as LLMs tend to underperform when prompted with unfamiliar information. (ii) Our preliminary results show that prompting an open-source LLM to generate an entire dialog yields low performance, often producing

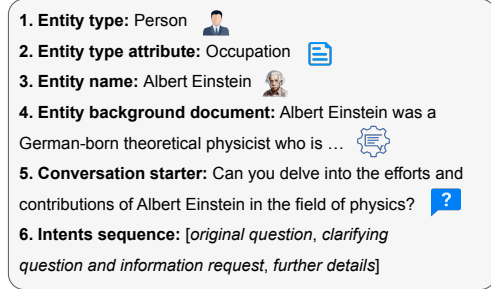


Figure 2: Example of a seed generated through self-seeding via SOLID.

too many or too few utterances that do not follow the intent sequence. However, generating utterances one by one improves generation quality.

SOLID. To address these challenges, we propose SOLID: a Self-seeding and multi-intent self-instructing method for generating Large-scale Intent-aware Information-Seeking Dialogs. SOLID has novel *self-seeding* and *multi-intent self-instructing* features: (i) *self-seeding* involves querying the LLM to generate a seed, which is then used in a subsequent step to prompt the same LLM to produce a dialog. (ii) *multi-intent self-instructing* allows our framework to generate utterances conditioned on multiple intents simultaneously.

SOLID-RL. While SOLID creates high-quality, intent-aware conversations by generating dialogs turn by turn, it incurs significant computational costs. To enhance efficiency, we propose SOLID-RL, which enables the generation of the full dialog in a single pass. Although prompting an LLM to generate an entire dialog reduces quality, similar to how solving complex tasks without step-by-step breakdowns results in less coherent reasoning (Wei et al., 2022), SOLID-RL mitigates this problem by fine-tuning the LLM on the high-quality dialogs produced by SOLID. This method preserves dialog quality while significantly reducing computational overhead. The framework is illustrated in Figure 3.

Contributions. Our contributions are as follows:

- SOLID: We propose a novel Self-seeding and multi-intent self-instructing method for generating Large-scale Intent-aware Information-Seeking Dialogues.
- SOLID-RL: We propose a fine-tuned method for efficiently generating entire dialogs using SOLID-generated synthetic dialogs.
- We provide two large-scale synthetic intent-aware information-seeking dialog datasets: SOLISpeak and SOLITurbo, each consisting of

316,697 dialogs. Our implementation is open-sourced as a Python library.²

- We demonstrate that IP methods trained on a mix of limited available human data and a subset of SOLISpeak and SOLITurbo achieve significantly higher effectiveness than those trained solely on human-labeled datasets.

2 Related work

Data augmentation with LLMs. Data augmentation using LLMs has gained significant attention in recent research (Abolghasemi et al., 2024; Askari et al., 2023b; Tunstall et al., 2023; Abbasiantaeb et al., 2023; Liu et al., 2024b). For example, ConvSDG is a framework that improves conversational search performance by leveraging synthetic training data generated at the dialog, session and query level (Mo et al., 2024). Similarly, LAPS (Joko et al., 2024) introduces a data augmentation framework where LLMs guide a human worker into constructing personalized multi-session dialogs, enabling the creation of large-scale conversational datasets that reflect real-world user preferences. (Lin et al., 2024) further contributes with a zero-shot method to generate refined data for intent detection, expanding the versatility of generated data for downstream tasks. The Self-Instruct method (Wang et al., 2022) allows LLMs to augment their own prompt instructions (Taori et al., 2023; Chiang et al., 2023; Zheng et al., 2023). Dialogic (Li et al., 2022) employs in-context learning to generate annotated dialogues in a controllable fashion, leading to a more structured data augmentation framework. Meanwhile, SynTOD (Samarinas et al., 2024) introduces an approach that simulates task-oriented dialogues using both state transition graphs and LLMs. However, none of these existing datasets have user intents associated, therefore they cannot be used for training IP models. Additionally, some studies use LLMs to augment training data for intent classification, but the utterances lack dialog context (Sahu et al., 2022) or are limited to "yes" and "no" responses (Kim et al., 2022), which differs from intent-aware conversation generation.

Intent Prediction in information-seeking dialogs. Utterance-level IP in dialog systems is a well-studied task aimed at identifying the intent at each utterance within a dialog (Meng et al., 2023; Colombo et al., 2020). MSdialog-Intent (Qu

et al., 2018b) and MANtIS (Penha et al., 2019b) are the largest available datasets on intent-aware information-seeking dialogs (statistics are shown in Table 4). However, these datasets still remain relatively small, because of the high cost of human annotations. As stated by (Qu et al., 2019a) annotating utterance intents of 2,100 information-seeking conversations costs approximately 1,700 dollars using Amazon MTurk. This study focuses on open-domain information-seeking dialogs, excluding research on IP in other types of dialogs, such as chit-chat dialogs (Ghazarian et al., 2022; Qu et al., 2023), dialogs with evidence from single or multi-knowledge sources (Wu et al., 2023), and task-oriented dialogs (Shah et al., 2018; Rastogi et al., 2020; King and Flanigan, 2024; Arora et al., 2024; Askari et al., 2023a).

3 Intent-Aware Dialog Generation

3.1 SOLID

SOLID generates dialogs almost entirely based on its internal knowledge, without relying on external seeds or conversations, with the exception of using intent labels from MSDialog-intent. This ensures a self-contained generation process.

Seed Generation. SOLID is initiated by producing dialog seeds that serve as the starting point for dialog generation. Using the same model for both seed and dialog generation ensures that the LLM has internal knowledge of the entity and its attributes, enabling a more informed dialog generation process. Methods such as Liu et al. (2024a) employ external seeds, relying on predefined content to generate dialogs using LLMs. While external seeds ensure factual background content, they do not guarantee that the LLM is familiar with specific information. See Figure 2 as an example of a seed that can be generated by SOLID. We take multiple steps to generate high-quality dialog seeds:

- *Entity type and attribute generation:* We first prompt the LLM to generate 100 *entity types* (e.g., 'Person', 'Artist'), along with 10 *entity attributes* associated with each entity type (e.g., 'Occupation', 'Financial').
- *Entity name generation:* For each entity type, we ask the LLM to generate an *entity name* (e.g., 'Albert Einstein'). To encourage diversity in the data, we prompt the LLM to generate 100 names for each letter in the English alphabet across all entity types ($26 \times 100 \times 100$), resulting in 50K entity names after automatic filtering. More details

²https://anonymous.4open.science/r/solid_repo/README.md

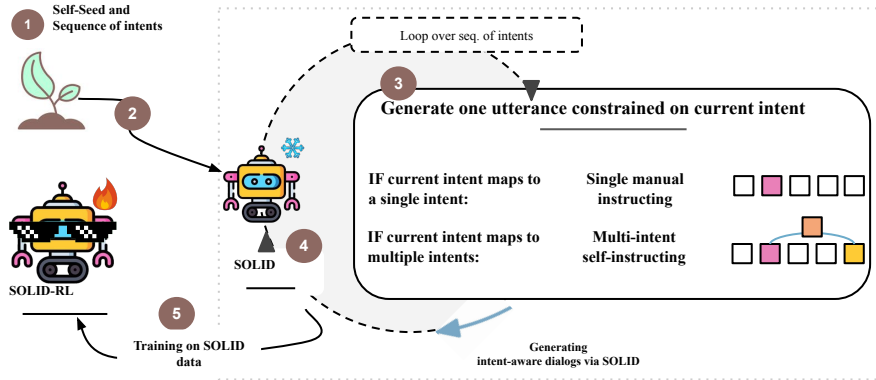


Figure 3: Illustrating SOLID and SOLID-RL: Starting with a self-generated seed and a sequence of real-world intents, SOLID produces utterances sequentially, with each utterance conditioned on one or more intents. During phase 3, each utterance falls into one of two categories: it either corresponds to a single intent, based on a definition from taxonomy 7, or combines multiple intent definitions. In the case of multiple intents, SOLID employs multi-intent self-instruction to merge these into a single cohesive instruction.

on filtering entities in Appendix A.4.

- *Background document generation:* Inspired by Yu et al. (2022), we prompt the LLM to generate a *background document* for each entity name (e.g., “Albert Einstein was a ...”).
- *Conversation starter:* Given the seed information generated so far, we prompt the LLM to create the first utterance, namely, the *conversation starter* (e.g., “Can you delve ...”). This is done by using the prompt shown in Figure 7.
- *Intent sequence:* We randomly sample a *sequence of intents* from the existing MSDialog-intent dataset (Qu et al., 2018a), which defines a set of intents for information-seeking conversations. By performing sampling with replacement, we expect the distribution of intent sequences to closely match the original distribution of the dataset. This intent set allows us to capture the natural flow and complexity of real dialogs. The number of intents in the sampled sequence of intents indicates the length of the dialog to be generated. E.g., the number of intents in Figure 1 is five, resulting in a dialog with five utterances. Figures 4 and 16 illustrate the seed generation process in more detail. All generated entity types are illustrated in Appendix A.5. Furthermore, all prompts are available in Appendix A.6.

Dialog Generation. Given a seed, we prompt the LLM to generate each subsequent utterance based on its associated intent. If the utterance is associated with a single intent, we extract the corresponding prompt instruction from Tables 8 and 9 (details in Appendix A.2). If the utterance is associated with multiple intents, i.e. any combi-

Dataset	# Dialogs	Avg. # Turns	Avg. Length (by token)	Avg. Utt. Length (by token)
MSDialog	2,199	4.56	297.12	65.16
MANtS	1,356	5.90	492.65	131.48
SOLISpeak (Ours)	316,697	4.57	1064.70	236.60
SOLITurbo (Ours)	316,697	4.57	1097.10	241.80

Table 1: Statistics of intent-aware information-seeking dialog datasets. ‘T’, ‘Dial’, ‘Utt’, and ‘L’ refer to Turns, Dialogs, Utterances, and Length respectively.

nation of two or more without repetition, we first retrieve the prompt instructions for each intent individually, and then query the LLM to create a customized prompt by merging the extracted instructions. *Multi-intent self-instructing* is essential, as manually creating prompt instruction for C_{12}^k as k approaches larger values becomes unmanageable and impractical. Following Algorithm 1, we generate each utterance and iteratively prompt the LLM with the seed, the next corresponding instruction, and the current dialog history.

Post-processing. In the final step, we post-process the generated dialogs and filter out the noisy ones. This involves three steps: (i) trimming to the last punctuation, as LLMs can reach their maximum token capacity without ending the last utterance, (ii) removing empty lines, and (iii) removing the dialog turn keywords, i.e., “User” or “Agent.”³

3.2 SOLID-RL

SOLID employs an utterance-by-utterance generation process. While this approach results in highly

³Post-processing implementation is available at https://anonymous.4open.science/r/solid_repo/solid_conversation/utils.py

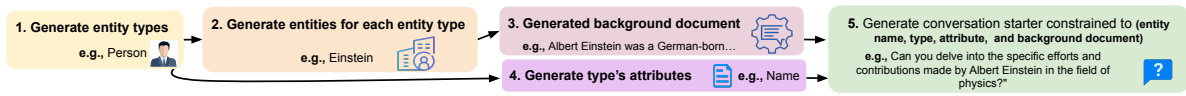


Figure 4: The illustration of seed generation through self-seeding via SOLID.

coherent dialogs that closely follow the dialog seed, it brings significant computational costs. To address this, we propose SOLID-RL, a method that enables the generation of an entire dialog in a single step. We leverage Direct Preference Optimization (DPO) (Rafailov et al., 2023) for our RL-training. DPO requires chosen and rejected samples for training. We label the generated dialog in an utterance-by-utterance manner by SOLID as ‘chosen’ and dialogs generated by a single instruction as ‘rejected’. The prompt for generating a dialog given a single instruction is presented in Figure 9 in the appendix. Since the quality of synthetic data generated by SOLID can vary significantly, establishing a dialog-level quality metric that can be used to guide the training process of SOLID-RL is crucial.

Through human assessment, we observed that shorter dialogs exhibit higher quality. We aimed to inject this observed signal during training by designing prefixes constrained to the length of the dialog. These prefixes are independent of the ‘chosen’ or ‘rejected’ labels of the dialog and are only injected at the beginning of the dialog given its length. For the shortest dialogs, we add the prefix “excellent dialog.” During inference, we use the prefix “excellent dialog” for all of our generations, regardless of the length of the dialog that is going to be generated. This approach triggers the LLM to generate dialogs that are more similar to higher-quality dialogs than to lower-quality ones. Our method is inspired by OpenChat (Wang et al., 2024) that proposes prefix-conditioned training, where different labeled data are annotated by more expert assessors. However, in our setup, both datasets are generated by the same LLM, but the length has a direct impact on the quality of the data.

We define this approach as Length-based quality estimation for Mixed-Quality training (LMQ), where dialog length serves as a quality indicator (See Appendix A.8.1 and A.8.2 for examples of low-quality training instances). We propose a simple yet effective heuristic. Before each dialog, we add a textual prompt representing its assessed quality: “excellent quality dialog:” for 1-3 utterances, “good quality dialog:” for 4 utterances, “average quality dialog:” for 5-10 utterances, “poor qual-

ity dialog:” for 11-15 utterances, and “very poor quality dialog:” for 16-20 utterances. Below is an example of distinct initial prompt tokens for the highest quality:

[3 utterances] Seed<|assistant|> Excellent quality dialog:

[9 utterances] Seed<|assistant|> Poor quality dialog:

Since longer dialogs often contain coherence issues, topic drifts, and repetitions, dialog length serves as a useful quality proxy. During inference, we assume that SOLID-RL has learned to distinguish higher-quality data distributions, considering that our objective is to exclusively generate high-quality responses by prompting them with:

[for any dialog] Seed<|assistant|> Excellent quality dialog:

3.3 Dataset Creation

Using SOLID to generate dialog data results in a 300k conversational dataset called **SOLISpeak**. Subsequently, we train SOLID-RL on SOLISpeak to generate dialogs more efficiently, resulting in a second dataset of 300k conversations called **SOLITurbo**. Both datasets are large-scale and intent-aware, with SOLITurbo benefiting from the enhanced data quality and efficiency provided by the fine-tuning and optimization process of SOLID-RL. Details for both datasets are presented in Table 4.

4 Experimental Setup

Training and Evaluation. We demonstrate the utility of SOLISpeak and SOLITurbo by using them to train IP models and evaluating their performance on existing human-annotated datasets. Given the large scale of our generated datasets, a small subset of the data is sufficient for effectively training a model. Therefore, we explore some training scenarios in which our generated datasets can be used independently or jointly with human-generated data to improve the trained models. We evaluate the synthetic data in two training scenarios:

- **Scenario 1: Synthetic-Only Training.** In this scenario, the models are trained exclusively on our synthetically generated data. We can utilize the entire dataset or strategically selected subsets, as using subsets reduces computational costs and potentially improves model effectiveness by fo-

cusing on the most relevant instances. To create these subsets, we employ a retrieval-based strategy: we randomly pick instances from the existing real dialog datasets, and then retrieve the top- k most relevant dialogs from SOLISpeak and SOLITurbo based on a dot product metric. In other words, we treat the real dialogs as queries, and the synthetic dialogs as a document collection. Note that in this scenario, the actual dialogs are used for retrieval purposes only, and not for model training. We use two retrieval models: *BM25* (Robertson and Zaragoza, 2009; Elasticsearch, 2023) and a *BERT-based bi-encoder* (Transformers, 2023). We use 1,760 dialogs from the training partition of MSDialog-intent as queries, and retrieve $k = 15$ dialogs from SOLISpeak/SOLITurbo for each, yielding 26,400 dialogs. After removing duplicates, we end up with 25,779 dialogs.⁴

- **Scenario 2: Mixed Real-Synthetic Training.** In this scenario, we train a model using a combination of human-labeled dialogs and synthetically generated data. We employ three sampling strategies to create balanced training sets. As in the previous paragraph, sampling is conducted on the synthetic data relative to the human-labeled data, to address label imbalance in the latter. *SeqInt-Bal* samples from synthetic data (SOLISpeak/SOLITurbo) to ensure that, when combined with the human-labeled dialogs, the final dataset’s intent sequence labels are balanced. *SeqInt-Bal* ensures that all intent sequences have at least 1000 data samples. *Int-Bal* works similarly to *SeqInt-Bal*, but focuses on balancing the individual intents rather than full intent sequences, ensuring that all intents are represented by the same number of samples. Finally, to demonstrate the effectiveness of balanced sampling, we compare the results with *Random-EQ* that randomly samples a fixed number of data points, ensuring a one-to-one correspondence.

LLM for generating data. We chose Zephyr-7b-beta as our LLM for data generation, as it was one of the most competitive and robust 7B models at the time of our experiments.

LLM supervised fine-tuning. We use Huggingface’s trainers for Supervised Fine-Tuning (SFT) of SOLID-RL. We use QLora 4bit (Detters et al.,

2023) for efficient fine-tuning. We employ a batch size and gradient accumulation of 4, and train for 4000 steps for both SFT and DPO, which takes about 7 hours to be trained on a single NVIDIA GPU P6000 40GB GPU memory.

Datasets. We use the MSDialog-Intent (Qu et al., 2019a) dataset to extract real-world user intents, ensuring a realistic order and distribution. We also use it for evaluating the IP models trained on our synthetic dataset, along with MANtIS (Penha et al., 2019b).

Intent Prediction models. For intent prediction, we use the IP models: (i) *cnn-context-rep*, a top-performing model on MSDialog; (ii) BERT, which has shown state-of-the-art results on MANtIS; (iii) We also fine-tune T5 for intent prediction, adapting it for intent generation; (iv) *zephyr-7b-beta* in a few-shot intent prediction setup. Further details and implementations link can be found in Appendix A.3.

Performance Assessment. We evaluate the effectiveness of our framework using MSDialog-intent dataset. We also assess our RL-trained LLM using the benchmark datasets from the Huggingface leaderboard (Huggingface, 2024). Although this leaderboard does not directly measure the conversational quality of chat models, it provides a useful signal to validate whether fine-tuning has introduced regressions on the base model’s reasoning and truthfulness capabilities. Further details can be found in Appendix A.9. Moreover, we verified there is no data leakage between Zephyr and Mistral, and our chosen evaluation datasets. More details can be found in Appendix A.10.

5 Results

We address the following research questions:

- **(RQ1)** How effective is SOLISpeak in training intent predictors across different scenarios with either zero or partial availability of human data, and to what extent does using the SOLITurbo dataset enhance the effectiveness of these intent predictors compared to SOLISpeak?
- **(RQ2)** What is the impact of self-seeding on SOLID compared to alternative external-seeding methods?
- **(RQ3)** How does SOLID-RL’s ability to generate a full intent-aware dialog from a single zero-shot instruction improve computational efficiency compared to SOLID’s utterance-by-utterance dialog generation?

⁴We experimented multiple values of $k \in [5, 20]$ in increments of 5, and found $k = 5$ as optimal, although the effectiveness across values was relatively similar.

Table 2: The evaluation results of the ‘cnn-context-rep’ model (Qu et al., 2018a) on the MSDialog-intent dataset with utilizing all designed sampling methods for each scenario. We also trained on MANtIS as reference. The terms ‘# of SOLISpeak’ and ‘# of MS’ denote the number of dialogs sampled from our synthetically generated dataset (SOLISpeak) and the human-annotated MSDialog-intent dataset, respectively. ‘P’ and ‘F1’ refer to Precision and F1 score.

Sampling method	# of SOLISpeak	# of MS	P	F1	
				Micro	Macro
Cross-Dataset Evaluation					
MANtIS	0	0	.5324	.5523	.4871
Scenario 1: Synthetic-Only Training					
—	316,697	0	.7167	.7197	.6406
Bi-Encoder	25,779	0	<u>.7626</u>	<u>.7989</u>	<u>.7228</u>
BM25	25,779	0	.7600	.7745	.7198
Scenario 2: Mixed Real-Synthetic Training					
—	0	1,760	.7321	.7562	.6713
SeqInt-Bal	111,889	1,760	.7641	.8076	.7246
Int-Bal	19,635	1,760	.7613	.7857	.7211
Random-Eq	1,760	1,760	.7402	.7640	.6940

Main results (RQ1). As a preliminary quality evaluation of the generated dialogs by SOLID, we manually reviewed a sample of 100 dialogs to ensure alignment between intents and utterances. Furthermore, table 2 presents the impact of different sampling methods within each scenario on MSDialog-intent. We use the state-of-the-art IP method for this dataset, aiming to compare the utility of generated data across different sampling approaches and offer insights rather than evaluate various IP models. Our findings indicate that the most effective sampling method is sequence of intents balancing (SeqInt-Bal), which incorporates a higher proportion of SOLID data. Intent balancing is the second-best method, using approximately six times less SOLISpeak data. We find that for all sampling methods, training the IP model on a combination of human and SOLISpeak leads to higher effectiveness than using human data alone. Furthermore, we find that using a bi-encoder for sampling ranks as the second most effective approach. An interesting observation is that Random-Eq outperforms scenarios where only the human-labeled data or only SOLID-generated data are used. Training exclusively on SOLISpeak achieves slightly lower effectiveness than using human data, demonstrating SOLISpeak’s strong utility in cases where human data is unavailable or restricted for IP model training. Moreover, we observe that the IP model

trained solely on our dataset (Scenario 1) achieves a precision of 0.7167, compared to 0.5324 for a model trained on the MANtIS human-annotated dataset. This shows that training on our large and diverse synthetic datasets can build effective IP models that are generalizable to other datasets.

To further explore **RQ1**, we investigate utility of the SOLISpeak and SOLITurbo datasets on an additional dataset and across various IP models. Focusing on Scenario 2, where a small human-generated dataset is available, we apply the most effective sampling method, SeqInt-Bal, alongside Random-EQ. We also consider Zephyr in a few-shot setup to determine whether it can serve as a slower but effective IP method. Our main observations from Table 3 are: (1) utilizing our synthetic dataset improves all three of the supervised IP methods compared only using human data by a large margin; this proves the usefulness of our dataset in downstream tasks; (2) SOLID-RL generates more valuable data for IP, as the T5 model trained on a combination of human and SOLID-RL data achieves the highest effectiveness compared to other IP methods; (3) Zephyr cannot predict the intents effectively in a few-shot setup, showing the challenge of prediction for LLMs, even though they can generate content following specific intents. This aligns with previous studies’ findings (Zhao et al., 2023b; Hu et al., 2023) and can be due to the fact that following instructions is a more natural task for LLMs than acting as an IP method in a few-shot setup.

To better assess the necessity of constructing an intent-annotated dialog dataset for IP prediction, we evaluate Zephyr-7b and GPT-4o (Zhao et al., 2024) in a few-shot setting as intent predictors. Table 5 shows that intent prediction models trained on SOLID and SOLID-RL significantly outperform strong few-shot models such as GPT-4o, even with 20 provided examples. We would also like to highlight the efficiency and cost benefits of using the T5 model trained on SOLID or SOLID-RL data, which has only 220 million parameters. Deploying such a model is much more feasible compared to using larger LLMs with 70B or 405B parameters. As a result, our approach is not only more effective but also far more efficient in practical applications. While the T5-base (SOLID-RL) model performs significantly better than the other models, there are ‘hard’ dialogs on which all models fail. We present an example of such a dialog in Appendix A.11.

Analyzing self-seeding impact (RQ2). We ex-

Table 3: The Results of intent predictors on the most effective sampling method (SeqInt-Bal) and the most simple while effective (RandomEq) in addition to the using only human as training data. ‘P’ refers to the precision.

Model	Sampling method	P	MSDialog		MANtIS		
			F1-Micro	F1-Macro	P	F1-Micro	F1-Macro
cnn-context-rep	SeqInt-Bal	.764	.808	.725	.801	.786	.625
	Random-Eq	.740	.764	.694	.784	.744	.595
	Only human	.732	.756	.671	.777	.736	.572
BERT-base	SeqInt-Bal	.810	.831	.743	.855	.814	.644
	Random-Eq	.762	.792	.711	.814	.772	.617
	Only human	.734	.758	.674	.779	.738	.591
T5-base	SeqInt-Bal	.839	.868	.771	.877	.833	.669
	Random-Eq	.791	.828	.737	.824	.812	.650
	Only human	.750	.772	.685	.780	.745	.587
T5-base on SOLID-RL	SeqInt-Bal	.846	.879	.786	.881	.839	.677
	Random-Eq	.807	.835	.746	.829	.814	.656
Zephyr-fewshot	3-shots	.416	.479	.405	.486	.437	.344

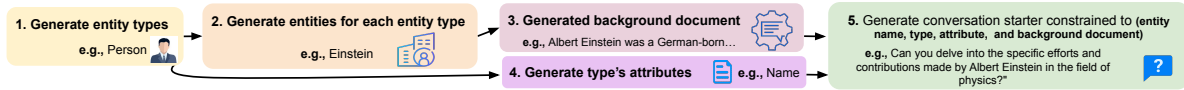


Figure 5: The illustration of seed generation through self-seeding via SOLID.

Model	Precision	F1 (Micro)	F1 (Macro)
GPT-4o (20-shots)	0.546	0.589	0.424
Zephyr-7b (3-shots)	0.416	0.479	0.405
T5-base (SOLID-RL)	0.846	0.879	0.786
T5-base (SOLID)	0.839	0.868	0.771

Table 4: Evaluation of LLMs in few-shot settings as IP predictors.

amine the impact of self-seeding in comparison to three alternative external seeding methods. For external seeding, we employ various sampling approaches and prompt an LLM to generate 10,000 intent-aware dialogs about 1,000 entities. This is achieved by first generating ten different questions for each entity to serve as the starting point of the conversation. The alternative external-seeding methods are as follows: *Top-most-popular entities*, we prompt the LLM with the top 1,000 most popular entities from Wikipedia; in *Sampling from popular entities*, we prompt the LLM with 1,000 entities randomly sampled from the top 10,000 most popular entities on Wikipedia; and for *Sampling random entities*, we randomly sample entities from Wikipedia. It is crucial to note that the sequence of intents remains consistent across this analysis. Table 5 shows that the IP models trained on data generated by the external seeding methods achieve lower effectiveness than those trained with self-seeding. This confirms the intuition behind self-seeding, which shows that it could improve the

Table 5: Comparative effectiveness of dialog generation using self-seeding vs. random selection of popular entities. The utterance classification model is T5 evaluated on MSDialog. ‘P’ refers to the precision.

	P	F1	
		Micro	Macro
Self-seeding (ours)	.7085	.7251	.6240
External-seeding methods			
Top-most-frequent entities	.6457	.6604	.5748
Sampling frequent entities	.6460	.6633	.5769
Sampling random entities	.5839	.6101	.4976

quality of data generated by the LLM.

SOLID-RL efficiency (RQ3). We evaluate the efficiency of SOLID-RL, which can generate an entire dialog given a single instruction, compared to SOLID’s utterance-by-utterance generation process that requires N LLM passes to generate a dialog with N turns. Using a batch size of 16, we measure the time required to generate 1,000 dialogs with each method. Our results indicate that SOLID takes 23,000 seconds to generate these dialogs, whereas SOLID-RL completes the task in just 2,000 seconds, making SOLID-RL nearly 12 times more efficient for intent-aware dialog generation.

6 Discussion

Multi-intent Self-instruction Analysis. We assess the effect of multi-intent self-instructing on IP models by generating dialogs with 10,000 seeds using SOLID, and comparing the self-instruction

Table 6: Analyzing the impact of self-instructing and length-based quality estimation for mixed-quality training in SOLID-RL. ‘P’ refers to the precision.

Model	P	F1	
		Micro	Macro
SOLID (ours)	.708	.725	.624
W/o self-instructing	.515	.537	.428
SOLID-RL (ours)	.741	.778	.68
W/o L-based MQ	.532	.554	.457

approach to an alternative rule-based method that combines multiple instructions with “and.” The top section of Table 6 reveals a significant performance difference between models trained on SOLID with self-instruction and those using the rule-based method, showing the potential of multi-intent self-instructing as a promising strategy.

Length-based mixed-quality training. Using the same 10,000 seeds as above, we examine the impact of LMQ on intent-aware dialog generation. To assess this, we train SOLID-RL both with and without LMQ’s class-conditioned input, starting from scratch in the latter case. For further details on the exclusion process, see Appendix A.8.2. Results in the bottom section of Table 6 reveal a significant advantage for SOLID-RL trained with LMQ, indicating that LMQ effectively optimizes and avoids low-quality dialogs impacting the LLM’s performance. It also shows that conditioning the model with LMQ for high-quality output during inference aids in generating higher-quality data.

7 Conclusion

This work presents a novel approach for automatically generating intent-aware, information-seeking dialog datasets through two zero-shot generation methods: SOLID and SOLID-RL. SOLID employs a novel zero-shot, multi-step process that uses self-seeding and self-instructing. SOLID-RL, optimized with supervised fine-tuning followed by DPO optimization, is trained on SOLID’s output. With the ability to generate dialogs from a single instruction, SOLID-RL achieves an efficiency level 11 times greater than SOLID’s utterance-by-utterance generation process. This efficiency is further enhanced by a unique length-based mixed-quality training method that dynamically assesses dialog quality. Leveraging these methods, we create two synthetic datasets that can significantly improve intent prediction accuracy. Despite LLMs’

current limitations in intent prediction, our results highlight their potential to enhance training datasets. Our generated datasets provide valuable resources for other related tasks such as next intent prediction and conversational search. Our self-seeding and self-instructing approaches are adaptable to various conversation types and languages with minimal modifications.

Limitations

In this paper, we demonstrated that LLMs can generate data to effectively train intent predictor models within the context of information-seeking dialogs. It is important to emphasize that this paper did not explore all aspects of evaluation, particularly the impact of biased information in the generated dialogs or the biases within the LLM’s conversational abilities, that shows through factually incorrect generations. In our study, we did not examine how these biases affect the performance of intent predictors, which could lead to a tendency for certain utterances to be predicted with biased intents. Factually incorrect generations in the generated data (as a result of LLM’s hallucination) are not likely to be harmful to the intent prediction models, as only information contained in human dialog will serve as input for the intent predictors in the test and real-world scenarios. Moreover, these predictors are limited to only the defined intents and cannot hallucinate new intents. We did not systematically investigate or quantify the impact of hallucinated data on the performance of intent predictor models and the potential bias they could introduce in the prediction. In the Appendix, we provide more information on this and estimate a generated entity that does not exist in sources like Wikipedia as a hallucination, but this is susceptible to errors. While our method intuitively could work effectively with other large language models (LLMs), we have not empirically tested its performance across a broad range of LLMs, as we aimed to limit the scope of this paper. We leave the comprehensive evaluation of our method using different LLMs to future work, where potential variations in performance and biases can be thoroughly explored. Our methodology can accommodate finer-grained taxonomies or additional annotations, thus its flexibility allows researchers to extend SOLID for more detailed tasks. Exploring these directions, including intent hierarchies and contextual annotations, is an exciting opportunity for future work.

References

- Zahra Abbasiantaeb, Yifei Yuan, Evangelos Kanoulas, and Mohammad Aliannejadi. 2023. Let the llms talk: Simulating human-to-human conversational qa via zero-shot llm-to-llm interactions. *arXiv preprint arXiv:2312.02913*.
- Amin Abolghasemi, Zhaochun Ren, Arian Askari, Mohammad Aliannejadi, Maarten de Rijke, and Suzan Verberne. 2024. CAUSE: Counterfactual assessment of user satisfaction estimation in task-oriented dialogue systems. In *ACL 2024: 62nd Annual Meeting of the Association for Computational Linguistics*. ACL.
- Gaurav Arora, Shreya Jain, and Srujana Merugu. 2024. Intent detection in the age of llms.
- Arian Askari, Mohammad Aliannejadi, Amin Abolghasemi, Evangelos Kanoulas, and Suzan Verberne. 2023a. Closer: Conversational legal longformer with expertise-aware passage response ranker for long contexts. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM '23*, page 25–35, New York, NY, USA. Association for Computing Machinery.
- Arian Askari, Mohammad Aliannejadi, Chuan Meng, Evangelos Kanoulas, and Suzan Verberne. 2023b. Expand, highlight, generate: RL-driven document generation for passage reranking. In *EMNLP*, pages 10087–10099.
- Weixin Chen and Bo Li. 2024. Grath: Gradual self-truthifying for large language models. *arXiv preprint arXiv:2401.12292*.
- Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*.
- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try ARC, the AI2 reasoning challenge.
- Pierre Colombo, Emile Chapuis, Matteo Manica, Emmanuel Vignon, Giovanna Varni, and Chloe Clavel. 2020. Guiding attention in sequence-to-sequence models for dialogue act prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7594–7601.
- Wentao Deng, Jiahuan Pei, Zhaochun Ren, Zhumin Chen, and Pengjie Ren. 2023. Intent-calibrated self-training for answer selection in open-domain dialogues. *Transactions of the Association for Computational Linguistics*, 11:1232–1249.
- Yang Deng, Wenxuan Zhang, Wai Lam, Hong Cheng, and Helen Meng. 2022. User satisfaction estimation with sequential dialogue act modeling in goal-oriented conversational systems. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 2998–3008. ACM.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.
- Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations.
- Elasticsearch. 2023. Elasticsearch: The official distributed search & analytics engine. <https://www.elastic.co/guide/en/elasticsearch/reference/current/index-modules-similarity.html#bm25h>.
- Sarik Ghazarian, Behnam Hedayatnia, Alexandros Pappangelis, Yang Liu, and Dilek Hakkani-Tur. 2022. What is wrong with you?: Leveraging user sentiment for automatic dialog evaluation. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 4194–4204. Association for Computational Linguistics.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding.
- Zhiyuan Hu, Yue Feng, Anh Tuan Luu, Bryan Hooi, and Aldo Lipani. 2023. Unlocking the potential of user feedback: Leveraging large language model as user simulators to enhance dialogue system. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM '23*, page 3953–3957, New York, NY, USA. Association for Computing Machinery.
- Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*.
- Huggingface. 2024. Huggingface open llm leaderboard: The huggingface open llm leaderboard aims to track, rank and evaluate open llms and chatbots. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard.
- Hideaki Joko, Shubham Chatterjee, Andrew Ramsay, Arjen P. de Vries, Jeff Dalton, and Faegheh Hasibi. 2024. Doing personal LAPS: llm-augmented dialogue construction for personalized multi-session conversational search. *CoRR*, abs/2405.03480.

- Gangwoo Kim, Sungdong Kim, Kang Min Yoo, and Jaewoo Kang. 2022. Generating information-seeking conversations from unlabeled documents. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2362–2378.
- Brendan King and Jeffrey Flanigan. 2024. [The power of the noisy channel: Unsupervised end-to-end task-oriented dialogue with llms](#). *CoRR*, abs/2404.15219.
- Atharva Kulkarni, Bo-Hsiang Tseng, Joel Ruben Antony Moniz, Dhivya Piraviperumal, Hong Yu, and Shruti Bhargava. 2024. Synthdst: Synthetic data is all you need for few-shot dialog state tracking. *arXiv preprint arXiv:2402.02285*.
- Harshit Kumar, Arvind Agarwal, and Sachindra Joshi. 2019. A practical dialogue-act-driven conversation model for multi-turn response selection. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1980–1989.
- Zekun Li, Wenhui Chen, Shiyang Li, Hong Wang, Jing Qian, and Xifeng Yan. 2022. [Controllable dialogue simulation with in-context learning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 4330–4347. Association for Computational Linguistics.
- I-Fan Lin, Faegheh Hasibi, and Suzan Verberne. 2024. [Generate then refine: Data augmentation for zero-shot intent detection](#).
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods.
- Zhenhua Liu, Tong Zhu, Jianxiang Xiang, and Wenliang Chen. 2024a. [Controllable and diverse data augmentation with large language model for low-resource open-domain dialogue generation](#). *CoRR*, abs/2404.00361.
- Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Chankyu Lee, Mohammad Shoeybi, and Bryan Catanzaro. 2024b. [Chatqa: Building GPT-4 level conversational QA models](#). *CoRR*, abs/2401.10225.
- Mediawiki. 2023. Wikipedia api search: Get request to perform an advanced search for wiki pages by title or content text match. <https://www.mediawiki.org/wiki/API:Search>.
- Chuan Meng, Mohammad Aliannejadi, and Maarten de Rijke. 2023. System initiative prediction for multi-turn conversational information seeking. In *CIKM*, pages 1807–1817.
- Fengran Mo, Bole Yi, Kelong Mao, Chen Qu, Kaiyu Huang, and Jian-Yun Nie. 2024. [Convsgd: Session data generation for conversational search](#). In *Companion Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, Singapore, May 13-17, 2024*, pages 1634–1642. ACM.
- Gustavo Penha, Alexandru Balan, and Claudia Hauff. 2019a. Introducing mantis: a novel multi-domain information seeking dialogues dataset. *arXiv preprint arXiv:1912.04639*.
- Gustavo Penha, Alexandru Balan, and Claudia Hauff. 2019b. [Introducing mantis: a novel multi-domain information seeking dialogues dataset](#).
- C. Qu, L. Yang, W. B. Croft, Y. Zhang, J. Trippas, and M. Qiu. 2019a. User intent prediction in information-seeking conversations. In *CHIIR '19*.
- Chen Qu, Liu Yang, W. Bruce Croft, Johanne R. Trippas, Yongfeng Zhang, and Minghui Qiu. 2018a. [Analyzing and characterizing user intent in information-seeking conversations](#). In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 989–992. ACM.
- Chen Qu, Liu Yang, W. Bruce Croft, Johanne R. Trippas, Yongfeng Zhang, and Minghui Qiu. 2018b. [Analyzing and characterizing user intent in information-seeking conversations](#). In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 989–992. ACM.
- Chen Qu, Liu Yang, W. Bruce Croft, Yongfeng Zhang, Johanne R. Trippas, and Minghui Qiu. 2019b. [User intent prediction in information-seeking conversations](#). In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval, CHIIR 2019, Glasgow, Scotland, UK, March 10-14, 2019*, pages 25–33. ACM.
- Renyi Qu, Lyle H. Ungar, and João Sedoc. 2023. [Conditioning on dialog acts improves empathy style transfer](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 13254–13271. Association for Computational Linguistics.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. [Direct preference optimization: Your language model is secretly a reward model](#).
- Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. [Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset](#). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8689–8696. AAAI Press.

- Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.
- Gaurav Sahu, Pau Rodriguez, Issam Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. 2022. Data augmentation for intent classification with off-the-shelf large language models. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 47–57, Dublin, Ireland. Association for Computational Linguistics.
- Chris Samarinas, Pracha Promthaw, Atharva Nijasure, Hansi Zeng, Julian Killingback, and Hamed Zamani. 2024. Simulating task-oriented dialogues with state transition graphs and large language models. *CoRR*, abs/2404.14772.
- Pararth Shah, Dilek Hakkani-Tür, Gökhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry P. Heck. 2018. Building a conversational agent overnight with dialogue self-play. *CoRR*, abs/1801.04871.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.
- Sentence Transformers. 2023. multi-qa-mpnet-base-dot-v1: a sentence-transformers model. <https://huggingface.co/sentence-transformers/multi-qa-mpnet-base-dot-v1>.
- Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment.
- Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2024. Openchat: Advancing open-source language models with mixed-quality data. In *The Twelfth International Conference on Learning Representations*.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Zequ Wu, Ryu Parish, Hao Cheng, Sewon Min, Prithviraj Ammanabrolu, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. Inscit: Information-seeking conversations with mixed-initiative interactions. *Transactions of the Association for Computational Linguistics*, 11:453–468.
- Liu Yang, Minghui Qiu, Chen Qu, Cen Chen, Jiafeng Guo, Yongfeng Zhang, W. Bruce Croft, and Haiqing Chen. 2020. Iart: Intent-aware response ranking with transformers in information-seeking conversation systems. In *Proceedings of The Web Conference 2020, WWW '20*, page 2592–2598, New York, NY, USA. Association for Computing Machinery.
- Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2022. Generate rather than retrieve: Large language models are strong context generators. *arXiv preprint arXiv:2209.10063*.
- Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*.
- Hamed Zamani, Johanne R Trippas, Jeff Dalton, Filip Radlinski, et al. 2023. Conversational information seeking. *Foundations and Trends® in Information Retrieval*, 17(3-4):244–456.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence?
- Justin Zhao, Timothy Wang, Wael Abid, Geoffrey Angus, Arnav Garg, Jeffery Kinnison, Alex Sherstinsky, Piero Molino, Travis Addair, and Devvret Rishi. 2024. Lora land: 310 fine-tuned llms that rival gpt-4, A technical report. *CoRR*, abs/2405.00732.
- Weixiang Zhao, Yanyan Zhao, Xin Lu, Shilong Wang, Yanpeng Tong, and Bing Qin. 2023a. Is chat-gpt equipped with emotional dialogue capabilities? *arXiv preprint arXiv:2304.09582*.
- Yukun Zhao, Lingyong Yan, Weiwei Sun, Chong Meng, Shuaiqiang Wang, Zhicong Cheng, Zhaochun Ren, and Dawei Yin. 2023b. DiQAD: A benchmark dataset for open-domain dialogue quality assessment. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15128–15145, Singapore. Association for Computational Linguistics.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. 2023. Lmsys-chat-1m: A large-scale real-world llm conversation dataset.

A Appendix

A.1 Intents

In the context of an information-seeking conversation, we refer to intent as the role or function that an utterance plays in the conversation. It can include acts like requesting information, giving positive/negative feedback, requesting further details, or repeating the question. Table 7 shows the list of 12 unique utterances that exist in the SOLID (ours) and MSDialog datasets. Intents can be used to help the chatbot understand the context and purpose of a user’s message and to determine how to respond appropriately. While similar in intuition, FQ (Follow-Up Question) and CQ (Clarifying Question) are distinct in their definitions, as established in prior studies. CQ focuses on clarification regarding specific aspects of a prior utterance, while FQ expands the discussion to related but broader topics. We followed these existing definitions to maintain consistency with established research and thus be able to compare with previous and future work.

Table 7: Intent taxonomy and distribution in MSDialog

CodeLabel	Description	%
OQ	Original Question	The first question from the user to initiate the dialog. 13
RQ	Repeat Question	Other users repeat a previous question. 3
CQ	Clarifying Question	User or agent asks for clarifications. 4
FD	Further Details	User or agent provides more details. 14
FQ	Follow Up Question	User asks for follow up questions about relevant issues. 5
IR	Information Request	Agent asks for information from users. 6
PA	Potential Answer	A potential answer or solution provided by agents. 22
PF	Positive Feedback	User provides positive feedback for working solutions. 6
NF	Negative Feedback	User provides negative feedback for useless solutions. 4
GG	Greetings/Gratitude	Greetings or expressing gratitude. 22
JK	Junk	No useful information in the utterance. 1
O	Others	Utterances cannot be categorized using other classes. 1

A.2 Manually-crafted instructions for User and Agent Intents

A total of 24 manually crafted instructions were created, covering 12 distinct intents, with separate instructions for both the user and agent sides, as shown in Tables 9 and 8. We want to note that the difference between the FQ and CQ is very fine, in linguistic nuance:

- **Further Question (FQ):** A question that builds on the previous information, i.e., it is a new question. This is emphasized by using the keyword **further**. Example: “How does this relate with the concept of supervised learning?” where **this** refers to the information in the previous turn.
- **Clarification Question (CQ):** A question that asks for more rephrasing, i.e., clarification, of the previous turn, because the user/sys-

tem cannot understand the meaning behind it. Example: “What exactly do you mean by image classification?” where the subject of the previous turn was still **image classification**.

The prompts were carefully engineered to capture this difference, and unfortunately, using the straight definition from Table 7 (which seems intuitive as human readers) did not yield satisfactory generations.

Table 8: Manually-crafted intent instructions. Actor type: Agent

Intent	Instruction
CQ	Reply with one follow-up response in conversation style.
FD	Reply with further details in conversation style.
GG	Continue the conversation by expressing gratitude for the user’s questions.
PA	Provide a potential solution or answer in conversation style.
IR	Ask the user to provide relevant information needed for their previous question.
OQ	Formulate an original question posed by an agent.
FQ	Formulate a follow-up question from an agent, seeking further clarification or information.
RQ	Now you are talking from the point of view of a third participant in the conversation. Repeat Question:
PF	Express satisfaction and appreciation for the conversation.
NF	Convey dissatisfaction for the previous response.
JK	Reply with gibberish information. It can contain emojis.
O	Reply with a system error. Return N/A

A.3 Details of intent predictors

We fine-tune BERT as the intent predictor model with a cross-entropy loss function. We fine-tune T5 for a seq-to-seq generation task with a cross-entropy loss function, using the prefix of "multi-label classification: " prompted to the input. For cnn-context-rep (Qu et al., 2019a), we utilized the released code by the original paper and used the most effective reported hyperparameter. The implementation of the baselines are available at baseline folder of our repository which can be accessed at [baselines folder](#).

A.4 Impact of hallucination

We categorize an entity as likely hallucinated if it cannot be located within Wikipedia. To facilitate this identification process, we leverage the

Table 9: Manually-crafted intent instructions. Actor type: User

Intent	Instruction
CQ	Reply with one question asking for clarification in conversation style.
FD	Reply with more details in conversation style.
GG	Continue the conversation by expressing gratitude for the agent’s help.
PA	Provide a potential solution or answer in conversation style.
IR	Reply with relevant information.
OQ	Formulate the first question posed by a user that initiates a QA dialog.
FQ	Formulate a follow-up question from a user, seeking further clarification or information.
RQ	Now you are talking from the point of view of a third participant in the conversation. Repeat Question:
PF	Express satisfaction and appreciation for a working solution.
NF	Convey dissatisfaction for the previous response.
JK	Reply with gibberish information. It can contain emojis.
O	Reply with a system error. Return N/A

Wikipedia search API (Mediawiki, 2023). It is important to acknowledge that our methodology, being automated, would inevitably generate hallucinated content. In Table 10, we examine the influence of hallucinated entities on subsequent tasks. Surprisingly, the results show that including hallucinated entities leads to higher effectiveness for intent prediction. We believe the reason lies in the nature of the task itself: intent prediction focuses on understanding the intent of the text rather than ensuring the factually correctness of the text.

Table 10: Analyzing impact of hallucination in SOLISpeak as training source on T5 intent predictor evaluated on MSDialog dataset.

	MSDialog		
	P	F1	
		Micro	Macro
Only hallucinated	.7295	.7498	.6692
Only non-hallucinated	.7289	.7502	.6684
Full	.7321	.7580	.6710

A.5 Self-seed generation process

Figure 16 presents a detailed illustration of the seed generation process. Additionally, Table 12 provides all 98 distinct entity types we generated for SOLISpeak and SOLITurbo. Entity types that look like duplicates are removed.

A.6 Prompts

We have provided the content of all of the prompts used in our experiments. Our prompts are designed empirically, based on a comprehensive analysis of dialogue quality across various prompt configurations.

A.6.1 Prompt for generating entity types

We utilize a simple prompt asking ‘Provide a list of 100 entity types.’ to generate entity types.

A.6.2 Prompts for generating entity names

We empirically figured out that this task cannot be addressed effectively without providing a one-shot example to the LLM. Therefore, we provide 10 examples of generated entities, and ask the LLM to generate 100 entities. We exclude lines with more than 20 characters and remove special characters. The prompt is shown in Figure 6.

A.6.3 Prompt for generating conversation starters

The prompt for generating a conversation starter is shown in Figure 7.

A.6.4 Prompt for building self-instruct instruction

When an utterance has more than one associated intent, we ask the LLM to generate its own instruction by merging the manually-crafted individual instructions presented in Tables 8 and 9. The prompt is shown in Figure 8.

A.6.5 Prompt for generating an intent-aware conversation

We combine the base prompt with the intent instructions. The base prompt is: ‘I will give you an entity, its type, a background document, a conversation history that ends in a...’. We empirically tested various prompts and refined our choices using ChatGPT, aiming to identify relatively the most effective prompt. The prompt we used was selected based on preliminary experiments that showed promising results in achieving the desired information-seeking dialog compared to other tested prompts. Using this prompt, we

ask the LLM to generate the next utterance based on the current conversation history. Our current approach involves a one-way forward pass to generate the dialog sequentially. To further enhance data quality, we propose a backward pass—starting from the final utterance and working back to the first. In this reverse process, we would prompt the LLM to rewrite each utterance so that it better aligns with the intended next utterance, ensuring that the assigned intents are better reflected.

Example1:
Instruction: Provide a list of at least 10 entities categorized as 'Person' whose names begin with the letter 'A' Please include a new line after each entity.
Entities:
 Abraham Lincoln
 Albert Einstein
 Alexander Graham Bell
 Amelia Earhart
 Anne Frank
 Anton Chekhov
 Arthur Conan Doyle
 Audrey Hepburn
 Auguste Rodin
 Abraham Lincoln

Example 2:
Instruction: Provide a list of at least 100 entities categorized as '{entity_type}' whose names begin with the letter '{letter}' Please include a new line after each entity.
Entities:

Figure 6: Prompt for generating entity names.

Entity name: {entity_name}
Entity description: {entity_description}
Entity type: {entity_type}
Attribute: {attribute}
 Come with a specific question that can initiate an information-seeking conversation about the information above.
Question:

Figure 7: Prompt for generating a conversation starter

Example1:
Instruction1: Reply with more details in conversation style. # Instruction for the first intent that the utterance should have. Here it is "further detail" intent.
Instruction2: Convey dissatisfaction for the previous response. # Instruction for the second intent that the same utterance should have. Here it is, for the "negative feedback" intent.
Merged Instruction: In a conversational style, reply with more details and express dissatisfaction for the previous response. # A merged instruction so the utterance can have multiple intents of "further detail and negative feedback".

Example2:
 {instructions_for_each_of_intents}
Merged Instruction:
 # For an utterance that should have multiple intents. The LLM generates a merged instruction based on instructions for each of the intents that are provided above. We then pass this instruction to the LLM itself for generating the utterance in intent-aware dialogue generation phase.

Figure 8: Prompt for generating self-instructions for utterances with multiple intents.

<[system]> You are a chatbot who generates dialogues, in which each utterance has a specific intent.
 Here is the definition of intents:
Intent: Name: "Original Question"
Definition: "The first question by a user that initiates the QA dialog."
Example: "If a computer is purchased with win 10 can it be downgraded to win 7?"
Intent: Name: "Repeat Question"
Definition: "Posters other than the user repeat a previous question."
Example: "I am experiencing the same problem ..."
Intent: Name: "Clarifying Question"
Definition: "Users or assistants ask for clarification to get more details." **Example:** "Your advice is not detailed enough. I'm not sure what you mean by ..."
Intent: Name: "Further Details"
Definition: "Users or assistants provide more details." **Example:** "Hi. Sorry for taking so long to reply. The information you need is ..."
Intent: Name: "Follow Up Question"
Definition: "Users ask follow-up questions about relevant issues."
Example: "Thanks. I really have one more simple question -- if I ..."
Intent: Name: "Information Request"
Definition: "Assistants ask for information from users." **Example:** "What is the make and model of the computer? Have you tried installing ..."
Intent: Name: "Potential Answer"
Definition: "A potential answer or solution provided by assistants."
Example: "Hi. To change your PIN in Windows 10, you may follow the steps below: ..."
Intent: Name: "Positive Feedback"
Definition: "Users provide positive feedback for working solutions."
Example: "Hi. That was exactly the right fix. All set now. Tx!"
Intent: Name: "Negative Feedback"
Definition: "Users provide negative feedback for useless solutions."
Example: "Thank you for your help, but the steps below did not resolve the problem ..."
Intent: Name: "Greetings/Gratitude"
Definition: "Users or assistants greet each other or express gratitude." **Example:** "Thank you all for your responses to my question ..."
Intent: Name: "Junk"
Definition: "There is no useful information in the post." **Example:** "Emojis. Sigh Thread closed by moderator ..."
Intent: Name: "Others"
Definition: "Posts that cannot be categorized using other classes."
 The user passes to you an entity name, entity background, and sequence of intents. Next, the dialogue you generate should have the requested sequence of intents. This is the format that the user passes the information to you:
Example of an input format by user: {entity name} {entity background document}
Sequence of intents: [utterance2 by assistant: further details, utterance3 by user: clarification question, utterance4 by assistant: positive feedback and potential answer].
 Utterance1 is already given by the user. "utterance by assistant" means the utterance should be generated in assistant style, and "utterance by user" means the utterance should be generated in user style. <[user]> **Entity Name:** {Entity name}
Entity Background Document:
 {Background document}
Sequence of Intents:
 [Utterance1 by user: Original Question, Utterance2 by assistant: {intent(s)}, ... UtteranceN by user: {intent(s)}].

Figure 9: Prompt for generating 'rejected' dialogs for DPO training.

A.7 Generating intent-aware dialogs.

We first initialize the dialog with the conversation starter. For each intent i in sequence \mathbf{S} , we prompt the LLM to generate a new conversational turn. An intent can be either simple or complex. A simple intent is $i \in \text{keys}(\mathbf{I})$, whereas a complex intent is a combination of two or more intents $i = \{i_1, i_2, \dots, i_n\}$, where $i_k \in \text{keys}(\mathbf{I})$ and $k \sim \text{Uniform}(1, 12)$. If the intent is complex, we apply the *multi-intent self-instructing* schema. For each intent in the sequence, the algorithm: (1) Retrieves its corresponding instruction or generates a merged instruction, (2) Concatenates the general instruction, entity, dialog history, and

new intent-specific instruction, (3) Passes the message to the LLM to generate a new utterance, (4) Applies post-processing on the new utterance, and (5) Inserts the new utterance at the end of the dialog. When the whole sequence **S** has been processed, the algorithm returns the complete generated dialog **D**.

A.8 Example of LMQ training instances.

Whether a sample is selected or rejected during LMQ-based DPO training, a textual prefix on dialog length is added to construct a class-conditioned input. When LMQ is not used, this prefix is omitted. Examples for each setup are provided below.

A.8.1 Including LMQ

We constrain the class-conditioned input only based on the dialog length. As a result, both the rejected and accepted samples have the same class-conditioned output "excellent quality dialog" due to their equal length in terms of the number of utterances.

excellent quality dialog:

Utterance 1:
Text: Can you delve into the specific efforts and contributions made by Albert Einstein in the field of physics?
Intent: original question

Utterance 2:
Text: Sure! Albert Einstein made groundbreaking contributions to physics, especially with his theory of relativity. What aspect would you like me to focus on or any specific topic you're interested in?
Intent: clarifying question

Utterance 3:
Text: Could you provide more details about his theory of relativity and how it revolutionized our understanding of space and time?
Intent: further details

Figure 10: A chosen training sample with LMQ.

excellent quality dialog:

Utterance 1:
Text: Can you delve into the specific efforts and contributions made by Albert Einstein in the field of physics?
Intent: original question

Utterance 2:
Text: Absolutely! Albert Einstein made revolutionary advancements in the field of physics, particularly through his theory of relativity.
Intent: clarifying question

Utterance 3:
Text: How is the weather?
Intent: further details

Figure 11: A rejected training sample with LMQ.

A.8.2 Excluding LMQ

Utterance 1:
Text: Can you delve into the specific efforts and contributions made by Albert Einstein in the field of physics?
Intent: original question

Utterance 2:
Text: Sure! Albert Einstein made groundbreaking contributions to physics, especially with his theory of relativity. What aspect would you like me to focus on or any specific topic you're interested in?
Intent: clarifying question

Utterance 3:
Text: Could you provide more details about his theory of relativity and how it revolutionized our understanding of space and time?
Intent: further details

Figure 12: A chosen training sample without LMQ.

Utterance 1: Text: Can you delve into the specific efforts and contributions made by Albert Einstein in the field of physics? Intent: original question
Utterance 2: Text: Absolutely! Albert Einstein made revolutionary advancements in the field of physics, particularly through his theory of relativity. Intent: clarifying question
Utterance 3: Text: How is the weather? Intent: further details

Figure 13: A rejected training sample without LMQ.

A.9 SOLID-RL on LLM benchmarks

We analyze the effect of using SOLID-RL through performance of Zephyr-7b-beta on multiple LLM benchmarks on the Open LLM Leaderboard (Huggingface, 2024). We gain insights into whether SOLID-RL introduces regressions on the base model’s reasoning and truthfulness capabilities. Table 11 provides an in-depth analysis of the impact of different training strategies for SOLID-RL on its performance. Our results demonstrate that SOLID-RL enhances Zephyr’s performance on both MMLU (Hendrycks et al., 2021) and Truthful QA (Lin et al., 2022) benchmarks. Our finding is relatively aligned with previous studies that show LLMs can enhance themselves when trained on high-quality self-generated data (Huang et al., 2022; Yuan et al., 2024; Chen and Li, 2024; Chen et al., 2024). The results indicate that excluding length-based mixed-quality training can significantly harm the general capability of SOLID-RL. We highlight the positive impact of DPO after SFT. We further investigate the effect of reversing chosen/rejected outputs in DPO training by using the output generated by SOLID as the rejected output during DPO optimization. This setup results in lower performance, confirming the chosen setup in SOLID-RL. We also experiment with training SOLID-RL alongside Zephyr’s training dataset in a multi-task setup; we found that this led to a decrease in the effectiveness of the LLM and to a generation of lower-quality dialogs. Our analysis suggests that SOLID-RL is the most robust method

compared to the variations investigated. Furthermore, we demonstrate the performance of SOLID-RL on 57 tasks on MMLU compared with Zephyr-7b, as illustrated in Figure 15. An example of an MMLU task is illustrated in Figure 14.

<p>A 6-sided die is rolled 15 times and the results are: side 1 comes up 0 times; side 2: 1 time; side 3: 2 times; side 4: 3 times; side 5: 4 times; side 6: 5 times. Based on these results, what is the probability of side 3 coming up when using Add-1 Smoothing?</p> <p>(A) 2/15 (B) 1/7 (C) 3/16 (D) 1/5</p>
--

Figure 14: A Machine Learning example of MMLU benchmark (Hendrycks et al., 2020).

A.10 Safeguard Data Leakage

To ensure there is no data leakage between the evaluation and training datasets used for Zephyr, we reviewed the Zephyr and Mistral training documentation (Tunstall et al., 2023). We checked if any of the BEIR collection data, such as MSDialog or MANTIS, exist in the training data of Mistral or Zephyr. We made sure that there is no mutual dataset within the used LLM training data, X (Tunstall et al., 2023) and our evaluation datasets.

A.11 Hard Examples

We include an example of a ‘hard’ dialog, which has been wrongly predicted by all models in our experimental setup:

User: Hi, the best way to view photos in Windows 10 is by using the Photos app. For us to better assist you, may we know how you are accessing your photos?

Agent: What changes have you made prior to getting this issue? What troubleshooting steps have you done so far? Can you provide a screenshot of the error message you’re getting?

User: We’ll be waiting for your response.

For this specific example, the predicted intents by each model are:

- **GPT-4o:** [‘Further Details’]
- **T5 (SOLID):** [‘Potential Answer’, ‘Clarification Question’]
- **T5 (SOLID-RL):** [‘Potential Answer’, ‘Clarification Question’]

while the ground truth intents are:

- [‘Information Request’, ‘Potential Answer’, ‘Clarification Question’]

Table 11: The evaluation results on the Open LLM Leaderboard (Huggingface, 2024), which measures the performance of LMs across four multiclass classification tasks: ARC (Clark et al., 2018), HellaSwag (Zellers et al., 2019), MMLU (Hendrycks et al., 2021), and Truthful QA (Lin et al., 2022). There are 57 tasks in MMLU - a detailed report on the effectiveness of SOLID-RL on all of them is shown in Figure 15. The bottom section of the table provides various analyses on SOLID-RL including: an ablation study on our length-oriented mixed-quality training, ablation study on DPO, reverting the preference data in DPO, and an analysis on multi-task training. The base model of SOLID-RL-7B is Zephyr-7B.

Model	Accuracy			
	Truthful QA	Hella Swag	MMLU	ARC
UltraLM-13B (Touvron et al., 2023)	52.00	61.32	50.45	57.25
Zephyr-7B (Tunstall et al., 2023)	57.44	84.52	61.44	62.03
SOLID-RL-7B (Ours)	58.13	<u>83.82</u>	61.68	<u>60.78</u>
Analysis on SOLID-RL-7B:				
W/o length-oriented MixedQ training	55.25	80.69	58.40	59.24
W/o DPO (i.e., only SFT)	53.77	81.72	60.47	59.73
Reverse chosen/rejected for DPO	52.36	79.82	60.14	58.96
W/ multi-task training	57.58	82.53	59.60	59.56

Algorithm 1 Generate Dialog

Input: G (global task instruction), I (intent instructions dictionary), E (entity), D_0 (conversation starter), S (sequence of intents)

Output: D (generated multi-turn dialog)

```

1: Initialize dialog  $D \leftarrow D_0$ 
2: for each intent  $i$  in sequence  $S$  do
3:   if  $\text{multiple\_intents}(i)$  then
4:     Set  $\text{instruction} \leftarrow \text{SelfInstruct}(i)$ 
5:   else
6:     Retrieve  $\text{instruction} \leftarrow I[s]$ 
7:   end if
8:    $\text{message} \leftarrow \text{Concat}(G, C, D, \text{instruction})$ 
9:    $\text{utterance} \leftarrow \text{GenUtterance}(\text{message})$ 
10:   $\text{utterance} \leftarrow \text{PostProcess}(\text{utterance})$ 
11:   $D \leftarrow \text{InsertNewTurn}(D, \text{utterance})$ 
12: end for
13: return  $D$ 

```

While it may seem clear to a human that the message encompasses all three intents in the ground truth, neither GPT-4o nor our models can successfully detect all of them. However, T5 models trained on SOLID and SOLID-RL are able to capture two out of the three ground truth intents, whereas GPT-4o fails to detect any of the correct intents.

Table 12: Diverse set of 98 entity types generated via SOLID.

Person	Historical Event	Beverage	Galaxy	Author	Deity
Book	Constellation	Artist	Mythology	Song	Astronomical Object
Politician	Mythical Creature	Animal	Natural Disaster	Actor	Supernatural Being
Game	Weather Phenomenon	Musician	Character (Fictional)	Gadget	Disease
Celebrity	Organization	Software	Medication	Actress	Company
App	Medical Procedure	Athlete	Government Agency	Website	Law
Historical Figure	Nonprofit Organization	Social Media Platform	Legal Case	Entrepreneur	Hospital
Device	Political Ideology	Inventor	University	Instrument	Social Movement
Scientist	Event	Tool	Philosophy	Mathematician	School
Furniture	Religion	Philosopher	Place	Clothing	Folklore
Explorer	Country	Artwork	Language	Poet	City
Painting	TM Brand	Photographer	Architectural Structure	Sculpture	Fashion Brand
Journalist	Festival	Cuisine	Style (Fashion, Art)	Activist	Conference
Recipe	Genre (Music, Film)	Sportsperson	Service	Scientific Concept	Technology
Sport	Product	Chemical Element	Programming Language	Team	Food
Particle	Movie	League	Vehicle	Planet	Historical Period
Plant	Star				

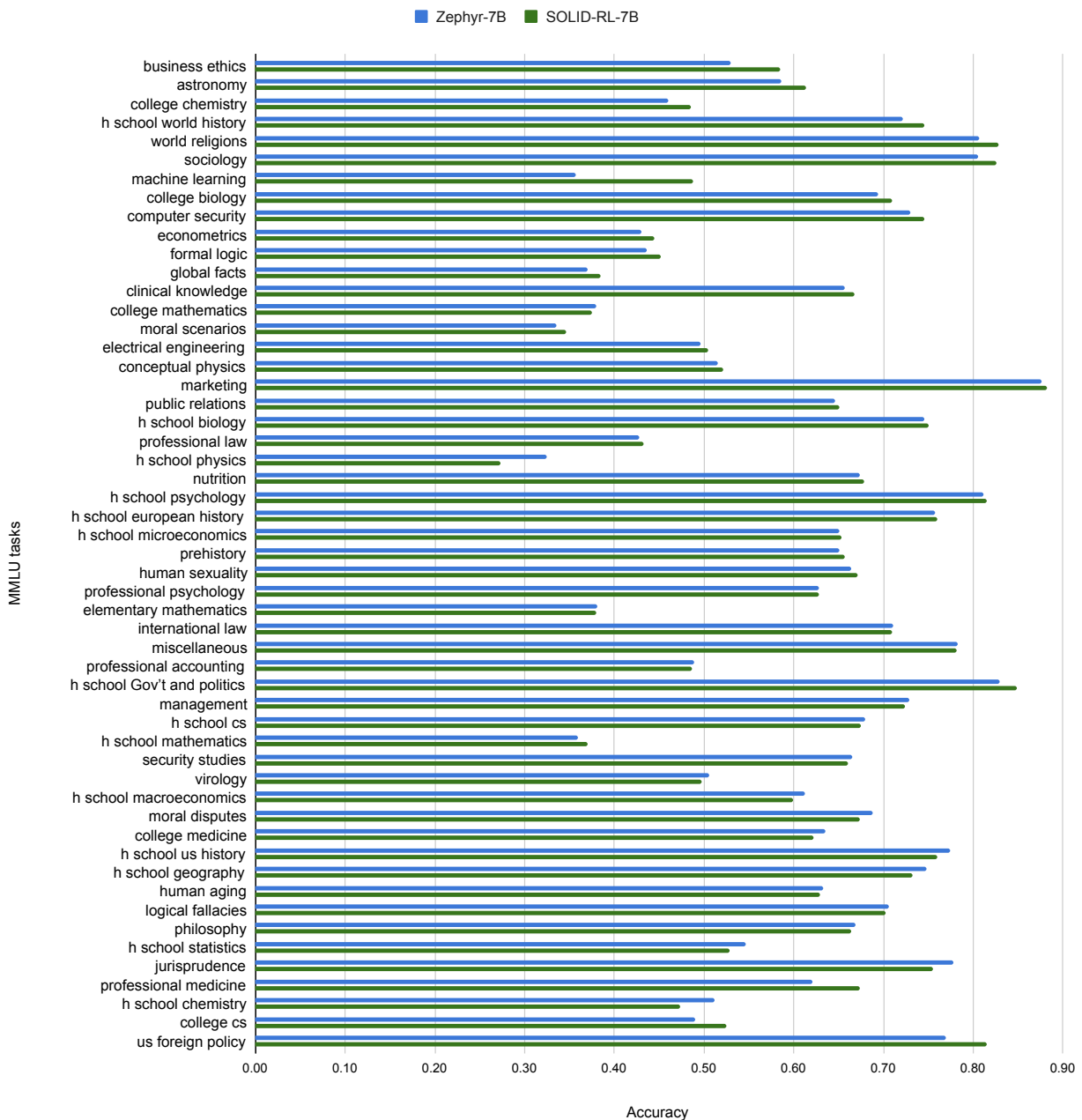


Figure 15: Effectiveness of SOLID-RL-7B vs. Zephyr-7B in terms of accuracy over tasks of Measuring Massive Multitask Language Understanding (MMLU) benchmark (Hendrycks et al., 2020).

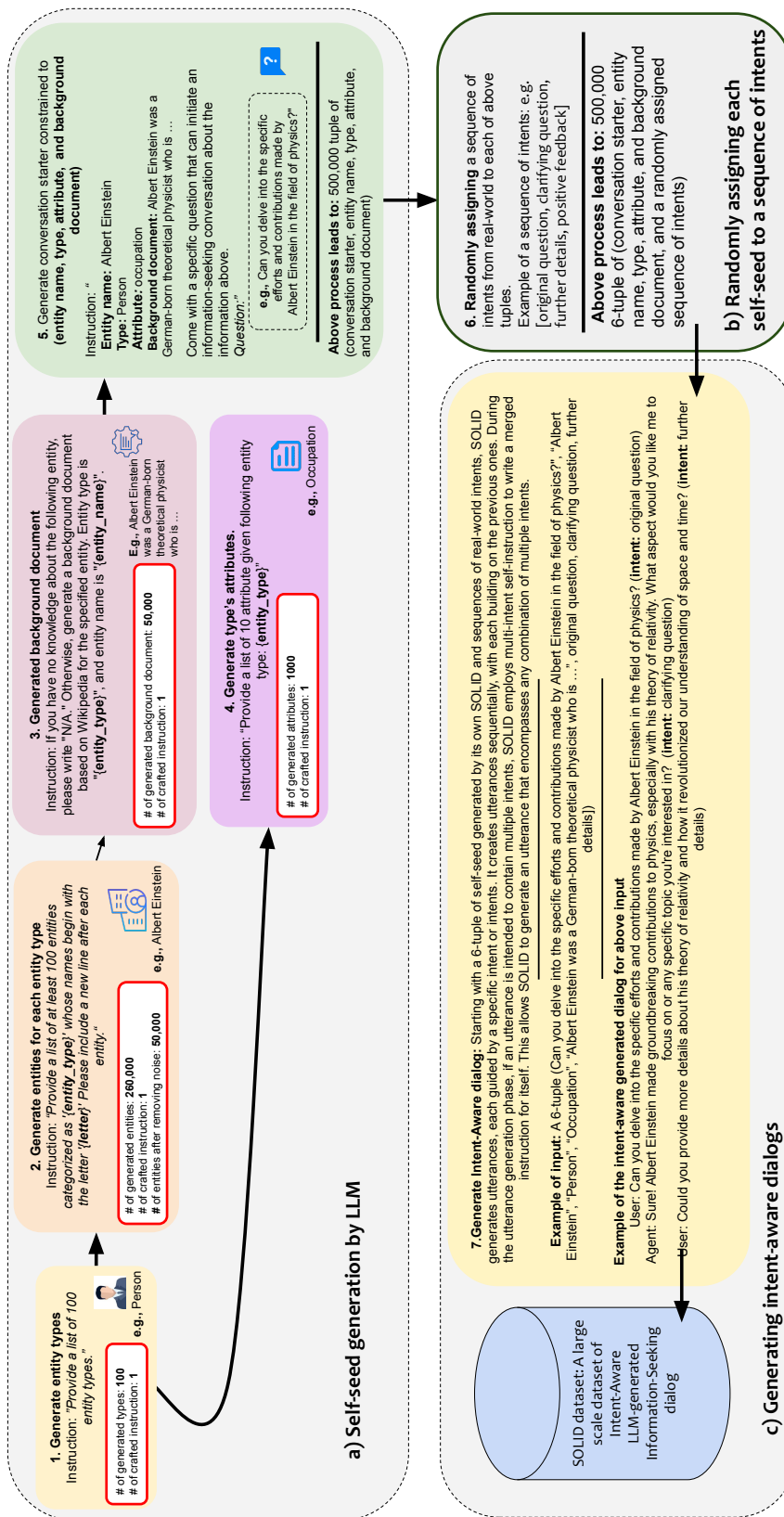


Figure 16: Comprehensive depiction of the self-seed generation process in SOLID, illustrated through examples and prompts, alongside a summary of intent-aware dialog generation leveraging these self-seeds.