Generate then refine: data augmentation for zero-shot intent detection

Lin, I.; Hasibi, F.; Verberne, S.; Al-Onaizan, Y.; Bansal, M.; Chen, Y.

Generate then Refine: Data Augmentation for Zero-shot Intent Detection

I-Fan Lin
Leiden University
Leiden, The Netherlands
i.lin@liacs.leidenuniv.nl

Faegheh Hasibi
Radboud University
Nijmegen, The Netherlands
faegheh.hasibi@ru.nl

Suzan Verberne
Leiden University
Leiden, The Netherlands
s.verberne@liacs.leidenuniv.nl

Abstract

In this short paper we propose a data augmentation method for intent detection in zero-resource domains. Existing data augmentation methods rely on few labelled examples for each intent category, which can be expensive in settings with many possible intents. We use a two-stage approach: First, we generate utterances for intent labels using an open-source large language model in a zero-shot setting. Second, we develop a smaller sequence-to-sequence model (the Refiner), to improve the generated utterances. The Refiner is fine-tuned on seen domains and then applied to unseen domains. We evaluate our method by training an intent classifier on the generated data, and evaluating it on real (human) data. We find that the Refiner significantly improves the data utility and diversity over the zero-shot LLM baseline for unseen domains and over common baseline approaches. Our results indicate that a two-step approach of a generative LLM in zero-shot setting and a smaller sequence-to-sequence model can provide high-quality data for intent detection.[1]

1 Introduction

Intent detection is a common component in task-oriented dialogue (TOD) systems. Its objective is to categorize user utterances into predefined classes of user intents (Ni et al., 2023). The advent of transformer-based models has significantly elevated the performance of intent detection, particularly when trained and evaluated within familiar domains and intents (Larson et al., 2019). However, when faced with previously unseen domains and intents, a considerable challenge emerges, primarily stemming from data scarcity.

Addressing this challenge involves maximizing the utility of limited training data and ensuring the adaptability of the trained intent classifier to novel intents. A combination of few-shot learning and meta-learning strategies has been employed in prior work (Zhang et al., 2020; Sauer et al., 2022; Wang et al., 2023; Cho et al., 2023).

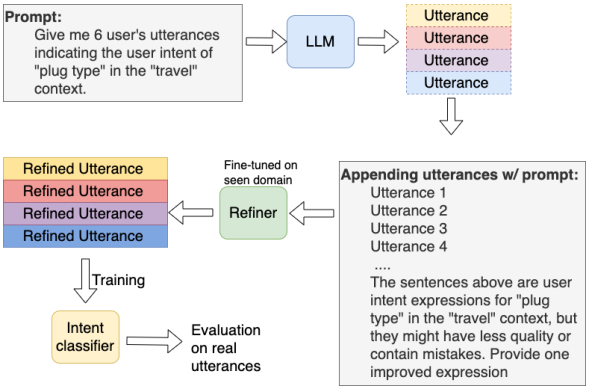

Figure 1: The pipeline of generating and refining utterances for out-of-domain intent detection.

A viable alternative solution for low-resource domain adaptation is augmenting and increasing the training data. This is achieved by direct utilization of a generative large language model (LLM) as a data generator (Xia et al., 2020; Lee et al., 2021; Sahu et al., 2022; Marceau et al., 2022; Fang et al., 2023). Synthetically generated data, however, is not always of high quality and this has direct effect on model performance. As a remedy, several works have attempted to filter out generated examples that are of low quality or relevance (Sahu et al., 2022; Meng et al., 2022; Lin et al., 2023).

In this paper, we study the setup where no labelled data is available for unseen domains (e.g., travel), while substantial number of user utterances with intent labels exist for seen domains (e.g., banking). Our goal is to create high-quality training data to train an intent classifier for completely unseen domains. In these unseen domains, the intent labels are known, but there are no utterances available. In line with previous work (Ye et al., 2022; Sahu et al., 2022; Meng et al., 2022; Lin et al., 2023), we obtain generated data from a generative LLM. The utility of this data, as measured by classification quality,

[1]Our code is available at: https://github.com/tom192180/Generate_then_Refine

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 13138–13146

lags behind a classifier trained on human utterances. One of the causes is that the LLM tends to generate the words from the intent label in the output utterances, which makes it less representative of real user's utterances. Therefore, we propose a Refiner model that transforms sub-optimal generated utterances into better ones. The goal of our Refiner model, trained on data from seen domains, is to enhance the performance of the intent classifier; see Figure 1 for illustration.

The main contribution of this paper is proposing a sequence-to-sequence learning method for enhancing the utility of LLM-generated utterances, while retaining sample size. This stands in contrast to the previous studies that utilize formula-based metrics for data selection to filter sub-optimal utterances (Meng et al., 2022; Lin et al., 2023).

In summary, the contributions of this paper are:

- Proposing a Refiner model for zero-shot intent classification and showing its effectiveness compared to state-of-the-art data augmentation approaches, including ChatGPT.
- Evaluating the Refiner (via extensive analysis and ablation study) as a compute-efficient model, showing that a smaller model, when trained on rich-resource domains, is capable of improving the output of a larger (7B-parameter) LLM in unseen domains.
- Showing the success of the Refiner in producing lexically diverse text, comparable to human data, thereby addressing the common issue of less diverse text generated by LLMs.

2 Related Work

Data augmentation is a commonly used solution to data scarcity in training conversational agents (Soudani et al., 2024). Recently, generating new examples directly from LLMs has emerged as a viable strategy (Meng et al., 2022; Fang et al., 2023; Dai et al., 2023; Yu et al., 2024; Guo and Chen, 2024). In addressing the challenge of data scarcity for intent detection in novel domains, some of the prior work has applied metric-based meta-learning algorithms (Zhang et al., 2022; Sauer et al., 2022; Wang et al., 2023). Data augmentation is an alternative. Lee et al. (2021) employ extrapolation techniques with a sequence-to-sequence model to generate new, under-represented utterances. Alternatively, leveraging the advancements in LLMs, Sahu et al. (2022) use GPT-3 with in-context learning to get additional utterances. Lin et al. (2023) adopted a similar appraoch, using LLMs to generate synthetic examples, and then selecting useful data for training based on the Pointwise V-Information metric. Our approach differs from previous research in two key ways. Firstly, we specifically consider the zero-shot (instead of the few-shot) setting. This increases the challenge for LLMs to provide high-quality data. Secondly, rather than relying on a formula-based metric, we opt for a potentially more flexible approach – the Refiner. The Refiner directly generates refined data instead of filtering out irrelevant samples.

3 Method

3.1 Task Formulation

We assume the labeled dataset $D = \{(u_i, y_i)|y_i \in Y_s\}_{i=1}^{N}$, where u_i denotes i_{th} utterance labelled with intent y_i, Y_s denotes a set of distinct intents for seen domains, and N is the number of labelled utterances. Assuming that D provides sufficient amount of training data for seen domains (i.e., N is large), our objective is to predict the unseen intent class set Y_t, where $Y_s \cap Y_t = \emptyset$ and t denotes unseen domains. We consider the zero-shot setting, assuming that we have no example utterances in the unseen domain.

3.2 Utterance Generation with LLMs

We use generative LLMs to generate utterances for unseen intents. We prompt the model directly with a given intent in a zero-shot setting, requesting it to provide corresponding utterances (see Figure 1). The zero-shot use of an LLM for this task leads to suboptimal utterances that cannot directly be used as a replacement for human-generated data to train the intent classifier (Ye et al., 2022). To improve the quality of the generated utterances, we use training data from seen domains with sufficient labelled utterances. We train an utterance Refiner: a sequence-to-sequence model that takes generated utterances as input and outputs their refined versions. After training the Refiner on seen domains, it is applied to LLM-generated utterances from unseen domains to improve them.

3.3 Refiner

Refiner goal. The objective of the Refiner is to generate diverse, high-quality utterances from lower-quality inputs. We anticipate that the Refiner can effectively fulfill two roles: (a) In the case of irrelevant or inaccurate utterances generated by the

LLM, the Refiner should be able to generate utterances accurately aligned with the intended intents. (b) The generation of a diverse set of utterances holds the promise of significantly improving the overall performance of the intent classifier.

Training objective. Let $\mathcal{U}_j^{gen}$ denote the set of generated utterances and $\mathcal{U}_j^{real}$ denote the set of human-written utterances for the j_{th} unique intent from the seen domain. $\mathcal{U}_j^{gen}$ is generated using an LLM, while $\mathcal{U}_j^{real}$ is derived from the provided labeled dataset. We ensure that the length of the generated set matches that of our labeled dataset, i.e., $|\mathcal{U}_j^{gen}| = |\mathcal{U}_j^{real}| = N_j$. For each j_{th} distinct intent from the seen domain, a sub-training set $\hat{D}_j$ is defined as $\hat{D}_j = \{(U_{ij}^{gen}, U_{ij}^{real})\}_{i=1}^{N_j}$, where $U_{ij}^{gen} \subset \mathcal{U}_j^{gen}$ and $U_{ij}^{real} \subset \mathcal{U}_j^{real}$; and $\sum_{j=1}^{k} N_j = N$, where k is the total number of distinct intents. The complete training set for the Refiner is then represented as $\hat{D} = \bigcup_{j=1}^{k} \hat{D}_j$.

To ensure our Refiner generalizes well, we select some domains from seen domains as validation set, using the remaining seen domains for training. We use the regular loss function of the sequence-to-sequence model for training. This loss evaluates to what extent the refined utterance (the model's output) is different from the real utterance used as ground truth. As validation loss, we use a distinct metric based on classification loss. The classifier is fine-tuned with the labeled dataset D and is used to monitor the Refiner's performance during training with the validation domains.

4 Experimental Settings

Datasets. We use the CLINC150 (Larson et al., 2019) and SGD (Rastogi et al., 2020) datasets for our experiments. **CLINC150** includes 10 domains, with a total of 150 intents (15 per domain). Each intent has 100 training examples and 30 test examples. **SGD** includes 20 domains, with a total of 46 intents. Intents in SGD have a diverse number of utterances, ranging from hundreds to thousands. The domains included in both datasets are detailed in Appendix A.

LLMs for utterances generation. We experiment with two LLMs to generate utterances: **Zephyr-7B Beta** (Tunstall et al., 2023) and **Llama3-8B-Instruct** (AI@Meta, 2024). An example of the prompt we use for zero-shot utterance generation is shown in Figure 1 (top left).

Training the Refiner. Flan-T5-large (Chung et al., 2024) with 783M parameters serves as the backbone model for the Refiner. Pairs of seen-domain utterances and LLM-generated utterances are used to train the Refiner, which is trained for 6 epochs with a batch size of 24 and early stopping to prevent overfitting. During training, validation loss is monitored using a fine-tuned DistilBERT (Sanh et al., 2019)[2] classifier (67M parameters), trained on data from all seen domains. The classifier is fine-tuned over 1,800 steps with a batch size of 60, also using early stopping to avoid overfitting. The prompt we use for the Refiner is:

> The sentences above are user intent expressions for "{*intent*}" in the "{*domain*}" context, but they might have less quality or contain mistakes. Provide one improved expression.

Here, "{*intent*}" and "{*domain*}" are placeholders derived from the datasets.

Data sampling for Refiner training. To allow the Refiner to observe the diversity of the sample as a whole instead of a single input, we experiment with a multiple-to-one setting, aligning with previous research (Lee et al., 2021). Based on experimentation with different values (See the ablation study in the Section 6), we set the input number to 7 during both training and inference. This includes the current utterance plus 6 randomly selected utterances, with sampling done with replacement. The results of Refiner are obtained with the 7to1 setting (unless otherwise specified).

Evaluation. The main criterion for the generated utterances is *utility*: to what extent can the same task be performed with the generated utterances as with real (human) utterences. We therefore report on the accuracy of intent classification evaluation to assess the efficacy of the refined synthetic utterances. For this, we fine-tune DistilBERT[2] on the generated utterances with intent labels. The evaluation is conducted on authentic test data (real user utterances). To make the experiment results more generalized, for every experiment, we randomly split the domains into "seen" and "unseen", and perform 5 experiments with different splits. The evaluation metrics are based on the average of these five runs. For training the intent classifier, for each unseen-domain intent, we use 100 generated

[2] https://huggingface.co/distilbert/distilbert-base-uncased

Experiment ID	# seen intents	# unseen intents
1	28 (5.3K)	18 (22.7K)
2	26 (5.1K)	20 (21,6K)
3	29 (5.7K)	17 (16,8K)
4	25 (4.8K)	21 (26,7K)
5	29 (5.6K)	17 (18,3K)

Table 1: SGD train and test split: the number in parentheses indicates the total training and testing examples.

utterances for **CLINC150** and 200 for **SGD**. All reported accuracy results share the same classifier setting: a batch size of 60 and 1,800 training steps. The utterances were split into train and validation sets with an 80%-20% ratio. Early stopping was applied to prevent overfitting.

For intrinsic evaluation of the generated data, we report on diversity (distinct-1 & 2) (Li et al., 2016; Nakamura et al., 2019) and account for penalization of longer text in computing distinct-n, following (Joko et al., 2024); see Appendix B for details about implementing distinct-n.

Train-test split For the **CLINC150** dataset, we randomly select 5 out of the 10 domains as unseen domains for each experiment. We train on 4 seen domains and monitor validation loss on 1 seen domain, using a total of 7,500 examples ($= 100 * 15 * 5$), and evaluate on the remaining 5 unseen domains (75 intents), with a total of 2,250 test examples ($= 30 * 15 * 5$). For the **SGD** dataset, to generalize the results and align with our task formulation (details in Appendix C), we merge all splits and randomly select 8 out of 20 domains as unseen domains. We train on 9 seen domains and monitor validation loss on 3 seen domains. Table 1 provides detailed splits for each experiment.

Baselines. We compare our method to Ye et al. (2022)'s **ZeroGen**, utilizing data directly generated from the LLM for downstream tasks. We also compare with Meng et al. (2022)'s **SuperGen** data selection approach, which performs data augmentation in the zero-shot setting by extensively sampling from an LLM and selecting a subset based on output confidence (see for more details Appendix D). In both baselines we use the same generative LLMs as in our own method (Zephyr and Llama-3). Because **GPT-3.5-turbo** (OpenAI, 2023), referred to as ChatGPT, is widely used and shows strong zero-shot performance for many tasks, we use it as an additional baseline in the ZeroGen setting.[3] For all baselines and our approach, we use the same prompt to generated synthetic utterances.

[3]We did not try SuperGen's data selection approach for ChatGPT because it requires oversampling and would incur ten times the cost of the API.

	SGD	Clinc150
ZeroGen (Zephyr)	61.3 (4.9)	69.9 (2.0)
ZeroGen (Llama3)	68.7 (4.0)	73.8 (1.9)
ZeroGen (ChatGPT)	60.2 (10.0)	71.3 (1.5)
SuperGen (Zephyr)	58.9 (5.0)	65.8 (2.5)
SuperGen (Llama3)	67.4 (2.4)	71.4 (2.2)
Refiner (Zephyr)	72.2 (5.3)	76.0 (0.8)
Refiner (Llama3)	**72.4** (5.2)	**76.9** (1.7)

Table 2: Intent prediction accuracy is compared across unseen domains (averaged over 5 trials), with standard deviation reported in parentheses.

	SGD		Clinc150	
	Dist-1	Dist-2	Dist-1	Dist-2
ZeroGen (Llama3)	0.068	0.163	0.139	0.309
ZeroGen (ChatGPT)	0.080	0.213	0.147	0.348
SuperGen (Llama3)	0.058	0.131	0.120	0.259
Refiner (Llama3)	**0.129**	**0.363**	**0.200**	**0.518**
Real (human) Data	0.125	0.378	0.167	0.436

Table 3: Lexical diversity results, averaged over 5 trials.

Hardware/resources used. One NVIDIA A100 GPU, equipped with 40GB of memory, was used in all experiments. Additionally, it cost approximately 4 USD for using the ChatGPT API.

5 Results

Effect of Refiner on unseen domains. Table 2 shows that our approach outperforms all the baselines. The table indicates that our method achieves better performance without the need for extensive sampling or using very large language models. The classifier trained with data generated from Llama3 performs better than that from Zephyr. After refinement, the refined data gives consistently better results, reducing the difference between Llama3 and Zephyr. As a comparison, a classifier trained on real (human) data is still substantially better, with an accuracy of 95.4% in CLINC150. For SGD, we cannot make this comparison as all human data from SGD's unseen domains is used as test data, we do not have real human data for SGD to train a model for comparison.

Lexical diversity and similarity. To validate our assumption that the refiner provides more diverse data, we report distinct-n as diversity metric. Table 3 shows that the refiner leads to substantially more diverse data than other approaches. Additionally, the diversity of **SuperGen** is lower than of ZeroGen.

Qualitative analysis We manually compared the differences between the real data, the LLM-generated data, and the refined data. We found that refined data generally looks more like real data in

terms of length and format. LLMs tend to make longer sentences because they often include the purpose or reason behind their intent. For example, for the intent of freezing an account, the real dataset includes the sentence "i want my chase account blocked immeditately", while an LLM output example is "Can you put a hold on my account? I'm going on a trip and I don't want anyone to access my funds.".

Additionally, LLMs sometimes provide unnecessary explanations for their generated utterance. For example, LLMs generate the utterance "Can you please provide me with the phone number to text customer support? (User is asking for the specific contact information to send a text message to a customer support team)". The refiner can remove these unwanted parts. However, the refiner can still occasionally generate incorrect utterances. For example, "how do i report online" is not clear enough to indicate the intent of reporting fraud. More details and examples are provided in Appendix E.

6 Ablation Study

Effect of fine-tuning on performance. The results in Table 2 and 3 have shown that using refined language is better than directly using utterances generated by the LLMs. We investigate to what extent this improvement is due to the knowledge acquired from fine-tuning on seen domains. Table 4 shows that the fine-tuned Refiner significantly improves overall accuracy over a non-fine-tuned Refiner, especially with a larger sample size. This confirms that the Refiner learns to improve utterances cross-domain.

Factors to improve the training efficiency and performance. We additionally conduct an ablation study to explore various factors that could potentially affect training efficiency and performance: 1) We apply LoRA fine-tuning (Hu et al., 2021) with only around 5M trainable parameters to see if efficient fine-tuning can achieve similar outcomes. The results are in Table 5. We observe a slight drop in accuracy. 2) Our main results were based on using an input of 7 utterances. We experimented with different input/output settings to see if it affects performance. The results are in Table 6). We find that increasing the number of inputs slightly improves performance, while varying the number of outputs has minimal impact.

Data source	SGD		Clinc150	
	Zephyr	Llama3	Zephyr	Llama3
1×				
Refiner (NFT)	65.0	67.5	73.3	73.8
Refiner	**72.2****	**72.4****	**76.0****	**76.9****
2×				
Refiner (NFT)	62.9	67.8	74.2	76.3
Refiner	**72.4****	**72.5****	**77.2****	**78.1****

Table 4: Intent prediction accuracy for unseen domains: with and without fine-tuning the Refiner (averaged over 5 trials). $**$ indicates the fine-tuned model significantly outperforms the non-fine-tuned (NFT) model at $\alpha = 0.05$ based on a one-tail paired t-test. 1× denotes the default evaluation sample size, and 2× denotes double.

Data source	SGD		Clinc150	
	Zephyr	Llama3	Zephyr	Llama3
Refiner	**72.2**	**72.4**	**76.0**	**76.9**
Refiner (LoRA)	70.5	71.5	75.5	76.0

Table 5: Intent prediction accuracy for unseen domains (averaged over 5 trials): Full fine-tuning vs. LoRA parameter-efficient fine-tuning.

Refiner(LoRA):	SGD		Clinc150	
	Zephyr	Llama3	Zephyr	Llama3
1to1	69.7	70.5	74.0	75.1
3to1	71.5	70.6	74.7	75.5
5to1	71.0	72.1	74.1	**76.8**
7to1	70.5	71.5	**75.5**	76.0
7to3	71.6	**73.4**	74.9	76.7
7to5	**71.8**	71.8	74.1	76.0

Table 6: Intent prediction accuracy for unseen domains (averaged over 5 trials) with varied input and output setting.

7 Conclusions

We found that (1) our proposed utterance Refiner can improve data quality on zero-resource domains without the need for larger LLMs or oversampling; (2) data from seen domains is still useful for generalizing to unseen domains; and (3) the Refiner enhances the lexical diversity of the LLM-generated data, comparable to human data. Our results indicate that a two-step approach of a generative LLM in zero-shot setting and a smaller sequence-to-sequence model can lead to high-quality data for intent detection. In future work, further work can be done to close the gap to the quality of human-labelled training data, which other models and other training strategies.

Acknowledgments

This publication is part of the project LESSEN with project number NWA.1389.20.183 of the research program NWA ORC 2020/21 which is (partly) financed by the Dutch Research Council (NWO).

Limitations

Language. Like most of the prior work, we only focused on English utterance generation. This is relevant because current LLMs are known to be more proficient in English than in other languages, and our results may not generalize to lower-resource languages.

Computational resources. Not all settings that we originally wanted to experiment with were feasible with the computational resources that we had at our disposal (such as larger numbers of n in the n-to-n settings).

References

AI@Meta. 2024. Llama 3 model card.

Hyunsoo Cho, Hyuhng Joon Kim, Junyeob Kim, Sang-Woo Lee, Sang-goo Lee, Kang Min Yoo, and Taeuk Kim. 2023. Prompt-augmented linear probing: Scaling beyond the limit of few-shot in-context learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12709–12718.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Zihao Wu, Lin Zhao, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, et al. 2023. Chataug: Leveraging chatgpt for text data augmentation. *arXiv preprint arXiv:2302.13007*.

Yihao Fang, Xianzhi Li, Stephen Thomas, and Xiaodan Zhu. 2023. Chatgpt as data augmentation for compositional generalization: A case study in open intent detection. In *Proceedings of the Fifth Workshop on Financial Technology and Natural Language Processing and the Second Multimodal AI For Financial Forecasting*, pages 13–33.

Xu Guo and Yiqiang Chen. 2024. Generative ai for synthetic data generation: Methods, challenges and the future. *arXiv preprint arXiv:2403.04190*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

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. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 796–806.

Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. 2019. An evaluation dataset for intent classification and out-of-scope prediction. 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 1311–1316.

Kenton Lee, Kelvin Guu, Luheng He, Tim Dozat, and Hyung Won Chung. 2021. Neural data augmentation via example extrapolation. *arXiv preprint arXiv:2102.01335*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.

Yen-Ting Lin, Alexandros Papangelis, Seokhwan Kim, Sungjin Lee, Devamanyu Hazarika, Mahdi Namazifar, Di Jin, Yang Liu, and Dilek Hakkani-Tur. 2023. Selective in-context data augmentation for intent detection using pointwise v-information. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1455–1468.

Louis Marceau, Raouf Belbahar, Marc Queudot, Nada Naji, Eric Charton, and Marie-Jean Meurs. 2022. Quick starting dialog systems with paraphrase generation. *arXiv preprint arXiv:2204.02546*.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. *Advances in Neural Information Processing Systems*, 35:462–477.

Ryo Nakamura, Katsuhito Sudoh, Koichiro Yoshino, and Satoshi Nakamura. 2019. Another diversity-promoting objective function for neural dialogue generation.

Jinjie Ni, Tom Young, Vlad Pandelea, Fuzhao Xue, and Erik Cambria. 2023. Recent advances in deep learning based dialogue systems: A systematic survey. *Artificial intelligence review*, 56(4):3055–3155.

OpenAI. 2023. Chatgpt: A large language model. https://www.openai.com/chatgpt.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Gaurav Sahu, Pau Rodriguez, Issam H Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. 2022. Data augmentation for intent classification with off-the-shelf large language models. *ACL 2022*, page 47.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Anna Sauer, Shima Asaadi, and Fabian Küch. 2022. Knowledge distillation meets few-shot learning: An approach for few-shot intent classification within and across domains. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 108–119, Dublin, Ireland. Association for Computational Linguistics.

Heydar Soudani, Roxana Petcu, Evangelos Kanoulas, and Faegheh Hasibi. 2024. A survey on recent advances in conversational data generation. *arXiv preprint arXiv:2405.13003*.

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.

Xuyang Wang, Yajun Du, Danroujing Chen, Xianyong Li, Xiaoliang Chen, Yongquan Fan, Chunzhi Xie, Yanli Li, Jia Liu, and Hui Li. 2023. Dual adversarial network with meta-learning for domain-generalized few-shot text classification. *Applied Soft Computing*, 146:110697.

Congying Xia, Chenwei Zhang, Hoang Nguyen, Jiawei Zhang, and Philip Yu. 2020. Cg-bert: Conditional text generation with bert for generalized few-shot intent detection. *arXiv preprint arXiv:2004.01881*.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. Zerogen: Efficient zero-shot learning via dataset generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2024. Large language model as attributed training data generator: A tale of diversity and bias. *Advances in Neural Information Processing Systems*, 36.

Jianguo Zhang, Kazuma Hashimoto, Wenhao Liu, Chien-Sheng Wu, Yao Wan, S Yu Philip, Richard Socher, and Caiming Xiong. 2020. Discriminative nearest neighbor few-shot intent detection by transferring natural language inference. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5064–5082.

Jianhai Zhang, Mieradilijiang Maimaiti, Gao Xing, Yuanhang Zheng, and Ji Zhang. 2022. Mgimn: Multi-grained interactive matching network for few-shot text classification. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1937–1946.

A Domains for both datasets

The domains for each dataset are:

- CLINC150: Banking, credit cards, kitchen/dining, home, auto/commute, travel, utility, work, small talk, and meta.
- SGD: Restaurants, media, events, music, movies, flights, ride sharing, rental cars, buses, hotels, services, homes, banks, calendar, weather, travel, alarm, payment, trains, and messaging.

B Distinct-n computation

We consider all utterances belonging to the same intent as a single text. For distinct-n, we calculate the ratio of unique n-grams to the total number of words in each document and then determine the average. Because distinct-n penalizes longer texts more, we follow (Joko et al., 2024) and truncate the texts to the same length through sampling to ensure a fair comparison across generated datasets for each intent.

C Details of train-test split for the SGD dataset

For the SGD dataset, there are 40 intents in the train-validation split and 34 in the test split, with 46 unique intents across both splits, of which 28 are shared. If we use the default splits, the unseen intents can only be selected from the 34 in the test split instead of all 46. To make our results more generalizable, we merge all splits together, allowing all intents to serve as possible evaluation. We randomly select 8 out of the 20 domains as unseen domains. We train refiners on 9 "seen domains and monitor the validation loss on 3 "seen" domains.

We did not choose a 10/10 split because the SGD dataset has substantial fewer intents compared to the CLINC150 dataset (46 vs. 150), and our problem setting assumes we have sufficient seen-domain data.

Additionally, in SGD, the number of intents is not equally distributed across all domains. There are 4 domains with only 1 intent each, and 3 domains with 4 intents each. If we used a 10/10 domain split, it increase the chance of too few intents for the training split, undermining our assumptions. The number of examples for each intent ranges from 128 to 3,291, with an average of 1,244. Since each real human example has at least one unique generated pair from LLMs, and larger sample sizes would require more resources, which does not align with our low computing resource setting, we limit the training data to a maximum of 200 examples per seen-domain intent because some intents have fewer than 200 human examples, and we evaluate the refiners on unseen-domain intents.

D Details of the SuperGen baseline

Applying SuperGen to our tasks. Meng et al. (2022) use an LLM to generate $10N$ examples for each unseen class. N is the number of training examples. Subsequently, they calculated the geometric mean of the conditional generation probability for each example. The probability is conditioned on previously generated token(s) and the prompt. They selected N examples with the highest score for training. Following their approach, we first generate $1,000 = 100 \times 10$ examples per intent for the Clinc150 dataset and $2,000 = 200 \times 10$ examples per intent for the SGD dataset. Then, we select the top 10% confidence samples for evaluation. It's noteworthy that Meng et al. (2022) also apply a regularization approach to train a classifier, but this aspect is not considered in our study.

E Some generated examples

Examples of generated data We manually checked the generated data and there are some findings:

First, we found in the Clinc150, data generated by LLMs often produced longer utterances compared to real data. However, these utterances were shortened after refinement. Below are examples of the "freeze account" intent from the Clinc150 dataset[4]:"

Real data

- "please freeze my account"
- "i want my chase account blocked immediately"
- "i need you to block my chase account immediately"

Zephyr:

- "User: I would like to freeze my bank account temporarily."
- "User: Can you please put a hold on my bank account for a specific period of time?"
- "User: I need to prevent any transactions on my account for a while."

[4]Some utterances are shown in their original format with lowercase at the beginning

Llama3:

- "I want to freeze my account to prevent any unauthorized transactions."
- "Can you put a hold on my account? I'm going on a trip and I don't want anyone to access my funds."
- "I need to freeze my account temporarily until I can get a new debit card."

Zephyr+Refiner

- "can you freeze my bank account until future notice"
- "can you freeze my account"
- "is there a way to freeze all my accounts"

Llama3+Refiner

- "i need you to stop opening my account"
- "put my account on hold for now"
- ""help, i am wanting my account to be frozen""

Llama3+Refiner(LoRA)

- "freeze my account from now on"
- "help, i am trying to freeze my card"
- "can you do frozen transactions please"

Secondly, we observed that in the refined data, new words were used to express the intent. Below are examples of the refined output from Llama3:

- "get your approval to immediately halt all activity on my account (intent: freeze account): The word 'halt' is not used in the Llama3 output"
- "can you tell me the steps for handing in the money to my mom's account (intent: transfer): The phrase 'hand in' is not used in the Llama3 output"

Thirdly, we observed that the Refiner filtered out some noise words. For instance, in Zephyr's outputs, there were instances where "User:" was frequently used as a prefix, and occasional extra explanations were provided (see examples below). These occurrences were reduced after refinement.

- "Can you please provide me with the phone number to text customer support? (User is asking for the specific contact information to send a text message to a customer support team)" (intent: text)
- "User: I want to set a timer for 60 minutes, but I don't want it to beep when it's done. Is that possible? (Note: This utterance also indicates a preference or request for customization.)" (intent: timer)

Fourthly, we found that the Refiner with LoRA fine-tuning generates incorrect utterances more frequently than the fully fine-tuned Refiner. Below are examples selected from both datasets, with the correct intent shown in parentheses:

- "help with my credit cards" (intent: freeze account)
- "how do i report online" (intent: report fraud)
- "please help" (intent: account blocked)
- "I want to get a cab to take me somewhere from the city of San Francisco." (intent: find bus)