Efficient Out-of-Scope Detection in Dialogue Systems via Uncertainty-Driven LLM Routing

Álvaro Zaera[*], Diana Nicoleta Popa, Ivan Sekulić, Paolo Rosso
Telepathy Labs GmbH, Zürich, Switzerland
`{firstname}.{lastname}@telepathy.ai`

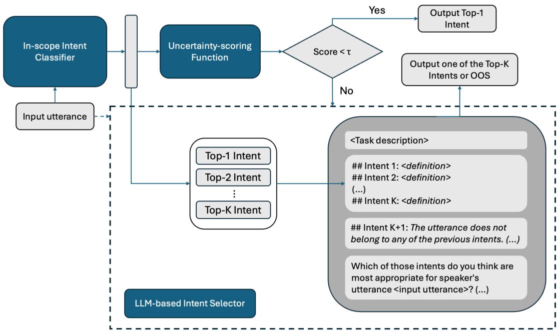
Abstract

Out-of-scope (OOS) intent detection is a critical challenge in task-oriented dialogue systems (TODS), as it ensures robustness to unseen and ambiguous queries. In this work, we propose a novel but simple modular framework that combines uncertainty modeling with fine-tuned large language models (LLMs) for efficient and accurate OOS detection. The first step applies uncertainty estimation to the output of an in-scope intent detection classifier, which is currently deployed in a real-world TODS handling tens of thousands of user interactions daily. The second step then leverages an emerging LLM-based approach, where a fine-tuned LLM is triggered to make a final decision on instances with high uncertainty. Unlike prior approaches, our method effectively balances computational efficiency and performance, combining traditional approaches with LLMs and yielding state-of-the-art results on key OOS detection benchmarks, including real-world OOS data acquired from a deployed TODS.

1 Introduction

Intent detection is a fundamental task in natural language understanding, enabling systems to accurately interpret and respond to user queries by identifying their underlying intention (Casanueva et al., 2020). While intent detection ensures that in-scope (INS) queries are mapped to predefined intents, detecting out-of-scope (OOS) intents is equally critical, especially in real-world applications, where users often interact in unpredictable ways, by, e.g., posing queries that fall outside the system's designed capabilities (Larson et al., 2019; Wang et al., 2024).

Without effective OOS detection, such inputs could lead to incorrect responses, reduced user trust, and eventual system failures as the universe of OOS queries for any TOD system is infinitely large (Arora et al., 2024). By identifying OOS queries, systems can gracefully handle such cases, by generating a predefined or dynamic response indicating its inability to process the request, by activating a fallback mechanism such as escalating the conversation to a human agent or by triggering updates to expand system coverage.



In-scope Intent Classifier
Uncertainty-scoring Function
Score < τ
Yes
No
Output Top-1 Intent
Output one of the Top-K Intents or OOS
Input utterance
Top-1 Intent
Top-2 Intent
⋮
Top-K Intent
<Task description>
## Intent 1: <definition>
## Intent 2: <definition>
(...)
## Intent K: <definition>
## Intent K+1: The utterance does not belong to any of the previous intents. (...)
Which of those intents do you think are most appropriate for speaker's utterance <input utterance>? (...)
LLM-based Intent Selector

Figure 1: Overview of **UDRIL**. An **uncertainty-scoring function** is applied to the output of an **in-scope classifier**. When a user utterance is potentially out-of-scope, ambiguous or misclassified, as indicated by the uncertainty score and a defined threshold, a **fine-tuned LLM** is prompted to correct the prediction; otherwise, the classifier's original prediction is maintained.

To address these challenges, we propose **Uncertainty-DRIven Large language models triggering, (`UDRIL`)**, a two-step method that combines efficiency with accuracy for robust intent detection. `UDRIL` is depicted in Figure 1 and consists of an in-scope intent classifier, an uncertainty prediction scoring function, and an LLM-based module. Specifically, we use a BERT-based classifier to ensure both effectiveness and efficiency in a task-oriented dialogue system (TODS) that is currently deployed in production and handling tens of thousands of user interactions daily. To refine predictions, we first apply NNK-Means (Gulati et al., 2024) to identify high-uncertainty instances. For these cases, an emerging LLM-based approach is

[*] This work was conducted as part of the author's internship at Telepathy Labs.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track), pages 328–335
July 28-30, 2025 ©2025 Association for Computational Linguistics

employed, where a fine-tuned LLM makes the final decision. This hierarchical approach leverages the efficiency of the BERT model for the majority of cases, while utilizing the LLM's capabilities for more ambiguous or complex inputs, including OOS detection. Our results demonstrate significant improvements in OOS detection, both on internal real-world dataset and on publicly available data. Notably, these gains are achieved with additional gains in effectiveness for INS intent detection (+5%), highlighting the method's overall robustness and practicality.

Our main contributions are as follows:

- a simple modular framework for joint INS and OOS intent detection, combining strengths of traditional intent classification, uncertainty modeling and LLMs;
- a design that selectively escalates user input to a more resource-intensive LLM, balancing efficiency and performance;
- state-of-the-art results on publicly available datasets and on real-world industry data from a deployed system, demonstrating practical applicability and effectiveness.

2 Related Work

Intent detection is an important task both in TODS (Casanueva et al., 2020) and in, now emerging, agent-based systems, where we aim to identify the right knowledge sources, APIs, and tools to use (Arora et al., 2024).

Non-LLM-based OOS Intent Detection. Previous research explored various approaches to intent detection using transformer-based classifiers. A key area of focus has been OOS detection, with methods generally falling into two categories: post-hoc methods that detect OOS instances after obtaining model representations, and approaches that enhance model robustness by modifying the training process to better handle OOS data (Gulati et al., 2024). We focus on the first category, as these methods are modular, adaptable, and easier to maintain, allowing for easy updates to the architecture without requiring intensive retraining. Particularly relevant in practice is the work by Gulati et al. (2024), in which the soft-clustering technique NNK-Means (Shekkizhar and Ortega, 2021) is applied for OOS detection. This enhances performance while also offering superior computational and memory efficiency compared to previous approaches.

LLM-based Intent Detection. Recently, LLM-based intent detection received significant attention, with studies analyzing the effect in intent detection performance produced by the incorporation of high-quality natural language intent descriptions (Hong et al., 2024). Off-the-shelf LLMs have been shown to outperform non-LLM based methods in few-shot settings where the training set only consist of a small number of utterances per intent class (Parikh et al., 2023). Hong et al. (2024) and Zhang et al. (2024) elaborate on this finding, showing that LLMs fine-tuned on intent detection datasets improve off-the-shelf LLMs, incorporating the ability to detect intents for domains unseen in training. Fine-tuning has also proven to be beneficial in few-shot settings, allowing to obtain better results with smaller LLMs compared to off-the-shelf LLMs (Parikh et al., 2023) and in-context learning (ICL) approaches (Mirza et al., 2024)).

However, the performance improvement achieved by LLM-based intent detection, as compared to earlier non-LLM methods, is primarily reported in few-shot settings, where the training is strictly constrained by the number of intents per class. Previous studies reporting comparisons in full-data settings show that LLMs still underperform relative to BERT-based approaches in such cases (Parikh et al., 2023; Mirza et al., 2024). This underscores the continued relevance of BERT-based methods for practical deployment. Combining the strengths of both LLMs and BERT-based approaches could lead to more flexible systems, capable of adapting to a wider range of training data scenarios and enhancing deployment versatility.

In the context of out-of-scope (OOS) detection, LLMs have been shown to struggle with effective detection when relying solely on text representations without additional training (Arora et al., 2024; Wang et al., 2024). To address this limitation, Liu et al. (2024) explore the use of fine-tuning via low-rank adaptation (LoRA) (Hu et al., 2021) on INS data, demonstrating that this approach enhances the utility of last-token representations for OOS detection through cosine similarity.

Hybrid Approach. Through the current proposal, we aim to adopt a hybrid approach that combines non-LLM-based OOS intent detection methods with fine-tuned LLMs, leveraging the distinct strengths of the previously discussed methods. A relevant related work to ours is that of Arora et al. (2024) who also propose a two-step approach to intent classification, albeit involving two LLM passes to determine if an utterance is OOS. Additionally, their proposal requires maintaining a vector storage of last token representations for a set of training examples per intent, performing negative data augmentation and employing multiple runs of monte carlo dropout, making the whole process less scalable. Also, contrary to Arora et al. (2024) who argue that fine-tuning an LLM for this purpose is impractical and prohibitive from development and maintenance perspective, our experiments as well as related work (Hong et al., 2024) show that fine-tuning with a set of guidelines is helpful for inference even when the said guidelines are later updated. Therefore, from the maintenance perspective, an update of the intent space and guidelines does not require extra work.

3 Uncertainty-Driven LLM-based Framework for OOS Intent Detection

We propose `UDRIL`, a framework for intent classification and OOS detection, consisting of an in-scope intent classifier and an LLM intent refiner, guided by an uncertainty scoring function f. The system first employs a classifier to generate an in-scope prediction. If the prediction is deemed confident by f, it is used directly; otherwise, the LLM refines it based on the classifier's output. The proposed framework enhances the cost-efficient classifier by enabling OOS detection while selectively leveraging the LLM, a computationally resource-heavy method, ensuring an accuracy - efficiency balance.

We next describe each component of our framework, noting that they can be replaced based on available resources and performance requirements.

3.1 In-scope Intent Classifier

Specifically, given user utterance u, the initial classifier's task is to model the probability distribution over a set of N classes $\mathcal{Y}$, selecting the one with highest probability as an output: $\hat{y}_C = \arg\max_{y \in \mathcal{Y}} P_C(y \mid u; \theta_C)$ where $P_C(y \mid u; \theta_C)$ is the classifier's predicted probability distribution and θ_C its parameters.

In order to meet the demands of low-latency applications, we model P_C with DistilBERT (Sanh et al., 2019), due to its strong balance between efficiency and effectiveness, making it suitable for an industry setting. Moreover, the training process only models θ_C and does not incorporate any methods specific to OOS detection, as this responsibility is entirely managed by the uncertainty-scoring function f and the LLM. Instead, the focus is on training the model to perform general classification tasks efficiently. We use focal loss (Ross and Dollár, 2017) during training to address the intent class imbalance that is likely to occur in the training dataset of real dialogue systems.

3.2 Uncertainty-Scoring Function

A function f provides an uncertainty score based on the output of the in-scope classifier, which aims to determine whether the prediction is sufficiently reliable or if further processing by the LLM is required. Specifically, score $s_u = f(u)$ indicates the uncertainty score for utterance u. If s_u exceeds a predefined threshold τ, the utterance is routed to the LLM. Otherwise, the classifier's prediction is used directly.

We model f with EC-NNK-Means (Gulati et al., 2024), a soft-clustering based method trained on utterance embeddings to learn a dictionary that minimizes the reconstruction error of the training data. At inference, s_u is the NNK-Means reconstruction error. In Gulati et al. (2024), it is shown that new data with high reconstruction error is more likely to be OOS. We observe that this method also has satisfactory results in identifying potentially misclassified INS data, making it valuable for detecting utterances that require prediction refinement. In our experiments, we apply EC-NNK-Means to the last output embedding of the DistilBERT `[CLS]` token.

Threshold τ can be tuned to route higher, or lower, ratio of utterances to the LLM, balancing the effectiveness and efficiency as needed. In this work, we experiment with three specific thresholds to showcase its effect on the routing ratio and the overall performance. The selected thresholds define low-routing ($\tau = 0.15$), moderate-routing ($\tau = 0.10$) and high-routing ($\tau = 0.05$) strategies.

3.3 LLM-Based Intent and OOS Detection

If the classifier is uncertain, i.e., $s_u > \tau$, the utterance u is forwarded to the LLM to make a final decision. Formally, given the top-k intent candidates $(\hat{y}_{(1)}, \ldots, \hat{y}_{(k)})$, as modeled by P_C, the LLM either selects the most appropriate intent among the top-k or determines that u is out-of-scope (OOS):

$$\hat{y}_{LLM} = \arg\max_{y \in \{\hat{y}_{(1)}, \ldots, \hat{y}_{(k)}, OOS\}} P_{LLM}(y \mid u, \hat{y}_{(1)}, \ldots, \hat{y}_{(k)}; \theta_{LLM}) \quad (1)$$

In this work, we learn θ_{LLM} of P_{LLM} via fine-tuning using LoRA (Hu et al., 2021) with a language modeling objective. Our method is designed to provide the LLM with OOS detection capabilities using only INS data. For the dataset creation, given each <utterance-gold label> pair (u, y_u), we additionally create one negative example (u, OOS), using k candidates $(y'_{(1)}, \ldots, y'_{(k)})$ sampled from $\mathcal{Y} \setminus \{y_u\}$, as described in Algorithm 1. We then train using the obtained dataset D' to maximize Eq. (1).

Algorithm 1 Fine-tuning Dataset Creation

```
Input: INS Dataset $D$, Classifier $P_C$, Param $\theta_C$
Output: Fine-tuning Dataset $D'$

Initialize: $D' \leftarrow \emptyset$
for each $(u, y_u)$ in $D$ do
    Use $P_C(\,\cdot\,|u;\theta_C)$ to obtain $(\hat{y}_{(1)}, \ldots, \hat{y}_{(k)})$
    Add $(u, (\hat{y}_{(1)}, \ldots, \hat{y}_{(k)}), y_u)$ to $D'$
    Sample $k$ distinct intents from $\mathcal{Y} \setminus \{y_u\}$:
        $(y'_{(1)}, \ldots, y'_{(k)})$
    Add $(u, (y'_{(1)}, \ldots, y'_{(k)}), OOS)$ to $D'$
end for
Return: Fine-tuning Dataset $D'$
```

For our experiments, we use Llama 3.1-8B (Dubey et al., 2024) as the LLM with $k = 3$ intent descriptions. The prompt contains a description of each of the k intents. Each epoch, the order of the k candidates is shuffled in the prompt. The fine-tuning set is created using 5 random utterances from the training set per intent class. In cases where the number of available utterances was lower than 5, we performed data augmentation. Having a limited number of examples, combined with using a parameter-efficient fine-tuning technique (LoRA), facilitates deployment in production environments.

3.4 Evaluation Setup and Data

Internal benchmark. Our main goal is to tackle intent detection in our deployed TOD system; thus, we primarily evaluate our approach on an internal benchmark. To this end, we extract 6492 real user utterances from our past user-system interactions and manually annotate them with one of 42 intents. We refer to this dataset as *BookData*.

Public benchmark. To ensure comparability to related work, we further evaluate our methods on the real-world data from the HINT3 collection (Arora et al., 2020), created from live chatbot interactions in diverse domains. The collection contains three datasets: *SOFMattress* (mattress products retail), *Curekart* (fitness supplements retail), and *Powerplay11* (online gaming). Utterances in the train sets are labeled with between 21–57 INS intents, while the test sets additionally contain a large number of OOS utterances.

Intent guidelines. While for internal data, we have access to annotation guidelines, for public benchmarks such guidelines are not made available. To solve this, we generate guidelines for each of the public datasets using OpenAI's GPT3.5: for each intent, we provide as input the intent name and all utterances that are part of the train set for that intent. We then ask the LLM to generate a definition such that, when presented along with such examples, a human would choose to label the examples with the given intent. We make no further adjustments or post-processing to the obtained guidelines.

4 Results and Discussion

Table 1 presents results on HINT3 public datasets, comparing state-of-the-art solutions (Arora et al., 2024) and our methods. We compare to three main categories of related work results: (1) non-LLM (*SNA*) and the best performing LLM-based approaches in Arora et al. (2024): *Mistral-7B*, *Claude v3 Haiku* and *Mistral Large*; (2) hybrid models and (3) the proposal of Arora et al. (2024) specifically designed for OOS intent detection.

4.1 Open-Source Data

Average F1-scores across all datasets show that `UDRIL` provides an average of 2-3% relative improvement compared to state-of-the-art methods that employ significantly larger LLMs, up to 13% relative improvement compared to traditional classifier-based approaches and up to 34% relative improvement when compared to similar-sized LLMs (see comparison to *Mistral-7B* (Arora et al., 2024)). The increase in performance holds regardless of the routing strategy employed. It also holds when using an LLM that was not fine-tuned for the task compared to other similar-sized LLMs (`UDRIL-noFT` can yield up to 10% increase compared to *Mistral-7B* (Arora et al., 2024)), validating the value of our architecture beyond fine-tuning.

`UDRIL` also outperforms hybrid approaches by up to 5%, despite these latter ones using much larger LLMs. Methodology-wise, `UDRIL` is also simpler: there is no need for negative data augmentation

Method	Curekart	SOFMattress	PowerPlay11	Avg Score	BookData	Param
SNA (Arora et al., 2024)	0.709	0.672	0.639	0.673	-	NA
Mistral-7B (Arora et al., 2024)	0.615	0.699	0.384	0.566	-	7B
Claude v3 Haiku (Arora et al., 2024)	0.775	**0.815**	0.646	0.745	-	NA
Mistral Large (Arora et al., 2024)	0.779	0.767	0.668	0.738	-	123B
SNA + Claude v3 Haiku (Arora et al., 2024)	0.756	0.730	0.690	0.725	-	NA
SNA + Mistral Large (Arora et al., 2024)	0.761	0.719	0.692	0.724	-	NA
Mistral-7B-2steps (Arora et al., 2024)	0.766	0.751	**0.739**	0.752	-	7B
UDRIL-noFT (low-route)	0.637	0.661	0.525	0.607	0.831	8B
UDRIL-noFT (moderate-route)	0.660	0.672	0.542	0.624	0.826	8B
UDRIL-noFT (high-route)	0.662	0.676	0.547	0.628	0.790	8B
UDRIL-noFT (full-route)	0.655	0.669	0.545	0.623	0.748	8B
UDRIL-FT (low-route)	0.727	0.764	0.677	0.722	0.852	8B
UDRIL-FT (moderate-route)	0.779	0.777	0.701	0.752	**0.857**	8B
UDRIL-FT (high-route)	**0.791**	0.784	0.710	**0.761**	0.853	8B
UDRIL-FT (full-route)	0.787	0.777	0.708	0.757	0.850	8B

Table 1: F1 scores across state-of-the-art methods and our proposed solution UDRIL, with different routing strategies. The postfix *-noFT* refers to off-the-shelf models that were not fine-tuned, while *-FT* refers to the fine-tuned version of Llama 3.1-8B. *Mistral-7B* is the model proposed in Arora et al. (2024), with comparable number of parameters to our method, while *Claude v3 Haiku* and *Mistral Large* are the best performing models of Arora et al. (2024) - albeit much bigger than our proposed solutions. *SNA + Mistral Large*; and *SNA + Claude v3 Haiku* are hybrid models and *Mistral-7B-2steps* is the best OOS model in (Arora et al., 2024). Best scores are in **bold**, second best are underlined.

for the classifier or multiple uncertainty estimation runs, unlike other hybrid proposals.

Finally, UDRIL yields improvements over the OOS-specific method of Arora et al. (2024) for Curekart and SOFMattress and incurs only slight degradation in the case of PowerPlay11, making it on average the better performing model of the two. Beyond performance, the simplicity of UDRIL also makes it easier to use in practice.

4.2 Real-World Data

We observe a performance increase on *BookData* when fine-tuning is employed and a progressive decrease as we route more utterances with the non-fine-tuned models. These results suggest that, for a real-world industry setting, fine-tuning LLM-based models on in-domain labeled data is still superior to switching to in-context learning with LLMs.

Furthermore, increasing the amount of training data, even with noisy labels, improves the performance of a DistilBERT-based classifier, thereby reducing the need for extensive routing to achieve optimal results. Additionally, fine-tuning the LLM on a small set of utterances enhances the framework's robustness across various routing strategies, enabling effective out-of-scope (OOS) handling without compromising in-scope (INS) performance.

4.3 Impact of Fine-Tuning on Performance

Fine-tuning improves the OOS detection capabilities of UDRIL by substantially increasing recall, with only a minor reduction in precision. For instance, in BookData with the full-route setting, the OOS recall increases from 0.403 to 0.698 and the precision is very similar, dropping from 0.514 to 0.508. The reduction in OOS precision could potentially lead to a slight decrease in INS performance. This is not the case for BookData, where the INS Accuracy increases from 0.768 with the off-the-shelf LLM to 0.856 with the fine-tuned version in the full-route setting. However, in the HINT3 dataset we do observe slight drops: Curekart 0.817 to 0.815, Sofmattress 0.806 to 0.743 and Powerplay11 0.599 to 0.547. We observe that incorporating OOS detection capabilities through fine-tuning is more likely to negatively impact INS performance for cases where the first-stage classifier performs worse (such as Powerplay11).

4.4 Balancing INS and OOS performance

Table 2 compares UDRIL with the method specifically designed for OOS detection in Arora et al. (2024). Our approach strikes a better balance between OOS recall and INS accuracy, leading to a superior overall F1 score on two out of three datasets. Powerplay11 is the only exception, where

		F1 Score	INS Accuracy	OOS Precision	OOS Recall
SOF Mattress	Mistral-7B-2steps (Arora et al., 2024)	0.751	**0.767**	-	0.715
	UDRIL-FT (high-route)	**0.784**	0.759	0.725	**0.840**
Curekart	Mistral-7B-2steps (Arora et al., 2024)	0.766	0.736	-	**0.782**
	UDRIL-FT (high-route)	**0.791**	**0.830**	0.888	0.744
Power Play11	Mistral-7B-2steps (Arora et al., 2024)	**0.739**	0.411	-	**0.950**
	UDRIL-FT (high-route)	0.710	**0.557**	0.857	0.748

Table 2: Best-performing Arora et al. (2024) method vs UDRIL, focusing on OOS and INS performance.

Arora et al. (2024) outperforms ours. However, this can be attributed to the fact that ∼68% of the utterances in the test split of Powerplay11 are OOS. Their method, which achieves a significantly high OOS recall at the cost of excessively low INS accuracy, has limited practical applicability compared to our more balanced approach. That said, our approach does not achieve ideal INS accuracy either - most likely due to the first-stage classifier: since Powerplay11's training set is of lower quality, this directly impacts both the DistilBERT classifier and the overall performance of the framework.

Intent guidelines Experiments showed that fine-tuning using guidelines of one dataset can be beneficial across datasets: results on SOFMattress and PowerPlay11 with UDRIL fine-tuned using Curekart-specific guidelines are comparable to those obtained when fine-tuning using their own guidelines directly. These findings are in line with recent work (Hong et al., 2024) and support the usability of the method in the lack of up-to-date dataset-specific guidelines at fine-tuning time.

Uncertainty measures and LLMs. We experimented with different LLMs, including recent *DeepSeek-R1-Distill-Llama-8B* and *DeepSeek-R1-Distill-Qwen-7B* models[1], as well as several uncertainty measures, such as Shannon Entropy and Energy, as proposed in (Sun et al., 2024). Results were similar to the reported ones with some degradation observed when using other uncertainty measures.

How good is our routing strategy? We observe routing strategies above *moderate* yield improvements over existing models, with the preferred approach consisting in *high* amount of routing.

The percentage of routed OOS utterances varies between 70-96% for Curekart, 84-98% for SOFMattress and 79-98% for PowerPlay11, depending on how conservative we are. Furthermore, of the incorrectly labeled INS utterances, our method routes between 40-88% in the case of Curekart, 53-87% for SOFMattress and 65-97% for Powerplay11, as seen from Figure 2. We also observe that when DistilBERT performs better, fewer correctly classified INS utterances are routed to the LLM, demonstrating that the routing method effectively captures prediction uncertainty. We conclude that our routing method benefits both OOS and INS labels.

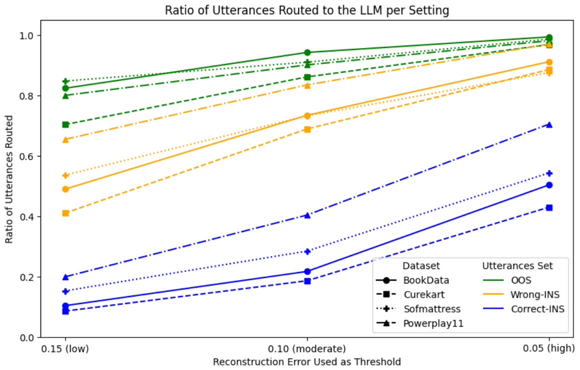


Figure 2: Impact of routing threshold to number of routed utterances across four datasets and three utterance label sets.

5 Conclusion

In this paper, we introduce UDRIL, a framework that achieves state-of-the-art performance in both in-scope (INS) intent classification and out-of-scope (OOS) intent detection. Unlike approaches that require modifying or retraining the base intent classifier, UDRIL operates by modeling its outputs, enabling OOS detection while preserving the efficiency of the existing classifier. This makes our framework particularly well-suited for real-world deployment, as shown by the results on our internal benchmark, derived from real user-system interactions, where maintaining low latency and computational efficiency is crucial.

Moreover, UDRIL is modular, allowing for the seamless substitution of different components: base classifier, uncertainty estimation method, and LLM. Furthermore, it provides a practical mecha-

[1] https://huggingface.co/deepseek-ai

nism for controlling efficiency-performance tradeoffs by adjusting the routing percentage threshold, ensuring adaptability to varying production constraints. By enabling reliable OOS detection without disrupting existing intent classification models, our approach offers a scalable solution for enhancing the robustness of deployed TOD systems.

Ethical Considerations

We prioritize user privacy and ensure that no real conversations are reported in this paper. Additionally, we do not release any data or model weights trained on user interactions. All data used in our study was collected with user consent, ensuring ethical use and compliance with the US privacy considerations.

References

Gaurav Arora, Chirag Jain, Manas Chaturvedi, and Krupal Modi. 2020. HINT3: Raising the bar for intent detection in the wild. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 100–105, Online. Association for Computational Linguistics.

Gaurav Arora, Shreya Jain, and Srujana Merugu. 2024. Intent detection in the age of LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, EMNLP'24, pages 1559–1570.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

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

Aryan Gulati, Xingjian Dong, Carlos Hurtado, Sarath Shekkizhar, Swabha Swayamdipta, and Antonio Ortega. 2024. Out-of-distribution detection through soft clustering with non-negative kernel regression. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12943–12959, Miami, Florida, USA. Association for Computational Linguistics.

Taesuk Hong, Youbin Ahn, Dongkyu Lee, Joongbo Shin, Seungpil Won, Janghoon Han, Stanley Jungkyu Choi, and Jungyun Seo. 2024. Exploring the use of natural language descriptions of intents for large language models in zero-shot intent classification. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 458–465, Kyoto, Japan. Association for Computational Linguistics.

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

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 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, Hong Kong, China. Association for Computational Linguistics.

Bo Liu, Li-Ming Zhan, Zexin Lu, Yujie Feng, Lei Xue, and Xiao-Ming Wu. 2024. How good are LLMs at out-of-distribution detection? In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, LREC-COLING'24, pages 8211–8222.

Paramita Mirza, Viju Sudhi, Soumya Ranjan Sahoo, and Sinchana Ramakanth Bhat. 2024. ILLUMINER: Instruction-tuned large language models as few-shot intent classifier and slot filler. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, LREC-COLING'24, pages 8639–8651.

Soham Parikh, Mitul Tiwari, Prashil Tumbade, and Quaizar Vohra. 2023. Exploring zero and few-shot techniques for intent classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, ACL'23, pages 744–751.

T-YLPG Ross and GKHP Dollár. 2017. Focal loss for dense object detection. In *proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2980–2988.

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

Sarath Shekkizhar and Antonio Ortega. 2021. Nnk-means: Dictionary learning using non-negative kernel regression. *CoRR*, abs/2110.08212.

Fanshu Sun, Heyan Huang, Puhai Yang, Hengda Xu, and Xianling Mao. 2024. Out-of-scope intent detection with intent-invariant data augmentation. *Knowledge-Based Systems*, 283:111167.

Pei Wang, Keqing He, Yejie Wang, Xiaoshuai Song, Yutao Mou, Jingang Wang, Yunsen Xian, Xunliang

Cai, and Weiran Xu. 2024. Beyond the known: Investigating LLMs performance on out-of-domain intent detection. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, LREC-COLING'24, pages 2354–2364.

Feng Zhang, Wei Chen, Fei Ding, Meng Gao, Tengjiao Wang, Jiahui Yao, and Jiabin Zheng. 2024. From discrimination to generation: Low-resource intent detection with language model instruction tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, ACL findings'24, pages 10167–10183.