Intent Detection in the Age of LLMs

Gaurav Arora
Amazon
gaurvar@amazon.com

Shreya Jain
IIT Jammu*
2020uee0135@iitjammu.ac.in

Srujana Merugu
Amazon
smerugu@amazon.com

Abstract

Intent detection is a critical component of task-oriented dialogue systems (TODS) which enables the identification of suitable actions to address user utterances at each dialog turn. Traditional approaches relied on computationally efficient supervised sentence transformer encoder models, which require substantial training data and struggle with out-of-scope (OOS) detection. The emergence of generative large language models (LLMs) with intrinsic world knowledge presents new opportunities to address these challenges. In this work, we adapt 7 SOTA LLMs using adaptive in-context learning and chain-of-thought prompting for intent detection, and compare their performance with contrastively fine-tuned sentence transformer (SetFit) models to highlight prediction quality and latency tradeoff. We propose a hybrid system using uncertainty based routing strategy to combine the two approaches that along with negative data augmentation results in achieving the best of both worlds (i.e. within 2% of native LLM accuracy with 50% less latency). To better understand LLM OOS detection capabilities, we perform controlled experiments revealing that this capability is significantly influenced by the scope of intent labels and the size of the label space. We also introduce a two-step approach utilizing internal LLM representations, demonstrating empirical gains in OOS detection accuracy and F1-score by >5% for the Mistral-7B model.

1 Introduction

Task oriented dialogue systems (TODS) have gained significant traction and investment from industry because of their efficiency, accessibility and 24x7 availability to serve customers. Automation through TODS is expected to save billions of dollars in labor costs by 2026 (Gartner, 2022).

Intent Detection is a vital part of natural language understanding (NLU) layer of TODS. Traditionally, intent detection has been used to understand and map the user query to a bot action (e.g., respond with a static answer, execute a pre-configured flow etc) (Dialogflow, 2010; LEX, 2017). With increasing use of LLMs such as ChatGPT (OpenAI, 2022), Claude (Anthropic, 2023), Mistral (Mistral, 2023), Llama (Meta, 2023) as retrieval augmented generators to generate answers to user queries in TODS, intent detection is being used to identify the right knowledge sources, APIs and tools to call for retrieval augmented generation. This ensures efficient utilization of tools, APIs and various other knowledge sources.

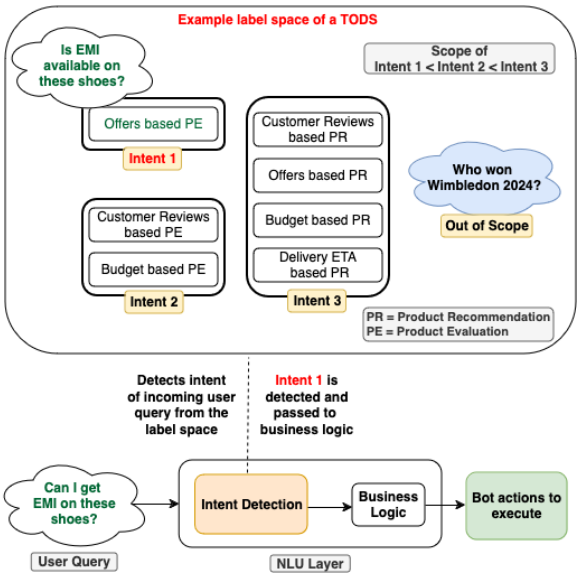


Figure 1: Example of broad/specific intent scopes and OOS queries which Intent Detection systems deal with in a typical TODS.

An intent detection system of a conversational AI service is expected to handle intents anywhere in the spectrum of very-broad to very-specific scopes[1] depending upon actionability of intents and bot usecases as shown in Fig 1. They are also

*Contributed to this work during her internship at Amazon

[1]By "scope of intent" we mean semantic space of all natural language utterances which can fall in that intent.

expected to accurately reject out-of-scope (OOS) queries[2] without having access to any training data for such queries as universe of OOS queries for any TODS is infinitely large. Since for a typical conversational AI service, data for intent detection training comes from bot developers who are not experts in ML, intent detection systems have to also deal with imbalanced training datasets. Additionally, these systems are expected to work with very few utterances per intent.

Traditionally, intent detection systems have been built using supervised classification or similarity based models (Zhang et al., 2021; Liu and Lane, 2016; Casanueva et al., 2020). LLMs, due to their few-shot learning capabilities, world knowledge and impressive performance across multiple NLP tasks (Qin et al., 2023; Zhao et al., 2023), have the potential to improve intent detection systems in TODS. In this work, we explore how LLMs can be best leveraged for the task of intent detection and assess their ability to handle OOS queries and varying scope of intents.

Contributions. 1. We employ generative LLMs using adaptive in-context learning (ICL) and chain of thought (CoT) prompting for the task of intent detection and compare them against contrastively fine-tuned sentence transformer (SetFit) models, highlighting performance/latency trade-offs. We evaluate 7 SOTA LLMs from Claude and Mistral families on 3 open-source and 3 internal real world datasets.

2. We propose a hybrid system that combines SetFit and LLM by conditionally routing queries to LLM based on SetFit's predictive uncertainty determined using Monte Carlo Dropout. We also propose a negative data augmentation technique that improves SetFit's performance by $>5\%$ across datasets. The resulting system achieves performance within $\sim 2\%$ of native LLM performance with $\sim 50\%$ less latency than native LLM.

3. We study the behavior of adaptive ICL based intent detection through controlled experiments and show that LLM's OOS detection capability significantly depends upon the scope of intent labels (class design) and the number of labels.

4. We also propose a novel two step methodology utilizing internal LLM representations to help improve LLM's OOS detection capabilities and show empirical gains in OOS detection accuracy and F1-score by >5% across datasets for Mistral-7B.

We intend to also share the three internal datasets after necessary approvals as a community resource and to ensure reproducibility.

[2]Out-of-scope (OOS) queries are the ones which do not fall into any of the system's supported intents (Larson et al., 2019).

2 Related Work

Evaluation of LLMs. LLMs like ChatGPT (OpenAI, 2022), GPT-4 (OpenAI et al., 2024), Claude (Anthropic, 2023), Mistral (Mistral, 2023), Llama (Meta, 2023) have shown impressive performance on multiple NLP tasks and benchmarks (Zhao et al., 2023). Supervised BERT (Devlin et al., 2018) based models have been widely used for intent detection but now with the advent of LLMs it is not clear what benefits they bring for intent detection in the real world. Hence in this work, we evaluate LLMs on the critical task of intent detection for TODS on real world intent detection datasets and highlight performance/latency tradeoffs by benchmarking LLMs with traditional sentence transformers. Recent work (Wang et al., 2024; Liu et al., 2024) majorly focused on evaluation of LLMs on datasets like CLINC150 (Larson et al., 2019), BANKING77 (Casanueva et al., 2020) which are: (i) not real world intent detection datasets (queries are not from deployed TODS), (ii) not multi-label (every query maps to single intent). Instead, our evaluation is on real world intent detection datasets wherein queries are from deployed TODS which have real world challenges like intents with very-broad to very-specific scopes, imbalanced training datasets with very few examples per intent and 3 out of 6 of our datasets are also multi-label which makes our evaluation more comprehensive.

Improving OOS detection performance of LLMs. Recent work (Liu et al., 2024) fine-tuned LLMs to improve OOS performance which is prohibitive both from development and maintenance perspective for a typical Conversational-AI platform which needs to support hundreds of different TODS (because fine-tuning and deploying a separate instance of LLM for every TODS is prohibitively expensive which makes fine-tuning LLMs impractical). Hence, we propose an alternative approach without LLM fine-tuning which improves both OOS accuracy and overall performance by >5% and allows use of the same instance of foundational LLM across TODS.

Hybrid intent detection system which uses LLMs. Unlike prior work, our focus is not just on evaluation of LLMs and/or improving OOS de-

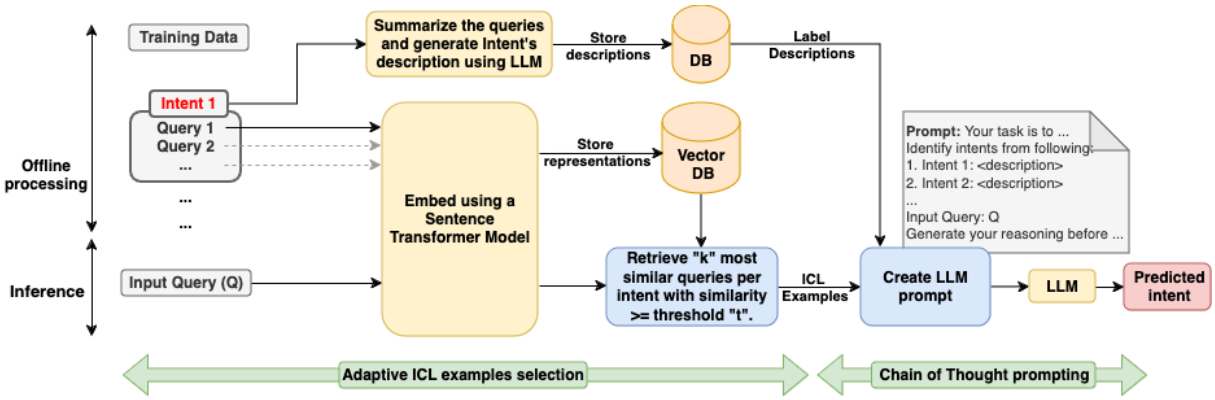


Figure 2: Methodology for adaptive ICL and CoT based intent detection using LLMs.

tection performance of LLMs, but we also focus on building a deployable intent detection system which can benefit from LLMs but does not have prohibitive cost and latency, as part of which we propose a hybrid system using uncertainty based routing strategy to combine LLMs and SetFit approaches that along with negative data augmentation results in achieving the best of both worlds (i.e. within 2% of native LLM accuracy with 50% less latency).

Better understanding of LLM's OOS detection capabilities. In this work we do controlled experiments to study the effect of scope of labels and size of label space. Recent work (Wang et al., 2024) also investigated the effect of the size of the label space on LLM's OOS performance and their findings are inline with our findings. However, our findings on how LLM OOS detection capabilities are influenced by the scope of intent labels are novel and would inform label space design during development of TODS.

3 Leveraging LLMs for Intent Detection

In this section we see how LLMs can be best leveraged for intent detection and propose a hybrid system which leverages LLMs conditionally, achieving a balance between performance and cost.

3.1 Methodology

3.1.1 Fine-Tuned Sentence Transformers

We fine tune sentence transformer (SetFit) models in two steps (Tunstall et al., 2022a) and use them as our baseline. In the first step, a sentence transformer model is fine-tuned on the training data in a contrastive, siamese manner on sentence pairs. In the second step, a text classification head is trained using the encoded training data generated by the fine-tuned sentence transformer from the first step.

Negative Data Augmentation. To help SetFit learn better decision boundaries, we augment training data by modifying keywords in sentences by (a) removing, or (b) replacing them with random strings. These modified sentences are considered OOS during training. Since these augmented OOS sentences have similar lexical pattern as in-scope training sentences, these are expected to help the model avoid latching onto any spurious patterns and help overall learning.

3.1.2 Adaptive ICL + CoT based Intent Detection using LLMs

Fig 2 shows how we use LLMs with adaptive ICL and CoT prompting for intent detection. During offline processing, we embed all training examples using a sentence transformer model and store the embedding vectors in a DB. Additionally, we generate and store descriptions for every intent from training data using LLM. During inference, we embed the user query using the same transformer model and retrieve top-k most similar queries per intent with similarity >t, where t is retriever threshold. We construct prompt for LLM using retrieved ICL examples, stored intent descriptions and static task specific instructions.

3.1.3 Uncertainty based Query Routing

High compute and latency costs of LLMs make them prohibitively expensive to use in production at scale.[3] Hence, we propose a hybrid system which routes incoming queries to LLMs for intent detection only if SetFit model is uncertain. We sample M predictions from the SetFit model using Monte Carlo (MC) dropout (Gal and Ghahramani, 2016)

[3]Mechanisms like caching can help somewhat but we skip their discussion for brevity.

	SOF Mattress	Curekart	Power Play11	ALC	ADP	OADP	Avg Score	Avg p50 Latency
Claude v1 Instant	0.613	0.528	0.295	0.840	0.687	0.630	0.599	2.297
Claude v2	0.763	0.773	0.665	0.891	0.703	0.630	**0.737**	11.795
Claude v3 Haiku	**0.815**	0.775	0.646	0.849	0.715	0.619	0.736	1.697
Claude v3 Sonnet	0.739	0.647	0.566	0.895	**0.765**	**0.653**	0.711	4.592
Mistral 7B	0.699	0.615	0.384	0.804	0.624	0.453	0.597	1.624
Mixtral 8x7B	0.694	0.614	0.434	0.824	0.653	0.587	0.634	1.992
Mistral Large	0.767	**0.779**	**0.668**	**0.907**	0.688	0.601	0.735	3.565
SetFit (Baseline)	0.632	0.511	0.612	0.769	0.617	0.462	0.600	0.030
SetFit + Neg Aug	0.672	0.709	0.639	0.848	0.625	0.459	0.658	0.030

Table 1: Comparison of F1 Score of various SOTA LLMs with fine tuned sentence transformer models across AID3 and HINT3 datasets

Dataset	No. of Intents	No. of Queries		
		Train	Valid	
			In Scope	OOS
ALC	8	150	338	128
ADP	13	683	803	91
OADP	13	-	430	56

Table 2: Data Statistics for AID3 dataset

and use variance of the predictions as an uncertainty estimate.

3.2 Datasets

We use SOFMattress, Curekart and Powerplay11 datasets from HINT3 (Arora et al., 2020). We also use **AID3**[4], a collection of three internal multi-label datasets shown in Table 2 - ALC, ADP and OADP, each containing diverse set of PII redacted in-scope and OOS real world queries from shopping domain. Both ALC and ADP contain queries from deployed shopping assistant, whereas OADP contains queries from single turn QnA forum. We use OADP to test out of distribution generalization while using ADP train set. See Appendix A.1 for more details on AID3. Label space size across HINT3 and AID3 datasets varies from 8 till 59 and all these datasets are real world intent detection datasets from deployed TODS which mimic real world scenarios and production challenges like handling intents with very-broad to very-specific scopes, imbalanced training datasets with very few examples per intent. By evaluating on HINT3 and AID3 datasets we include scenarios where there are large number of intents (59 being the maximum label space size) and also include multi-label scenarios (3 out of 6 of our datasets are also multi-label), which makes our evaluation comprehensive.

[4]The splits of all three datasets in AID3 were prepared specifically for experiments done as part of this work and performance on them does not reflect our production system's performance.

3.3 Experiment Setup

SetFit. We use MPNet (Transformers, 2021; Song et al., 2020) as the backbone and use linear layer with sigmoid as differentiable head. We do hyper-parameter search over search space given in Table 6 using Optuna (Akiba et al., 2019) and report best valid set results across all datasets. For MC Sampling, we use 0.1 dropout across hidden and attention layers in the backbone.

LLMs. We use BGE sentence transformer (BAAI, 2023) as the retriever and do grid search over k and t with search space specified in Table 7 and report best valid set results. To prevent LLMs from using any spurious patterns from intent label names, especially for open source datasets, we randomly mask them to Label-xx, where xx is some random integer. We use Claude v3 Sonnet to generate label descriptions for each intent for all datasets and keep them consistent across all LLMs.

Metrics. We use F1-Score as the primary performance metric. Additionally, we use OOS Recall (Larson et al., 2019) and OOS AUCROC to compare model's OOS detection capabilities and use in-scope accuracy to compare their in-scope performance.

See Appendix A.2 for more details on implementation and experiment setup across models.

3.4 Results

Evaluation results from 7 SOTA LLMs across two LLM families (Claude, Mistral) are shown in Table 1. Overall Claude v2, v3 LLMs and Mistral Large have similar performance, but Claude v3

	SOFMattress	Curekart	PowerPlay11	ALC	ADP	OADP	Avg Score
Claude v1 Instant	0.229	0.241	0.122	0.742	0.143	0.000	0.246
Claude v2	0.688	0.701	0.580	0.945	0.330	0.232	0.579
Claude v3 Haiku	**0.736**	0.716	0.561	**0.961**	0.593	0.036	0.601
Claude v3 Sonnet	0.479	0.436	0.402	0.953	0.440	0.036	0.458
Mistral 7B	0.465	0.376	0.205	0.781	0.154	0.018	0.333
Mixtral 8x7B	0.382	0.391	0.455	0.914	0.264	0.036	0.407
Mistral Large	0.646	**0.771**	0.602	0.945	**0.615**	**0.268**	**0.641**
SetFit (Baseline)	0.563	0.293	**0.798**	0.594	0.022	0.000	0.378
SetFit + Neg Aug	0.681	0.592	0.665	0.844	0.154	0.000	0.489

Table 3: Out of Scope Recall at best F1 Score of various SOTA LLMs with fine tuned sentence transformer models across AID3 and HINT3 datasets

	M	SOF Mattress	Cure Kart	Power Play11	ALC	ADP	OADP	Avg score		Avg latency	Delta Avg score		Latency fraction
									w/o OADP			w/o OADP	
SNA	**-**	0.672	0.709	0.639	0.848	0.625	0.459	0.658	0.698	0.030	-0.078	-0.061	0.013
v3 Haiku	**-**	0.815	0.775	0.646	0.849	0.715	0.619	0.736	0.760	2.345	0.000	0.000	1.000
SNA + v3 Haiku	5	0.719	0.734	0.654	0.849	0.653	0.473	0.680	0.722	0.748	-0.056	-0.038	0.319
	10	0.740	0.747	0.671	0.863	0.666	0.489	0.696	0.737	1.005	-0.040	-0.022	0.429
	20	0.730	0.756	0.690	0.855	0.668	0.485	0.697	0.740	1.287	-0.039	-0.020	0.549
Mistral-L	-	0.767	0.779	0.668	0.907	0.688	0.601	0.735	0.762	3.867	0.000	0.000	1.000
SNA + Mistral-L	5	0.712	0.739	0.648	0.872	0.651	0.481	0.684	0.724	1.063	-0.051	-0.037	0.275
	10	0.726	0.747	0.668	0.879	0.662	0.497	0.696	0.736	1.453	-0.038	-0.025	0.376
	20	0.719	0.761	0.692	0.872	0.664	0.498	0.701	0.742	1.657	-0.034	-0.020	0.428

Table 4: Table showing F1 score of two best LLMs (Claude v3 Haiku and Mistral Large) and SetFit + Neg Aug (SNA) hybrid system with varying number of samples (M) from MC dropout.

Haiku is better amongst them with respect to latency. We see that adding negative augmentation to baseline SetFit improves performance by >5%, but still has ∼8% poor predictive performance with respect to best performing LLM. SetFit is about 56 times faster than overall best LLM (v3 Haiku). Additionally, all models see lower performance for OADP as compared to ADP but SetFit has one of the largest drop in performance (∼15%) for OADP as compared to ADP. This shows lack of generalization ability of smaller SetFit models in comparison to LLMs. Table 3 shows that all models including LLMs struggle with OOS detection with poor OOS recall across datasets.

Table 4 shows hybrid system results for two best performing LLMs. We see that with the hybrid system we are able to bring performance gap further down to ∼2% (from ∼6%) for all datasets for which train and test data were from same distribution (i.e. except OADP) and down to ∼4% (from ∼8%) including OADP at ∼50% reduced latency[5]. Increasing number of samples (M) in MC dropout does not increase performance significantly.

[5]Latency would reduce further if we do MC sampling in batches. See latency discussion in Appendix A.2.

4 LLMs and OOS Detection

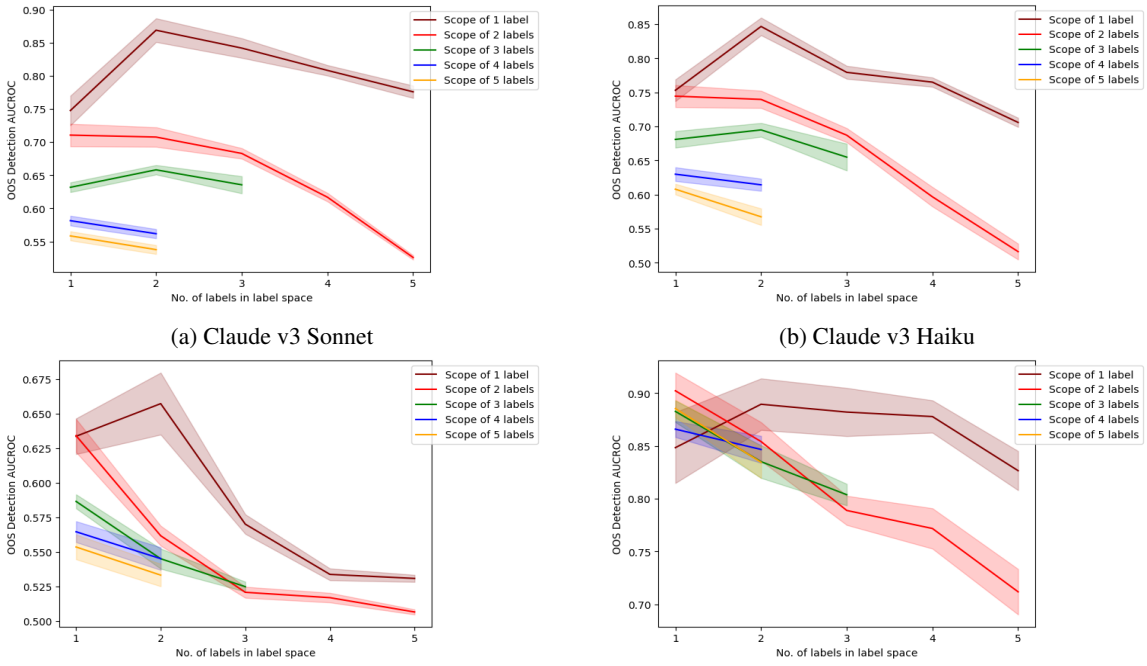
Evaluation results in Sec 3.4 showed that LLMs struggle with OOS detection. Hence, in this section we do a controlled study to better understand behavior of LLM based intent detection with special focus on their OOS detection capabilities (Sec 4.1) and based on the insights propose a novel methodology for OOS detection to improve LLMs performance (Sec 4.2).

4.1 Analyzing LLMs OOS Detection Abilities

We first describe how we setup a controlled experiment to understand how varying "scope of intents" and "no. of labels" in the label space affects LLM performance, and then share our analysis results.

Dataset. We hand curate a dataset with hierarchical label space consisting of 20 leaf intents/labels and two unique parent intents as shown in Table 8. From it, we create new intents with varying scope of $S \in [1, 5]$ labels by randomly combining S leaf intents from the same parent, without replacement. This is realistic because in real world intent scope is driven by bot usecases and scope of APIs/systems which TODS can access.

Experiment Setup. We experiment by varying



(a) Claude v3 Sonnet

(b) Claude v3 Haiku

(c) Mistral 7B

(d) SetFit + Neg Aug (SNA)

Figure 3: Change in OOS detection performance with number of labels in label space and scope of labels.

"scope of intents" by choosing intents from the newly created intents with scope of S labels with $S \in [1, 5]$ and experiment with varying "no. of labels" in label space by randomly picking L different intents of the required scope with $L \in [1, 5]$. Higher S leads to intents with broader scope. We report results based on runs on 10 randomly created datasets for every experiment. See Appendix A.3 for more details on the setup.

Results and Analysis. Fig 3[6] shows how OOS detection AUCROC for LLMs is affected with change in "scope of intents" and "no. of labels" in the label space. We see significantly more performance degradation across all LLMs in comparison to SNA model with increase in "scope of intents" and "no. of labels" in label space. This highlights greater importance of class design for LLMs and suggests that fine grained labels and smaller label spaces are better for LLM's OOS detection capabilities. From Fig 5 in Appendix A.3 we see that in-scope accuracy of LLMs is relatively immune to change in "scope of intents" but degrades with increase in label space size. However, degradation in OOS detection AUCROC is worse than in-scope accuracy degradation with increase in label space size. SNA model on the other hand does show degradation in in-scope accuracy as well with both increase in "scope of intents" and "no. of labels" in label space.

[6]Curves with scope of label > 2 are truncated because we sample and combine leaf nodes without replacement to create non-conflicting intents with bigger scope.

4.2 OOS Detection using LLMs Internal Representations

Motivated by the insights from controlled experiment, we propose a two step methodology using LLM's internal representations to improve its performance which we describe in this section.

4.2.1 Methodology

Fig 4 shows our proposed methodology. During offline processing, we generate representation of each sentence in the training data by obtaining LLM decoder layer's last prompt token's representation. Then during inference, we perform following steps.

Step 1. Firstly, we prompt the LLM to predict one of the in-scope labels without asking it to predict out of scope by completely discarding out of scope from label space given to LLM in the prompt.

Step 2. Then, based on in-scope label predicted from the previous step, we generate incoming query's representation in similar way as done during offline processing using LLM's decoder layer. We then compare this representation with representations of training instances of predicted in-scope label from the first step.

This ensures reduced label space for OOS detection but adds low latency overhead for generating

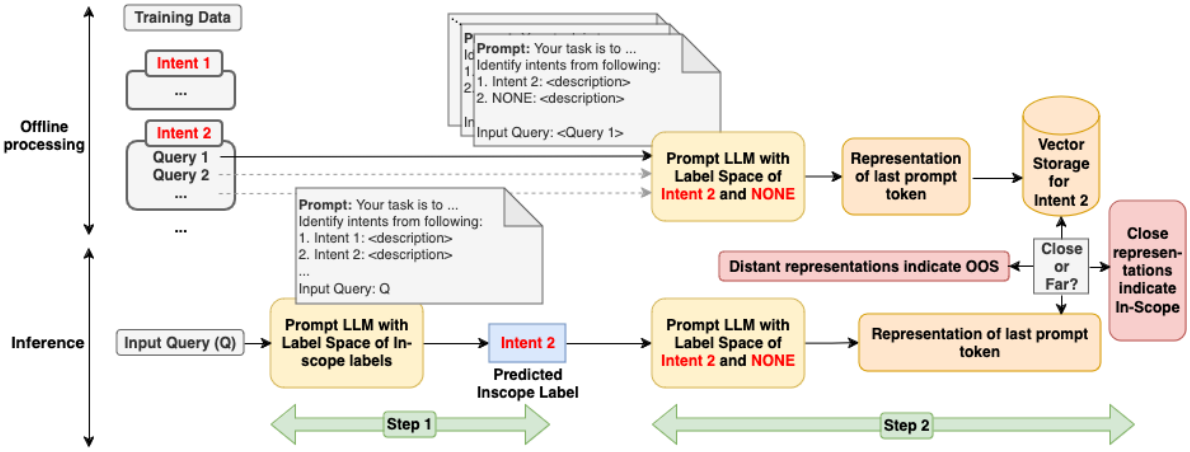


Figure 4: OOS detection using LLM's internal representations

		Overall Accuracy	F1 Score	Inscope Accuracy	Out of Scope Recall
SOF Mattress	**Mistral-7B**	0.705	0.699	0.842	0.465
	Ours	0.748	0.751	0.767	0.715
Curekart	**Mistral-7B**	0.601	0.615	0.863	0.376
	Ours	0.761	0.766	0.736	0.782
Power Play11	**Mistral-7B**	0.357	0.384	0.689	0.205
	Ours	0.780	0.739	0.411	0.950

Table 5: Comparison of our two step methodology with baseline across HINT3 datasets

representations in Step 2. But since we just need to do a forward pass for encoding the prompt, it is significantly faster than autoregressive generation.

Additionally, our proposed OOS detection methodology using LLM's internal representations can be used to improve OOS detection performance of both fine-tuned and non-fine-tuned (base instruct tuned) LLMs. We choose to experiment and show results on non-fine-tuned LLM in Sec 4.2.2 because that is a more practical scenario (as fine-tuning and deployment of a separate instance of LLM for every TODS is prohibitively expensive), but the methodology is generic enough to be used with fine-tuned LLMs as well.

4.2.2 Experiments and Results

Setup. We experiment with base instruct tuned Mistral-7B since its weights are open source. We use cosine similarity for comparing representations in Step 2 and take mean of scores over all training sentences of the predicted intent.

Results. Table 5 compares results of our methodology against baseline LLM methodology discussed in Sec 3.1.2 for HINT3 datasets. We see >5% improvement in performance across datasets at ∼300ms additional latency cost on 1 32GB V100 GPU because encoding the prompt through LLM is cheap. There is drop in in-scope performance as well but that is overcome by significant gains in OOS recall to lead to better overall performance. If needed, threshold in Step 2 of our methodology can be chosen such that drop in in-scope performance is less than an upper limit which in-turn would limit the gains in OOS performance though.

5 Conclusion

Various idiosyncrasies of intent detection task like varying scope of intents within a dataset, need to reject out of scope queries, imbalanced datasets and low resource regime make it a challenging task. In this work we evaluate multiple open source and closed source SOTA LLMs across multiple internal and external datasets for the task of intent detection using adaptive ICL and CoT prompting, compare them with SetFit models and discuss their performance/latency trade-offs. We build a hybrid system which routes queries to LLM when needed and achieves balance between performance and cost. We also propose a novel two step methodology which improves overall LLM performance by >5% across datasets and share insights on how varying scope of intents and number of labels in label space affect LLM performance. We hope our work will be useful for the community to build better TODS.

Limitations

While our current work has broad applicability for the design of accurate and computationally efficient task-oriented dialog systems, there are a few limitations:

Interactive Intent Design. Our current work assumes that intents are specified one-time in the form of examples by human experts, which has been the norm for designing task-oriented conversational assistants. However, there is potential for leveraging LLMs for an interactive class design process. In the future, we plan to investigate the benefits of enabling domain experts to directly interact with these LLMs to interactively define and refine the scope of intents.

Multilingual Support. While our current empirical evaluation was primarily focused on English datasets, the SOTA LLMs we explore already provide multilingual support. To fully harness the potential of our approach, we aim to generalize our ideas to the multilingual setting and evaluate them on diverse dialog datasets across various languages.

Alternative Hybrid Strategies. In the current work, we employ a cascade routing strategy that uses SetFit's uncertainty to combine the SetFit models and LLMs yielding promising results. However, there are additional hybrid strategies worth exploring. Drawing inspiration from active learning literature, we could investigate alternative utility functions, such as information gain to determine when to invoke the LLM alongside the SetFit model. We also plan to compare our approach with model distillation strategies, where the LLM is used to generate synthetic training data to enhance the SetFit models.

Ethics Statement

Our motivation for the current work is to develop computationally efficient and accurate solutions for intent detection, leveraging prior research on sentence transformers and generative language models. As the focus is on intent classification rather than generation, the typical risks associated with generative content do not directly apply. However, as with any machine learning system, there are other important considerations, such as potential biases in the training data or constituent pre-trained models, the possibility of misuse, and challenges in establishing full accountability. Since our approach incorporates generative LLMs, any application of the proposed ideas needs to be mindful of any biases present in those models. Overall, the proposed methodological innovations are intended for benign applications and are not associated with any direct negative social impact. The datasets used in this research include public benchmarks and proprietary datasets from safe ecommerce categories, with personally identifiable information (PII) redacted to ensure customer privacy. To enable reproducibility, we plan to share these datasets as a community after internal approvals.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631.

Anthropic. 2023. Welcome to claude. https://docs.anthropic.com/claude/docs/intro-to-claude. Accessed: 30-04-2024.

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.

BAAI. 2023. Bge retriever. https://huggingface.co/BAAI/bge-base-en-v1.5. Accessed: 30-04-2024.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Dialogflow. 2010. Intent in dialogflow. https://cloud.google.com/dialogflow/cx/docs/concept/intent. Accessed: 11-07-2024.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *Preprint*, arXiv:1506.02142.

Gartner. 2022. Gartner predicts conversational ai will reduce contact center agent labor costs by $80 billion in 2026. https://www.gartner.com/en/newsroom/press-releases/2022-08-31-gartner-predicts-conversational. Accessed: 11-07-2024.

Maarten Grootendorst. 2020. Keybert: Minimal keyword extraction with bert.

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.

AWS LEX. 2017. Intent in aws lex. https://docs.aws.amazon.com/lex/latest/dg/how-it-works.html. Accessed: 11-07-2024.

Bing Liu and Ian R. Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *CoRR*, abs/1609.01454.

Bo Liu, Liming Zhan, Zexin Lu, Yujie Feng, Lei Xue, and Xiao-Ming Wu. 2024. How good are llms at out-of-distribution detection? *Preprint*, arXiv:2308.10261.

Meta. 2023. Meta llama. https://llama.meta.com/. Accessed: 11-07-2024.

Mistral. 2023. Mistral ai models. https://docs.mistral.ai/getting-started/models/. Accessed: 11-07-2024.

OpenAI. 2022. Introducing chatgpt. https://openai.com/index/chatgpt. Accessed: 30-04-2024.

OpenAI, Josh Achiam, and Steven Adler et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *Preprint*, arXiv:2302.06476.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *arXiv preprint arXiv:2004.09297*.

Sentence Transformers. 2021. Sentence transformer mpnet base v2. https://huggingface.co/sentence-transformers/all-mpnet-base-v2. Accessed: 10-07-2024.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022a. Efficient few-shot learning without prompts. *Preprint*, arXiv:2209.11055.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022b. Setfit - efficient few-shot learning with sentence transformers. https://github.com/huggingface/setfit. Accessed: 10-07-2024.

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. *Preprint*, arXiv:2402.17256.

Haode Zhang, Yuwei Zhang, Li-Ming Zhan, Jiaxin Chen, Guangyuan Shi, Xiao-Ming Wu, and Albert Y. S. Lam. 2021. Effectiveness of pre-training for few-shot intent classification. *CoRR*, abs/2109.05782.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *Preprint*, arXiv:2303.18223.

A Appendix

A.1 AID3 Dataset

ALC contains upper funnel shopping queries for 1 HCTP[7] category while **ADP** contains lower funnel queries for 6 HCTP categories. **OADP** also contains lower funnel queries from >10 HCTP categories.

A.2 Experiment Setup

For training SetFit models, we use SetFit library (Tunstall et al., 2022b) for implementation. Hyperparameter search space for SetFit model's training is given in Table 6.

For **negative augmentation**, we use KeyBERT (Grootendorst, 2020) for identifying keywords. For every identified keyword, random 50% of the times we completely remove it, and remaining 50% of the times we replace it with a randomly generated string of 5 characters. For eg: "looking for a gaming laptop" can get converted into "looking for a" or "looking for a XYCVD QSDER" or "looking for a RTYUH". Since these augmented OOS sentences have similar lexical pattern as in-scope training sentences, these are expected to help the model avoid latching onto any spurious patterns and help overall learning, which shows up in results as well (See 3.4). If U is the set of randomly sampled augmentations to add to train set, then we keep $|U| = 0.2*|D|$, where $|D|$ is size of train set.

For **choosing ICL examples** for LLMs, we do grid search over ideal number of ICL examples and retriever threshold whose search space is shown in Table 7. We keep ordering of labels in the prompt

[7] High Consideration Technical Products

Hyperparameter Name	Range of Values
body_learning_rate	From 5e-6 till 5e-5
head_learning_rate	From 1e-3 till 1e-2
num_epochs	From 3 till 10
batch_size	Amongst [8, 16, 32, 64]
n_trials	10

Table 6: Hyperparameter search space for SetFit model training

Hyperparameter Name	Range of Values
k (no. of ICL examples)	[0, 1, 5, 10, 20]
t (retriever threshold)	[0.00001, 0.3, 0.5, 0.7]

Table 7: Hyperparameter search space for choosing ICL examples for LLM based intent detection

fixed across all experiments and keep ICL examples within a label in descending order of similarity with incoming query.

For **Monte Carlo (MC) sampling** from SetFit models for hybrid system, we look at variance of the predictions as an uncertainty estimate. Specifically, let $p_i \in P \forall i \in [1, M]$ be the predicted label with maximum score from i^{th} sample, where M is the maximum number of samples. Then, we consider the prediction to be uncertain if number of different values of $p_i \forall i \in [1, M]$ is greater than 1 or less than $M/2$. We add upper limit of $M/2$ for stability.

For **latency calculations of hybrid system**, we also add time for doing multiple forward passes sequentially through SetFit in MC sampling procedure keeping memory needs constant. Since maximum $M = 20$ in our experiments, if we consider that sampling can be done in batches, then latency of hybrid system would go further down.

For SetFit models, we calculate OOS AUCROC by considering max predicted score amongst all labels. For black box LLMs, we calculate OOS AUCROC by considering score as 1 if LLM predicts an in-scope label, 0 otherwise.

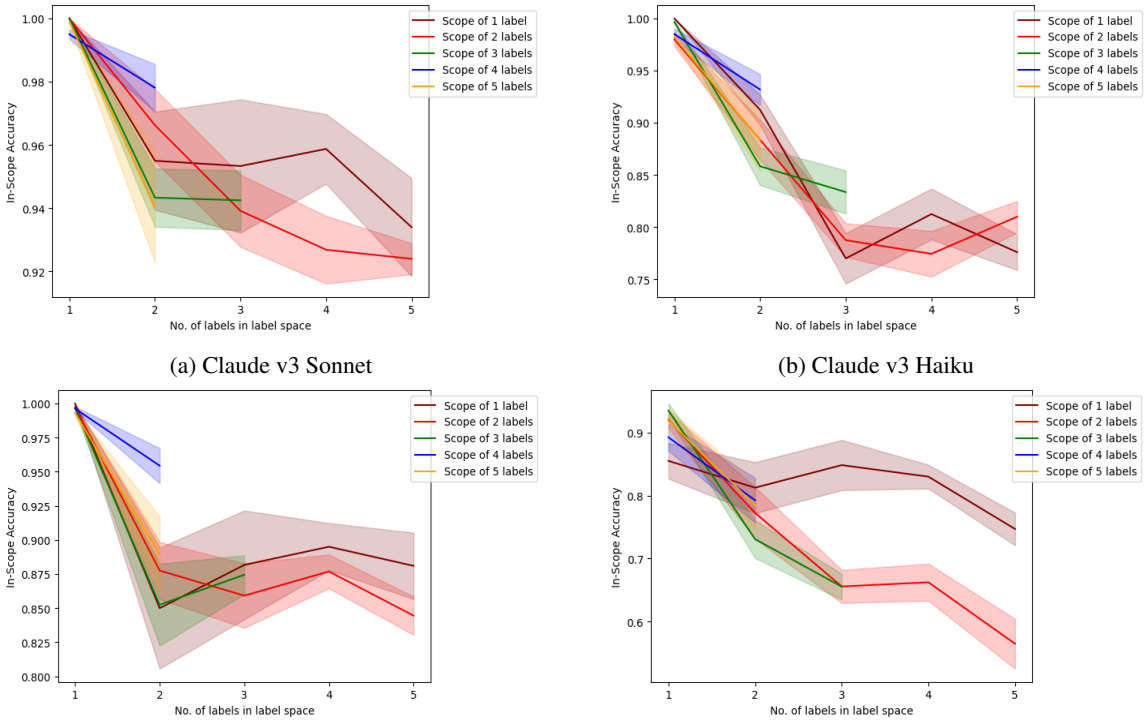
A.3 Controlled Experiment

Setup. For our controlled experiment dataset, we hand-curate 10 utterances per leaf intent, random 5 of which we use in train and other 5 we use in test for every run. We also use three paraphrases (pre-curated) of each test utterance in our test set for every run to test generalization across utterance variants. For controlled experiment, we train all SetFit models with batch size of 16 and 5 epochs. For ICL examples selection with LLMs, we use max 5 ICL examples with retriever threshold of 1e-5. Since we execute every experiment 10 times with randomly created dataset, we are unable to experiment with other hyperparameters due to compute costs. Since we do controlled experiments to develop better understanding of LLM behavior, keeping these hyper-parameters fixed is okay.

Results. Table 8 shows example queries from each intent from controlled experiment dataset. From controlled experiments, Fig 5 and Fig 6 show change in In-Scope accuracy and OOS Recall with number of labels in label space and scope of labels, respectively.

Level 1 class	Level 2 class	Example Utterance
Product Recommendation	Static Product Attribute based	show laptop with 8gb RAM
Product Recommendation	Similarity/Comparison with other products based	show laptop comparable to the Dell XPS 13
Product Recommendation	Compatibiliy with other products based	show laptop bags compatible with Dell XPS 15
Product Recommendation	Offers based	show laptop with HDFC bank EMI offers
Product Recommendation	Customer Reviews/Ratings based	show laptops whose battery life is highly praised by users
Product Recommendation	Budget based	show laptops under 50k
Product Recommendation	Purpose/Usecase based	show laptops suitable for graphic design work
Product Recommendation	Warranty/Return policy based	show laptops with hassle-free return options
Product Recommendation	Delivery ETA based	show laptops that can be delivered within the next week
Product Recommendation	Past sales based	show the most popular laptop models recently
Product Evaluation	Static Product Attribute based	does this laptop have 8gb RAM
Product Evaluation	Similarity/Comparison with other products based	is this laptop comparable to the Dell XPS 13
Product Evaluation	Compatibiliy with other products based	are these laptop bags compatible with Dell XPS 15
Product Evaluation	Offers based	does this laptop have HDFC bank EMI offers
Product Evaluation	Customer Reviews/Ratings based	are these laptops whose battery life is highly praised by users
Product Evaluation	Budget based	are these laptops under 50k
Product Evaluation	Purpose/Usecase based	are these laptops suitable for graphic design work
Product Evaluation	Warranty/Return policy based	do these laptops have hassle-free return options
Product Evaluation	Delivery ETA based	can these laptops be delivered within the next week
Product Evaluation	Past sales based	are these the most popular laptop models recently

Table 8: Example utterance for each leaf intent from controlled experiment dataset used to understand behavior of LLM based intent detection.



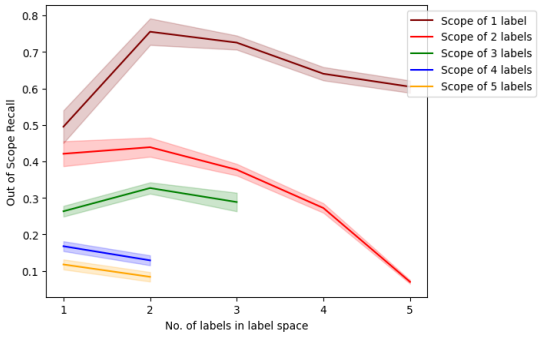
(a) Claude v3 Sonnet

(b) Claude v3 Haiku

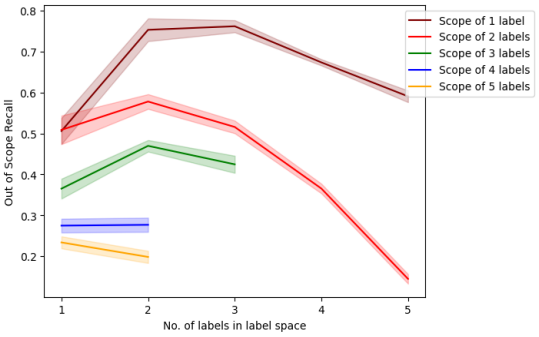
(c) Mistral 7B

(d) SetFit + Neg Aug

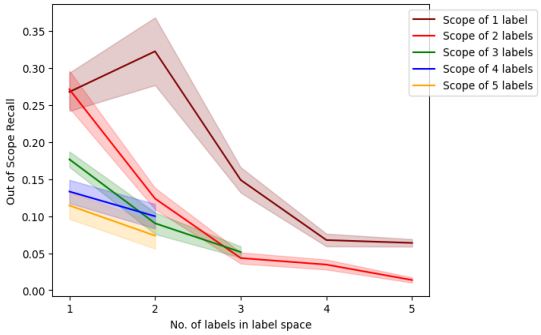
Figure 5: Change in In-Scope accuracy with number of labels in label space and scope of labels.



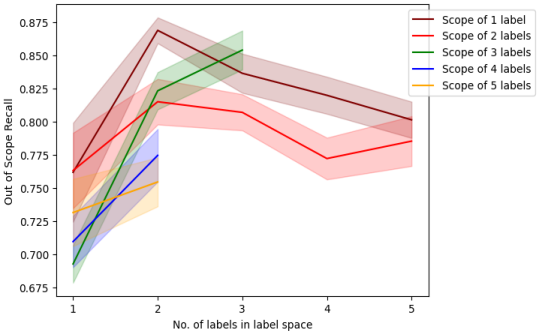
(a) Claude v3 Sonnet



(b) Claude v3 Haiku



(c) Mistral 7B



(d) SetFit + Neg Aug

Figure 6: Change in OOS Recall with number of labels in label space and scope of labels.