Navigating the Unknown: Intent Classification and Out-of-Distribution Detection Using Large Language Models

Anonymous ACL submission

Abstract

Out-of-Distribution (OOD) detection is a challenging task that requires great generalization capability for the practicality and safety of task-oriented dialogue systems (TODS). With the dawn of large language models (LLMs), their enhanced ability to handle diverse patterns and contexts may aid in addressing this challenging task. In this paper, we investigate the current performance of LLMs in the near-OOD setting, where OOD queries belong to the same domain but different intents. To take advantage of out-of-the-shelf capabilities of LLMs, we do not use fine-tuning. We study the performance of one of the leading frontier models, GPT-4o, in 3 well-known public datasets and 3 in-house datasets, using 10 different methods and prompt variations. We study the performance of different prompts and techniques in Gemini 1.5 Flash and Llama 3.1-70b. We investigate the effect of increasing the number of In-Distribution (ID) intents. We propose a novel hybrid method that is cost-efficient, high-performing, highly robust, and versatile enough to be used with smaller LLMs without sacrificing performance. This is achieved by combining ID success of smaller text classification models and high generalization capabilities of LLMs in OOD detection.

1 Introduction

As the field of natural language processing (NLP) progresses rapidly, task-oriented dialogue systems (TODS) are experiencing a significant increase in their overall capabilities. Their efficiency, accessibility, and coverage have improved with the emergence of the Large Language Model (LLM) paradigm, as shown by Zhao et al. (2024).

In TODS, natural language understanding (NLU) tasks begin with intent detection, where the user query is mapped to a set of known intents to control the flow of the dialogue, select appropriate knowledge sources, and so on. Prior to LLMs, this task was handled by transformer models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). These models operate under the closed world hypothesis, as discussed by Lang et al. (2023), and can only detect what is present in their training data. However, the real world is open-ended, and users often submit unseen or unrelated queries. The need to reject such out-of-scope queries — illustrated in Figure 1 — has led to the development of Out-of-Distribution (OOD) detection systems, described in surveys by Lang et al. (2023) and Yang et al. (2024). The improved generalization capabilities introduced by LLMs hold great promise for addressing this challenge.

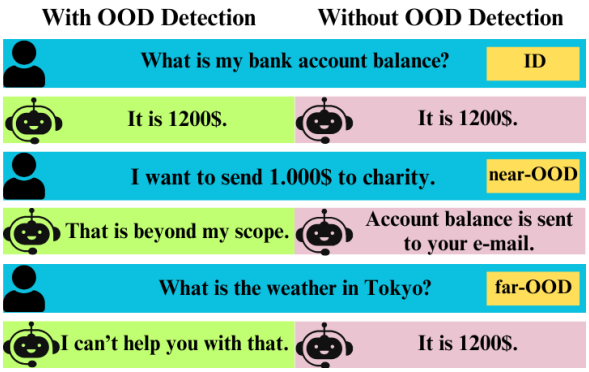

Figure 1: An example of a TODS dialogue in banking domain. Without OOD detection support, conversations may go astray.

As an intent detection model is trained with a fixed set of intents and utterances, the problem of OOD detection can be framed as the task of identifying a distribution shift. In NLP, two primary types of distributional shift are typically discussed. The first is *semantic shift*, where OOD queries arise from a different intent space and must be filtered before being incorrectly mapped to known intents. The second is *covariate shift*, where the intent space remains the same, but the input distribution changes, leading to novel utterances for

familiar intents, as discussed by Lang et al. (2023) and Yang et al. (2024). In this paper, we focus on semantic shifts.

The challenges of OOD detection extend beyond distributional shifts. Training datasets, often constructed by domain experts without machine learning expertise, may be imbalanced, with certain intents represented by very few examples. Additionally, the scope of intents can vary widely — some may be fine-grained while others are overly broad.

This work focuses on a more demanding subset of the problem, known as *near-OOD* detection (Liu et al., 2024a), also referred to as In-Domain and Out-of-Scope Detection (ID-OOS) by Zhang et al. (2022). In this setting, unknown examples do not come from entirely different domains but instead belong to different intent labels within the same general distribution. Near-OOD detection mimics the presence of long-tail intents unseen during training by identifying semantically similar yet label-divergent in-domain examples.

In this paper, we investigate the performance of LLMs on near-OOD detection without fine-tuning. We aim to provide insights into the latest models, their performance with different techniques, and how they can be better utilized. Our main contributions are:

1. We investigate the performance of GPT-4o in the challenging task of near-OOD detection using 3 well-known public datasets, and 3 in-house datasets. We use zero-shot and few-shot prompts and their k-Nearest (k-N) variations, as well as a hybrid method with various prompting strategies. The results demonstrate exceptional performance, surpassing prior studies on the benchmarks.

2. We introduce a novel hybrid method. Our method, on average, is the highest performing amongst all strategies. It also improves the performance of smaller or open source LLMs to match GPT4o, reduces the input-token numbers and cost, is robust, and easy to implement.

3. We compare the performance of Gemini 1.5 Flash, Llama 3.1-70b, and GPT-4o when using zero-shot, few-shot, and two hybrid methods.

4. We study the effect of increasing the number of ID intents in 3 datasets using GPT-4o.

2 Related Work

Hendrycks et al. (2020) systematically measures performance in the OOD detection task. Various older methods are seen to perform worse than random guess, but pre-trained models such as BERT and RoBERTa (Devlin et al., 2019; Liu et al., 2019) have performed well and are accepted as the industry standard. Numerous studies have been conducted on the fine-tuning performance of such models. Uppaal et al. (2023) presents a systematic comparison of fine-tuning methods. It is observed that pre-trained models achieve near-perfect OOD detection in *far-OOD*, which is the case where the distributional shift corresponds to a domain shift.

In near-OOD problems where no examples of OOD data are given, there have been various efforts. Zhang et al. (2022) fine-tune different varieties of BERT-based models and observe that fine-grained near-OOD problems with few examples remain a significant challenge. Zhan et al. (2021) and Wang et al. (2023) employ discriminative fine-tuning methods, while Zhou et al. (2021) and Zeng et al. (2021) investigate contrastive fine-tuning methods. Since models in real-world scenarios typically have no access to OOD data, Baran et al. (2023) focus on post hoc methods.

The main categories of methods for OOD detection problems are summarized in recent surveys by Lang et al. (2023) and Yang et al. (2024) as:

(1) *output-based* (Hendrycks and Gimpel, 2018; Liu et al., 2020; Qian et al., 2022),

(2) *gradient-based* (Huang et al., 2021),

(3) *density-based* (Arora et al., 2021), and

(4) *distance-based* (Sun et al., 2022; Podolskiy et al., 2022).

Zawbaa et al. (2024) present an output-based method called Dual Encoder for Threshold-Based Re-Classification (DETER) that achieves significant improvements in near-OOD problems. In addition, as seen in the works of Rawat et al. (2021) and Kim et al. (2023), creating synthetic OOD data is also a valuable approach. Li et al. (2024) study the effect of employing ChatGPT (OpenAI, 2022) in creating synthetic near-OOD data.

LLMs such as GPT-4o (OpenAI, 2024a), Gemini 1.5 Flash (Google, 2024), and Llama 3.1 (Grattafiori, 2024) have become the leading paradigm in NLP. Their performance in multiple NLP tasks such as machine translation, information extraction, summarization, and clustering is impressive (Zhao et al., 2024). However, the study of the performance of LLMs on the OOD detection task is still lacking. Arora et al. (2024) investigates the In-Context Learning (ICL) ability in the far-OOD detection task. Seven of the most recent LLMs are tested together with a hybrid model that

utilizes SetFit (Tunstall et al., 2022) with negative data augmentation. Their hybrid model aims to reduce cost and latency, and they also propose a two-step methodology that utilizes the representation of the last prompt token of the LLM decoder layer. Wang et al. (2024) examines the zero-shot and few-shot capabilities of ChatGPT in the near-OOD setting and compares them with unsupervised SOTA methods, including those proposed by Mou et al. (2022). They find that ChatGPT struggles when the number of in-distribution (ID) intents is large. In addition, they claim that fine-grained intent labels are challenging for ChatGPT and that it is difficult to transfer knowledge from ID examples to OOD tasks.

3 Methodology

3.1 Problem Formulation

Let $S = \{i_1, i_2, \ldots, i_N\}$ be the predefined set of N intents. Let $q = \{t_1, t_2, \ldots, t_n\}$ be the user input query, composed of the tokens t_i. The output is prediction $i_{pre} \in S \cup \{\text{OOD}\}$. We employ 4 different performance metrics: total (ALL), OOD and ID macro-F1; OOD recall.

3.2 Datasets

We employ 3 widely used benchmarks and a collection of internal production data. The benchmark datasets are CLINC150 (Larson et al., 2019), BANKING77 (Casanueva et al., 2020), and DSTC Finance dataset from DSTC11 Track 2 (Gung et al., 2023b,a). In the test splits we limit the utterance per intent number to 10. To the best of our knowledge, DSTC Finance is used in similar tasks like intent clustering and open intent induction, but it is the first time an OOD detection paper has utilized it. We employ their utterance test set and split it $50\% - 50\%$ in a stratified fashion to be used as train and test sets. The in-house collection dataset is called **BIT3**[1], consisting of banking, insurance, and telecommunication domains. The corresponding subsets are called **BIT3**-*bank*, **BIT3**-*ins*, and **BIT3**-*tele*, and their statistics can be seen in Table 1. These datasets are collected directly from real-world applications. They are PII redacted and cleaned by 3 experts.

[1]This dataset is created for the purpose of this study; hence, experiment results may differ from production performance.

Dataset	# Intent	Train		Test	
		Size	UPL	Size	UPL
CLINC150	150	15000	100	1500	10
BANKING77	77	10003	130	770	10
DSTC	38	565	15	365	9.6
BIT3-*bank*	88	852	10	546	6
BIT3-*ins*	62	340	5	336	5
BIT3-*tele*	58	273	5	270	4.5

Table 1: The statistics of datasets. DSTC stands for DSTC Finance. UPL stands for average utterance per label.

3.3 Methods

3.3.1 Baseline

As a baseline method, we adopt a threshold-based classification system using a fine-tuned, quantized version of "sentence-transformers/all-distilroberta-v1" (Reimers and Gurevych, 2019; Sanh et al., 2020), reflecting a basic, production-oriented text classification pipeline. We attach a lightweight feedforward classification head, train it with cross-entropy loss while keeping the encoder frozen, and classify examples as OOD when prediction confidence falls below a threshold ($T = 0.9$). Full architectural and training details are provided in Appendix A.1.

3.3.2 Zero-Shot Detection

We use two variations of zero-shot detection. First, we create tools for each intent using only intent names, adding an OOD class as a tool named "fallback." All tools are then fed to the LLM. No intent utterance is used; only intent names. In this and the following experiments, we force LLMs to use tool calling, removing formatting issues. To our knowledge, this is the first use of tool calling for this task. Example is given in Appendices A.2.

Second; using OpenAI's "text-embedding-ada-002" (OpenAI, 2023) as a text embedding model, we choose the k-Nearest tools with $k = 5$, and only feed them to the LLM. This variation is called *zero-shot k-Nearest* method. Related prompts are in the Appendices A.3.

3.3.3 Few-Shot Detection

We employ two variations of few-shot detection.

First, we create tools for each intent using both intent names and example utterances. We experiment with 10 and 45 example utterances per intent. In the **BIT3** dataset, there are cases where an intent has fewer examples. In those cases, we use all the examples in a few-shot prompting.

Second, using OpenAI's "text-embedding-ada-002" as a text embedding model, we choose the k-Nearest tools with $k = 5$, and only feed them to the LLM. This method is called *few-shot k-Nearest*. Related prompts are in the Appendices A.4.

3.3.4 Hybrid-Methods

Arora et al. (2024) shows that LLMs' OOD detection performance drops as the number and scope of intents grow. Moreover, with a high number of intents, few-shot prompting becomes costly and slow. To address these challenges, we devise a two-step hybrid method. First, the baseline model predicts an ID intent, as its ID performance is satisfactory. Then, using examples and specialized prompts, we ask GPT-4o to verify or reject the baseline's prediction—thus focusing LLM usage solely on OOD detection and avoiding unrelated examples in prompts. Chain-of-thought style prompts are generated with OpenAI o1 (OpenAI, 2024b) and customized for the task. Few-shot toy examples highlighting ID vs. OOD distinctions are added, as detailed in the appendices. We use three different prompts: Balanced, OOD-Focused, and Contrastive.

Balanced. This is a concise prompt that aims to be unbiased in predicting ID or OOD. It also prioritizes efficiency and clarity and uses a step-by-step approach. We use at most 45 utterances of the predicted intent.

OOD-Focused. This is a skeptic version of the balanced prompt, more suited for cases where OOD intents are easier to miss. The inclination is towards detecting the OOD examples.

Contrastive. This prompt uses cross-referencing between two sets of examples, one of which is a set of positive examples and the other is a set of negative but similar examples that are obtained from the second and third predictions of the front end. This aims for higher coverage around the query and creates further separation between challenging near-OOD examples.

4 Experiments and Results

We devise three different experiments. In Sec 4.1, we want to see how different approaches compare to each other. To that end, we employ all the methods in Sec 3.3 using GPT-4o as the LLM representative. In Sec 4.2, we aim to compare the performance of different LLMs using zero-shot and few-shot prompting with all the intents, and using the hybrid method with balanced prompt and contrastive prompt. For this purpose, we use GPT-4o (OpenAI, 2024a), Gemini 1.5 Flash (Google, 2024), and Llama 3.1-70b-instruct (Grattafiori, 2024) . Lastly, in Sec 4.3, the goal is to see the effect of the number of ID intents on OOD detection performance. We use the same LLMs and methods as in the previous experiment.

4.1 Comparison of Methods

To compare the methods, we use all 6 datasets. We split the labels $50\% - 50\%$ as ID and OOD, respectively. We use F1 scores as well as OOD recall, since it is important to see how much of the OOD examples are captured without sacrificing ID performance. The results are shown in Table 2. The best score is shown in bold, and the second best is underlined. In Figure 2, the performance in BANKING77 is visualized.

The baseline model's ID performance is comparable to LLMs on public datasets. It achieves the highest ID F1 on CLINC150 and outperforms GPT-4o on BANKING77 in both zero-shot and few-shot settings. In **BIT3**, GPT-4o leads. Few-shot generally outperforms zero-shot, though gains diminish or slightly drop beyond 10–45 examples, indicating an upper bound. In OOD performance, we see a significant improvement when using LLMs. OOD F1 is significantly higher than the baseline in the vast majority of cases. It is also important to note that in all the datasets, the highest OOD F1 belongs to the hybrid methods. In public sets, different prompts perform better, but in **BIT3** the contrastive method seems to outperform the rest, which may suggest the robustness of this strategy in real life. Through all datasets but **BIT3**-*ins* we see few-shot improves performance over zero-shot in terms of ALL F1 scores. In public datasets, zero-shot k-Nearest decreases zero-shot performance, whereas in **BIT3** it results in a slight increase. This may be the result of noise in intent names in **BIT3**. A similar situation occurs in few-shot as well. In public datasets, k-Nearest is beneficial both in 10 and 45 example few-shots whereas in **BIT3** there may be a slight drop in performance.

Comparing balanced prompt and OOD-focused prompt variations of the hybrid method, it is observed that OOD recall is significantly increased on all 6 datasets (up to 21 points) with a similar or lower decrease on ID F1. This suggests that a simple prompt engineering can shift the balance in the ID-OOD trade-off without having a significant decrease on the either side.

In all 6 datasets, it is observed that OOD recall

		BANKING77				CLINC150				DSTC Finance			
		ALL	ID	OOD		ALL	ID	OOD		ALL	ID	OOD	
Model	**Method**	**F1**	**F1**	**Recall**	**F1**	**F1**	**F1**	**Recall**	**F1**	**F1**	**F1**	**Recall**	**F1**
Baseline	Threshold	76.7	92.7	42.5	61.1	82.4	**95.9**	56.0	70.4	86.1	89.2	78.5	84.2
GPT-4o	Z-S	69.4	77.2	66.0	71.6	83.7	88.5	82.9	84.5	87.7	85.7	92.3	90.4
	Z-S k-N	67.2	77.2	58.9	67.0	76.5	83.8	74.5	78.7	87.3	87.0	87.8	90.4
	F-S (10)	78.7	90.6	54.9	68.9	91.3	94.8	87.1	90.6	93.3	91.3	92.7	94.5
	F-S (10) k-N	79.6	90.5	58.6	71.6	90.0	92.2	91.1	90.9	94.5	91.9	94.2	94.8
	F-S (45)	78.4	90.7	53.9	68.8	91.6	95.7	85.6	90.2	94.9	**93.0**	92.6	95.0
	F-S (45) k-N	79.5	90.8	56.4	70.4	92.2	94.8	90.5	92.4	94.8	92.6	93.1	95.1
Hybrid	Balanced	83.2	93.0	62.7	75.0	**95.7**	95.8	96.8	**96.2**	**95.1**	92.1	95.9	**96.0**
	OOD-Focused	**86.3**	89.7	**84.0**	**85.9**	91.4	90.5	**99.4**	93.6	92.3	88.4	**98.4**	94.0
	Contrastive	85.8	**93.3**	70.7	80.2	95.3	95.5	96.4	95.7	94.4	91.9	94.1	94.7
		BIT3-*bank*				**BIT3**-*ins*				**BIT3**-*tele*			
		ALL	ID	OOD		ALL	ID	OOD		ALL	ID	OOD	
Model	**Method**	**F1**	**F1**	**Recall**	**F1**	**F1**	**F1**	**Recall**	**F1**	**F1**	**F1**	**Recall**	**F1**
Baseline	Threshold	72.1	74.7	81.7	79.1	69.6	72.3	85.6	79.4	63.9	66.3	86.3	79.1
GPT-4o	Z-S	68.7	72.9	84.9	77.5	74.4	77.7	80.7	80.3	77.8	82.6	81.2	85.4
	Z-S k-N	69.8	74.1	84.1	77.9	75.5	81.1	76.1	79.2	77.8	**84.1**	77.8	84.2
	F-S (10)	86.0	90.0	84.3	86.9	83.7	**91.1**	70.7	80.3	73.4	78.0	78.6	81.3
	F-S (10) k-N	84.5	88.4	85.1	85.7	83.1	87.0	82.7	85.9	75.6	78.7	85.5	83.2
	F-S (45)	**86.8**	**90.2**	84.5	87.5	83.7	90.5	72.2	81.3	72.6	76.7	80.4	81.0
	F-S (45) k-N	85.3	89.0	86.3	87.0	83.9	87.6	83.0	86.5	72.3	75.0	86.6	82.7
Hybrid	Balanced	84.8	87.3	84.6	86.6	82.6	86.4	79.8	84.0	80.2	83.1	83.4	87.6
	OOD-Focused	81.3	81.4	**94.0**	85.9	80.0	82.3	85.6	84.3	75.6	75.6	**93.0**	87.4
	Contrastive	86.1	87.7	89.6	**88.5**	**84.5**	85.7	**90.2**	**88.5**	**80.6**	82.3	88.5	**88.3**

Table 2: The performance metrics across different methods and datasets. All results are average of 3 runs. Z-S stands for zero-shot, and F-S stands for few-shot.

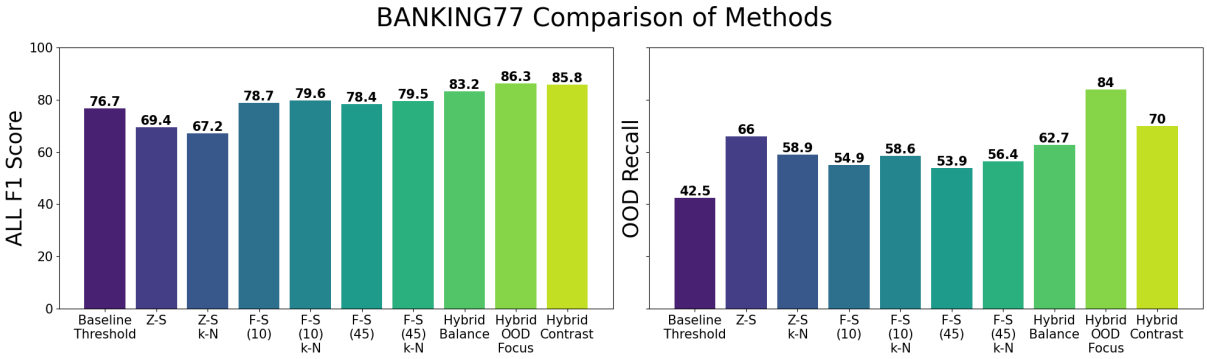

Figure 2: The comparison of all the methods in BANKING77 in terms of ALL F1 and OOD Recall.

is decreased (up to 8 points) on zero-shot prompt when the k-Nearest method is used, while it is increased on few-shot prompt (up to 12 points) when the k-Nearest method is used. This may suggest that using at least a few example utterances may help to detect related intents much better. However, this observation needs further investigation.

4.2 Comparison of LLMs

To see whether the high performance in OOD detection depends on the success of GPT-4o, we compare the performance of GPT-4o, Gemini 1.5 Flash, and Llama 3.1-70b in CLINC150, DSTC Finance, and **BIT3**-ins. We split the labels $50\% - 50\%$ into ID and OOD, respectively. We compare each LLM using the zero-shot and few-shot (10) prompting methods. Then, we complete using the hybrid method with balanced prompt and contrastive prompt. The results are shown in Table 3. In Figure 3 results of CLINC150 are visualized.

The largest model of all performs the best. The greatest performance difference is in zero-shot method. Few-shot generally improves ALL F1 score by about $\sim$10 points. In Gemini 1.5 Flash and Llama 3.1-70b, there is significant improvement when using hybrid methods. This may indicate that

		CLINC150				DSTC Finance				BIT3-*ins*			
		ALL	ID	OOD		ALL	ID	OOD		ALL	ID	OOD	
Method	**Model**	**F1**	**F1**	**Recall**	**F1**	**F1**	**F1**	**Recall**	**F1**	**F1**	**F1**	**Recall**	**F1**
Zero-Shot	GPT-4o	**83.7**	**89.0**	**81.3**	**83.9**	**88.2**	**85.8**	**93.6**	**91.1**	**75.0**	**78.5**	**79.5**	**80.1**
	Gemini	74.9	86.1	54.8	67.3	76.0	81.1	52.2	67.2	69.9	77.3	48.9	61.5
	Llama	72.6	81.6	55.5	66.7	81.9	84.8	73.1	82.4	68.0	79.3	45.6	58.0
Few-Shot	GPT-4o	**91.3**	**94.8**	**87.1**	**90.6**	**93.8**	91.5	**92.7**	94.5	**83.7**	**91.1**	**70.7**	**80.3**
	Gemini	83.7	87.1	86.4	87.9	93.7	90.6	95.7	**95.1**	76.3	84.2	63.9	74.1
	Llama	80.7	78.3	56.8	69.9	91.3	**91.5**	85.2	91.5	76.3	84.3	55.6	68.9
Hybrid Balanced	GPT-4o	**95.7**	95.8	**96.8**	**96.2**	**95.1**	92.1	**95.9**	**96.0**	**82.6**	86.4	**79.8**	84.0
	Gemini	94.7	94.9	96.5	95.4	93.5	92.1	91.2	93.3	82.5	86.7	78.9	**84.7**
	Llama	95.0	**95.8**	95.1	95.4	94.5	**92.1**	91.4	94.2	81.1	**87.2**	72.5	81.0
Hybrid Contrastive	GPT-4o	**95.3**	95.5	**96.4**	**95.7**	94.4	91.9	94.1	94.7	**84.5**	85.7	**90.2**	**88.5**
	Gemini	94.8	95.2	96.1	95.3	93.9	91.6	93.7	94.4	81.7	84.7	85.4	85.6
	Llama	94.9	**95.6**	95.6	95.4	**94.4**	**92.1**	**94.1**	**94.8**	82.9	**86.5**	82.3	84.9

Table 3: Performance comparison for LLMs using various techniques. GPT-4o stands for gpt-4o-2024-08-06, Gemini stands for Gemini 1.5 Flash, and Llama stands for Llama 3.1-70b. All results are average of 3 runs.

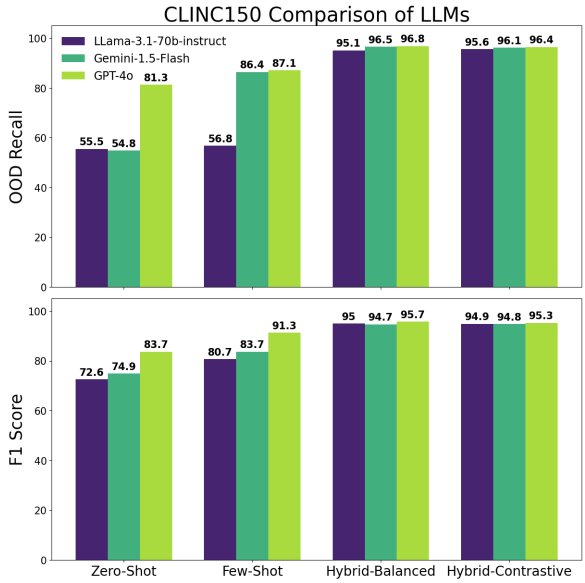

Figure 3: The comparison of different LLMs in CLINC150 in terms of ALL F1 and OOD recall.

a large classification task may be challenging for them, and narrowing the task scope given to them is highly beneficial. We see comparable performance for all the models when we use hybrid methods, which demonstrates the robustness of our proposed system. This shows that through intelligent design, smaller models may perform on par with the largest models.

4.3 Comparison of Number of Intents

Wang et al. (2024) observes that LLMs may struggle with fine-grained near-OOD cases where there is also a large number of intents (30–40). As LLMs progress rapidly, checking the performance in this aspect is a necessity. As a good representative of current LLMs, we use GPT-4o in this experiment. CLINC150 , DSTC Finance, and **BIT3**-ins datasets are used. We split 25% of their intents to the OOD class. All of the OOD examples are used in the test set. Then, we sample the rest of the intents $33\%, 66\%, 100\%$ as ID intents. To make a fair comparison, these splits are crafted to ensure that smaller splits are a subset of the larger splits. We report OOD recall and binary F1 between ID and OOD classes in Table 4. To calculate the binary F1 metric, we assume all ID classes have the same label; thus, we only measure the ability of models to differentiate OOD samples from ID samples. The results are visualized in Figure 4.

In CLINC150 the OOD test sets make up 38 intents resulting in $33\%, 66\%, 100\%$ sets to have $37, 74, 112$, respectively. We see high OOD performance even in zero-shot methods through all intent numbers. The performance drops from 37 intents to 112 intents significantly ($\sim$10) in zero-shot and few-shot methods. The performance drop in hybrid methods is approximately half of that amount, indicating the robustness of our hybrid approach.

In DSTC Finance, the OOD test sets make up about 9 intents, leaving $9, 18, 29$ for the ID splits. The overall performances are higher, with almost perfect scores, in hybrid methods. The worst performing of all, few-shot method, shows the least deviation.

In **BIT3**-*ins* we have 17 intents reserved for OOD, and ID splits have $17, 33, 51$ intents. Most of the intents have less utterances than enough to fill few-shot prompts. In the smaller splits few-shot

		CLINC150			DSTC Finance			BIT3-*ins*		
Method	**Metric**	**33%**	**66%**	**100%**	**33%**	**66%**	**100%**	**33%**	**66%**	**100%**
Zero-Shot	OOD Recall	**87.4**	82.1	78.7	**98.1**	94.6	92.3	**89.9**	82.5	81.9
	Binary F1	**87.6**	81.7	78.6	**98.1**	90.7	84.0	**85.4**	75.1	72.1
Few-Shot	OOD Recall	**91.0**	86.8	78.1	**96.1**	90.8	92.7	**87.9**	80.0	63.7
	Binary F1	**93.8**	90.6	87.1	**95.2**	95.1	95.1	**92.7**	88.2	81.3
Balanced Prompt	OOD Recall	**97.9**	96.6	94.9	**98.8**	96.6	96.6	**88.1**	79.1	82.3
	Binary F1	**97.4**	95.0	93.3	**99.1**	96.7	94.9	**90.7**	85.8	82.1
Contrastive Prompt	OOD Recall	**96.8**	95.7	94.0	**99.6**	94.7	93.8	90.2	91.0	**93.4**
	Binary F1	**96.2**	93.9	91.7	**99.4**	95.9	93.3	**89.4**	87.3	79.6

Table 4: Effect of number of intents on GPT-4o across three datasets. The highest score per each split in each dataset is bolded. All results are average of 3 runs.

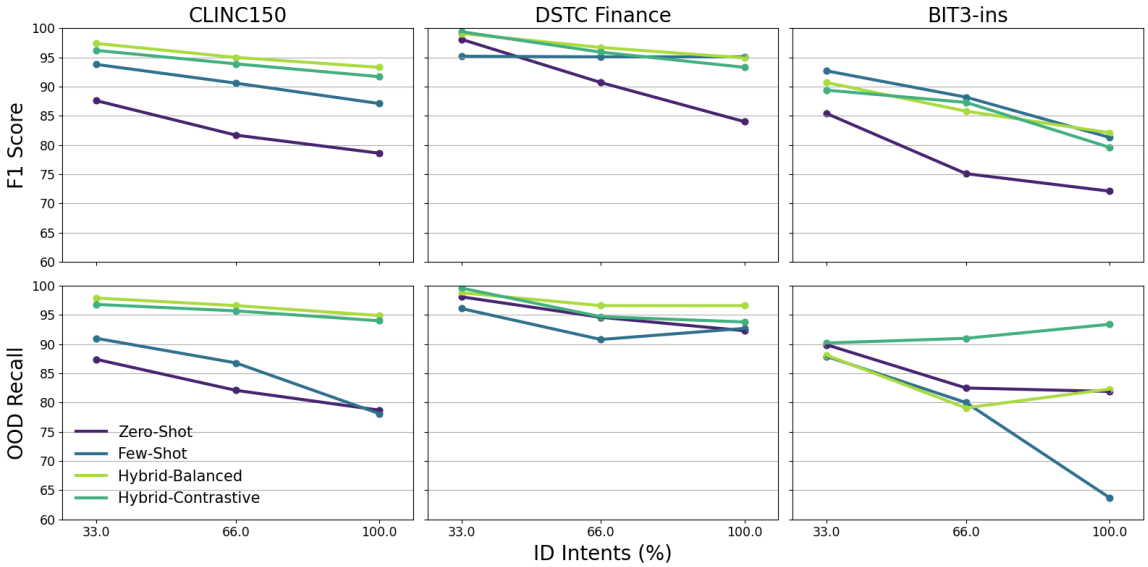

Figure 4: Performance of different methods with increasing number of intents in terms of F1 Score and OOD recall.

performs the best, however, in 100%, hybrid balanced method takes the lead. In all the methods, drop in F1 score is approximately ∼10 points. This may indicate the effect of having extremely few examples per intent.

5 Discussion

OOD detection is a challenging task that is critical for the practicality and safety of AI systems. Our study sheds light on the current state of one of the frontier models, GPT-4o, which performs significantly better than its older successors like GPT-4 and GPT-3.5, as reported by Wang et al. (2024). Their experiments on BANKING77 using GPT 3.5 with zero-shot prompting show 33.8 OOD F1, whereas our GPT-4o experiments score 71.6 with the same combination, as can be seen in Table 2. There are prompt details and split differences that can change the result; however, approximately ∼40 points improvement indicates the model gets better at an impressive pace. In CLINC150 a similar ∼35 point improvement is seen in Table 2.

In few-shot prompting, Wang et al. (2024) reports the performance of GPT 3.5 with different numbers of intents and different numbers of utterances per intent. They demonstrate a significant performance drop in OOD F1. From 5 intents to 40 intents, GPT 3.5 suffers a loss of approximately ∼50 points. In Table 4 we see a less drop through all datasets. In CLINC150, our few-shot prompt (with 10 examples) suffers approximately ∼6 points from 37 intents to 112 intents. This shows that the current frontier models are significantly more robust in this aspect.

We also believe the performance of OOD detection strongly depends on the dataset quality. If any intent has a misleading label name, has noisy examples, or it has utterance examples that can

be used in another intent as well, the results get affected significantly. We suggest that the low performance in BANKING77 in comparison to other public datasets is due to these kinds of effects, as reported by Ying and Thomas (2022).

In **BIT3** datasets, the results in Table 2 show a greater challenge than public data. The main reasons are the differences in the nature of the datasets and the imbalanced number of utterance examples for each intent. The imbalance affects few-shot prompting, and the hybrid method the most. The ID classifier in the hybrid method is dependent on our baseline method, which is more sensitive to such imbalances.

We also note, Wang et al. (2024) shows that the longer a prompt takes, the less consistent the output labels become. This may result from forgetting the middle part of the prompt, as suggested by Liu et al. (2024b). This limits the prompt length, the domain information we can inject, and the number of few-shot examples. However, by using a transformer-based classifier as a front end, we avoid prompt length issues in our hybrid systems and achieve higher performance with lower cost and latency.

Liu et al. (2024a) shows that OOD detection success improves with the scale of LLMs. Our results in Table 3 support this claim. However, the pre-training and supervised fine-tuning steps also play a substantial role in task success. To factor these out, detailed comparison of same-family models needs to be done. In addition, our results demonstrate that using a hybrid method may remove this disadvantage.

6 Conclusion and Future Work

In this paper, we extensively study the current state of LLMs in one of the challenging tasks of NLU, the near-OOD detection task. Using 10 methods across 3 public and 3 in-house datasets, we evaluate GPT-4o and compare it with Gemini 1.5 Flash and Llama 3.1-70b. We study how the increasing number of intents affects the performance of GPT-4o. We introduce a novel hybrid method that is robust, high-performing, easy to use, that enables the usage of smaller or open-source models without sacrificing performance.

Despite broad coverage, several research directions remain:

Cost and Latency. By combining cost and latency into a metric we may find the most beneficial strategy to run large scale intent detection with OOD detection support.

Alternative Hybrid Methods. The hybrid method provided here focuses on combining the ID strength of transformer models with the generalization capability of LLMs. It is cost-efficient as it doesn't require every intent and utterance to be fed to LLMs; however, we still send each intent even though the baseline method is extremely confident. Using a post hoc method and using LLMs only when the method is unsure may provide further efficiency and speed.

Detailed LLM Comparison. To clearly see the comparison of different providers and open-source models, we need thorough experimentation. The frontier models, the newly emerging reasoning models, and small language models are all rich future directions.

Multi-Label Intent Detection. Intent detection systems often do a simplification that is rather unrealistic of human communication. It is to expect each utterance to have a single intent. In reality, communication often involves more intricate information having multiple levels of intentions. One may ask a couple of things at once, or ask a more abstract concept with non-trivial bounds, or ask something that can be understood in different levels of detail. The LLMs open the door for such advanced NLU cases.

Limitations

First of all, we do not include thorough LLM comparisons. Comparison of different sizes of the same family of models, comparison of the largest models of all providers, comparison of generative models and reasoning models are required to come to a conclusion about the state of the LLMs.

In the hybrid method as a front-end, we rely on a simple yet effective sentence transformer that is known to be $\sim$ 1-2 points below the SOTA in the public datasets we use. The change may be caused by the performance gap should be studied.

The hybrid method contrastive prompt uses slightly more utterance examples than other prompts as it includes the second and the third prediction examples. We omit its further investigations due to page limitations.

It is possible that few-shot example size and intent number are correlated. Their extensive study is required. In this paper, we only have insights on these.

References

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*, pages 1559–1570, Miami, Florida, US. Association for Computational Linguistics.

Udit Arora, William Huang, and He He. 2021. Types of Out-of-Distribution Texts and How to Detect Them. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10687–10701, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mateusz Baran, Joanna Baran, Mateusz Wójcik, Maciej Zięba, and Adam Gonczarek. 2023. Classical Out-of-Distribution Detection Methods Benchmark in Text Classification Tasks. *Preprint*, arXiv:2307.07002.

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. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Gemini Team Google. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Aaron Grattafiori. 2024. The Llama 3 Herd of Models. *Preprint*, arXiv:2407.21783.

James Gung, Emily Moeng, Wesley Rose, Arshit Gupta, Yi Zhang, and Saab Mansour. 2023a. NatCS: Eliciting Natural Customer Support Dialogues. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9652–9677, Toronto, Canada. Association for Computational Linguistics.

James Gung, Raphael Shu, Emily Moeng, Wesley Rose, Salvatore Romeo, Arshit Gupta, Yassine Benajiba, Saab Mansour, and Yi Zhang. 2023b. Intent Induction from Conversations for Task-Oriented Dialogue Track at DSTC 11. In *Proceedings of the Eleventh Dialog System Technology Challenge*, pages 242–259, Prague, Czech Republic. Association for Computational Linguistics.

Dan Hendrycks and Kevin Gimpel. 2018. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. *Preprint*, arXiv:1610.02136.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained Transformers Improve Out-of-Distribution Robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751, Online. Association for Computational Linguistics.

Rui Huang, Andrew Geng, and Yixuan Li. 2021. On the Importance of Gradients for Detecting Distributional Shifts in the Wild. *Preprint*, arXiv:2110.00218.

Jaeyoung Kim, Kyuheon Jung, Dongbin Na, Sion Jang, Eunbin Park, and Sungchul Choi. 2023. Pseudo Outlier Exposure for Out-of-Distribution Detection using Pretrained Transformers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1469–1482, Toronto, Canada. Association for Computational Linguistics.

Hao Lang, Yinhe Zheng, Yixuan Li, Jian Sun, Fei Huang, and Yongbin Li. 2023. A Survey on Out-of-Distribution Detection in NLP. *Preprint*, arXiv:2305.03236.

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.

Zhijian Li, Stefan Larson, and Kevin Leach. 2024. Generating Hard-Negative Out-of-Scope Data with ChatGPT for Intent Classification. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7634–7646, Torino, Italia. ELRA and ICCL.

Bo Liu, Li-Ming Zhan, Zexin Lu, Yujie Feng, Lei Xue, and Xiao-Ming Wu. 2024a. 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 2024)*, pages 8211–8222, Torino, Italia. ELRA and ICCL.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024b. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. 2020. Energy-based Out-of-Distribution Detection. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Yutao Mou, Pei Wang, Keqing He, Yanan Wu, Jingang Wang, Wei Wu, and Weiran Xu. 2022. UniNL: Aligning Representation Learning with Scoring Function for OOD Detection via Unified Neighborhood Learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7317–7325, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

OpenAI. 2023. New and improved embedding model. https://openai.com/index/new-and-improved-embedding-model/.

OpenAI. 2024a. Hello GPT-4o. OpenAI Blog.

OpenAI. 2024b. OpenAI o1 System Card. *Preprint*, arXiv:2412.16720.

Alexander Podolskiy, Dmitry Lipin, Andrey Bout, Ekaterina Artemova, and Irina Piontkovskaya. 2022. Revisiting Mahalanobis Distance for Transformer-Based Out-of-Domain Detection. *Preprint*, arXiv:2101.03778.

Cheng Qian, Haode Qi, Gengyu Wang, Ladislav Kunc, and Saloni Potdar. 2022. Distinguish Sense from Nonsense: Out-of-Scope Detection for Virtual Assistants. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 502–511, Abu Dhabi, UAE. Association for Computational Linguistics.

Mrinal Rawat, Ramya Hebbalaguppe, and Lovekesh Vig. 2021. PnPOOD : Out-Of-Distribution Detection for Text Classification via Plug andPlay Data Augmentation. *Preprint*, arXiv:2111.00506.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. 2022. Out-of-Distribution Detection with Deep Nearest Neighbors. *Preprint*, arXiv:2204.06507.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient Few-Shot Learning Without Prompts. *Preprint*, arXiv:2209.11055.

Rheeya Uppaal, Junjie Hu, and Yixuan Li. 2023. Is Fine-tuning Needed? Pre-trained Language Models Are Near Perfect for Out-of-Domain Detection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12813–12832, Toronto, Canada. Association for Computational Linguistics.

Pei Wang, Keqing He, Yutao Mou, Xiaoshuai Song, Yanan Wu, Jingang Wang, Yunsen Xian, Xunliang Cai, and Weiran Xu. 2023. APP: Adaptive Prototypical Pseudo-Labeling for Few-shot OOD Detection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3926–3939, Singapore. Association for Computational Linguistics.

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 2024)*, pages 2354–2364, Torino, Italia. ELRA and ICCL.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. 2024. Generalized Out-of-Distribution Detection: A Survey. *Preprint*, arXiv:2110.11334.

Cecilia Ying and Stephen Thomas. 2022. Label Errors in BANKING77. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 139–143, Dublin, Ireland. Association for Computational Linguistics.

Hossam Zawbaa, Wael Rashwan, Sourav Dutta, and Haytham Assem. 2024. Improved Out-of-Scope Intent Classification with Dual Encoding and Threshold-based Re-Classification. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8708–8718, Torino, Italia. ELRA and ICCL.

Zhiyuan Zeng, Keqing He, Yuanmeng Yan, Zijun Liu, Yanan Wu, Hong Xu, Huixing Jiang, and Weiran Xu. 2021. Modeling Discriminative Representations for Out-of-Domain Detection with Supervised Contrastive Learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 870–878, Online. Association for Computational Linguistics.

Li-Ming Zhan, Haowen Liang, Bo Liu, Lu Fan, Xiao-Ming Wu, and Albert Y.S. Lam. 2021. Out-of-Scope Intent Detection with Self-Supervision and Discriminative Training. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3521–3532, Online. Association for Computational Linguistics.

Jianguo Zhang, Kazuma Hashimoto, Yao Wan, Zhiwei Liu, Ye Liu, Caiming Xiong, and Philip Yu. 2022. Are Pre-trained Transformers Robust in Intent Classification? A Missing Ingredient in Evaluation of Out-of-Scope Intent Detection. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 12–20, Dublin, Ireland. Association for Computational Linguistics.

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. 2024. A Survey of Large Language Models. *Preprint*, arXiv:2303.18223.

Wenxuan Zhou, Fangyu Liu, and Muhao Chen. 2021. Contrastive Out-of-Distribution Detection for Pre-trained Transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1100–1111, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Appendix

A.1 Training Details for Threshold-Based Baseline

The classification head is a four-layer structure starting with an input layer (embedding size 768), followed by two hidden layers with 512 and 64 neurons respectively, and concluding with an output layer matching the number of classes. We use ReLU activations for hidden layers, and cross-entropy loss for optimization. Training was performed over 100 epochs with a learning rate of 0.001, while keeping the sentence-transformer backbone frozen.

A.2 Tool Calling

We define tools as

```
tool = {
"name": name,
"description": desc_limited,
...
}
```

where

```
desc_limited = f"""Below are some example utterances for when to call this tool:

Utterances:
- <utterance1>
- <utterance2>
...
    """
```

In zero-shot setting we put empty string to description field.

A.3 Zero-Shot Prompt

```
Classify given utterances into pre-defined intents,
using the "fallback" intent for out of scope samples.
You will receive a list of intents, each with a name but no description,
and an additional intent called "fallback".
If an utterance aligns with one of the predefined intents,
classify it under that intent. Otherwise, classify it as "fallback".

# Steps
1. **Identify Intents**: Receive a list of intent names without descriptions,
including an additional "fallback" intent.
2. **Analyze Utterance**: Carefully analyze the provided utterance
to determine its intent.
3. **Classify Utterance**:
   - If the utterance matches any given intent,
   classify it under that specific intent.
   - If it does not match any of the provided intents,
   classify it as "fallback".

Remember that you need to choose intents via tool calling only.
```

A.4 Few-Shot Prompt

```
Classify user utterances based on a set of predefined tools
and identify any out-of-scope requests, assigning them
to a fallback tool if necessary.

You will be provided with a list of tools,
each defined by a name and a description
containing sample utterances. Your task is to match user utterances
to the appropriate tool name, evaluating their relevance and context.

# Steps
1. **Analyze User Input:**
Carefully read and understand the given user utterance.
2. **Evaluate Tool Descriptions:**
   - Review the names and descriptions of all provided tools,
   including sample utterances.
   - Compare the user utterance to the samples and context provided.
3. **Determine Intent:**
   - Identify if the user utterance corresponds closely with any tool
   based on the descriptions.
   - If the utterance does not clearly match any tool,
   assign it to the "fallback" tool.
4. **Output the Appropriate Tool Name:**
   - Output the name of the identified tool
   or "fallback" if the intent is out-of-scope.

# Output Format
Output the result as a simple text line containing
only the name of the matched tool or
"fallback" if no match is found.

# Example
tools = ["credit_card_cancellation", "billing_payment",
"lost_or_stolen_card", "i_am_hungry", "fallback"]

**Input**: "I want to pay my bills"
You need to call "billing_payment" tool.

**Input**: "I want to cancel my credit card."
You need to call "credit_card_cancellation" tool.

**Input**: "I want to cancel my account."
You need to call "fallback" tool.

**Input**: "I want to cancel my miles&fly card."
You need to call "fallback" tool.

**Input**: "I lost my card, so I want to deactivate it"
You need to call "lost_or_stolen_card" tool.

**Input**: "I want to learn my credit card balance."
You need to call "fallback" tool.

**Input**: "I want to change my billing address."
You need to call "fallback" tool.

**Input**: "I am starving. Lets eat something."
You need to call "i_am_hungry" tool.

**Input**: "Do you want to have a drink tonight?"
You need to call "fallback" tool.

# Notes
- Be precise in intent matching and ensure the fallback is used only
when no other tool is applicable.
- Handle ambiguities in user input by leveraging
the breadth of tool descriptions and samples.
- Continuously refer back to the tool descriptions
to ensure accuracy in intent classification.
```

A.5 Hybrid Method

A.5.1 Balanced Prompt

```
Decide if a new utterance belongs to a given intent. You will be provided with
an intent name and sample utterances for that intent,
followed by a new utterance.
Determine if the new utterance corresponds to the intent.
- Analyze the provided intent name
and sample utterances to understand the common thematic elements or purpose.
- Compare the new utterance with the sample utterances
to decide if it conveys the same intent.
- Consider variations in phrasing and terminology when deciding.
# Steps
1. **Understand the Intent:** Review the intent name and
sample utterances provided to capture the core idea or action they represent.
2. **Analyze the New Utterance:** Examine the new utterance,
```

identifying key elements and themes.
3. **Compare and Decide:** Cross-reference the analysis of
the new utterance with the intent description
and sample utterances to check for alignment.
4. **Conclusion:** Decide if the new utterance matches the intent.
If yes, output "yes"; if not, output "no".
Output Format
- A single word response: "yes" or "no".
Examples
Example 1
- **Intent Name:** "Order Food"
- **Sample Utterances:** "I'd like to place an order for a pizza.",
"Can I get a delivery for sushi?", "I want to order dinner."
- **New Utterance:** "I'd like to schedule a meal."
- **Output:** yes
Example 2
- **Intent Name:** "Order Food"
- **Sample Utterances:** "I'd like to place an order for a pizza.",
"Can I get a delivery for sushi?", "I want to order dinner."
- **New Utterance:** "What's the weather like today?"
- **Output:** no
Notes
- Be mindful of synonyms and different phrases conveying the same intent.
- Consider the context and overarching theme rather than specific word matches.

A.5.2 OOD-Focused Prompt

You will be provided with an intent name and a set of sample utterances
associated with that intent.
Additionally, a new utterance will be given.
Your task is to determine whether the new utterance
aligns with the defined intent
based on the provided samples. Provide a thorough analysis to ensure
the decision accounts for subtle differences,
recognizing when a new utterance closely resembles
but does not fully match the intent.
Steps
1. **Review the Intent Name**:
Understand the overall purpose
and category defined by the intent name.
2. **Analyze Sample Utterances**:
Examine the given sample utterances
to identify common themes, language patterns, and key details
that characterize the defined intent.
3. **Evaluate New Utterance**:
Compare the new utterance against
the identified characteristics of the intent's sample utterances.
- Consider synonyms, language variations,
and context to determine similarity.
- Pay close attention to subtle differences
that may indicate a different intent.
4. **Decision Making**:
- If the new utterance matches the intent characteristics,
conclude with “yes.”
- If it does not match, conclude with “no.”
Output Format
- A single word response: 'yes' or 'no'.
Examples
Example 1
Input:
- Intent Name: BookFlight
- Sample Utterances:
["I want to book a flight", "Can you help me reserve a ticket?",
"Find me a flight ticket"]
- New Utterance: "I need to book a cab"
Reasoning:
Review the sample utterances for intent "BookFlight."
They all involve reserving or booking flight tickets.
The new utterance refers to booking a cab,
which is different from booking a flight despite structural similarities.
Output:- "no"
Example 2
Input:
- Intent Name: BookFlight
- Sample Utterances:
["I want to book a flight", "Can you help me reserve a ticket?",
"Find me a flight ticket"]
- New Utterance: "Can you book me a flight?"
Reasoning:
The new utterance closely aligns with the intent to book a flight,
sharing both context and action with the sample utterances.
Output:- "yes"
Notes
- Pay attention to context shifts, even when phrasing similarities exist.
- Carefully consider synonyms and phrasing that might
subtly change the intent."""

A.5.3 Contrastive Prompt

Based on the given intent, sample in-scope utterances,
and out-of-scope utterances, classify whether
a new utterance belongs to the intent.
Output must strictly be 'yes' or 'no'.
- You will receive the intent name, a set of sample utterances
that fall under this intent, and another set that do not.
- Carefully evaluate both sets of utterances to determine
the defining characteristics and scope of the intent.
- Analyze the new utterance in this context.
- Decide if the new utterance appropriately falls under
the specified intent.,
Steps,1.
Understand the Intent Scope:
- Review the name and characteristics of the intent.
- Examine in-scope utterances to understand
typical expressions that belong to the intent.
- Analyze out-of-scope utterances to understand
boundaries and what differentiates them from in-scope utterances.
2. **Evaluate the New Utterance**:
- Compare the new utterance against
the characteristics derived in the previous step.
- Determine whether it aligns more closely with
the in-scope or out-of-scope examples.
3. **Classify the New Utterance**:
- If it aligns with in-scope, conclude with 'yes'.
- If it matches out-of-scope features, conclude with 'no'.,
Output Format,
- The output must be a single word: either 'yes' or 'no'.
No other text, explanations, or alterations are permitted.
Examples
Example 1:
- **Input**:
- Intent: Credit Card Cancellation
- In-Scope Utterances: "I want to cancel my credit card",
"How do I close my credit card account?"
- Out-of-Scope Utterances: "What is my credit card balance?",
"I need to change my credit card limit."
- New Utterance: "Please terminate my credit card",
Output: yes,
Example 2:
- **Input**:
- Intent: Travel Request
- In-Scope Utterances: "I need to book a flight for my trip",
"Can you arrange transportation for my travel?"
- Out-of-Scope Utterances: "What are the hotel prices in New York?",
"I need to know my car rent balance."
- New Utterance: "When is the best time to visit Paris?",
Output: no,
Notes,
- Remember that it's crucial to understand the nuanced differences
between intents, especially when they are closely related.
- Avoid assumptions beyond the given utterances;
strictly adhere to the linguistic patterns and examples provided.