IDALC: A Semi-Supervised Framework for Intent Detection and Active Learning based Correction

Ankan Mullick, Sukannya Purkayastha, Saransh Sharma, Pawan Goyal, and Niloy Ganguly, *Senior Member, IEEE*

***Abstract*—Voice-controlled dialog systems have become immensely popular due to their ability to perform a wide range of actions in response to diverse user queries. These agents possess a predefined set of skills or intents to fulfill specific user tasks. But every system has its own limitations. There are instances where even for known intents, if any model exhibits low confidence, it results in rejection of utterances that necessitate manual annotation. Additionally, as time progresses, there may be a need to re-train these agents with new intents from the system-rejected queries to carry out additional tasks. Labeling all these emerging intents and rejected utterances over time is impractical, thus calling for an efficient mechanism to reduce annotation costs. In this paper, we introduce IDALC (Intent Detection and Active Learning based Correction), a semi-supervised framework designed to detect user intents and rectify system-rejected utterances while minimizing the need for human annotation. Empirical findings on various benchmark datasets demonstrate that our system surpasses baseline methods, achieving a ∼5-10% higher accuracy and a ∼4-8% improvement in macro-F1. Remarkably, we maintain the overall annotation cost at just ∼6-10% of the unlabelled data available to the system. The overall framework of IDALC is shown in Fig. 1.**

***Impact Statement*—Conversational AI systems are widely used across industries, yet they often reject user inputs due to low confidence in intent recognition. This results in poor user experience and high manual annotation costs. Our framework, IDALC, offers a practical alternative by combining intent detection with active learning to automatically correct low-confidence predictions. It reduces labeling effort by over 90% while maintaining high accuracy across languages and domains. Unlike large language models, which are resource-intensive and difficult to deploy in real-time or on edge devices, IDALC is lightweight, efficient, and more adaptable to evolving user needs. It performs comparably, or even better, than LLMs for this task, without the heavy computational burden. By making intent detection more scalable and accessible, IDALC has strong potential to support voice-driven services in education, healthcare, and public-facing digital assistants where speed, cost, and reliability matter most.**

***Index Terms*—Active Learning based Intent Correction, Intent Detection, Intent Rejection Handling,**

I. Introduction

Over the past few years, task-oriented dialog systems have become ubiquitous. The use of these agents to perform tasks based on voice commands has opened up a new dimension in the field of Natural Language Understanding (NLU). These agents are characterized by various skills to perform various tasks based on a user's query. These predefined sets of skills are known as intents. For eg., the query **'add Michael Wittig music to country icon playlist'** requires to do the task of adding 'Michael Wittig music' to the 'country icon playlist' and it would be associated with the *AddToPlaylist* intent category (SNIPS Dataset). The task-oriented systems have some initial set of known intents that can easily be identified and responded to accordingly, but when users interact with voice-controlled systems or chatbots, there are instances where the system fails to comprehend the user's query or classify it wrongly into a known intent. The system even struggles to classify complex utterances from the known set of intents due to various reasons such as ambiguity, uncommon user requests, or untrained intents. These hard-instanced queries eventually get rejected because of lower confidence. Effective handling of system-rejected search queries is a critical challenge. To improve the user experience and enhance the system's performance, resolving this issue by understanding novel intents and detecting known intents with higher accuracy is essential.

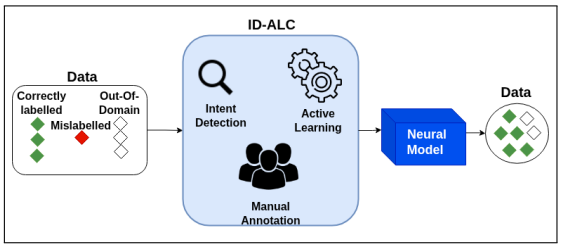

Fig. 1: Overview of IDALC Framework: The system processes user utterances, identifies known intents with high confidence, and flags low-confidence or unknown ones for correction

To address the challenge of handling system-rejected queries, several requirements must be met. First, a comprehensive evaluation of multiple datasets needs to be set up to analyze the problem of rejected queries and the necessity for manual annotation. Moreover, the development of a robust intent detection algorithm plays a crucial role in accurately classifying queries into their respective intents. So, with the emerging new intents in the system, the conversational agents need to be retrained to identify these newer intents aka *novel classes* without compromising known intent accuracy. However, it is infeasible to label all the new instances as well as the rejected utterances because of the sheer volume, and hence an effective mechanism to reduce human annotation cost would

Ankan Mullick, Pawan Goyal, and Niloy Ganguly are with the Department of Computer Science and Engineering, IIT Kharagpur, India (e-mail: ankanm@kgpian.iitkgp.ac.in).

Sukannya Purkayastha is with the Department of Computer Science, Technische Universität Darmstadt, Germany. Work done while at IIT Kharagpur.

Saransh Sharma is with Adobe Research, Bangalore, India (e-mail: sarsharma@adobe.com).

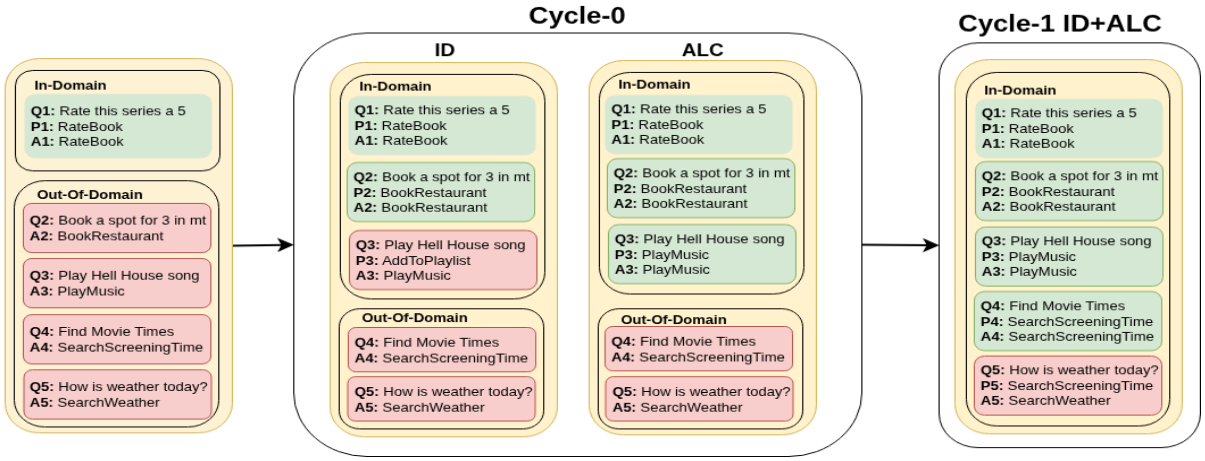

Fig. 2: End-to-end framework of IDALC (Intent Detection and Active Learning-based Correction) is illustrated using predicted (**P1–P5**) and actual intent labels (**A1–A5**). Initially, only two intents, RateBook and AddToPlayList, are known. Q1 is correctly classified. Q2–Q5 have unknown intents. In Cycle 0, Q2 is correctly classified during the Intent Detection (ID) step. Q3 is incorrectly predicted as 'AddToPlayList' but belongs to a new intent: 'PlayMusic'. As the system is unsure, it rejects Q3. In the ALC step, Q3 is corrected, and PlayMusic is added to the list of known intents. In the next cycle, Q4 is correctly classified as SearchScreeningTime. A5 is added to the set of known intents, but Q5 is still misclassified. After each ID step, some previously unknown queries are mapped to known intents. After each ALC step, incorrect predictions are corrected. By the end, some queries may remain misclassified or unrecognized.

be required. So, an algorithm should be designed to handle both known and novel intents, ensuring that the system can adapt and recognize emerging user requests effectively with optimized annotation. Active learning methods can further enhance the intent detection model by selecting the most informative queries for manual annotation. Regular retraining of the model with new data, including rejected queries and annotated queries, allows the system to continually improve its performance and stay up-to-date with evolving user intents.

In this paper, we adopt a two-step semi-supervised setting to identify known as well as novel out-of-domain (OOD) intents using an active learning framework. We also experiment with different scenarios of similarities among known and novel intents. Corresponding to our problem setting, we start with an initial labeled training data which consists of a set of known intents and a large unlabeled data that comprises known and novel intents. The evaluation is performed on a predefined test set that consists of all the known and new classes. Our framework consists of two architectures running sequentially - Intent Detection (ID) and Active Learning based Correction (ALC). Fig. 1 gives a high-level view of the IDALC framework. It shows how the system processes user queries by first identifying known intents with high confidence. If the system is unsure or encounters a new intent, it flags the query for correction through active learning.

Fig. 2 provides a detailed step-by-step example of how this process works. At the start, the system only knows two intents, RateBook and AddToPlayList. It correctly identifies the intent of the first query (Q1), but the intents for Q2, Q3, and Q4 are unknown. In the first cycle (Cycle 0), the system processes Q2 and Q3. It gets Q2 right but misclassifies Q3 as AddToPlayList, a known intent, even though the correct label is PlayMusic. This mistake is caught and corrected in the Active Learning phase. In the next cycle (Cycle 1), the system has now learned the intent behind Q4 and can correctly classify it.

Our system is compared with similar models and evaluated across several NLU standard public datasets. Our framework is language agnostic and performs significantly well on different multilingual (English, Spanish, and Thai) datasets. IDALC outperforms the existing baselines by a margin of $\sim$ 4-8% macro-F1 and $\sim$ 5-10% in terms of accuracy on various datasets. We also show that our model is able to achieve better outcomes than large language models like ChatGPT, and LLaMA-2. The annotation cost incurred by our model varies from 6-10% depending on the size of the unlabeled dataset.

II. Related Work

In the last decade, researchers have focused on intent detection in various scenarios. People also explore various active learning methods to get optimum accuracy with minimum labeling cost. We review related work on intent class detection as well as active learning-based strategies to improve classification and lower human effort.

A. Intent Class Detection

In dynamic and evolutionary real-world applications, various flexible adaptive intent class detection models are developed. [1, 2, 3, 4] work on incremental learning in a dynamic environment for evolving new classes in streams. **(author?)** [5] aims at the class evolution phenomenon of the emergence and disappearance of classes. Some advanced approaches are proposed to explore novel class detection problems in data streams. ECSMiner [6] resolve streaming problems but the time delay assumption and fixed-size chunks are unrealistic in real scenarios. SAND [7] is a semi-supervised framework to identify concept drifts. **(author?)** [8] identifies novel intents

but does not apply any approach to retain the accuracy of known intents and does not handle system-rejected utterances. In different specialized contexts, researchers also work on intent phrase identification task [9, 10, 11, 12, 13, 14, 15, 16], various entity-extraction [17, 18, 19, 20], opinion context detection [21, 22, 23, 24, 25, 26] and others [27, 28]. Based on the Frequent-Directions technique [29], SENC-MaS [30] is a matrix sketch method to detect new classes and update the classifier. SENCForest [31] utilizes random trees as a single common core to identify new classes. But both these approaches rely on a large labelled initial training set and do not use unlabelled data. [32, 33, 34] explore outlier detection approaches but most of them are applicable on images. [35, 36, 37, 38, 39, 40] detect new class in the form of outlier detection. However, these algorithms fail to identify new classes from rejected utterances and classify all. SEEN [41] uses both labeled and unlabelled data to detect emerging new classes and classify known classes. Zero-shot-OOD [42] proposes a new method to tackle the OOD detection task in low-resource settings, without adversely affecting performance on the few-shot ID classification. Aligner [43], CEA-Net [44] explore joint multiple intent detection with slot filling task. **(author?)** [45] apply the joint-BERT model for domain-specific intent classification. SENNE [46] and IFSTC [47] handle streaming class classification problems. TARS [48] identifies in-domain data and classifies them. We use SEEN, SENC-MaS, Zero-shot-OOD, SENNE, IFSTC, and TARS as baselines.

B. Active Learning based Correction

Active learning (AL) is a machine learning paradigm that aims to reduce the annotation effort by selectively choosing the most informative data points to label. Instead of randomly labeling large volumes of data, AL focuses on identifying uncertain or representative instances that are likely to improve the model's performance when labeled. This is particularly useful in tasks like intent detection, where obtaining labeled data can be expensive and time-consuming, especially when dealing with large or diverse datasets. **(author?)** [49] proposes an active learning strategy, based on label error statistical methods to find critical instances. **(author?)** [50] uses active learning for heterogeneous data. **(author?)** [51, 52] apply active learning (AL) method for intent classification. **(author?)** [53] is a cluster-based active learning method works on data streams. But none of these works focused on accuracy improvements of intent classes using AL approaches. **(author?)** [54] presents a consistency-based semi-supervised AL framework and a simple pool-based AL selection metric to select data for labeling. But these models do not handle system-rejected samples and are useful for images, not in textual contexts. **(author?)** [55] proposes various AL strategies to improve system accuracy. **(author?)** [56] develops a maximal marginal uncertainty-based AL model to maximize accuracy with minimal interactions. We use the last two approaches as baselines (MaxMU-AL and MinU-AL).

None of the earlier works focus on developing a single system to utilize labelled and unlabelled data together to detect unknown intents, improve known intent identification with optimized human annotation cost. Our approach can improve intent detection accuracy while handling system-rejected utterances and also lower the human labelling task.

III. Dataset

We perform our experiments on three NLU benchmarked datasets - SNIPS [57], ATIS [58] and Facebook Multi-lingual data [59].

(A) SNIPS: This dataset is collected from the SNIPS personal voice assistant. It has a total of 14,484 sentences with 7 intent types. We perform our experiments considering 5 intents as known and the rest 2 as unknown / novel. The split of the dataset is shown in Table I. We use acronyms such as **BR** (BookRestaurant), **AP** (AddToPlaylist), **GW** (GetWeather), **RB** (RateBook), **SSE** (SearchScreeningEvent), **SCW** (SearchCreativeWork) and **PM** (PlayMusic) to represent the intents. To understand the effect of choosing various types of novel intents and how the system performs in different scenarios, we perform 3 different types of division as follows:

(i) **Type 1** (*Unknowns are dissimilar from each other but similar to the known intents*): For this experiment, we consider 'SCW' and 'PM' as unknowns which resemble the known intents, 'SSE' and 'AP', respectively. (ii) **Type 2** (*Unknowns are dissimilar from each other and also from the known intents*): In this case, we consider 'GW' and 'RB' as unknown intents which are distinct classes and bear no resemblance to any other classes. (iii) **Type 3** (*Unknowns are similar to each other but dissimilar from the known intents*): The intents 'SSE' and 'SCW' are considered as unknown intents here which are similar to each other but have no similarity with any other known intents.

(B) Facebook-multilingual Dataset: The dataset consists of data from three different languages: English (FB-EN), Spanish (FB-ES), and Thai (FB-TH) which consist of 43,323, 8,643, and 5,083 utterances, respectively. The utterances have been collected from three domains: Alarm, Reminder, and Weather with 12 different intents. We consider 9 intents as known and the rest 3 as unknown.

(C) ATIS Dataset: Airline Travel Information System (ATIS) consists of audio recordings of people making flight reservations. It consists of 5,871 user queries with 17 different intent types. We consider 10 intents as known and 7 as unknown.

The detailed statistics of these datasets, including our experimental framework are shown in Table I. We take the existing benchmarked NLU publicly available datasets, omit a few intents, mark them as OOD, and detect them. Since the datasets are fully labeled, annotation here just means using the given labels. We use them for training and hide them during prediction, so there are no issues like labeling errors or differences between annotators, and the data stays consistent.

IV. Approach

We propose the algorithm "ID-ALC" (Intent Detection and Active Learning based Correction) which is shown in Algorithm 1 with two major procedures: intent detection (ID - Algorithm 2) and active learning based correction (ALC -

Dataset	# Intent Class	Total data	Type	# Lab	#Unlab			#Test		
					#Ind	#OOD	#Tot	#Ind	#OOD	#Tot
SNIPS	7 (5 + 2)	14484	1	2990	5661	3449	9110	1679	705	2384
			2	2990	5629	3481	9110	1709	675	2384
			3	2990	5689	3421	9110	1692	692	2384
FB	12 (9 + 3)	43323	EN	10000	9978	15022	25000	3849	4474	8323
		8643	ES	2000	2321	1678	3999	1512	1132	2644
		5083	TH	1000	1203	1797	3000	318	765	1083
ATIS	17 (10 + 7)	5871	-	1501	2688	682	3370	802	198	1000

TABLE I: Dataset statistics based on our custom splits for multiple intent detection benchmarks. The datasets include SNIPS, Facebook Multilingual (FB), and ATIS. Each dataset is partitioned into labeled, unlabeled, and test subsets, further broken down into in-distribution (Ind) and out-of-distribution (OOD) samples. For SNIPS, we report three different Type settings.

Algorithm 3). The ID module detects out-of-domain samples on unlabeled data, identifies new intent classes as described in Sec IV-A and the ALC, described in Sec IV-B, handles the rejected utterances while reducing the annotation cost and also ensuring higher overall accuracy on the known intents. In our framework, we keep adding data to the initial pool of labeled data at every step using the two components (running multiple iterations), we call each of these model retraining steps as *cycles*. The ID-ALC framework is shown in Fig. 3 with its two components.

Algorithm 1 Intent Detection and Active Learning based Correction (ID-ALC) Algorithm

Input
  $\mathcal{L}$ Labelled Data (**Cycle 0**)
  $\mathcal{U}$ Unlabelled Data
  $\mathcal{T}$ Blind Test Data
  $\mathcal{TH}$ A predefined Threshold
  $\mathcal{MV}$ Majority Voting Count
Paramters
  $\mathcal{M}$ Neural Model
$\mathcal{L}$, $\mathcal{U}_{rem}$, $\mathcal{M} \leftarrow$ ID($\mathcal{L}$,$\mathcal{U}$, $\mathcal{M}$, T)
$\mathcal{L}$, $\mathcal{U}_{rem}$, $\mathcal{M} \leftarrow$ ALC($\mathcal{L}$,$\mathcal{U}_{rem}$, $\mathcal{M}$, T, $\mathcal{TH}$, $\mathcal{MV}$)

Algorithm 2 Intent Detection: ID($\mathcal{L}$,$\mathcal{U}$, $\mathcal{M}$, T)

procedure INTENT DETECTION
  Train $\mathcal{M}$ on $\mathcal{L}$ and predict on T.
  Train OOD-SDA on $\mathcal{L}$ and predict on $\mathcal{U}$ to get OOD samples, $\mathcal{U}'$
  Consider $m\%$ of the samples from $\mathcal{U}'$ for manual annotation, say $\mathcal{U}'_m$
  Use various labelling strategies to label $\mathcal{U}'$ based on $\mathcal{U}'_m$
  $\mathcal{L} = \mathcal{L} \cup \mathcal{U}'$ and $\mathcal{U}_{rem} = \mathcal{U} - \mathcal{U}'$
  Train $\mathcal{M}$ on $\mathcal{L}$ and predict on T (**Cycle 1**)
end procedure

Neural Model ($\mathcal{M}$): We use the JointBERT [60] model as our learning model. The model performs intent classification. We do not use the auxiliary task of slot-filling. We use English uncased BERT-Base[1] model [61] as the base model for JointBERT. In the case of other languages, we use the bert-base-multilingual-uncased model. For the neural model, we perform a hyperparameter search and report the results

[1]We explore different models but it performs the best

Algorithm 3 Active Learning based Correction: ALC($\mathcal{L}$,$\mathcal{U}_{rem}$, $\mathcal{M}$, T, $\mathcal{TH}$, $\mathcal{MV}$)

procedure AL CORRECTION
  Apply Model $\mathcal{M}$ on $\mathcal{U}_{rem}$ to predict intent class label and respective probability score
  for Each sample in $\mathcal{U}_{rem}$ **do**
    if score $< \mathcal{TH}$ **then**
      Apply Classifiers - RF, LR, Bg, KNN and LDA
      Label the sample with Majority Voting predictions (Vote Count $\leq$MV)
      Add Sample and Label to Majority Voting Set ($\mathcal{MVS}$)
      Add Rejected Samples to $\mathcal{R}$ set
    end if
  end for
  Manually Annotate $\mathcal{R}$ set
  $\mathcal{L} = \mathcal{L} \cup \mathcal{MVS} \cup \mathcal{R}$
  $\mathcal{U}_{rem} = \mathcal{U}_{rem} - \mathcal{MVS} - \mathcal{R}$
  Retrain $\mathcal{M}$ on $\mathcal{L}$ and predict on T
  Repeat Steps 2 to 14 for K times
end procedure

of the settings that achieve the best results and then fix the same for all the models. The batch size is 16, the number of epochs is 10, the optimization algorithm used is AdamW and the learning rate is $5e-5$ with cross-entropy as the loss function.

A. Intent Detection (ID)

The algorithm (Algorithm 2: steps 1-8) for our Detection consists of the following Steps:

Step 1 *Cycle 0*: With the initial labeled data ($\mathcal{L}$) that consists of examples only from the set of known intents, we train a Neural Model ($\mathcal{M}$).

Step 2: *Out-Of-Domain Sample Detection Algorithm (OOD-SDA)*: In this step, we consider an algorithm to detect Out-Of-Domain samples $\mathcal{U}'$ on the Unlabeled Data ($\mathcal{U}$).

Step 3: *Efficient Labeling of New Intents in OOD*: In this step, we consider $m\%$ of the OOD samples $\mathcal{U}'$ for manual annotation ($\mathcal{U}'_m$). The $\mathcal{U}'$ OOD samples can be labeled with new as well as old intents. We experiment with various labeling strategies to add the rest of the OOD samples back to the Cycle 0 training set ($\mathcal{L}$).

Step 4 *Cycle 1*: We re-train $\mathcal{M}$ on the modified training set and perform prediction on the Test Set.

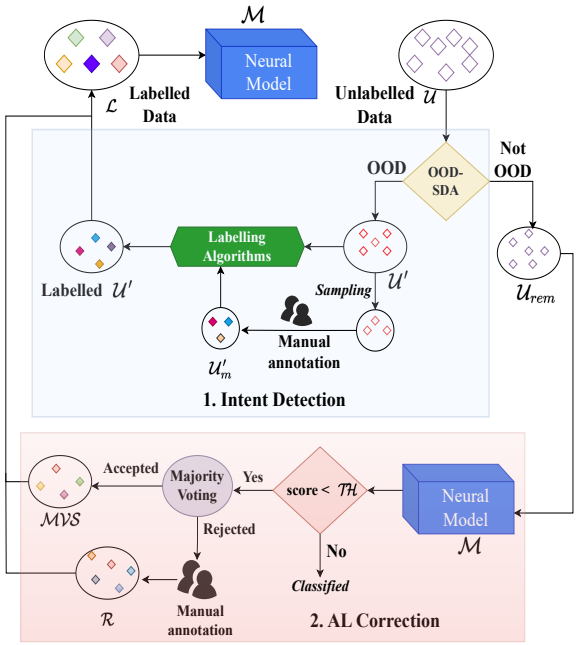

Fig. 3: End-to-end framework of IDALC:Intent Detection and Active Learning based Correction

B. Active Learning based Correction (ALC)

Next, we devise an active learning-based framework to take care of the rejected intent utterances with low confidence for rectification. As shown in Figure 3, we take the utterances from the unlabeled data and do active learning-based auto corrections of low confidence instances (i.e., the utterances with a score below a threshold after passing through the neural model $\mathcal{M}$) by majority voting. Then we manually annotate the low-confidence and auto-rejected utterances and add both to the training set. The steps (1-16) of the Algorithm 3) are as follows:

Step 1 *Cycle 1*: We use the retrained Neural Model ($\mathcal{M}$) (Cycle 1 of intent detection) to predict intent scores on the unlabeled dataset after removing OODs ($\mathcal{U}_{rem}$).

Step 2 *Cycle 1*: We set a pre-defined threshold ($\mathcal{TH}$), and the samples with score ¡ $\mathcal{TH}$ are then fed to five different classifiers and we use majority voting prediction to identify their labels (*auto-corrected*). Samples that do not find a majority are manually annotated. These auto-corrected and annotated samples are added back to the labeled data and removed from the unlabeled data.

Step 3 *Cycle 2*: We retrain the Neural Model ($\mathcal{M}$) with an updated labelled dataset and redo **Steps** 1-2 for the same threshold value.

Step 4: We run this approach for $2 \leq K \leq 5$ cycles.

V. Experimental Results

We discuss our experimental set-up detailing various algorithms and strategies utilized for both the components, competing baselines, and the experimental results. We categorize the experimental outcomes into two compartments - intent detection and active learning-based correction.

A. Intent Detection

For identification of new intents, we consider the following settings:

Out-of-Domain Sample Detection Algorithms (OOD-SDA): We consider three algorithms for detecting out-of-domain samples. i) Softmax Prediction Probability (MSP) [62], ii) Deep Open Classification (DOC) [63] method and iii) Local Outlier Factor (LOF) [64]. For MSP, we use the minimum prediction score on the training set to decide the threshold and report for the best threshold. We experiment on all English datasets as shown in Table II. DOC performs the best in terms of both accuracy (Acc) and macro-f1(F1), so we use the DOC model in our pipeline and for different experiments.

Dataset	Method	Threshold	Acc (%)	F1 (%)
SNIPS	MSP	0.7	86.72	89.27
	LOF	-	81.21	83.44
	DOC	-	**90.3**	**89.7**
ATIS	MSP	0.3	85.74	86.32
	LOF	-	78.2	78.05
	DOC	-	**88.88**	**87.19**
FB	MSP	0.7	88.63	89.11
	LOF	-	81.49	82.37
	DOC	-	**93.42**	**93.5**

TABLE II: Accuracy(Acc) and Macro-F1(F1) of various OOD-SDA on the three datasets

Labelling Strategies: Once we obtain the OOD samples from the unlabeled data using OOD-SDA, the next step is to add this data back to the training set. The $m\%$ samples chosen uniformly and randomly for manual annotation are set to 20% in our case based on various experiments. The different Labelling Strategies used are discussed below:

K-Means (KM): Here, we decide on the number of clusters based on the majority labels of the manually annotated OOD samples. The labels whose frequency is greater than a pre-decided threshold, t, are considered the majority labels. The threshold is decided as t = (no of annotated samples) / (total number of labels discovered during annotation). We now run k-means algorithm on the entire set of OOD samples on the unlabeled data set and annotate clusters as per their majority labels.

Majority Voting (MV): With the manually annotated OOD samples, we train several traditional machine learning models to compute the accuracy of the system. We check on a total of twenty different classifiers and the top nine results (accuracy for each dataset and overall average [Avg]) are shown in Table III for Random Forest (RF), Bagging (Bg), Support Vector Machine One-vs-all (SVM), Logistic Regression (LR), XGBoost (XGB), Multinomial NB Classifier (NB), Linear Discriminant Analysis (LDA), K-Nearest Neighbour (KNN),

Class	Accuracy (%)					
	SNIPS	ATIS	FB-EN	FB-ES	FB-TH	Avg
RF	95.4	89.8	91.2	90.6	88.3	91.1
Bg	97.9	87.8	92.3	89.5	88.4	91.2
SVM	87.4	85.8	86.42	83.5	80.4	84.7
LR	97.8	96.5	94.7	95.9	97.4	96.5
XGB	87.6	75.3	86.5	84.0	85.1	83.7
NB	85.1	86.9	83.6	80.0	81.6	83.4
LDA	97.8	97.02	95.4	96.7	97.6	96.9
KNN	95.1	89.6	90.5	82.3	92.4	90.0
Gau	90.2	79.1	80.2	84.0	85.7	83.9

TABLE III: 5-Fold Cross Validation Accuracy over 5 different datasets for various classifiers. The best results are **bolded**.

Method	FB-EN		FB-ES		FB-TH		ATIS		SNIPS	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
SEEN*	84.9	86.2	75.4	79.8	71.9	77.5	84.3	71.4	85.8	86.2
Zero-Shot-OOD*	86.7	87.4	62.1	70.1	68.6	72.1	84.8	81.7	87.4	88.5
SENC-MaS*	82.3	83.9	69.4	70.5	64.6	72.8	80.1	69.4	82.0	81.4
SENNE*	61.4	60.5	56.1	58.9	58.2	59.4	53.2	51.5	52.7	55.8
IFSTC	84.7	82.0	80.4	54.3	86.1	87.2	71.1	71.4	70.4	71.1
TARS	71.2	72.3	61.1	60.2	81.1	80.1	78.4	72.4	79.6	81.1
LLaMA2-FT	92.2	78.1	**89.1**	71.5	83.5	79.0	81.6	73.3	90.6	90.1
LLaMA2-FT+ALC(2)	92.7	82.4	91.3	76.2	86.3	84.1	82.5	77.5	92.6	92.9
ChatGPT 3.5T-Pr	70.5	69.7	70.4	64.5	51.7	52.1	68.8	51.9	70.8	72.2
ID (0)	61.9	72.4	66.6	74.2	68.1	80.2	82.6	81.1	73.7	80.2
ID + RA	66.7	52.4	56.6	52.3	48.1	60.7	52.6	51.9	73.7	78.7
ID (1) + KM	**95.7**	**92.3**	87.6	**94.9**	**89.3**	**88.6**	**90.3**	**86.4**	**94.7**	**93.0**
ID (1) + MV	92.2	89.6	85.2	90.3	85.3	87.23	86.2	81.6	91.9	90.5
ID (1) + CL	94.1	91.8	87.3	92.1	**89.3**	88.4	89.6	83.5	94.6	92.6
ID-ALC (2)	**98.0**	**93.1**	**91.5**	**96.1**	**92.5**	**89.5**	**92.1**	**88.9**	**97.9**	**95.5**

TABLE IV: Results in terms of Overall Accuracy (Acc) and Macro average F1-score (F1) on Facebook - English, Spanish, Thai, ATIS and SNIPS Datasets. (0), (1) and (2) denote the cycle number. All results are in (%).

Gaussian (Gau).We take the majority vote of the predictions of these classifiers on the rest of the OOD samples to label them. 5-fold cross-validation is performed on the labeled data to determine the performance of these classifiers to identify intents. Based on the accuracy scores, the top 5 (out of 9) classifiers (**Bold** in Table III) are chosen for experiments.

Cluster Then Label (CL): Here, we start with a predefined number of clusters, k. We cluster the entire set of OOD samples on the unlabeled data set using k-means. We then label those clusters based on their majority labels as per $m\%$ annotated data. If two or more clusters have the same label, we merge these clusters. We set the value of k as double the number of known intents.

*1) **Competing Baselines**:* IDALC aims to enhance task-oriented dialog systems by handling known intents, detecting unseen ones, and adapting in streaming settings with limited annotation. We use various baselines from the last few years - all are closely matching solutions to the problem definition, and compare our framework in different directions - OOD detection, zero/Few-shot approaches, etc. Since all the baselines are state-of-the-art approaches addressing related problems over the last decade, including them in our comparisons is essential to rigorously establish the superiority of our framework. For fair comparison, we evaluated baselines across three categories: (i) streaming/semi-supervised, (ii) zero-shot/OOD detection, and (iii) large language models. These categories capture the main challenges of evolving label spaces, limited data, and annotation efficiency.

Notation for Complexity: We denote n_{train} and n_{test} as the number of training and test examples, respectively, with e epochs and sequence length L (fixed at $L \leq 512$). The hidden dimension is d ($d = 4096$ for LLaMA2-7B), with ℓ transformer layers ($\ell_{\text{BERT}} = 12$, $\ell_{\text{LLaMA}} = 32$). We cap the number of clusters at $k \leq 20$. For QLoRA, the number of trainable parameters is p_{ft}; with rank-$r = 4$ LoRA adapters applied to Q, K, V, O projections across layers, this yields $p_{ft} = 8 \cdot d \cdot r \cdot \ell_{\text{LLaMA}} \approx 4.2$M. ChatGPT-3.5 API calls incur monetary cost c. Batch size is kept at 16.

A) Streaming and Semi-Supervised Methods

SEEN [41] is designed specifically for streaming intent detection, where new labels can appear continuously in the incoming data. It combines two key components: (i) *SEEN-Forest*, an ensemble of random trees that serves as an anomaly detector, and (ii) *SEEN-LP*, a graph-based label propagation module for semi-supervised classification of queries. This hybrid design makes SEEN particularly relevant for dialog systems, as it explicitly addresses the challenge of detecting novel intents with minimal human supervision. In our setup, we initialize their model with a small amount of labeled data in the initial stream (stream 0), incrementally update it with unlabeled data from the next stream (stream 1), and finally evaluate it on the held-out test set. The complexity of label propagation is $\mathcal{O}(n_{\text{train}}^2)$ in dense graphs or $\mathcal{O}(n_{\text{train}})$ when sparsified, while training decision trees costs $\mathcal{O}(n_{\text{train}} \log n_{\text{train}})$.

SENC-MaS [30] builds compact matrix sketches that summarize the distribution of known-class data and then leverages these sketches for both classification and novelty detection. By operating on compressed sketches rather than the raw data, it prioritizes efficiency, making it well-suited for high-throughput dialog streams where retraining large neural models on every update is infeasible. In our experiments, the algorithm incrementally updates its sketches with streaming unlabeled data, after which new queries are either classified or flagged as novel. The update complexity is $\mathcal{O}(n_{\text{train}}\, d)$, reflecting its linear dependence on training data and hidden dimension.

SENNE [46] is a nearest-neighbor ensemble approach tailored for emerging-class detection, especially in scenarios where new intents are closely related to existing ones. This characteristic is important for dialog systems, as subtle distinctions (e.g., *PlayMusic* vs. *AddToPlaylist*) can critically impact performance. SENNE works by constructing ensembles of nearest-neighbor classifiers for each known class and checking whether a query falls within or outside the decision boundary of these ensembles. This makes it particularly strong in handling borderline cases where unseen intents resemble existing categories. The computational cost is $\mathcal{O}(n_{\text{train}}\, d)$ at inference.

IFSTC [47] frames intent detection as a textual entailment problem, which enables few-shot classification of new intents without retraining on the entire dataset. This is especially relevant for IDALC, as both approaches aim to generalize

quickly from small amounts of annotated data. IFSTC employs a hybrid strategy: it initializes models with XNLI pretraining and then fine-tunes them using only a few labeled examples of new intents. This combination of cross-lingual pretraining and few-shot adaptation makes it robust across languages and efficient in data-scarce settings. Its fine-tuning cost is $\mathcal{O}(e\, n_{\text{train}}\, L^2\, d\, \ell_{\text{BERT}})$.

B) Zero-Shot and OOD Methods

Zero-Shot-OOD [42] is built on an OOD-resistant prototypical network, which learns compact clusters of known-class examples and rejects queries that fall outside them. The method directly benchmarks IDALC's OOD detection capabilities, since both target the challenge of handling unseen intents with limited supervision. In our experiments, prototypes were constructed using meta-learning episodes, and incoming queries were tested for membership within known clusters. Queries sufficiently distant from all prototypes were flagged as out-of-distribution. Training scales as $\mathcal{O}(e\, n_{\text{train}}\, d)$ and inference as $\mathcal{O}(k\, d)$ per query.

TARS [48] reformulates text classification as a label-text matching task, where each input query is paired with an intent label and the model decides whether they match. This approach is highly relevant for our evaluation, as it probes whether intent names alone can serve as sufficient supervision. Unlike IDALC, which uses a more elaborate loop of OOD detection, clustering, and annotation, TARS relies directly on semantic alignment between queries and labels. For our benchmarks, we trained TARS models on dialog datasets using BERT-base for English and multilingual BERT for non-English data. The training cost is $\mathcal{O}(e\, n_{\text{train}}\, L^2\, d\, \ell_{\text{BERT}})$, with inference requiring a single forward pass per label.

C) Large Language Model Baselines

LLaMA2-FT[2] fine-tunes a LLaMA2-7B model using parameter-efficient methods such as QLoRA and PEFT. While it serves as a strong benchmark in high-resource setups, it is less suited for incremental adaptation in streaming scenarios. Including this baseline highlights the trade-off between raw accuracy and computational cost. To further investigate adaptability, we also report results with IDALC's ALC module applied on top (LLaMA2-FT+ALC), demonstrating how annotation-driven clustering can complement large pre-trained models. The training complexity is $\mathcal{O}(e\, n_{\text{train}}\, L^2\, d\, \ell_{\text{LLaMA}})$ with $p_{ft} = 4.19$M trainable parameters.

ChatGPT 3.5 Turbo[3] represents in-context learning with no model retraining. Queries are answered by constructing prompts that include 100–150 training samples along with the test query, following structured instruction templates. Its relevance lies in evaluating the performance of large hosted APIs that incur zero training cost, but at the expense of latency, privacy, and reliance on external infrastructure. The local complexity is negligible, but usage cost scales with c, the monetary cost of API calls.

D) Random Annotation (RA) This baseline randomly selects samples from the unlabeled pool for annotation, equal in number to those annotated by IDALC. It provides a natural lower bound for annotation-driven methods by removing any intelligent sampling strategy. While simple, RA allows us to quantify how much of IDALC's gains are attributable to active annotation selection rather than annotation volume. The additional computational cost apart from BERT classifier training is negligible, involving only uniform random sampling.

Evaluation Note: Some baselines detect only one novel intent at a time. For fairness, we merge all novel intents into a single class and report accuracy and macro-F1 over known vs. merged classes[4]. Unlike IDALC, these methods use no manual annotation; variants with equal annotation performed worse and are not reported. IDALC alternates between neural training, OOD detection, clustering, annotation, and retraining. The dominant cost comes from BERT training, with complexity $\mathcal{O}(e\, n_{\text{train}}\, L^2\, d\, \ell_{\text{BERT}})$. The OOD component is another transformer-based classifier, which scales similarly as $\mathcal{O}(e\, n_{\text{train}}\, L^2\, d\, \ell_{\text{BERT}})$. For clustering, we apply k-means on hidden representations, costing $\mathcal{O}(n_{\text{train}}\, k\, d)$. At inference, optional nearest-neighbor assignment adds $\mathcal{O}(n_{\text{test}}\, d)$. The overall cost is $\mathcal{O}(e\, n_{\text{train}}\, L^2\, d\, \ell_{\text{BERT}} + n_{\text{train}}\, k\, d)$.

2) Results and Discussion: In Table IV, we show results in multilingual Facebook datasets (English, Spanish, and Thai), ATIS, and SNIPS (average results from Table V). In Table V, we report the performance of our algorithm on the different types of SNIPS data. We observe that in almost all the datasets, the **KM** labeling strategy outperforms others in terms of overall accuracy and Macro-F1 score. We report overall accuracy and Macro-F1 as they give a better idea of performance across both frequent and rare classes. We also observe that **CL** performs significantly well on all the datasets and even outperforms **KM** in the SNIPS Type 3 setting as in Table V. Thus, **CL** can be a decent alternative to **KM** and may significantly reduce the overhead of deciding the number of clusters. ChatGPT 3.5 Turbo with Prompt does not produce good results which shows limitations of large language models without finetuning for specific tasks. LLaMA2[5] fine-tuned with a prompting model achieves good performances for a few cases (e.g. - accuracy in Spanish without ALC) but does not produce better results than IDALC for different datasets in Table 4. This may be due to the facts like - intent data skewness, similar intent categories in task-oriented datasets (e.g. - *atis-flight* and *atis-flight-no* in ATIS), and limitation of LLaMA2 base model in case of finetuning with fewer data sample and a quantization parameter. In a few cases, 'LLaMA2-FT+ALC(2)' outperforms IDALC like in the case of SNIPS Type 2 (in Table V) where new intents are dissimilar to each other and with known intents also. In other scenarios, LLaMA2-FT produces better accuracy than other models (without ALC) in FB-Spanish (Table IV) but IDALC performs better than 'LLaMA2-FT+ALC(2)'. The ID Random annotation (RA) baseline fails to perform well on the datasets since the random selection of samples leads to a uniform distribution of all the known and unknown classes and a lower number of OOD samples are added back to training.

[2]https://ai.meta.com/llama/

[3]https://platform.openai.com/docs/models/gpt-3.5-turbo

[4]This setting favors such baselines, as they do not separate multiple novel intents.

[5]We explore LLaMA2 (LLaMA2-7b) without fine-tuning but results are not good, hence we do not report that.

Method	Type 1		Type 2		Type 3	
	Acc	F1	Acc	F1	Acc	F1
SEEN*	82.6	81.4	83.4	84.5	91.3	92.7
Zero-Shot-OOD*	85.7	83.2	86.2	89.1	90.4	93.1
SENC-MaS*	80.7	78.3	78.5	82.1	86.9	83.7
SENNE*	50.4	51.1	56.1	61.5	51.6	54.8
IFSTC	69.1	7.6	73.6	73.5	68.5	69.2
TARS	78.1	78.3	84.5	84.8	76.2	80.2
LLaMA2-FT	84.0	82.8	97.1	97.1	90.6	90.3
LLaMA2-FT+ALC(2)	86.7	87.2	**98.9**	**98.1**	92.2	93.4
ChatGPT 3.5T-Pr	73.3	76.3	79.2	79.0	59.9	61.4
ID (0)	79.4	81.6	71.1	80.6	70.8	78.3
ID (1) + RA	70.4	76.6	76.2	81.3	74.7	78.1
ID (1) + KM	96.3	94.8	95.3	93.2	92.4	91.1
ID (1) + MV	93.6	89.9	91.9	91.5	90.2	90.0
ID (1) + CL	93.1	91.1	94.9	92.5	95.9	94.2
ID-ALC (2)	**98.6**	**95.9**	98.8	95.7	**96.2**	**94.8**

TABLE V: Results in terms of Accuracy (Acc) and F1-score (F1) (in %) on test set for different types of SNIPS Data. (Number) represents Cycle no. **Type 1** (*unknown intents are similar to known intents*). **Type 2** (*unknown intents are dissimilar to each other and also known intents*). **Type 3** (*unknown intents are similar to each other but dissimilar to the known intents*)

Our method 'IDALC(KM based)' outperforms other baselines in most of the cases by $\sim$ 5-10% in terms of accuracy and $\sim$ 4-8% in terms of macro F1. Also, Table V infers that all the baselines perform well for the Type 2 setting in the SNIPS dataset. The reason behind this would be the high level of dissimilarity among the unknown intents and with the known intents which makes it easier to detect these intent classes. Despite the fact that the competing baselines have the added advantage of not having to differentiate between the new intents, IDALC still achieves higher results (in terms of both accuracy and macro-f1) in most of the cases. The improvement of IDALC (KM based) over baselines is statistically significant ($p < 0.05$) as per McNemar's Test in Table IV and V.

B. Active Learning based Correction

We use Active Learning Classifier (Majority Voting) on top of ID (using KM) and LLaMA2-FT to detect intents more accurately as shown in Tables IV and V. It helps to increase the accuracy of known intents and identify new cases. Since IDALC (using KM) performs well for most of the datasets we further evaluate ALC on top of ID. We set different threshold values on top of the Neural Model ($\mathcal{M}$) to identify rejected utterances and optimum results are obtained when the threshold is set to 75% of the maximum classification probability score for a particular dataset. We pass these rejected utterances through a majority voting classifier (the same five classifiers as used previously). If three or more classifiers detect the same label for a rejected utterance, it is auto-labeled (auto-corrected). The rest of the samples (i.e. Majority voting rejection samples) are annotated manually. These auto-corrected and manually annotated samples are removed from the unlabeled set and added back to the labeled dataset, $\mathcal{L}$ for retraining purposes. Thus the auto-correction criteria, save significant human labeling costs.

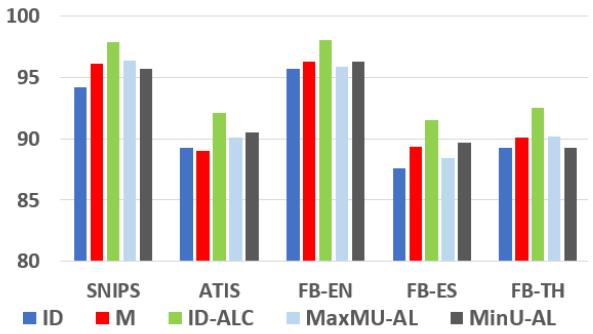

Fig. 4: Accuracies (%) for Final ID, Joint Bert, IDALC, Max and Min Uncertainty-AL models

1) Baselines: We apply Maximum Marginal Uncertainty (MaxMU-AL) with minimum interaction [56] and Minimum Uncertainty (MinU-AL) based Active Learning model [55] on top of intention detection (ID(1)+KM) output. We use ID(1)+KM, Joint Bert Neural Model ($\mathcal{M}$), MaxMU-AL, and MinU-AL as baselines.

2) Results and Discussion: The addition of the ALC component on top of ID gives a higher boost in performance across all datasets as shown in Tables IV and V. IDALC outperforms other baseline methods (along with different active learning based) across all datasets as shown in Fig 4. We experiment with applying Active learning frameworks (ALC on top of ID) for five different cycles across all datasets as in Fig 5. From the third cycle onwards, improvements are not significant for particular threshold values so we can ignore that to save extra retraining costs.

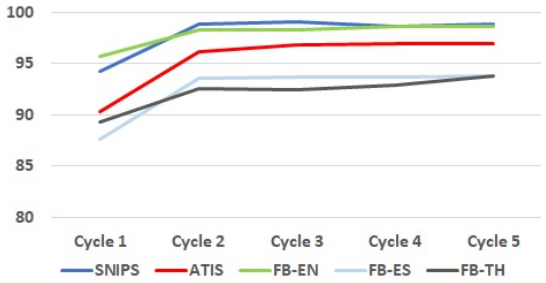

Fig. 5: Accuracy (%) across different cycles of ALC module, showing iterative improvement and supporting the choice of limiting the process to two cycles for efficiency.

We evaluate the ALC model performance by checking what fraction of data is getting automatically labeled by the Majority Voting algorithm and added back to training data. Majority Voting can predict $\sim$90% low threshold samples with more than 80% accuracy as shown in Table VI. Accuracy is very good for SNIPS (95.9%), Facebook-English (92.69%), and Facebook Spanish (94.35%) where ALC can auto-correct

Dataset	SNIPS	ATIS	FB-EN	FB-ES	FB-TH
Corrections (%)	98.21	91.86	89.71	93.91	90.62
Acc (%)	95.9	81.4	92.69	94.35	82.65

TABLE VI: Majority Voting Corrections of Low Threshold Samples and Accuracy (Acc) in %

ATIS and FB-Thai with decent accuracy. Thus majority voting module is very important for the overall architecture as it boosts the overall performance without extra neural model retraining costs and optimized annotations. Another performance assessment is done for ALC model by evaluating the annotation count for the active learning system. Here, we also annotate majority voting rejected samples during annotation. Table VII shows the percentage of unlabeled samples required for annotation when we use different Majority Voting - No Voting (No MV), at least three (MV($\leq$3)), at least four (MV($\leq$4)) and all classifier agreement (MV($\leq$5)) in ALC. The best results are obtained when we use MV($\leq$3) - around 10% of the dataset needs to be annotated to get good results.

C. Human Annotation Cost

In Table VIII, we report the overall annotation cost of IDALC on the unlabeled data ($\mathcal{U}$). The cost ranges from 6–10% of the samples, which represents a reasonable trade-off given the substantial performance gains. This indicates that IDALC can reduce labeling effort by more than 90% compared to a fully supervised setup (i.e. 100% labeled), where the entire dataset would need to be annotated. Despite this significant reduction in manual labeling, IDALC achieves high accuracy and macro-F1, thereby substantially lowering annotation costs. In contrast, other baselines differ in their labeling requirements: semi-supervised and streaming approaches (e.g., SEEN, SENC-MaS, SENNE) often assume a large initial labeled pool or require frequent labeling during deployment, while zero-shot and few-shot methods rely on minimal labels but typically suffer from lower accuracy and adaptability. To further validate our claim, we ran an experiment where baselines were provided with a random subset of the same size as IDALC's labeling budget, in addition to the initial labeled set. Their performance was comparable or lower, confirming that IDALC's advantage stems from its informed sample selection rather than labeling budget size alone.

D. Interpretability

Interpretability in intent detection appears in different ways depending on the method. Clustering-based and semi-supervised approaches provide some clarity through cluster centroids or prototype assignments, which highlight typical examples and show how new intents are grouped. Zero-shot and OOD methods rely on embedding similarity, which offers a basic explanation by pointing to the closest known intent or label description, though it does not show token-level details. Large language models mainly provide output-level explanations rather than fine-grained reasoning. In IDALC, interpretability comes from k-means clustering, which gives centroid-based insights, and from base classifiers like logistic regression and linear discriminant analysis, where learned weights can be examined. Components such as JointBERT and the ensemble voting in the correction module are primarily designed for improving accuracy. At the user feedback level, IDALC provides additional transparency through the human-in-the-loop setup, where annotators can view errors and guide the system with feedback. While interpretability is valuable and will be explored further, the main focus of this study is on performance: finding new intents without reducing accuracy on known ones, while keeping annotation needs low. This makes the approach practical for real-world applications and adaptable across multiple languages.

Dataset	No MV	MV($\geq$3)	MV($\geq$4)	MV($\geq$5)
SNIPS	46.72	1.79	4.61	7.50
ATIS	42.31	8.14	13.53	19.62
FB-EN	39.45	10.29	15.34	23.85
FB-ES	49.68	6.09	10.78	16.75
FB-Th	41.57	10.38	16.82	22.86

TABLE VII: Annotation requirement (%) for different conditions of Majority Voting across different datasets

E. Real World Scenarios

We explore the applicability of the IDALC approach in various real-world scenarios.

(i) In practical applications, user queries often contain multiple intents. To evaluate our approach in such cases, we used two benchmark datasets for multi-intent detection: Mix-SNIPS [65] and Mix-ATIS [65]. These datasets consist of queries with multiple intent labels. We extended the IDALC architecture to support sentence-level multi-label detection, enabling the system to predict multiple intents for a single query. The modified model achieved 76.45% accuracy on Mix-SNIPS and 70.37% on Mix-ATIS, showing its effectiveness and robustness in handling multi-intent queries.

(ii) In scenarios requiring quick responses, our algorithm can detect both known and unknown intents by combining zero-shot or few-shot learning with a majority-voting-based automated method within the IDALC framework. This allows the system to generate responses without extensive manual labeling while maintaining reliable performance. Our approach achieves 81.76% accuracy on SNIPS, 70.23% on ATIS, and 78.45% on Fb-EN. A key advantage is the reduced reliance on manual annotation, which in traditional methods requires labeling 6-10% of unlabeled data. In large datasets, this translates to hundreds or thousands of samples, creating bottlenecks for real-time applications such as customer service bots, virtual assistants, or emergency response systems. By reducing manual labeling needs, our method improves scalability and responsiveness.

(iii) Our system can also operate in streaming settings where unlabeled data arrives sequentially. At each step, IDALC applies the BERT-based classifier (110M parameters, requiring about 4- 5 GB of GPU memory), making it efficient enough for local or edge devices. High-confidence predictions are accepted directly, while low-confidence cases are checked by an ensemble of lightweight classifiers (Random Forest, Logistic Regression, Bagging, KNN, LDA). If consensus is reached, the corrected intent is returned immediately; otherwise, the sample is flagged as "unknown" or queued for manual annotation. This incremental process supports real-time deployment with low latency, while the BERT model can be updated continuously with new labeled data.

VI. Experimental Setup

All experiments were conducted on two Tesla P100 GPUs (16 GB RAM, 6 Gbps, GDDR5) for BERT-based models and one A100 GPU (48 GB RAM, dual 960 GB SSDs) for LLaMA2. Most models trained within 120 GPU minutes, while LLaMA2-7B took about 3–5 hours. LLaMA2-7B was fine-tuned using QLoRA, which applies 4-bit quantization with low-rank adapters (LoRA). A rank of 8 was used for all projection matrices (Q, K, V, O) with LoRA $\alpha = 32$. For in-context prompting, we used ChatGPT-3.5 with default settings (temperature = 1.0, top-p = 0.8, top-k = 50, max output tokens = 10). Code/Data details are in link[6]All fine-tuned models were trained for 10 epochs with AdamW (weight decay = 0.01, $\beta = (0.9, 0.999)$). The learning rate was chosen through a small-scale empirical search and fixed at 5×10^{-5}, using cross-entropy loss. The dataset was split 80:20 for training and validation, specifically during hyperparameter search. The batch size was fixed at 16 to keep the model lightweight and practical for real-world use. The BERT-based IDALC model already required about 4–5 GB of memory for inference, with additional overhead from optimizer states, activations, and handling long sequences. Larger batch sizes quickly exceeded GPU capacity, while smaller ones only slowed training without any performance gains. This balance ensured the model remained efficient and deployable under realistic resource constraints. Preprocessing used NLTK, SpaCy, and Scikit-learn for tokenization, lowercasing, and removal of stop words, punctuation, and non-alphabetic tokens. Evaluation used Scikit-learn with accuracy, precision, recall, and F1 score.

VII. Error Analysis

We perform an error analysis of different parts of our proposed framework to gain more insights.

(i) The trained model, $\mathcal{M}$, sometimes falters when two intents are very similar. On the SNIPS dataset, the sentence "*i m wondering when i can see hurry sundown*" has gold label "*SearchScreeningEvent*" but the model predicts as "*SearchCreativeWork*" since both are similar. Similarly, the system wrongly classifies "*play hell house song*" as 'AddToPlayList' but the actual intent is 'PlayMusic' which are very close to each other.

(ii) Majority Voting does not perform well in ID due to less annotated data (novel classes) for training, but works well in ALC as training is done on more data (labeled data ($\mathcal{L}$)).

(iii) Out-of-Distribution (OOD) detection and clustering approaches exhibit limited effectiveness with ATIS datasets, primarily due to the limited amount of data and the high similarity among known and unknown intents. Our current strategy does not specifically address the challenges posed by datasets with very few samples. We plan to explore further.

VIII. Limitations

While our study shows promising results, it has several limitations that open important directions for future work: (1) ID-ALC relies on human-in-the-loop feedback, which improves accuracy but requires costly manual annotation; we plan to explore zero-shot adaptation using self-learning and large language models. (2) The framework currently supports only text, whereas real-world applications often involve voice, images, or mixed modalities; we will scale datasets and replace classifiers with multimodal models such as SDIF-DA [66], MulT [67], MISA [68], and MAG-BERT [69]. (3) Our experiments are limited to English, Thai, and Spanish, leaving out code-mixed or code-switched settings; extending to such cases will require multilingual models with alignment across framework components. (4) We also plan to evaluate low-resource languages and domain-specific datasets (e.g., education, healthcare, law) to test adaptability beyond general-purpose intent detection. (5) While the current work focuses on optimizing performance with minimal annotation, future extensions will address the interpretability of IDALC, as understanding which features or interactions drive clustering and adaptation can improve trust, usability, and applicability in expert-facing domains.

[6]https://github.com/IDALC-TAI/IDALC

Dataset	Type	# ID	# ALC	# Total	# Unlab	% annot
SNIPS	1	589	29	618	9110	**6.78**
	2	669	43	712	9110	**7.82**
	3	613	37	650	9110	**7.14**
FB	EN	1600	984	2584	25000	**10.36**
	ES	240	55	295	3999	**7.38**
	TH	250	54	304	3000	**10.13**
ATIS	-	100	260	360	3370	**10.70**

TABLE VIII: Summary of annotated instances for the IDALC framework across different datasets and configurations. The columns indicate the number of annotated intent (ID) and active learning cycles (ALC), the total annotated samples, the size of the corresponding unlabeled pool, and the percentage of annotated data relative to the unlabeled set.

IX. Conclusion

We present **IDALC** (Intent Detection and Active Learning based Correction), a semi-supervised framework designed to handle rejected utterances and discover new intents with minimal human annotation. IDALC combines active learning with majority voting to select informative samples, enabling iterative improvement while reducing annotation cost. The framework is scalable, integrates easily with existing dialog systems, and remains effective even in low-resource scenarios. Experiments on multiple benchmarks demonstrate consistent gains over baselines, with 5–10% higher accuracy and 4–8% better macro-F1, while requiring annotation for only 6–10% of unlabeled data. Looking ahead, we aim to extend IDALC to dynamically evolving intent classes, incorporate zero- and few-shot techniques for further reducing supervision, and study interpretability to understand which features and interactions drive model decisions. We also plan to evaluate the framework in real-world domains such as finance, healthcare, and law, where both efficiency and transparency are critical. IDALC can serve as a building block for next-generation dialog systems with structured intent understanding, ensuring reliable, adaptive, and trustworthy user interactions.

ETHICAL IMPLICATIONS

We use publicly available datasets and do not contain any sensitive personal information or harmful content. They come with existing annotations, which we use as labels. During prediction, we mask these labels and predict them. We also have not used any AI-generated text.

REFERENCES

[1] G. Liao, P. Zhang, H. Yin, X. Deng, Y. Li, H. Zhou, and D. Zhao, "A novel semi-supervised classification approach for evolving data streams," *Expert Systems with Applications*, vol. 215, p. 119273, 2023.

[2] I. Kuzborskij, F. Orabona, and B. Caputo, "From n to n+ 1: Multiclass transfer incremental learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3358–3365.

[3] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 7, pp. 1757–1772, 2012.

[4] A. Degirmenci and O. Karal, "Efficient density and cluster based incremental outlier detection in data streams," *Information Sciences*, vol. 607, pp. 901–920, 2022.

[5] Y. Sun, K. Tang, L. L. Minku, S. Wang, and X. Yao, "Online ensemble learning of data streams with gradually evolved classes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1532–1545, 2016.

[6] M. Masud, J. Gao, L. Khan, J. Han, and B. M. Thuraisingham, "Classification and novel class detection in concept-drifting data streams under time constraints," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 6, pp. 859–874, 2010.

[7] A. Haque, L. Khan, and M. Baron, "Sand: Semi-supervised adaptive novel class detection and classification over data stream," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[8] A. Mullick, S. Purkayastha, P. Goyal, and N. Ganguly, "A framework to generate high-quality datapoints for multiple novel intent detection," *arXiv preprint arXiv:2205.02005*, 2022.

[9] A. Mullick, A. Nandy, M. N. Kapadnis, S. Patnaik, and R. Raghav, "Fine-grained intent classification in the legal domain," *arXiv preprint arXiv:2205.03509*, 2022.

[10] A. Mullick, "Novel intent detection and active learning based classification (student abstract)," in *Proceedings of the AAAI Conference on Artificial Intelligence*, no. 13, 2023, pp. 16 286–16 287.

[11] Mullick, "Exploring multilingual intent dynamics and applications," in *IJCAI*, 2023, pp. 7087–7088.

[12] A. Mullick, A. Nandy, M. Kapadnis, S. Patnaik, R. Raghav, and R. Kar, "An evaluation framework for legal document summarization," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 2022, pp. 4747–4753.

[13] A. Mullick, S. Bose, A. Nandy, G. Chaitanya, and P. Goyal, "A pointer network-based approach for joint extraction and detection of multi-label multi-class intents," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, pp. 15 664–15 680.

[14] A. Mullick, I. Mondal, S. Ray, R. Raghav, G. Chaitanya, and P. Goyal, "Intent identification and entity extraction for healthcare queries in indic languages," in *Findings of the Association for Computational Linguistics: EACL 2023*, 2023, pp. 1825–1836.

[15] A. Mullick, M. Gupta, and P. Goyal, "Intent detection and entity extraction from biomedical literature," *LREC-COLING 2024*, p. 271, 2024.

[16] A. Mullick, S. Sharma, A. Jana, and P. Goyal, "Text takes over: A study of modality bias in multimodal intent detection," *arXiv preprint arXiv:2508.16122*, 2025.

[17] A. Mullick, A. Ghosh, G. S. Chaitanya, S. Ghui, T. Nayak, S.-C. Lee, S. Bhattacharjee, and P. Goyal, "Matscire: Leveraging pointer networks to automate entity and relation extraction for material science knowledge-base construction," *Computational Materials Science*, vol. 233, p. 112659, 2024.

[18] S. Guha, A. Mullick, J. Agrawal, S. Ram, S. Ghui, S.-C. Lee, S. Bhattacharjee, and P. Goyal, "Matscie: An automated tool for the generation of databases of methods and parameters used in the computational materials science literature," *Computational Materials Science (Comput. Mater. Sci.)*, vol. 192, p. 110325, 2021.

[19] A. Mullick, S. Pal, T. Nayak, S.-C. Lee, S. Bhattacharjee, and P. Goyal, "Using sentence-level classification helps entity extraction from material science literature," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 2022, pp. 4540–4545.

[20] A. Mullick, A. Bera, and T. Nayak, "Rte: A tool for annotating relation triplets from text," *arXiv preprint arXiv:2108.08184*, 2021.

[21] A. Mullick, S. Maheshwari, P. Goyal, and N. Ganguly, "A generic opinion-fact classifier with application in understanding opinionatedness in various news section," in *Proceedings of the 26th International Conference on World Wide Web Companion*, 2017, pp. 827–828.

[22] A. Mullick, P. Goyal, and N. Ganguly, "A graphical framework to detect and categorize diverse opinions from online news," in *Proceedings of the Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media (PEOPLES)*, 2016, pp. 40–49.

[23] A. Mullick, S. Ghosh D, S. Maheswari, S. Sahoo, S. K. Maity, and P. Goyal, "Identifying opinion and fact subcategories from the social web," in *Proceedings of the 2018 ACM International Conference on Supporting Group Work*, 2018, pp. 145–149.

[24] A. Mullick, P. Goyal, N. Ganguly, and M. Gupta, "Harnessing twitter for answering opinion list queries," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 4, pp. 1083–1095, 2018.

[25] A. Mullick, S. Pal, P. Chanda, A. Panigrahy, A. Bharadwaj, S. Singh, and T. Dam, "D-fj: Deep neural network based factuality judgment," *Technology*, vol. 50, p. 173, 2019.

[26] A. Mullick, P. Goyal, N. Ganguly, and M. Gupta, "Extracting social lists from twitter," in *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, 2017, pp. 391–394.

[27] A. Mullick and M. Gupta, "Avenues in iot with advances in artificial intelligence," in *2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, 2024, pp. 314–319.

[28] A. Mullick, S. Bose, R. Saha, A. K. Bhowmick, A. Vempaty, P. Dey, R. Kokku, P. Goyal, and N. Ganguly, "Introducing spotlight: A novel approach for generating captivating key information from documents," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025, pp. 35 449–35 477.

[29] E. Liberty, "Simple and deterministic matrix sketching," in *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 581–588.

[30] X. Mu, F. Zhu, J. Du, E.-P. Lim, and Z.-H. Zhou, "Streaming classification with emerging new class by class matrix sketching," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[31] X. Mu, K. M. Ting, and Z.-H. Zhou, "Classification under streaming emerging new classes: A solution using completely-random trees," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 8, pp. 1605–1618, 2017.

[32] J. Cui, L. Zong, J. Xie, and M. Tang, "A novel multi-module integrated intrusion detection system for high-dimensional imbalanced data," *Applied Intelligence*, vol. 53, no. 1, pp. 272–288, 2023.

[33] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, and S. Krishnaswamy, "Anynovel: detection of novel concepts in evolving data streams," *Evolving Systems*, vol. 7, no. 2, pp. 73–93, 2016.

[34] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, "Detecting twenty-thousand classes using image-level supervision," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*. Springer, 2022, pp. 350–368.

[35] G. S. Na, D. Kim, and H. Yu, "Dilof: Effective and memory efficient local outlier detection in data streams," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1993–2002.

[36] L.-M. Zhan, H. Liang, B. Liu, L. Fan, X.-M. Wu, and A. Lam, "Out-of-scope intent detection with self-supervision and discriminative training," *arXiv preprint arXiv:2106.08616*, 2021.

[37] S. Larson, A. Mahendran, J. J. Peper, C. Clarke, A. Lee, P. Hill, J. K. Kummerfeld, K. Leach, M. A. Laurenzano, L. Tang, and J. Mars, "An evaluation dataset for intent classification and out-of-scope prediction," 2019.

[38] G. Yan, L. Fan, Q. Li, H. Liu, X. Zhang, X.-M. Wu, and A. Y. Lam, "Unknown intent detection using gaussian mixture model with an application to zero-shot intent classification," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 1050–1060.

[39] Y. Zhou, P. Liu, and X. Qiu, "Knn-contrastive learning for out-of-domain intent classification," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 5129–5141.

[40] M. Firdaus, A. Ekbal, and E. Cambria, "Multitask learning for multilingual intent detection and slot filling in dialogue systems," *Information Fusion*, vol. 91, pp. 299–315, 2023.

[41] Y.-N. Zhu and Y.-F. Li, "Semi-supervised streaming learning with emerging new labels," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 7015–7022.

[42] M. Tan, Y. Yu, H. Wang, D. Wang, S. Potdar, S. Chang, and M. Yu, "Out-of-domain detection for low-resource text classification tasks," *arXiv preprint arXiv:1909.05357*, 2019.

[43] Z. Zhu, X. Cheng, Y. Li, H. Li, and Y. Zou, "Aligner^2: Enhancing joint multiple intent detection and slot filling via adjustive and forced cross-task alignment," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, 2024, pp. 19 777–19 785.

[44] D. Wu, L. Jiang, L. Yin, Z. Li, and H. Huang, "Ceanet: a co-interactive external attention network for joint intent detection and slot filling," *Neural Computing and Applications*, pp. 1–13, 2024.

[45] S. Park, C. C. Menassa, and V. R. Kamat, "Joint bert model for intent classification and slot filling analysis of natural language instructions in co-robotic field construction work," in *Computing in Civil Engineering 2023*, 2024, pp. 453–460.

[46] X.-Q. Cai, P. Zhao, K.-M. Ting, X. Mu, and Y. Jiang, "Nearest neighbor ensembles: An effective method for difficult problems in streaming classification with emerging new classes," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 970–975.

[47] C. Xia, W. Yin, Y. Feng, and P. Yu, "Incremental few-shot text classification with multi-round new classes: Formulation, dataset and system," *arXiv preprint arXiv:2104.11882*, 2021.

[48] K. Halder, A. Akbik, J. Krapac, and R. Vollgraf, "Task-aware representation of sentences for generic text classification," in *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 3202–3213. [Online]. Available: https://aclanthology.org/2020.coling-main.285

[49] M. Wang, K. Fu, F. Min, and X. Jia, "Active learning through label error statistical methods," *Knowledge-Based Systems*, vol. 189, p. 105140, 2020.

[50] M. Salama, H. Abdelkader, and A. Abdelwahab, "A novel ensemble approach for heterogeneous data with active learning," *International Journal of Engineering Business Management*, vol. 14, p. 18479790221082605, 2022.

[51] G. Senay, B. Y. Idrissi, and M. Haziza, "Viraal:

Virtual adversarial active learning," *arXiv preprint arXiv:2005.07287*, 2020.

[52] H. Zhang, H. Xu, and T.-E. Lin, "Deep open intent classification with adaptive decision boundary," *arXiv preprint arXiv:2012.10209*, 2020.

[53] C. Yin, S. Chen, and Z. Yin, "Clustering-based active learning classification towards data stream," *ACM Transactions on Intelligent Systems and Technology*, 2023.

[54] M. Gao, Z. Zhang, G. Yu, S. Ö. Arık, L. S. Davis, and T. Pfister, "Consistency-based semi-supervised active learning: Towards minimizing labeling cost," in *European Conference on Computer Vision*. Springer, 2020, pp. 510–526.

[55] T. Danka and P. Horvath, "modAL: A modular active learning framework for Python," available on arXiv at https://arxiv.org/abs/1805.00979. [Online]. Available: https://github.com/modAL-python/modAL

[56] A. Tian and M. Lease, "Active learning to maximize accuracy vs. effort in interactive information retrieval," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011, pp. 145–154.

[57] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, M. Primet, and J. Dureau, "Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces," 2018.

[58] G. Tur, D. Hakkani-Tür, and L. Heck, "What is left to be understood in atis?" in *2010 IEEE Spoken Language Technology Workshop*. IEEE, 2010, pp. 19–24.

[59] S. Schuster, S. Gupta, R. Shah, and M. Lewis, "Cross-lingual transfer learning for multilingual task oriented dialog," *arXiv preprint arXiv:1810.13327*, 2018.

[60] Q. Chen, Z. Zhuo, and W. Wang, "Bert for joint intent classification and slot filling," 2019.

[61] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[62] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," 2018.

[63] L. Shu, H. Xu, and B. Liu, "Doc: Deep open classification of text documents," 2017.

[64] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA: Association for Computing Machinery, 2000, p. 93–104.

[65] L. Qin, X. Xu, W. Che, and T. Liu, "Agif: An adaptive graph-interactive framework for joint multiple intent detection and slot filling," *arXiv preprint arXiv:2004.10087*, 2020.

[66] S. Huang, L. Qin, B. Wang, G. Tu, and R. Xu, "Sdif-da: A shallow-to-deep interaction framework with data augmentation for multi-modal intent detection," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 10 206–10 210.

[67] Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov, "Multimodal transformer for unaligned multimodal language sequences," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 6558–6569. [Online]. Available: https://aclanthology.org/P19-1656/

[68] D. Hazarika, R. Zimmermann, and S. Poria, "Misa: Modality-invariant and-specific representations for multimodal sentiment analysis," in *Proceedings of the 28th ACM international conference on multimedia*, 2020, pp. 1122–1131.

[69] W. Rahman, M. K. Hasan, S. Lee, A. Bagher Zadeh, C. Mao, L.-P. Morency, and E. Hoque, "Integrating multimodal information in large pretrained transformers," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 2359–2369. [Online]. Available: https://aclanthology.org/2020.acl-main.214