Multi-Label Intent Detection via Contrastive Task Specialization of Sentence Encoders

Ivan Vulič, Iñigo Casanueva, Georgios Spithourakis, Avishek Mondal, Tsung-Hsien Wen and Paweł Budzianowski

PolyAI Limited
London, United Kingdom
poly.ai

Abstract

Deploying task-oriented dialog (ToD) systems for new domains and tasks requires natural language understanding models that are (1) resource-efficient and work under low-data regimes; (2) adaptable, efficient, and quick-to-train; (3) expressive and can handle complex ToD scenarios with multiple user intents in a single utterance. Motivated by these requirements, we introduce a novel framework for multi-label intent detection (mID): MULTI-CONVFiT (Multi-Label Intent Detection via Contrastive Conversational Fine-Tuning). While previous work on efficient single-label intent detection learns a classifier on top of a fixed sentence encoder (SE), we propose to 1) transform general-purpose SEs into task-specialized SEs via contrastive fine-tuning on annotated multi-label data, 2) where task specialization knowledge can be stored into lightweight adapter modules without updating the original parameters of the input SE, and then 3) we build improved mID classifiers stacked on top of fixed specialized SEs. Our main results indicate that MULTI-CONVFiT yields effective mID models, with large gains over non-specialized SEs reported across a spectrum of different mID datasets, both in low-data and high-data regimes.

1 Introduction

Task-oriented dialog (ToD) systems allow users to interact with computer applications through conversation in order to solve a particular task with well-defined semantics (Levin and Pieraccini, 1995; Young, 2010). ToD supports a multitude of applications such as automating different customer service tasks, facilitating bookings in hospitality and travel industries, or providing assistance in healthcare or finance (Raux et al., 2003; El Asri et al., 2017; Xu et al., 2017; Budzianowski et al., 2018).

Intent detection (ID), a task of recognizing the user’s intent or goal from their utterance, is a crucial component of any ToD system (Hemphill et al., 1990; Tür et al., 2010; Coucke et al., 2018). Intent detectors that adhere to industry standards should satisfy the following three requirements (Casanueva et al., 2020; Larson and Leach, 2022).

(R1) Scalability and resource efficiency. They must be quickly bootstrapped for new domains and tasks. However, this process requires creating expensive annotations for each domain and task of interest, which calls for sample-efficient ID methods that achieve strong performance in low-data scenarios.

(R2) Lightweight design and modularity. While large language models (LMs) have shown strong performance in the ID task, running full-fledged fine-tuning and storing separate fine-tuned models per each domain or task yields prohibitive storage and memory costs (Ding et al., 2022). Enabling fast training of intent detectors (Casanueva et al., 2020) also speeds up ToD development cycles.

(R3) Expressiveness: supporting complex ToD scenarios. Previous work has typically focused on more limited single-label ID scenarios (Liu et al., 2019a; Larson et al., 2019; Wu et al., 2020; Mehri et al., 2020, inter alia). Such setups are not realistic in more complex industry settings and even lead to limited and simplified conversational scenarios with artificial intent sets. Intent detectors should thus tackle the more challenging multi-label ID (mID) task, which enables more complex ‘real-life’ natural language understanding for ToD (Qin et al., 2021; Hou et al., 2021; Casanueva et al., 2022).

In this work, we propose a novel framework for mID, termed MULTI-CONVFiT (Multi-Label Intent Detection via Contrastive Conversational Fine-Tuning), that satisfies the three requirements R1-R3. The framework’s pipeline is illustrated in Figure 1. Previous work typically used fixed general-purpose (Henderson et al., 2020; Casanueva et al., 2020) sentence encoders (SEs) (Cer et al., 2018) combined with tunable intent
classifiers for efficient single-label ID. In this work, we propose a modular framework that contrastively fine-tunes general-purpose encoders, using mID data annotations implicitly, to yield task-specialized sentence encoders. All the task-specific ‘adaptation’ knowledge after contrastive fine-tuning can be injected into small adapter modules. (Houlsby et al., 2019; Pfeiffer et al., 2020a).

Such adapter modules are then used to adapt the underlying general-purpose SE which already stores plenty of useful semantic knowledge – a single large model that serves multiple tasks and domains – into the task-specialized SE. The contrastive procedure creates a semantic space which better aligns with intent classes of a particular mID task, as demonstrated in Figure 2. Consequently, such fixed task-specialized SEs enable learning improved mID classifiers that outperform mID classifiers learnt on top of the original general SEs.

Our key results indicate effectiveness and robustness of multi-CONVFIT, yielding state-of-the-art results across four representative mID datasets, both in low-data and high-data scenarios, while offering modularity and efficient fine-tuning and inference. Additional analyses indicate multi-CONVFIT’s versatility and wide applicability: it can be used with a range of pretrained SEs and LMs, and it leads to gains across different domains, dataset sizes, and intent set sizes.

2 Methodology

The full overview of multi-CONVFIT, aiming to satisfy the requirements R1-R3 from §1, is provided in Figure 1. In what follows, we discuss its main components in detail: contrastive task-specialization of general-purpose input encoders using annotated mID data (§2.1); a classifier for multi-label ID stacked on top of the fixed encoder (§2.2); and a more efficient variant of the framework which combines contrastive tuning with adapters (§2.3).

Preliminaries. For any input text \( t \), we obtain its encoding \( f(t) \), where \( f \) is an encoding function of any input encoder model (i.e., LM, general-purpose or task-specialized SE). \( t \) is tokenized into subwords using each encoder’s dedicated tokenizer. The final encoding \( t \) is created via a pooling operation such as (a) using the [CLS] token as the text representation, (b) or mean-pooling the output subword encodings. Following prior work (Reimers and Gurevych, 2019; Liu et al., 2021), we opt for mean-pooling as a better-performing option.

Further, we assume that \( |S| \) annotated mID data examples are available: they comprise a set of pairs \( S = \{(s_1, L_1), \ldots, (s_i, L_i), \ldots, (s_{|S|}, L_{|S|})\} \), where \( s_i \) is a sentence/example, each annotated with a set of \( L_i = \{l_{i,1}, \ldots, l_{i,M_i}\} \) labels, where \( M_i \geq 0 \). Each label \( l \) is in fact one of the \( |C| \) intent classes from the set \( C = \{c_1, \ldots, c_{|C|}\} \).

2.1 Contrastive Specialization

Motivation. The main idea is to specialize the input general-purpose encoder relying on available ID annotations so that the encoder better aligns with the actual ID task semantics. Such specialization of general-purpose encoders has been proven effective in prior work on single-label nonparametric ID (Zhang et al., 2020, 2021; Mehr et al., 2021; Vulić et al., 2021). Whereas the ‘ID task seman-
The rationale is to enable the encoder to focus on parts of the sentences that yield shared labels/intent(s), and re-shape the semantic space so that sentences with a larger proportion of shared intents end up more similar in the fine-tuned space. For instance, imagine a toy scenario with three classes $c_x$, $c_y$, $c_z$: the procedure should cluster together all single-label examples (i.e., all $c_x$ examples should be grouped together, and another two coherent clusters are $c_y$ and $c_z$ examples). At the same time, all two-label examples should also create coherent clusters, and examples labeled with $c_x$ and $c_y$ should end up closer to the single-label $c_x$ and $c_y$ clusters than to the $c_z$ cluster, etc. This effect is indeed observed with real mID data, as plotted in Figure 2. For instance, we observe that sentences labeled with intents (cancel, account) are indeed in encoded in a

2. The OCL loss, among other tasks, demonstrated strong performance in single-label ID tasks in prior work (Vulić et al., 2021). The ‘online’ formulation typically results in quicker convergence and better performance, also confirmed in our preliminary experiments. See also www.sbert.net/docs/package_reference/losses.html. Future work will delve deeper into experimenting with other contrastive losses.
subspace between the clusters of ‘cancel-only’ and ‘account-only’ sentences, while the sentences with \( \text{make, account} \) are encoded in another cluster between ‘make-only’ and ‘account-only’ sentences.

### 2.2 Multi-Label Classifier

A standard approach to efficient intent detection in single-label scenarios is to stack a classifier on top of a fixed sentence encoder (Casanueva et al., 2020; Gerz et al., 2021). While it is much more lightweight than fine-tuning the entire SE (Mehri et al., 2020), this approach typically yields on-par performance in single-label ID tasks (Casanueva et al., 2020). Following prior work, our classifier is a standard Multi-Layer Perceptron (MLP), stacked on top of the fixed sentence representations \( f(s) \), which were previously obtained with any fixed input encoder (see Figure 1). The MLP classifier comprises a single hidden layer with non-linearity, followed by a \textit{sigmoid} layer to allow for multi-label classification. It is trained via standard binary cross-entropy loss. A tunable threshold \( \theta \) determines the final classification: only intents with their probability scores \( \geq \theta \) are taken as positives. This way, the threshold \( \theta \) effectively controls the trade-off between precision and recall of the classifier.

### Label Smoothing

In contrast to prior work in single-label ID setups, we propose to add label smoothing (Müller et al., 2019) into classifier training, and later validate its impact on mID performance. This label smoothing regularization should mitigate overfitting and classification overconfidence in low-data setups (Bai et al., 2021), where such overconfidence might get even more pronounced with contrastively specialized encoders. Since the label smoothing technique has not been used in prior work on ID and mID, we provide a full description in what follows.

We leverage a standard label smoothing technique, additionally ‘corrected’ for multi-label classification (Hou et al., 2019). Without label smoothing, for the item \( \{s_i, L_i\} \in S \) the conversion of \( L_i \) into a \( |C| \)-dimensional vector \( Y_i = [y_{i,1}, \ldots, y_{i,|C|}] \) of binary labels assigns 1-s to labeled classes from \( C \), and 0-s otherwise. Adding label smoothing with the value \( \lambda \) then means reassigning all the individual binary indicators \( y_i \)-s to the following \( y'_i \)-values:

\[
y'_{i,k} = \begin{cases} 
    \lambda & \text{if } y_{i,k} = 1, \\
    \frac{1 - \lambda}{|C|} & \text{otherwise.}
\end{cases}
\]

\( M_i \) is the number of positive labels for the example \( s_i \) (i.e., the number of 1-s in \( Y_i \)). This reassigns some of the probability mass from the positive labels to the negative ones, this way reducing the classifier’s (over)confidence (Pereyra et al., 2017).

### 2.3 Sentence Encoders with Adapters

We always learn the classifier on top of fixed sentence encoders. However, the contrastive task specialization must adapt the weights of the general-purpose input encoder. A standard variant, termed C-FFT, does full fine-tuning of all the weights, meaning that a separate full copy of the specialized model must be stored per each specialization.

However, with multiple dialog domains and tasks, requirements such as model compactness, fine-tuning and storage efficiency become crucial features; see again the main requirements listed in §1. We thus propose to combine fine-tuning of general-purpose SEs with lightweight tunable adapter modules (Houlsby et al., 2019; Pfeiffer et al., 2021). Such adapters, whose size is typically only a fraction of the size of the full input neural model, are inserted within each Transformer layer of the underlying model. At fine-tuning, only adapter parameters are updated while all the other parameters of the large model are kept fixed, which enables parameter-efficient and modular adaptation of large neural models (He et al., 2022).

Unlike prior work which typically combined adapters with off-the-shelf large LMs (Pfeiffer et al., 2020a; He et al., 2022), here we focus on inserting adapter modules directly into general-purpose sentence encoders. We create small task-specialized modules (Madotto et al., 2020; Pfeiffer et al., 2020b) that transform a large general-purpose SE into a particular domain- or task-specialized SE without full fine-tuning. Since a single general-purpose SE can serve multiple domains and tasks without incurring catastrophic forgetting and interference (McCloskey and Cohen, 1989; Hashimoto et al., 2017), this approach increases modularity and decreases storage demands. We note that MULTI-COVfIT can be directly applied to the single-label ID scenario (Mehri et al., 2021; Vulić et al., 2021; Zhang et al., 2021) as a special case. The efficient adapter-based SE tuning variant, illustrated in Figure 1, is dubbed C-ADAPT.

### 3 Experimental Setup

In what follows, we outline our experimental setup, focused on evaluating and improving performance
and sample-efficiency of multi-label intent detectors, relying on the standard mID benchmarks.

**Input Sentence Encoders.** We experiment with several representative and popular sentence encoders as input, aiming to (i) validate the robustness of the proposed methodology across different underlying encoders, as well as to (ii) analyze the impact of the chosen encoder on the final mID task performance. We opt for the following SEs that offer a good trade-off between model size and performance in sentence-level semantic tasks (Reimers and Gurevych, 2019). 1) **MLM12**, 2) **MPNET**, 3) **DROB** are sentence encoders which transform the respective pretrained LMs – the 12-layer MiniLM (Wang et al., 2020), MPNet (Song et al., 2020), and DistilRoBERTa (Sanh et al., 2019) – into SEs via standard contrastive dual-encoder (i.e., bi-encoder) frameworks (Reimers and Gurevych, 2019; Henderson et al., 2020).^3^ MLM12 (its size is 120 MB) comprises \( L_T = 12 \) Transformer layers, with hidden size \( h_T = 384 \); \( L_T = 12 \) and \( h_T = 768 \) for MPNET (490 MB); \( L_T = 6 \) and \( h_T = 768 \) for DROB (290 MB). All SEs have been obtained in prior work (Reimers and Gurevych, 2019) via contrastively fine-tuning their underlying LMs on a set of more than 1B sentence pairs, which comprises various data such as Reddit 2015-2018 comments (Henderson et al., 2019), Natural Questions (Kwiatkowski et al., 2019), PAQ (question, answer) pairs (Lewis et al., 2021), etc.^4^

Furthermore, in order to test the impact of the chosen input/underlying model (i.e., sentence encoders versus language models) we also contrastively fine-tune the original LMs instead of their SE counterparts (see Figure 1 again): we refer to the respective input LMs as MLM12-LM, MPNET-LM, and DROB-LM.

**Multi-Label ID Datasets.** Until very recently, have been few and far between (Casanueva et al., 2022), as prior ID research predominantly focused on single-label scenarios (Larson and Leach, 2022). We experiment with a representative set of multi-label ID datasets, covering 1) four diverse domains, 2) ontologies with different sizes of the intent sets (from 18 to 118 intents), 3) varied dataset size, and 4) different average number of intents per example. A complete summary is provided in Table 1.

**MIXATIS** (Qin et al., 2020) is the only dataset whose ‘multi-label nature’ was achieved synthetically through concatenation of single-label examples from the original ATIS dataset (Hemphill et al., 1990). On the other hand, the other datasets in our evaluation are natively multi-label, relying on the concept of combining the so-called intent modules; see the work of Casanueva et al. (2022) for further details. Some examples of multi-label sentences from each mID dataset are provided in Appendix B.

**Low-Data versus High-Data Setups.** Due to the high cost of task-specific annotations, prior work on single-label ID has recognized the importance of building and bootstrapping intent detectors in low-data regimes (Casanueva et al., 2020; Mehri et al., 2021; Vulić et al., 2021), and the same naturally holds also for multi-label ID. In order to understand the behaviour of MULTI-CONVFIT in such scenarios versus setups with more abundant task-annotated data, we conduct experiments in two standard data setups: 1) **low-data** and 2) **high-data**.

For the BANKING, HOTELS, and INSURANCE-FAQ datasets (see Table 1), we adopt the standard 10-fold cross-validation (Casanueva et al., 2022). Then, in **low-data** setups we use 1 fold as our training data for contrastive fine-tuning and MLP train-

---

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Domain</th>
<th>Abbreviation</th>
<th># of Intents</th>
<th># of Examples</th>
<th>Avg. Intents per Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLU++ (Casanueva et al., 2022)</td>
<td>e-banking</td>
<td>BANKING</td>
<td>48</td>
<td>2,071</td>
<td>2.25</td>
</tr>
<tr>
<td>NLU++ (Casanueva et al., 2022)</td>
<td>hotel reservations and FAQ</td>
<td>HOTELS</td>
<td>40</td>
<td>1,009</td>
<td>1.52</td>
</tr>
<tr>
<td>(internal)</td>
<td>insurance FAQ</td>
<td>INSURANCEFAQ</td>
<td>118</td>
<td>4,356</td>
<td>1.91</td>
</tr>
<tr>
<td>MixATIS (Qin et al., 2020)</td>
<td>flight info</td>
<td>MIXATIS</td>
<td>18</td>
<td>18k/1k/1k</td>
<td>2.19/1.22/1</td>
</tr>
</tbody>
</table>

Table 1: Multi-label intent detection datasets in our experiments with key statistics. MIXATIS is the only dataset with a standardized train/dev/test split, whereas on the other datasets we run suggested 10-fold experiments; see (Casanueva et al., 2022) and §3 for further details.
ing (see Figure 1), and test the model on the remaining 9 folds. The high-data setups are effectively the same, but with swapped training and test data: we now use merged data from 9 folds as training data, and test on the single held-out fold. All the reported scores are averages over all folds. The folding evaluation comes with several benefits: 1) we avoid overfitting to any particular test set; 2) we reach more stable results with smaller training and test data (i.e., simulating low-data regimes typically met in production) through averaging over different folds; 3) variations in results due to potentially different random seeds are reduced.

For MIXATIS, we leverage its development portion for our low-data experiments, and its training portion for the high-data setup (without leveraging development data for model selection; see next for our hyper-parameter selection procedure).

Contrastive Specialization Setup. Following the suggested settings (Reimers and Gurevych, 2019; Vulic et al., 2021), we use the AdamW optimizer (Loshchilov and Hutter, 2018). The learning rate for the C-FFT variants is set to the standard value of 2e-5, while a higher learning rate of 4e-4 is used for the adapter-based C-ADAPT variants. For C-ADAPT, we opt for a standard efficient bottleneck adapter configuration, following Pfeiffer et al. (2021): ReLU activation (Nair and Hinton, 2010), with the adapter reduction factor of 4.6

The warmup rate of 0.1 with cosine decay is used; weight decay rate is 0.02. We fine-tune for 10 epochs in low-data, and for 3 epochs in high-data setups, with the number of negatives \( n = 2 \);\(^7\) batch size is 32, and max sequence length is 128.

Classifier Setup. We adopt the MLP classifier architecture from Casanueva et al. (2020): it contains 1 hidden layer of size 512 with ReLU as non-linear activation. The values for the hyperparameters were largely adopted from prior work (Reimers and Gurevych, 2019; Casanueva et al., 2020; Vulic et al., 2021), both for contrastive fine-tuning and MLP training. We further fine-tuned them relying solely on one randomly sampled fold (Fold 5)

\(^{6}\)Sec., e.g., (Pfeiffer et al., 2020a) for the definition of the reduction factor. When combined with the MLM12 SE, this adapter config requires only 3.5 MB additional parameters for each task specialization of the base MLM12 model.

\(^{7}\)We also experimented with higher \( n \) values, which substantially increase fine-tuning time while offering diminishing/negligible performance gains in the mID task in our preliminary experiments. A similar finding for single-label ID scenarios was reported by Vulic et al. (2021).

from BANKING with MLM12 and MLM12-LM in the low-data setup, and applied the same hyperparameters across all other models, setups, and runs. The classifiers’ dropout rate is fixed to 0.4, and the threshold \( \theta \) is fixed to 0.3 in all runs. We again train with AdamW, with the standard warmup rate of 0.1, weight decay of 0.02, and the learning rate is set to 0.003; 600 epochs with the batch size of 32. Unless noted otherwise, we always apply label smoothing with \( ls = 0.95 \).

Evaluation Details. All the reported scores are averaged across three runs with three different random seeds. We report standard ID evaluation metrics: \( F_1 \) and exact match accuracy (Acc).

4 Results and Discussion

The main results are summarized in Table 2 and Figure 3, while further results and analyses are available in §4.1, with additional results in Appendix C. These results offer multiple axes of comparison and analysis, discussed in what follows.

Impact of Contrastive Task Specialization. First, the results clearly demonstrate substantial and consistent gains achieved via contrastive task-specialization. The strong performance boosts are present across the board, and span all input encoders (including both SEs and LMs), both fine-tuning variants (C-FFT and C-ADAPT), all mID datasets, both low-data and high-data scenarios, and also all groups of examples with a different number of intents per example (see also Figure 9 in Appendix C). While prior work has proven that even general-purpose SEs can support effective and efficient (single-label) ID (Casanueva et al., 2020; Gerz et al., 2021; Zhang et al., 2021), here we demonstrate that 1) the efficient SE-based approach is also beneficial for the multi-label ID task, and 2) large performance improvements are achieved by transforming/adapting such general-purpose SEs into task-specialized sentence encoders.

C-FFT versus C-ADAPT. Importantly, the comparison in Figure 3 validates that the efficient C-ADAPT variant maintains strong performance across the board, offering on-par or even slightly improved performance across different setups. It indicates that task-specialized modules can be combined with large sentence encoders to obtain their task specialization. The strong performance with C-ADAPT is maintained over both data setups and using different input SEs. The C-ADAPT variant in
Table 2: $F_1$ ($\times 100\%)$ and exact match Accuracy scores (Acc; $\times 100\%)$, in the format $F_1 / Acc$, in the multi-label ID task with full model fine-tuning via supervised contrastive learning (+C-FFT). (a) C-FFT starts from a sentence encoder; (b) C-FFT starts from a language model (results with MLM12-LM, following the same trends, omitted for brevity). **Bold** numbers indicate a better-scoring configuration per each individual SE or LM architecture, whereas **underlined** numbers denote the best overall performance in each column (which includes both sub-tables).

![Figure 3: Comparison of full-model (C-FFT) and adapter-based (C-ADAPT) contrastive fine-tuning, demonstrating the competitive performance of much more parameter-efficient C-ADAPT. $F_1$ scores shown; see also Appendix C.](source)

Fact allows for building efficient, high-performing and modular multi-label intent detectors, satisfying the motivating requirements from §1.

Here, we use standard bottleneck adapters, but we believe that it is possible to strike an even better trade-off between parameter-efficiency and mID performance. A further exploration of and efficiency optimization with different adapter configurations (Pfeiffer et al., 2020a; He et al., 2022), including more efficient variants (Li and Liang, 2021; Mahabadi et al., 2021; Ansell et al., 2021, 2022), is beyond the scope of this paper, and we leave it for future work.

**Input Encoders.** As expected, the choice of the input encoder also impacts final mID performance. First, we mark that starting from SEs yields higher performance than starting from their LM-based counterparts (e.g., MPNET +C-FFT outperforms MPNET-LM +C-FFT, and the same holds with other models and fine-tuning variants). The gap is substantial in low-data setups, and it also exists even in high-data setups. This finding suggests the usefulness of conducting the adaptive fine-tuning step (Mehri et al., 2019; Henderson et al., 2020; Ruder, 2021), transforming LMs into general-purpose SEs through more suitable objectives such as response selection and paraphrase detection. Our finding corroborates a similar result in single-label ID scenarios (Vulić et al., 2021). Contrastively fine-tuning LMs with task-annotated mID data does yield large benefits in the mID task, but they cannot reach performance peaks of SEs as starting encoders.

A comparison of different input SEs reveals that the SE with the highest capacity (MPNET) yields highest absolute scores across the board. However, even the most lightweight input SE (MLM12) displays very competitive performance in all the experiments, also with the efficient C-ADAPT specialization variant. Similarly, when we start from LMs instead of SEs, MPNET-LM has a slight edge over DROB-LM. Again, we stress that applying the MULT-CONVFIT specialization yields large...
benefits regardless of the starting input encoder.

**Low-Data versus High-Data Setups.** MULTICONVFiT yields performance boosts in both data setups. As expected, absolute improvements with contrastive learning are higher in *low-data* setups (e.g., +12.6 $F_1$ in *low-data* versus +4.5 in *high-data* with MPNET +C-FFT on INSURANCEFAQ; +9.5 versus 6.9 with MLM12 +C-ADAPT on BANKING). However, we observe prominent boosts even in *high-data* setups with several thousand annotated instances (e.g., more than 4k for INSURANCEFAQ), rendering task specialization of SEs as universally useful for multi-label intent detection.

What is more, while the primary focus of MULTICONVFiT is the trade-off of performance and efficiency, the results in *high-data* setups on BANKING, HOTELS, and MIXATIS are current state-of-the-art results on all these datasets. The scores on BANKING increase from $F_1$ of 93.0 (Casanueva et al., 2022) to 94.3, while the previous high score on HOTELS of 86.7 is supersed by $F_1$ scores from Table 2 and Figure 3, reaching up to 93.4 $F_1$. The previous high scores were obtained via QA-based intent models (Namazifar et al., 2021; Casanueva et al., 2022) which require much more computationally demanding and slower training and inference regimes. The previous best-reported results on MIXATIS (*high-data*) are $F_1$ of 81.2 (Qin et al., 2020) and Acc of 76.3 (Qin et al., 2021), while we report respective peak scores of 91.5 and 81.1.8

In brief, our results illustrate the important aspect of sample efficiency of the MULTICONVFiT framework. On top of offering better scores in *high-data* setups, it also allows for reaching strong mID performance relying on smaller amounts of the most ‘precious’ resource: annotated task data (cf. the scores in *low-data* scenarios).

### 4.1 Further Discussion

We now analyze other important aspects of MULTICONVFiT, running a series of side experiments. Due to a large number of experiments and to avoid clutter, we plot results from a representative subset of possible experimental configurations (i.e., encoders, fine-tuning variants, datasets, data setups), but we note that very similar patterns in results have been observed with other configurations.

**Impact of Label Smoothing.** Figure 4 suggests the importance of applying label smoothing, especially in *low-data* scenarios (e.g., drops in $F_1$ scores even up to 4-5 points) and with contrastively tuned encoders, where there is a higher chance of overfitting that leads to classification overconfidence. Switching off label smoothing (i.e., effectively setting $ls = 1.0$) is less severe in *high-data* setups, but our results render it almost universally useful for different MULTICONVFiT model variants.

![Figure 4: Change in $F_1$ performance when no label smoothing is used versus the standard variant with label smoothing ($\Delta F_1$ on the y-axis), with all other parts kept equal. Similar trends are observed with other input models and with C-ADAPT, and also with Acc scores (see Appendix C).](image)

![Figure 5: Impact of contrastive specialization duration (i.e., the number of epochs) on the final mID performance in *low-data* scenarios on BANKING. $F_1$ scores; C-FFT. Very similar trends are observed on the other ID datasets, with C-ADAPT, and with other SEs and LMs.](image)

![Figure 6: Impact of (random) sampling of positive examples (*high-data* scenarios) for contrastive SE specialization on the final mID performance ($F_1$ scores shown). (a) MPNET is the underlying SE, C-FFT; (b) MLM12 is the underlying encoder, C-ADAPT. $x$-axis is in the log-scale for clarity; straight lines of the same color and style refer to respective model configurations without any contrastive specialization.](image)
Training Duration. Figure 5 indicates that the highest gains in mID performance are achieved in the first few epochs of contrastive fine-tuning. There is a large leap already after a single epoch of C-FFT or C-ADAPT, with more gains achieved in subsequent epochs before the procedure starts converging. Figure 5 also illustrates similar learning patterns both for SEs and LMs: the starting gap between SEs and their corresponding LMs does not get mitigated through contrastive specialization, again suggesting the importance of using SEs instead of LMs as the underlying text encoders.

Subsampling Positive Examples. The complexity of contrastive fine-tuning scales quadratically with the number of task-annotated examples, i.e., its complexity is $O(|S|^2)$. Therefore, despite observed gains in high-data setups, the procedure might become prohibitively expensive if the datasets are too large. We investigate if a model variant where (i) we randomly subsample a smaller number of examples from the set $S$ for the creation of sets $PosP$ and $NegP$, while (ii) keeping the full set for the much less expensive part of the model, MLP training, still maintains the benefits stemming from contrastive specialization.

The impact of such random sampling of positive examples is illustrated in Figures 6a (C-FFT) and 6b (C-ADAPT). The plots suggest several findings. 1) As expected, relying on more positive examples yields higher absolute scores, but the large increase in training time does come with diminishing returns in terms of performance (i.e., $F_1$ scores start saturating already with 500-1000 examples. 2) Even a small number of examples (100-200) already yields large benefits over the model variant that does not apply any contrastive specialization, suggesting that it is possible to trade off a fraction of performance for large efficiency benefits. Finally, similar patterns are again observed for C-FFT and for C-ADAPT.

5 Conclusion and Future Work

We proposed MULTI-CONVFiT, a contrastive fine-tuning framework for multi-label intent detection (mID) that transforms general-purpose language models and sentence encoders (SEs) into task-specialized SEs. Such specialized SEs facilitate efficient learning of mID classifiers stacked on top of the fixed sentence encodings (Casanueva et al., 2020). Moreover, we demonstrate how to combine SEs with lightweight adapter modules, resulting in a modular multi-tenant design of the MULTI-CONVFiT framework. We demonstrate effectiveness and robustness of contrastive mID task specialization across a representative set of mID datasets, different input encoders, with large improvements especially in the most demanding low-resource scenarios. We hope that MULTI-CONVFiT will inspire more work on sample-efficient, modular, and highly adaptable multi-label intent detectors.

There are multiple avenues for future research that can further improve various aspects of the proposed MULTI-CONVFiT framework. For instance, in this work, for simplicity and clarity, we rely on globally set fixed threshold values $\theta$, while such thresholds can also be adaptable, with differently calibrated values for different (sets of) intents (Hou et al., 2021). Further, this work relied on a particular class of parameter-efficient fine-tuning methods, bottleneck adapters, as one of the most established methods available. However, as mentioned in §4, future work should also explore other parameter-efficient methods (Pfeiffer et al., 2020a; Ding et al., 2022), aiming to achieve an even better trade-off between performance and parameter-efficiency. In particular, driven by the efficiency requirements, we will investigate parameter-efficient methods that do not increase the model size at all, and thus maintain the same time efficiency at inference, such as methods based on low-rank adaptation (Hu et al., 2022) or sparse fine-tuning (Sung et al., 2021; Ansell et al., 2022).

Acknowledgements

We are grateful to our colleagues at PolyAI for many fruitful discussions and their encouragement to pursue this project. We also thank the anonymous reviewers for their helpful suggestions.

Limitations

We believe there is room for enhancing the underlying contrastive fine-tuning technique. In this work we evaluated only a single contrastive loss, OCL, following the suggestions and empirical analyses from prior work (see §2.1) as well as our preliminary experiments, demonstrating its strong performance. However, other contrastive losses can also be applied within the MULTI-CONVFiT framework. Further, we relied on random sampling of negative examples, as well as random sampling of positive examples in §4.1: we believe that additional performance gains might be achieved through more
sophisticated and semantically guided sampling strategies (Kalantidis et al., 2020; Robinson et al., 2021, inter alia).

This work focused only on multi-label intent detection as a well-defined downstream application, and the methodology was inspired by the desiderata of (efficient) mID. While the proposed methodology is not tied to the mID task and should be equally applicable to other multi-label sentence classification tasks, we did not evaluate the capacity and usefulness of the proposed methods in other tasks as part of this paper.

Finally, we point to the limitations of the current mID datasets and their design which cannot be mitigated solely through improving mID models: current mID datasets provide user utterances without any previous dialog context, and they still fail to distinguish between very subtle meaning differences in more difficult examples (e.g., using a toy example of sentences No, I want the booking and No, I don’t want the booking, both sentences will be annotated with labels deny and booking).

References


A  Models and Data

URLs to the models used in this paper are provided in Table 3.

Our code is based on PyTorch, and relies on the following two widely used repositories:

- sentence-transformers: www.sbert.net
- huggingface.co/transformers/

Publicly available mLID data can be accessed following these links:

- github.com/PolyAI-LDN/task-specific-datasets/tree/master/nlupp (NLU++: BANKING and HOTELS)
- github.com/LooperXX/AGIF/tree/master/data (MIXATIS)

(Due to concerns with privacy and security, the INSURANCEFAQ dataset cannot be publicly released in full.)

B  Examples of Multi-Label Sentences

Some examples from the four multi-label ID datasets in our evaluation (see §3 and Table 1 in the main paper) are provided in Table 4.

C  Additional Results

Additional empirical evidence and analyses, which further support the claims in the main paper, have been relegated to the appendix for clarity and compactness of the presentation in the main paper. These results to a large extent follow the trends observed in the results which are presented in the main paper, or offer additional supporting evidence for the main claims. In summary, we provide the following additional results:

Figure 7 is similar to Figure 2 in the main paper; it shows t-SNE plots that illustrate the effects of contrastive task specialization relying on three related encoders: MPNET-LM, MPNET, and MPNET +C-FFT, and on a subset of intent classes from the mLID dataset HOTELS. Unlike this figure, Figure 2 in the main paper focuses on another set of encoders, and relies on +C-ADAPT contrastive fine-tuning, and plots examples from the BANKING dataset. Both of them demonstrate the desirable effect on the semantic space achieved by contrastive task specialization.

Table 5 shows the exact mLID scores ($F_1$ and Acc) for several C-ADAPT variants on all four mLID datasets in both data setups; see also related Figure 3 in the main paper.

Table 6 demonstrates the impact of duration of contrastive task specialization on mLID Accuracy scores; Figure 5 in the main paper demonstrates the impact on mLID $F_1$ scores.

Figure 8 demonstrates the impact of disabling label smoothing on final mLID Accuracy scores (while Figure 4 in the main paper demonstrates the impact on mLID $F_1$ scores).

Intents per Example. Finally, Figure 9 shows $F_1$ scores over examples with a different number of intents, while a similar plot with Acc scores is provided in Figure 10. Contrastive task specialization leads to pronounced improvements over all groups of examples, especially in low-data setups. Interestingly, while Acc scores are naturally higher for the groups with a fewer number of intents (see Figure 10), the 1-label group displays lower $F_1$ scores than 2-label or 3-label groups on BANKING and INSURANCEFAQ. We attribute it to the modular ontology design (Casanueva et al., 2022); as a consequence, 1-label examples in those mLID datasets are typically very short sentences (e.g., 1-3 word tokens), which are known to pose a challenge for sentence encoders (Chen et al., 2019a).
<table>
<thead>
<tr>
<th>Name</th>
<th>Abbreviation</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpnet-base</td>
<td>MPNET-LM</td>
<td>huggingface.co/microsoft/mpnet-base</td>
</tr>
<tr>
<td>distilroberta-base</td>
<td>DROB-LM</td>
<td>huggingface.co/distilroberta-base</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Sentence</th>
<th>Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>BANKING</td>
<td>I want to apply for a loan, what should I do?</td>
<td>loan, make, request_info</td>
</tr>
<tr>
<td>HOTELS</td>
<td>The pin for my card is not the same as the one for my account, right?</td>
<td>pin, account, card, request_info</td>
</tr>
<tr>
<td>HOTELS</td>
<td>Cancel the restaurant reservation for 18:45 under Jane Doe.</td>
<td>cancel_close, restaurant, booking</td>
</tr>
<tr>
<td>INSURANCEFAQ</td>
<td>I have a reservation and I need to change the number of adults.</td>
<td>change, existing, booking</td>
</tr>
<tr>
<td>INSURANCEFAQ</td>
<td>I’m stuck at the tax identification number.</td>
<td>tax_id, not_working</td>
</tr>
<tr>
<td>MIXATIS</td>
<td>How do I reset my security questions?</td>
<td>how, change, security_question, atis_distance, atis_meal</td>
</tr>
<tr>
<td>MIXATIS</td>
<td>what is the distance between Pittsburgh airport and downtown Pittsburgh?</td>
<td>atis_abbreviation, atis_city</td>
</tr>
<tr>
<td></td>
<td>and what are my meal options from Boston to Denver?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>and show me the cities served by nationalair</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Encoder / Epoch</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPNET-LM</td>
<td>23.9</td>
<td>33.3</td>
<td>36.1</td>
<td>35.8</td>
<td>38.0</td>
<td>40.0</td>
<td>41.1</td>
<td>39.9</td>
<td>40.3</td>
<td>40.3</td>
<td>40.5</td>
</tr>
<tr>
<td>MPNET</td>
<td>30.0</td>
<td>42.7</td>
<td>46.5</td>
<td>47.0</td>
<td>47.4</td>
<td>47.9</td>
<td>48.0</td>
<td>48.3</td>
<td>48.6</td>
<td>49.1</td>
<td>49.1</td>
</tr>
<tr>
<td>DROB - C-ADAPT</td>
<td>31.0</td>
<td>39.2</td>
<td>44.5</td>
<td>45.8</td>
<td>45.9</td>
<td>46.7</td>
<td>47.0</td>
<td>47.5</td>
<td>47.1</td>
<td>47.9</td>
<td>47.6</td>
</tr>
</tbody>
</table>

Figure 7: t-SNE plots (van der Maaten and Hinton, 2012) of encoded utterances from the mID dataset HOTELS (see §3) associated with a subset of intent classes, demonstrating the effects of contrastive task specialization of the input encoder with mID data. **Left:** sentence encodings with the original MPNet LM (Song et al., 2020); **Middle:** encodings with MPNet transformed into a universal SE (Reimers and Gurevych, 2019); **Right:** encodings with a task-specialized SE obtained after contrastively fine-tuning the universal MPNet-based SE.
Figure 8: Change in Acc performance when no label smoothing is used, with all other parts kept equal. Similar trends are observed with other input models and with C-ADAPT.

Figure 9: $F_1$ scores over examples with a particular number of intents; (a) low-data, (b) high-data.

Figure 10: Acc scores over examples with a particular number of intents; (a) low-data, (b) high-data.