Knowledge Distillation Meets Few-Shot Learning: An Approach for Few-Shot Intent Classification Within and Across Domains

Anna Sauer
University of Stuttgart
anna.sauer@ims.uni-stuttgart.de

Shima Asaadi and **Fabian Küch**
Fraunhofer IIS
{shima.asaadi,fabian.kuech}@iis.fraunhofer.de

Abstract

Large Transformer-based natural language understanding models have achieved state-of-the-art performance in dialogue systems. However, scarce labeled data for training, the large model size, and low inference speed hinder their deployment in low-resource scenarios. Few-shot learning and knowledge distillation techniques have been introduced to reduce the need for labeled data and computational resources, respectively. However, these techniques are incompatible because few-shot learning trains models using few data, whereas, knowledge distillation requires sufficient data to train smaller, yet competitive models that run on limited computational resources. In this paper, we address the problem of distilling generalizable small models under the few-shot setting for the intent classification task. Considering in-domain and cross-domain few-shot learning scenarios, we introduce an approach for distilling small models that generalize to new intent classes and domains using only a handful of labeled examples. We conduct experiments on public intent classification benchmarks, and observe a slight performance gap between small models and large models. Overall, our results in both few-shot scenarios confirm the generalization ability of the small distilled models while having lower computational costs.

1 Introduction

Transformer-based language models, such as BERT (Devlin et al., 2019), contribute widely to the development of dialogue systems. A key component in the development of these systems is natural language understanding (NLU), such as intent classification (IC). Intent classification refers to determining the intent of the speaker's utterance in a given domain in dialogue systems. Recently, BERT-based language models have achieved state-of-the-art performance in intent classification through fine-tuning on task-specific datasets (Chen et al., 2019). However, there are two main challenges in the development of BERT-based intent classification models for task-oriented dialogue systems. First, training such models across many domains needs labeled training data from multiple domains. Due to the lack of large amounts of multi-domain training data, few-shot learning (FSL) methods, such as metric-based meta-learning techniques (Vinyals et al., 2016; Snell et al., 2017), have been used to adapt BERT-based intent classification models to new domains (Li et al., 2021). In cross-domain few-shot learning methods, the model learns transferable knowledge from large-scale source domain data and generalizes to unseen target domains using only a handful of training samples.

The second challenge is the large model size and long inference time of Transformer-based models, which hinder the deployment of such models when limited computational resources are available. Approaches to reduce the size of models, e.g., knowledge distillation (KD; Hinton et al. 2015), have been introduced. It has been shown that the new compressed models retain a high percentage of the performance while having a shorter inference time than the original models (Liu et al., 2019). Task-specific knowledge distillation approaches require sufficiently large training datasets (Tang et al., 2019), ideally with labels (Hinton et al., 2015), to distill a powerful small model. However, to obtain both generalized and small models, knowledge distillation methods seem to be incompatible with few-shot learning due to the large need of sufficient training data. Therefore, an adaptation of knowledge distillation to few-shot learning is necessary. To the best of our knowledge, task-specific knowledge distillation in cross-domain few-shot learning has largely remained unexplored with a few exceptions in computer vision (Zhang et al., 2020b; Li et al., 2020) and natural language processing (NLP; Pan et al. 2021; Zhou et al. 2021).

In this paper, we propose a task-specific approach for distilling small models with generaliza-

tion ability to new classes and domains in two few-shot learning scenarios: 1) in-domain target class generalization in single- and multi-domain intent classification; 2) target domain adaptation in multi-domain intent classification. To this end, we first pretrain a Transformer-based prototypical teacher network (Snell et al., 2017) on source classes and domains using meta-learning. Then, we design a prototypical student network and pass the transferable knowledge to the student using knowledge distillation. During the distillation process, we consider a prototype loss as a new component in the standard distillation loss function. This loss measures how much each prototype that is produced by the student model resembles the respective prototype produced by the teacher model. Moreover, as opposed to standard batch training in knowledge distillation, we introduce an episodic distillation process. This way, we obtain a small student model that is compatible with few-shot scenarios and generalizes to unseen target classes and domains.

Our contributions are summarized as follows: 1) We propose a new knowledge distillation approach compatible with few-shot learning by introducing an episodic distillation process and using the prototype-based distillation loss. Our novel approach combines advantages of few-shot learning with knowledge distillation. 2) We perform extensive experiments on four public NLU benchmarks and compare the distilled small model with the large model in the few-shot intent classification scenario. Results show a slight performance drop for the small model while having lower memory consumption and a slightly faster inference speed. 3) We show that the small model can effectively generalize and adapt to target domains without the teacher supervision in the few-shot target domain adaptation. This is a more challenging and realistic scenario for small student models.

2 Background and Related Work

2.1 Few-shot learning

Few-shot learning has received substantial interest in NLP. One prominent technique in FSL is meta-learning, such as metric-based meta-learning techniques (Vinyals et al., 2016; Snell et al., 2017). In these techniques, a model is trained on source training tasks with sufficient labeled instances, called meta-training, and generalizes or adapts to new tasks with only a handful of labeled examples, called meta-testing. The meta-training step is performed through episodes. In each episode, a set of N classes (N-way) is chosen per task. For each class, a support set, which contains K labeled examples, and a query set are created for training and evaluating the performance of the classifier for updating the model parameters. The learning process is performed in the form of N-way K-shot classification task. During meta-testing, an adaptation to new tasks using a few labeled examples is performed similarly to meta-training.

Recent attempts in few-shot intent classification focus on both in-domain and cross-domain generalization using different meta-learning techniques. Some approaches introduce metric-based meta-learning, such as Prototypical networks (Snell et al., 2017) to train models on large-scale source class or domain data and generalize to emerging classes or domains using only a handful of training samples (Geng et al., 2019; Nguyen et al., 2020; Krone et al., 2020; Li et al., 2021). In metric-based methods, a metric function is trained to classify new examples by comparing them with labeled examples. Other approaches propose to pretrain models on different source tasks and transfer them to the few-shot intent detection task (Casanueva et al., 2020; Zhang et al., 2020a). Alternatively, Xia et al. (2020) propose a novel model to augment training data by generating utterances for unseen intent class labels.

2.2 Knowledge distillation

Knowledge distillation approaches transfer the knowledge and generalization ability of a large trained model, called teacher, to a small model, called student (Ba and Caruana, 2014; Hinton et al., 2015). In the simplest case, the objective function during distillation is to minimize the difference between the soft labels produced by the teacher and the student predictions. As an alternative, the logits, i.e., the inputs to the final softmax function, can be used instead of the soft labels for training the student (Bucila et al., 2006). Hinton et al. (2015) The teacher and student models can have different architectures. For instance, Liu et al. (2019) explore Transformer-based teacher and both Transformer- and LSTM-based student models for multi-task knowledge distillation in NLP. KD has received special attention in Transformer-based teacher models to train light-weight generic students (Sanh et al., 2019; Sun et al., 2019; Jiao et al., 2020; Sun et al., 2020; Wu et al., 2020) and task-specific stu-

dents with practical applications (Tsai et al., 2019; Liu et al., 2019; Clark et al., 2019) including intent classification (Jiang et al., 2021).

2.3 Knowledge distillation and few-shot learning

In NLP models, knowledge distillation for improving the overall efficiency and generalization ability to new classes and domains is not straightforward under the few-shot learning scenario. Recent investigations suggest that larger models show a better few-shot performance than smaller models because of higher model capacity (Brown et al., 2020).[1] At the same time, knowledge distillation needs sufficiently large training data, ideally with labels (Hinton et al., 2015), to distill a small model with small performance gap. Thus, employing few-shot learning and knowledge distillation methods jointly seems to be conflicting.

There have been only a few attempts to apply knowledge distillation in the context of the few-shot learning scenario in computer vision (Zhang et al., 2020b; Li et al., 2020; Liu et al., 2020). To the best of our knowledge, attempts in NLP are restricted to the work by Pan et al. (2021) and Zhou et al. (2021). In their work, Pan et al. (2021) train a multi-domain Transformer-based meta-teacher and introduce a meta-distillation approach to obtain domain-specific student models. Similar to our work, they consider in-domain generalization and target domain adaptation scenarios during the distillation process. However, we focus on a more challenging scenario where the student model does not have access to the teacher for emerging domains. That is, the student adapts to new target domains using a handful of labeled examples independently and without any distillation process. Thus, our model architecture is different from that of Pan et al. (2021) to preserve the model capacity for generalization and adaptation purposes. Zhou et al. (2021) propose a meta-learning approach for knowledge distillation in which both teacher and student are trained through interacting with each other. The teacher learns to improve its transfer ability by receiving feedback about the performance of the student on a new data split called quiz set. Alternative approaches to KD in a low-resource setting consider data augmentation to generate unlabeled data and distill small models using the augmented data (Melas-Kyriazi et al., 2019).

[1] Although there has recently been a discussion around this assumption (Schick and Schütze, 2021).

3 Approach

We first describe the teacher and student model architectures, followed by our proposed model training procedure. We elaborate on details of the proposed episodic distillation process and show how our approach preserves the generalization ability of the distilled models under few-shot learning scenarios.

3.1 Model architecture

Since we consider the few-shot learning scenario, both teacher and student models are designed as a prototypical network (Protonet; Snell et al. 2017), which is a metric-based meta-learning approach.

A teacher Protonet $\mathcal{T}$ with trainable parameters $\theta_{\mathcal{T}}$ is composed of an encoding block, which is a Transformer with L layers ($L >= 2$), followed by two linear hidden layers. The objective of the network is to learn a metric space by training model parameters $\theta_{\mathcal{T}}$. The input to the teacher is a sequence $x = t_1 \dots t_k$ with k tokens. The fixed-length encoded sequence is the mean pooling of the token embeddings from the output of the last layer of the Transformer $e(x) = \frac{1}{k}\sum_{i=1}^{k} h^L(t_i)$. Then, $e(x)$ serves as the input to the hidden layers and the output is an m-dimensional sequence representation. Given C classes, $\mathcal{T}$ computes m-dimensional class representations $r_c \in \mathbb{R}^m$ for $c \in \{1, \dots, C\}$, called prototype, as the mean aggregation of the m-dimensional representations of support instances in the respective class. For each new sequence, a classification is performed by computing the Euclidean distance between the class prototypes and the created m-dimensional sequence representation.

The student Protonet $\mathcal{S}$ with trainable parameters $\theta_{\mathcal{S}}$ consists of a Transformer with two layers in the encoding block, followed by two linear hidden layers. The Transformer layers are initialized from the first two layers of the teacher's encoding block. Class prototypes are computed in the same way as the teacher. In both architectures, all model parameters are trainable and shared across all domains in multi-domain intent classification.

3.2 Model training and testing

Inspired by meta-learning, we implement meta-training and meta-testing steps. Given two few-shot scenarios in our work, we adjust these steps accordingly. The first scenario is in-domain tar-

get class generalization and the second scenario is target domain adaptation in multi-domain classification. Due to the joint FSL and KD approach, meta-training consists of two steps: 1) teacher pretraining on source classes (domains), referred to as *episodic pretraining*, 2) student pretraining on source classes (domains) using the proposed episodic knowledge distillation, referred to as *episodic distillation*. At meta-testing, we implement an additional target domain adaptation step for the second scenario, called *Mini-episodic adaption*. In the following, we explain the details of (mini-)episode construction and the training steps.

3.2.1 Episode construction

Assume there are disjoint sets of source classes C_{train} and target classes C_{test} for meta-training and meta-testing, respectively. These sets belong to source and target domains splits, D_{train} and D_{test}. In the in-domain target class generalization scenario, $D_{train} = D_{test}$. To construct an episode, a domain d is uniformly chosen from domains D_{split} where $split$ is either $train$ or $test$. Then, we create variable size episodes by sampling the number of ways n, support shot k_s, and query shot k_q from the selected domain d, following the work by Krone et al. (2020) and Triantafillou et al. (2020). Then the support set S_c and the query set Q_c for each class c are sampled from the domain splits. As discussed in Krone et al. (2020)'s work, by setting variable shots and ways per episode, our approach is more compatible with real-world cases where unbalanced classes are available in the datasets. Please refer to Appendix A.1 for the details of episode construction. Meta-training consists of epochs and each epoch contains distinct episodes. Therefore, in line with Krone et al. (2020), once an episode is constructed, we remove the respective samples from the meta-training split until all samples are seen in an epoch.

3.2.2 Episodic pretraining

To pretrain a teacher $\mathcal{T}$ on source classes (domains), we implement the standard meta-learning approach. At each step, an episode is created through the described variable episode construction approach. Then, class prototypes $r_c \in \mathbb{R}^m$ are computed utilizing the labeled support set of each class S_c:

$$r_c = \frac{1}{|S_c|} \sum_{(x_i, y_i) \in S_c} \mathcal{T}_\theta(x_i). \tag{1}$$

Next, the model computes the negative of the squared Euclidean distance between each query example representation and the class prototypes, denoted as *logits*. Finally, we use the cross-entropy loss between the computed logits of the query set and query labels y as the classification loss:

$$\mathcal{L}_{cls} = \sum_{c \in C_{train}} \sum_{i=1}^{|Q_c|} \text{cross-entropy}(logits_i, y_i), \tag{2}$$

and update the model parameters $\theta_\mathcal{T}$ using the Adam optimizer.

3.2.3 Episodic distillation process

Our goal is to obtain efficient small student models that generalize to unseen classes (domains). Therefore, we combine the advantages of FSL and KD and introduce episodic knowledge distillation as the main component in our approach. It is performed during the pretraining step of the student on source classes (domains).

Given a pretrained teacher $\mathcal{T}$ on source classes (domains), we distill a student $\mathcal{S}$ on the same classes (domains). The distillation process consists of epochs. At each distillation step in an epoch, we create an episode. The support set is used to compute the class prototypes in both $\mathcal{T}$ and $\mathcal{S}$ for the classification of the query set. We define the overall distillation loss function as follows:

$$\mathcal{L}_{\text{kd}} = \mathcal{L}_{\text{soft}} + \mathcal{L}_{\text{pt}}, \tag{3}$$

where $\mathcal{L}_{soft}$ is the Kullback-Leibler (KL) divergence between the soft labels of the teacher and student output layer on the query set, which is computed as follows:

$$\begin{aligned} p_\mathcal{T} &= \text{softmax}(\text{logits}^\mathcal{T}) \\ p_\mathcal{S} &= \text{softmax}(\text{logits}^\mathcal{S}) \\ \mathcal{L}_{\text{soft}}(\mathcal{T}, \mathcal{S}) &= \text{KL}(p_\mathcal{T}, p_\mathcal{S}). \end{aligned} \tag{4}$$

To transfer the generalization ability of the teacher, we use a new term $\mathcal{L}_{pt}$ in the distillation loss function, which is specific to the few-shot learning setting. $\mathcal{L}_{pt}$ computes the difference between the class prototypes in $\mathcal{T}$ and $\mathcal{S}$. It is computed as the Mean-Squared Error (MSE) on the class prototypes in the teacher and student:

$$\mathcal{L}_{pt}(\mathcal{T}, \mathcal{S}) = \sum_{c=1}^{C_{train}} \text{MSE}(r_c^\mathcal{T}, r_c^\mathcal{S}). \tag{5}$$

After computing the loss, the student model parameters θ_S are updated. Note that the student does not have access to the query set labels during distillation.

3.2.4 Mini-episodic adaptation

Both pretrained teacher and student models can be adapted on target domains in multi-domain IC. However, our assumption is that the models have access to only a handful of labeled examples. For this purpose, similar to the episodic pretraining step, the standard meta-learning approach is applied for adapting the teacher or student on the target domain. To simulate the few-shot assumption, we create mini-episodes from episodes, devised originally by Luo et al. (2017). At each adaptation step, akin to the k-fold cross-validation approach, the k_s instances in the support set of the created episode are repeatedly split into n-way $k_s - 1$ mini-support instances and one mini-query instance. Model parameters are updated after each mini-episode. The episode's query set is left for evaluation purposes at inference time. We adapt the teacher using mini-episodic adaptation. The student is also adapted using the same procedure without the teacher supervision.

3.2.5 Meta-testing

Model performance is evaluated at meta-testing time through random test episodes on the meta-testing split C_{test} following the experimental setup in (Krone et al., 2020; Li et al., 2021). In the first scenario, we use the support and query sets at each random test episode for prototype computations and performance evaluation, respectively. In the second scenario, we adapt the model to target domains using mini-episodic adaptation and use the mini-support and mini-query sets for model parameters update. Then, the episode's support set is used for prototype computations while the query set is used for performance evaluation. If the model is evaluated on target domains without any adaptations, we use the support and query sets for prototype computations and performance evaluation.

4 Experiments

We conduct extensive experiments to evaluate the proposed approach on public intent classification datasets. We simulate two scenarios: in-domain target class generalization and the more challenging scenario, target domain adaptation in multi-domain intent classification. Experiments have been implemented in PyTorch and performed on a single NVIDIA 8GB GPU in Ubuntu 16.04.6 LTS.

4.1 Experiment setup

4.1.1 Datasets and splits

We use four public NLU benchmarks in our experiments: SNIPS (Coucke et al., 2018), ATIS (Hemphill et al., 1990), TOP (Gupta et al., 2018), and Clinc150 (Larson et al., 2019). To simulate few-shot class generalization in intent classification, we use the proposed splits by Krone et al. (2020). They create a meta-training split (train split) and a meta-testing split (test split) from the classes in each dataset. To simulate few-shot domain adaptation, we use the proposed splits by Li et al. (2021). The statistics on the datasets and splits for both scenarios are provided in Tables 7 and 8 in Appendix B.1, respectively. Intent classes of each split can be found in (Krone et al., 2020). We only remove the *atis_day_name* intent from the test split of ATIS as it contains only two utterances. Moreover, we use *Work*, *Banking*, and *Credit card* domains as the source domains and *Home* and *Kitchen and Dining* as the target domains in the Clinc150 dataset for the second scenario. We choose this split to minimize the overlap between the source and target domains. Moreover, we do not utilize any validation set for model parameters optimization. In this way, we increased the difficulty level for a meaningful comparison in few-shot scenarios. Furthermore, SNIPS is in fact a multi-domain dataset and contains cross-domain intent classes, and ATIS and TOP are highly unbalanced resulting in rather difficult datasets for comparison in the few-shot setting. TOP also contains various intent classes in the *navigation and events* domain.

4.1.2 Training and testing Settings

In all scenarios, we use the Adam optimizer during pretraining and distillation with a learning rate of $1e^{-5}$. Following the experiments setup in (Krone et al., 2020) and (Li et al., 2021), training epochs for both teacher and student are set to 30. At test time, we report the average accuracy and standard deviation of the models over three random seeds and 100 random test episodes on the test split. We use $\text{BERT}_{\text{base_uncased}}$ as the base language model with hidden size of 768. All hidden layers and output features m in the Protonet are set to 200 based on practical experiments.

In the in-domain target class generalization, we pretrain the teacher using episodic training with two different maximum support set size (K_{max}) for episodes: 20 and 100. This way, we compare our results directly with the results of Krone et al. (2020). In the target domain adaptation scenario, we report the results of the distilled student on target domains without adaptation and with 10 epochs of mini-episodic adaptation. In line with Li et al. (2021)'s work, the ways n is set to the number of the intent classes in the target domain during the target domain adaptation of the student. Moreover, we fix both k_s and k_q to 10 during the adaptation. Therefore, variable episode construction is not utilized in this step. Pretraining and episodic distillation steps remain the same as before.

4.2 Results and discussions

4.2.1 In-domain target class generalization

We investigate the generalization ability of the small student model on unseen classes and compare with the proposed models in (Krone et al., 2020). They study different encoding blocks (GloVe, ELMo, BERT) and algorithms (Fine-tune, foMAML, Proto) for joint IC and slot filling under the few-shot learning scenario. We report the results of the BERT+Proto model (Baseline BERT+Proto), which is the $\text{BERT}_{\text{base_uncased}}$ model with a Protonet, and the best results obtained among all models (Baseline best result). Note that the reported Baseline BERT+Proto model is approximately as large as the teacher model in the number of parameters. Table 1 shows the evaluation results on the three benchmark datasets, considering two different values of K_{max}. For each domain dataset, we train a teacher model on the train split via the episodic pretraining step, and distill an in-domain student using the episodic distillation process. We then evaluate the performance of the student on the unseen intent classes, i.e., the test split, in the respective dataset without further adaptation. Moreover, following the experiments in (Krone et al., 2020), we train a multi-domain teacher using the train splits of all datasets jointly. We then evaluate two types of distilled students on the test split of each dataset individually: 1) a multi-domain student distilled on all datasets, and 2) a domain-specific student. Table 2 shows the results of the multi-domain intent class generalization.

As can be computed from Table 1, the domain-specific student retains 95.7% of the domain-specific teacher's performance on average, which confirms its generalization ability given the limited capacity of small models. The student outperforms the Baseline BERT+Proto model by 5.6 points in $K_{max} = 20$ and 1.75 points in $K_{max} = 100$ on average. Note that Krone et al. (2020) proposed a joint few-shot learning approach for IC and slot filling tasks, which results in a more challenging final task. Therefore, for fairness, we refrain from comparing our teacher results with their models. The performance boost by larger K_{max} in the student is 2.4 points. Since there is a semantic overlap between the train and test intent classes in ATIS, the student shows competitive performance with the teacher. SNIPS contains semantically distant classes. Similarly, TOP contains diverse intent classes besides being highly unbalanced, which explains the performance gap between the student and the teacher in these datasets.

Table 2 shows that the multi-domain and domain-specific students distilled from the multi-domain teacher, achieve 82.31% and 92.06% of the teacher performance, respectively. As is expected, the multi-domain student underperforms the domain-specific student by 7.35 accuracy points on average since its representational capacity is limited for several domains. However, the multi-domain student outperforms the Baseline BERT+Proto in the ATIS domain. This demonstrates that multi-domain training is beneficial when the test set is highly imbalanced, like the ATIS dataset. Compared to the Baseline BERT+Proto, the domain-specific student achieves a higher performance in four out of six experiments and falls behind in the other two experiments by 1.79 points on average. Therefore, there is a trade-off between less memory consumption by deploying a multi-domain small model and a higher accuracy performance by deploying several distinct domain-specific models in an application. Slight improvements with $K_{max} = 100$ can be observed in our model.

4.2.2 Target domain adaptation

In this experiment, a multi-domain teacher is pretrained on source domains (pretrained $\mathcal{T}$) and a student is distilled on source domains using the episodic knowledge distillation (pretrained $\mathcal{S}$). To evaluate the generalization ability of the student on unseen domains, we adapt the student to a target domain without teacher access (adapted $\mathcal{S}$) using mini-episodic adaptation. We compare its performance with the teacher adapted to the respective

Model	$K_{max} = 20$			$K_{max} = 100$		
	SNIPS	**ATIS**	**TOP**	**SNIPS**	**ATIS**	**TOP**
Baseline best result	85.53 ± 0.35	65.95 ± 2.29	52.76 ± 2.26	87.69 ± 1.05	70.25 ± 0.39	61.30 ± 0.32
Baseline BERT+Proto	81.39 ± 1.85	58.84 ± 1.33	52.76 ± 2.26	83.51 ± 0.88	66.89 ± 2.31	61.30 ± 0.32
Domain-specific teacher	87.66 ± 1.69	69.44 ± 1.21	63.08 ± 1.80	87.58 ± 1.95	72.98 ± 2.29	64.65 ± 2.74
Domain-specific student	82.83 ± 0.92	69.45 ± 3.21	57.49 ± 3.17	84.19 ± 0.83	71.57 ± 2.98	61.21 ± 1.46

Table 1: Average test accuracy on in-domain target class generalization. Models are trained and tested on each domain (dataset) separately.

Model	$K_{max} = 20$			$K_{max} = 100$		
	SNIPS	**ATIS**	**TOP**	**SNIPS**	**ATIS**	**TOP**
Baseline best result	87.64 ± 0.73	65.19 ± 1.29	52.64 ± 2.58	88.90 ± 0.18	71.89 ± 1.45	62.51 ± 1.79
Baseline BERT+Proto	81.44 ± 2.91	58.82 ± 1.55	52.64 ± 2.58	86.29 ± 1.09	65.70 ± 2.31	62.51 ± 1.79
Multi-domain teacher	87.74 ± 0.48	79.65 ± 6.27	62.83 ± 2.00	86.91 ± 3.06	83.77 ± 0.89	65.72 ± 0.77
Multi-domain student	72.97 ± 0.62	72.03 ± 2.07	45.36 ± 0.94	75.57 ± 0.82	68.90 ± 2.02	51.82 ± 0.76
Domain-specific student	85.74 ± 0.49	72.08 ± 3.16	56.37 ± 3.60	85.58 ± 0.73	71.36 ± 2.16	59.64 ± 3.64

Table 2: Average test accuracy on in-domain target class generalization. Multi-domain models are trained on all three datasets and tested on each dataset separately.

domain (adapted $\mathcal{T}$) using mini-episodic adaptation. Train and test splits are reported in Table 8 in Appendix B.1.

Table 3 shows the average results on three target domains. We also report the results of two cross-domain models proposed by Li et al. (2021), referred to as Base Protonet and Base best. The Base Protonet utilizes BERT as the encoding block, which is approximately in the same size as our teacher model. The Base best is the best results obtained among different models. As can be seen, the adapted student without teacher supervision shows a significant improvement over its pretrained counterpart. It also achieves 95% of the adapted teacher's performance and even outperforms it on SNIPS slightly. Moreover, the adapted student outperforms the large baselines by 7.03 points on average. This leads to a conclusion that our proposed approach brings benefits in the few-shot generalization problem on small distilled models with limited representational capacity. Note that Li et al. (2021) proposed a joint meta-learning approach for cross-domain IC and slot filling, which results in a more challenging final task. Therefore, for fairness, we refrain from comparing our teacher results with their models.

We extend the experiments with the Clinc150 dataset, which is a balanced dataset. Table 4 presents evaluation results for the Clinc150 target domain split. Following the same discussion, the pretrained teacher outperforms the pretrained student. The adapted student achieves higher accuracy than its pretrained counterpart and retains 87% of the adapted teacher, which is slightly lower than the previously studied domains. We argue that it is due to the more challenging target domains with larger number of intent types (15 intents per domain) and highly overlapping intents (e.g., *todo_list* and *todo_list_update*, *restaurant_suggestion* and *restaurant_review*), which should be handled by a single student in the Clinc150 dataset. This limits the application of our approach in such multi-domain settings.

Model	SNIPS	ATIS	TOP
Base best	90.9 ± 0.3	76.0 ± 0.8	61.9 ± 1.1
Base Protonet	90.9 ± 0.3	75.3 ± 0.7	61.9 ± 1.1
Pretrained $\mathcal{T}$	79.11 ± 1.68	82.20 ± 1.56	62.97 ± 1.91
Pretrained $\mathcal{S}$	75.24 ± 3.02	76.56 ± 2.28	57.16 ± 0.73
Adapted $\mathcal{T}$	89.90 ± 0.13	94.70 ± 0.33	76.12 ± 0.90
Adapted $\mathcal{S}$	90.41 ± 0.89	92.36 ± 0.73	66.78 ± 0.97

Table 3: Average test accuracy on target domain adaptation in SNIPS, ATIS, and TOP

To compare the computational cost of the teacher and student, we report the memory size and average inference time of the models per episode on target domains. The number of parameters (in millions) for teacher and student is 109.68M and 38.80M. The student consumes 64% less memory (2.8 times fewer parameters) than the teacher. The average inference speed of the student for one episode in-

Model	Home	Kitchen_dining
Pretrained $\mathcal{T}$	78.08 ± 0.70	79.39 ± 0.40
Pretrained $\mathcal{S}$	63.74 ± 0.86	69.76 ± 0.94
Adapted $\mathcal{T}$	91.91 ± 0.07	91.44 ± 0.44
Adapted $\mathcal{S}$	76.60 ± 0.29	82.17 ± 0.78

Table 4: Average test accuracy on Clinc150 target domains

Source\Target	SNIPS	ATIS	TOP
SNIPS	–	58.42 ± 3.65 89.45 ± 1.99	40.50 ± 0.71 62.58 ± 0.87
ATIS	75.92 ± 2.78 89.51 ± 1.17	–	53.88 ± 0.39 66.93 ± 0.54
TOP	65.59 ± 3.18 82.46 ± 1.18	70.08 ± 1.90 91.42 ± 0.28	–

Table 5: Average test accuracy of models trained on one source domain. Upper rows report pretrained student and lower rows report adapted student.

cluding prototype computations and query set predictions on the Clinc150 target domains (Home and Kitchen) is 5.6 and 1.1 times faster than teacher on CPU and GPU, respectively.[2]

4.2.3 Ablation Study

We analyze the impact of source domains on the performance of the student model on target domains in the target domain adaptation scenario. For this purpose, we pretrain the teacher and student on one source domain and evaluate the pretrained and adapted student on the two other target domains individually and compare with the results in Table 3. Table 5 shows the results of the pretrained and adapted student in each target domain. We observe a performance gap between one versus multiple source domains in pretrained students, specially when we opt out the source ATIS; The performance of the pretrained student on TOP is 40.50 with source SNIPS and 57.16 with source ATIS and SNIPS. This demonstrates that the pretrained student takes an advantage of diverse source domains for evaluation on target domains. Moreover, the average higher performance of the student in the multiple source domain setting indicates that the knowledge is transferred effectively through the episodic distillation process. Small performance gap between one versus multiple source domains is also observed in the adapted student.

Lastly, we analyze how FSL and KD influence the IC performance separately. For this, we measure the performance of the teacher and the distilled student, which are pretrained on the Clinc150 source domains and tested on the Clinc150-Home target domain without adaptation. We test these models with support shot $k_s = 10$ and $k_s = 70$, called FSL and many-shot learning (MSL) scenario, respectively. We use the first 10 and 70 instances of each class in the official train set of the Home domain as the support set. The official test set with 30 instances per class is also used as the query set. Evaluation results are shown in Table 6. We observe an accuracy drop from teacher to student in both scenarios (15.19 and 16.60 points), however, with a negligible difference . Therefore, the distilled student loses approximately the same amount of teacher's performance accuracy in few- and many-shot learning settings. This indicates the effectiveness of the proposed episodic distillation process in knowledge transfer under the FSL setting. Moreover, the difference in the performance loss from MSL to FSL in both teacher and student models is small ($9.63 - 8.22 = 1.41$ point). This implies the capability of the proposed approach for obtaining generalizable small models. Note that the discrepancy between the performance results in this section and previous section is due to the different support and query splits at meta-testing.

	Teacher	Student	Student - Teacher
MSL	77.78 ± 0.59	62.59 ± 0.92	-15.19
FSL	$69.56 \pm 2,94$	52.96 ± 2.44	-16.60
FSL - MSL	-8.22	-9.63	–

Table 6: Average test accuracy on the effect of FSL and KD on Clin150-Home

5 Conclusion

We address the nontrivial merging problem of meta-learning and knowledge distillation. Our proposed approach distills large Transformer-based models into smaller student models, which are compatible with few-shot learning scenarios in intent classification. Through a multi-step meta-training with an episodic knowledge distillation, we obtain a small distilled model that is generalizable and adaptable to new classes and domains using only a few labeled examples. Our results in target domain adaptation show that the small model can adapt effectively to new domains without teacher supervision.

[2]The CPU is a 3.1 GHz Quad-Core Intel Core i7.

This removes the need for a large teacher when time and computational resources are limited. Compared to the large model, we observe a slight performance loss and less memory consumption in the distilled model. In summary, our results provide insights into the advantages and limitations of a joint few-shot learning and knowledge distillation approach to foster future research in this area.

Our primary findings suggest that it is worthwhile to explore different FSL techniques jointly with KD for cross-domain few-shot performance improvements. Overall, this topic still merits more attention to aid the practical deployment of NLU models in dialogue systems under low-resource scenarios. As future research, we will study novel joint methods for the cross-domain generalization problem under low-resource scenarios. Moreover, we will investigate the methods in joint NLU tasks, specifically slot filling and IC.

Acknowledgements

This research was carried out while the first author was affiliated with the Fraunhofer IIS. This work is supported by the German Federal Ministry for Economic Affairs and Energy (BMWi) through the SPEAKER project (FKZ 01MK19011).

References

Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Cristian Bucila, R. Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *KDD'06*.

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

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. BERT for joint intent classification and slot filling. *CoRR*, abs/1902.10909.

Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D. Manning, and Quoc V. Le. 2019. BAM! born-again multi-task networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5931–5937, Florence, Italy. Association for Computational Linguistics.

A. Coucke, A. Saade, Adrien Ball, Théodore Bluche, A. Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, F. Caltagirone, Thibaut Lavril, Maël Primet, and J. Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *ArXiv*, abs/1805.10190.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3904–3913, Hong Kong, China. Association for Computational Linguistics.

Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic parsing for task oriented dialog using hierarchical representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2787–2792, Brussels, Belgium. Association for Computational Linguistics.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*, Hidden Valley, Pennsylvania.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Yidi Jiang, Bidisha Sharma, Maulik Madhavi, and Haizhou Li. 2021. Knowledge distillation from bert transformer to speech transformer for intent classification.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Jason Krone, Yi Zhang, and Mona Diab. 2020. Learning to classify intents and slot labels given a handful of examples. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 96–108, Online. Association for Computational Linguistics.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Shang-Wen Li, Jason Krone, Shuyan Dong, Yufu Zhang, and Y. Al-Onaizan. 2021. Meta learning to classify intent and slot labels with noisy few shot examples. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 1004–1011.

Tianhong Li, Jianguo Li, Zhuang Liu, and Changshui Zhang. 2020. Few sample knowledge distillation for efficient network compression. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14627–14635.

Benlin Liu, Yongming Rao, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. 2020. Metadistiller: Network self-boosting via meta-learned top-down distillation. In *Computer Vision – ECCV 2020*, pages 694–709, Cham. Springer International Publishing.

L. Liu, Haiquan Wang, Jimmy J. Lin, R. Socher, and Caiming Xiong. 2019. Mkd: a multi-task knowledge distillation approach for pretrained language models.

Zelun Luo, Yuliang Zou, Judy Hoffman, and Li Fei-Fei. 2017. Label efficient learning of transferable representations across domains and tasks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 164–176, Red Hook, NY, USA. Curran Associates Inc.

Luke Melas-Kyriazi, George Han, and Celine Liang. 2019. Generation-distillation for efficient natural language understanding in low-data settings. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 124–131, Hong Kong, China. Association for Computational Linguistics.

Hoang Nguyen, Chenwei Zhang, Congying Xia, and Philip Yu. 2020. Dynamic semantic matching and aggregation network for few-shot intent detection. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1209–1218, Online. Association for Computational Linguistics.

Haojie Pan, Chengyu Wang, Minghui Qiu, Yichang Zhang, Yaliang Li, and Jun Huang. 2021. Meta-KD: A meta knowledge distillation framework for language model compression across domains. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3026–3036, Online. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS 2019*, Vancouver.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2339–2352. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for BERT model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4323–4332, Hong Kong, China. Association for Computational Linguistics.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. MobileBERT: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170, Online. Association for Computational Linguistics.

Raphael Tang, Yao Lu, L. Liu, Lili Mou, Olga Vechtomova, and Jimmy J. Lin. 2019. Distilling task-specific knowledge from bert into simple neural networks. *ArXiv*, abs/1903.12136.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. 2020. Meta-dataset: A dataset of datasets for learning to learn from few examples.

In *International Conference on Learning Representations (ICLR 2020)*, Addis Ababa, Ethiopia. ICLR.

Henry Tsai, Jason Riesa, Melvin Johnson, Naveen Arivazhagan, Xin Li, and Amelia Archer. 2019. Small and practical BERT models for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3632–3636, Hong Kong, China. Association for Computational Linguistics.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3637–3645, Red Hook, NY, USA. Curran Associates Inc.

Bowen Wu, Huan Zhang, MengYuan Li, Zongsheng Wang, Qihang Feng, Junhong Huang, and Baoxun Wang. 2020. Towards non-task-specific distillation of BERT via sentence representation approximation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 70–79, Suzhou, China. Association for Computational Linguistics.

Congying Xia, Chenwei Zhang, Hoang Nguyen, Jiawei Zhang, and Philip S. Yu. 2020. CG-BERT: conditional text generation with BERT for generalized few-shot intent detection. *CoRR*, abs/2004.01881.

Jianguo Zhang, Kazuma Hashimoto, Wenhao Liu, Chien-Sheng Wu, Yao Wan, Philip Yu, Richard Socher, and Caiming Xiong. 2020a. Discriminative nearest neighbor few-shot intent detection by transferring natural language inference. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5064–5082, Online. Association for Computational Linguistics.

Min Zhang, Donglin Wang, and Sibo Gai. 2020b. Knowledge distillation for model-agnostic meta-learning. In *ECAI 2020*, pages 1355–1362. IOS Press.

Wangchunshu Zhou, Canwen Xu, and Julian McAuley. 2021. Meta learning for knowledge distillation. *ArXiv*, abs/2106.04570.

A Approach

A.1 Variable episode construction

Following the work by Krone et al. (2020) and Triantafillou et al. (2020), to create an episode, first, the way n is uniformly selected from the range $[3, |C_{split}|]$ for each domain $d \in D_{split}$. Then, the query shot k_q is computed as follows:

$$k_q = \min(10, (\min_{c \in C_{split}} \lfloor 0.5 * |U_c| \rfloor)),$$

where U_c is the set of instances in class c in domain d. Then, we compute the overall support set size:

$$|S| = \min \{K_{max}, \sum_{c \in C_{split}} \lceil \beta \ \min\{20, |U_c| - k_q\} \rceil\},$$

where β is sampled uniformly from $(0, 1]$. K_{max} is a constant value indicating the maximum size of the support set as a whole. Finally, we calculate the support shot k_s for each class c:

$$k_s = \min\{\lfloor R_c * (|S| - |C_{split}|) \rfloor + 1, |U_c| - k_q\},$$

where R_c noisily approximates the ratio of instances belonging to class c in domain d:

$$R_c = \frac{\exp(\alpha_c) * |U_c|}{\sum_{c' \in C_d} \exp(\alpha_{c'}) * |U_{c'}|}.$$

α_c is uniformly sampled from the interval $[\log(0.5), \log(2))$. Then, we construct distinct random episodes by choosing the set of support and query instances of each class, S_c and Q_c, from the corresponding split.

B Experiment setup

B.1 Datasets and splits

Following (Krone et al., 2020) and Li et al. (2021), the statistics on the datasets and splits for in-domain target class generalization and target domain adaptation in cross-domain intent classification are provided in Table 7 and 8, respectively.

Split\Dataset	SNIPS	TOP	ATIS
Train	(8230,4)	(20345,7)	(4373,5)
Test	(6254,3)	(4426,6)	(827,6)

Table 7: Statistics of train and test splits in NLU datasets for in-domain class generalization with (number of utterances in the split, number of intents in the split).

Target domain\Split	Train	Test
SNIPS	(TOP,20345,7) (ATIS,4373,5)	(SNIPS,6254,3)
TOP	(SNIPS, 8230,4) (ATIS,4373,5)	(TOP,4426,6)
ATIS	(TOP,20345,7) (SNIPS, 8230,4)	(ATIS,827,6)
Clinc150-Home	(Work,1500,15) (Banking,1500,15) (Credit-card,1500,15)	(Home,450,15)
Clinc150-Kitchen_dining	(Work,1500,15) (Banking,1500,15) (Credit-card,1500,15)	(Kitchen_dining,450,15)

Table 8: Statistics of train and test splits in NLU datasets for target domain adaptation with (domain, number of utterances, number of intents)