Revisit Few-shot Intent Classification with PLMs: Direct Fine-tuning vs. Continual Pre-training

Haode Zhang[1] **Haowen Liang**[1] **Liming Zhan**[1]
Xiao-Ming Wu[1*] **Albert Y.S. Lam**[2]
Department of Computing, The Hong Kong Polytechnic University, Hong Kong S.A.R.[1]
Fano Labs, Hong Kong S.A.R.[2]
{haode.zhang,michaelhw.liang,lmzhan.zhan}@connect.polyu.hk
xiao-ming.wu@polyu.edu.hk, albert@fano.ai

Abstract

We consider the task of few-shot intent detection, which involves training a deep learning model to classify utterances based on their underlying intents using only a small amount of labeled data. The current approach to address this problem is through continual pre-training, i.e., fine-tuning pre-trained language models (PLMs) on external resources (e.g., conversational corpora, public intent detection datasets, or natural language understanding datasets) before using them as utterance encoders for training an intent classifier. In this paper, we show that continual pre-training may not be essential, since the overfitting problem of PLMs on this task may not be as serious as expected. Specifically, we find that directly fine-tuning PLMs on only a handful of labeled examples already yields decent results compared to methods that employ continual pre-training, and the performance gap diminishes rapidly as the number of labeled data increases. To maximize the utilization of the limited available data, we propose a context augmentation method and leverage sequential self-distillation to boost performance. Comprehensive experiments on real-world benchmarks show that given only two or more labeled samples per class, direct fine-tuning outperforms many strong baselines that utilize external data sources for continual pre-training. The code can be found at https://github.com/hdzhang-code/DFTPlus.

1 Introduction

Intent detection is a critical module in task-oriented dialogue systems. The target is to classify utterances according to user intents. Recent progress in intent detection highly relies on deep models and datasets with well-crafted annotations. Using large-scale models or datasets has been recognized as a de facto recipe for many tasks in natural language processing (NLP) including intent detection . However, large training datasets are often not available due to the cost of labeling. Therefore, few-shot intent detection, which aims to train a classifier with only a few labeled examples, has attracted considerable attention in recent years (Dopierre et al., 2021; Zhang et al., 2022; Mi et al., 2022).

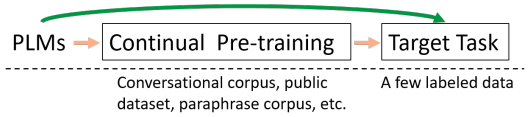


Figure 1: Illustration of continual pre-training (orange) and direct fine-tuning (green).

The main obstacle for few-shot learning is commonly believed to be overfitting, i.e. the model trained with only a few examples tends to overfit to the training data and perform much worse on test data (Vinyals et al., 2016; Zhang et al., 2022). To alleviate the problem, the mainstream approach is to transfer knowledge from *external resources* such as another labeled dataset, which has been widely used for few-shot image classification (Fei-Fei et al., 2006; Snell et al., 2017) and few-shot intent detection (Yu et al., 2018; Geng et al., 2019; Nguyen et al., 2020).

Since recently emerged large-scale pre-trained language models (PLMs) have achieved great success in various NLP tasks, most recent few-shot intent detection methods propose to fine-tune PLMs on external resources before applying them on the target task, which is known as *continual pre-training* (Gururangan et al., 2020; Ye et al., 2021), as illustrated in Fig 1. The external resources utilized for continual pre-training include conversational corpus (Wu et al., 2020a; Mehri et al., 2020; Vulić et al., 2021), natural language understanding datasets (Zhang et al., 2020a), public intent detection datasets (Zhang et al., 2021a; Yu et al., 2021), and paraphrase corpus (Ma et al., 2022). While these methods have achieved state-of-the-

* Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 11105–11121

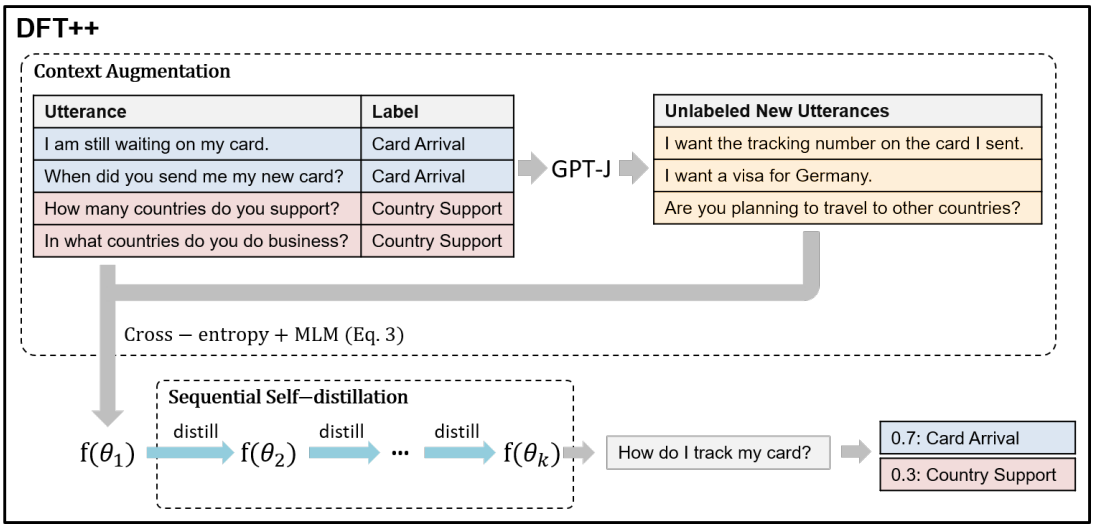


Figure 2: Illustration of DFT++ with 2 classes and 2 labeled examples per class. GPT-J is employed to generate contextually relevant unlabeled utterances. Sequential self-distillation is performed to further boost the performance.

art results, the use of external training corpora induces extra data processing effort (e.g., SBERT-Paraphrase Ma et al. (2022) uses 83 million sentence pairs from 12 datasets) as well as model bias (e.g., the trained model may be biased to the intent classes used in continual pre-training) (Xia et al., 2020b, 2021a; Nguyen et al., 2020).

It is commonly believed that directly fine-tuning PLMs with a small amount of data may generate unacceptable variance (Lee et al., 2020; Dodge et al., 2020). However, it has been recently found that the instability may be caused by incorrect use of optimizer and insufficient training (Mosbach et al., 2021; Zhang et al., 2020c). Further, some studies (Hao et al., 2019; Li et al., 2019) have revealed that in sentiment analysis and paraphase detection tasks, when directly fine-tuned with a small dataset, PLMs sush as BERT (Devlin et al., 2019) demonstrate a certain level of resilience to overfitting. Therefore, a thorough investigation is needed to explore the direct fine-tuning of PLMs for few-shot intent detection. In this work, we make the following contributions:

- We take an empirical investigation into the overfitting issue when directly fine-tuning PLMs on few-shot intent detection tasks. Our study suggests that overfitting may not be a significant concern, since the test performance improves rapidly as the size of training data increases. Further, the model's performance does not degrade as training continues. It implies that early stopping is not necessary, which is often employed to prevent overfitting in few-shot learning and requires an additional set of labeled data for validation.

- We find that direct fine-tuning (DFT) already yields decent results compared with continual pre-training methods. We further devise a DFT++ framework to fully exploit the given few labeled data and boost the performance. DFT++ introduces a novel *context augmentation* mechanism by using a generative PLM to generate *contextually relevant unlabeled data* to enable better adaptation to target data distribution, as well as a sequential self-distillation mechanism to exploit the multi-view structure in data. A comprehensive evaluation shows that DFT++ outperforms state-of-the-art continual pre-training methods with only the few labeled data provided for the task, without resorting to external training corpora.

2 Direct Fine-tuning

We investigate a straightforward approach for few-shot intent detection – directly fine-tuning (DFT) PLMs with the few-shot data at hand. However, it is a common belief that such a process may lead to severe overfitting. Before going into detail, we first formally define the problem.

2.1 Problem Formulation

Few-shot intent detection aims to train an intent classifier with only a small labeled dataset $\mathcal{D} = \{(x_i, y_i)\}_N$, where N is the dataset size, x_i denotes the i_{th} utterance, and y_i is the label. The number of

samples per label is typically less than 10.

We follow the standard practice (Sun et al., 2019; Zhang et al., 2021a) to apply a linear classifier on top of the utterance representations:

$$\mathrm{p}(y|\mathbf{h}_i) = \mathrm{softmax}\left(\mathbf{W}\mathbf{h}_i + \mathbf{b}\right) \in \mathbb{R}^L, \quad (1)$$

where $\mathbf{h}_i \in \mathbb{R}^d$ is the representation of the i_{th} utterance in $\mathcal{D}$, $\mathbf{W} \in \mathbb{R}^{L\times d}$ and $\mathbf{b} \in \mathbb{R}^L$ are the parameters of the linear layer, and L is the number of classes. We use the representation of the [CLS] token as the utterance embedding $\mathbf{h}_i$. The model parameters $\theta = \{\phi, \mathbf{W}, \mathbf{b}\}$, with ϕ being the parameters of the PLM, are trained on $\mathcal{D}$. We use a cross-entropy loss $\mathcal{L}_{\text{ce}}(\cdot)$ to learn the model parameters:

$$\theta = \arg\min_{\theta} \mathcal{L}_{\text{ce}}\left(\mathcal{D}; \theta\right). \quad (2)$$

Unlike the popular approach of continual pre-training (Zhang et al., 2020a, 2022, 2021b), DFT fine-tunes PLMs directly on the few-shot data, which may experience overfitting, leading to sub-optimal performance. To examine this issue, we conduct the following experiments.

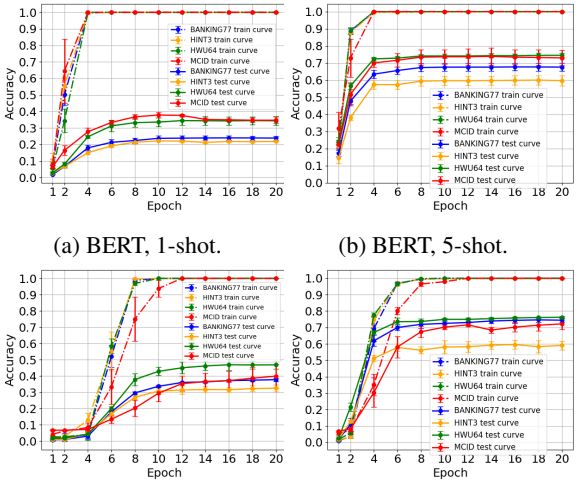
2.2 Experiments

Datasets We utilize four large-scale practical datasets. **HINT3** (Arora et al., 2020b) is created from live chatbots with 51 intents. **BANKING77** (Casanueva et al., 2020) is a fine-trained dataset focusing on banking services, containing 77 intents. **MCID** (Arora et al., 2020a) is a cross-lingual dataset for "Covid-19" with 16 intents, and we use the English version only. **HWU64** (Liu et al., 2019a) is a large-scale multi-domain dataset with 64 intents. The statistics of the datasets are given in Table 1. To simulate few-shot scenarios, we randomly sample K samples per label from the training set of each dataset to form the dataset $\mathcal{D}$.

Dataset	#Intent	#Train	#Dev	#Test
OOS	150	15000	3000	4500
BANKING77	77	10003	0	3080
HINT3	51	1579	0	676
HWU64	64	8954	1076	1076
MCID	16	1258	148	339

Table 1: Dataset statistics.

Baselines To evaluate DFT, we compare it against IsoIntentBERT (Zhang et al., 2022), a competitive baseline applying continual pre-training with public intent detection datasets. We follow the original work to pre-train BERT on OOS (Larson et al., 2019), a multi-domain public intent detection dataset containing diverse semantics, and then perform in-task fine-tuning on the small dataset $\mathcal{D}$.



(a) BERT, 1-shot. (b) BERT, 5-shot.

(c) RoBERTa, 1-shot. (d) RoBERTa, 5-shot.

Figure 3: Training and test learning curves of DFT with BERT and RoBERTa as text encoder respectively.

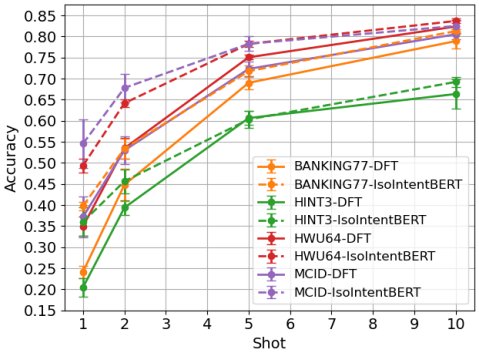


Figure 4: Comparison between DFT (solid lines) and IsoIntentBERT (dashed lines). The benefit from continued pre-training(IsoIntentBERT) decays quickly.

Results and Findings We plot the learning curves of DFT in Fig. 3, where the following observations can be drawn. First, comparing the results in 1-shot and 5-shot scenarios, the test performance of DFT improves drastically as the number of labeled examples rises from 1 to 5, leading to a fast reduction in the performance gap between the training and test performance. Second, the test performance does not deteriorate as the training progresses, and the learning curves exhibit a flat trend. These observations are consistent across various datasets and different models (BERT and

RoBERTa), including both 1-shot and 5-shot[1] scenarios. The observations also align with previous findings in sentiment analysis (Li et al., 2019) and paraphrase detection (Hao et al., 2019) tasks.

The flat learning curves indicate that early stopping is not necessary, which is often used to prevent overfitting and requires an additional set of labeled data. This is important for practitioners because model selection has been identified as a roadblock for *true few-shot learning* (Perez et al., 2021), where the labeled data is so limited that it is not worth setting aside a portion of it for early stopping. On the other hand, the rapidly reduced performance gap between DFT and IsoIntentBERT (Fig. 4) casts doubt on the necessity of continual pre-training. Thus, we raise an intriguing question:

- With only the given few labeled data, is it possible to achieve comparable or better performance than continual pre-training methods?

Our attempt to answer the question leads to DFT++, a framework designed to fully exploit the given few labeled data, which provides an affirmative answer.

3 Push the Limit of Direct Fine-Tuning

To push the limit of few-shot intent detection with only a few labeled data at hand and without using any external training corpora, DFT++ introduces two mechanisms, as shown in Fig. 2. The first is a novel context augmentation mechanism, wherein the few data are used to prompt a generative PLM to generate contextually relevant unlabeled utterances to better model target data distribution. The second is a sequential self-distillation mechanism.

3.1 Context Augmentation

Unlike continual pre-training methods that leverage external training corpora, we use the few data to solicit knowledge from generative PLMs. An intuitive way is data augmentation, which prompts the model to generate new utterances with the given intent class. However, as suggested by Sahu et al. (2022) and our analysis (Section 3.4), data augmentation for intent detection with tens of intent classes is challenging. Hence, we propose to exploit contextual relevance in an unsupervised manner instead. Specifically, for each intent class, we compose the few data into a prompt and then feed

[1]We observe the same patterns when further increasing the shot number beyond 5.

Prompt:
The following sentences belong to the same category 'cancel transfer':
Example 1: How can I cancel a transfer I made?
Example 2: Cancel transaction.
Example 3: I need to cancel a transfer.
Example 4: I want to revert a transaction I did this morning.
Example 5: I made a mistake and performed a transaction on the wrong account.
Example 6:

Generated Utterances:
I want to cancel this transaction.
How can I cancel an already invisible order?
I made a mistake on a financial transaction that I executed on the wrong account.
This transaction has already been completed.
I want to reverse a mistake I did last year.

Figure 5: An example of the prompt and generated utterances in a 5-shot scenario. Green utterances are successful cases, while the red one is a failure case.

it to GPT-J (Wang and Komatsuzaki, 2021), a powerful generative PLM, to generate novel unlabeled utterances. Fig. 5 gives an example of the prompt and generated results. The generated unlabeled data is combined with the given utterances in $\mathcal{D}$ to compose a corpus $\mathcal{D}_{\text{aug}} = \{x_i\}_i$, which can be used for masked language modeling (MLM). Hence, the model parameters θ are learned by simultaneously minimizing both the cross-entropy loss $\mathcal{L}_{\text{ce}}$ and the MLM loss $\mathcal{L}_{\text{mlm}}$:

$$\theta = \arg\min_{\theta} \left(\mathcal{L}_{\text{ce}}(\mathcal{D}; \theta) + \lambda \mathcal{L}_{\text{mlm}}(\mathcal{D}_{\text{aug}}; \theta) \right), \tag{3}$$

where λ is a balancing parameter.

Notice that there is a critical difference between the proposed context augmentation and conventional data augmentation methods. Context augmentation generates contextually relevant data (i.e., utterances with similar context to the given input but not necessarily belong to the same label class), and we use the generated data in an unsupervised manner via MLM. In contrast, conventional data augmentation methods generate new utterances with the same label as the given utterance and utilize them in a supervised manner.

3.2 Sequential Self-distillation

To further boost performance, we employ self-distillation (Mobahi et al., 2020; Allen-Zhu and Li, 2020) (Fig. 2). The knowledge in the learned model is distilled into another model with the same

architecture by matching their output logits[2]:

$$\theta_k = \arg\min_{\theta_k} \mathrm{KL}\left(\frac{\mathrm{f}\left(\mathcal{D};\theta_k\right)}{t}, \frac{\mathrm{f}\left(\mathcal{D};\theta_{k-1}\right)}{t}\right), \quad (4)$$

where $\mathrm{KL}(\cdot)$ is the Kullback-Leibler (KL) divergence, $\mathrm{f}(\cdot)$ is the output logit of the model, and t is the temperature parameter. We adopt the born-again strategy (Furlanello et al., 2018) to iteratively distill the model into a sequence of generations. Hence, the model at k_{th} generation with parameters θ_k is distilled to match the $(k-1)_{\text{th}}$ generation with parameters θ_{k-1}.

Self-distillation can provably improve model performance if the data has a multi-view structure, i.e., the data has multiple features (views) to help identify its class (Allen-Zhu and Li, 2020). Such structures naturally exist in utterances. For instance, given the following utterance of label "travel alert",

"How safe is visiting Canada this week",

both "safe" and "visiting" indicate the intent label, and it is likely that the model learns only one of them because a single feature may be sufficient to discriminate the above utterance from others with different labels, especially with limited training data. Sequential self-distillation can help to learn both features, as shown in Allen-Zhu and Li (2020).

3.3 Experiments

We evaluate DFT++ on the same benchmarks used to evaluate DFT. We compare DFT++ with state-of-the-art continual pre-training methods. Since early stopping is not necessary, as demonstrated in subsection 2.2, we combine the validation and test sets for a more comprehensive evaluation.

Baselines We compare with the following baselines. **TOD-BERT** (Wu et al., 2020a) conducts continual pre-training on dialogue corpus with MLM and response objectives. **DNNC-NLI** (Zhang et al., 2020b) and **SE-NLI** (Ma et al., 2022) employ NLI datasets. DNNC-NLI is equipped with a BERT-style pair-wise similarity model and a nearest neighbor classifier. SE-NLI employs sentence encoder (Reimers and Gurevych, 2019) with siamese and triplet architecture to learn the semantic similarity. **DNNC-Intent**, **CPFT** (Zhang et al., 2021b), **IntentBERT** (Zhang et al., 2021a) and **IsoIntentBERT** (Zhang et al., 2022) use external intent detection datasets. DNNC-Intent shares the same model structure as DNNC-NLI. CPFT adopts contrastive learning and MLM. IntentBERT employs standard supervised pre-training, based on which IsoIntentBERT introduces isotropization to improve performance. **SE-Paraphrase** (Ma et al., 2022) exploits paraphrase corpus, using the same model architecture for sentence encode as SE-NLI.

For all the baselines, we download the publicly released model if available. Otherwise, we follow the original work's guidelines to perform continual pre-training. Next, we perform standard fine-tuning similar to DFT, using hyperparameters searched within the same range as our method, with three exceptions: DNNC-NLI, DNNC-Intent, and CPFT. For these methods, we use the original design and training configuration for in-task fine-tuning.

In addition, we compare DFT++ against **CINS** (Mi et al., 2022), the most recent prompt-based method. CINS addresses intent detection by converting it into a cloze-filling problem through a carefully designed prompt template. Similar to our method, CINS directly fine-tunes PLMs on a limited amount of data.

Our method We evaluate our method and the baselines based on two popular PLMs: BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019b). The representation of the token [CLS] is used as the utterance embedding. For a fair comparison, we select the hyper-parameters with the same validation data as used by the baselines, i.e., we follow IsoIntentBERT to use a portion of the OOS dataset as the validation data. The best hyper-parameters and the parameter range are given in the appendix.

Main results We first examine the performance using a moderately small amount of data, specifically 5-shot and 10-shot scenarios. The results are summarized in Table 2. Remarkably, DFT++ performs comparably to a diverse set of baselines that leverage external resources, despite the fact that it solely utilizes the limited few-shot data available. The superiority of DFT++ can be attributed to the effective utilization of context augmentation and sequential self-distillation, both of which demonstrate improved results when applied independently in most cases. We notice that DFT++ performs better when using the stronger base model RoBERTa. As shown in Table 2b, DFT++ outperforms all the baselines in most cases. Moreover, as shown in Table 3, in most cases, DFT++ also outperforms

[2]We have also tried to add a cross-entropy term (Tian et al., 2020), but find it hurts the performance.

Method	BANKING77		HINT3		HWU64		MCID	
	5-shot	10-shot	5-shot	10-shot	5-shot	10-shot	5-shot	10-shot
TOD-BERT	67.69(1.37)	79.71(0.91)	56.33(2.14)	66.42(2.19)	74.83(1.11)	82.15(0.47)	66.37(2.65)	74.66(1.52)
DNNC-NLI	68.48(1.15)	74.53(4.83)	59.05(1.02)	65.12(1.96)	72.25(1.39)	77.91(1.11)	67.35(2.09)	75.20(1.28)
DNNC-Intent	70.36(1.85)	78.85(1.56)	58.08(4.98)	64.56(3.64)	69.86(4.27)	74.87(3.02)	70.80(3.16)	78.60(1.49)
CPFT	70.96(2.45)	79.44(.80)	61.63(2.64)	69.85(1.21)	73.63(1.74)	80.59(.61)	71.54(4.97)	79.38(1.60)
IntentBERT	70.64(1.02)	81.18(.34)	58.96(1.50)	68.96(1.50)	**77.60(.31)**	**83.55(.21)**	76.67(.84)	81.60(1.41)
IsoIntentBERT	71.78(1.40)	**81.30(.50)**	60.33(1.95)	69.23(1.16)	**78.26(.69)**	**83.70(.59)**	**78.28(1.72)**	**82.51(1.23)**
SE-Paraphrase	**71.92(.84)**	81.18(.33)	**62.28(.77)**	**70.00(1.01)**	76.75(.63)	82.88(.48)	**78.32(2.12)**	**83.08(1.32)**
SE-NLI	70.03(1.47)	80.58(1.13)	**61.69(1.59)**	68.37(1.55)	75.10(1.17)	82.57(.79)	74.54(1.86)	81.20(1.80)
DFT	69.01(1.54)	78.92(1.69)	60.65(1.60)	66.36(3.48)	75.07(.53)	82.38(1.49)	72.32(1.80)	80.53(1.15)
DFT++ (w/ CA)	**72.23(1.80)**	**82.33(.72)**	60.53(2.73)	**70.36(1.90)**	76.73(1.05)	82.61(.23)	77.45(1.66)	81.27(1.41)
DFT++ (w/ SSD)	68.86(1.49)	80.32(.81)	61.51(1.88)	68.82(2.49)	75.05(1.36)	82.14(.92)	74.17(1.09)	81.44(1.08)
DFT++ (w/ CA, SSD)	**72.90(.89)**	**82.66(.50)**	**63.08(1.17)**	**70.47(2.56)**	**77.73(1.02)**	**83.45(.38)**	**79.43(.84)**	**82.83(.76)**

(a) BERT-based evaluation results.

Method	BANKING77		HINT3		HWU64		MCID	
	5-shot	10-shot	5-shot	10-shot	5-shot	10-shot	5-shot	10-shot
DNNC-NLI	73.90(1.27)	79.51(2.56)	59.73(0.89)	64.05(2.30)	73.06(1.70)	78.12(1.86)	63.74(3.79)	73.72(1.82)
DNNC-Intent	72.97(1.46)	77.69(5.06)	61.15(1.74)	66.45(1.06)	69.74(1.85)	72.30(3.61)	72.44(2.50)	78.64(1.69)
CPFT	70.94(1.08)	78.57(.75)	58.17(3.44)	61.07(2.37)	74.36(1.15)	79.46(.81)	78.20(1.72)	83.04(1.74)
IntentRoBERTa	75.23(.89)	83.94(.33)	60.77(1.60)	68.91(1.24)	**78.97(1.26)**	84.26(.84)	77.25(2.05)	82.67(1.43)
IsoIntentRoBERTa	75.05(1.92)	84.49(.43)	59.79(2.72)	69.08(1.59)	78.09(1.06)	84.15(.58)	78.40(2.03)	83.20(1.89)
SE-Paraphrase	76.03(.64)	82.85(.89)	**63.96(.02)**	69.14(2.08)	76.50(.45)	81.25(.97)	**80.78(1.36)**	83.12(.86)
SE-NLI	**76.56(.69)**	84.65(.26)	62.60(2.45)	69.91(1.82)	78.53(.84)	84.81(.45)	**79.43(3.17)**	**84.13(1.25)**
DFT	76.11(1.16)	84.77(.43)	61.39(1.51)	68.40(1.21)	76.72(.94)	84.00(.34)	76.39(1.18)	82.55(1.15)
DFT++ (w/ CA)	**78.74(1.00)**	**85.95(.34)**	**63.17(2.20)**	**71.30(1.54)**	**79.02(.89)**	**85.49(.35)**	76.51(2.77)	**83.98(1.17)**
DFT++ (w/ SSD)	76.25(1.67)	**84.95(.53)**	61.30(2.31)	**70.12(1.35)**	77.57(.62)	**84.91(.45)**	78.73(2.30)	83.37(1.64)
DFT++ (w/ CA, SSD)	**78.90(.50)**	**86.14(.19)**	**63.61(1.80)**	**71.80(1.88)**	**79.93(.92)**	**86.21(.28)**	**80.16(2.74)**	**84.80(.79)**

(b) RoBERTa-based evaluation.

Table 2: Results of DFT++ and state-of-the-art methods. The mean value and standard deviation are reported. CA denotes context augmentation. SSD denotes sequential self-distillation. The top 3 results are highlighted.

5-shot	Bank	Home
CINS¶	89.1	80.2
DFT++ (BERT)	**91.39(.78)**	**82.11(4.09)**
DFT++ (RoBERTa)	**93.76(.46)**	**86.21(2.94)**
5-shot	Utility	Auto
CINS¶	95.4	**93.7**
DFT++ (BERT)	**96.16(.41)**	90.64(.93)
DFT++ (RoBERTa)	**97.39(.50)**	**93.31(1.21)**

Table 3: Comparison of DFT++ against CINS. ¶ denotes results copied from Mi et al. (2022). DFT++ is better in most cases, especially when RoBERTa is employed. The top 2 results are highlighted.

CINS, the most recent prompt-based method, despite that CINS employs T5-base (Raffel et al., 2020) with 220 million parameters, which is almost twice the size of our base model.

To study the impact of the number of labeled data on performance, we reduce the number to only 1 sample per label and present the results in Fig. 6. We experiment with BANKING77, a challenging fine-grained dataset. When using BERT, we observe that DFT++ begins to outperform the baselines at a crossing point of 4. When using RoBERTa, the crossing point is even smaller, at 2, which is quite surprising. We have also observed similar phenomena on other datasets, as detailed in the appendix. The observations confirm our claim that the overfitting issue in directly fine-tuning PLMs for few-shot intent detection may not be as severe as initially presumed. The performance disadvantage resulting from overfitting can be effectively alleviated by leveraging other techniques to exploit the limited available data, even without resorting to the continual pre-training approach. However, in scenarios with an extremely small number of labeled data, the transferred knowledge from continual pre-training still provides significantly better performance compared to DFT++.

3.4 Analysis

Comparison between contextual augmentation and conventional data augmentation methods

Method	BANKING77		HINT3		HWU64		MCID	
	5-shot	10-shot	5-shot	10-shot	5-shot	10-shot	5-shot	10-shot
DFT	69.01(1.54)	78.92(1.69)	60.65(1.60)	66.36(3.48)	75.07(.53)	82.38(1.49)	72.32(1.80)	80.53(1.15)
EDA	68.81(1.97)	72.97(.94)	60.50(3.06)	59.94(1.10)	74.68(.81)	72.76(5.16)	73.10(.64)	80.99(.16)
BT	69.65(1.39)	78.42(.83)	60.50(1.40)	66.33(2.69)	74.15(.84)	79.12(1.65)	75.15(2.04)	81.36(1.6)
PrompDA	71.62(.72)	80.61(2.95)	**61.51(2.20)**	69.17(1.91)	76.59(.89)	**83.29(.56)**	77.16(.98)	**81.47(2.19)**
SuperGen	64.83(1.06)	77.48(0.37)	57.30(1.41)	64.44(2.64)	69.52(0.56)	77.26(0.88)	72.55(1.37)	78.78(1.01)
GPT-J-DA	71.84(1.41)	78.34(.87)	60.24(.83)	67.40(2.41)	70.72(.78)	76.66(1.3)	73.92(2.77)	78.77(2.39)
CA	**72.23(1.80)**	**82.33(.72)**	60.53(2.73)	**70.36(1.90)**	**76.73(1.05)**	82.61(.23)	**77.45(1.66)**	81.27(1.41)

(a) BERT-based evaluation results.

Method	BANKING77		HINT3		HWU64		MCID	
	5-shot	10-shot	5-shot	10-shot	5-shot	10-shot	5-shot	10-shot
DFT	76.11(1.16)	84.77(.43)	61.39(1.51)	68.40(1.21)	76.72(.94)	84.00(.34)	76.39(1.18)	82.55(1.15)
EDA	74.74(1.08)	81.84(.59)	62.04(2.49)	66.78(1.53)	75.88(1.59)	81.91(.67)	77.17(1.85)	83.12(1.30)
BT	75.12(1.03)	84.12(.28)	60.83(1.16)	68.34(1.33)	77.31(.72)	82.89(.21)	77.49(2.71)	82.05(1.45)
PrompDA	76.56(1.15)	82.69(.99)	60.56(1.37)	69.44(1.57)	77.57(1.12)	82.94(1.29)	77.60(1.94)	83.86(2.27)
SuperGen	70.42(0.19)	81.74(0.16)	57.64(1.33)	65.88(0.54)	71.28(0.78)	81.16(0.35)	73.99(1.79)	80.08(0.89)
GPT-J-DA	76.58(1.30)	83.01(.87)	62.16(1.83)	**71.45(1.86)**	76.59(.94)	81.65(.73)	**77.91(2.22)**	82.51(1.90)
CA	**78.74(1.00)**	**85.95(.34)**	**63.17(2.20)**	71.30(1.54)	**79.02(.89)**	**85.49(.35)**	76.51(2.77)	**83.981.17)**

(b) Roberta-based evaluation results.

Table 4: Comparison of our proposed contextual augmentation against conventional data augmentation methods. CA denotes contextual augmentation. The best results are highlighted.

We compare our proposed context augmentation with the following conventional data augmentation methods. Easy Data Augmentation (EDA) (Wei and Zou, 2019) modifies a small number of utterances, e.g., through word swapping, to generate new augmented instances. Back-translation (BT) (Edunov et al., 2018) translates an utterance into another language and then translates it back[3]. PromDA (Wang et al., 2022) and SuperGen (Meng et al., 2022) are recent data augmentation methods leveraging generative PLMs. GPT-J-DA (Sahu et al., 2022) exploits the data generated by GPT-J in a supervised manner. The results in Table 4 show context augmentation is more robust against data shift. Note that SuperGen is designed for coarse-grained tasks with only two or three labels, such as sentiment classification. As a result, it may not scale effectively to intent detection tasks that involve a larger number of intents, typically ranging in the tens. The comparison between context augmentation and GPT-J-DA highlights the superiority of unsupervised exploitation of the generated data. The inconsistent effectiveness of GPT-J-DA is also reported by Sahu et al. (2022).

[3]We use French as the intermediate language, and utilize T5-base (Raffel et al., 2020) and opus-mt-fr-en (Tiedemann and Thottingal, 2020) for translation.

Quality of context augmentation To demonstrate the quality of the data generated by context augmentation, we provide some good and bad examples of generated utterances in Table 5. It is observed that GPT-J is able to generate grammatically fluent utterances that exhibit a high level of contextual relevance to the input utterances, which are utilized by DFT++ to better model the target data distribution. On the other hand, as also observed in Sahu et al. (2022), some of the generated utterances deviate from the original label and, therefore, are not suitable for data augmentation. However, DFT++ mitigates this issue by focusing solely on leveraging contextual relevance, resulting in improved robustness against data shift (Table 4).

Complementarity of continual pre-training and DFT++ Continual pre-training and DFT++ mitigate overfitting from different aspects. The former leverages external data, while the latter maximizes the utilization of the limited available data. Hence, it is likely that they are complementary. To support this claim, we present empirical results demonstrating their complementarity in Table 6. It is observed that when combined with DFT++, the two competitive methods, IsoIntentBERT and SE-Paraphrase, both demonstrate improved performance.

Input	Good	Bad
"Is there a reason why my card was declined when I attempted to withdraw money?", "How come I can not get money at the ATM?", "Why can not I withdraw cash from this ATM?", "Why will not the ATM give me cash?", "This morning, I wanted to make a withdrawal before work but my card was declined, please double check it for me as this is the first time it was declined."	"ATM will not let me withdraw my money my card as refused please help", "I withdrew less than I expected from the ATM on monday", "My wallet was stolen but my ATM card was within safely", "I spent a fortune last week and have none left on my card can you reverse refund the fees","Please give me the code that I can use in the ATM for my face to use my card"	"Why did my card never get a their villages and journey?", "An autofill took place but there was nothing to approve.", "Can I get one form my card after I have made a ctifre?", "Family needs money for the holidays they said they can not make it I hope you can help even if it is not much."
"Please order take from Jasons Deli.", "Can you please order some food for me?", "Can you look up Chinese takeout near here?", "Can i order takeaway from Spanish place?", "Find and order rasgulla of janta sweet home pvt ltd."	"I need to get some gluten free cookies for my daughter", "Can you do ticket counter take away", "How can I order Chinese food", "Delivery service please order some takeaway jahdi", "Order beef kasundi bewa rasgulla and dosa will be ready in 10 mins"	"Please make some reservation if you want booking on myhotelcom", "Drive take from a taxi", "Warehouse 26723", "Please make some reservation if you want booking on myhotelcom"

Table 5: Utterances generated by GPT-J. The first row corresponds to the label "Declined Cash Withdrawal" from BANKING77. The second row corresponds to the label "Takeaway Order" from HWU64. Good examples exhibit semantic relevance to the input data, while bad examples are irrelevant. Green words are highlighted to indicate semantic relevance, while the underlined words deviating the sentence from the original label.

IsoIntentBERT	DFT++	BANKING77	HWU64
✓		71.78(1.40)	78.26(.69)
✓	✓	**73.53(1.33)**	**80.20(1.20)**
SE-Paraphrase	DFT++	BANKING77	HWU64
✓		71.92(.84)	76.75(.63)
✓	✓	**73.21(1.24)**	**78.34(.31)**

Table 6: Complementarity of DFT++ and continued pre-training with experiments conducted on 5-shot tasks.

Method	BANKING77	
	5-shot	10-shot
DFT	69.01(1.54)	78.92(1.69)
DFT + External	67.84(.82)	81.23(.66)
DFT + EDA	70.61(1.78)	81.83(.41)
DFT + GPT-J	**72.22(1.80)**	**82.33(.72)**

Table 7: Comparison of our proposed GPT-J-based context augmentation with other alternatives. "External" denotes a corpus collected from Wikipedia.

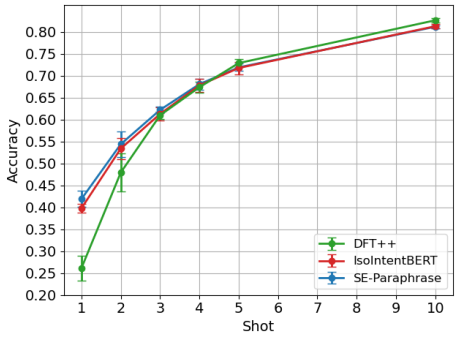
Impact of hyper-parameters We study the impact of two key hyper-parameters, the size of the generated data and the number of self-distillation generations. As visualized in Fig. 7a, a positive correlation is found between the performance and the size of the augmented data. The performance saturates after the data size per label reaches 50. It is noted that when only the given data are used for MLM, i.e., the generated data size is 0, MLM has an adversarial effect probably due to overfitting on the few given data. Such negative effect is successfully alleviated by context augmentation. As for self-distillation generations, we find that multiple generations of self-distillation are necessary to achieve better performance. In the appendix, we further analyze the impact of of the temperature parameters of GPT-J and self-distillation.

Comparison with alternative context augmentation methods We have also studied alternative context augmentation methods. The first one is Easy Data Augmentation (EDA) (Wei and Zou, 2019) with random synonym replacement, insertion, swap, and deletion. The second approach involves manually collecting a domain-specific corpus. We conduct experiments on BANKING77, since it focuses on a single domain, making it convenient to collect the corpus. We extract web pages from Wikipedia[4] with keywords that are closely relevant to "Banking", such as "Bank" and "Credit card". The keywords can be found in the appendix. As shown by Table 7, our GPT-J-based context augmentation outperforms the alternatives. We attribute the superiority to the grammatical fluency achieved by leveraging the generative power of GPT-J, which is typically compromised by EDA. Additionally, the high degree of semantic relevance observed in our approach is rarely guaranteed in the noisy corpus collected from Wikipedia.

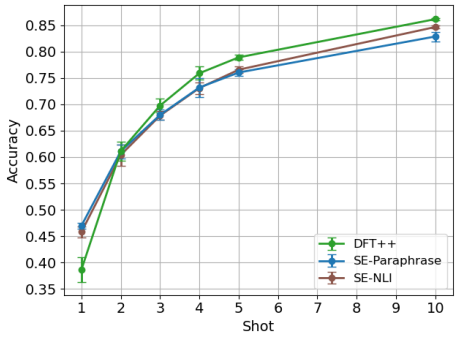
4 Related Works

Few-shot Intent Detection Before the era of PLMs, the study of few-shot intent detection focuses on model architecture (Geng et al., 2019; Xia et al., 2020a; Nguyen et al., 2020). Recently, fine-tuning PLMs has become the mainstream methodology. Zhang et al. (2020b) fine-tune pair-wise encoder on natural language inference (NLI) tasks. Zhang et al. (2021b) fine-tune PLMs in a con-

[4]https://en.wikipedia.org



(a) BERT-based experiments.



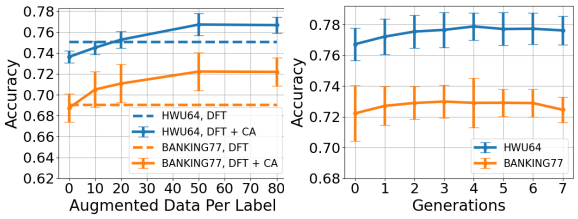
(b) RoBERTa-based experiments.

Figure 6: Impact of the size of labeled data on performance. The experiments are conducted on BANKING77. We compare DFT++ with the top 2 baselines.

trastive manner. Zhang et al. (2021a) leverage public intent detection dataset, which is further improved by isotropization (Zhang et al., 2022). Other settings are also studied, including semi-supervised learning (Dopierre et al., 2020, 2021) and incremental learning (Xia et al., 2021b). Unlike the mainstream strategy, our method does not require continual pre-training on extra resources.

Continual Pre-training of PLMs Continual pre-training of PLMs is helpful (Gururangan et al., 2020; Ye et al., 2021; Luo et al., 2021). For dialogue understanding, many works leverage conversational corpus to perform continual pre-training. Li et al. (2020) conducts continual pre-training with a dialogue-adaptive pre-training objective and a synthesized in-domain corpus. Wu et al. (2020b) further pre-trains BERT with dialogue corpora through masked language modeling and contrastive loss. Henderson et al. (2020) use Reddit conversational corpus to pre-train a dual-encoder model. Vulić et al. (2021) adopts adaptive conversational fine-tuning on a dialogue corpus.

PLM-based Data Augmentation Rosenbaum et al. (2022) fine-tune PLMs to generate data for intent detection and slot tagging. Jolly et al. (2020) develop novel sampling strategies to improve the generated utterances. Kumar et al. (2022) pre-train a token insertion PLM for utterances generation. However, these methods require slot values, which are assumed unavailable in this work. Papangelis et al. (2021) fine-tune PLMs with reinforcement learning, but our augmentation method adopts off-the-shelf PLM without further training. The closest work to ours is Sahu et al. (2022), which utilizes off-the-shelf PLMs for data augmentation. However, our method focuses solely on leveraging contextual relevance to achieve improved robustness. PLM-based data augmentation has been explored for other tasks, e.g. sentiment classification (Yoo et al., 2021; Wang et al., 2022; Chen and Liu, 2022) and natural language inference (Meng et al., 2022; Ye et al., 2022). However, these approaches may fail to scale to intent detection tasks with tens of intent classes, as shown by Sahu et al. (2022) and our experiments.



(a) (b)

Figure 7: Impact of the size of the augmented data (a) and the number of self-distillation generations (b). The experiments are conducted in 5-shot scenarios. CA denotes context augmentation.

5 Conclusions and Limitations

We revisit few-shot intent detection with PLMs by comparing two approaches: direct fine-tuning and continual pre-training. We show that the overfitting issue may not be as significant as commonly believed. In most cases, our proposed framework, DFT++, demonstrates superior performance compared to mainstream continual pre-training methods that rely on external training corpora.

One limitation of DFT++ is the computational overhead caused by generative PLMs. Additionally, our current approach includes all utterances generated by the PLM, even those that might lack contextual relevance or contain noise. These issues are left for future exploration.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments. This research was partially supported by the grant of HK ITF ITS/359/21FP.

References

Zeyuan Allen-Zhu and Yuanzhi Li. 2020. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*.

Abhinav Arora, Akshat Shrivastava, Mrinal Mohit, Lorena Sainz-Maza Lecanda, and Ahmed Aly. 2020a. Cross-lingual transfer learning for intent detection of covid-19 utterances.

Gaurav Arora, Chirag Jain, Manas Chaturvedi, and Krupal Modi. 2020b. Hint3: Raising the bar for intent detection in the wild. *arXiv preprint arXiv:2009.13833*.

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.

Yanan Chen and Yang Liu. 2022. Rethinking data augmentation in text-to-text paradigm. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1157–1162, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping.

Thomas Dopierre, Christophe Gravier, and Wilfried Logerais. 2021. ProtAugment: Intent detection meta-learning through unsupervised diverse paraphrasing. 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 2454–2466, Online. Association for Computational Linguistics.

Thomas Dopierre, Christophe Gravier, Julien Subercaze, and Wilfried Logerais. 2020. Few-shot pseudo-labeling for intent detection. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4993–5003.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Li Fei-Fei, Robert Fergus, and Pietro Perona. 2006. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born again neural networks. In *International Conference on Machine Learning*, pages 1607–1616. PMLR.

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.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2019. Visualizing and understanding the effectiveness of bert. *arXiv preprint arXiv:1908.05620*.

Matthew Henderson, Iñigo Casanueva, Nikola Mrkvsi'c, Pei hao Su, Tsung-Hsien, and Ivan Vulic. 2020. Convert: Efficient and accurate conversational representations from transformers. *ArXiv*, abs/1911.03688.

Shailza Jolly, Tobias Falke, Caglar Tirkaz, and Daniil Sorokin. 2020. Data-efficient paraphrase generation to bootstrap intent classification and slot labeling for new features in task-oriented dialog systems. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, pages 10–20, Online. International Committee on Computational Linguistics.

Manoj Kumar, Yuval Merhav, Haidar Khan, Rahul Gupta, Anna Rumshisky, and Wael Hamza. 2022. Controlled data generation via insertion operations for NLU. In *Proceedings of the 2022 Conference of the North American Chapter of the Association*

for Computational Linguistics: Human Language Technologies: Industry Track, pages 54–61, Hybrid: Seattle, Washington + 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.

Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. 2020. Mixout: Effective regularization to finetune large-scale pretrained language models. In *International Conference on Learning Representations*.

Junlong Li, Zhuosheng Zhang, Hai Zhao, Xi Zhou, and Xiang Zhou. 2020. Task-specific objectives of pre-trained language models for dialogue adaptation. *ArXiv*, abs/2009.04984.

Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. 2019. Exploiting bert for end-to-end aspect-based sentiment analysis. *arXiv preprint arXiv:1910.00883*.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2019a. Benchmarking natural language understanding services for building conversational agents. In *Increasing Naturalness and Flexibility in Spoken Dialogue Interaction - 10th International Workshop on Spoken Dialogue Systems, IWSDS 2019, Syracuse, Sicily, Italy, 24-26 April 2019*, volume 714 of *Lecture Notes in Electrical Engineering*, pages 165–183. Springer.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ruikun Luo, Guanhuan Huang, and Xiaojun Quan. 2021. Bi-granularity contrastive learning for post-training in few-shot scene. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1733–1742.

Tingting Ma, Qianhui Wu, Zhiwei Yu, Tiejun Zhao, and Chin-Yew Lin. 2022. On the effectiveness of sentence encoding for intent detection meta-learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3806–3818, Seattle, United States. Association for Computational Linguistics.

Shikib Mehri, Mihail Eric, and Dilek Hakkani-Tur. 2020. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *arXiv preprint arXiv:2009.13570*.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. In *NeurIPS*.

Fei Mi, Yasheng Wang, and Yitong Li. 2022. Cins: Comprehensive instruction for few-shot learning in task-oriented dialog systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11076–11084.

Hossein Mobahi, Mehrdad Farajtabar, and Peter Bartlett. 2020. Self-distillation amplifies regularization in hilbert space. *Advances in Neural Information Processing Systems*, 33:3351–3361.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. In *9th International Conference on Learning Representations*, CONF.

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.

Alexandros Papangelis, Karthik Gopalakrishnan, Aishwarya Padmakumar, Seokhwan Kim, Gokhan Tur, and Dilek Hakkani-Tur. 2021. Generative conversational networks. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 111–120, Singapore and Online. Association for Computational Linguistics.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. In *Advances in Neural Information Processing Systems*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Andy Rosenbaum, Saleh Soltan, Wael Hamza, Yannick Versley, and Markus Boese. 2022. LINGUIST: Language model instruction tuning to generate annotated utterances for intent classification and slot tagging. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 218–241, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Gaurav Sahu, Pau Rodriguez, Issam Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. 2022. Data augmentation for intent classification with off-the-shelf large language models. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 47–57, Dublin, Ireland. 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*, pages 4077–4087.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China national conference on Chinese computational linguistics*, pages 194–206. Springer.

Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. 2020. Rethinking few-shot image classification: a good embedding is all you need? *arXiv preprint arXiv:2003.11539*.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638.

Ivan Vulić, Pei-Hao Su, Samuel Coope, Daniela Gerz, Paweł Budzianowski, Iñigo Casanueva, Nikola Mrkšić, and Tsung-Hsien Wen. 2021. ConvFiT: Conversational fine-tuning of pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1151–1168, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Yufei Wang, Can Xu, Qingfeng Sun, Huang Hu, Chongyang Tao, Xiubo Geng, and Daxin Jiang. 2022. PromDA: Prompt-based data augmentation for low-resource NLU tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4242–4255, Dublin, Ireland. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. 2020a. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 917–929, Online. Association for Computational Linguistics.

Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. 2020b. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 917–929, Online. Association for Computational Linguistics.

Congying Xia, Caiming Xiong, and Philip Yu. 2021a. Pseudo siamese network for few-shot intent generation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2005–2009.

Congying Xia, Caiming Xiong, Philip Yu, and Richard Socher. 2020a. Composed variational natural language generation for few-shot intents. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3379–3388, Online. Association for Computational Linguistics.

Congying Xia, Wenpeng Yin, Yihao Feng, and Philip Yu. 2021b. Incremental few-shot text classification with multi-round new classes: Formulation, dataset and system. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1351–1360, Online. Association for Computational Linguistics.

Congying Xia, Chenwei Zhang, Hoang Nguyen, Jiawei Zhang, and Philip Yu. 2020b. Cg-bert: Conditional text generation with bert for generalized few-shot intent detection. *arXiv preprint arXiv:2004.01881*.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. Zerogen: Efficient zero-shot learning via dataset generation.

Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. 2021. CrossFit: A few-shot learning challenge for cross-task generalization in NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7163–7189, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyoung Park. 2021. GPT3Mix: Leveraging large-scale language models for text augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2225–2239, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dian Yu, Luheng He, Yuan Zhang, Xinya Du, Panupong Pasupat, and Qi Li. 2021. Few-shot intent classification and slot filling with retrieved examples. *arXiv preprint arXiv:2104.05763*.

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. 2018. Diverse few-shot text classification with multiple metrics. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1206–1215.

Haode Zhang, Haowen Liang, Yuwei Zhang, Li-Ming Zhan, Xiao-Ming Wu, Xiaolei Lu, and Albert Lam. 2022. Fine-tuning pre-trained language models for few-shot intent detection: Supervised pre-training and isotropization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 532–542, Seattle, United States. Association for Computational Linguistics.

Haode Zhang, Yuwei Zhang, Li-Ming Zhan, Jiaxin Chen, Guangyuan Shi, Xiao-Ming Wu, and Albert Y.S. Lam. 2021a. Effectiveness of pre-training for few-shot intent classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1114–1120, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jianguo Zhang, Trung Bui, Seunghyun Yoon, Xiang Chen, Zhiwei Liu, Congying Xia, Quan Hung Tran, Walter Chang, and Philip Yu. 2021b. Few-shot intent detection via contrastive pre-training and fine-tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1906–1912, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Jianguo Zhang, Kazuma Hashimoto, Wenhao Liu, Chien-Sheng Wu, Yao Wan, Philip Yu, Richard Socher, and Caiming Xiong. 2020b. 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.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2020c. Revisiting few-sample bert fine-tuning. In *International Conference on Learning Representations*.

A Appendix

Hyper-parameters We determine the hyper-parameters by grid search. The best hyper-parameters and the search range are summarized in Table 8 and Table 9, respectively. The grid search is performed with OOS dataset. Specifically, we follow IsoIntentBERT to use the two domains "Travel" and "Kitchen dining" as the validation set. To guarantee a fair comparison, the same validation set is also employed for all the baselines.

PLM	Hyper-parameter
BERT	$\text{lr}_{\text{PLM}} = 2e-4$, $\text{lr}_{\text{cls}} = 2e-5$, $\lambda = 1.0$, context_size =50, $t = 100$, iteration=6.
RoBERTa	$\text{lr}_{\text{PLM}} = 2e-5$, $\text{lr}_{\text{cls}} = 2e-3$, $\lambda = 0.1$, context_size =50, $t = 40$, iteration=5.

Table 8: Hyper-parameters of DFT++. lr_{PLM} and lr_{cls} denote the learning rate of the PLM and the linear classifier, respectively. context_size is the size of the augmented contextual utterances per label. iteration is the number of iterations/generations in sequential self-distillation.

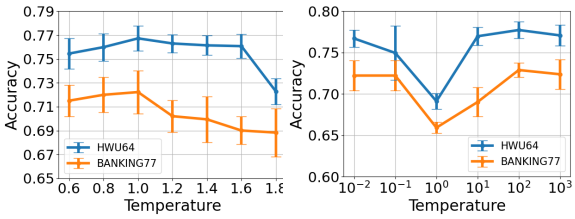
Parameter	Range
lr_{PLM}	$\{2e-5, 2e-4, 2e-3\}$
lr_{cls}	$\{2e-5, 2e-4, 2e-3\}$
λ	$\{0.01, 0.1, 1.0, 10.0\}$
context_size	$\{1, 2, 5, 10, 20, 50, 80\}$
t	$\{0.1, 1, 10, 40, 80, 100, 200, 500\}$
iteration	$\{1, 2, 3, 4, 5, 6, 7\}$

Table 9: Grid search range of hyper-parameters.

Implementation details We use Python, PyTorch library and Hugging Face library [5] to implement the model. We adopt *bert-base-uncased* and *roberta-base* with around 110 million parameters. We use AdamW as the optimizer. We use different learning rates for PLMs and the linear classifier, determined by grid-search. The parameter for weight decay is set to $1e-3$. We employ a linear scheduler with the warm-up proportion of 5%. We fine-tune the model for 200 epochs to guarantee convergence. The experiments are conducted with Nvidia RTX 3090 GPUs. We repeat all experiments for 5 times, reporting the averaged accuracy and standard deviation.

Impact of the number of labeled data on performance We provide the full results in Fig. 9. It is

[5] https://github.com/huggingface/transformers



(a) (b)

Figure 8: Impact of the temperature parameter of GPT-J (a) and self-distillation (b). The experiments are conducted in 5-shot scenarios.

observed that DFT++ outperforms many competitive methods fine-tuned on extra data even when the number of labeled data is small.

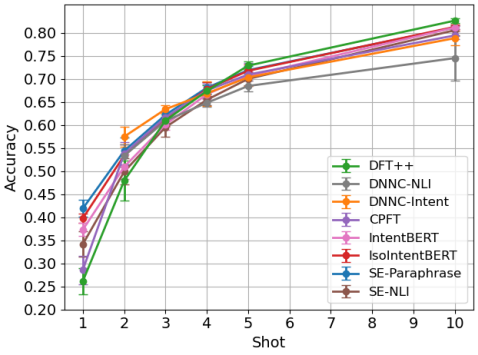
Keywords used to collect the corpus for an alternative context augmentation method As introduced in subsection 3.4, one alternative context augmentation method involves manually collecting a domain-specific corpus. We experiment with BANKING77. To collect an external corpus, we extract web pages from Wikipedia[6] with keywords closely related to "Banking", such as "Bank" and "Credit card". The adopted keywords are summarized in Table 10.

"Bank", "Credit", "Debt", "Payment", "Fund", "Credit card", "Banking agent", "Bank regulation", "Cheque", "Coin", "Deposit account", "Electronic funds transfer", "Finance", "Internet banking", "Investment banking", "Money", "Wire transfer", "Central bank", "Credit union", "Public bank", "Cash", "Call report", "Ethical banking", "Loan", "Mobile banking", "Money laundering", "Narrow banking", "Private banking"

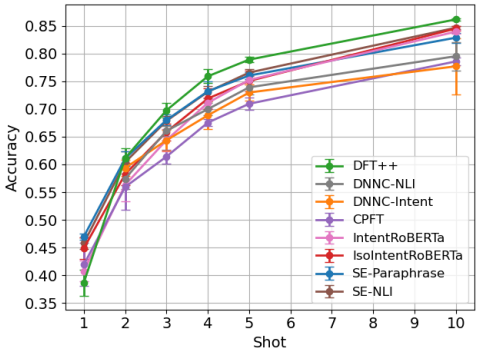
Table 10: Key words used to collect the corpus from Wikipedia.

Analysis of hyper-parameters We show the impact of the temperature parameter of GPT-J and self-distillation in Fig. 8. The temperature parameter of GPT-J controls the diversity of the generated context. A higher temperature makes the generated text more diverse. As shown in the figure, the best performance is reached when the diversity is moderate.For self-distillation, both small and large temperatures can produce good results.

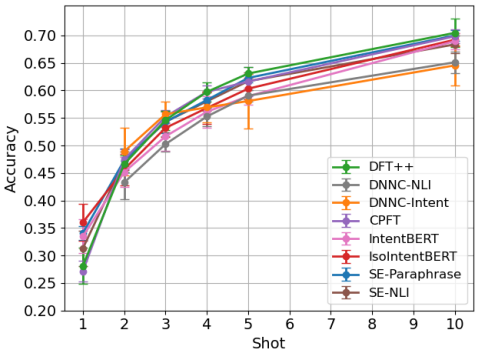
[6]https://en.wikipedia.org



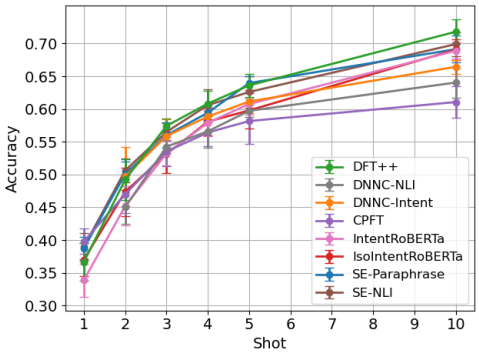
(a) BERT-based experiments on BANKING77.



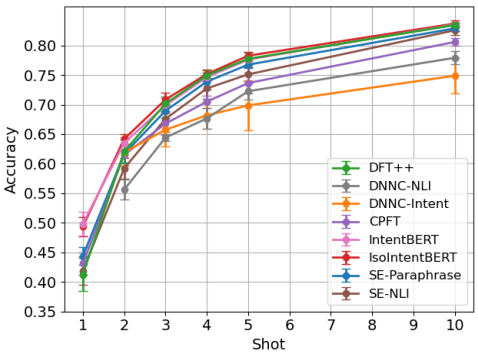
(b) RoBERTa-based experiments on BANKING77.



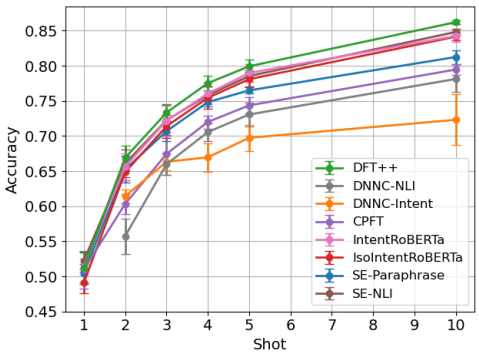
(c) BERT-based experiments on HINT3.



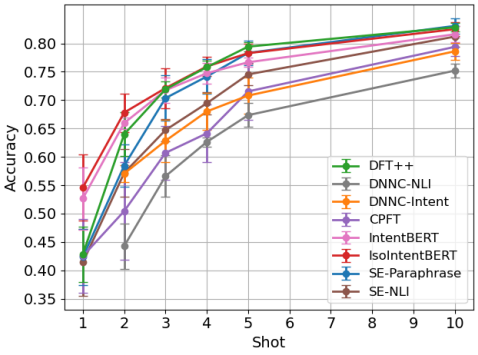
(d) RoBERTa-based experiments on HINT3.



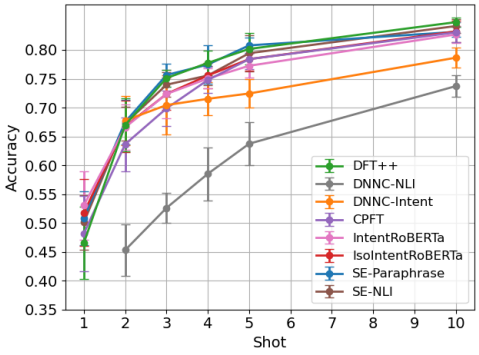
(e) BERT-based experiments on HWU64.



(f) RoBERTa-based experiments on HWU64.



(g) BERT-based experiments on MCID.



(h) RoBERTa-based experiments on MCID

Figure 9: Impact of the number of labeled data on model performance.

ACL 2023 Responsible NLP Checklist

A For every submission:

☐ A1. Did you describe the limitations of your work?
Left blank.

☐ A2. Did you discuss any potential risks of your work?
Left blank.

☐ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☐ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☐ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C ☐ Did you run computational experiments?

Left blank.

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Left blank.

D ☐ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.