Task-adaptive Pre-training and Self-training are Complementary for Natural Language Understanding

Shiyang Li
UC Santa Barbara
shiyangli@ucsb.edu

Semih Yavuz
Salesforce Research
syavuz@salesforce.com

Wenhu Chen *
UC Santa Barbara
wenhuchen@ucsb.edu

Xifeng Yan
UC Santa Barbara
xyan@cs.ucsb.edu

Abstract

Task-adaptive pre-training (TAPT) and Self-training (ST) have emerged as the major semi-supervised approaches to improve natural language understanding (NLU) tasks with massive amount of unlabeled data. However, it's unclear whether they learn similar representations or they can be effectively combined. In this paper, we show that TAPT and ST can be complementary with simple TFS protocol by following **T**APT → **F**inetuning → **S**elf-training (TFS) process. Experimental results show that TFS protocol can effectively utilize unlabeled data to achieve strong combined gains consistently across six datasets covering sentiment classification, paraphrase identification, natural language inference, named entity recognition and dialogue slot classification. We investigate various semi-supervised settings and consistently show that gains from TAPT and ST can be strongly *additive* by following TFS procedure. We hope that TFS could serve as an important semi-supervised baseline for future NLP studies.

1 Introduction

Deep neural networks (Goodfellow et al., 2016) often require large amounts of labeled data to achieve state-of-the-art performance (Xie et al., 2020). However, acquiring high-quality labels is a costly process, which inspires research on methods that can effectively utilize unlabeled data to improve performance (He et al., 2020). Large pre-trained language models like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and T5 (Raffel et al., 2020) can learn general language understanding abilities from large-scale unlabeled corpora and have reduced this annotation cost. In this paradigm, large neural networks are first pre-trained on massive amounts of unlabeled data in a self-supervised manner and then finetuned on large amount of labeled data for downstream tasks, which has led

---

*Now at Google Research.

to large improvements for natural language understanding on standard benchmarks (Wang et al., 2019b,a). However, their success still relies on large amount of data during finetuning stage. For example, Wu et al. (2020) shows that BERT only achieves 6.4% joint goal accuracy with 1% finetuning data for dialogue state tracking task, a core component of task-oriented dialogue systems, making it far behind its full counterpart 45.6%. This data-intensive finetuning poses several challenges for many real-world applications, where collecting large amount of labeled data is not only expensive and time-consuming, but also infeasible in many cases due to data access and privacy constraints (Wang et al., 2021).

Semi-supervised learning (Thomas, 2009) provides a plausible solution to address aforementioned data hungry issue by making effective use of freely available unlabeled data. One of the most popular semi-supervised learning algorithms is self-training (Scudder, 1965). In self-training, a teacher model is first trained on available labeled data and then used to generate pseudo labels for unlabeled data. The original hand-annotated labeled data and the pseudo-labeled data are combined to train a student model. The student model is assigned as a teacher model in next round and the teacher-student training procedure is repeated until convergence or reaching maximum rounds. Self-training utilizes unlabeled data in a *task-specific* way during pseudo labeling process (Chen et al., 2020b) and has been successfully applied to a variety of tasks, including image recognition (Xie et al., 2020; Zoph et al., 2020), automatic speech recognition (Kahn et al., 2020), text classification (Du et al., 2021; Mukherjee and Awadallah, 2020), sequence labeling (Wang et al., 2021) and neural machine translation (He et al., 2020).

Recently, task-adaptive pre-training (TAPT) (Gururangan et al., 2020) was further proposed, which can adapt pre-trained language models, e.g. BERT

and RoBERTa to unlabeled in-domain training set to improve performance (Gururangan et al., 2020). The intuition of TAPT is that datasets curated to capture specific tasks of interest tend to cover only a subset of the text available within the broader domain and continued pretraining on the task dataset itself or data relevant to the task can be helpful (Gururangan et al., 2020). TAPT tends to adapt its linguistic representation by utilizing the unlabeled data in a *task-agnostic* way (Chen et al., 2020b). With the recent success of task-adaptive pre-training and self-training in natural language understanding (NLU), a research question arises: *Are task-adaptive pre-training (TAPT) and self-training (ST) complementary for natural language understanding (NLU)?*

In this paper, we show that TAPT and ST can be complementary with simple TFS protocol by following **T**APT → **F**inetuning → **S**elf-training process (TFS). TFS protocol follows three steps: (1) TAPT on unlabeled corpus drawn from a task (2) Standard supervised finetuning on labeled data inheriting parameters from TAPT as initialization to train a teacher model (3) Teacher model generates pseudo labels for *the same* unlabeled corpus in (1) and trains a student model in a self-training framework until convergence or reaching maximum rounds as shown in Figure 1. The first step utilizes unlabeled corpus in a task-agnostic way to learn general linguistic representations while the third one utilizes unlabeled corpus in a task-specific way during pseudo-labeling process. Therefore, unlabeled data are utilized *twice* through two different ways by taking advantages of TAPT and ST. TFS can effectively utilize unlabeled data to achieve strong combined gains of TAPT and ST consistently across six datasets covering sentiment classification, paraphrase identification, natural language inference, named entity recognition and dialogue slot classification. We further investigate various semi-supervised settings and consistently show that gains from TAPT and ST can be strongly *additive* by following TFS procedure.

2 Related Work

Pre-training. Unsupervised or self-supervised pre-training have achieved remarkable successes in natural language processing (Devlin et al., 2019; Liu et al., 2019; Radford et al., 2019; Raffel et al., 2020; Brown et al., 2020). However, these models are pre-trained on a very large general domain cor-

pus, e.g. Wikipedia, and may limit their performance on a specific task due to distribution shift (Lee et al., 2020; Wu et al., 2020; Gururangan et al., 2020). To better handle aforementioned issue, domain-adaptive pre-training (DAPT) by continuing pre-training of existing language models, e.g. BERT and RoBERTa, on a large corpus of unlabeled domain-specific text data has been proposed and achieved great successes in specific domains (Gururangan et al., 2020; Lee et al., 2020; Wu et al., 2020). Lee et al. (2020) proposed BioBERT by continuing pre-training of BERT on biomedical domain corpus and outperformed BERT in biomedical text mining significantly. Following a similar idea, Wu et al. (2020) proposed ToD-BERT by continuing pre-training of BERT on nine dialogue datasets for NLU tasks in task-oriented dialogue systems and achieved great successes in various few-shot NLU tasks in dialogue domain. Gururangan et al. (2020) took one step further and continued pre-training of language models on a much smaller amount of unlabeled data but drawn from the same distribution for a given task (TAPT), which not only can achieve competitive results with DAPT but also is complementary with it.

Self-training. Self-training as one of the earliest and simplest semi-supervised learning has recently shown state-of-the-art performance for tasks like image classification (Xie et al., 2020; Sun et al., 2019), object detection (Zoph et al., 2020) and can perform at par with fully supervised models while using much less training data. On natural language processing, Mukherjee and Awadallah (2020) applied self-training for few-shot text classification and incorporated uncertainty estimation of the underlying neural network for unlabeled data selection. Wang et al. (2021) improved self-training with meta-learning by adaptive sample re-weighting to mitigate error propagation from noisy pseudo-labels for named entity recognition and slot tagging in task-oriented dialog systems. He et al. (2020) injected noise to the input space as a noisy version of self-training for neural sequence generation and obtained state-of-the-art performance for tasks like neural machine translation. Du et al. (2021) utilized information retrieval to retrieve task-specific in-domain data from a large bank of web sentences for self-training. Beyond these applications of self-training, Wei et al. (2021) further theoretically proved that self-training and input-consistency regularization will achieve high

accuracy with respect to ground-truth labels under certain assumptions.

There also exists works combing pre-training with self-training. Chen et al. (2020b) first conducted self-supervised pre-training with SimCLR (Chen et al., 2020a) on ImageNet (Russakovsky et al., 2015) in a task-agnostic way, then finetuned pre-trained models on limited labeled data and finally did self-training/knowledge distillation (Hinton et al., 2015) via the same unlabeled examples as pre-training in a task-specific way. Such a framework enables models to make use of data twice in both pre-training and self-training/knowledge distillation stage. (Xu et al., 2020) followed this framework on speech recognition and achieved state-of-the-art performance only with very limited labeled data. However, it's unclear that language models like BERT that have already been pre-trained in a very large general corpus can benefit this framework or not since Chen et al. (2020b) and Xu et al. (2020) conducted pre-training from scratch. In addition, they only did self-training in one round, making it unclear whether iterative self-training without pre-training can achieve comparable results in the end. A recent work (Du et al., 2021) did both continuing pre-training and self-training in retrieved data from open domains but only observe gains for self-training while our work utilizes existing in-domain unlabeled data and found that both TAPT and self-training are effective.

3 Algorithms

3.1 Problem setup

Denote $D_l = \{x_i, y_i\}^N$ to be a set of N labeled instances, where x_i is a sequence of m tokens: $x_i = \{x_{i1}, x_{i2}, ..., x_{im}\}$ with y_i being its label. Also, consider $D_u = \{x_j\}^M$ to be a set of M unlabeled instances drawn from the same distribution of $\{x_i\}^N$, where $M \gg N$. Assuming that we can only access a small amount of labeled data along with a much larger amount of unlabeled data, our goal is to fully leverage unlabeled data D_u to improve model performance.

3.2 Task-adaptive Pre-training (TAPT)

One simple yet effective way to improve BERT-like models with unlabeled data is task-adaptive pre-training (TAPT). The approach of TAPT is quite straightforward – simply continuing pre-training BERT-like models with masked language modeling (MLM) (Devlin et al., 2019) on unlabeled text data

for a given task (Gururangan et al., 2020).

Specifically, during MLM process, a proportion of random sample of tokens in the input sequence is selected and replaced with the special token [MASK]. We conduct dynamical token masking during batch training following (Liu et al., 2019; Wu et al., 2020). The MLM loss function is the cross entropy loss on predicting the masked tokens:

$$\mathcal{L}_{\text{mlm}} = -\sum_{j=1}^{M}\sum_{k=1}^{m} \mathbb{1} * \log(\, p(x_{jk})), \quad (1)$$

where $\mathbb{1}$ is 1 if x_{jk} is masked out in the input, otherwise 0.

3.3 Self-training (ST)

Self-training begins with a *teacher* model p^t trained on the labeled data D_l. The teacher model is used to generate pseudo labels for unlabeled data D_u. The augmented data $D_l \cup D_u$ is then used to train a student model p^s. Specifically, $\forall x_j \in D_u$, we use teacher model to generate its soft label and then student model is trained with standard cross-entropy loss for labeled data and KL divergence for unlabeled data, which can be formulated as:

$$\mathcal{L}_{\text{st}} = - \sum_{(x_i, y_i) \in D_l} \log(y_i | p^s(x_i))$$
$$- \sum_{x_j \in D_u} \text{KL}(p^t(x_j) || p^s(x_j)), \quad (2)$$

where teacher model p^t is fixed in the current round. After training of student model with objective $\mathcal{L}_{\text{st}}$, it is assigned as a new teacher model in the next round and the teacher-student training procedure is repeated until convergence or reaching maximum rounds.

3.4 TAPT → Finetuning → Self-training (TFS)

Although TAPT has been proven effective to utilize unlabeled data, it's *task-agnostic* in the sense that it's unaware of specific tasks, e.g. classification or name entity recognition. This paradigm learns general linguistic representations buried under unlabeled data, which are not directly tailored to a specific task. Utilizing data in a *task-agnostic* may lose the information of unlabeled data key to the task at hand. On the contrary, self-training utilizes unlabeled data in a *task-specific* way. Pseudo labels are obtained through trained models and task-specific information can be encoded into pseudo
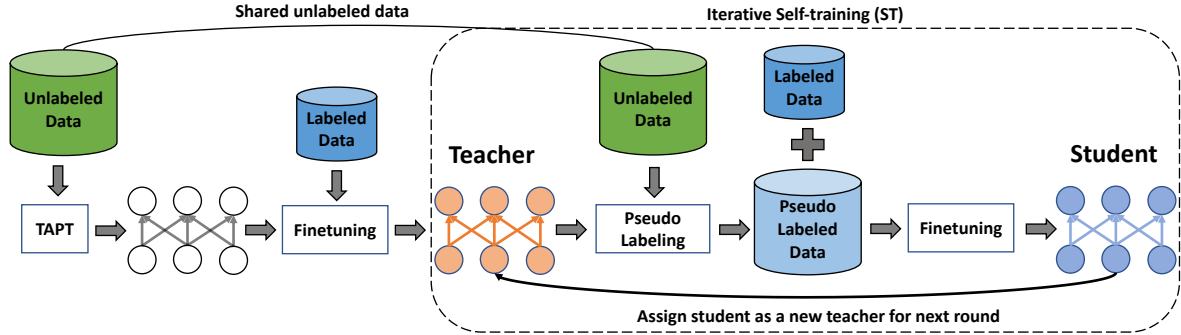
Figure 1: The overall pipeline of TFS. It has three steps (1) TAPT on unlabeled corpus drawn from a task (2) Train a teacher model on labeled data with TAPT as initialization (3) Teacher generates pseudo labels from *share* unlabeled corpus with (1) and trains a student model with both labeled and pseudo labeled data in an iterative self-training framework.

labels. However, this method may be effective only when a good fraction of the predictions on unlabeled samples are correct (He et al., 2020), otherwise early mistakes made by *teacher* model p^t due to scarcity of labeled data can reinforce itself by generating incorrectly labeled data and re-training with this data will lead to an even worse p^s in the next round (Zhu and Goldberg, 2009).

TFS protocol by following **T**APT → **F**inetuning → **S**elf-training (TFS) process can take advantages of TAPT and ST but at the same time avoid their weakness. The overall pipeline of TFS is shown in Figure 1. TFS first utilizes unlabeled data in a *task-agnostic* way by TAPT to have a better initialization for finetuning in next step and then finetunes a teacher model initializing its parameters from TAPT on labeled data in a standard supervised way. These two steps can build a better teacher model, avoid early mistakes and generate more accurate

---

Algorithm 1 TFS Protocol

Input: Labeled corpus D_l, unlabeled corpus D_u and initialized model p_θ

1: Update model p_θ with TAPT on unlabeled corpus D_u by Equation 1
2: Train a teacher model p_τ initialized with p_θ by finetuning on labeled corpus D_l
3: **repeat**
4:     Apply p_τ to the unlabeled corpus D_u to obtain $\hat{D}_u := \{(x_j, p_\tau(x_j)) | \forall x_j \in D_u\}$
5:     Train a student model p_τ on $D_l \cup \hat{D}_u$ by Equation 2
6:     Assign p_τ as a teacher for the next round
7: **until** Convergence or maximum rounds are reached

---

predictions for students, which is key to the success of self-training. The unlabeled data is leveraged again during self-training process in a *task-specific* way to further boost the performance of models at hand. We summarize the workflow of TFS in Algorithm 1.

4 Experiments

Here we conduct comprehensive experiments and analysis on different NLU datasets to demonstrate the effectiveness of TFS.

4.1 Experimental Setup

We use six popular large-scale datasets covering sentiment classification, paraphrase identification, natural language inference, named entity recognition and dialogue slot classification as follows.

(1) **SST-2** (Socher et al., 2013) consists of sentences from movie reviews and human annotations of their sentiment. The task is to predict the sentiment of a given sentence (Wang et al., 2019b).

(2) Both **QNLI** (Wang et al., 2019b) and **MNLI** (Williams et al., 2018) are natural language inference datasets. QNLI is adapted from the SQuAD (Rajpurkar et al., 2016) question answering dataset and the task is to determine whether the context sentence contains the answer to the question (Wang et al., 2019b), which can be regarded as a binary classification problem. MNLI is slightly different from QNLI as it has multiple genres. Given a premise sentence and a hypothesis sentence, the task is to predict whether the premise entails the hypothesis (*entailment*), contradicts the hypothesis (*contradiction*), or neither (*neutral*) (Wang et al., 2019b).

(3) **QQP** (Chen et al., 2018) is a paraphrase iden-

Dataset	Task	Train size	Number of classes	Evaluation metrics
SST-2	Sentiment analysis	67,349	2	Accuracy
QNLI	Natural language inference	104,743	2	Accuracy
MNLI	Natural language inference	100,000*	3	Accuracy
QQP	Paraphrase identification	100,000*	2	F1
CoNLL 2003	Named entity recognition	14,041	9	F1
MultiWOZ 2.1	Slot classification	56,557	30	Micro-F1

Table 1: Dataset summary for evaluation. * are datasets that we randomly sample 100K instances from original training sets due to the high cost of iterative self-training.

tification dataset. The goal is to determine if two questions asked on Quora are semantically equivalent (Wang et al., 2019b), which can also be formulated as a binary classification problem.

(4) **CoNLL 2003** (Tjong Kim Sang and De Meulder, 2003) is a name entity recognition dataset and the task is to recognize four types of named entities: *persons*, *locations*, *organizations* and *miscellaneous* entities, where *miscellaneous* type does not belong to any of the previous three.

(5) **MultiWOZ 2.1** (Eric et al., 2020) is a large-scale, multi-domain dialogue dataset of human-human conversations. We convert each dialogue into turns and the task is to predict whether a slot, e.g. restaurant name, is mentioned in a turn and can be cast as a multi-label binary slot classification problem (Li et al., 2021).

SST-2, QNLI, MNLI and QQP datasets are from GLUE benchmark [1] and we only report their results on development sets as extensive experiments don't allow us to submit predictions on their test sets to the official leaderboard due to submission limitations [2]. Note that for both MNLI and QQP, we randomly downsample their training sets into 100K and development sets into 5K otherwise iterative self-training in various semi-supervised setups can be too costly and for MNLI, we report results on the matched development set. On both CoNLL 2003 and MultiWOZ 2.1, we report results of their test sets. For SST-2, MNLI and QNLI, we use standard accuracy metric and for QQP and CoNLL 2003 we report their F1 scores. For MultiWOZ 2.1, we report micro-F1. We summarize details of each dataset including task, full training data size, number of classes and their evaluation metric in Table 1.

TAPT. We use `BERT-base` and `BERT-large` as our backbone to leverage both labeled and unlabeled data. Both labeled and unlabeled data are used for TAPT in our implementation so that we can use the same checkpoint for different data split and labeled data size without repeating costly pre-training process on the same dataset. During TAPT process, we use MLM objective with random token masking probability 0.15 for each training set listed in Table 1 following previous work (Wu et al., 2020).

Finetuning. We follow standard supervised finetuning paradigm (Devlin et al., 2019) by adding a linear projection layer with weight $W \in \mathbb{R}^{K \times I}$ on top of BERT in labeled data for each dataset listed in Table 1, where K is the number of classes and I is the dimensionality of representations of BERT. Specifically, for SST-2, QNLI, MNLI and QQP, we pass the representation of `[CLS]` token H^{CLS} to a linear layer followed by a `Softmax` function. Models are trained with cross-entropy loss between the predicted distributions `Softmax`$(W(H^{CLS}))$ and their ground truth labels. For CoNLL 2003 name entity recognition task, we feed the representation of each token into a linear layer followed by a `Softmax` function. Models are trained with average cross-entropy loss between the predicted distributions and their labels over all tokens [3]. For multilabel binary slot classification task on MultiWOZ 2.1, we pass the representation of `[CLS]` token H^{CLS} into a linear layer followed by a `Sigmoid` function. Models are trained with mean binary cross-entropy loss between the predicted distributions `Sigmoid`$(W(H^{CLS}))$ and their ground truth labels.

Self-training. We use the finetunned models with labeled data as *teachers* to generate pseudo soft labels on unlabeled data following (Du et al., 2021).

---

[1] We only consider datasets with training data size larger than 10K in GLUE benchmark.

[2] See more about FAQ 1 at https://gluebenchmark.com/faq

[3] We only calculate loss of the first token for words with multiple tokens after tokenization.

Dataset	Model	FT	TAPT	ST	TFS
SST-2	BERT_{base}	$87.3_{1.5}$	$88.5_{0.7}$ (+1.2)	$88.4_{1.0}$ (+1.1)	$\textbf{89.4}_{0.8}$ (**+2.1**)
	BERT_{large}	$89.0_{4.2}$	$90.7_{0.7}$ (+1.7)	$90.1_{4.2}$ (+1.1)	$\textbf{91.4}_{0.4}$ (**+2.4**)
QNLI	BERT_{base}	$79.1_{0.8}$	$82.0_{0.5}$ (+2.9)	$80.2_{0.7}$ (+1.1)	$\textbf{83.1}_{0.6}$ (**+4.0**)
	BERT_{large}	$82.6_{0.4}$	$83.2_{0.6}$ (+0.6)	$83.7_{0.4}$ (+1.1)	$\textbf{84.4}_{0.6}$ (**+1.8**)
MNLI	BERT_{base}	$57.3_{1.9}$	$58.8_{1.3}$ (+1.5)	$59.2_{2.1}$ (+1.9)	$\textbf{60.9}_{1.4}$ (**+3.6**)
	BERT_{large}	$66.4_{2.6}$	$67.6_{1.5}$ (+1.2)	$68.7_{2.2}$ (+2.3)	$\textbf{69.4}_{1.4}$ (**+3.0**)
QQP	BERT_{base}	$71.3_{0.8}$	$74.3_{0.8}$ (+3.0)	$72.3_{0.6}$ (+1.0)	$\textbf{75.1}_{0.9}$ (**+3.8**)
	BERT_{large}	$73.1_{1.7}$	$75.1_{0.9}$ (+2.0)	$74.2_{1.8}$ (+1.1)	$\textbf{76.1}_{0.9}$ (**+3.0**)
CoNLL 2003	BERT_{base}	$78.8_{1.1}$	$79.3_{1.6}$ (+0.5)	$81.6_{1.1}$ (+2.8)	$\textbf{82.2}_{1.3}$ (**+3.4**)
	BERT_{large}	$76.3_{2.4}$	$79.8_{1.0}$ (+3.5)	$79.4_{2.4}$ (+3.1)	$\textbf{82.2}_{1.1}$ (**+5.9**)
MultiWOZ 2.1	BERT_{base}	$75.6_{0.7}$	$79.8_{0.5}$ (+4.2)	$76.6_{0.8}$ (+1.0)	$\textbf{80.2}_{0.4}$ (**+4.6**)
	BERT_{large}	$77.7_{0.4}$	$81.4_{0.2}$ (+3.7)	$78.7_{0.6}$ (+1.0)	$\textbf{81.8}_{0.3}$ (**+4.1**)

Table 2: Results comparison (%) of finetuned baselines on labeled data (FT), TAPT, ST and TFS of BERT_{base} and BERT_{large} on six different datasets with 1% labeled data. Mean results along with their standard deviation in the subscript are listed and values inside the parentheses are gains over FT.

Pseudo labeled data are combined with original labeled data to trained student models by optimizing objective function in Equation 2. In the first round, *students* utilize the same pre-trained checkpoints as their teachers and in the following rounds, *students* inherit parameters from *teachers*. We set maximum rounds as 3 since we observe that setting a much larger round brings the same results or very marginal gains on both SST-2 and CoNLL 2003.

4.2 Main results

In this section, we simulate data scarcity scenarios for these mentioned datasets in Table 1 for both BERT_{base} and BERT_{large}. Specifically, for each dataset we randomly sample 1% training data as labeled corpus and left 99% as unlabeled data. Both labeled and unlabeled corpus is used as the input of TAPT while only unlabeled corpus is used for self-training. For all datasets, we randomly choose three data splits and have three different runs for each of them except BERT_{large} on CoNLL 2003 and MultiWOZ 2.1 to combat their instability by leveraging their results on development sets. In these two datasets, we use ten different runs for each data split on BERT_{large} and report corresponding test set results based on top three runs on development sets. Results are summarized in Table 2.

Comparison between TAPT and ST. TAPT and ST in both BERT_{base} and BERT_{large} models consistently outperform finetuned baseline results across six different datasets, demonstrating their effectiveness as semi-supervised methods on NLU tasks. However, TAPT has inconsistent results compared to ST. TAPT outperforms ST in SST-2, QQP and MultiWOZ 2.1 datasets but underperforms ST or only achieves comparable results for QNLI, MNLI and CoNLL 2003. These results indicate that they learn different representations from unlabeled data since they utilize data from different perspectives.

TFS shows strong *additive* gains over individual TAPT and ST. On QNLI, TFS on BERT_{base} improves 4% accuracy over finetuned baselines (FT), equal to the sum of gains from TAPT (+2.9%) and ST (+1.1%), and on BERT_{large} improves 1.8% accuracy, even slightly larger than the sum of gains from TAPT (+0.6%) and ST (+1.1%). On MNLI, TFS on BERT_{base} improves 3.6% accuracy, larger than the sum of gains from TAPT (+1.5%) and ST (+1.9%). Similar results on BERT_{base} also hold for CoNLL 2003. For results of other settings, improvements of TFS can also be well approximated by simply adding gains from corresponding TAPT and ST over FT. These consistent and significant results show that TAPT and ST are complementary to each other and TFS can effectively add their gains.

4.3 Varying size of labeled data

We have demonstrated the effectiveness of TFS on both BERT_{base} and BERT_{large} in six different datasets with 1% training data in section 4.2. We further explore different sizes of labeled data on six datasets in Table 1 with BERT_{base} model.

Specifically, on relatively simple dataset SST-2, we vary labeled data ratio as
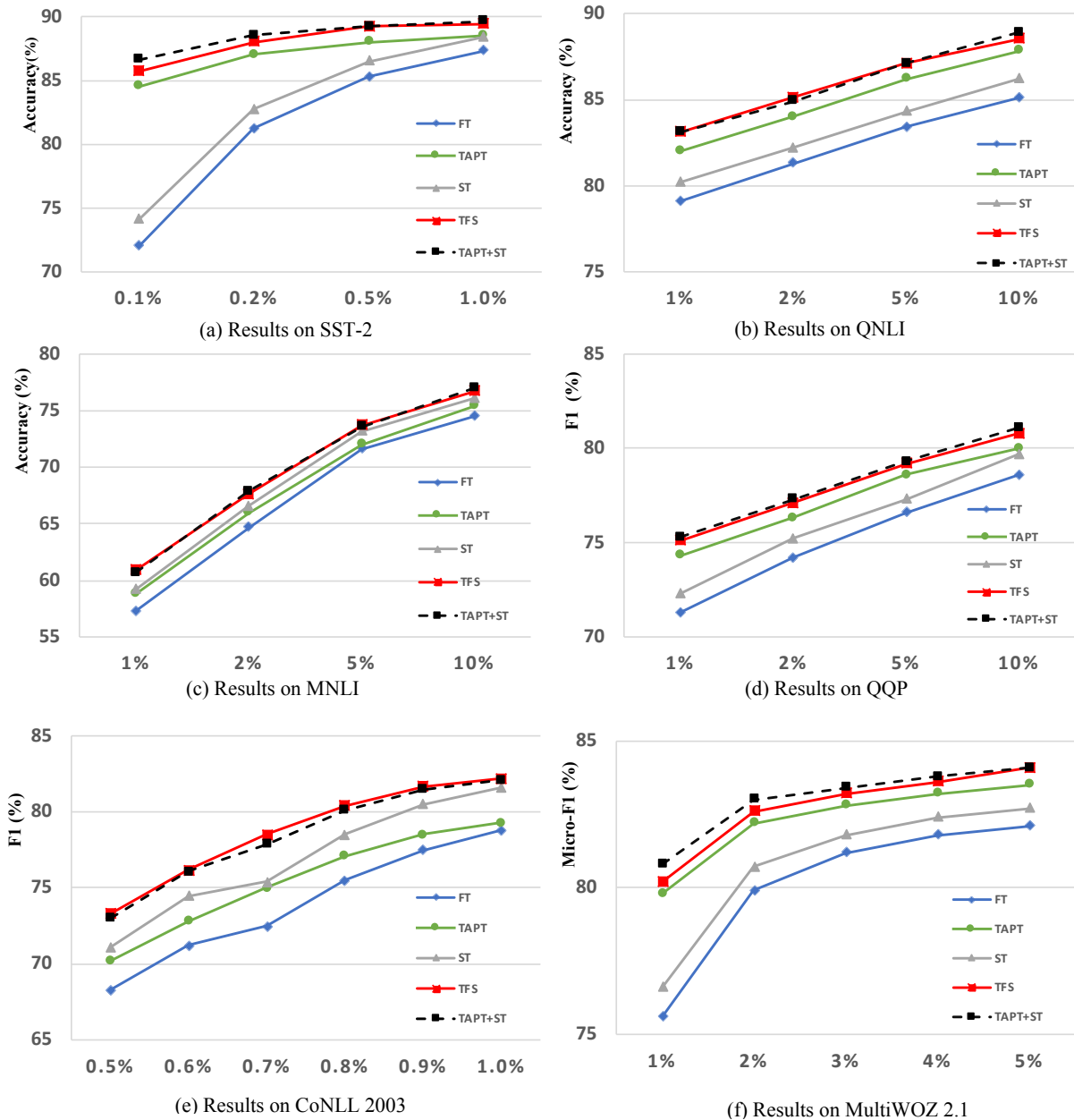
Figure 2: Results comparison (%) of FT, TAPT, ST and TFS with different sizes of labeled data on six datasets. TAPT+ST is ***not*** a method but references to demonstrate additive gains of TFS.

$\{0.1\%, 0.2\%, 0.5\%, 1.0\%\}$ and on more difficult QNLI, MNLI and QQP datasets, we vary their labeled ratios as $\{1\%, 2\%, 5\%, 10\%\}$. For CoNLL 2003, we explore labeled data ratio in $\{0.5\%, 0.6\%, 0.7\%, 0.8\%\ 0.9\%, 1.0\%\}$ and for MultiWOZ 2.1, we set labeled data ratio as $\{1\%, 2\%, 3\%, 4\%, 5\%\}$. Following previous settings, for each labeled data ratio among these six datasets, we randomly select 3 data splits and each data split has three different runs. Final average results for each data ratio are reported over these nine runs. To better measure additive property of

TFS, we introduce TAPT+ST in our results for references, which directly adds performance gains of TAPT and ST on FT.

Results of six different datasets among different sizes of labeled data are summarized in Figure 2. TAPT outperforms ST in SST-2, QNLI, QQP and MutiWOZ 2.1 datasets in various labeled data setups but underperforms it on MNLI and CoNLL 2003, indicating that TAPT and ST learn different representations from unlabeled data and have their pros and cons. However, TFS consistently and significantly outperforms TAPT and ST in all

scenarios among six different datasets and again proves its effectiveness over TAPT and ST alone. For example, in CoNLL 2003 with 0.5% labeled data, TFS has relative 4.4% and 3.1% improvement over TAPT and ST, respectively. More importantly, TFS overall has very similar results with `TAPT+ST` in various labeled data size of different datasets, which further strengthens that TFS protocol can yield strong additive gains over TAPT and ST.

4.4 Analysis

Given the promising results in the previous experiments, we aim to answer why TFS outperforms ST consistently and significantly. Indeed, the differences between TFS and ST lie in two aspects: (1) TFS uses initialization from the checkpoint of TAPT rather than original BERT as ST. (2) TFS utilizes pseudo labels generated from `TAPT finetuned models` while ST uses pseudo labels generated from `BERT finetuned models` (FT). To further investigate these two perspectives, we design a variant of original ST, **ST** with **T**APT **I**nitilization (STTI), which utilizes pseudo labels generated by BERT finetuned models as ST but is initialized with the same checkpoints from TAPT as TFS during the first round of self-training. The intermediate variant can help us better understand what makes TFS work. We run experiments on SST-2 with 0.1% and 1.0% labeled data for $BERT_{base}$ to compare ST, STTI and TFS. The results of STTI are obtained by running over the same three data splits as ST and TFS, and having three different runs for each data split. Results are averaged and summarized in Table 3.

Importance of initialization. Table 3 shows that STTI consistently outperforms ST in both 0.1% and 1% labeled setup. Comparing its difference with ST, we can conclude that its improvement over ST comes from its TAPT initialization. Results of MNLI and CoNLL 2003 in Figure 2 (c) and (e) also validate the importance of initialization. In these two datasets, although ST can consistently generate more accurate labels than TAPT finetuned models, meaning that it can match TAPT finetuned performance during self-training process, it still underperforms TFS in the end. These results again indicate the importance of initialization. Without TAPT as initialization, even if ST itself can outperform TAPT finetuned models, who are teachers of TFS in self-training process, but still at its end will

	FT	TAPT	ST	STTI	TFS
Init.	BERT	TAPT*	BERT	TAPT*	TAPT*
Pseud.	-	-	FT	FT	TAPT
Acc. (0.1%)	72.0	84.5	74.1	75.4	**85.7**
Acc. (1.0%)	87.3	88.5	88.4	88.8	**89.4**

Table 3: Results comparison of FT, TAPT, ST, STTI and TFS on SST-2 dataset. Rows with *Init.* and *Pseud.* show initialization and pseudo labeler of different models, respectively. Last two rows list accuracy with 0.1% and 1.0% labeled training data of these models. * represents models without finetuning on labeled data.

be left behind of TFS.

Importance of pseudo label correctness. Table 3 also shows that STTI underperforms TFS in both 0.1% and 1% labeled setup although both of them inherit the same parameters from TAPT. These results indicate that beyond initialization, accurate pseudo labels also matter for self-training process. STTI takes pseudo labels generated from BERT finetuned baselines (FT) that have more errors while TFS utilizes more accurate pseudo labels generated from TAPT finetuned models. Suffering from more incorrect pseudo labels in the beginning of self-training process, STTI may converge to a worse local optima than that of TFS. This is even more severe when labeled data is 0.1% and FT is left far behind of TAPT finetuned models, causing that STTI has 10.3% accuracy gap compared to TFS. These results prove the importance of accurate pseudo labels for self-training.

Combining these findings, we argue that TFS can outperform ST at least for two reasons: (1) it has a better initialization from TAPT compared to ST from BERT (2) it utilizes more accurate pseudo labels from TAPT finetuned models than ST.

5 Conclusion

In this paper, we demonstrate that TAPT and ST are complementary for NLU tasks with TFS by following TAPT → Finetuning → Self-training process. Our extensive experiments in various semi-supervised setups across six popular datasets show that they are not only complementary but also strongly *additive* with TFS protocol. We further show that TFS outperforms ST through (1) a better initialization from TAPT (2) more accurate predictions from TAPT finetuned models. We hope that TFS could serve as an important semi-supervised baseline for future NLP studies.

Acknowledgement

References

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.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020a. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. 2020b. Big self-supervised models are strong semi-supervised learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 22243–22255. Curran Associates, Inc.

Zihan Chen, Hongbo Zhang, Xiaoji Zhang, and Leqi Zhao. 2018. Quora question pairs.

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.

Jingfei Du, Edouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Veselin Stoyanov, and Alexis Conneau. 2021. Self-training improves pre-training for natural language understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5408–5418, Online. Association for Computational Linguistics.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 422–428, Marseille, France. European Language Resources Association.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*, volume 1. MIT press Cambridge.

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.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2020. Revisiting self-training for neural sequence generation. In *International Conference on Learning Representations*.

Geoffrey E. Hinton, Oriol Vinyals, and J. Dean. 2015. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531.

Jacob Kahn, Ann Lee, and Awni Hannun. 2020. Self-training for end-to-end speech recognition. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7084–7088.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, D. Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36:1234 – 1240.

Shiyang Li, Semih Yavuz, Kazuma Hashimoto, Jia Li, Tong Niu, Nazneen Rajani, Xifeng Yan, Yingbo Zhou, and Caiming Xiong. 2021. Coco: Controllable counterfactuals for evaluating dialogue state trackers. In *International Conference on Learning Representations*.

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

Subhabrata Mukherjee and Ahmed Hassan Awadallah. 2020. Uncertainty-aware self-training for text classification with few labels. *ArXiv*, abs/2006.15315.

Alec Radford, Jeffrey Wu, Rewon Child, Luan David, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Olga Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Zhiheng Huang, A. Karpathy, A. Khosla, Michael S. Bernstein, A. Berg, and Li Fei-Fei. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252.

H. Scudder. 1965. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Qianru Sun, Xinzhe Li, Yaoyao Liu, S. Zheng, Tat-Seng Chua, and B. Schiele. 2019. Learning to self-train for semi-supervised few-shot classification. In *NeurIPS*.

Philippe Thomas. 2009. Semi-supervised learning by olivier chapelle, bernhard schölkopf, and alexander zien (review). *IEEE Transactions on Neural Networks*, 20:542.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. In *NeurIPS*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Yaqing Wang, Subhabrata Mukherjee, Haoda Chu, Yuancheng Tu, Ming Wu, Jing Gao, and Ahmed Hassan Awadallah. 2021. Adaptive self-training for neural sequence labeling with few labels.

Colin Wei, Kendrick Shen, Yining Chen, and Tengyu Ma. 2021. Theoretical analysis of self-training with deep networks on unlabeled data. In *International Conference on Learning Representations*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. 2020. 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.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. 2020. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Qiantong Xu, Alexei Baevski, T. Likhomanenko, Paden Tomasello, Alexis Conneau, Ronan Collobert, Gabriel Synnaeve, and Michael Auli. 2020. Self-training and pre-training are complementary for speech recognition. *ArXiv*, abs/2010.11430.

Xiaojin Zhu and Andrew B. Goldberg. 2009. *Introduction to Semi-Supervised Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. 2020. Rethinking pre-training and self-training. In *NeurIPS*.