Learning Instructions with Unlabeled Data for Zero-Shot Cross-Task Generalization

Yuxian Gu, Pei Ke, Xiaoyan Zhu, Minlie Huang[†]
The CoAI group, Tsinghua University, Beijing, China
Institute for Artificial Intelligence, State Key Lab of Intelligent Technology and Systems,
Beijing National Research Center for Information Science and Technology,
Department of Computer Science and Technology, Tsinghua University, Beijing, China
guyx21@mails.tsinghua.edu.cn, kepei1106@outlook.com
{zxy-dcs,aihuang}@tsinghua.edu.cn

Abstract

Training language models to learn from human instructions for zero-shot cross-task generalization has attracted much attention in NLP communities. Recently, instruction tuning (IT), which fine-tunes a pre-trained language model on a massive collection of tasks described via human-craft instructions, has been shown effective in instruction learning for unseen tasks. However, IT relies on a large amount of human-annotated samples, which restricts its generalization. Unlike labeled data, unlabeled data are often massive and cheap to obtain. In this work, we study how IT can be improved with unlabeled data. We first empirically explore the IT performance trends versus the number of labeled data, instructions, and training tasks. We find it critical to enlarge the number of training instructions, and the instructions can be underutilized due to the scarcity of labeled data. Then, we propose **U**nlabeled **D**ata Augmented **I**nstruction **T**uning (UDIT) to take better advantage of the instructions during IT by constructing pseudo-labeled data from unlabeled plain texts. We conduct extensive experiments to show UDIT's effectiveness in various scenarios of tasks and datasets. We also comprehensively analyze the key factors of UDIT to investigate how to better improve IT with unlabeled data. The code is publicly available at https://github.com/thu-coai/UDIT.

1 Introduction

The instruction learning paradigm (Weller et al., 2020), where language models learn from human instructions to perform unseen tasks in zero-shot scenarios, has received increasing attention recently. Compared to conventional machine learning paradigms that mainly learn from data examples, instruction learning requires models to complete tasks based on the understanding of human-written task descriptions without task-specific data, which is closer to general AI systems. For instance, in summarization tasks, a model is only given an explicit instruction "Summarize the following article in brief:" and an article to generate the corresponding summary. To realize instruction learning, recent works such as FLAN (Wei et al., 2022) and T0 (Sanh et al., 2022) propose instruction tuning (IT), which fine-tunes pre-trained language models (PLMs) (Han et al., 2021) on a large collection of tasks with human-annotated data specified in descriptive instructions. Through IT, PLMs learn to follow the human-written instructions to complete the corresponding tasks, which enables them to perform instruction learning in unseen tasks.

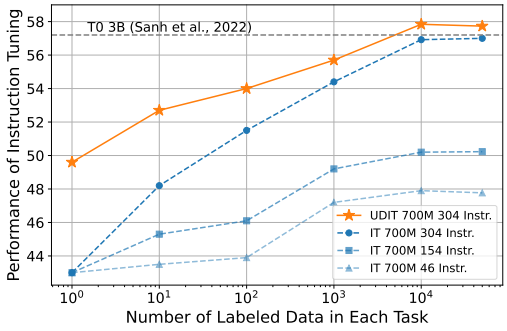


Figure 1: The performance of IT and UDIT with respect to the instruction numbers and labeled data amounts. We follow Sanh et al. (2022) to fine-tune a 700M PLM and then test its zero-shot generalization ability on unseen tasks. The x-axis represents the number of labeled samples in each training task, and the y-axis is the average performance on evaluation tasks. We control the instruction numbers by gradually adding training tasks.

An intuitive way to boost the performance of IT is to increase the number of training instructions and data examples. As shown in Figure 1, the number of training instructions largely determines the best performance of IT, and the corresponding human-annotated data should be also sufficient for the model to learn these instructions well. However, the amount and domain diversity of labeled data in

[†] Corresponding author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 1617 - 1634
December 7-11, 2022

different tasks vary greatly. In practice, many low-resource tasks lack sufficient multi-domain human-annotated examples. This can lead to easy overfitting to specific domains or examples when learning the corresponding instructions, which affects the zero-shot performance in instruction learning.

Introducing unlabeled data is a common approach to alleviating the data scarcity problem in supervised learning (Brown et al., 2020; Xie et al., 2020; Du et al., 2021) because large-scale unlabeled plain texts are much easier to access. However, we argue that their benefit to IT is still inconclusive. This is because IT is much more challenging, requiring learning the mapping between human instructions and task semantics rather than that between samples and labels in a single task.

Therefore, in this work, we investigate incorporating unlabeled data into instruction learning. We focus on the two questions: (1) *Is it possible to perform IT with unlabeled plain texts when there are few or even no human-annotated data?* and (2) *How to better use unlabeled plain texts to improve IT for zero-shot cross-task generalization?*

To study (1), we propose **U**nlabeled **D**ata Augmented **I**nstruction **T**uning (UDIT) to effectively use unlabeled data to help instruction learning. Specifically, we construct pseudo-labeled data from unlabeled plain texts according to task instructions. The pseudo-labeled data which enlarge training samples and diversify data domains help to learn the meanings of the corresponding task instructions better. We test UDIT under various scenarios of training tasks and labeled data to verify that learning instructions from unlabeled data is possible.

To study (2), we compare UDIT with previous methods to show its superior performance in using unlabeled data. We also conduct extensive experiments to reveal the underlying factors to the success of UDIT. Specifically, our contributions are summarized as follows:

- We introduce UDIT, a training framework that incorporates unlabeled data into instruction tuning for zero-shot cross-task generalization.
- Through UDIT, we empirically verify that PLMs can learn to follow human-written instructions with unlabeled data when there are few or even no annotated samples.
- We show that UDIT is a significantly better way to use unlabeled data to improve instruction tuning, making a 700M PLM with UDIT outperform the 3B counterpart based on IT.
- We comprehensively analyze the key factors of UDIT and give some insights into using unlabeled data to improve instruction learning.

2 Related Works

Instruction Learning. Recently, large PLMs like GPT-3 (Brown et al., 2020) have shown promising performance in learning from human instructions to solve tasks in few-shot and zero-shot scenarios (Liu et al., 2021). Several works propose benchmarks (Weller et al., 2020; Efrat and Levy, 2020; Mishra et al., 2022; Finlayson et al., 2022) to evaluate instruction learning for zero-shot cross-task generalization (Ye et al., 2021). To enhance instruction understanding, many works adopt IT, which fine-tunes PLMs on massive task clusters described by instructions in a multi-task fashion, such as FLAN (Wei et al., 2022), T0 (Sanh et al., 2022), ZeroPrompt (Xu et al., 2022a), and InstructGPT (Ouyang et al., 2022). These models show superior zero-shot performance on unseen tasks. To better understand IT and zero-shot learning of PLMs, Wang et al. (2022) compares different model architectures and training objectives. Some works also incorporate unlabeled data to improve the zero-shot performance of IT (Zhou et al., 2022; Lin et al., 2022). But they assume the existence of unlabeled samples in evaluation tasks, while we only use plain texts, which is more in line with the zero-shot cross-task evaluation scenario.

Semi-Supervised Learning. Semi-supervised learning adopts unlabeled data to improve supervised learners (Chapelle et al., 2009). Many previous works use consistency training to regularize model predictions (Bachman et al., 2014; Rasmus et al., 2015; Xie et al., 2020). Self-training (Scudder, 1965; Du et al., 2021) is also widely used, which assigns synthetic labels to unlabeled data with a teacher model. These data are then used to train the student model. However, these methods typically assume the availability of unannotated task-specific data while we focus on using task-agnostic plain texts, which is more practical.

Self-Supervised Training in NLP. Training with self-supervised tasks is also related to our method, which helps models obtain versatile knowledge from large-scale plain texts and boosts the model

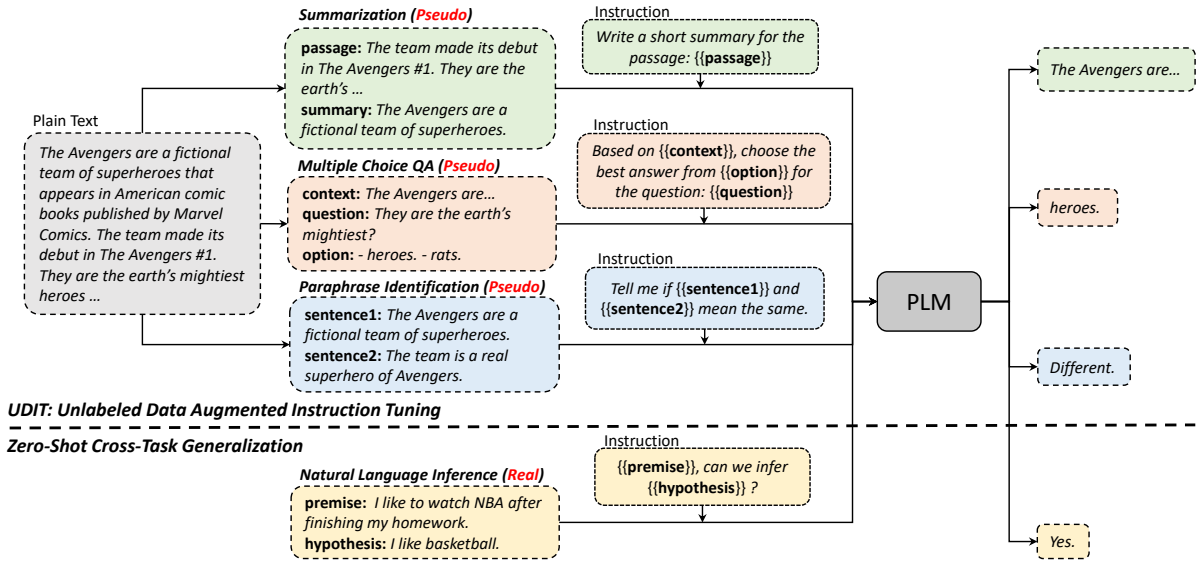


Figure 2: An Example of UDIT. The pseudo-labeled data of Summarization, Multiple-Choice QA, and Paraphrase Identification are constructed from plain texts. Then the corresponding instructions are applied to these samples. We fine-tune the PLM on these samples in a multi-task text-to-text language modeling objective and test it for zero-shot cross-task generalization where the evaluation task (Natural Language Inference) is unseen during training.

performance (Devlin et al., 2019; Radford et al., 2019; Raffel et al., 2020; Lewis et al., 2020; Lan et al., 2020; Fang et al., 2020). Some works also find that carefully designed self-supervised tasks can bring further improvement to low-resource tasks (Bansal et al., 2020; Gu et al., 2022; Chen et al., 2022). However, conventional self-supervised tasks are designed independent of human instructions (Aroca-Ouellette and Rudzicz, 2020), while tasks in UDIT match the instruction semantics closely, which is crucial to instruction learning for zero-shot cross-task generalization.

3 Method

3.1 Background

In this section, we first give a formal description of IT. We define a "task" as a pair (D, I), where D is the task-specific dataset, and I is a set of instructions describing the task. We assume that the tasks can be divided into n clusters $\mathcal{T} = \{T_1, T_2, \cdots, T_n\}$ according to the task similarities, where the i^{th} cluster $T_i = \{(D_1, I_1), (D_2, I_2), \cdots, (D_{k_i}, I_{k_i})\}$ contains k_i tasks. For example, in the cluster "Multiple-Choice QA", a data sample typically consists of a passage, a question, several answer options, and the answer. As shown in Figure 2, the instructions serve as templates to convert the inputs and outputs to natural texts and formulate all tasks into text-to-text language modeling problems. In IT, a PLM is first fine-tuned on several clusters $\mathcal{T}_{\text{Train}} \subsetneqq \mathcal{T}$ in a multi-task fashion. Then the model is evaluated on the tasks in novel clusters $\mathcal{T}_{\text{Test}} = \mathcal{T} \backslash \mathcal{T}_{\text{Train}}$ with instructions only, as shown in the "Zero-Shot Cross-Task Generalization" part of Figure 2.

In this paper, we mainly follow the settings of T0 (Sanh et al., 2022), including the training tasks, instructions, and the split of task clusters. However, our findings can also be applied to other scenarios. T0 is a representative model instruction-tuned on 8 task clusters based on the pre-trained T5 model (Raffel et al., 2020) and tested on 4 task clusters. The instructions are collected from the Public Pool of Prompts (P3) (Bach et al., 2022) which contains thousands of crowdsourced instructions.

3.2 Overview

Figure 2 shows an overview of UDIT. To better learn the instructions in $\mathcal{T}_{\text{Train}}$, we construct pseudo-labeled data from the unlabeled plain texts according to the meaning of the instructions in each task cluster $T_i \in \mathcal{T}_{\text{Train}}$. The plain texts are a mixture of multi-domain corpora, including BookCorpus (Zhu et al., 2015), CC-News (Sebastian, 2016), OpenWebText (Gokaslan et al., 2019), Wikipedia (Foundation, 2022), and IMDB Review (Maas et al., 2011), totaling about 37.2G. The details of these corpora are shown in Appendix A. The constructing process is based on heuristic rules, widely used

NLP toolkits like NLTK[1], and basic data augmentation techniques like back-translation (Sennrich et al., 2016). Then, we apply the instructions in T_i to the pseudo-labeled samples and fine-tune the PLM on pseudo-labeled and labeled samples with a multi-task language modeling objective. Although the pseudo-labeled data are constructed at the level of task clusters rather than single tasks, we find they match the meanings of most instructions in the corresponding cluster due to the task similarities. Note that we do not assume the existence of labeled data during the constructing process, which means that UDIT is applicable under various settings with or without labeled data. The following section briefly introduces the pseudo-labeled data construction for the 8 task clusters in T0. We provide some examples of the constructed data in Appendix D.

3.3 Constructing Pseudo-Labeled Data

Multiple-Choice QA (MCQA). The sample in MCQA consists of a passage, a related question, an answer, and several options. Given a plain-text document, we design two methods to construct pseudo-labeled data: (1) We first randomly replace one noun in a randomly selected sentence with a "_" symbol. Then, we add a "?" mark to the end of the sentence to form a question. We treat the texts before the sentence as the passage and the replaced word as the answer. The options are sampled from the words with the same part of speech as the answer. (2) We observe that many questions are naturally followed by its answer in our corpus. Therefore, we search for questions in the document and treat previous texts as the passage and the following sentence as the answer. The options are sampled from the sentences after the answer.

Extractive QA (EXQA). EXQA aims to answer the questions using the phrases in the given passages. We mainly follow Fabbri et al. (2020) which first selects entities in the plain-text documents as the answers and uses templates to convert the sentences containing the answers to questions.

Close-Book QA (CBQA). CBQA is similar to EXQA except for the absence of the passage. Therefore, we use the question-answer pair from the pseudo-labeled data of EXQA.

Sentiment (SENT). SENT requires identifying the sentiment labels of given texts. We use a keyword-based sentiment analyzer in NLTK to annotate sentiment labels. To improve the label quality, we only construct pseudo-labeled data from the IMDB Review corpus for this cluster.

Topic Classification (TC). TC requires finding proper topic labels for input passages. We notice that many URLs of the CC-News corpus contain the topic of passages. Therefore, we devise heuristic rules to extract topic labels from the URLs to build the pseudo-labeled data. We first split the URL by "/" and search for topic words from left to right. We stop at the first string that is composed of English letters, shorter than 20 characters, and not in ["news", "en", "story", "us", "articles", "local", "english", "tag", "post"]. Then, we choose the most frequent 14 strings as the topic labels and the corresponding passages as the inputs.

Structure-to-Text (S2T). S2T requires generating natural sentences that describe input structural data like graphs. Since the input data are usually linearized as word sequences in the instructions, we adopt a keyword-to-text generation task that takes a random subset of the notional words in a sentence as the input and the sentence as the output.

Summarization (SUM). For summarization, we adopt Leading Sentence Generation (LSG) and Gap Sentence Generation (GSG) from Liu et al. (2022). LSG takes the title of a passage as the summary and the body as the input. GSG treats the sentence overlapping other document parts the most as the summary and the remaining sentences as the input.

Paraphrase Identification (PARA). PARA aims to identify whether two sentences have the same meaning. Given a sentence s in the plain texts, we add word-level perturbation to s to get s_{pert}. We consider two kinds of perturbations: (1) Randomly choosing a word and replacing it with its antonym via NLTK. (2) Picking out nouns in the sentence via NLTK and shuffling their order. Then we get $\widetilde{s}$ and $\widetilde{s}_{\text{pert}}$ by adopting back-translation to s and s_{pert}, respectively. We treat $(s, \widetilde{s})$ as the positive pair and $(s, \widetilde{s}_{\text{pert}})$ as the negative pair.

4 Experiment

4.1 Setup

Settings. We consider three scenarios in which unlabeled data can be utilized to enhance instruction learning: (1) **No Labeled Data**, where only the instructions for each task are available. (2) **Few**

[1] https://www.nltk.org/

Labeled Data, where only a small part of the labeled data is available. (3) **Full Labeled Data**, where all the labeled data are available during IT.

Datasets. Following Sanh et al. (2022), we use 8 task clusters as $\mathcal{T}_{\text{Train}}$, which contains 36 datasets and 304 instructions. $\mathcal{T}_{\text{Test}}$ contains 6 task clusters consisting of 9 text classification tasks[2] and 2 language generation tasks. Detailed data information can be found in Appendix A.

Training and Evaluation Details. For computational efficiency, we conduct our experiments mainly based on a 700M T5 model. We mix the labeled and pseudo-labeled data for multi-task fine-tuning. Unless specified otherwise, we use at most 10k labeled/pseudo-labeled samples for each task because we find more samples bring little improvement. We choose the best checkpoint on the merged validation splits of datasets in $\mathcal{T}_{\text{Train}}$ for evaluation. More hyper-parameter details are shown in Appendix B. In evaluation, we report the mean (Section 4.2) and median (Appendix C.2) of the performance across different instructions on the validation set of each task in $\mathcal{T}_{\text{Test}}$. For the multiple-choice tasks, we select the option with the highest log-likelihood (Brown et al., 2020) as the answer.

Baselines. We consider the following baselines:
(1) *Direct Zero-Shot* (**DirectZS**): The PLM is directly evaluated on $\mathcal{T}_{\text{Test}}$ without fine-tuning.
(2) *Vanilla Instruction Tuning* (**Vanilla-IT**): The model is instruction-tuned on the labeled data in $\mathcal{T}_{\text{Train}}$, which stays the same with Sanh et al. (2022).
(3) *Self-Supervised Training*: Besides the labeled data in IT, the model is also tuned on our unlabeled plain-text corpus with the language modeling objective (**ExtraLM**) or the four self-supervised objectives proposed in Chen et al. (2022) (**SelfSup-IT**). The proportion of training samples to our pseudo-labeled samples is 1:1.
(4) *Data Augmentation* (**DataAug-IT**): For the tasks with few labeled data, we perform back-translation and augment the labeled data to twice as large (Xu et al., 2022b).

4.2 Results

4.2.1 No Labeled Data

Table 1 shows the results where no labeled data are available, from which we have 3 observations.

[2]The ANLI task contains datasets of 3 versions that share the same instructions. We only evaluate our model on the R1 version for simplicity.

First, all methods that use unlabeled data (ExtraLM, SelfSup-IT, and UDIT) outperform DirectZS, suggesting that PLMs can learn to follow instructions for zero-shot cross-task generalization with unlabeled data when human-labeled samples are absent.

Second, among different methods using unlabeled data, self-supervised training only brings marginal improvement, while UDIT boosts the performance largely on most tasks. This indicates that using unlabeled data to improve instruction learning is non-trivial. Simple self-supervised tasks cannot reflect the characteristics of human instructions, while UDIT directly helps the PLM learn the mapping between instructions and task semantics.

Third, UDIT can be combined with self-supervised training when we mix the training samples augmented by these two methods. The row "UDIT + SelfSup-IT" achieves the best average performance, which means that these two methods are complementary in this scenario.

4.2.2 Few Labeled Data

In this section, we study a more practical scenario where only a small set of labeled data are available for IT and show the results in Table 2. We explore three different data scarcity settings. "Few Tasks" simulates the setting where only a few task clusters have enough labeled data. Here, we choose EXQA as the data-sufficient cluster, where each task contains 10K samples. The results of other choices are shown in Appendix C.1. "Few Datasets" means only 10% human-labeled datasets exist in each task cluster. And the "Few Samples" block shows the results where IT is performed on all task clusters, but each dataset contains only 100 samples. Note that data augmentation (DataAug-IT) can only be applied to the "Few Samples" setting because there are no source data for back-translation in other settings. UDIT adds the pseudo-labeled data to those data-scarce tasks to enhance instruction learning.

Our findings from Table 2 are as follows:

1. SelfSup-IT and DataAug-IT fail to bring significant improvement over Vanilla-IT. It is probably because the self-supervised tasks do not use instructions, and the augmented data are too similar to the source samples.

2. UDIT performs the best on average under all the three settings, indicating that learning instruction semantics and training on sufficient

Method	Size	Coref.		NLI			Sentence Comp.			WSD	Avg.
		WSC	Wino.	CB	RTE	ANLI	COPA	H-Swag	Story	WiC	
DirectZS	700M	50.3	50.9	32.8	48.4	32.8	42.3	26.4	52.7	50.9	43.0
DirectZS	3B	49.5	50.9	31.1	47.9	32.5	46.4	25.7	55.3	50.5	43.3
DirectZS	11B	**54.1**	50.6	34.3	53.0	32.9	54.9	27.0	48.2	50.3	45.0
ExtraLM	700M	52.0	51.5	29.3	52.3	32.3	48.7	23.9	51.6	50.5	43.6
SelfSup-IT	700M	50.5	**54.0**	41.9	53.0	32.9	50.7	24.0	51.4	50.2	45.4
UDIT	700M	53.6	52.9	44.2	54.0	**33.0**	59.4	**29.4**	67.4	**53.0**	49.6
UDIT + SelfSup-IT	700M	53.6	53.1	**45.1**	**56.6**	32.4	**59.5**	28.1	**71.8**	52.0	**50.2**

Table 1: The zero-shot cross-task generalization results of classification tasks in the "No Labeled Data" scenario. We report the average accuracy of different testing instructions on the official validation set of each dataset. We reprint the DirectZS scores of the 11B model from Sanh et al. (2022). The best results on each dataset are in **boldface** and the second-best results are underlined.

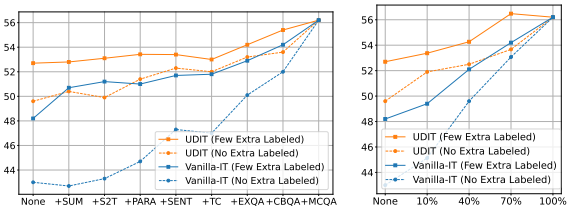
Setting	Method	Coref.		NLI			Sentence Comp.			WSD	Avg.
		WSC	Wino.	CB	RTE	ANLI	COPA	H-Swag	Story	WiC	
Few Tasks	Vanilla-IT	51.4	53.0	50.3	53.7	31.5	**60.9**	**28.3**	51.6	50.3	47.9
	SelfSup-IT	50.6	55.2	50.4	53.2	**33.1**	57.2	27.7	52.9	50.4	47.8
	UDIT	**54.4**	54.1	**57.5**	**57.2**	32.2	**60.9**	27.6	66.2	52.0	**51.3**
	UDIT + SelfSup-IT	51.7	**55.8**	48.0	56.2	32.9	60.4	26.3	**70.0**	**52.1**	50.4
Few Datasets	Vanilla-IT	49.8	51.3	32.8	50.5	**32.9**	59.0	25.8	53.1	51.0	45.1
	SelfSup-IT	50.6	54.1	41.9	54.0	**32.9**	55.0	26.4	59.0	51.4	47.2
	UDIT	**51.1**	51.2	48.3	**65.9**	31.8	**66.8**	**29.4**	71.0	51.6	**51.9**
	UDIT + SelfSup-IT	50.8	**55.1**	**48.8**	57.9	32.0	60.5	27.4	**75.9**	**53.2**	51.3
Few Samples	Vanilla-IT	52.0	51.6	54.2	58.2	31.2	65.9	26.9	71.5	51.7	51.5
	SelfSup-IT	50.8	53.3	47.2	63.8	31.7	60.9	26.5	73.1	54.0	51.2
	DataAug-IT	50.9	52.0	**55.1**	56.3	31.2	67.2	28.1	72.0	53.0	51.8
	UDIT	**53.8**	52.9	51.0	**64.4**	31.3	68.4	**30.5**	80.0	54.2	**54.0**
	UDIT + SelfSup-IT	51.6	**54.3**	51.8	59.9	**32.4**	68.6	27.9	**81.0**	**55.4**	53.6
	UDIT + DataAug-IT	53.4	51.6	48.5	61.3	31.5	**70.7**	29.7	79.4	53.0	53.2

Table 2: The scores of classification accuracy in the "Few Labeled Data" scenario. In the "Few Tasks" block, IT is performed on EXQA, and UDIT adds pseudo-labeled data to other clusters. In the "Few Datasets" block, IT is performed on 10% tasks from each cluster, and UDIT adds pseudo-labeled data to the remaining tasks. In the "Few Samples" block, IT is performed on 100 samples from each dataset, and UDIT adds pseudo-labeled data to all tasks. All experiments are based on the 700M model. The best scores in each block are in **boldface**.

diverse data are crucial to zero-shot cross-task generalization of PLMs.

3. Unlike the observations in Section 4.2.1, the benefit of combining self-supervised tasks with UDIT vanishes with the existence of the few labeled data.

We also evaluate Vanilla-IT and UDIT when the number of data-sufficient tasks varies. In Figure 3(a), we incrementally add task clusters containing full labeled data. And in Figure 3(b), we gradually increase the proportion of data-sufficient tasks in each cluster. In these processes, other tasks are considered data-scarce, containing 100 (Few Extra Labeled) or no (No Extra Labeled) labeled samples to simulate the situation when both data-sufficient and data-scarce tasks exist. UDIT enhances the data-scarce tasks with pseudo-labeled data.[3]



(a) Task Clusters (b) Task Proportions

Figure 3: The performance trend when number of data-sufficient tasks varies. The y-axis means the average classification results. "Few/No Extra Labeled" means there are few/no labeled data in the data-scarce tasks.

[3]We take "+TC" in Figure 3(a) as an example. All tasks in SUM, S2T, PARA, SENT, and TC have enough data (10K

Method	Size	Coref.		NLI			Sentence Comp.			WSD	Avg.
		WSC	Wino.	CB	RTE	ANLI	COPA	H-Swag	Story	WiC	
Vanilla-IT	700M	53.0	53.4	53.0	69.2	**34.8**	75.2	27.4	89.6	50.7	56.9
SelfSup-IT	700M	49.2	**56.4**	56.5	67.7	34.1	75.7	26.1	90.4	51.0	56.3
UDIT	700M	**55.6**	53.5	**57.8**	**73.1**	33.5	74.8	**30.1**	88.7	53.5	**57.8**
UDIT + SelfSup-IT	700M	53.4	55.7	54.2	65.5	33.6	**76.7**	28.5	**90.6**	**56.6**	57.2
Vanilla-IT	3B	61.2	52.5	42.6	73.7	36.3	79.2	27.4	89.9	52.0	57.2
Vanilla-IT	11B	61.4	59.9	70.1	80.8	43.6	90.0	33.6	92.4	56.6	65.4

Table 3: The scores of classification accuracy in the "Full Labeled Data" scenario. The **boldfaced** scores are the best results among the methods based on the 700M model. We reprint the results of the 11B model from Sanh et al. (2022) but re-evaluate the 3B model because the numerical values of this size are not provided in Sanh et al. (2022).

By comparing the solid and dashed lines, we conclude that training on more tasks and instructions is beneficial, even if some tasks contain only 100 samples, which is consistent with Figure 1. Also, by comparing the orange and blue lines, we can see that UDIT leads to further improvement when applied to those data-scarce tasks, regardless of the existence of the 100 labeled samples. From Figure 3(a), we notice the performance drop when adding TC, which matches the observation in Xu et al. (2022a) that not all the task clusters are helpful. But in general, the performance of IT has a positive correlation with the task number and diversity.

4.2.3 Full Labeled Data

When the labeled data are sufficient, we can also mix them with the pseudo-labeled data to perform IT. Table 3 shows that adding 10K pseudo-labeled data can improve the IT performance, making our 700M model outperform the 3B model with Vanilla-IT. But increasing labeled data to 50k only leads to little further improvement (Figure 1). This indicates that the pseudo-labeled data do not merely contribute to the data amount per task. We conjecture that these samples also help avoid overfitting to the domain of specific datasets during IT, owing to the domain diversity of unlabeled corpora, which will be further analyzed in Section 5.2.

4.2.4 Language Generation Tasks

We also test the instruction-tuned models on two language generation tasks. From Table 4, we observe the similar phenomenon that UDIT improves IT the most in all scenarios. We also notice that self-supervised training is more beneficial to language generation than classification. This is likely because the self-supervised tasks include Next Sentence Generation and Next Phrase Generation (Chen et al., 2022), which resemble the generation tasks used in the zero-shot evaluation.

4.3 Discussion

Based on the results in Section 4.2, we conclude that UDIT is effective under all three settings on both classification and generation tasks.

We observe that UDIT brings larger improvements to Natural Language Inference (NLI) and Sentence Completion (Sentence Comp.) compared to Coreference Resolution (Coref.) and Word Sense Disambiguation (WSD), which resembles the phenomenon of Vanilla-IT. We suspect that our training tasks are mostly sentence-level, while the tasks in Coref. and WSD are word-level. Although IT enables cross-task generalization for PLMs, it is still challenging to generalize from sentence-level tasks to word-level tasks. This also emphasizes the importance of using instructions from more diverse tasks for IT.

Besides, we also find that the performance variance across different testing instructions is high on some tasks (Figure 5 in Appendix C.3), which is consistent with the observations in Sanh et al. (2022). Reducing the sensitivity of PLMs to prompts and instructions has been largely discussed in previous literature (Zhao et al., 2021; Zhou et al., 2022). Most of these methods are applicable to our settings.

5 Analysis

5.1 Effect of Instruction Tuning

IT brings two effects: (1) helping the model get familiar with the input form containing human instructions and (2) enabling the model to learn the mapping between the instructions and task semantics. To differentiate these effects, we construct

samples). Each task in EXQA, CBQA, and MCQA contains 100 (Few Extra Labeled) or no (No Extra Labeled) labeled samples. UDIT is applied only to EXQA, CBQA, and MCQA.

Setting	Method	SQuAD	Story	Avg.
None	DirectZS	8.6	8.7	8.6
	ExtraLM	13.3	15.2	14.2
	SelfSup-IT	12.3	15.3	13.8
	UDIT	14.9	15.7	15.3
	UDIT + SelfSup-IT	**16.3**	**17.2**	**16.8**
Few Tasks	Vanilla-IT	25.9	**16.3**	21.1
	SelfSup-IT	25.3	15.4	20.4
	UDIT	25.6	15.9	20.8
	UDIT + SelfSup-IT	**27.3**	15.2	**21.2**
Few Datasets	Vanilla-IT	17.5	14.5	16.0
	SelfSup-IT	19.5	15.4	17.4
	UDIT	**27.0**	16.4	**21.7**
	UDIT + SelfSup-IT	25.5	**16.7**	21.1
Few Samples	Vanilla-IT	22.9	15.3	19.1
	SelfSup-IT	25.1	15.9	20.5
	DataAug-IT	24.0	14.4	19.2
	UDIT	25.4	15.4	20.4
	UDIT + SelfSup-IT	**25.6**	**16.0**	**20.8**
	UDIT + DataAug-IT	24.4	14.7	19.5
Full	Vanilla-IT	28.5	15.3	21.9
	SelfSup-IT	**30.3**	15.2	22.8
	UDIT	28.6	**19.2**	**23.9**
	UDIT + SelfSup-IT	29.4	17.5	23.4

Table 4: Results on language generation tasks. We report the average Rouge-L score across different testing instructions. "None" represents the "No Labeled Data" scenario. "Few Tasks/Datasets/Samples" means the three settings in "Few Labeled Data" discussed in Section 4.2.2. "Full" represents "Full Labeled Data". All experiments are based on the 700M model.

tasks that do not match the instructions by randomly setting the labels in the labeled and pseudo-labeled samples. As shown in Table 5, although we randomize the labels, the results of IT are still slightly better than No IT, suggesting that the input form matters. Furthermore, a large performance gap exists between the random and correct labels, indicating that the model learns the instruction-task mapping in addition to the instruction form.

5.2 Effect of Domain Diversity

As described in Section 3.2, our unlabeled data are a mixture of multi-domain plain-text corpora. To investigate the domain diversity effect, we construct pseudo-labeled data only from Wikipedia for the task clusters other than SENT and TC, where we still use IMDB Review and CC-News. This ensures that each cluster contains a single domain. We also maintain the same amount of training samples as the multi-domain circumstance. From the

Method	Labeled	Pseudo	CLS	GEN
No IT	-	-	43.0	8.6
Vanilla-IT	Random	-	44.8	8.6
	Correct	-	56.9	15.3
UDIT	-	Random	43.9	13.8
	-	Correct	49.6	21.9
	Correct	Random	52.3	22.1
	Correct	Correct	**57.8**	**23.9**

Table 5: The results on comparing whether we assign random labels to labeled and pseudo-labeled data. "-" means that we do not use the corresponding data. "CLS" and "GEN" stand for the average performance on the classification and generation tasks, respectively.

Setting	Method	CLS	GEN
None	DirectZS	43.0	8.6
	UDIT (Single)	48.3	14.6
	UDIT (Multiple)	**49.6**	**15.3**
Full	Vanilla-IT	56.9	21.9
	UDIT (Single)	57.0	21.1
	UDIT (Multiple)	**57.8**	**23.9**

Table 6: The results when we use the pseudo-labeled data of single or multiple domains for UDIT. "CLS" and "GEN" stand for the average performance on classification and generation tasks, respectively. "None/Full" represents the "No/Full Labeled Data" scenario.

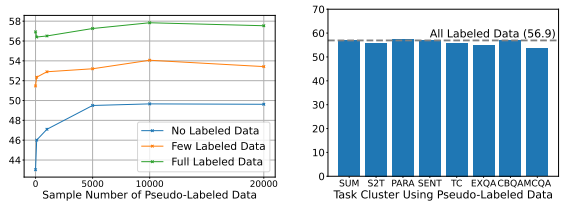
results in Table 6, we observe that reducing the domain diversity hurts UDIT. In the "No Labeled Data" scenario, the performance of UDIT mostly comes from the additional instructions. But in the "Full Labeled Data" scenario, the domain diversity contributes most to the improvement of UDIT.

5.3 Effect of Data Amount

Since the pseudo-labeled data are constructed from the plain-text corpus, we can obtain numerous training samples for UDIT. However, as shown in Figure 4(a), the performance converges when the number of pseudo-labeled training samples per task reaches 10k in all the three scenarios we consider in Section 4.2. This is different from other methods using unlabeled data, such as self-supervised pre-training, where increasing the data amount continuously improves downstream performance (Liu et al., 2019; Kaplan et al., 2020). These results suggest that UDIT is not data-hungry and does not consume much more training resources than Vanilla-IT.

5.4 Effect of Individual Task Clusters

The pseudo-labeled data inevitably contain noises which may hurt the model performance. Therefore,



(a) Data Amount (b) Individual Task Clusters

Figure 4: The effect of the data amount and individual task clusters. The y-axis means the average results on the classification tasks.

we investigate the influence of these noises in each task cluster. We choose one task cluster at a time, replace the labeled samples with pseudo-labeled samples, and perform IT on the mixed data. In Figure 4(b), we can see that using pseudo-labeled data in MCQA affects the zero-shot performance the most. But in general, replacing one task cluster does not bring much influence. This means that pseudo-labeled data in each cluster are of high quality and UDIT is robust to the noises in individual task clusters. However, comparing Table 1 and Table 3, we find that UDIT still has a performance drop that cannot be ignored when replacing all the labeled data. It is probably caused by the noise accumulation in the pseudo-labeled data of multiple tasks. We leave how to further reduce the noises in the pseudo-labeled data as future work.

6 Conclusion and Future Work

In this work, we investigate performing IT with unlabeled data for zero-shot cross-task generalization. We first empirically find that the IT performance is largely restricted by the number of distinct tasks, instructions, and training samples in data-scarce tasks. Then, we propose UDIT to take better advantage of the instructions by constructing pseudo-labeled data from the unlabeled plain texts. Through UDIT, it is possible to perform IT with unlabeled data when there are few or no human-annotated samples, which offers a better way to incorporate unlabeled data compared with other approaches. Through comprehensive analysis, we find that the domain diversity and the matching between the pseudo-labeled data and corresponding instructions are essential for UDIT. In contrast, noises in individual task clusters and colossal data amount are less influential. There are three directions for future work: (1) Designing automatic and generalizable methods to construct pseudo-labeled data for instruction tuning. (2) Mining novel instructions from the unlabeled corpus to enlarge the amount of instructions during training. (3) Further denoising the pseudo-labeled data built from unlabeled plain texts.

Limitations

The limitation of our work is that the process of constructing pseudo-labeled data from unlabeled plain texts still needs manual design. Although the strategies we use are easy to implement and our pseudo-labeled data have covered a big part of classic NLP tasks, there may exist some "hard tasks" where finding suitable methods to construct high-quality pseudo-labeled data is not easy. However, this is not a severe problem in practice because UDIT boosts instruction learning for zero-shot cross-task generalization. This means we can still improve the performance on the "hard tasks" with UDIT based on the pseudo-labeled data from the "easy tasks". We believe that more generalizable and elaborate data construction methods would further improve performance. We leave this as future work, and the findings in this work can guide the design of these methods.

Acknowledgements

This paper was supported by the National Key Research and Development Program of China (No. 2021ZD0113304), the National Science Foundation for Distinguished Young Scholars (with No. 62125604), and the NSFC projects (Key project with No. 61936010 and regular project with No. 61876096). This work was also supported by the Guoqiang Institute of Tsinghua University, with Grant No. 2019GQG1 and 2020GQG0005, and sponsored by Tsinghua-Toyota Joint Research Fund.

References

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. 2021. Explanations for CommonsenseQA: New Dataset and Models. In *Proceedings of ACL*.

Stéphane Aroca-Ouellette and Frank Rudzicz. 2020. On losses for modern language models. In *Proceedings of EMNLP*.

Stephen H Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V Nayak, et al. 2022.

PromptSource: An integrated development environment and repository for natural language prompts. In *Proceedings of ACL (demo)*.

Philip Bachman, Ouais Alsharif, and Doina Precup. 2014. Learning with pseudo-ensembles. In *Proceedings of NeurIPS*.

Trapit Bansal, Rishikesh Jha, Tsendsuren Munkhdalai, and Andrew McCallum. 2020. Self-supervised meta-learning for few-shot natural language classification tasks. In *Proceedings of EMNLP*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, et al. 2020. Language models are few-shot learners. In *Proceedings of NeurIPS*.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. 2009. Semi-supervised learning. *IEEE Transactions on Neural Networks*.

Mingda Chen, Jingfei Du, Ramakanth Pasunuru, Todor Mihaylov, Srini Iyer, Veselin Stoyanov, and Zornitsa Kozareva. 2022. Improving in-context few-shot learning via self-supervised training. In *Proceedings of NAACL*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Proceedings of Machine Learning Challenges: Evaluating Predictive Uncertainty*.

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *Proceedings of EMNLP*.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The CommitmentBank: Investigating projection in naturally occurring discourse. In *Proceedings of Sinn und Bedeutung 23*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

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 NAACL-HLT*.

Avia Efrat and Omer Levy. 2020. The Turking Test: Can language models understand instructions? *arXiv preprint arXiv:2010.11982*.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-News: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of ACL*.

Alexander R Fabbri, Patrick Ng, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. Template-based question generation from retrieved sentences for improved unsupervised question answering. In *Proceedings of ACL*.

Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. CERT: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766*.

Matthew Finlayson, Kyle Richardson, Ashish Sabharwal, and Peter Clark. 2022. What makes instruction learning hard? an investigation and a new challenge in a synthetic environment. *arXiv preprint arXiv:2204.09148*.

Wikimedia Foundation. 2022. Wikimedia downloads.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization (EMNLP2019)*.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. 2019. Openwebtext corpus.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*.

Giovanni Grano, Andrea Di Sorbo, Francesco Mercaldo, Corrado A Visaggio, Gerardo Canfora, and Sebastiano Panichella. 2017. Software applications user reviews.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2022. PPT: Pre-trained prompt tuning for few-shor learning. In *Proceedings of ACL*.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, et al. 2021. Pre-trained models: Past, present and future. *AI Open*.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of EMNLP*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *Proceedings of AAAI*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of ICLR*.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of EMNLP*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, and Christian Bizer. 2014. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Proceedings of KR*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of ACL*.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, et al. 2021. Datasets: A community library for natural language processing. In *Proceedings of EMNLP*.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of COLING*.

Bill Yuchen Lin, Kangmin Tan, Chris Miller, Beiwen Tian, and Xiang Ren. 2022. Unsupervised cross-task generalization via retrieval augmentation. In *Proceedings of NeurIPS*.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of EMNLP*.

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning over paragraph effects in situations. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering (EMNLP2019)*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Xiaochen Liu, Yu Bai, Jiawei Li, Yinan Hu, and Yang Gao. 2022. PSP: Pre-trained soft prompts for few-shot abstractive summarization. In *Proceedings of COLING*.

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 preprint arXiv:1907.11692*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of ACL*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of ACL*.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of RecSys*.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed precision training. In *Proceedings of ICLR*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of ACL*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of NAACL-HLT*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of EMNLP*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of ACL*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *Proceedings of NeurIPS*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of NAACL-HLT*.

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

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. *JMLR*.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. ZeRO: Memory optimizations toward training trillion parameter models. In *Proceedings of SC20*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of EMNLP*.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. DeepSpeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of KDD*.

Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. 2015. Semi-supervised learning with ladder networks. In *Proceedings of NeurIPS*.

Yuanhang Ren, Ye Du, and Di Wang. 2018. Tackling adversarial examples in QA via answer sentence selection. In *Proceedings of the Workshop on Machine Reading for Question Answering (ACL2018)*.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. 2020. Getting closer to ai complete question answering: A set of prerequisite real tasks. In *Proceedings of AAAI*.

Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. 2018. DuoRC: Towards complex language understanding with paraphrased reading comprehension. In *Proceedings of ACL*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *Proceedings of ICLR*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of EMNLP*.

Henry Scudder. 1965. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*.

Nagel Sebastian. 2016. CC-news.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of ACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of ACL*.

Lakshay Sharma, Laura Graesser, Nikita Nangia, and Utku Evci. 2019. Natural language understanding with the quora question pairs dataset. *arXiv preprint arXiv:1907.01041*.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge data set and models for dialogue-based reading comprehension. *TACL*.

Oyvind Tafjord, Peter Clark, Matt Gardner, Wen-tau Yih, and Ashish Sabharwal. 2019a. QUAREL: A dataset and models for answering questions about qualitative relationships. In *Proceedings AAAI*.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019b. QuaRTz: An open-domain dataset of qualitative relationship questions. In *Proceedings of EMNLP*.

Niket Tandon, Bhavana Dalvi, Keisuke Sakaguchi, Peter Clark, and Antoine Bosselut. 2019. WIQA: A dataset for "what if..." reasoning over procedural text. In *Proceedings of EMNLP*.

Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. 2022. What language model architecture and pretraining objective work best for zero-shot generalization? In *Proceedings of ICML*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *Proceedings of ICLR*.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text (ACL2017)*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *TACL*.

Orion Weller, Nicholas Lourie, Matt Gardner, and Matthew E. Peters. 2020. Learning from task descriptions. In *Proceedings of EMNLP*.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. In *Proceedings of NeurIPS*.

Hanwei Xu, Yujun Chen, Yulun Du, Nan Shao, Yanggang Wang, Haiyu Li, and Zhilin Yang. 2022a. Zeroprompt: Scaling prompt-based pretraining to 1,000 tasks improves zero-shot generalization. *arXiv preprint arXiv:2201.06910*.

Jiahao Xu, Yubin Ruan, Wei Bi, Guoping Huang, Shuming Shi, Lihui Chen, and Lemao Liu. 2022b. On synthetic data for back translation. In *Proceedings of NAACL-HLT*.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of EMNLP*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of EMNLP*.

Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. 2021. CrossFit: A few-shot learning challenge for cross-task generalization in nlp. In *Proceedings of EMNLP*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of ACL*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of NeurIPS*.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of NAACL-HLT*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of ICML*.

Chunting Zhou, Junxian He, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Prompt consistency for zero-shot task generalization. *arXiv preprint arXiv:2205.00049*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of ICCV*.

Appendices

A Data Information

A.1 Training Tasks

Following Sanh et al. (2022), we adopt 8 task clusters containing 36 datasets. The datasets and the number of instructions in each cluster are shown in Table 7. All instructions are taken from the Public Pool of Prompts (P3) (Bach et al., 2022)[4].

A.2 Evaluation Tasks

We evaluate our model on 4 text classification task clusters and 2 language generation task clusters. The text classification task clusters and datasets include: (1) *Coreference Resolution* (Coref.): WSC and Winogrande (Wino.) (Levesque et al., 2012); (2) *Natural Language Inference* (NLI): CB (De Marneffe et al., 2019), RTE (Dagan et al., 2006), and ANLI-R1 (Nie et al., 2020); (3) *Sentence Completion* (Sentence Comp.): COPA (Gordon et al., 2012), HellaSwag (H-Swag) (Zellers et al., 2019), and Story Cloze (Mostafazadeh et al., 2016); (4) *Word Sense Diasambiguation* (WSD): WIC (Pilehvar and Camacho-Collados, 2019). The language generation task clusters and datasets include: (5) *Question Generation* (QG): SQuAD (Rajpurkar et al., 2016); (6) *Open-Ended Natural Language Generation* (ONLG): Roc Story (Mostafazadeh et al., 2016). All instructions are obtained from the Public Pool of Prompts (P3) (Bach et al., 2022).

A.3 Unlabeled Data

Our unlabeled plain texts consist of the multi-domain corpus, including BookCorpus (Zhu et al., 2015) (5.5G), Wikipedia (Foundation, 2022) (20G), CC-News (Sebastian, 2016) (1.7G), OpenWebText (Gokaslan et al., 2019) (10G), IMDB Review (Maas et al., 2011) (65M). We access these data from the HuggingFace Datasets (Lhoest et al., 2021)[5]. For OpenWebText, we randomly sample 10GB sample from the original 38GB samples to balance the data sources.

B More Training Details

We run IT on a 700M T5 model[6]. The max input sequence lengths of the encoder and the decoder are 512 and 128, respectively. We first run Vanilla-IT to select the hyper-parameters that yield the best performance on the validation splits of the training datasets. Then, we fix the hyper-parameters in all our experiments. We search for the learning rate in [3e-5, 5e-5, 1e-4], the batch size in [512, 1024, 2048], and the max training steps in [10K, 30K]. We finally set the learning rate to 5e-5, batch size to 1024, and the max training steps to 10K for both Vanilla-IT and UDIT. We use the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-8$, and weight_decay $= 0.01$. We follow Wei et al. (2022) to balance each dataset by treating each task at most 3K samples per instruction for sampling.

To improve the training efficiency, we adopt the mixed-precision training (Micikevicius et al., 2018)

Cluster	#Instr.	Datasets
MCQA	80	COS-E (Aggarwal et al., 2021), DREAM (Sun et al., 2019), QuAIL (Rogers et al., 2020), QuaRTz (Tafjord et al., 2019b), Social-IQA (Sap et al., 2019), WiQA (Tandon et al., 2019), CosmosQA (Huang et al., 2019), QASC (Khot et al., 2020), QUAREL (Tafjord et al., 2019a), SciQ (Welbl et al., 2017), Wiki-Hop (Welbl et al., 2018)
EXQA	46	Adversarial-QA (Ren et al., 2018), Quoref (Dasigi et al., 2019), ROPES (Lin et al., 2019), DuoRC (Saha et al., 2018)
CBQA	26	Hotpot-QA (Yang et al., 2018), Wiki-QA (Yang et al., 2015)
SENT	43	Amazon (McAuley and Leskovec, 2013), App-Reviews (Grano et al., 2017), IMDB (Maas et al., 2011), Rotten-Tomatoes (Pang and Lee, 2005), Yelp (Zhang et al., 2015)
TC	29	AG-News (Zhang et al., 2015), DBPedia (Lehmann et al., 2014), TREC (Li and Roth, 2002)
S2T	14	Common-Gen (Lin et al., 2020), Wiki-Bio (Lebret et al., 2016)
SUM	41	CNN-Daily-Mail (See et al., 2017), Gigaword (Graff et al., 2003), MultiNews (Fabbri et al., 2019), SAMSum (Gliwa et al., 2019), XSum (Narayan et al., 2018)
PARA	25	MRPC (Dolan and Brockett, 2005), PAWS (Zhang et al., 2019), QQP (Sharma et al., 2019)

Table 7: Task clusters and datasets use for training. "#Instr." means the number of instructions in each cluster.

[4] https://github.com/bigscience-workshop/promptsource

[5] https://huggingface.co/datasets

[6] https://huggingface.co/liangtaiwan/t5-v1_1-lm100k-large

Cluster	Dataset	#Valid Set	#Instr.
Coref.	WSC	104	10
	Winogrande	1,267	5
NLI	CB	57	15
	RTE	277	10
	ANLI-R1	1,000	15
Story Comp.	COPA	100	12
	HellaSwag	10,042	5
	Story Cloze	1,871	5
WSD	WIC	637	10
QG	SQuAD	10,570	2
ONLG	Roc-Story	1,871	1

Table 8: Task clusters and datasets for zero-shot cross-task evaluation. "#Valid Set" stands for the number of samples in the validation split of each dataset, which we use as the test sets. "#Instr". means the number of instructions in each dataset.

and ZeRO (stage-1) (Rajbhandari et al., 2020) implemented in DeepSpeed (Rasley et al., 2020)[7]. Note that the T0-3B model is evaluated in FP32 precision. Our experiments are all conducted on the NVIDIA 32G V100 GPU. We use two GPUs for each run of IT, which completes in about 12 hours, depending on the total training data amount. The inference of a single model occupies one GPU and takes about 10 minutes.

C More results

C.1 Other Choices of the "Few Tasks" Setting

In Table 9, we present the results when we use different task clusters as the data-sufficient cluster, as a complementary to Table 2. From the results, we can see that UDIT improves Vanilla IT in most cases. One exception is the language generation tasks when we train the model on EXQA. We think the reason is that some tasks in EXQA are also formulated to question generation tasks, which are too similar to the evaluation task and cover the effect of UDIT. Note that the zero-shot performance of Vanilla-IT on language generation tasks is really poor when the model is trained only on SENT, TC, or PARA, which mainly consist of text classification tasks. We observe that all output texts are biased to the labels of corresponding training datasets, which means the model overfits the text classification tasks during IT and fails to learn to follow instructions in unseen tasks.

[7] https://github.com/microsoft/DeepSpeed

Full-Data Cluster	Classification		Generation	
	Vanilla-IT	UDIT	Vanilla-IT	UDIT
MCQA	52.9	**56.5**	19.8	**21.1**
EXQA	47.9	**51.3**	**21.1**	20.8
CBQA	45.4	**50.7**	10.2	**17.5**
SENT	47.2	**50.6**	0.2	**15.9**
TC	44.3	**49.7**	1.1	**16.4**
S2T	42.6	**50.8**	14.0	**19.0**
SUM	42.7	**50.4**	16.1	**17.2**
PARA	46.4	**49.4**	0.8	**15.9**

Table 9: The results when we choose different task clusters as the data-sufficient cluster in the "Few Tasks" setting. "Full-Data Cluster" means the data-sufficient task cluster. We report the average performance on the text classification and generation tasks.

C.2 Median Results Across Different Testing Instructions

Following Sanh et al. (2022), we also report the median of the performances across different testing instructions in Table 10, Table 11, and Table 12 as a supplement to the mean of the performances in Table 1, Table 2, and Table 3, respectively. Comparing different approaches, we can draw similar conclusions as Section 4.2 that UDIT offers a significantly better way to incorporate unlabeled data into IT and improves the zero-shot cross-task generalization. We also observe that the mean and median do not differ much on most datasets, except for CB where the median is much better.

C.3 Variance Across Instructions

We draw the box plot of UDIT and some baselines under the "No Labeled Data" (Section 4.2.1) and the "Full Labeled Data" (Section 4.2.3) settings to show the variance across different instructions. From Figure 5, we can see that the results vary across instructions in all methods, which is also observed in Sanh et al. (2022). There are plenty of studies on how to reduce the variance across prompts or instructions (Zhou et al., 2022; Zhao et al., 2021; Lu et al., 2022). Most of them can be combined with our methods.

C.4 Human Evaluation on Pseudo-Labeled Data

We conduct human evaluation on the pseudo-labeled data, and the results are shown in Table 13. For each task cluster, we randomly select 50 sample-instruction pairs and recruit 3 different annotators from Amazon Mechanical Turk[8] to evalu-

[8] https://www.mturk.com/

Method	Size	Coref.		NLI			Sentence Comp.			WSD	Avg.
		WSC	Wino.	CB	RTE	ANLI	COPA	H-Swag	Story	WiC	
DirectZS	700M	50.0	51.1	33.3	47.4	33.1	42.7	26.6	53.0	50.3	43.0
DirectZS	3B	50.0	51.0	33.3	47.6	32.7	45.8	25.6	55.6	50.6	43.6
DirectZS	11B	**57.7**	50.7	33.9	51.8	32.7	55.0	27.7	48.8	50.3	45.4
ExtraLM	700M	53.1	51.5	27.1	52.6	32.6	49.5	24.2	51.8	50.6	43.6
SelfSup-IT	700M	50.8	**54.6**	41.7	52.9	**33.2**	51.0	23.7	50.7	50.2	45.4
UDIT	700M	53.3	52.6	**50.0**	53.7	**33.2**	57.8	29.8	68.0	52.7	50.0
UDIT + SelfSup-IT	700M	53.3	52.9	45.8	**56.4**	32.8	**58.8**	**29.9**	**71.5**	**52.8**	**50.5**

Table 10: The median classification accuracy of the experiments in Table 1 (No Labeled Data).

Setting	Method	Coref.		NLI			Sentence Comp.			WSD	Avg.
		WSC	Wino.	CB	RTE	ANLI	COPA	H-Swag	Story	WiC	
Few Tasks	Vannila-IT	50.8	54.4	62.5	53.7	31.2	**61.4**	28.2	50.7	50.8	49.2
	SelfSup-IT	50.8	54.9	51.4	52.9	**33.3**	58.0	27.4	53.0	50.3	48.0
	UDIT	**53.8**	55.0	**68.8**	**57.3**	32.4	58.3	**29.4**	66.8	**54.9**	**53.0**
	UDIT + SelfSup-IT	51.6	**56.6**	45.8	55.9	**33.3**	59.9	26.7	**67.9**	51.9	50.0
Few Datasets	Vannila-IT	50.0	52.2	39.6	49.8	**33.0**	57.8	26.2	54.2	51.0	45.1
	SelfSup-IT	50.8	54.3	41.7	53.7	**33.0**	56.2	27.3	60.2	50.4	47.5
	UDIT	50.8	50.9	**56.2**	**66.9**	30.7	**68.2**	**30.2**	71.3	51.5	**53.0**
	UDIT + SelfSup-IT	**51.6**	**55.6**	45.8	58.4	32.2	61.4	27.6	**77.0**	**52.6**	51.4
Few Samples	Vanilla-IT	50.8	51.8	56.2	57.4	31.4	66.7	26.5	71.0	53.4	51.7
	SelfSup-IT	50.8	53.2	47.9	62.7	31.8	61.4	26.4	77.2	54.6	51.8
	DataAug-IT	50.8	52.2	**60.6**	54.6	30.8	64.8	28.0	72.0	53.0	51.9
	UDIT	**53.1**	52.7	56.2	**65.1**	31.4	69.8	**29.6**	79.6	55.2	**54.7**
	UDIT + SelfSup-IT	52.3	**54.3**	52.1	55.7	**32.7**	70.3	28.9	**82.0**	**56.0**	53.6
	UDIT + DataAug-IT	50.8	51.3	56.2	61.0	31.8	**72.9**	29.1	80.2	52.6	54.2

Table 11: The median classification accuracy of the experiments in Table 2 (Few Labeled Data).

ate whether the pseudo-labeled sample is aligned with the instruction (scored as 1) or not (scored as 0). The final score for each task cluster is averaged over all the samples and 3 different annotators. From the results, we can see that although most of the pseudo-labeled samples make sense to humans, there inevitably exist some mislabeled samples that may be harmful to the model. We leave how to further denoise the pseudo-labeled data to future work.

D Examples of Pseudo-Labeled Data

We list a few examples of the pseudo-labeled data for the 8 task clusters in Table 14. In MCQA, EXQA, and CBQA, although the constructing process relies on some assumptions, the pseudo-labeled data reflect the task semantics well and thus match the meanings of the corresponding instructions. We notice some incoherence and typos in the pseudo-labeled data, but this does not affect the general meanings of the sentences. For TC, SENT, and SUM, we find the pseudo-labeled data to be of high quality. For S2T and PARA, we observe that the pseudo-labeled data is much easier than the labeled data. This may harm conventional supervised learning since these data can hurt the model's ability to solve hard samples. However, we argue that this issue is not that severe in instruction learning because "learning to follow instructions" only requires the correct mapping between the instructions and the task semantics, which is satisfied in the pseudo-labeled data, despite its simpleness.

Method	Size	Coref.		NLI			Sentence Comp.			WSD	Avg.
		WSC	Wino.	CB	RTE	ANLI	COPA	H-Swag	Story	WiC	
Vannila-IT	700M	52.3	53.1	62.5	68.8	**35.0**	76.3	27.5	89.8	50.7	58.0
SelfSup-IT	700M	48.4	**56.8**	60.4	66.5	34.3	79.7	25.8	90.5	50.7	57.0
UDIT	700M	**53.9**	53.4	**70.8**	**73.2**	33.3	78.6	**29.0**	88.3	53.3	**59.3**
UDIT + SelfSup-IT	700M	51.6	56.2	56.2	64.7	34.3	**80.2**	26.9	**90.7**	**56.7**	57.5
Vannila-IT	3B	62.5	52.4	50.0	72.5	35.2	78.1	28.3	88.9	52.4	57.8
Vannila-IT	11B	64.4	60.5	78.6	81.2	44.7	90.8	33.6	94.7	57.2	67.3

Table 12: The median classification accuracy of the experiments in Table 3 (Full Labeled Data).

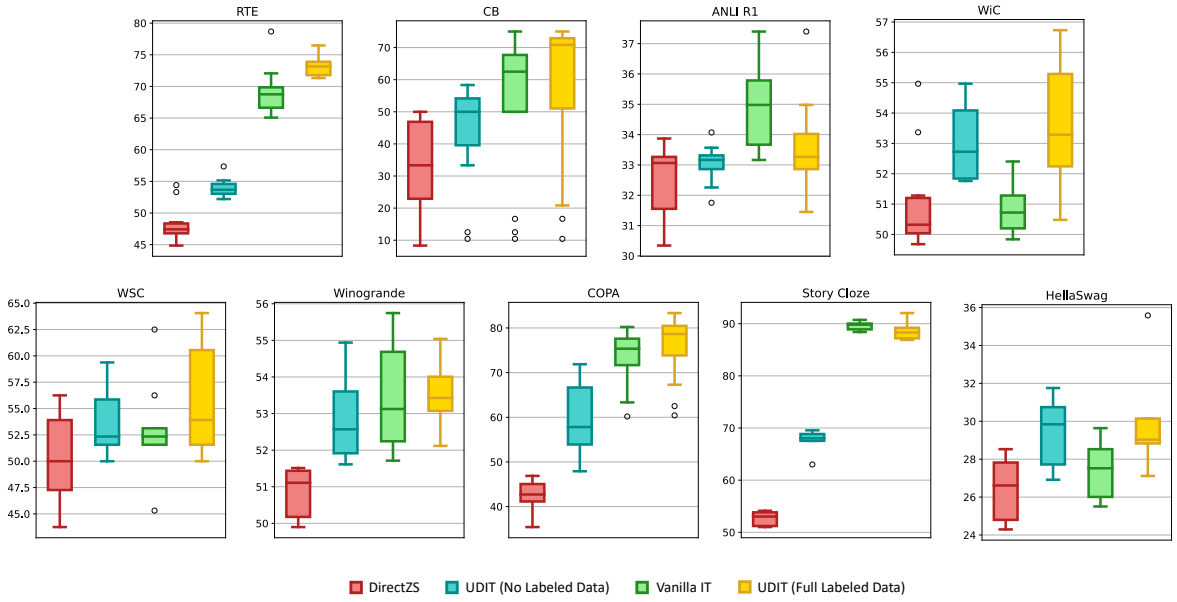


Figure 5: Evaluation results on classification tasks with variance across different instructions.

Data Cluster	MCQA	EXQA	CBQA	TC	SENT	S2T	SUM	PARA
Score	0.91	0.64	0.65	0.88	0.83	0.77	0.76	0.89

Table 13: Human evaluation results of the pseudo-labeled data.

Task	Plain Texts	Pseudo-Labeled Data
MCQA	"Just open-source it" is not realistic. I've received a couple of questions about fred following the failure of ... The big one is: "why don't you just open-source it as-is?". My answer is: it's impractical, and it wouldn't help anyone as-is.	**Passage**: "Just open-source it" is not realistic. I've received a couple of questions about fred following the failure of ... **Question**: why don't you just open-source it as-is? **Answer**: My answer is: it's impractical, and it ... **Option A**: My answer is: it's impractical, and it ... **Option B**: Both Harrison and Andrew laughed ...
EXQA	It lies west of Liniewo, east of Kościerzyna, and south-west of the regional capital Gdańsk. ... South-east of Kościerzyna, and south-west of the regional capital Gdańsk, it lies approximately south of Liniewo.	**Passage**: It lies west of Liniewo, east of Kościerzyna, and south-west of the regional capital Gdańsk. ... **Question**: Where south-east of Kościerzyna, and south-west of the regional capital Gdańsk, it lies approximately south of? **Answer**: Liniewo
CBQA	Psychroflexus planctonicus is a gram-negative bacteria which has been isolated from the Lake Xiaochaidan in the Qinghai Province In China.	**Question**: Where in China, Psychroflexus planctonicus is a gram-negative bacteria which has been isolated from the Lake Xiaochaidan in? **Answer**: the Qinghai Province
TC	URL: https://www.mydailyregister.com/sports/14501/eagles-topple-trimble-8-5 The Eastern baseball team trailed 2-0, two innings into Monday night's Tri-Valley Conference Hocking Division showdown with ...	**Passage**: The Eastern baseball team trailed 2-0, two innings into Monday night's Tri-Valley Conference Hocking Division showdown with ... **Label**: Sports
SENT	AMAZING! The staff was so friendly, welcoming and the food was superb!	**Passage**: AMAZING! The staff was so friendly, welcoming and the food was superb! **Label**: Positive
S2T	Other features will help make Henn-na the most futuristic low-cost hotel in the industry.	**Keyword**: most; industry; hotel; Henn-na **Text**: Other features will help make Henn-na the most futuristic low-cost hotel in the industry.
SUM	Title: NVIDIA Deep Learning Platform Gives Enterprise Customers Instant Access to AI ... Passage: Baidu and NVIDIA are long-time partners in advancing the state of the art in AI ...	**Passage**: Baidu and NVIDIA are long-time partners in advancing the state of the art in AI... **Summary**: NVIDIA Deep Learning Platform Gives Enterprise Customers Instant Access to AI...
PARA	You're likely vulnerable to online attacks.	**Sent1**: You're likely vulnerable to online attacks. **Sent2**: You are incredibly vulnerable to online pilots. **Label**: not

Table 14: Examples of the unlabeled plain texts and the corresponding pseudo-labeled data for each task cluster. The important parts in the unlabeled plain texts for data construction are underlined.