Persian Natural Language Inference: A Meta-learning approach

Anonymous ACL submission

Abstract

Incorporating information from other languages can improve the results of tasks in low-resource languages. A powerful method of building functional natural language processing systems for low-resource languages is to combine multilingual pre-trained representations with cross-lingual transfer learning. In general, however, shared representations are learned separately, either across tasks or across languages. This paper proposes a meta-learning approach for inferring natural language in Persian. Alternately, meta-learning uses different task information (such as QA in Persian) or other language information (such as natural language inference in English). Also, we investigate the role of task augmentation strategy for forming additional high-quality tasks. We evaluate the proposed method using four languages and an auxiliary task. Compared to the baseline approach, the proposed model consistently outperforms it, improving accuracy by roughly six percent. We also examine the effect of finding appropriate initial parameters using zero-shot evaluation and CCA similarity.

1 Introduction

In natural language processing (NLP), the goal is to improve models for the processing and production of human languages. As part of NLP, several tasks are defined, each of which covers a different level of natural language understanding. Meanwhile, natural language inference (NLI) is considered an appropriate and rigorous measure of language comprehension. This task requires to recognize the consequences of natural language sentences, which indicates how well it understands the language (MacCartney, 2009).

NLI aims to determine the inferential relationship between a premise p and a hypothesis h. The problem involves a three-class classification in which every pair (p, h) falls into one of three categories: entailment, contradiction, and neutral. If the hypothesis can be inferred from the premise, pair (p, h) will be assigned to the entailment class. For a hypothesis that contradicts the premise, pair (p, h) will be assigned to the contradiction and neutral otherwise (Amirkhani et al., 2020).

As we know, the Persian language lacks sufficient linguistic resources when it comes to natural language understanding. The lack of data can be addressed by collecting annotated data, but this process is both time-consuming and expensive (Nooralahzadeh et al., 2020). FarsTail (Amirkhani et al., 2020) is currently available for Persian, which is created using the same method as SciTail (Khot et al., 2018). It contains 10,367 samples. Also, ParsiNLU (Khashabi et al., 2021) is created for high-level tasks in Persian and for NLI , it consists of 2700 samples. As it turns out, this amount of data is too small compared with resource-rich languages (such as English, which has only 550,000 samples in the SNLI (Bowman et al., 2015) dataset).

Researchers have tried to solve the data scarcity problem by using cross-language methods. Recent work on cross-lingual learning has mainly focused on transfer between languages already covered by pre-trained representations (Wu and Dredze, 2019). Nonetheless, these techniques do not readily transfer to low-resource languages in which (1) large monolingual corpora are unavailable for pre-training, and (2) sufficient labeled data is lacking for fine-tuning downstream tasks (Xia et al., 2021).

The results of experimental studies for Persian using different embedding methods including word2vec (Mikolov et al., 2013), fastText (Bojanowski et al., 2017), ELMo (Peters et al., 2018), and BERT (Devlin et al., 2019) and various models, such as, DecompAtt (Parikh et al., 2016), ESIM (Chen et al., 2016), HBMP (Talman et al., 2019), and ULMFiT (Howard and Ruder, 2018) is reported in FarsTail (Amirkhani et al., 2020). Although this cross-lingual information sharing has

enabled success in various natural language processing tasks, it raises the question of how we can achieve more effective collaborative learning between languages or even between different tasks.

Recently, meta-learning has shown to be effective for a variety of machine learning tasks, including NLP (Koch et al., 2015; Ravi and Larochelle, 2016; Qian and Yu, 2019). This paper uses a meta-learning-based method for learning parameters in the joint space of tasks and languages. Auxiliary languages include English, Spanish, French, and German, while QA is the auxiliary task.

Alternatively, an essential prerequisite for the successful application of meta-learning is a task distribution from which a large number of tasks can be sampled to train the meta-learner. However, in NLP, datasets are usually considered as tasks (Nooralahzadeh et al., 2020; Qi and Du, 2020). There are two main problems with treating entire datasets as tasks. The first problem is overfitting, in which a meta-learner is overfitted to a small number of training tasks since there is only a small number of supervised datasets for each NLP problem. A second concern is that the heterogeneity of NLP datasets may result in learning episodes that lead to memorization overfitting, where a meta-learner ignores the support set and fails to adapt (Murty et al., 2021). To improve the quality and quantity of tasks, we use the DReCa (Murty et al., 2021) approach as our data augmentation strategy.

In this paper, we employ meta-learning algorithms and DReCa task augmentation to enhance the Persian NLI task. Our models are evaluated on the FarsTail dataset. Experimental results show that we push Persian NLI accuracy forward more than 6% and zero-shot accuracy by about 4%, setting a new state-of-the-art result for this task. In summary, the main contributions of our research are:

- We have enabled effective parameter sharing across multiple languages and tasks by providing a meta-learning approach. To the best of the authors' knowledge, this is the first study of the interaction between several languages and tasks at different levels of abstraction to solve a high-level problem in the Persian language. The evaluation results are based on the FarsTail dataset as a reference dataset in the Persian language. The datasets available in the XTREME benchmark (Hu et al., 2020) has also been used for auxiliary languages and tasks.

- We examine a metadata augmentation strategy named DReCa (Murty et al., 2021) that takes as input a set of tasks (entire datasets). We then decompose them to approximate some of the latent reasoning categories underlying these datasets, such as various syntactic constructs within a dataset, or semantic categories such as quantifiers and negation.

- We also evaluate the trained model in zero-shot mode, which means that the target language (Persian) data never appears during the training process. The test indicates the generality of the model.

The remainder of this paper is organized as follows: Section 2 briefly describes related work. Section 3 introduces our method, and in section 4, we explain the details of the experimental setup. Section 5 presents practical results. The results analysis and some justification are described in section 6. We conclude the paper and summarize future directions in section 7.

2 Related Work

In this section, we briefly outline related work in three areas. The first area is models based on cross-lingual algorithms. In the second area, we highlight methods based on meta-learning. Finally, we summarize existing data augmentation strategies.

2.1 Models based on Cross-lingual

Cross-lingual learning is a method for transferring knowledge from one natural language to another (Pikuliak et al., 2021). Pre-trained models are one of the most widely used examples of cross-lingual learning. Since these models have achieved good results, so Wu and Dredze (2019) explored the broader cross-lingual potential of mBERT (multilingual BERT) as a zero-shot language transfer model with five NLP tasks including NLI , covering a total of 39 languages. Also, Wang et al. (2019) provides a comprehensive study of the contribution of different components in mBERT to its cross-lingual ability. In addition, it examines the impact of the linguistic properties of the languages, the architecture of the model, and the learning objectives. Conneau and Lample (2019) proposed two methods for learning cross-lingual language models, one

using monolingual data and the other using parallel data and a new cross-lingual language model objective. Singh et al. (2019) introduced a cross-lingual data augmentation method that substitutes part of the input text with its translation in another language.

Huertas-Tato et al. (2021) designed a new architecture called Siamese Inter-Lingual Transformer (SILT), to align multilingual embeddings for NLI efficiently. The paper points out that transformer models are unable to generalize to other domains and have problems with multilingual and inter-linguistic scenarios. A new network has been developed to overcome these weaknesses by combining three parts: a multilingual transformer as pre-trained embedding, an alignment matrix to compute the similarity between two sentences, and a multi-head self-attention block to interpret input strings.

Despite the advances that Cross-lingual methods have made, building NLP systems in these settings is challenging for several reasons. First, the target language does not contain sufficient annotated data for effective fine-tuning. Secondly, pre-trained multilingual representations are not directly transferable due to language disparities (Xia et al., 2021). In contrast to these methods, we consider setting up training models simultaneously on multiple languages and tasks.

2.2 Meta-learning

Meta-learning addresses the problem of learning to learn. By examining many learning problems, a meta-learner learns a model (Liu et al., 2020). Specifically, the meta-learner uses a meta training set $\mathbb{MS} = \{(\mathbb{S}_i^s, \mathbb{T}_i^s)\}_{i=1}^{N^s}$, where $(\mathbb{S}_i^s, \mathbb{T}_i^s)$ are the training (support) and test (query) set of the i^{th} learning problem and N^s is the number of learning problems used for training; and a meta test set $\mathbb{MT} = \left\{\left(\mathbb{S}_i^t, \mathbb{T}_i^t\right)\right\}_{i=1}^{N^t}$, where $\left(\mathbb{S}_i^t, \mathbb{T}_i^t\right)$ are the support and query set of the i^{th} test learning problem, while N^t is the number of learning problems used for the test. Given $\mathbb{MS}$, the meta-learner learns how to map a pair $(\mathbb{S}, \mathbb{T})$ into an algorithm that leverages $\mathbb{S}$ to optimally solve $\mathbb{T}$.

Due to the lack of well-defined task distribution, meta-learning has not yet succeeded in NLP , leading to attempts that treat datasets as tasks. An ad hoc task distribution causes problems with quantity and quality. Murty et al. (2021) provide a way to break down heterogeneous tasks such as datasets into a set of appropriate subtasks. With this method, data is transferred to the feature space using a pre-trained model. They use k-means to decompose data into k clusters and create tasks by combining these clusters.

Recently, however, the combination of cross-lingual techniques in meta-learning frameworks has also been extensively studied. To train a model for low-resource languages on NLI and QA tasks, Nooralahzadeh et al. (2020) uses the MAML algorithm and auxiliary languages. van der Heijden et al. (2021) study the text documents classification problem in monolingual and multilingual modes, using different algorithms such as, Prototypical Networks (Snell et al., 2017), MAML (Finn et al., 2017), Reptile (Nichol et al., 2018), and ProtoMAML (Triantafillou et al., 2019). Also, Tarunesh et al. (2021) examine the interaction between different languages and tasks to learn an appropriate common feature space.

Additionally, transfer-learning can be helpful for low-resource languages. Xia et al. (2021) introduce a meta-learning-based framework called MetaXL for extremely low-resource languages. MetaXL learns an intelligent representational conversion from several auxiliary languages to the target language, bringing the feature space of these languages closer together for more efficient conversion. The main idea is to use a Representation Transformation network between the main model layers which are trained only with target language data.

To the best of our knowledge, this paper is the first attempt to study meta-learning for solving the NLI problem in the Persian language. Also, we are pioneers in using task-language pairs as meta-learning tasks in the Persian language.

2.3 Task Augmentation

Machine learning algorithms usually assume that the train and test data have the same distribution. In contrast, the meta-learning framework treats tasks as training examples and trains a model to adapt to all of them. Meta-learning also assumes that the training and new tasks are drawn from the same distribution of tasks $p(\tau)$. In NLP, datasets are typically treated as tasks, and meta-learners are then overfitting their adaptation mechanisms. NLP datasets are highly heterogeneous, which causes many learning episodes to have the poor transfer between their support and query sets, which dis-

suades meta-learners from adapting (Murty et al., 2021).

To deal with overfitting challenges, Yin et al. (2019) propose a meta-regularizer to mitigate memorization overfitting, but don't study learner overfitting. Rajendran et al. (2020) study task augmentation for mitigating meta-learners overfitting in the context of few-shot label adaptation. SMLMT method (Bansal et al., 2020) creates new self-supervised tasks that improve meta-overfitting, but this does not directly address the dataset-as-tasks problem. In contrast, the DReCa method (Murty et al., 2021) addresses the dataset-as-tasks problem and focuses on using clustering as a way to subdivide and fix tasks that already exist. In this paper, we use DReCa as a task augmentation strategy for our method since it mitigates meta-overfitting without any additional unlabeled data.

	NLI	QA
FA	FarsTail (10.3K)	PersianQA (9K)
EN	XNLI (392k)	—
ES	tr. XNLI (392k)	—
DE	tr. XNLI (392k)	—
FR	tr. XNLI (392k)	—

Table 1: Overview of datasets from a variety of sources. For the NLI task, we use the XNLI dataset for English, and their translated versions (tr.) for Spanish(ES), German(DE), and French(FR) provided in XTREME. For each dataset, the number of training instances is also mentioned.

3 The Proposed Methodology

In our setting, firstly, we prepare a set of task-language pairs to provide meta-learning tasks. Afterward, in each episode, we sample some tasks and feed them to the meta-learner. In the rest of this section, we describe the proposed task sampling strategy, the proposed meta-learning algorithm, and the proposed task augmentation strategy.

3.1 The Proposed Task Sampling Strategy

In meta-learning, task selection has a profound impact on model performance. For this reason, we create a queue of tasks first. We can create this queue using different scenarios such as selecting languages for a target task (Gu et al., 2018), selecting tasks for a target language (Dou et al., 2019), and picking from various auxiliary languages and auxiliary tasks. In the meta-training section, we sample some tasks from the queue. Formally, the queue's tasks are represented by $\mathcal{D}$. We need to sample tasks from $\mathcal{M}$, which is a Multinomial distribution over $P_{\mathcal{D}}(i)$s. Thus, we investigate temperature-based heuristic sampling (Aharoni et al., 2019), which defines the probability of any dataset as a function of its size as,

$$P_{\mathcal{D}}(i) = q_i^{1/\tau} / \left(\sum_{k=1}^{n} q_k^{1/\tau} \right) \quad (1)$$

where $P_{\mathcal{D}}(i)$ is the probability of sampling the ith task, q_i is the size of ith task, and τ is the temperature parameter. With $\tau = 1$, tasks are randomly sampled proportionately to their dataset sizes, and with $\tau \rightarrow \infty$, they follow a uniform distribution.

3.2 The Proposed Meta-learning Algorithms

Meta-learning is the process of building a model that can solve a new task with only a few labeled examples by training on a variety of tasks with rich annotations. The key idea is to train the model's initial parameters such that the model has maximal performance on a new task after the parameters have been updated through zero or a couple of gradient steps (Yin, 2020). MAML (model-agnostic meta-learning) (Finn et al., 2017) is one of the most significant algorithms. We describe one episode of the MAML algorithm in Appendix A.1. MAML is quite difficult to train, since there are two levels of training. Therefore, we use the following two optimization-based and metric-based meta-learning algorithms in this work.

Reptile (Nichol et al., 2018) is a first-order optimization-based algorithm that moves weights toward a manifold of the weighted averages of task-specific parameters $\theta_i^{(m)}$. It samples training tasks from $p(\mathcal{T}) : \tau_1, \cdots, \tau_i, \cdots, \tau_n$. For each training task, it generates an episode that just contains the support set data. For training task τ_i, let's assume the original parameters θ have gone through m steps of updating and become $\theta_i^{(m)}$ (i.e., $\theta_i^{(m)} = \text{AdamW}\left(L_{\tau_i}, \theta, m\right)$ (2)), then Reptile updates θ as follows (Yin, 2020):

$$\theta \leftarrow \theta + \beta \frac{1}{|\{\mathcal{T}\}|} \sum_{\tau_i \sim \mathcal{M}} \left(\theta_i^{(m)} - \theta \right) \quad (3)$$

Prototypical Networks (Snell et al., 2017) is a metric-based meta-learning algorithm. Prototypical networks learn a metric space in which classification can be performed by computing distances to prototype representations of each class. In general,

they are composed of an embedding network f_θ and a distance function $d(x_1, x_2)$. Using the following equation, the embedding network encodes the support set samples S_c and computes prototypes μ_c per class based on the mean sample encodings for that class.

$$\mu_c = \frac{1}{|S_c|} \sum_{(x_i, y_i) \in S_c} f_\theta(x_i). \quad (4)$$

A Prototypical network classifies a new sample according to the following rule.

$$p(y = c \mid x) = \frac{\exp\left(-d\left(f_\theta(x), \mu_c\right)\right)}{\sum_{c' \in C} \exp\left(-d\left(f_\theta(x), \mu_{c'}\right)\right)} \quad (5)$$

Thus, we define the *distance-based cross entropy* (DCE) loss as follows:

$$\text{Loss}(DCE) = -\log P(y = c \mid x) \quad (6)$$

To ensure that the feature space is robust to noise, we also use the Cross Entropy (CE) loss (more details can be found in Appendix A.3.1).

3.3 The Task Augmentation Strategy

First, we use dataset-as-tasks strategy that is the most common method for selecting tasks for meta-learning in NLP applications. Next, we employ DReCa to form additional high quality tasks. The goal of DReCa is to take a heterogeneous task (such as a dataset) and produce a decomposed set of tasks. Given a training task T_i^{tr}, DReCa first groups examples by their labels, and then embeds examples within each group with an embedding function EMBED(.). Concretely, for each N-way classification task T_i^{tr}, it forms groups $g_l^i = \{(\text{EMBED}(x_i), y_i) \mid y_i = l\}$. Then, it proceeds to refine each label group into K clusters via k-means clustering to break down T_i^{tr} into groups $\left\{C^j\left(g_l^i\right)\right\}_{j=1}^{K}$ for $l = 1, 2, \ldots, N$. These cluster groups can be used to produce K^N potential DReCa tasks. Each task is obtained by choosing one of K clusters for each of the N label groups, and taking their union.

4 Experimental Setup

4.1 Datasets

We use FarsTail (Amirkhani et al., 2020) for the target dataset. FarsTail is the only large-scale Persian corpus for the NLI task, with 10,367 samples. The samples are generated from 3,539 multiple-choice questions with the least amount of annotators' interventions or selected from natural sentences that already exist independently in the wild, similarly to the SciTail dataset (Khot et al., 2018).

We also use XTREME (Hu et al., 2020) as an auxiliary dataset. XTREME is a multilingual multi-task benchmark consisting of classification, structured prediction, QA , and retrieval tasks. We use this benchmark to prepare NLI data for auxiliary languages. Note that, large-scale datasets for NLI were only available in English. However, the authors of XTREME developed a custom-built translation system to get translated datasets for NLI. Furthermore, we consider the QA as an auxiliary task. Therefore, we use PersianQA (Ayoubi and Davoodeh, 2021) which is a Persian reading comprehension dataset for QA, containing over 9000 entries. Table 1 summarizes the employed dataset specifications.

4.2 Baselines

On the FarsTail dataset, Amirkhani et al. (2020) present results of various traditional and deep learning-based methods. According to the results of this paper, the highest test accuracy is obtained by using a translation-based approach, i.e., *Translate-Source* with fastText embeddings. In *Translate-Source*, the Persian-translated MultiNLI training set is combined with FarsTail training data for training an ESIM model. Furthermore, FarsTail's authors reported mBERT fine-tuning results in FarsTail webpage[1]. Therefore, we use these results as baselines.

4.3 Implementation Details

In this study, we aim to compare the effects of meta-learning algorithms on classification accuracy with those of fine-tuning and non-episodic algorithms. To make a fair comparison, we first fine-tune our pre-trained models using training data of the auxiliary task in a non-episodic approach. Afterward, we fine-tune the obtained model using the training data of the target task. In this approach, we use mBERT (Devlin et al., 2019), and XLM-R (Conneau et al., 2020), which are known as the state-of-the-art multilingual pre-trained models, and ParsBERT (Farahani et al., 2021) as a monolingual transformer-based model for the Persian language.

In the meta-learning approach, we use the XLM-R model with output layers tailored for each task

[1] https://github.com/dml-qom/FarsTail

and train it with Reptile and Prototypical algorithms. To select the hyperparameters of the Reptile algorithm, we utilize the experiments done in Tarunesh et al. (2021). Appendix A.2 provides further details. The Prototypical algorithm is used only in cross-lingual experiments, and we use Euclidean distance as its distance function. The auxiliary languages are arranged in two scenarios. In the first scenario, support and query set data are generated from auxiliary languages, while in the second scenario, the query set is drawn from both auxiliary and target languages. Detailed information is provided in Appendix A.3.2.

Furthermore, we fine-tune the obtained models on Persian training data using the following two methods. The first method is **non-episodic**, which involves fine-tuning models in batches. The second method is **episodic**, in which episodes are constructed first, and then the models are fine-tuned according to the algorithm used.

5 Results

The meta-learning model is tested on different combinations and configurations of the auxiliary tasks. The accuracy results of the Reptile algorithm are presented in Table 2. In addition to the zero-shot and fine-tuning results, we report the accuracy of another scenario. In this scenario, training data of the target language is placed in the meta-training stage along with other auxiliary tasks and cooperate in a training process. Consequently, this scenario does not involve fine-tuning phase. The results of the mentioned scenario are shown in the last column of Table 2.

Table 3 shows the accuracy scores using the Prototypical Network. In the first section of this table, we generate both support and query sets from Persian language data, without using auxiliary tasks. In the second section of this table, the results of the first multi-lingual scenario (where both the support and query sets are generated from auxiliary languages data) are reported in rows 5 to 12. In rows 13 to 16, we show the results of the second multilingual scenario (where the support set is drawn from auxiliary language data and the query set is drawn from both auxiliary and Persian language data). Lastly, we added the DReCa strategy and presented the results in rows 17 to 20.

Additionally, we conducted zero-shot evaluations of both algorithms. Zero-shot results are presented in the first accuracy column of Tables 2 and 3. The confusion matrices of the best-performing models for both Reptile and Prototypical algorithms are also depicted in Appendix A.4.

6 Discussion and Analysis

Table 2 shows that the multi-lingual models are always better than the multi-task models. Due to the fact that tasks like NLI (which require deeper semantic representations) are more likely to benefit from combining data from different languages. We found that our meta-learned models perform better than baselines and non-episodic models. The reason is that the goal of standard meta-learning is to find a model that generalizes well to a new target task. In addition, we compared two different meta-learning algorithms to evaluate their superiority in this paper. From Tables 2 and 3, we can see that Prototypical performed better than Reptile. It is because Prototypical networks use class representations instead of example representations. Therefore, it finds a suitable representation for each class during the meta-train stage.

As part of another experiment, we combined data from the target language with data from other auxiliary tasks for meta-training. Based on the results of these experiments (last column of Table 2 for Reptile and rows 13 to 16 of Table 3 for Prototypical), the model's accuracy has decreased. This is due to the fact that target language data is so small when compared with auxiliary language data. So, unbalanced training data confuses the training process and decreases the model's accuracy. In any case, the cooperation of the target language during the training process is a great idea for future work.

As indicated in the last two columns of Table 3, episodic fine-tuning is significantly superior to normal fine-tuning. It demonstrates that episodic training is effective even on single language data and creates a generality in the level of training and test data.

We examined the proximity between the feature spaces of the auxiliary languages and the target language quantitatively and qualitatively. At first, we collect representations of the auxiliary and target languages from non-episodic, Reptile, and Prototypical models. In Fig. 1, we present 2-component PCA visualization for comparison. We also evaluated the models using a distance metric commonly used in vision and NLP tasks (Huttenlocher et al., 1993; Dubuisson and Jain, 1994; Patra et al., 2019; Xia et al., 2021). Informally, the

Row	Model	Shot	Aux. Tasks	Zero-shot	non-episodic fine-tune	Add NLI-fa in m.t.
			Baselines			
1	Translate-Source*	—	—	—	78.13	—
2	mBERT*	—	—	—	83.38	—
			Non-episodic approach			
3	ParsBERT	—	—	—	74.64	—
4	mBERT	—	—	—	81.95	—
5		—	nli-en	56.53	81.38	—
6		—	nli-(en, es, de, fr)	67.88	82.34	—
7	XLM-R	—	—	—	81.97	—
8		—	nli-en	**69.49**	**86.55**	—
9		—	nli-(en, es, de, fr)	69.09	84.69	—
			Meta-learning approach			
10	XLM-R	1	nli-en	64.19	84.31	83.37
11		4		70.96	87.17	86.00
12		8		70.70	87.11	86.65
13		16		71.03	87.43	86.52
14		1	nli-(en, es, de, fr)	65.17	85.21	83.91
15		4		**72.27**	87.57	85.74
16		8		71.61	**88.35**	88.22
17		16		71.22	88.02	87.76
18		1	qa-fa	34.18	81.48	81.58
19		4		34.18	81.38	84.96
20		8		33.79	82.14	83.59
21		16		34.18	82.23	84.70
22		1	nli-en, qa-fa	46.42	83.53	85.16
23		4		66.02	86.52	86.26
24		8		64.26	86.98	86.46
25		16		46.88	86.52	86.13

Table 2: Average test accuracy of the Reptile algorithm with baselines and non-episodic approach results on the Persian NLI task. The first accuracy column shows results before fine-tuning on the Persian NLI train-set (called zero-shot). In the second accuracy column, we provided results after fine-tuning on the Persian NLI train-sets. The last accuracy column reports results of using the Persian NLI train-set in the meta-training phase (m.t.). The data with * comes from FarsTail's paper and webpage.

Row	Model	Shot	Support	Query	Zero-shot	non-episodic fine-tune	episodic fine-tune
			Without auxiliary tasks				
1	XLM-R	1	nli-fa	nli-fa	—	70.38	79.30
2		4			—	81.97	85.22
3		8			—	83.98	84.64
4		16			—	85.29	85.74
			With auxiliary tasks				
5	XLM-R	1	nli-en	nli-en	68.10	84.83	86.07
6		4			70.57	86.72	87.50
7		8			70.77	86.72	87.37
8		16			**73.18**	87.76	88.54
9		1	nli-(en, es, de, fr)	nli-(en, es, de, fr)	69.15	85.01	85.97
10		4			70.25	86.78	87.63
11		8			71.09	**88.48**	**89.39**
12		16			72.20	88.15	88.28
13		1	nli-(en, es, de, fr)	nli-(en, es, de, fr)	—	84.15	85.12
14		4			—	86.33	86.78
15		8			—	86.33	86.46
16		16			—	86.78	87.24
17	XLM-R+ DReCa	8	nli-en	nli-en	70.44	87.96	88.87
18		16			71.94	87.24	88.74
19		8	nli-(en, es, de, fr)	nli-(en, es, de, fr)	71.16	87.74	88.48
20		16			71.61	87.30	88.22

Table 3: Average test accuracy on the Persian NLI task using Prototypical algorithm with and without auxiliary tasks. The last accuracy column reports results after episodic fine-tuning on the Persian NLI train-set.

Hausdorff distance measures the distance between data representations of auxiliary languages and the target language. Given a set of representations of the auxiliary language $\mathcal{S} = \{s_1, s_2, \ldots, s_m\}$ and a set of representations of the target language $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$ we compute the Hausdorff distance as follows:

$$\max\left\{\max_{s\in\mathcal{S}}\min_{t\in\mathcal{T}} d(s,t), \max_{t\in\mathcal{T}}\min_{s\in\mathcal{S}} d(s,t)\right\} \quad (7)$$

where cosine distance is used as the inner distance, i.e.,

$$d(s,t) \triangleq 1 - \cos(s,t) \quad (8)$$

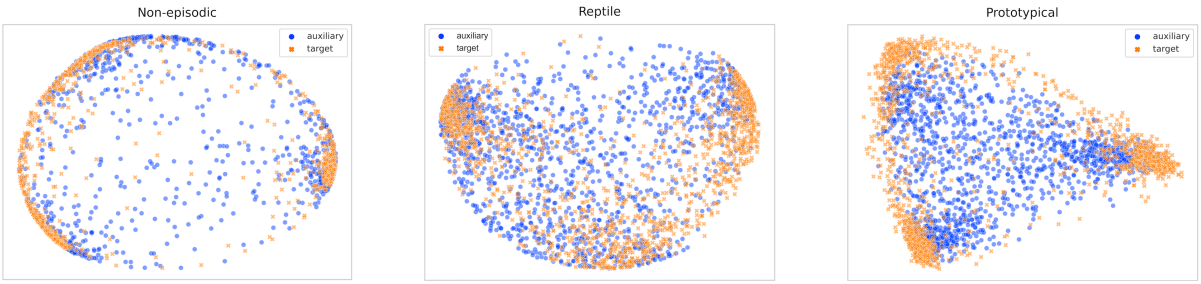

Figure 1: PCA visualization of non-episodic, Reptile, and Prototypical models to examine the closeness of the auxiliary and target languages feature spaces.

Compared to the non-episodic method, we observe a drastic drop of Hausdorff distance from 0.18 to 0.05 for Prototypical and also, we see a minor decline of Hausdorff distance from 0.18 to 0.13 for Reptile. Both qualitative visualization and quantitative metrics confirm that meta-learning approaches bring the distributions of auxiliary and target language data closer together, thus increasing the accuracy on the target language.

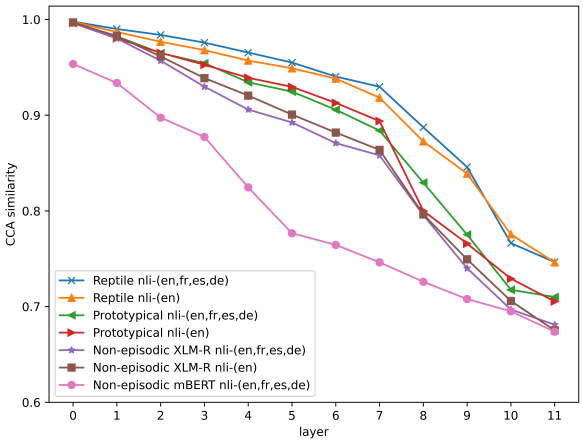

Figure 2: CCA similarity for each transformer layer. We calculate the similarity before and after fine-tuning on the FarsTail training data.

The advantage of meta-learning methods is that they obtain the appropriate initial parameters for the target language, as mentioned. The zero-shot test is used as a criterion to evaluate this point, and it shows that meta-learning-based models are more accurate than other methods. The generality of the initial parameters can also be assessed via canonical correlation analysis (CCA) (Raghu et al., 2017; Morcos et al., 2018). Using this criterion, we compare the output of each layer before and after fine-tuning, and the results are presented in Fig. 2. The meta-learning models have a higher CCA similarity, which indicates the model obtained more general parameters before fine-tuning.

In the next experiment, we apply the DReCa strategy and train the model with the Prototypical algorithm. According to Table 3, some results have improved, while others have remained the same. It illustrates that task augmentation in meta-algorithms affects the model's accuracy. However, defining the appropriate task augmentation strategy still needs research.

7 Conclusion

We present effective use of meta-learning to benefit from other tasks or languages. We advantageously leverage this approach to improve NLI in Persian as a low-source language. We found that our meta-learning model outperformed competitive baseline models. In response to the concept of treating entire datasets as tasks, we use DReCa as a general-purpose task augmenting approach. Finally, zero-shot evaluations illustrate the generality of the results obtained by meta-learning. This work will be extended to other cross-lingual NLP tasks in Persian in the future. Furthermore, we would like to use a self-supervised approach to provide a useful starting point for parameters.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Hossein Amirkhani, Mohammad Azari Jafari, Azadeh Amirak, Zohreh Pourjafari, Soroush Faridan Jahromi, and Zeinab Kouhkan. 2020. FarsTail: A persian natural language inference dataset. *arXiv preprint arXiv:2009.08820*.

Sajjad Ayoubi and Mohammad Yasin Davoodeh. 2021. Persianqa: a dataset for persian question answering.

Trapit Bansal, Rishikesh Jha, Tsendsuren Munkhdalai, and Andrew McCallum. 2020. Self-supervised meta-learning for few-shot natural language classification tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced LSTM for natural language inference. *arXiv preprint arXiv:1609.06038*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems*.

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)*, Minneapolis, Minnesota. Association for Computational Linguistics.

Zi-Yi Dou, Keyi Yu, and Antonios Anastasopoulos. 2019. Investigating meta-learning algorithms for low-resource natural language understanding tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics.

Marie-Pierre Dubuisson and Anil K. Jain. 1994. A modified hausdorff distance for object matching. In *Proceedings of 12th International Conference on Pattern Recognition*. IEEE.

Mehrdad Farahani, Mohammad Gharachorloo, Marzieh Farahani, and Mohammad Manthouri. 2021. ParsBERT: Transformer-based model for persian language understanding. *Neural Process. Lett.*

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*. PMLR.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. In *International Conference on Machine Learning*. PMLR.

Javier Huertas-Tato, Alejandro Martín, and David Camacho. 2021. SML: a new semantic embedding alignment transformer for efficient cross-lingual natural language inference. *arXiv preprint arXiv:2103.09635*.

Daniel P Huttenlocher, Gregory A. Klanderman, and William J Rucklidge. 1993. Comparing images using the hausdorff distance. *IEEE Transactions on pattern analysis and machine intelligence*.

Daniel Khashabi, Arman Cohan, Siamak Shakeri, Pedram Hosseini, Pouya Pezeshkpour, Malihe Alikhani, Moin Aminnaseri, Marzieh Bitaab, Faeze Brahman, Sarik Ghazarian, Mozhdeh Gheini, Arman Kabiri, Rabeeh Karimi Mahabagdi, Omid Memarrast, Ahmadreza Mosallanezhad, Erfan Noury, Shahab Raji, Mohammad Sadegh Rasooli, Sepideh Sadeghi, Erfan Sadeqi Azer, Niloofar Safi Samghabadi, Mahsa Shafaei, Saber Sheybani, Ali Tazarv, and Yadollah Yaghoobzadeh. 2021. ParsiNLU: A Suite of Language Understanding Challenges for Persian. *Transactions of the Association for Computational Linguistics*.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. SciTaiL: A textual entailment dataset from science question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*.

Bo Liu, Hao Kang, Haoxiang Li, Gang Hua, and Nuno Vasconcelos. 2020. Few-shot open-set recognition using meta-learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam.

Bill MacCartney. 2009. *Natural language inference*. Stanford University.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*.

Ari Morcos, Maithra Raghu, and Samy Bengio. 2018. Insights on representational similarity in neural networks with canonical correlation. In *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc.

Shikhar Murty, Tatsunori B. Hashimoto, and Christopher Manning. 2021. DReCa: A general task augmentation strategy for few-shot natural language inference. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.

Farhad Nooralahzadeh, Giannis Bekoulis, Johannes Bjerva, and Isabelle Augenstein. 2020. Zero-shot cross-lingual transfer with meta learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Barun Patra, Joel Ruben Antony Moniz, Sarthak Garg, Matthew R. Gormley, and Graham Neubig. 2019. Bilingual lexicon induction with semi-supervision in non-isometric embedding spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Matúš Pikuliak, Marián Šimko, and Mária Bieliková. 2021. Cross-lingual learning for text processing: A survey. *Expert Systems with Applications*.

Kunxun Qi and Jianfeng Du. 2020. Translation-based matching adversarial network for cross-lingual natural language inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Kun Qian and Zhou Yu. 2019. Domain adaptive dialog generation via meta learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. 2017. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc.

Janarthanan Rajendran, Alexander Irpan, and Eric Jang. 2020. Meta-learning requires meta-augmentation. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

Sachin Ravi and Hugo Larochelle. 2016. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations*.

Jasdeep Singh, Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2019. XLDA: cross-lingual data augmentation for natural language inference and question answering. *arXiv preprint arXiv:1905.11471*.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*.

Aarne Talman, Anssi Yli-Jyrä, and Jörg Tiedemann. 2019. Sentence embeddings in nli with iterative refinement encoders. *Natural Language Engineering*.

Ishan Tarunesh, Sushil Khyalia, Vishwajeet Kumar, Ganesh Ramakrishnan, and Preethi Jyothi. 2021. Meta-learning for effective multi-task and multilingual modelling. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, et al. 2019. Meta-dataset: A dataset of datasets for learning to learn from few examples. *arXiv preprint arXiv:1903.03096*.

Niels van der Heijden, Helen Yannakoudakis, Pushkar Mishra, and Ekaterina Shutova. 2021. Multilingual and cross-lingual document classification: A meta-learning approach. In *Proceedings of the 16th Conference of the European Chapter of the Association*

for Computational Linguistics: Main Volume. Association for Computational Linguistics.

Zihan Wang, Stephen Mayhew, Dan Roth, et al. 2019. Cross-lingual ability of multilingual bert: An empirical study. *arXiv preprint arXiv:1912.07840*.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics.

Mengzhou Xia, Guoqing Zheng, Subhabrata Mukherjee, Milad Shokouhi, Graham Neubig, and Ahmed Hassan Awadallah. 2021. MetaXL: Meta representation transformation for low-resource cross-lingual learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Mingzhang Yin, George Tucker, Mingyuan Zhou, Sergey Levine, and Chelsea Finn. 2019. Meta-learning without memorization. *arXiv preprint arXiv:1912.03820*.

Wenpeng Yin. 2020. Meta-learning for few-shot natural language processing: A survey. *arXiv preprint arXiv:2007.09604*.

A Appendix

A.1 MAML Description

MAML is one of the most popular meta-learning algorithms and it has proven its effectiveness in various fields (e.g., computer vision). MAML is able to find good initialization parameter values and adapt to new tasks quickly. This algorithm can be performed in one episode by following these steps:

- Make a copy of the model with its initial parameters θ.
- Use the training set D_i^{train} to train the model as
$$\hat{\theta} = \theta - \alpha \nabla_\theta \mathcal{L}_i \left(\theta, \mathcal{D}_i^{train}\right) \quad (9)$$
- Apply the model with the updated parameters $\hat{\theta}$ to the validation set D_i^{val}.
- Use the loss on the validation set to update the initial parameters θ
$$\theta = \theta - \beta \nabla \theta \sum_i \mathcal{L}_i \left(\hat{\theta}, \mathcal{D}_i^{val}\right) \quad (10)$$

A.2 Hyperparameters

Models are implemented using the PyTorch[2] framework. ParsBERT, mBERT and XLM-R implementations are taken from the HuggingFace library [3].

In our experiments, we used the AdamW optimizer (Loshchilov and Hutter, 2018) with learning rate 1e-5 to perform the inner loop of the Reptile algorithm (2), which is known as meta-step. The hyperparameters for the Reptile algorithm are listed in Table 4.

Hyperparameter	Value
epochs	2
number of iterations	20000
sequence length (for NLI)	128
sequence length (for QA)	384
dropout	0.1
optimizer	AdamW
learning rate	1e-5
update steps (m)	3
number of class per episode (way)	2
queue length	4
temperature parameter (τ)	1

Table 4: Hyperparameters for the Reptile algorithm

The hyperparameters for the Prototypical algorithm are also shown in Table 5. Some parameters are calculated based on a grid search, such as Distance Cross-Entropy (DCE) and Cross-Entropy (CE) coefficients, and others are chosen similar to the Reptile algorithm.

Hyperparameter	Value
epochs	2
number of iterations	20000
sequence length (for NLI)	128
dropout	0.1
optimizer	AdamW
learning rate	1e-5
number of class per episode (way)	3
DCE coefficient (λ_1)	1.0
CE coefficient (λ_2)	1.0

Table 5: Hyperparameters for the Prototypical algorithm

The number of iterations parameter varies according to the value of the shot, and is chosen to ensure that all instances in the dataset appear at least once in each epoch.

[2] https://pytorch.org/
[3] https://huggingface.co/

A.3 Prototypical Networks

A.3.1 Loss Function

As we mentioned in section 2.2, the primary loss function of the Prototypical algorithm is DCE. Since a prototype consists of distribution information from instances associated with it, the choice of these instances may introduce noise in the learned representation if the neural network is trained only by using the DCE loss. We use CE loss in addition to the DCE loss to make the feature space robust to noise. As a whole, we train the model with a combination of DCE loss and CE loss given by the following equation.

$$\text{Loss}(\text{overall}) = \lambda_1 \, \text{Loss}(DCE) + \lambda_2 \, \text{Loss}(CE) \tag{11}$$

A.3.2 Scenarios

We considered two scenarios for making the episodes. In the first scenario, the model is trained only on auxiliary languages, then fine-tuned using the target language. Therefore, Only auxiliary languages are used to generate support and query sets. An episode of the first scenario is shown in Table 6.

In the second scenario, in addition to auxiliary languages, we also used the target language for training. So, the support set is constructed from auxiliary language data and the query set is generated from both auxiliary and Persian language data. Table 7 shows an episode of the second scenario.

A.4 Confusion Matrices

The confusion matrices for the top-performing models (8-shot with four auxiliary languages) is depicted in Fig. 3 showing the success of this method in improving the accuracy in all classes specially the neutral class.

Example	Category
Support set (or Query set)	
In the midst of this amazing amalgam of cultures is a passion for continuity ⇒ *A passion for continuity is not the most important of these cultures*	neutral
The river plays a central role in all visits to Paris. ⇒ *The river is central to all vacations to Paris*	entailment
For the moment, he sought refuge in retreat, and left the room precipitately. ⇒ *He stayed put and sat on the floor.*	contradiction

Table 6: Example for a 3-way 1-shot episode in the first scenario. In this example we select support set and query set samples from English dataset. As support and query sets are generated similarly, only one set is shown in this table.

Example	Category
Support set	
Recuerda que una vez mencionó que su padre era médico? ⇒ *Ella mencionó que su padre era médico hace mucho tiempo*	neutral
Dies ist etwas anderes als eine Cantina-Leuchte ⇒ *Dies ist sicherlich keine Cantina-Leuchte*	entailment
Ensuite, il enfonce un tube respiratoire dans la gorge du patient mort. ⇒ *Le patient vit toujours.*	contradiction
English Translation	
You remember her once mentioning that her father was a doctor? ⇒ She mentioned her father being a doctor a long time ago.	neutral
This is something other than a cantina fixture. ⇒ This is certainly not a cantina fixture.	entailment
Next he shoves a breathing tube down the dead patient 's throat . ⇒ The patient is still alive.	contradiction
Query set	
Une pièce qualifie Frank Lloyd Wright de terrible ingénieur. ⇒ *Piece a également déclaré que Wright était un bien meilleur concepteur.*	neutral
Sus rápidos oídos captaron el sonido del tren que se acercaba. ⇒*Escuchó que el tren se acercaba rápidamente.*	entailment
از قرن دوازدهم به بعد ارقام عربی برای نخستین بار در ایتالیا کاربرد یافت. ⇒فرانسه اولین کشوری بود که از ارقام عربی استفاده کرد.	contradiction
English Translation	
A piece calls Frank Lloyd Wright an awful engineer. ⇒Piece also stated Wright was a much better designer.	neutral
Her quick ears caught the sound of the approaching train. ⇒She heard the train approaching fast.	entailment
From the twelfth century onwards, Arabic numerals were first used in Italy. ⇒France was the first country to use Arabic numerals.	contradiction

Table 7: Example for a 3-way 1-shot episode in the second scenario. In this example, the support set samples are selected from French, Spanish, and German datasets, respectively, and the query set samples are selected from French, Spanish, and Persian datasets, respectively.

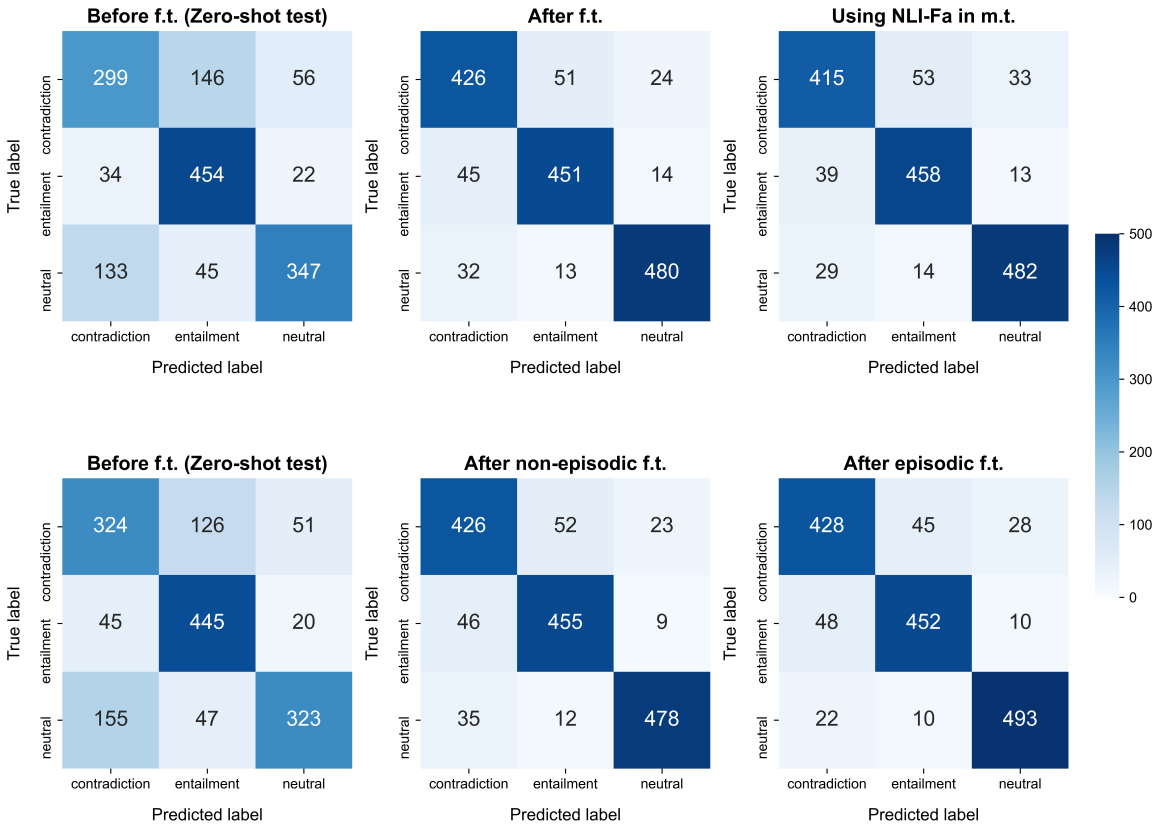

Figure 3: Confusion matrices of the best-obtained model (8-shot with four auxiliary languages) in both meta-learning algorithms on the FarsTail test set. (Top): Reptile algorithm results. (Bottom): Prototypical algorithm results.