Meta-Tsallis-Entropy Minimization: A New Self-Training Approach for Domain Adaptation on Text Classification

Menglong Lu[1] , **Zhen Huang**[1] , **Zhiliang Tian**[1] , **Yunxiang Zhao**[2] , **Xuanyu Fei**[3] and **Dongsheng Li**[1]
[1]National Key Laboratory of Parallel and Distributed Computing, National University of Defense Technology, China
[2]Beijing Institute of Biotechnology, China
[3]Suiren Information,China
{lumenglong, huangzhen, tianzhiliang, dsli}@nudt.edu.cn, {zhaoyx1993, xuanyufelix}@163.com

Abstract

Text classification is a fundamental task for natural language processing, and adapting text classification models across domains has broad applications. Self-training generates pseudo-examples from the model's predictions and iteratively train on the pseudo-examples, i.e., mininizes the loss on the source domain and the Gibbs entropy on the target domain. However, Gibbs entropy is sensitive to prediction errors, and thus, self-training tends to fail when the domain shift is large. In this paper, we propose Meta-Tsallis Entropy minimization (MTEM), which applies meta-learning algorithm to optimize the instance adaptive Tsallis entropy on the target domain. To reduce the computation cost of MTEM, we propose an approximation technique to approximate the Second-order derivation involved in the meta-learning. To efficiently generate pseudo labels, we propose an annealing sampling mechanism for exploring the model's prediction probability. Theoretically, we prove the convergence of the meta-learning algorithm in MTEM and analyze the effectiveness of MTEM in achieving domain adaptation. Experimentally, MTEM improves the adaptation performance of BERT with an average of 4 percent on the benchmark dataset.

1 Introduction

Text classification plays a crucial role in language understanding and anomaly detection for social media text. With the recent advance of deep learningf [Kipf and Welling, 2017; Devlin *et al.*, 2019], text classification has experienced remarkable progress. Despite the success, existing text classification approaches are vulnerable to domain shift. When transferred to a new domain, a well-performed model undergoes severe performance deterioration. To address such deterioration, domain adaptation, which aims to adapt a model trained on one domain to a new domain, has attracted much attention [Du *et al.*, 2020; Lu *et al.*, 2022].

A direct way to achieve domain adaptation is to build a training set that approximates the distribution of the target domain. For this purpose, self-training [Zou *et al.*, 2019; Liu *et al.*, 2021] uses the unlabeled data from the target domain to bootstrap the model. In specific, self-training first uses the model's prediction to generate pseudo-labels and then uses the pseudo-labeled data to re-train the model. In this process, self-training forces the model to increase its confidence in the confident class, which is a Gibbs entropy minimization process in essence [Lee and others, 2013].

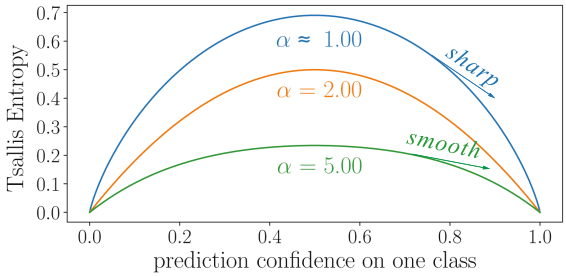


Figure 1: Tsallis entropy curve with respect to different entropy index (i.e., α below the curve).

.

However, Gibbs entropy minimization is sensitive to prediction errors [Mukherjee and Awadallah, 2020]. To handle the intractable label noise (i.e., prediction errors), data selection strategies are designed to select reliable pseudo labels [McClosky *et al.*, 2006; Reichart and Rappoport, 2007; Rotman and Reichart, 2019]. Among them, many qualified achievements [RoyChowdhury *et al.*, 2019; Shin *et al.*, 2020] are grounded on prior knowledge about the tasks (e.g., the temporal consistency on video [RoyChowdhury *et al.*, 2019]), and thus hard to be applied in text classification tasks. Since the Gibbs entropy minimization process in self-training is to minimize the model's uncertainty on the new domain, [Liu *et al.*, 2021] recently proposes to replace the Gibbs entropy with the Tsallis entropy, which is another effective metric for measuring uncertainty.

Tsallis entropy is a generalization of Gibbs entropy, referring to a set of entropy types controlled by the entropy index. Fig. 1 shows the change of Tsallis entropy with different entropy indexes for binary problems. When the entropy index is small (the resultant entropy curve is sharp),

the entropy minimization process tends to increase one dimension to 1.0 sharply, thus only being suitable for the scenario where pseudo labels are reliable. Otherwise, Tsallis entropy with a larger entropy index (smoother curve) is more suitable for scenarios with large label noise, e.g., domain adaptation scenarios with a large domain shift. Researchers [Liu *et al.*, 2021] tried to use the Tsallis entropy to improve self-training, but the proposed objective only involves a unified entropy index for all unlabeled data in the target domain. As illustrated in [Kumar *et al.*, 2010; Kumar *et al.*, 2020], different instances in the target domain have different degrees of shifts from the source domain. Thus, a unified entropy index cannot fully exploit the different pseudo instances in the target domain.

In this paper, we propose Meta-Tsallis-Entropy Minimization (MTEM) that uses an instance adaptive Tsallis entropy minimization process to minimize the model's prediction uncertainty on the target domain. Since the best entropy indexes changes along with the training, manually selecting an appropriate entropy index for each unlabeled data is intractable. Thus, we employ meta-learning to adaptively learn a suitable entropy index for each unlabeled data. The meta-learning process iterates over the *inner loop* on the target domain and *outer loop* on the source domain. In this process, the parameters optimized on the target domain also achieves a low loss on the source domain, which forces the model to obtain task informations on the target domain. However, the proposed MTEM still faces two challenges.

Firstly, the meta-learning algorithm in MTEM involves a Second-order derivation (i.e., the gradient of the entropy index), which requires much computation cost, especially when the model is large. Hence, it is hard to apply MTEM for prevailing big pre-trained language models. To this end, we propose to approximate the Second-order derivation via a Taylor expansion, which reduces the computation cost substantially.

Secondly, minimizing Tsallis entropy requires the guidance of pseudo labels (see § 2.2 and § 2.3). Previous self-training approaches generate pseudo labels by selecting the prediction with the largest probability (i.e., greedy selection), which tends to collapse when the model's prediction is unreliable [Zou *et al.*, 2019]. To this end, we propose to sample pseudo labels from the model's predicted distribution instead of a greedy selection. Further, we propose an annealing sampling mechanism to improve the sampling efficiency.

To summarize, our contributions are in three folds[1]:

(i) We propose Meta-Tsallis-Entropy Minimization (MTEM) for domain adaptation on text classification. MTEM involves an approximation technique to accelerate the computation, and an annealing sampling mechanism to improve the sampling efficiency.

(ii) We provide theoretical analysis for the MTEM, including its effectiveness in achieving domain adaptation and the convergence of the involved meta-learning process.

(iii) Experiments on two benchmark datasets demonstrate the effectiveness of the MTEM. Specifically, MTEM improves BERT on cross-domain sentiment classification tasks with an average of 4 percent, and improves BiGCN on cross-domain rumor detection task with an average of 21 percent.

[1]As the rest paper involves many mathematic symbols, we provide a symbol list (Tab. 7 in Appendix A) for reading convenience.

2 Preliminary

2.1 Domain Adaptation on Text Classification

Text classification is a task that aims to map a text to a specific label space. On a correct classification case, the process is expressed as $y_i = \arg\max_k f_{[k]}(x_i;\theta)$, where $x_i \in \mathcal{X}$ is an input text, $y_i \in \{0,1\}^K$ is the corresponding one-hot label with K classes, and f is a model with parameters θ, $f(x_i;\theta)$ is the prediction probability. Domain adaptation is to adapt a text classification model trained on the source domain (denoted as $\mathbb{D}_S$) to the target domain (denoted as $\mathbb{D}_T$). On the source domain, we have a set of labeled instances, i.e., $D_S = \{(x_i, y_i)\}_{i=1}^N$, which satisfies that $D_S \subseteq \mathbb{D}_S$. On the target domain, unlabeled text in the target domain is available, which we denote as $D_T^u = \{(x_m)\}_{m=1}^U$.

2.2 Tsallis Entropy

In information theory, Tsallis entropy refers to a set of entropy types, where the entropy index is used to identify a specific entropy. Formally, Tsallis entropy with α denoting the entropy index is written as Eq. (1),

$$e_\alpha(p_i) = \frac{1}{\alpha-1}(1-\sum_{j=1}^K p_{i[j]}^\alpha) \quad (1)$$

where p_i is the prediction probability. When $\alpha > 1$, e_α is a concave function [Plastino and Plastino, 1999]. When $\alpha \to 1$, e_α recovers the Gibbs entropy, as shown in Eq. (2) [2]:

$$e_{\alpha\to 1}(p_i) = \frac{\lim_{\alpha\to 1} 1-\sum_{j=1}^K p_{i[j]}^\alpha}{\lim_{\alpha\to 1}\alpha-1} = \sum_{j=1}^K -p_{i[j]} log(p_{i[j]}) \quad (2)$$

More intuitively, Fig. 1 exhibits the impact of the entropy index on the curves of the Tsallis entropy type. Specifically, a larger entropy index makes the curve more smooth, while a smaller entropy index exerts a more sharp curve.

Extending from the unsupervised Tsallis entropy, the corresponding *Tsallis loss* $\ell_\alpha(p_i, y_i)$ is expressed as Eq. (3). When $\alpha \to 1$, the corresponding supervised loss is the widely used cross-entropy loss (see Appendix B.1).

$$\ell_\alpha(p_i, y_i) = \frac{1}{\alpha-1}(1-\sum_{j=1}^K y_{i[j]} \cdot p_{i[j]}^{\alpha-1}) \quad (3)$$

2.3 Self-Training for Domain Adaptation

Self-training aims to achieve domain adaptation by optimizing the model's parameters with respect to the supervised loss on the source domain and the unsupervised loss (prediction uncertainty) on the target domain, as shown in Eq. (4).

$$\min_\theta \mathcal{L}_{ST}(\theta|D_S, D_T^u) = \mathcal{L}_S(\theta|D_S) + \lambda \cdot \mathcal{L}_T(\theta|D_T^u) \quad (4)$$

[2]The second equation is obtained by L'Hôpital's rule.

Algorithm 1 Meta-Tsallis-Entropy Minimization

Require: labeled source dataset D_S, unlabeled target dataset D_T^u, initial entropy-index on D_T^u, i.e., $\psi_1 = [\psi_{1[i]}|_{i=1}^{|D_T^u|}]_t$

1: **for** t = 1 $\rightarrow T_{\max}$ **do**
2: Sampling training batch $\mathcal{B} = \{x_j\}$ from D_T^u
3: Sampling validation batch $\mathcal{V}$ from D_S
4: Set κ_t with Eq.(15)
5: Sampling $\tilde{y}_j \sim p(\bullet|\theta, x_j, \kappa_t)$ for each instance in $\mathcal{B}$
6: $\hat{\theta}(\psi_t) = \theta_t - \eta_t \cdot \frac{\partial \mathcal{L}_T(\theta, \psi_t|\mathcal{B})}{\partial \theta}|_{\theta=\theta_t}$ ▷ Inner-Loop
7: $\psi_{t+1} = \psi_t - \beta_t \cdot \frac{\partial \mathcal{L}_S(\hat{\theta}(\psi_t)|\mathcal{V})}{\partial \psi}|_{\psi=\psi_t}$ ▷ Outer-Loop
8: $\theta_{t+1} = \theta_t - \eta_t \cdot \frac{\partial \mathcal{L}_T(\theta, \psi^{t+1}|\mathcal{B})}{\partial \theta}|_{\theta=\theta_t}$

where $\mathcal{L}_S$ is a supervised loss, $\mathcal{L}_T$ is an unsupervised loss, and λ is a coefficient to balance $\mathcal{L}_S$ and $\mathcal{L}_T$. Due to the simplicity, Gibbs entropy is widely used to measure the prediction uncertainty on the target domain [Zou *et al.*, 2019; Zou *et al.*, 2018], which is expressed as below:

$$\mathcal{L}_T(\theta|D_T^u) = \frac{1}{|D_T^u|} \sum_{x_i \in D_T^u} -f(x_i;\theta) \cdot log(f(x_i;\theta)) \tag{5}$$

However, as $\mathcal{L}_T(\theta|D_T^u)$ in Eq. (5) is a concave function, minimizing $\mathcal{L}_T(\theta|D_T^u)$ is hard to converge because the gradients on the minimal are larger than 0 [Benson, 1995]. For this purpose, self-training uses pseudo labels to guide the entropy minimization process, i.e., replacing Eq. (5) with Eq (6) where $\tilde{y}_i = \arg\max_k f_{[k]}(x_i;\theta)$ is the pseudo label.

$$\mathcal{L}_T(\theta|D_T^u) = \frac{1}{|D_T^u|} \sum_{x_k \in D_T^u} -\tilde{y}_i^T \cdot log(f(x_i;\theta)) \tag{6}$$

3 Meta-Tsallis-Entropy Minimization

MTEM inherits the basic framework of self-training, i.e., (i) simultaneously minimizing the supervised loss on the source domain and the unsupervised loss (prediction uncertainty) on the target domain; (ii) generating the pseudo labels to guide the entropy minimization process. The improvements of the MTEM are in three folds. Firstly, we propose an instance adaptive Tsallis entropy to measure the prediction uncertainty (§ 3.1). Secondly, we propose to use a meta-learning algorithm to minimize the joint loss (§ 3.2), which involves an approximation technique to reduce computation cost (§ 3.3). Thirdly, we propose to generate pseudo labels with an annealing sampling (§ 3.4). Fig. 2 exhibits the overview of the MTEM, and Algorithm 1 presents the core process.

3.1 Instance Adaptive Tsallis Entropy

The instance adaptive Tsallis Entropy, i.e., the unsupervised loss on the target domain, is as below:

$$\mathcal{L}_T(\theta, \psi|D_T^u) = \frac{1}{|D_T^u|} \sum_{x_k \in D_T^u} e_{\psi_{[k]}}(f(x_k;\theta)) \tag{7}$$

where $\psi_{[k]}$ indicates the entropy index for unlabeled data x_k, $e_{\psi_{[k]}}$ is the resultant Tsallis entropy.

Such an instance Tsallis entropy minimization is more effective in exploiting the model's prediction. In general, the prediction correctness is different on different instances. Thus, entropy index should be different on different instances too. For the instances with wrong prediction, we can increase the entropy index to make Tsallis entropy more smooth, then the model is updated more cauciously. Otherwise, for instances with correct prediction, we can set a small entropy index to update the model more aggressively.

However, as we are not aware of the label, setting the appropriate entropy indexes for each unlabeled data is intractable. Furthermore, as prediction errors can be corrected during the model training, the best entropy indexes change along with the updates of the model. To handle the above issues, we propose to use meta-learning to determine the entropy indexes automatically.

3.2 Meta-Learning

Meta-learning can help MTEM to find the appropriate entropy indexes due to the following reasons. Firstly, parameters optimized on a well-determined instance adaptive Tsallis entropy (entropy indexes are determined appropriately) should be more generalizable, which corresponds to the characteristics of meta-learning, i.e., training a model that can be fast adapted to a new task [Finn *et al.*, 2017]. Secondly, the meta-learning process updates the entropy indexes dynamically, thus maintaining the consistency between the model's parameters and the entropy indexes along with the whole training process. In specific, the meta-learning algorithm in MTEM iterates over the *Inner-Loop* on the target data and the *Outer-Loop* on the source domain.

In the *Inner-Loop*, we fix the entropy indexes to optimize the model's parameters with respect to the instance-adaptive Tsallis entropy on the target domain. In specific, we sample a batch of unlabeled data $\mathcal{B}$ from D_T^u and update the model with respect to their instance adaptive Tsallis entropy ($\mathcal{L}_T(\theta, \psi|\mathcal{B})$), as shown in Eq. (8):

$$\hat{\theta}_{t+1}(\psi_t) = \theta_t - \eta \cdot \frac{\partial \mathcal{L}_T(\theta, \psi_t|\mathcal{B})}{\partial \theta}|_{\theta=\theta_t}, \tag{8}$$

However, as introduced in § 2.2 and § 2.3, e_ψ in Eq. (7) is a concave function hard to be minimized. Following the way in self-training, we use the equation in Eq. (9) [3] to transform a concave function to a convex function,

$$e_\psi(f(x_i;\theta_t)) = \mathop{\mathbb{E}}_{\tilde{y}_i \sim f(x_i;\theta_t)} \ell_{\psi_{[i]}}(f(x_i;\theta_t), \tilde{y}_i) \tag{9}$$

where $\tilde{y}_i \in \{0,1\}^K$ is a one-hot pseudo label sampled from the model's prediction probability (i.e., $\tilde{y}_i \sim f(x_i;\theta_t)$). Then, the objective in the Inner-Loop is as Eq. (10):

$$\min_\theta \mathcal{L}_T(\theta, \psi|D_T^u) = \mathop{\mathbb{E}}_{\tilde{y}_i \sim f(x_i;\theta)} \frac{1}{|D_T^u|} \sum_{x_i \in D_T^u} \ell_{\psi_{[i]}}(f(x_i;\theta), \tilde{y}_i) \tag{10}$$

[3] Deduction is provided in Appendix B.2

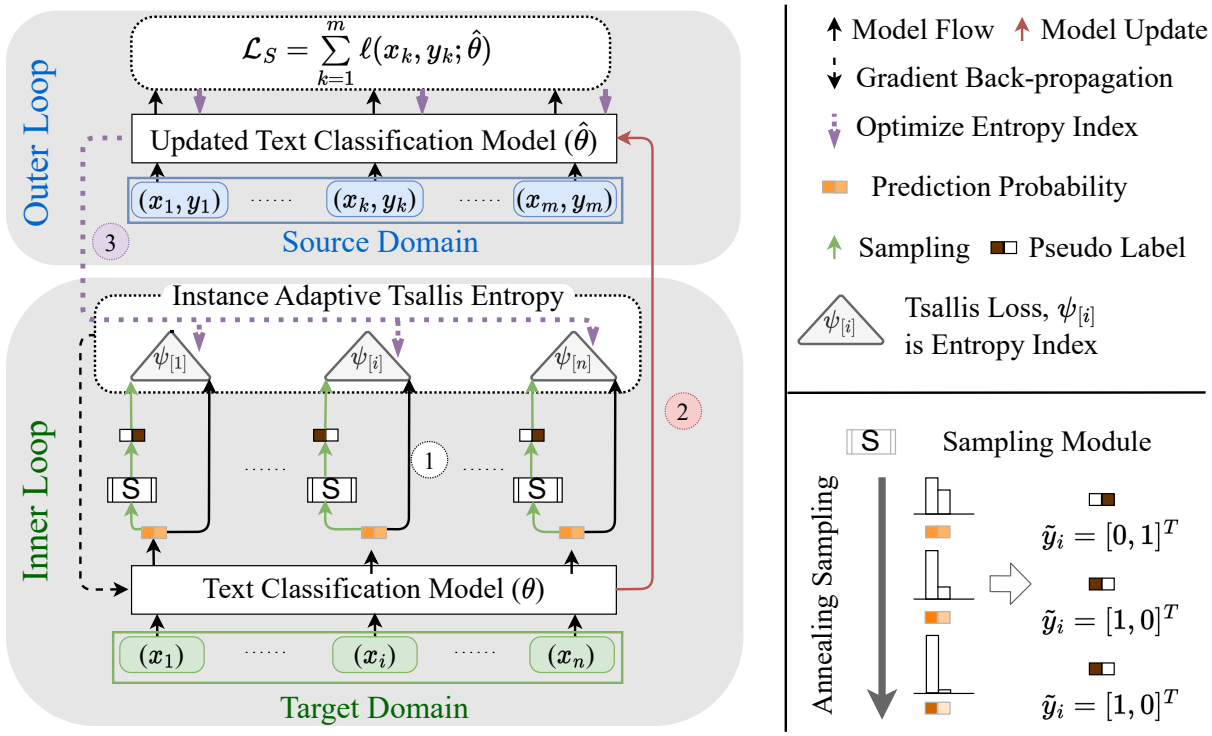


Figure 2: Meta-Tsallis-Entropy Minimization for domain adaptation on text classification: ① generate pseudo labels with annealing sampling module, and then update the model with the instance adaptive Tsallis entropy; ② validate the model on the source domain; ③ update the entropy indexes with respect to the validation performance.

In the *Outer-Loop*, we validate the model updated (i.e., $\hat{\theta}_{t+1}(\psi_t)$ in Eq. (8)) with labeled data from the source domain. Since different entropy indexes ψ leads to different $\hat{\theta}_{t+1}(\psi_t)$, we adjust ψ to find the better $\hat{\theta}_{t+1}(\psi^t)$ that can be fast adapted to the validation set. For this purpose, we optimize the entropy indexes ψ to minimize the validation loss. In each meta-validation step, we sample a valid batch of labeled data from the source domain, i.e., $\mathcal{V} \sim D_S$, and use the validation loss $\mathcal{L}_S(\hat{\theta}_{t+1}(\psi^t)|\mathcal{V})$ to evaluate the model, then update entropy indexes ψ with $\bigtriangledown_\psi \mathcal{L}_S(\hat{\theta}_{t+1}(\psi^t)|\mathcal{V})$. With the updated entropy indexes ψ_{t+1}, we return to update the model's parameters, as shown in line 8 of Algorithm 1.

3.3 Taylor Approximation Technique

The first challenge in the above meta-learning algorithm is the computation cost carried out in the $\bigtriangledown_\psi \mathcal{L}_S(\hat{\theta}_{t+1}(\psi^t)|\mathcal{V})$. Formally, the computation of $\bigtriangledown_\psi \mathcal{L}_S(\hat{\theta}_{t+1}(\psi^t)|\mathcal{V})$ is as:

$$\begin{aligned} \frac{\partial \mathcal{L}_S(\hat{\theta}_{t+1}(\psi^t)|\mathcal{V})}{\partial \psi} &= \frac{\partial \mathcal{L}_S(\hat{\theta}_{t+1}(\psi_t)|\mathcal{V})}{\partial \hat{\theta}_{t+1}(\psi_t)} \cdot \frac{\partial \hat{\theta}_{t+1}(\psi_t)}{\partial \psi} \\ &= -\eta \bigtriangledown_{\hat{\theta}} \mathcal{L}_S(\hat{\theta}_{t+1}(\psi_t)|\mathcal{V}) \\ &\quad \cdot \frac{\partial^2 \mathcal{L}_T(\theta, \psi_t|\mathcal{B})}{\partial \theta \partial \psi} \end{aligned} \tag{11}$$

where the second equation is obtained by substituting $\hat{\theta}_{t+1}(\psi^t)$ with Eq. (8). Since $\frac{\partial^2 \mathcal{L}_T(\theta, \psi^t|\mathcal{B})}{\partial \theta \partial \psi}$ in Eq. (11) is a hessian matrix (Second-order derivation), the computation in Eq. (11) is intractable. Although deep learning codebase, i.e. Pytorch and TensorFlow, provide tools for computing the Second-order derivation, the computation cost is quadratic to the model's parameters, which is thus unacceptable for recent big pre-trained language models (e.g., BERT).

Inspired by the research in [Liu *et al.*, 2018; Chen *et al.*, 2021], we propose an approximation technique for computing Eq. (11). In specific, we employ the Taylor Expansion to rewrite the term $\frac{\partial \mathcal{L}_S(\hat{\theta}(\psi))}{\partial \hat{\theta}} \frac{\partial^2 \mathcal{L}_T(\psi)}{\partial \theta \partial \psi}$ in Eq. (11) with Eq. (12).

$$\begin{aligned} &\bigtriangledown_{\hat{\theta}} \mathcal{L}_S(\hat{\theta}_{t+1}(\psi_t)|\mathcal{V}) \cdot \frac{\partial^2 \mathcal{L}_T(\theta)}{\partial \theta \partial \psi} \\ &= \frac{\bigtriangledown_\psi \mathcal{L}_T(\theta^+) - \bigtriangledown_\psi \mathcal{L}_T(\theta^-)}{2 * \epsilon} \end{aligned} \tag{12}$$

where ϵ is a small scalar, θ^+ and θ^- are defined as below:

$$\begin{aligned} \theta^+ &= \theta + \epsilon \cdot \bigtriangledown_{\hat{\theta}} \mathcal{L}_S(\hat{\theta}_{t+1}(\psi_t)|\mathcal{V}), \\ \theta^- &= \theta - \epsilon \cdot \bigtriangledown_{\hat{\theta}} \mathcal{L}_S(\hat{\theta}_{t+1}(\psi_t)|\mathcal{V}) \end{aligned} \tag{13}$$

where $\mathcal{L}_T(\theta)$ is the abbreviation of $\mathcal{L}_T(\theta, \psi_t|\mathcal{B})$. As demonstrated in [Liu *et al.*, 2018], Eq. (12) would be accurate enough for approximation when ϵ is small. However, computing $\bigtriangledown_\psi \mathcal{L}_T$ in Eq. (12) still requires much computation cost as

it involves a forward operation (i.e., $\mathcal{L}_T$) and a backward operation (i.e., $\bigtriangledown_\psi \mathcal{L}_T$). To this end, we derive the explicit form of $\bigtriangledown_\psi \mathcal{L}_T$ as Eq. (14) (details are in Appendix B.3).

$$\begin{aligned}\bigtriangledown_{\psi_{[i]}} \mathcal{L}_T(\theta) &= \frac{1}{\psi_{[i]}-1} \times [l_1(x_i,\tilde{y}_i) - l_{\psi_{[i]}}(x_i,\tilde{y}_i)] \\ &\quad -l_1(x_i,\tilde{y}_i) \times l_{\psi_{[i]}}(x_i,\tilde{y}_i) \end{aligned} \tag{14}$$

$l_1(x_i,\tilde{y}_i)$ and $l_{\psi_{[i]}}$ in Eq. (14) can be computed without gradients, thus preventing the time-consuming back-propagation process. Therefore, computing $\bigtriangledown_\psi \mathcal{L}_T$ with the above explicit form can further reduce the computation cost.

3.4 Annealing Sampling

In domain adaptation, the naive sampling mechanism in the Inner-Loop can suffer from the lowly efficient sampling problem. When the domain shift is large, the model usually performs worse in the target domain than in the source domain. As a result, the model's prediction confidence (i.e., the sampling probability) on the true class is small. Considering an extreme binary classification case, where the instance's ground truth label is $[0,1]_t$ but the model's prediction is $[0.99, 0.01]_t$, the probability of sampling the ground truth label is 0.01. In this case, most of the training cost is wasted on the pseudo instances with error labels.

To improve the sampling efficiency, we propose an annealing sampling mechanism. With a temperature parameter κ, we control the sharpness of the model's prediction probability (sampling probability) by $p(\bullet;\theta,x_i,\kappa) = softmax(\frac{score}{\kappa})$, where p is the sampling probability and $score$ is the model's original prediction score. In the earlier training phase, the model's prediction is not that reliable, so we set a high-temperature parameter κ to smooth the model's prediction distribution. With this setting, different class labels are sampled with roughly equal probability, which guarantees the possibility of sampling the correct pseudo label. Along with the convergence of the training process, the model's prediction is more and more reliable, thus the temperature scheduler will decrease the model's temperature. We design a temperature scheduler as Eq. (15):

$$\kappa_t = \kappa_{max} - (\kappa_{max} - \kappa_{min})\sigma(s - 2s \times \frac{t}{T_{\max}}) \tag{15}$$

where σ denotes the *sigmoid* function[4], κ_{max} and κ_{min} are the expected maximum temperature and minimum temperature, s is a manual set positive scalar. t is the index of the current training iteration, $T_{\max}$ is the maximum of the training iterations. Thus, $\frac{t}{T_{\max}}$ increases from 0.0 to 1.0, and the input $s - 2s \times \frac{t}{T_{\max}}$ decreases from s to $-s$. In our implementation, s is large value that satisfies $\sigma(s) \approx 1.0$ and $\sigma(-s) \approx 0.0$, which gurantees that κ_t will decrease from κ_{max} to κ_{min}.

4 Theoretical Analysis

Proofs of Lemma 1, Theorem 1, Theorem 2, and Theorem 3 are detailed in Appendix A.

[4] $\sigma(x) = \frac{1}{1+e^x}$, which approaches to 0 when $x < -5.0$ and saturates to 1.0 when $x > 5.0$

Lemma 1. *Suppose the operations in the base model is Lipschitz smooth, then $\ell_{\psi_{[i]}}(f(x_i,\theta),\tilde{y}_i)$ is Lipschitz smooth with respect to θ for $\forall\psi_{[i]} > 1$ and $\forall x_i \in D_S \bigcup D_T^u$, i.e., there exists a finite constant ρ_1 and a finite constant L_1 that satisfy:*

$$||\frac{\partial \ell_{\psi_{[i]}}(f(x_i,\theta),\tilde{y}_i)}{\partial\theta}||_2 \le \rho_1,$$
$$||\frac{\partial^2 l_{\psi_{[i]}}(f(x_i,\theta),\tilde{y}_i)}{\partial\theta^2}||_2 \le L_1$$

Also, for $\forall\psi_{[i]} > 1$ and $\forall x_i \in D_T^u$, $\ell_{\psi_{[i]}}(f(x_i,\theta),\tilde{y}_i)$ is Lipschitz smooth with respect to $\psi_{[i]}$, i.e., there exists a finite constant ρ_2 and a finite constant L_2 that satisfy:

$$||\frac{\partial \ell_{\psi_{[i]}}(f(x_i,\theta),\tilde{y}_i)}{\partial\psi_{[i]}}||_2 \le \rho_2,$$
$$||\frac{\partial^2 \ell_{\psi_{[i]}}(f(x_i,\theta),\tilde{y}_i)}{\partial\psi_{[i]}^2}||_2 \le L_2$$

Assumption 1. *The learning rate η_t (line 10 of Algorithm 1) satisfies $\eta_t = min\{1, \frac{k_1}{t}\}$ for some $k_1 > 0$, where $\frac{k_1}{t} < 1$. In addition, The learning rate β_t (line 8 of Algorithm 1) is a monotone descent sequence and $\beta_t = min\{\frac{1}{L}, \frac{k_2}{\sqrt[3]{t^2}}\}$ for some $k_2 > 0$, where $L = \max\{L_1, L_2\}$ and $\frac{\sqrt[3]{t^2}}{k_2} \ge L$.*

Based on the Assumption 1 and Lemma 1, we deduce Theorem 1 and Theorem 2. Theorem 1 demonstrates that, by adjusting ψ, the model trained on the target domain can be generalized to the source domain immediately. In other words, by adjusting ψ, the learning process on the target domain (i.e., Eq. (10)) has learned the domain agnostic features. At the same time, Theorem 2 guarantees the convergence of the learning process on the target domain.

Theorem 1. *The training process in MTEM can achieve $\mathbb{E}[\|\nabla_\psi \mathcal{L}_S(\hat{\theta}_t(\psi_t)|D_S)\|_2^2] \le \epsilon$ in $\mathcal{O}(\frac{1}{\epsilon^3})$ steps:*

$$\min_{0\le t\le T} \mathbb{E}[\|\nabla_\psi \mathcal{L}_S(\hat{\theta}_t(\psi_t)|D_S)\|_2^2] \le \mathcal{O}(\frac{C}{\sqrt[3]{T}})$$

where C is an independent constant.

Theorem 2. *With the training process in MTEM, the instance adaptive Tsallis entropy is guaranteed to be converged on unlabeled data. Formally,*

$$\lim_{t\to\infty} \mathbb{E}[\|\nabla_\theta \mathcal{L}_T(\theta_t,\psi_{t+1}|D_T^u)\|_2^2] = 0 \tag{16}$$

We use hypothesis $h : \mathcal{X} \to \Delta^{K-1}$ to analyze the effectiveness of MTEM in achieving domain adaptation. Formally, $h_\theta(x_i) = \arg\max_k f_{[k]}(x_i;\theta)$. We let $R_D(h)$ denote the model's robustness to the perturbations on dataset D. We let $\hat{\mathcal{R}}(\mathcal{H}|_D)$ denote the Rademacher complexity [Gnecco *et al.*, 2008] of function class $\mathcal{H}$ ($h \in \mathcal{H}$) on dataset D. Rademacher complexity evaluates the ability of the worst hypothesis $h \in \mathcal{H}$ in fitting random labels. If there exists a $h \in \mathcal{H}$ that fits most random labels on D, then $\hat{\mathcal{R}}(\mathcal{H}|_D)$ is large. With the above definitions, we deduce Theorem 3.

Theorem 3. *Suppose D_S and D_T^u satisfy $(q,c)-$ **constant expansion** [Wei* et al.*, 2021] for some constant $q, c \in (0,1)$. With the probability at least $1-\delta$ over the drawing of D_T^u*

from $\mathbb{D}_T$, the error rates of the model h_θ ($h \in \mathcal{H}$) on the target domain (i.e., $\epsilon_{\mathbb{D}_T}(h_\theta)$) is bounded by:

$$\begin{aligned}\epsilon_{\mathbb{D}_T}(h_\theta) \quad \leq \quad & \mathcal{L}_S(\theta|D_S) + \mathcal{L}_T(\theta,\psi|D_T^u) + 2q + 2K\cdot\hat{\mathcal{R}}(\mathcal{H}|_{D_S}) \\ & +4K\cdot\hat{\mathcal{R}}(\tilde{\mathcal{H}}\times\mathcal{H}|_{D_T^u}) + \frac{R_{D_S\cup D_T^u}(h)}{\min\{c,q\}} + \zeta \qquad (17)\end{aligned}$$

where $\zeta = \mathcal{O}(\sqrt{\frac{-log(\delta)}{|D_S|}} + \sqrt{\frac{-log(\delta)}{|D_T^u|}})$ is a low-order term. $\tilde{\mathcal{H}}\times\mathcal{H}$ refers to the function class $\{x \to h(x)_{[h'(x)]} : h, h' \in \mathcal{H}\}$.

With Theorem 3, we demonstrate that:

1. Theorem 1 and Theorem 2 prove that MTEM can simultaneously optimize the ψ and θ to minimize $\mathcal{L}_S(h|D_S)+\mathcal{L}_T(h,\psi|D_T^u)$, i.e., the first two term in Eq. (17).
2. With the bi-level optimization process, the learning process on D_T^u is regularized by supervised loss on the source domain. As D_S does not overlap with D_T^u, fitting the random labels on D_T^u cannot carry out the decrease of the supervised loss on the source domain (i.e., $\mathcal{L}_S(\theta|D_S)$). Thus, $h \in \mathcal{H}$ fits less noise on D_T^u, reducing $\hat{\mathcal{R}}(\tilde{\mathcal{H}}\times\mathcal{H}|_{D_T^u})$. At the same time, as D_S is unseen in the training process, it is also hard to fit the random label on D_S, thereby reducing $\hat{\mathcal{R}}(\mathcal{H}|_{D_S})$.
3. Instance adaptive Tsallis-entropy is an unsupervised loss. As accessing unlabeled data is easier than accessing the labeled data, MTEM provides the possibility of sampling a larger unlabeled data to make ζ smaller.
4. $R_{D_S\cup D_T^u}(h)$ is a term that can be minimized in the training process technically, e.g., adversarial training [Jiang *et al.*, 2020] or SAM (Sharpness-Aware-Minimization) optimizer [Foret *et al.*, 2020].

5 Experiments

5.1 Experiment Settings

Datasets. On the rumor detection task, we conduct experiments with the dataset TWITTER [Zubiaga *et al.*, 2016], which contains five domains: "Cha.", "Ger.", "Fer.", "Ott.", and "Syd.". On the sentiment classification task, we conduct experiments with the dataset Amazon [Blitzer *et al.*, 2007], which contains four domains: books, dvd, electronics, and kitchen. Preprocess and statistics on the TWITTER dataset and the Amazon dataset can be found in Appendix D.

Comparing Methods. We compare MTEM with previous domain adaptation approaches on both *semi-supervised*[5] and *unsupervised* domain adaptation scenarios. Under the unsupervised domain adaptation, we compare MTEM with Out [Chen *et al.*, 2021], DANN [Ganin *et al.*, 2016], FixMatch [Sohn *et al.*, 2020], and CST [Liu *et al.*, 2021]. Under the semi-supervised domain adaptation, MTEM[6] is compared with In+Out [Chen *et al.*, 2021], MME [Saito *et al.*, 2019], BiAT [Jiang *et al.*, 2020], and Wind [Chen *et al.*, 2021]. Out and In+Out are two straightforward ways for realizing unsupervised and semi-supervised domain adaptation, where Out means the base model is trained on the out-of-domain data (i.e., D_S) and In+Out means the base model is trained on both the in-domain and the out-of-domain data. DANN realizes domain adaptation by min-max the domain classification loss. CST and FixMatch are self-training approaches that generates pseudo instances to augment domain adaptation. Although CST also involves Tsallis entropy, the entropy-index is a mannually set hyper-parameters and is not instance-adaptative. WIND is a meta-reweigting based domain adaptation approach that learns-to-learn suitable instance weights of different labeled samples in the source domain. More details about the baseline methods can be found in the references.

Implementation Details. The base model on the Amazon dataset is BERT [Devlin *et al.*, 2019] and the base model on the TWITTER dataset is BiGCN [Bian *et al.*, 2020]. Domain adaptation experiments are conducted on every domain on the benchmark datasets. For every domain on the benchmark dataset, we seperately take them as the target domain and merges the rest domains as the source domain. For example, when the "books" domain in the Amazon dataset is taken as the target domain, the "dvd", "electronics" and "kitchen" domains are merged as the source domain. All unlabeled data from the target domain are involved in the training process, meanwhile the labeled data in the target domain are used for evaluation (with a ratio of 7:3). Since the TWITTER dataset does not contain extra unlabeled data, we take 70% of the labeled data on the target domain as the unlabeled data for training the model and preserve the rest ones for evaluation. The experiments on TWITTER are conducted on "Cha.", "Fer.", "Ott.", and "Syd."[7]. For the symbols in Algorithm 1, we set η_t and β_t with respect to Assumption 1.

5.2 General Results

We use all baseline approaches (including MTEM) to adapt BiGCN across domains on TWITTER, and to adapt BERT across domains on Amazon. We validate the effectiveness of the proposed MTEM on both unsupervised and semi-supervised domain adaptation scenarios. For semi-supervised domain adaptation scenario, 100 labeled instances in the target domain are randomly selected as the in-domain dataset. As the rumor detection task mainly concerns the classification performance in the 'rumor' category, we use the F1 score to evaluate the performance on TWITTER. On the sentiment classification task, different classes are equally important. Thus, we use the accuracy score to evaluate different models, which is also convenient for comparison with previous studies. Experiment results are listed in Table 1, Table 2.

The results in Table 1 and Table 2 demonstrate the effectiveness of the proposed MTEM algorithm. In particular, MTEM outperforms all baseline approaches on all benchmark datasets. Compared with the self-training approaches, i.e., FixMatch and CST, MTEM maintains the superiority of an average of nearly 2 percent on the Amazon dataset and an average of 4 percent on the TWITTER set. Thus, regularizing

[5] A small set of labeled data in the target domain can be accessed, named *in-domain* dataset.

[6] For semi-supervised domain adaptation, we insert the labeled target data into D_S to run MTEM.

[7] The labeled data in "Ger." domain is too scare to provide extra unlabeled data.

Base Model (BiGCN)	Unsupervised domain adaptation					Semi-Supervised domain adaptation				
	Out	DANN	FixMatch	CST	MTEM	In+Out	MME	BiAT	Wind	MTEM
Cha.	0.561	0.501	0.614	0.573	**0.627**	0.586	0.601	0.547	0.552	**0.637**
Fer.	0.190	0.387	0.473	0.446	**0.549**	0.200	0.381	0.256	0.291	**0.635**
Ott.	0.575	0.544	0.672	0.649	**0.728**	0.599	0.612	0.614	0.633	**0.817**
Syd.	0.438	0.461	0.694	0.653	**0.729**	0.424	0.677	0.661	0.628	**0.750**
Mean	0.441	0.473	0.613	0.598	**0.658**	0.452	0.567	0.520	0.526	**0.709**

Table 1: F1 scores on the TWITTER dataset

Base Model (BERT)	Unsupervised Domain Adaptation					Semi-Supervised Domain Adaptation				
	Out	DANN	CST	FixMatch	MTEM	In+Out	MME	BiAT	WIND	MTEM
books	0.902	0.912	0.912	0.906	**0.939**	0.912	0.923	0.922	0.917	**0.946**
dvd	0.902	0.909	0.923	0.907	**0.937**	0.908	0.924	0.903	0.911	**0.947**
electronics	0.894	0.934	0.923	0.913	**0.935**	0.926	0.927	0.930	0.931	**0.945**
kitchen	0.895	0.934	0.924	0.922	**0.937**	0.931	0.931	0.933	0.940	**0.942**
Mean	0.898	0.922	0.920	0.912	**0.937**	0.919	0.926	0.922	0.925	**0.945**

Table 2: Accuracy scores on the Amazon dataset

the self-training process with an instance adaptative is beneficial. Moreover, MTEM also surpasses the meta-reweighting algorithm, i.e., WIND, by an average of nearly 2 percent on the Amazon dataset and nearly 18 percent on the TWITTER dataset. Thus, the meta-learning algorithm in MTEM, i.e., learning to learn the suitable entropy indexes, is a competitive candidate in the domain adaptation scenario.

5.3 Ablation Study

We separately remove the meta-learning module (- *w/o M*), the temperature scheduler (- *w/o T*), and the sampling mechanism (- *w/o S*) to observe the adaptation performance across domains on the benchmark datasets. - *w/o M* means all instances in the target domain will be allocated the same entropy index (determined with manually attempt). - *w/o T* means removing the temperature scheduler, and the temperature κ is fixed to be 1.0. - *w/o S* means to remove the sampling mechanism, i.e., generates pseudo labels with greedy strategies as previous self-training approaches. The experiments are conducted under the unsupervised domain adaptation scenarios. We validate the effectiveness with F1 score on TWITTER, and use the accuracy score on Amazon. The experiment results are listed in Tab. 3 and Tab. 4.

From Tab. 3 and Tab. 4, we can find that all variants perform worse than MTEM on two benchmark datasets: (i) MTEM surpasses MTEM - *w/o M* on the Amazon dataset with an average of 2 percent, and on the TWITTER dataset with an average of 7 percent. Thus, allocating an instance adaptative entropy index for every unlabeled instance in the target domain is superior to allocating the same entropy index. Furthermore, since the unified entropy index in MTEM - *w/o M* is searched manually, MTEM - *w/o M* should be better than Gibbs Entropy. Otherwise, the entropy index would be determined as 1.0 (Gibbs Entropy). Thus, the instance adaptive Tsallis-entropy in MTEM is better than Gibbs Entropy. (ii) MTEM surpasses MTEM - *w/o S* on the Amazon dataset with an average of 1.4 percent, and on the TWITTER dataset with an average of 1.5 percent, which is attributed to the sampling mechanism can directly correct the model's prediction errors. (iii) MTEM surpasses MTEM - *w/o T* with an average decrease of 0.9 percent on the TWITTER dataset, and with an average of 0.7 percent on the Amazon dataset, which is consistent with our claims that the annealing mechanism is beneficial to align the domains gradually.

Domain	Cha.	Fer.	Ott.	Syd.	Mean
MTEM	**0.627**	**0.549**	**0.728**	**0.729**	**0.658**
- *w/o M*	0.569	0.452	0.633	0.647	0.575
- *w/o A*	0.621	0.537	0.716	0.722	0.649
- *w/o S*	0.622	0.529	0.707	0.714	0.643

Table 3: Ablation Study on TWITTER

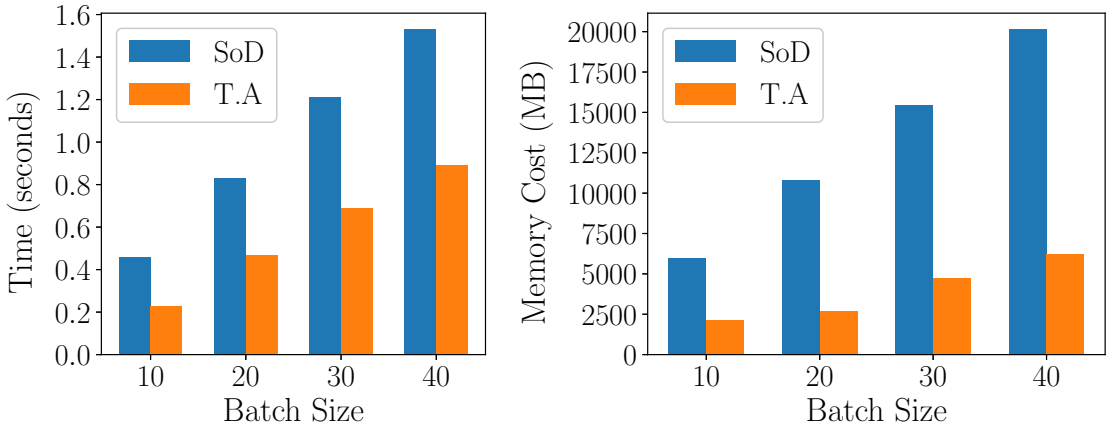
Domain	books	dvd	electronics	kitchen	Mean
MTEM	**0.939**	**0.937**	**0.935**	**0.937**	**0.937**
- *w/o M*	0.912	0.917	0.919	0.919	0.916
- *w/o A*	0.931	0.935	0.927	0.929	0.930
- *w/o S*	0.929	0.912	0.927	0.922	0.923

Table 4: Ablation Study on the Amazon dataset

5.4 Computation Cost

We conduct experiments on the Amazon dataset to compare the computation cost in the Taylor approximation and the original Second-order derivation. We separately count the time and the memory consumed in computing the gradient of the entropy indexes with different batch sizes. Experiments are deployed on an Nvidia Tesla V100 GPU. From Fig. 3, the time cost in the Second-order derivation is almost two times higher than in the Taylor approximation, and the memory cost in the Second-order derivation is 3-4 times higher than in the Taylor approximation technique.

We also count the different performances of adapting the BERT model to the 'kitchen' domain with respect to different batch sizes. The experiment results are listed in Tab. 5, where '/' means the memory cost is out of the device's capacity. From Tab. 5, we can observe that the Taylor Approximation technique keeps a similar performance with the Second-order derivation. What's more, the best performance is achieved



(a) Time Cost (b) Memory Cost

Figure 3: Computation Cost: Taylor Approximation (T.A) v.s. Second-order Derivation (SoD).

Batch Size	10	20	30	40	50	60
T.A	0.873	0.892	0.914	0.924	0.935	0.935
SoD	0.876	0.897	0.915	0.927	/	/

Table 5: Domain Adaptation with Different Batch Size: Taylor Approximation (T.A) v.s. Second-order Derivation (SoD)

$\psi \approx 1.0$	i bought this at amazon , but it's cheaper at cutlery and more, $ 9.95 , so is the $ 89.00 wusthof santoku 7 knife ($ 79.00), and they have free shipping ! check yahoo shopping before amazon ! ! !
ψ=5.0	i like and dislike these bowls, what i like about them is the shape and size for certain foods and for the dish--washer. they are too small for cereal if you would like to add fruit to your cereal, perfect for oatmeal or ice cream but too small for soup or stew.

Table 6: Unlabeled instances with different entropy index (ψ).

by using a batch with more than 50 instances (the setting in § 5.2), which would exceed the memory capacity if we use the Second-order derivation. Thus, the benefit of reducing the computation cost is apparent, as a larger batch size leads to better adaptation performance.

5.5 Case Study

In Tab. 6, we present two cases with different entropy indexes learned in the meta-learning process (more cases are provided in Appendix. E). Experiments are conducted on sentiment classification tasks, and the settings are the same as 'kitchen' in § 5.2. On the sentences with smaller entropy index (updating the model aggressively), the sentiment words are more transferrable across domains in e-commerce , e.g., 'cheaper' and 'free shipping'. Otherwise, sentences with larger entropy index contain more domain discriminative words, e.g., the n-gram 'but too small for soup or stew' in the kitchen domain are less relevant to the other domains (electronics, books, dvd). In this case, MTEM uses a large entropy index to update the model more cautiously.

6 Related Work

6.1 Domain Adaptation

To adapt a model to a new domain, feature-alignment approaches [Ganin *et al.*, 2016; Saito *et al.*, 2019; Saito *et al.*, 2019] focus on explicitly aligning the feature space across domains. For example, DANN [Ganin *et al.*, 2016] proposes to align the feature space by min-max the domain classification loss. With similar efforts, MME [Saito *et al.*, 2019] min-max the conditional entropy on the unlabeled data. BiAT [Jiang *et al.*, 2020] proposes to decouple the min-max optimization process in DANN, i.e., firstly maximize the risk loss to obtain a gradient-based perturbation on the input space and then minimize the objective on the perturbed input cases. On the other hand, data-centric approaches use the unlabeled data in the target domain or the labeled data from the source domain to align the feature space implicitly. To select labeled data from the source domain, researchers [Moore and Lewis, 2010] design a technique based on topic models for measuring the domain similarity, while [Chen *et al.*, 2021] takes a meta-learning algorithm to implicitly measure the domain similarity. To exploit the unlabeled data from the target domain, pseudo labeling approaches, including self-training [Zou *et al.*, 2019], co-training [Chen *et al.*, 2011], and tri-training [Saito *et al.*, 2017], are widely applied and become an important direction. The difference lies in that self-training [Zou *et al.*, 2018; Zou *et al.*, 2019; Liu *et al.*, 2021] uses the model's prediction to improve the model, while co-training [Chen *et al.*, 2011] and tri-training [Saito *et al.*, 2017] involves more models which learn the task information from each other. In the research of self-training for domain adaptation, many efforts tried to use prediction confidence to reduce the label noise of pseudo instances [Zou *et al.*, 2018; Zou *et al.*, 2019; Liu *et al.*, 2021], i.e., they preserve only the easy examples that have high prediction confidences while discarding the hard examples that have low prediction confidences. However, fitting the model on easy pseudo instances cannot effectively improve the performance, as the model is already

confident about its prediction.

6.2 Meta-Learning

Meta-learning is an emerging new branch in machine learning that aims to train a model that can adapt to a new task or new domain quickly given a few new samples. For this purpose, previous studies tried to learn better initial model parameters [Finn *et al.*, 2017], or better learning rates [Li *et al.*, 2017]. *Learning to Compare* methods, e.g., relation network [Sung *et al.*, 2018] and prototypical learning [Snell *et al.*, 2017], are investigated more widely in text classification tasks[Tan *et al.*, 2019; Geng *et al.*, 2020]. With the recent success of the pre-trained language model, Network Architect Search (NAS) methods, e.g., DARTs [Liu *et al.*, 2018]), are also widely studied in Natural Language Processing (NLP) tasks [Xu *et al.*, 2021; Dong *et al.*, 2021]. Some meta-learning algorithm learns to knowledge distillation [Zhou *et al.*, 2022], i.e., increase the number of teacher models to train a meta-teacher model that works better than a single teacher model. Meta reweighting algorithm [Ren *et al.*, 2018], which proposes to dynamically reweight the risk on different instances, has also inspired some NLP tasks [Li *et al.*, 2020; Chen *et al.*, 2021]. Our research is similar to the meta reweighting algorithm, i.e., the training objective is instance adaptive, the difference lies in that the entropy index controls the loss function on different instances while the instance weights do not change the loss function.

7 Conclusion

This paper proposes a new meta-learning algorithm for domain adaptation on text classification, namely MTEM. Inheriting the principle of entropy minimization, MTEM imposes an instance adaptative Tsallis entropy minimization process on the target domain, and such a process is formulated as a meta-learning process. To reduce the computation cost, we propose a Taylor approximation technique to compute the gradient of the entropy indexes. Also, we propose an annealing sampling mechanism to generate pseudo labels. In addition, we analyze the proposed MTEM theoretically, i.e., we prove the convergence of the meta-learning algorithm in optimizing the instance-adaptative entropy and provide insights for understanding why MTEM is effective in achieving domain adaptation. Extensive experiments on two popular models, BiGCN and BERT, verify the effectiveness of MTEM.

Acknowledgements

This work is supported by the following foundations: the National Natural Science Foundation of China under Grant No. 62025208, the Xiangjiang Laboratory Foundation under Grant No. 22XJ01012, 2022 International Postdoctoral Exchange Fellowship Program (Talent-Introduction Program) under Grant No. YJ20220260.

Contribution Statement

Menglong Lu and Zhen Huang contributed equally to this work. Zhiliang Tian and Yunxiang Zhao are the corresponding authors.

References

[Benson, 1995] Harold P Benson. Concave minimization: theory, applications and algorithms. In *Handbook of Global Optimization*, pages 43–148. Springer, 1995.

[Bian *et al.*, 2020] Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. Rumor detection on social media with bi-directional graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 549–556, 2020.

[Blitzer *et al.*, 2007] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, pages 440–447, 2007.

[Chen *et al.*, 2011] Minmin Chen, Kilian Q Weinberger, and John C Blitzer. Co-training for domain adaptation. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 2456–2464, 2011.

[Chen *et al.*, 2021] Xiang Chen, Yue Cao, and Xiaojun Wan. Wind: Weighting instances differentially for model-agnostic domain adaptation. In *Findings of the Annual Meeting of the Association for Computational Linguistics*, pages 2366–2376, 2021.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, 2019.

[Dong *et al.*, 2021] Chenhe Dong, Guangrun Wang, Hang Xu, Jiefeng Peng, Xiaozhe Ren, and Xiaodan Liang. Efficientbert: Progressively searching multilayer perceptron via warm-up knowledge distillation. In *Findings of the Association for Computational Linguistics*, pages 1424–1437, 2021.

[Du *et al.*, 2020] Chunning Du, Haifeng Sun, Jingyu Wang, Qi Qi, and Jianxin Liao. Adversarial and domain-aware bert for cross-domain sentiment analysis. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 4019–4028, 2020.

[Finn *et al.*, 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135, 2017.

[Foret *et al.*, 2020] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2020.

[Ganin *et al.*, 2016] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17:2096–2030, 2016.

[Geng *et al.*, 2020] Ruiying Geng, Binhua Li, Yongbin Li, Jian Sun, and Xiaodan Zhu. Dynamic memory induction

networks for few-shot text classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1087–1094, 2020.

[Gnecco *et al.*, 2008] Giorgio Gnecco, Marcello Sanguineti, et al. Approximation error bounds via rademacher complexity. *Applied Mathematical Sciences*, 2:153–176, 2008.

[Jiang *et al.*, 2020] Pin Jiang, Aming Wu, Yahong Han, Yunfeng Shao, Meiyu Qi, and Bingshuai Li. Bidirectional adversarial training for semi-supervised domain adaptation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 934–940, 2020.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations*, pages 1592–1601, 2017.

[Kumar *et al.*, 2010] M Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. *Advances in Neural Information Processing Systems*, 23, 2010.

[Kumar *et al.*, 2020] Ananya Kumar, Tengyu Ma, and Percy Liang. Understanding self-training for gradual domain adaptation. In *International Conference on Machine Learning*, pages 5468–5479, 2020.

[Lee and others, 2013] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *International Conference on Machine Learning*, page 896, 2013.

[Li *et al.*, 2017] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few shot learning. *CoRR*, abs/1707.09835, 2017.

[Li *et al.*, 2020] Zhenzhen Li, Jian-Yun Nie, Benyou Wang, Pan Du, Yuhan Zhang, Lixin Zou, and Dongsheng Li. Meta-learning for neural relation classification with distant supervision. In *Proceedings of the ACM International Conference on Information & Knowledge Management*, pages 815–824, 2020.

[Liu *et al.*, 2018] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *Proceedings of the International Conference on Learning Representations*, pages 934–940, 2018.

[Liu *et al.*, 2021] Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle self-training for domain adaptation. *Advances in Neural Information Processing Systems*, 34:22968–22981, 2021.

[Lu *et al.*, 2022] Menglong Lu, Zhen Huang, Binyang Li, Yunxiang Zhao, Zheng Qin, and DongSheng Li. Sifter: A framework for robust rumor detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:429–442, 2022.

[McClosky *et al.*, 2006] David McClosky, Eugene Charniak, and Mark Johnson. Reranking and self-training for parser adaptation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 337–344, 2006.

[Moore and Lewis, 2010] Robert C. Moore and William D. Lewis. Intelligent selection of language model training data. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 220–224, 2010.

[Mukherjee and Awadallah, 2020] Subhabrata Mukherjee and Ahmed Awadallah. Uncertainty-aware self-training for few-shot text classification. *Advances in Neural Information Processing Systems*, 33:21199–21212, 2020.

[Plastino and Plastino, 1999] ARPA Plastino and AR Plastino. Tsallis entropy and jaynes' information theory formalism. *Brazilian Journal of Physics*, 29:50–60, 1999.

[Reichart and Rappoport, 2007] Roi Reichart and Ari Rappoport. Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, pages 616–623, 2007.

[Ren *et al.*, 2018] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *Proceedings of the International Conference on Machine Learning*, pages 4334–4343, 2018.

[Rotman and Reichart, 2019] Guy Rotman and Roi Reichart. Deep contextualized self-training for low resource dependency parsing. *Transactions of the Association for Computational Linguistics*, 7:695–713, 2019.

[RoyChowdhury *et al.*, 2019] Aruni RoyChowdhury, Prithvijit Chakrabarty, Ashish Singh, SouYoung Jin, Huaizu Jiang, Liangliang Cao, and Erik Learned-Miller. Automatic adaptation of object detectors to new domains using self-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 780–790, 2019.

[Saito *et al.*, 2017] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 2988–2997, 2017.

[Saito *et al.*, 2019] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8050–8058, 2019.

[Shin *et al.*, 2020] Inkyu Shin, Sanghyun Woo, Fei Pan, and In So Kweon. Two-phase pseudo label densification for self-training based domain adaptation. In *European Conference on Computer Vision*, pages 532–548, 2020.

[Snell *et al.*, 2017] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, 2017.

[Sohn *et al.*, 2020] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33:596–608, 2020.

[Sung *et al.*, 2018] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[Tan *et al.*, 2019] Ming Tan, Yang Yu, Haoyu Wang, Dakuo Wang, Saloni Potdar, Shiyu Chang, and Mo Yu. Out-of-domain detection for low-resource text classification tasks. In *Conference on Empirical Methods in Natural Language Processing*, 2019.

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

[Xu *et al.*, 2021] Jin Xu, Xu Tan, Renqian Luo, Kaitao Song, Jian Li, Tao Qin, and Tie-Yan Liu. Nas-bert: task-agnostic and adaptive-size bert compression with neural architecture search. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1933–1943, 2021.

[Zhou *et al.*, 2022] Wangchunshu Zhou, Canwen Xu, and Julian McAuley. Bert learns to teach: Knowledge distillation with meta learning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 7037–7049, 2022.

[Zou *et al.*, 2018] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European Conference on Computer Vision*, pages 289–305, 2018.

[Zou *et al.*, 2019] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5982–5991, 2019.

[Zubiaga *et al.*, 2016] Arkaitz Zubiaga, Maria Liakata, and Rob Procter. Learning reporting dynamics during breaking news for rumour detection in social media. *CoRR*, abs/1610.07363, 2016.