Open-Set Semi-Supervised Text Classification via Adversarial Disagreement Maximization

Junfan Chen[1,2], **Richong Zhang**[1,3*], **Junchi Chen**[1], **Chunming Hu**[1,2,3]
[1]CCSE, School of Computer Science and Engineering, Beihang University, Beijing, China
[2]School of Software, Beihang University, Beijing, China
[3]Zhongguancun Laboratory, Beijing, China
{chenjf, zhangrc}@act.buaa.edu.cn, {sy2206115, hucm}@buaa.edu.cn

Abstract

Open-Set Semi-Supervised Text Classification (OSTC) aims to train a classification model on a limited set of labeled texts, alongside plenty of unlabeled texts that include both in-distribution and out-of-distribution examples. In this paper, we revisit the main challenge in OSTC, i.e., outlier detection, from a measurement disagreement perspective and innovatively propose to improve OSTC performance by directly maximizing the measurement disagreements. Based on the properties of in-measurement and cross-measurements, we design an Adversarial Disagreement Maximization (ADM) model that synergeticly optimizes the measurement disagreements. In addition, we develop an abnormal example detection and measurement calibration approach to guarantee the effectiveness of ADM training. Experiment results and comprehensive analysis of three benchmarks demonstrate the effectiveness of our model.

1 Introduction

Text classification is a fundamental task in natural language processing. With the development of modern deep learning techniques, text classification has achieved significant advancement. However, deep learning models usually require substantial labeled data, which is expensive in many real-world applications. To tackle this problem, Semi-supervised Text Classification (STC) has been proposed, which only needs a small set of labeled examples along with plenty of unlabeled examples (Lee et al., 2013; Tarvainen and Valpola, 2017; Meng et al., 2018; Gururangan et al., 2019; Chen et al., 2020; Lee et al., 2021; Tsai et al., 2022; Yang et al., 2023). By utilizing unlabeled texts in training machine learning models, these approaches reduce the need to expensively annotate abundant data. However, the STC assumption that all unlabeled texts are sampled within the intended scope is impractical in real-world applications. Therefore, researchers recently explored Open-set Semi-supervised Text Classification (OSTC) (Chen et al., 2023), which allows the inclusion of out-of-distribution examples in the unlabeled text set.

The main challenge in OSTC is the commonly known false positive inference problem (Chen et al., 2023), which indicates a phenomenon that out-of-distribution texts are prone to be recognized as an in-distribution class, leading to unsatisfactory OSTC outcomes. To address this issue, prior studies have integrated an STC model with different outlier detection techniques. For example, LMCL (Lin and Xu, 2019) and Softmax (Yan et al., 2020) learn discriminative embeddings and utilize local outlier factor (LOF) (Breunig et al., 2000) to identify in-distribution (ID) and out-of-distribution (OOD) examples. MSP (Hendrycks and Gimpel, 2017) and LOS (Chen et al., 2023) respectively employ maximum softmax probability and normalized entropy function to identify ID and OOD examples.

We provide a general understanding of existing OSTC models. Specifically, we revisit outlier detection in existing methods from a measurement disagreement perspective. It can be concluded that existing approaches are fundamentally grounded in the assumption that the differentiation between ID and OOD examples hinges on the presence of *disagreement* in specific measurements associated with the examples. For instance, the density measurement in LOF of LMCL and Softmax, and the confidence and entropy measurements in MSP and LOS. With this assumption, existing methods distinguish ID and OOD examples by setting certain thresholds on the measurement.

The above discussion motivates us to answer a question: Since ID and OOD examples can be more easily distinguished when the measurement disagreement is larger, can we improve outlier detection in OSTC by directly maximizing such measurement disagreement between ID and OOD exam-

* Corresponding author

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2170–2180

ples? To answer this question, we make formal definitions and assumptions for measurement disagreements and reveal several useful properties: (1) the disagreement bounds can be increased by maximizing in-measurement disagreement between ID and OOD examples. (2) the comparative consistency assumption allows us to synergisticly optimize two different measurements. (3) the in-example consistency assumption enables us to detect abnormal examples and calibrate measurements.

Based on the above motivations, we propose an Adversarial Disagreement Maximization (ADM) model for OSTC. Concretely, we treat the cross-entropy loss of the ID softmax classifier and the outlier detection confidence as two different measurements. To perform a synergistically measurement optimization, we leverage an adversarial learning approach to iteratively enlarge the disagreements of the two measurements. In addition, to guarantee the effectiveness of disagreement maximization, we design an abnormal-example detection approach to correct the measurement optimization direction.

To summarize, we make the following contributions: (1) We provide a general understanding of outlier detection in OSTC and revisit it from a measurement disagreement perspective. (2) We propose an ADM model to directly maximize measurement disagreements combined with measurement calibration. (3) We evaluate our ADM model on three benchmark datasets and demonstrate its effectiveness by comprehensive analysis.

2 Related Works

Semi-supervised text classification (STC) trains a model with a few labeled texts and many unlabeled texts. Various regularization techniques, consistency training approach are developed for STC (Miyato et al., 2017; Gururangan et al., 2019; Liu et al., 2021; Li et al., 2021; Xu et al., 2022). Among these models, UDA (Xie et al., 2020) utilizes substituting noising operations to construct data augmentations combined with consistency training. MixText (Chen et al., 2020) design a novel text data augmentation method with Mixup and used in consistency training. Some other works leverage pseudo-labeling to annotate unlabeled texts as additional training data (Lee et al., 2013; Tarvainen and Valpola, 2017; Meng et al., 2018; Li et al., 2021; Lee et al., 2021; Li et al., 2022b).

Open-set semi-supervised learning is understudied task. The pipeline approaches first filters out OOD examples and then conduct semi-supervised training with filtered data. These methods often design special outlier detectors (Saito et al., 2021; Huang et al., 2021; Liu et al., 2022) or build new optimization process (Yu et al., 2020; Zhu and Li, 2022; Li et al., 2022a). In recent work LOS (Chen et al., 2023), the authors design a set of pipeline methods utilizing different outlier detection approaches, including MSP (Hendrycks and Gimpel, 2017), DOC (Shu et al., 2017), LMCL (Lin and Xu, 2019) and LSoftmax (Yan et al., 2020). LOS unify semi-supervised training and outlier detection within probabilistic latent variable modeling. We understand OSTC in a new measurement disagreement perspective and leverages adversarial learning (Goodfellow et al., 2014) to achieve disagreement maximization.

3 Methodology

3.1 Prior Art

Task Definition. In open-set semi-supervised text classification (OSTC), we expect to train a text classification model with a few labeled texts and many unlabeled texts. The unlabeled texts contain both in-distribution (ID) and out-of-distribution (OOD) examples. We will use $\mathcal{Y}$ to denote the set of in-distribution classes. We assume that we have access to a labeled text set $\mathcal{L} = \{(x_l, y_l)\}_{i=1}^{L}$ with each class involving k examples, an unlabeled text set $\mathcal{U} = (\mathcal{U}_+, \mathcal{U}_-) = \{x_i\}_{i=1}^{n}$, which consists of an in-distribution text set $\mathcal{U}_+$ and an out-of-distribution text set $\mathcal{U}_-$. The goal of the OSTC task is to identify whether a given text in the test set is an ID or OOD example and predict the exact class type if the given text is an ID example.

Outlier Detection in OSTC. Outlier detection is a key component in OSTC. It contributes to model training and evaluation of OSTC. As demonstrated in previous work (Chen et al., 2023), the key challenge in OSTC is the false positive inference problem, which forces the OOD texts to be recognized as an ID class when conducting semi-supervised learning. An intuitive solution to this problem is utilizing outlier detection to filter unlabeled OOD examples during semi-supervised training. Second, the outlier detector is required to identify whether a given text is an ID or OOD example during evaluation. Considering the vital role of outlier detection in OSTC, in this work, we provide a general understanding of it and propose a novel optimization framework to improve its performance.

Table 1: Analysis for Existing Outlier Detection Methods in OSTC

Outlier Detection Model	$\mathcal{M}(f;x,\mathcal{Y})$	OOD Identification Condition
MSP (Hendrycks and Gimpel, 2017)	$\max_{y\in\mathcal{Y}} \frac{f(x,y)}{\sum_{c\in\mathcal{Y}} f(x,c)}$	$\mathcal{M}(f;x,\mathcal{Y}) < \eta_1$
DOC (Shu et al., 2017)	$\max_{y\in\mathcal{Y}} f_y(x)$	$\mathcal{M}(f;x,\mathcal{Y}) < \eta_2$
LMCL (Lin and Xu, 2019) LSoftmax (Yan et al., 2020)	$\frac{\sum_{x'\in\mathcal{N}_k(x)} f(x')}{\lvert\mathcal{N}_k(x)\rvert\cdot f(x)}$	$\mathcal{M}(f;x,\mathcal{Y}) > 1$
LOS (Chen et al., 2023)	$\frac{\mathcal{H}^\lambda(f(x))}{(\log\lvert\mathcal{Y}\rvert)^\lambda}$	$\mathcal{M}(f;x,\mathcal{Y}) > 0.5$

3.2 Motivation

Revisit Outlier Detection from Measurement Disagreement Perspective. To tackle the false positive inference problem, existing works either adopt a pipeline approach that first trains an outlier detector to filter the OOD examples and then conducts semi-supervised training on the rest of unlabeled data (Shu et al., 2017; Yan et al., 2020) or integrate supervised training and outlier detection as a unified framework (Chen et al., 2023) during optimization. We provide a formal understanding of these outlier detection methods and reveal that all of these methods identify OOD examples following a *measurement disagreement* assumption. Formally, define a measurement $\mathcal{M}(f;x,\mathcal{Y})$, which involves an internal function f and two inputs: a text example $x\in\mathcal{U}$ and the in-distribution class set $\mathcal{Y}$. Based on the specified measurement $\mathcal{M}$, existing outlier detection methods introduce a threshold to distinguish ID and OOD examples. Concretely, we give the measurement formulation and OOD identification condition for each outlier detection method in Table 4 and explain as follows:

- **MSP**: $\mathcal{M}$ is the maximum softmax probability and f is specified as a softmax classifier, with a condition $\mathcal{M} < \eta_1$.
- **DOC**: $\mathcal{M}$ is the maximum 1-vs-rest probability and each f is a 1-vs-rest sigmoid classifier, with a condition $\mathcal{M} < \eta_2$.
- **LMCL** and **LSoftmax**: $\mathcal{M}$ is the Local Outlier Factor and f is the local reachability density (Breunig et al., 2000). $\mathcal{N}_k$ are the k-nearest neighbors with a condition $\mathcal{M} > 1$.
- **LOS**: $\mathcal{M}$ is the normalized entropy and f is a softmax classifier, with a condition $\mathcal{M} > 0.5$. $\mathcal{H}$ is the entropy function.

The above analysis indicates that the OOD examples are detected under the assumption that the measurements of an ID example $x_+\in\mathcal{U}_+$ and an OOD example $x_-\in\mathcal{U}_-$ satisfy the following disagreement with some specified threshold η:

$$\begin{cases}\mathcal{M}(f;x_+,\mathcal{Y})<\eta,\\ \mathcal{M}(f;x_-,\mathcal{Y})\geqslant\eta.\end{cases} \text{ or } \begin{cases}\mathcal{M}(f;x_+,\mathcal{Y})>\eta,\\ \mathcal{M}(f;x_-,\mathcal{Y})\leqslant\eta\end{cases} \quad (1)$$

Formal Definitions for Measurement Disagreement. For the convenience of formal analysis, we make the following definitions:

Definition 1 Measurement Disagreement. *For any ID example $x_+\in\mathcal{U}_+$ and OOD example $x_-\in\mathcal{U}_-$, given a specified measurement $\mathcal{M}(f;x,\mathcal{Y})$, we define the measurement disagreement of the two examples x_+ and x_- as follows:*

$$d_{\mathcal{M}}(x_+,x_-)=|\mathcal{M}(f;x_+,\mathcal{Y})-\mathcal{M}(f;x_-,\mathcal{Y})| \quad (2)$$

We will also call $d_{\mathcal{M}}(x_+,x_-)$ the in-measurement disagreement of examples x_+ and x_-.

Definition 2 Cross-Measurement Disagreement. *For any example $x\in\mathcal{U}$ and two different specified measurements $\mathcal{M}_1(f;x,\mathcal{Y})$ and $\mathcal{M}_2(f;x,\mathcal{Y})$, we define the cross-measurement disagreement of example x under the two measurement as follows:*

$$d(x,\mathcal{M}_1,\mathcal{M}_2)=|\mathcal{M}_1(f;x,\mathcal{Y})-\mathcal{M}_2(f;x,\mathcal{Y})| \quad (3)$$

Definition 3 ϵ-Bounded Disagreement. *For a real number $\epsilon\geqslant 0$ and a given measurement $\mathcal{M}(f;x,\mathcal{Y})$, we define that ID examples $x_+\in\mathcal{U}_+$ and OOD examples $x_-\in\mathcal{U}_-$ has a ϵ-bounded disagreement, if the measurement disagreement for each (x_+,x_-) pair satisfy:*

$$d_{\mathcal{M}}(x_+,x_-)-\epsilon\geqslant 0 \quad (4)$$

Motivations from Measurement Disagreement. From Definition 1, we know that the measurement disagreement gives the amount of difference between the measurements of an ID and an OOD example. Thus, when the disagreement $d_{\mathcal{M}}(x_+,x_-)$

is larger, the model can distinguish x_+ and x_- more easily. This motivates us to improve outlier detection in OSTC by enlarging the measurement disagreements. However, as existing outlier detection methods are trained with softmax or sigmoid classifiers and representation learning losses for LOF in a pipeline or unified framework, *their training objectives are not developed to directly increase the measurement disagreements*. This design overlooks the potential advantage of enlarging measurement disagreements over outlier detection.

Furthermore, based on Definition 3, if a measurement disagreement $d_{\mathcal{M}}(x_+, x_-)$ satisfy ϵ-bounded disagreement, the worst case disagreement it can reach is ϵ. Thus, if we want to increase the measurement disagreements for (x_+, x_-) pairs, we may correspondingly maximize disagreement bound ϵ. From Formulation (1), we can easily derive that existing outlier detection methods only satisfy the *0-bounded disagreement*, which significantly limits the models' ability to distinguish ID and OOD examples. The above observations inspire us to ask: **Can we increase the disagreement bounds of outlier detection by directly maximizing the measurement disagreements?**

Another motivation is that we may unleash the advantages of cross-measurement disagreement in the outlier detection of OSTC. To formally understand the potential of cross-measurement disagreement, we make the following assumptions:

Assumption 1 Comparative Consistency. *For two examples $x, x' \in \mathcal{U}$ and given two different measurements $\mathcal{M}_1(f; x, \mathcal{Y})$ and $\mathcal{M}_2(f; x, \mathcal{Y})$, we assume the two measurements are comparative consistency. Namely, if $\mathcal{M}_1(f; x, \mathcal{Y}) > \mathcal{M}_1(f; x', \mathcal{Y})$, then $\mathcal{M}_2(f; x, \mathcal{Y}) > \mathcal{M}_2(f; x', \mathcal{Y})$ is satisfied.*

Under the above assumption, when comparing two different examples, different measurements demonstrate similar behavior. This property manifests that when one measurement is optimized on a set of examples, the other measurement may be accordingly optimized. Thus, we may jointly optimize two different measurements in a synergistic way to mutually maximize the measurement disagreements. However, existing methods only consider a single measurement, which *overlooks the mutual enhancement between different measurements*.

Assumption 2 In-Example Consistency. *For any example $x \in \mathcal{U}$ and given two different measurements $\mathcal{M}_1(f; x, \mathcal{Y})$ and $\mathcal{M}_2(f; x, \mathcal{Y})$, we assume the two measurements satisfy in-example consistency. Namely, existing a small real value δ that lets all x satisfy $d(x, \mathcal{M}_1, \mathcal{M}_2) \leqslant \delta$.*

Under the in-example consistency assumption, we expect that different measurements are consistent on each example. However, this consistency may not be guaranteed when a misidentified ID or OOD example is sent for optimization, we name them *abnormal examples*. This motivates us to employ cross-measurement disagreements to detect abnormal examples and calibrate the measurements during training. However, existing OSTC methods overlook such availability and *lack reliable mechanism to correct misidentified OOD examples during training*. This motivates us to consider: **How can we exploit the properties of cross-measurement disagreements to promote disagreement maximization and rectify abnormal examples?**

3.3 Adversarial Disagreement Maximization

In light of the formal discussion on measurement disagreements, we present an Adversarial Disagreement Maximization (ADM) model for OSTC that effectively addresses the above questions. In ADM, we first specify two measurements using a cross-entropy loss on the in-distribution classifier and confidence on the outlier detector. An adversarial learning approach is then proposed to synergistically maximize disagreements of the two measurements. To guarantee the optimization quality, we present an abnormal-example detection method to calibrate the disagreement maximization process. The entire structure of our ADM and its implementation process is shown in Figure 1.

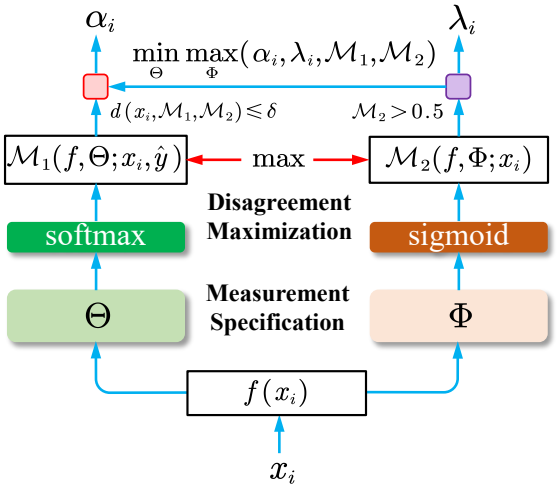


Figure 1: The implementation process of our Adversarial Disagreement Maximization (ADM) model.

Specification of Measurements. To perform OSTC, we build a softmax classifier on $\mathcal{Y}$ to classify ID examples and a sigmoid outlier detector to identify OOD examples. Since OOD examples may cause larger cross-entropy losses of the classifier and lower outlier detection confidence, we naturally treat the cross-entropy loss and the confidence as measurements. Specifically, a pre-trained language model is first used to obtain text representations. For later use, we formulate this text representation process as function f. Then, we can define the measurement of the cross-entropy loss

$$\mathcal{M}_1(f, \Theta; x, \hat{y}) = -\log \frac{\theta_{\hat{y}}^{\mathrm{T}} f(x)}{\sum_{y \in \mathcal{Y}} \theta_y^{\mathrm{T}} f(x)}, \quad (5)$$

where $\Theta = \{\theta_y | y \in \mathcal{Y}\}$ is the parameter in the softmax classifier on $\mathcal{Y}$. The measurement of the outlier-detection confidence is defined as

$$\mathcal{M}_2(f, \Phi; x) = \sigma(\Phi^{\mathrm{T}} f(x)), \quad (6)$$

where Φ is the parameter of the outlier detector and σ is the logistic function. Now the two measurements M_1 and M_2 are specified.

Disagreement Maximization via Adversarial Learning. To synergistically maximize disagreements between the two measurements, we propose an adversarial learning approach to iteratively enlarge the two disagreements. Concretely, we define the following adversarial training process for OSTC

$$\min_{\Theta} \max_{\Phi} \sum_{i=1}^{n} (-1)^{\lambda_i} \mathcal{M}_2(f, \Phi; x_i) \mathcal{M}_1(f, \Theta; x_i, \hat{y}_i) \quad (7)$$

where λ_i is a binary indicator that satisfies

$$\lambda_i = \begin{cases} 1, \mathcal{M}_2(f, \Phi; x_i) > 0.5, \\ 0, otherwise. \end{cases} \quad (8)$$

The above objective adopts a min-max optimization process. At the minimization step, the outlier detection confidence $\mathcal{M}_2$ performs as a weight on the cross-entropy loss $\mathcal{M}_1$ and λ_i performs as a switcher to determine maximize or minimize $\mathcal{M}_2$. Specifically, when $\lambda_i = 1$, x_i is treated as an ID example and the model minimizes its corresponding loss $\mathcal{M}_1(f, \Theta; x_i, \hat{y}_i)$. Otherwise, when $\lambda_i = 0$, x_i is treated as an OOD example and the model minimizes the negative loss $-\mathcal{M}_1(f, \Theta; x_i, \hat{y}_i)$, i.e., the model maximizes $\mathcal{M}_1$. Similarly, At the maximization step, $\mathcal{M}_1$ becomes a weight and the outlier detection confidence $\mathcal{M}_2$ is accordingly maximized or minimized according to λ_i. In this way, the measurements $\mathcal{M}_1$ and $\mathcal{M}_2$ for ID examples and OOD examples are updated in opposite directions, thus maximize the in-measurement disagreements $d_{\mathcal{M}_1}(x_+, x_-)$ and $d_{\mathcal{M}_2}(x_+, x_-)$ and therefore increase the disagreement bounds.

Abnormal Example Detection and Measurement Calibration. According to Equation (8), the binary indicator λ_i rely on the value of the outlier detector $\mathcal{M}_2(f, \Phi; x_i)$. As in the adversarial training process, the outlier detector has no supervision signals, it inevitably produces error indicators. An incorrect indicator λ_i will *reverse the optimization direction* and result in performance degradation. However, no information guides us to distinguish the examples that may lead to incorrect indicators.

Fortunately, the in-example consistency assumption provides us with an opportunity to mitigate the problem. Specifically, when an example is misclassified by the outlier detector, i.e., an abnormal example, it is likely to fail to satisfy in-example consistency. Thus, we use this property to detect abnormal examples and reverse corresponding indicator λ_i to maintain a correct optimization direction. To realize abnormal example detection, we specify the cross-measurement disagreement as

$$d(x_i, \mathcal{M}_1, \mathcal{M}_2) = |\mathcal{M}_2(f, \Phi; x_i) - \frac{\mathcal{M}_1(f, \Theta; x_i, \hat{y}_i)}{\max_j \mathcal{M}_1(f, \Theta; x_j, \hat{y}_j)}| \quad (9)$$

The measurement $\mathcal{M}_1$ is normalized with max-normalization to keep consistent scope $[0, 1]$ with measurement $\mathcal{M}_2$. Under this specification, we modify the adversarial learning objective to

$$\min_{\Theta} \max_{\Phi} \sum_{i=1}^{n} (-1)^{\alpha_i} (-1)^{\lambda_i} \mathcal{M}_2(f, \Phi; x_i) \cdot \mathcal{M}_1(f, \Theta; x_i, \hat{y}_i) \quad (10)$$

where α_i is a binary indicator that indicates whether x_i is an abnormal example and its value is determined by whether x_i satisfies the in-example consistency. Namely, α_i is computed by

$$\alpha_i = \begin{cases} 0, d(x_i, \mathcal{M}_1, \mathcal{M}_2) > \delta, \\ 1, otherwise. \end{cases} \quad (11)$$

Note that as $\mathcal{M}_1$ and $\mathcal{M}_2$ are converse measurements, i.e., larger $\mathcal{M}_1$ and smaller $\mathcal{M}_2$ indicate

OOD examples, they are more consistent when a larger margin between $\mathcal{M}_1$ and $\mathcal{M}_2$ is presented, which is different from the isotropic measurements defined in Assumption 2. Under this design, when an example x_i is detected as an abnormal example, i.e., $\alpha_i = 1$, the model will reverse the optimization direction in the learning objective (10) and calibrate the measurements towards a correct optimization direction, which guarantees the effectiveness of adversarial disagreement maximization.

Optimization Process of ADM. We optimize ADM with the following three training stages:

Pre-stage1: To provide an initialized model, we pre-train the classifier using labeled texts in $\mathcal{L}$ with cross-entropy loss and treat labeled texts as ID examples and unlabeled texts in $\mathcal{U}$ as OOD examples to pre-train the outlier detector with BCE loss.

Pre-stage2: Assign pseudo labels for unlabeled texts using the model trained in pre-stage1 and use the pseudo-labeled texts to further refine the classifier and outlier detector.

ADM: Perform adversarial disagreement maximization to iteratively optimize the two measurements and further update the classifier and outlier detector combined with measurement calibration.

4 Experiment

4.1 Datasets and Experiment Setting

Datasets. We use the 3 benchmark datasets created from existing text classification datasets in the previous work to evaluate OSTC (Chen et al., 2023), including AGNews (Radford et al., 2019), DBPedia (**?**) and Yahoo (Chang et al., 2008). These datasets are all widely used in text classification and includes enough amounts of common classes to evaluate OSTC models.

Table 2: The statistics of the datasets. *# lab.*, *#unl.*, *#val.* and *#test* respectively denote the labeled, unlabeled, validation and test texts for each class. *#ID* , *#OOD* denote the number of ID and OOD classes, respectively.

Dataset	#lab.	#unl.	#val.	#test	#ID	#OOD
AGNews	10/50/100	5k	2k	1.9k	2	2
Yahoo	10/50/100	5k	2k	6k	6	4
DBPedia	10/50/100	5k	2k	5k	8	6

Dataset Construction and Experimental Settings. We follow the same dataset splitting setting in previous work (Chen et al., 2023). We use the same ID and OOD class sets and data for training, validation and test described in the previous work. The statistics of these benchmarks are shown in Table 2. We also follow the two metrics used in LOS to evaluate the OSTC performance. We use *Acc* and *F1* to denote the overall accuracy and F1 value, which jointly evaluates a model's performance in both ID classification and OOD detection.

4.2 Implementation and Baseline Models

Implementation. Our ADM model is implemented with PyTorch. We leverage the BERT encoder to build the Θ component and introduce an MLP with a sigmoid function to build the Φ component. When optimizing, we execute 100 updates for the first two pre-training stages respectively and choose model parameters with the best *Acc* on the validation set. During the ADM training stage, we iteratively execute 100 updates at the minimization step and 100 updates at the maximization step. The labeled training batch size is set to 4 and the unlabeled training batch size is set to 8. The learning rate of the model is set to $5e-5$. The hyper-parameter δ is set to 0.25 on AGNews and Yahoo and is set to 0.15 on DBPedia. All hyper-parameters are selected by grid search on the validation set. We run each experiment 3 times and report the average result and standard deviation. We run all experiments on two NVIDIA Tesla A100 GPUs with 40GB memory.

Baseline Models ADM is compared with the following baselines:

UDA+MSP, MixText+MSP: Pipeline OSTC models combine UDA (Xie et al., 2020), MixText (Chen et al., 2020) with maximum softmax probability to implement OOD detection (Hendrycks and Gimpel, 2017).

UDA+LSoftmax, MixText+LSoftmax: Pipeline OSTC models combine UDA, MixText that learn discriminative text features with an uncomplicated softmax and achieve OOD detection using Local Outlier Factor (Yan et al., 2020).

UDA+DOC, MixText+DOC: Pipeline models combine UDA, MixText with m 1-vs-rest sigmoid classifiers for OOD detection (Shu et al., 2017).

UDA+LMCL, MixText+LMCL: Pipeline OSTC models combine UDA, MixText using large margin cosine loss for OOD detection (Lin and Xu, 2019).

UDA+LOS, MixText+LOS: OSTC models unify semi-supervised training and outlier detection within probabilistic latent variable modeling which optimized with EM algorithm based on UDA and Mixtext (Chen et al., 2023).

Table 3: The open-set semi-supervised text classification results on AGNews, Yahoo and DBPedia.

Method	AGNews						Yahoo						DBPedia					
	$k=10$		$k=50$		$k=100$		$k=10$		$k=50$		$k=100$		$k=10$		$k=50$		$k=100$	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
UDA+MSP	35.16	31.42	50.83	31.90	60.57	56.51	40.97	12.44	47.80	27.87	55.51	44.84	52.06	37.14	81.14	83.31	81.53	84.46
	(0.19)	(0.34)	(1.25)	(0.12)	(1.49)	(2.40)	(0.28)	(0.73)	(0.89)	(1.68)	(0.86)	(1.29)	(1.90)	(2.43)	(1.44)	(0.91)	(0.93)	(0.69)
UDA+LSoftmax	37.49	35.93	38.75	38.80	41.45	39.63	38.95	41.94	44.49	49.62	44.92	51.35	53.96	65.01	55.40	65.62	55.29	63.74
	(0.91)	(0.47)	(0.29)	(0.29)	(1.79)	(0.84)	(3.28)	(1.21)	(0.96)	(0.85)	(0.08)	(0.96)	(1.40)	(3.97)	(0.63)	(2.37)	(0.53)	(0.86)
UDA+DOC	33.26	30.89	49.95	26.39	60.59	55.16	39.93	9.87	40.01	8.21	47.98	28.39	42.84	6.73	48.89	18.79	82.88	82.30
	(0.65)	(1.11)	(0.33)	(0.73)	(2.29)	(2.04)	(0.17)	(0.40)	(0.07)	(0.02)	(0.62)	(1.27)	(0.07)	(0.05)	(1.20)	(2.96)	(1.06)	(1.36)
UDA+LMCL	35.20	31.48	44.66	28.03	48.26	41.98	41.92	37.06	46.62	54.05	47.65	55.26	53.13	59.46	56.16	69.94	56.34	69.79
	(0.66)	(0.50)	(1.61)	(7.69)	(6.82)	(5.77)	(4.28)	(20.49)	(0.59)	(1.10)	(0.22)	(0.49)	(3.38)	(12.82)	(0.27)	(0.47)	(0.09)	(0.80)
MixText+MSP	35.17	32.23	50.39	28.40	63.02	59.21	40.49	10.41	48.26	28.32	54.30	42.63	53.04	38.17	80.96	82.97	80.77	83.83
	(0.89)	(0.40)	(0.49)	(3.58)	(0.41)	(1.35)	(0.37)	(1.72)	(0.68)	(1.35)	(1.13)	(3.12)	(0.65)	(1.70)	(0.57)	(0.36)	(0.83)	(0.29)
MixText+LSoftmax	37.93	37.80	39.53	39.69	40.98	39.41	43.13	48.31	45.62	52.32	45.75	51.50	53.84	65.80	56.64	67.21	57.83	64.93
	(1.15)	(1.58)	(0.53)	(0.59)	(3.16)	(2.61)	(2.12)	(1.24)	(0.62)	(0.66)	(0.58)	(1.12)	(2.32)	(2.92)	(1.71)	(3.36)	(2.03)	(0.98)
MixText+DOC	38.17	36.54	50.10	22.56	58.41	51.58	40.12	9.13	40.03	8.26	47.38	26.61	42.85	7.27	47.65	16.93	81.74	82.55
	(0.79)	(1.89)	(0.04)	(0.13)	(1.27)	(4.02)	(0.02)	(0.07)	(0.44)	(1.00)	(0.79)	(1.89)	(0.02)	(0.58)	(1.76)	(3.50)	(3.61)	(2.23)
MixText+LMCL	34.20	27.41	40.55	38.44	51.27	47.66	43.89	49.35	47.04	53.77	48.42	55.70	55.40	69.79	56.05	70.24	56.24	72.08
	(0.93)	(3.78)	(8.87)	(11.91)	(10.70)	(7.06)	(2.65)	(2.56)	(0.17)	(0.79)	(0.63)	(1.12)	(0.08)	(0.23)	(0.06)	(0.13)	(0.11)	(1.93)
UDA+LOS	56.44	42.45	75.12	72.52	75.53	74.64	46.49	45.88	63.71	65.60	67.50	**67.67**	77.89	80.52	83.28	86.07	85.68	88.18
	(0.84)	(6.24)	(3.48)	(3.10)	(1.72)	(1.37)	(1.00)	(3.57)	(1.65)	(1.01)	(2.38)	(1.67)	(3.99)	(3.55)	(1.57)	(0.65)	(3.24)	(3.41)
MixText+LOS	52.89	31.80	69.11	62.04	76.60	73.33	57.19	51.07	66.57	66.18	**68.09**	66.99	76.67	79.23	**91.92**	**92.91**	88.31	90.33
	(0.57)	(1.54)	(4.90)	(7.82)	(2.81)	(3.83)	(0.35)	(0.41)	(0.09)	(0.54)	(0.49)	(0.94)	(4.48)	(4.23)	(3.88)	(3.05)	(9.57)	(7.39)
ADM	**67.47**	**62.35**	**77.65**	**74.84**	**79.93**	**78.10**	**57.95**	**54.11**	**68.08**	**66.07**	67.67	66.34	**86.49**	**85.52**	91.55	91.33	**90.75**	**90.50**
	(5.21)	(3.76)	(0.92)	(1.29)	(1.09)	(1.22)	(1.26)	(1.71)	(0.20)	(0.59)	(0.59)	(0.80)	(2.87)	(3.36)	(1.60)	(1.37)	(1.23)	(0.40)

Table 4: Ablation studies of ADM on the AGNews dataset.

Metric	ADM pre-stage1		ADM pre-stage2		ADM 0-threshold		ADM	
	$k=10$	$k=50$	$k=10$	$k=50$	$k=10$	$k=50$	$k=10$	$k=50$
Acc	53.97	56.22	54.07	69.14	50.52	68.67	**67.47**	**77.65**
F1	36.79	41.08	40.22	62.30	33.47	67.96	**62.35**	**74.84**
In	52.68	64.42	53.02	69.18	50.00	**84.47**	**71.49**	82.60
Out	58.88	60.51	60.21	76.50	55.78	74.64	**79.70**	**85.32**

4.3 Experiment Results

Main Results. The open-set semi-supervised text classification results on benchmark datasets are shown in Table 3. From the table, we observe that ADM achieves the best performance in all settings of the AGNews dataset and most of the settings on Yahoo and DBPedia datasets and ADM achieves more improvement when k is smaller. These results demonstrate that our adversarial disagreement maximization approach is effective in OSTC and it is more effective in a challenging setting. Another observation is that LOS combined with UDA and MixText significantly improve the performance over pipeline models and ADM further improves LOS in most of the settings. These results manifest that unified training of semi-supervised classification and outlier detection can be better adapted to OSTC than the pipeline approach and directly maximizing measurements disagreement is more effective than existing optimization methods in OSTC.

Ablation Study. To analyze the contributions of each component and each training stage in ADM, we make ablation studies. Specifically, we compare 3 ablated models: ADM pre-stage1, pre-stage1 and ADM 0-threshold that disables the abnormal-example detection module with a setting of $\delta = 0$. The ablated results show that pre-stage1 provides an initialized model for ADM and the pre-stage2 further improves upon pre-stage1. However, when training ADM without abnormal example detection, ADM 0-threshold performs even worse than in pre-stage2. When employing an appropriate abnormal-example detection threshold, ADM significantly improves OSTC results. This result demonstrates that abnormal-example detection and measurement calibration guarantees the effectiveness of ADM.

ID Classification and OOD Detection Results. To analyze the models' ability to classify ID examples and detect OOD examples, we report the ID classification accuracy and OOD detection accu-

Table 5: The evaluation results of accuracy on ID examples (In) and OOD detection accuracy on all examples (Out).

Method	AGNews						Yahoo						DBPedia					
	$k=10$		$k=50$		$k=100$		$k=10$		$k=50$		$k=100$		$k=10$		$k=50$		$k=100$	
	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out
UDA+MSP	70.33	49.99	82.90	51.67	86.10	62.97	63.70	41.54	76.23	48.44	78.46	56.75	96.98	52.17	98.42	81.40	98.55	81.82
UDA+LSoftmax	70.54	44.43	84.10	42.90	86.47	46.37	63.83	56.12	77.66	55.58	79.53	55.94	96.98	55.38	**98.52**	56.06	**98.57**	55.87
UDA+DOC	66.21	49.61	83.5	50.32	86.66	62.62	59.95	40.70	75.91	40.01	78.70	48.58	94.72	47.61	98.29	48.89	98.39	83.01
UDA+LMCL	70.40	50.33	83.55	47.19	86.07	51.10	65.92	52.90	77.41	59.21	79.49	59.67	96.78	54.38	98.45	57.02	98.49	57.20
MixText+MSP	70.35	49.93	83.32	50.81	86.88	65.08	66.82	40.75	77.54	48.91	79.48	55.72	95.79	53.23	98.10	81.30	98.35	81.11
MixText+LSoftmax	73.90	44.29	**85.47**	43.20	87.08	44.57	**70.61**	59.86	78.30	57.28	79.13	56.84	**97.26**	55.13	97.97	57.47	98.32	58.53
MixText+DOC	**76.17**	49.92	83.97	50.11	86.82	60.48	64.58	40.29	76.51	40.03	77.93	47.90	95.75	42.86	97.88	47.65	98.46	81.90
MixText+LMCL	69.35	45.17	83.54	44.71	86.97	55.01	67.86	55.15	**78.50**	59.88	**79.96**	60.33	96.96	57.32	98.12	57.11	98.28	57.22
UDA+LOS	59.18	60.34	84.23	77.78	86.23	78.94	60.63	59.73	76.27	70.66	78.12	73.96	93.49	79.53	96.06	84.40	97.55	86.40
MixText+LOS	61.97	53.59	83.50	70.48	**87.27**	77.97	69.34	61.29	78.03	70.30	79.73	71.77	92.15	79.17	97.75	**92.61**	98.33	88.83
ADM	71.49	**79.70**	82.60	**85.32**	84.62	**87.06**	63.03	**72.86**	74.17	**78.21**	76.82	**77.71**	95.16	**88.20**	97.78	92.54	97.71	**91.90**

racy in Table 5. Metrics *In* and *Out* are introduced to denote the ID accuracy on only ID examples and outlier-detection accuracy on all examples, respectively. The results in the table show that our ADM model outperforms existing OSTC models with large margins on the outlier detection (Out) accuracy results in most of the settings. These results demonstrate that the superiority of ADM is dominated by its good performance on OOD detection although it sacrifices ID classification results in some settings. These results suggest that ADM is effective in alleviating false positive inference.

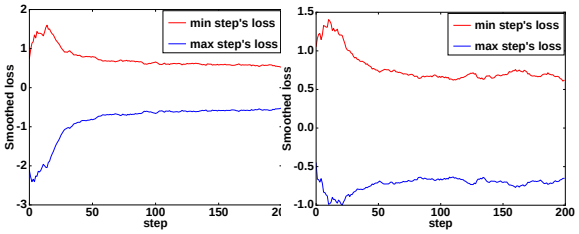
Comparison with Large Language Models. We provide an empirical study using the large language model (LLM) LLaMA2-7B and in-context learning for OSTC in Table 6. Concretely, we design prompts that can guide the LLaMA2-7B model to complete classification and outlier detection, and we evaluate it in the 0-shot and 10-shot settings on the AGNews dataset. From the results, we can conclude that LLaMA2-7B can perform outlier detection and the OSTC task. However, in the 0-shot setting LLaMA2-7B, it performs poorly in both outlier detection and OSTC and achieves worse performance than baseline models. In the 10-shot setting of LLaMA2-7B, it achieves a good outlier detection result but a worse OSTC result than baseline models. And ADM outperforms LLaMA2-7B in both outlier detection and OSTC. These results demonstrate that in the OSTC task, carefully designed lightweight models may still be necessary and useful with the background of LLMs.

4.4 Experimental Analysis

Analysis on the Adversarial Learning Process. To study how our adversarial learning process works, we make an analysis of the changes in the losses during training. Specifically, we record the loss values of the maximization and minimization steps during training on the $k = 10$ and $k = 50$ settings of AGNews in Figure 2. From the figure, we observe that the loss values of the maximization and minimization steps respectively increase and decrease as the training progresses. These results demonstrate that our adversarial training approach is effective. Since the losses of the maximization and minimization steps can be respectively viewed as weighted summed measurements, the results demonstrate that the measurement disagreements are synergistically maximized.

Table 6: Comparison with Large Language Models.

Model	Outlier Detection	OSTC
LLaMA2-7B (0-shot)	49.6	25.4
LLaMA2-7B (10-shot)	71.6	45.3
UDA+LOS ($k = 10$)	60.3	56.4
Mixtext+LOS ($k = 10$)	53.4	52.9
ADM ($k = 10$)	**79.7**	**67.4**



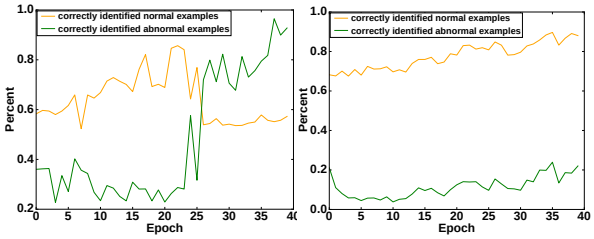
(a) AGNews 10 (b) AGNews 50

Figure 2: The smoothed loss values change trend in min-max optimization during training.

Analysis on the Abnormal-Example Detection Approach. In order to investigate the effective-

Table 7: Case study on four selected text examples from the AGNews dataset.

Id	Text	k	True Label	Pre-stage2	Initial $\mathcal{M}_2$	ADM $\mathcal{M}_2$
1	Jennifer Canada knew she was entering a boy's club when she enolled in Southern Methodist University's Guildhall school of video game making.	10	Sci/Tech	\	0.0706	0.9771
2	AP - The smallest man on the court always seems to make the biggest plays for the Washington Huskies.	10	\	Business	0.8621	0.2369
3	The company said most cuts would come in its network operations division, where work has become increasingly automated, and in the customer service group.	50	Business	\	0.4134	0.9995
4	Myanmar's Opposition National League for Democracy (NLD) party accused the military regime of endangering party leader Aung San Suu Kyi's life by restricting her access to a doctor and non-junta security.	50	\	Sci/Tech	0.9963	0.0174



(a) AGNews 10 (b) AGNews 50

Figure 3: The percent of correctly identified normal and abnormal examples accumulated in previous training epochs in $k = 10$ and $k = 50$ settings of AGNews.

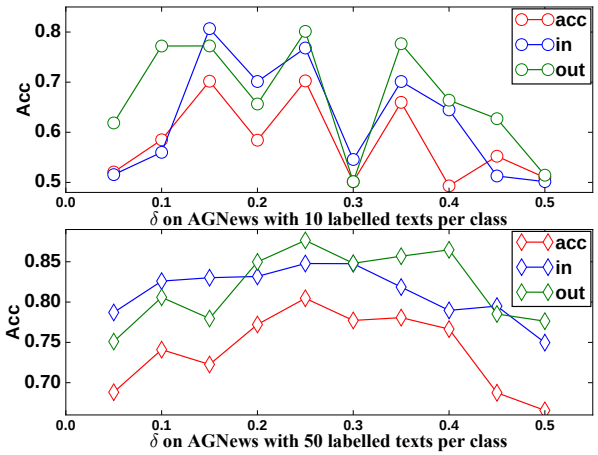


Figure 4: Hyper-parameter analysis of δ on AGNews.

ness of our abnormal-example detection approach during adversarial disagreement maximization, we report the percent of correctly identified normal and abnormal examples accumulated in previous training epochs in Figure 3. From the recording results, we observe that the percentage of correctly identified abnormal examples increase as the training progress. This result indicates that the abnormal-example detection approach is effective. In $k = 10$ setting, although the model sacrifices normal example identification performance, it significantly increase abnormal-example detection performance.

Hyper-Parameter Analysis. To study how the hyper-parameters affect the training process, we make a hyper-parameter analysis. Specifically, we report the OSTC evaluation results of $k = 10$ and $k = 50$ settings on AGNews using different hyper-parameter δ in Figure 4. From the results, we observe that the performance of ADM is sensitive to the hyper-parameter δ and an inappropriate configuration of δ may result in poor performance of ADM. This phenomenon indicates that abnormal-example detection and measurement calibration is necessary to guarantee the effectiveness of ADM.

4.5 Case Study

We make a case study in Table 7 on selected text examples to analyze the outlier detection ability of ADM. As shown in case 1 and case 3, the ground-true labels are respectively *Sci/Tech* and *Business*, but the pre-stage2 model makes wrong predictions ($\mathcal{M}_2 < 0.5$). After ADM training, the outlier detector successfully reverses the wrong predictions to the correct ones. Similarly, in case 2 and case 4, pre-stage2 incorrectly identifies the example as OOD. Nevertheless, ADM successfully rectifies the predictions, demonstrating its effectiveness.

5 Conclusion

We reveal the potential of measurement disagreement in OSTC. To fully employ the advantage of measurement disagreement in OSTC, we propose to directly maximize it and design a novel adversarial disagreement maximization (ADM) approach combined with abnormal-example detection to improve OSTC performance. Experiment results demonstrate the effectiveness of ADM.

6 Limitations

Although our measurement disagreement maximization model is demonstrated effective, it may have two limitations. First, ADM relies on a two-stage pre-training process to obtain an initialized ID classifier and outlier detector, which guarantees the effectiveness of the min-max optimization. Second, ADM requires the proposed abnormal-example detection and measurement calibration approach to guarantee the correct optimization direction.

Acknowledgments

This work was supported by the National Science and Technology Major Project under Grant 2022ZD0120202, in part by the National Natural Science Foundation of China (No. U23B2056 and No. 62306026), in part by China Postdoctoral Science Foundation (No. 2023M740184), in part by the Fundamental Research Funds for the Central Universities, and in part by the State Key Laboratory of Complex & Critical Software Environment.

References

Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *SIGMOD*, pages 93–104. ACM.

Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *AAAI*, volume 2, pages 830–835.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *ACL*, pages 2147–2157.

Junfan Chen, Richong Zhang, Junchi Chen, Chunming Hu, and Yongyi Mao. 2023. Open-set semi-supervised text classification with latent outlier softening. In *KDD*, pages 226–236.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *NIPS*, pages 2672–2680.

Suchin Gururangan, Tam Dang, Dallas Card, and Noah A. Smith. 2019. Variational pretraining for semi-supervised text classification. In *ACL*, pages 5880–5894. Association for Computational Linguistics.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*.

Junkai Huang, Chaowei Fang, Weikai Chen, Zhenhua Chai, Xiaolin Wei, Pengxu Wei, Liang Lin, and Guanbin Li. 2021. Trash to treasure: Harvesting OOD data with cross-modal matching for open-set semi-supervised learning. In *ICCV*, pages 8290–8299.

Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896.

Ju Hyoung Lee, Sang-Ki Ko, and Yo-Sub Han. 2021. Salnet: Semi-supervised few-shot text classification with attention-based lexicon construction. In *AAAI*, pages 13189–13197.

Changchun Li, Ximing Li, and Jihong Ouyang. 2021. Semi-supervised text classification with balanced deep representation distributions. In *ACL*, pages 5044–5053.

Haoran Li, Chun-Mei Feng, Tao Zhou, Yong Xu, and Xiaojun Chang. 2022a. Prompt-driven efficient open-set semi-supervised learning. *CoRR*, abs/2209.14205.

Shujie Li, Min Yang, Chengming Li, and Ruifeng Xu. 2022b. Dual pseudo supervision for semi-supervised text classification with a reliable teacher. In *SIGIR*, pages 2513–2518.

Ting-En Lin and Hua Xu. 2019. Deep unknown intent detection with margin loss. In *ACL*, pages 5491–5496.

Chen Liu, Mengchao Zhang, Zhibing Fu, Panpan Hou, and Yu Li. 2021. Flitext: A faster and lighter semi-supervised text classification with convolution networks. In *EMNLP*, pages 2481–2491.

Yen-Cheng Liu, Chih-Yao Ma, Xiaoliang Dai, Junjiao Tian, Peter Vajda, Zijian He, and Zsolt Kira. 2022. Open-set semi-supervised object detection. In *ECCV*, pages 143–159.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-supervised neural text classification. In *CIKM*, pages 983–992.

Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *ICLR*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Kuniaki Saito, Donghyun Kim, and Kate Saenko. 2021. Openmatch: Open-set consistency regularization for semi-supervised learning with outliers. In *NeurIPS*.

Lei Shu, Hu Xu, and Bing Liu. 2017. DOC: deep open classification of text documents. In *EMNLP*, pages 2911–2916.

Antti Tarvainen and Harri Valpola. 2017. Weight-averaged consistency targets improve semi-supervised deep learning results. *CoRR*, abs/1703.01780.

Austin Cheng-Yun Tsai, Sheng-Ya Lin, and Li-Chen Fu. 2022. Contrast-enhanced semi-supervised text classification with few labels. In *AAAI*, pages 11394–11402.

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

Hai-Ming Xu, Lingqiao Liu, and Ehsan Abbasnejad. 2022. Progressive class semantic matching for semi-supervised text classification. In *NAACL*, pages 3003–3013.

Guangfeng Yan, Lu Fan, Qimai Li, Han Liu, Xiaotong Zhang, Xiao-Ming Wu, and Albert Y. S. Lam. 2020. Unknown intent detection using gaussian mixture model with an application to zero-shot intent classification. In *ACL*, pages 1050–1060.

Weiyi Yang, Richong Zhang, Junfan Chen, Lihong Wang, and Jaein Kim. 2023. Prototype-guided pseudo labeling for semi-supervised text classification. In *ACL*, pages 16369–16382.

Qing Yu, Daiki Ikami, Go Irie, and Kiyoharu Aizawa. 2020. Multi-task curriculum framework for open-set semi-supervised learning. In *ECCV*, volume 12357, pages 438–454.

Ronghang Zhu and Sheng Li. 2022. Crossmatch: Cross-classifier consistency regularization for open-set single domain generalization. In *ICLR*.