Alleviate Dataset Shift Problem in Fine-grained Entity Typing with Virtual Adversarial Training

Haochen Shi[1] , **Siliang Tang**[1*] , **Xiaotao Gu**[2] , **Bo Chen**[1] , **Zhigang Chen**[3] , **Jian Shao**[1] and **Xiang Ren**[4]

[1]Zhejiang University
[2]University of Illinois at Urbana-Champaign
[3] State Key Laboratory of Cognitive Intelligence, Hefei, China
[4]University of Southern California
{hcshi, siliang, chenbo123, jshao}@zju.edu.cn, xiaotao2@illinois.edu, zgchen@iflytek.com, xiangren@usc.edu

Abstract

The recent success of Distant Supervision (DS) brings abundant labeled data for the task of fine-grained entity typing (FET) without human annotation. However, the heuristically generated labels inevitably bring a significant distribution gap, namely *dataset shift*, between the distantly labeled training set and the manually curated test set. Considerable efforts have been made to alleviate this problem from the *label perspective* by either intelligently denoising the training labels, or designing noise-aware loss functions. Despite their progress, the *dataset shift* can hardly be eliminated completely. In this work, complementary to the label perspective, we reconsider this problem from the *model perspective*: Can we learn a more robust typing model with the existence of dataset shift? To this end, we propose a novel regularization module based on virtual adversarial training (VAT). The proposed approach first uses a self-paced sample selection function to select suitable samples for VAT, then constructs virtual adversarial perturbations based on the selected samples, and finally regularizes the model to be robust to such perturbations. Experiments on two benchmarks demonstrate the effectiveness of the proposed method, with an average 3.8%, 2.5% and 3.2% improvement in accuracy, Macro F1 and Micro F1 respectively compared to the next best method.

1 Introduction

Fine-grained entity typing (FET) aims at assigning types to mentions of entities based on thier conext. It is an important task as the type information provided by it is useful for many downstream tasks, such as entity linking and event extraction [Yang *et al.*, 2019; Ding *et al.*, 2015]. State-of-the-art FET systems usually utilize distant supervision (DS) to fetch abundant training data by first linking a target mention to an existing entity in a knowledge base (KB) and then assigning all possible types of the entity to the target mention [Onoe and Durrett, 2019; Chen *et al.*, 2019]. Despite of the convenience in reducing human annotation, DS inevitably introduces noises to the heuristically generated labels. For example, the mention *Shelton* in the sentence "The telephone number for the charity in *Shelton*, Conn., has been disconnected" is assigned with types {*person, state province*} by DS, while only {*state province*} is the correct label for manual labeling. Such inconsistency brings a significant distribution gap, namely *dataset shift* [Moreno-Torres *et al.*, 2012], between the distantly labeled training set and the manually curated test set. Figure 1 is a hypothetical demonstration of the dataset shift problem, where Δs denotes the dataset shift. As shown in Figure 1, due to the existence of dataset shift, even though we achieve a local minimum of loss function on training set, the loss on test set can still be large.

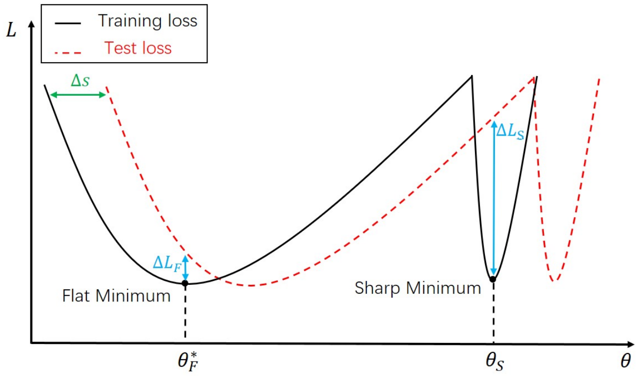

Figure 1: An hypothetical demonstration of dataset shift. The X-axis indicates the parameters of the model and Y-axis indicates the value of loss function. Δs denotes the dataset shift between training set and test set. θ_F^* and θ_S are the parameters where training loss achieves flat minumum and sharp minimum respectively. ΔL is the loss gap between training set and test set.

With aware of the dataset shift problem, prior arts attempt to alleviate this problem from the *label perspective* can be briefly summaried into two categories: (1) the first kind of

[*] Corresponding Author

"denoising" models aim to directly model the label noise and denoise the training labels. For instance, [Gillick *et al.*, 2014] uses heuristic rules to filter out noisy labels, [Ren *et al.*, 2016b] uses the prior knowledge of KB and the partial loss based label embedding to remove noisy labels, [Onoe and Durrett, 2019] relabels and filters out noisy data with a neural network. (2) the other kind of methods try to infer the correct labels dynamically and use the inferred labels to direct the model during the training phase. For example, [Xu and Barbosa, 2018] proposes a variant of partial label loss to handle noisy labels and [Chen *et al.*, 2019] proposes to infer the correct labels with a graph-based algorithm. Actually, all the above denoising methods are dedicated to reduce the gap of joint distribution $P(y, x)$ between the training set and test set, that is, the dataset shift Δs in Figure 1. However, since it is impossible to completely eliminate the datataset shift in distantly-supervised FET, the test loss can still be large even we achieve the minimum loss on the training set. Specifically, the denoising methods try to get low test loss by solving $argmin_\theta\, f(\Delta s|\theta) * f(L_{train}|\theta)$. As is shown in Figure 1, even though the model achieves the minimum loss on the training set at θ_s and the dataset shift is reduced to a small value Δs, the test loss can still be large.

In this work, complementary to the label perspective, we reconsider this problem from the *model perspective*: Can we learn a more robust typing model with the existence of dataset shift? As is shown in Figure 1, when dataset shift $\Delta s \neq 0$ is fixed, test loss not only depends on training loss $f(L_{train}|\theta)$ but also depends on the gap ΔL between test loss and training loss. Therefore, we propose to learn a robust typing model by solving $argmin_\theta\, f(\Delta L|\Delta s, \theta) * f(\Delta s|\theta) * f(L_{train}|\theta)$, where $f(\Delta L|\Delta s, \theta)$ is a measure of the generalization ability and robustness to dataset shift of models. As is shown in Figure 1 and pointed out in [Keskar *et al.*, 2016], the vulnerability of the models to dataset shift (*i.e.*, large $f(\Delta L|\Delta s, \theta)$) is due to the fact that models tend to converge to sharp minimizers. Based on this observation, we propose to improve the robustness of typing models to dataset shift with a variant of VAT. Specifically, we first use a self-paced sample selection function to select suitable samples for VAT, then construct virtual adversarial perturbations masked by the noisy labels for selected smaples, finally regularize the model to be robust to such perturbations. Since the construct perturbations can also be seen as a series of dataset shifts, the model is regularized to be robust to dataset shifts and the generalization ability of the model is finally improved. Since this process is orthogonal to denoising process, both our model and denoising models can serve as plug-and-play modules to enhance existing neural typing systems in a complementary way. Experiments on two benchmarks shows the effectiveness of the proposed method, with average 3.8%, 2.5% and 3.2% improvements in accuracy, macro F1 and Micro F1 respectively compared to next best method.

The major contributions of this paper are summarized as follows:

1. This work provides a new perspective on the problems caused by distant supervision for fine-grained entity typing and explores a new way to improve FET systems by regularizing the model to be robust to dataset shift.
2. A novel regularization module for FET is proposed, where a variant of VAT and a sample selection function is proposed to control the robustness of typing model.
3. Extensive experiments on standard benchmarks with two different base models demonstrate that our method brings stable and significant improvement over the base models. Finally, the proposed method consistently outperforms several state-of-the-art (SOTA) FET systems by a significant margin.

2 Related Work

Fine-grained entity typing was first proposed by [Ling and Weld, 2012] who used distant supervision to induce a relatively large training corpus for FET. Due to the context-agnostic nature of the distant supervision, the training data labeled in this way is inevitably noisy by assigning mentions with the types that cannot be inferred from the context of the mentions. Some works ignore such noise: [Yogatama *et al.*, 2015] proposed to jointly learn feature and type representations utilizing embedding techniques and [Lin and Ji, 2019] proposed to use a hybrid type classifier to capture latent type interdependency.

However, the major challenge of FET remains the problems posed by the noisy labels of the training corpus. With respect to the label noise, [Gillick *et al.*, 2014] proposed context dependent FET and cleaned the noisy labels with heuristics, which suffers from losing training data. To mine the information in noisy labels, [Ren *et al.*, 2016a] proposed partial label loss (**PLL**) to distinguish the impact of clean data and noisy data. [Xu and Barbosa, 2018] proposed a variant of PLL. From data perspective, [Ren *et al.*, 2016b] proposed to reduce label noise using partial-label embedding, which brings a significant improvement to FET and can be viewed as a milestone for FET. More recently, [Onoe and Durrett, 2019] further proposed to reduce the label noise of the Ontonotes dataset augmented by [Choi *et al.*, 2018] via relabeling and filtering out noisy data, which is the SOTA FET system on the Ontonotes dataset before this work. From data consistency perspective, [Chen *et al.*, 2019] proposed Compact Latent Space Clustering (**CLSC**) method to regularize the representation of mentions with the same types to form compact clusters, which achieves a great success in FET. Instead of just trying to reduce the noise in the dataset, this work is devoted to train a robust model with the noisy dataset and denoise the training data jointly. In this work, we propose a regularization based framework to enhance the robustness of the model to dataset shift utilizing a variant of VAT.

3 Task Formulation

The task of FET is to uncover the entity type information for entity mentions (*i.e.*, a sequence of tokens representing the appearance of an entity) in natural language sentences [Ren *et al.*, 2016a]. The task takes a corpus D labeled with a predefined entity type hierarchy Y as input and predicts the most suitable type-path in Y for each entity mention from the test set D_t based on the mention's context.

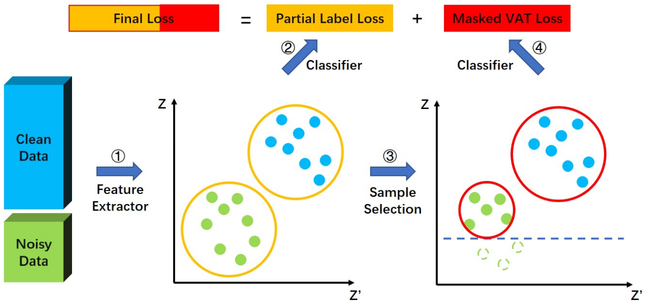

Figure 2: The overall framework of our method.

Because manually labeling a large training set for FET is too expensive and error-prone, current entity typing systems annotate the training corpus utilizing DS. Specifically, the entity typing systems first detect mentions m_i and link them to one or more entity e_i in a KB, and then assign the entity types Y_i of e_i in KB to m_i as candidate types. Formally, a labeled training corpus can be represented as a set of triples $D = \{(m_i, c_i, Y_i)\}$, where m_i is the i-th detected mention, c_i is the context of m_i and Y_i is the set of candidate types of m_i. We denote the all terminal types for each type path in Y_i as target type set Y_i^t (*e.g.* for $Y_i = \{person, doctor, artist\}$, $Y_i^t = \{doctor, artist\}$). This setting is also adopted by [Xu and Barbosa, 2018; Chen *et al.*, 2019].

Note that candidate types in Y_i can form multiple type paths, which may cause *out-of-context* noise (*i.e.* Y_i contains type paths that are irrelevant to the mention m_i in context c_i). We treat samples labeled with single type path (*i.e.* triples (m_i, c_i, Y_i) in D whose corresponding $|Y_i^t| = 1$) as **clean data** and others as **noisy data**. The major challenge of DS entity typing systems is to utilize both the clean data and the noisy data to produce a high-performance typing classifier.

4 Methodology

4.1 Overview

Our method is mainly based on the following assumption: the model should predict smoothly around the mention points in the feature space because mention points close to each other in feature space have similar context. For this assumption, we apply masked VAT to construct gradient-based local perturbation in the feature space for mentions and regularize the classifier to be robust to such kind of perturbation, so that the model's prediction is smoothed.

As demonstrated in Figure 2, our model mainly consists of three parts: (1) a feature extractor to project mentions with their contexts to the feature space, (2) a self-paced sample selection function to select some noisy samples for VAT, (3) and a classifier regularized by the VAT loss. All the clean data is used for adversarial training, but some noisy data will be select for VAT. Specifically, for the i-th sample (e_i, c_i, Y_i^t) in the training set, the mention with its context (m_i, c_i) is transformed into a vector z_i in the feature space using a feature extractor $F(z|(m_i, c_i), \theta_f)$ parameterized by θ_f. Then the posterior $p(y \mid z_i,\ \theta_C)$ will be given by a classifier C parameterized by θ_C. After that a self-paced filter function Γ is applied to decide whether the sample will be used for VAT.

4.2 Feature Extractors and Classifier

We have explored two different feature extractors, namely NFETC [Xu and Barbosa, 2018] and BERT [Devlin *et al.*, 2019].

NFETC. For fair comparison, We first adopt NFETC as the feature extractor, which is the same feature extractor as in [Xu and Barbosa, 2018] and [Chen *et al.*, 2019]. In shot, NFETC can be summarized as follows: (1) the word embedding of NFETC is the concatenation of Glove [Pennington *et al.*, 2014] embedding and word position embedding; (2) the mention representation u_{m_i} of NFETC is the concatenation of the average of the embedding sequence of m_i and the last hidden state of a LSTM over m_i; (3) the context representation u_{c_i} is the result of a word-level attention over the hidden states of a bidirectional LSTM on the context; (4) the final representation of the mention with the context is $z_i = FN([u_{m_i}, u_{c_i}])$, where FN is a feedforward neural network of n layers.

BERT. To further explore the potential of our method, we have also performed experiments using the popular pretrained language model BERT as a new baseline for fine-grained entity typing. Given a mention $m_i = (w_l, ..., w_r)$ with its context $c_i = (w_1, ..., w_L)$, we simply feed the sequence $([CLS], c_i, [SEP], m_i, [SEP])$ to BERT encoder and use the output of $[CLS]$ token as the representation of the mention with its context z_i.

Classifier. With the representation z_i of a mention with its context, we employ a softmax classifier parameterized by $\theta_C = [W_C, b_C]$ to get the posterior: $P(y|z_i) = softmax(W_c z_i + b_C)$, where $W_C \in \mathbb{R}^{K \times d_z}$ can be treated as the type embeddings, $b_c \in \mathbb{R}^{d_z}$ is the type bias, where K is the number of types. The predicted type $\hat{y}$ is the type with maximum posterior probability: $\hat{y} = \arg\max_y P(y|z_i)$.

4.3 Masked Virtual Adversarial Training for FET

The basic idea of VAT is to find the adversarial perturbation r_{adv} (*i.e.* the perturbation on the input that lead to the most different posterior from the original posterior) and regularize the model to be robust to such perturbation, so that the prediction of the model is smoothed. In this work, we propose to improve distantly-supervised FET which is a partially labeled problem with a masked VAT. The process of applying masked VAT to FET can be decomposed into the following three parts: (1) sample selection; (2) masked adversarial perturbation generation; (3) local distributional smoothness (LDS) regularization. The details of the process are introduced in following sections.

Sample Selection

In the early stage of training, the model is prone to give wrong prediction for a mention, in which case the use of VAT will make the classifier further fit the wrong result. To alleviate this problem, we design a self-paced sample selection function to select some samples for VAT, on which the model is not confident in its predictions. Specifically, given a sample (m_i, c_i, Y_i^t), we first find all the types Y_i in the type path terminating at Y_i^t, then convert Y_i to an type index vector $\tilde{Y_i} \in \mathbb{R}^K$, where $\tilde{Y}_{ij} = 1$ if the j-th type is in Y_i, otherwise $\tilde{Y}_{ij} = 0$. After that, we could get the posterior $P(y|(m_i, c_i))$ using the feature extractor and the classifier. With the type index vector $\tilde{Y_i}$ and the posterior $P(y|(m_i, c_i))$, we define the self-paced filter function Γ as following:

$$\Gamma(P(y|(m_i, c_i), \tilde{Y_i} = \mathbf{1}(\arg\max_{y \in \tilde{Y_i}} P(y|(m_i, c_i)) > \tau_i) \quad (1)$$

where $\mathbf{1}$ is an indicator function, $\tau_i = sigmoid(\beta(|\tilde{Y_i}| - \eta))$ is a real value indicating how noisy the i-th sample is, β and η are two hyper-parameters. Only when a sample's $\Gamma(P(y|(m_i, c_i), \tilde{Y_i}) = 1$, the sample will be used for VAT.

Perturbation Generation and Regularization

Adversarial training. Since our method is closely related to adversarial training (AT), we first introduce the process of AT. Given the representation z_i of a mention m_i with its context c_i, and a classifier parameterized by θ_C, AT adds the following regularization term to the objective function:

$$L_{AT} = D(q(y|z_i), p(y|z_i + r_{adv}, \theta_C)) \quad (2)$$

where $r_{adv} = \arg\max_{r, ||r|| <= \epsilon} D(q(y|z_i), p(y|z_i + r, \theta_C))$, $q(y|z_i)$ is the ground truth, r_{adv} is the perturbation on the representation z_i, ϵ is the maximum perturbation step size which is a hyper-parameter, $D(q, p)$ is a non-negative function measuring the divergence between two distributions q and p, and in practice we use KL divergence. Since exact minimization with respect to r is intractable for many neural networks, [Goodfellow *et al.*, 2014] proposed to approximate r_{adv} using the Fast Gradient Signed Method (FGSM):

$$r_{adv} \approx \epsilon sign(\nabla_{z_i} D(q(y|z_i), p(y|z_i, \theta_C))) \quad (3)$$

where $sign()$ is the sign function.

Masked virtual adversarial training. Since some type information is unreachable in semi-supervised learning, [Miyato *et al.*, 2018] proposed using the posterior $p(y|z_i; \theta_C)$ given by classifier as an estimation of the ground truth $q(y|z_i)$ and defined the local distributional smoothness (LDS) to be a measure of the local smoothness of the current classifier at each data point z_i:

$$LDS(z_i, \theta_C) = D(p(y|z_i; \hat{\theta}_C), p(y|z_i + r_{v-adv}; \theta)) \quad (4)$$

where $r_{v-adv} = \arg\max_{r, ||r|| <= \epsilon} D(p(y|z_i; \hat{\theta}_C), p(y|z_i + r; \hat{\theta}))$, $\hat{\theta}_C$ denotes the current parameters of the classifier. For the task of FET which is a partially labeled problem, we propose to use masked VAT to further utilize the noisy labels. Given the representation z_i of a sample (m_i, c_i, Y_i^t), we first compute the type index vector $\tilde{Y_i}$, then apply the modified virtual adversarial loss:

$$L_{vat,i} = D(\tilde{p}(y|z_i; \hat{\theta}_C), \tilde{p}(y|z_i + r_{v-adv}; \theta)) \quad (5)$$

where $\tilde{p}(y|*; \hat{\theta}_C) = softmax(\log(\tilde{Y_i} \otimes (\hat{W}_C * + \hat{b}_C)))$, $r_{v-adv} = \arg\max_{r, ||r|| <= \epsilon} D(\tilde{p}(y|z_i; \hat{\theta}_C), \tilde{p}(y|z_i + r; \hat{\theta}))$, $\tilde{p}(y|z_i; \hat{\theta}_C)$ is the posterior masked by all candidate types Y_i, and $\otimes$ denotes element-wise product. The intuition of masked VAT is to prevent the model from strengthening totally wrong predictions and to encourage the model to distinguish between the noisy candidate types.

Dataset	Ontonotes	BBN
#types	89	47
Type hierarchy depth	3	2
#mentions-train	253241	86078
#mentions-test	8963	12845
%clean mentions-train	73.13	75.92
%clean mentions-test	94.00	100

Table 1: Dataset statistics.

Methods	Strict Acc.	Macro F1	Micro F1
BERT$_{Clean}$	66.1	82.1	76.4
BERT$_{Full}$	66.7	82.7	77.1
BERT$_{SS}$	67.3	83.2	77.3
BERT-VAT$_{Clean}$	70.1	84.1	78.3
BERT-VAT$_{Full}$	69.8	84.6	78.8
BERT-VAT$_{SS}$	69.9	84.9	79.2

Table 2: Ablation study of individual components on the Ontonotes test set. The subscripts $_{Clean}$, $_{Full}$ and $_{SS}$ indicate the model is trained with clean data, full training data and sample selection function respectively.

Solving virtual adversarial perturbations. The evaluation of r_{v-adv} cannot be performed using Eq.(3) because the gradient of $D(\tilde{p}(y|z_i; \hat{\theta}_C), \tilde{p}(y|z_i + r; \hat{\theta}))$ with respect to r is always 0 at $r = 0$. To solve this problem, [Miyato *et al.*, 2018] proposed to approximate r_{v-adv} using the second order Taylor expansion of D and solve the r_{v-adv} via the power iteration method. Specifically, we can approximate r_{v-adv} by repeatedly applying the following update n_t times (n_t is a hyper-parameter):

$$r_{v-adv} \leftarrow \epsilon \overline{\nabla_r D(\tilde{p}(y|z_i; \hat{\theta}_C), \tilde{p}(y|z_i + r; \hat{\theta}))} \quad (6)$$

where the sign $\overline{v}$ means the unit vector of v.

4.4 Overall Objective

The final loss function consists of two parts: the supervision loss L_{sup} and the VAT loss L_{vat}. Given a batch of samples $\{(m_i, c_i, Y_i^t)\}_{i=1}^B$, we use the modified version of partial label loss for softmax classifier to calculate L_{sup}: $L_{sup} = \frac{1}{B} \sum_{i=1}^B \sum_{k=1}^K y_{ik} log(P(y_i|z_i; \theta_C))_k$, where K is the number of target types and B is the batch size. Besides, the corresponding VAT loss is $L_{vat} = \frac{1}{B} \sum_{i=1}^B \Gamma(P(y|(m_i, c_i), \tilde{Y_i}) * L_{vat,i}$. So the final loss function is $L_{final} = L_{sup} + \lambda_{vat} \times L_{vat}$, where λ_{vat} is a hyper-parameter controlling the influence of VAT.

5 Experiments

5.1 Dataset

We evaluate out method on two benchmarks: Ontonotes and BBN. Statistics of the datasets are shown in Table 1.

Ontonotes. The Ontonotes dataset is derived from the Newswire part of Ontonotes corpus and annotated by [Gillick *et al.*, 2014]. The training set of Ontonotes is annotated utilizing DBpedia spotlight, while the test set is manually labeled.

Methods	Ontonotes			BBN		
	Strict Acc.	**Macro F1**	**Micro F1**	**Strict Acc.**	**Macro F1**	**Micro F1**
NFGEC+LME [Xin *et al.*, 2018]	52.9	72.4	65.2	-	-	-
UFET [Choi *et al.*, 2018]	61.6	77.3	71.8	-	-	-
LABELGCN [Xiong *et al.*, 2019]	61.6	77.3	71.8	-	-	-
NFETC-CLSC [Chen *et al.*, 2019]	62.8	77.8	72	74.7	80.7	80.5
FET [Lin and Ji, 2019]	63.8	82.9	77.3	55.9	79.3	78.1
NFETC [Xu and Barbosa, 2018]	60.2	76.4	70.2	73.9	78.8	79.4
NFETC-VAT	63.8	78.7	73	**76.7**	80.7	80.9
D-NFETC-VAT	66.5	83.4	77.5	-	-	-
DenoiseET [Onoe and Durrett, 2019]	64.9	84.5	79.2	-	-	-
BERT-VAT	**69.9**	84.9	79.2	75.9	**83.0**	**83.8**
D-BERT-VAT	68.9	**86.4**	**81.2**	-	-	-

Table 3: Performance comparison of FET systems. The prefix "D" indicates the model is trained on the augumented Ontonotes dataset offered by **DenoiseET**.

BBN. The BBN dataset is derived from 2,311 Wall Street Journal articles. We use the version processed by [Ren *et al.*, 2016a].

In this work, we use the preprocessed dataset provided by [Chen *et al.*, 2019; Onoe and Durrett, 2019].

5.2 Compared Methods

We compare the proposed method with several SOTA FET systems: **NFGEC+LME** [Xin *et al.*, 2018] measures the compatibility between context sentences and labels utilizing a language model; **NFETC** [Xu and Barbosa, 2018] models type hierarchy with the hierarchical loss function; **NFETC-CLSC** [Chen *et al.*, 2019] compresses the clusters in the latent space with a manifold regularization loss; **UFET** [Choi *et al.*, 2018] augments the Ontonotes training set with new sources of distant supervision; **LABELGCN** [Xiong *et al.*, 2019] introduces a label-relational inductive bias with a graph propagation layer; **FET** [Lin and Ji, 2019] models type interdependency with a hybrid type classifier; **DenoiseET** [Onoe and Durrett, 2019] augments the Ontonotes training set and reduces the label noise with a filter and a relabel function; **NFETC-VAT** and **BERT-VAT** are the proposed models using VAT to regularize the classifiers. The methods with prefix "D" indicates the model is trained on Ontonotes dataset denoised by **DenoiseET**, which can be seen as the combination of our method with **DenoiseET**.

5.3 Evaluation Settings

For evaluation metrics, we evaluate the performance by strict accuracy, loose macro F-score and loose micro F-score, which is the most widely used evaluation setting for FET systems [Ling and Weld, 2012].

For BERT feature extractor, we use the pretrained BERT-Base, cased model with a step size of 2e-5 and batch size 32. For NFETC feature extractor, we follow the setting of [Xu and Barbosa, 2018; Chen *et al.*, 2019] using the 300 dimensional pretrained GloVe [Pennington *et al.*, 2014] word vectors.

5.4 Performance Comparison and Analysis

Table 3 shows the performance comparison between the two VAT improved base models and several SOTA FET systems. Note that the proposed method can be further improved when combining with **DenoiseET** by training the model on the augmented and denoised Ontonotes dataset. On both benchmarks, the proposed method consistently outperforms other methods by a significant margin on all three metrics.

To evaluate the influence of individual components of our method, an ablation study is conducted as shown in Table 2. To further evaluate the proposed method, we conduct another ablation study, where several variants of our method are evaluated, and the result is shown in Table 4. Our analysis of the results in Table 4 is as following.

What if we use adversarial training. By comparing the base models with AT improved models, we can see that AT improves the performance of FET systems. Although AT only regularize model to be robust to perturbations around clean data, it can still improve the generalization ability of models. Besides, with the observation that VAT improved models consistently outperform AT improved models, we cloud draw a conclusion that masked VAT makes full use of noisy data to improve FET systems.

Why not generate perturbation in input space. We performed the experiment of generating perturbations on word embedding (**NFETC-VAT***), and there is an improvement over the base model **NFETC**. But performing VAT in such way is inefficient, because it requires much more computational resources than constructing perturbations in the feature space. Besides, the performance gap between **NFETC-VAT*** and **NFETC-VAT** maybe because of the accumulation of a series of perturbations on word embedding may cause a large drift in semantics of the sentences.

Can masked VAT be effectively combined with different denoising methods. We combined masked VAT with two different denoising methods, namely **NFETC-CLSC** and **DenoiseET**. Experiment results show that the combined methods are superior to the base methods to a great extent. This is because mask VAT and denoising focus on handling dataset

Methods	Ontonotes			BBN		
	Strict Acc.	**Macro F1**	**Micro F1**	**Strict Acc.**	**Macro F1**	**Micro F1**
NFETC [Xu and Barbosa, 2018]	60.2	76.4	70.2	73.9	78.8	79.4
NFETC-AT	62.4(+2.2)	77.9(+1.5)	72.1(+1.9)	74.8(+0.9)	79.4(+0.6)	79.7(+0.3)
NFETC-VAT*	62.7(+2.5)	77.9(+1.5)	72(+1.8)	76.2(+2.3)	80.4(+1.6)	80.6(+1.2)
NFETC-VAT	63.8(+3.6)	78.7(+2.3)	73(+2.8)	76.7(+2.8)	80.7(+1.9)	80.9(+1.5)
D-NFETC-VAT	**66.5**(+6.3)	**83.4**(+7.0)	**77.5**(+7.3)	-	-	-
NFETC-CLSC [Chen *et al.*, 2019]	62.8(+2.6)	77.8(+1.4)	72(+1.8)	74.7(+0.8)	80.7(+1.9)	80.5(+1.1)
NFETC-CLSC-VAT	63.9(+3.7)	78.6(+2.2)	73.1(+2.9)	**76.9**(+3.0)	**81.2**(+2.4)	**81.4**(+2.0)
BERT	66.7	82.7	77.1	75.5	80.8	81.5
BERT-AT	68.1(+1.4)	83.7(+1.0)	77.4(+0.3)	74.8(-0.7)	82.6(+1.8)	82.9(+1.4)
BERT-VAT	**69.9**(+3.2)	84.9(+2.2)	79.2(+2.1)	**75.9**(+0.4)	**83.0**(+2.2)	**83.8**(2.3)
D-BERT-VAT	68.9(+2.2)	**86.4**(+3.7)	**81.2**(+4.1)	-	-	-

Table 4: Ablation study of our method with two base models. The prefix "D" indicate the model is trained on Ontonotes dataset augmented and denoised by **DenoiseET**. The suffix "AT" denotes adversarial training. The suffix * indicates the virtual adversarial perturbation is applied on word embedding rather than in feature space.

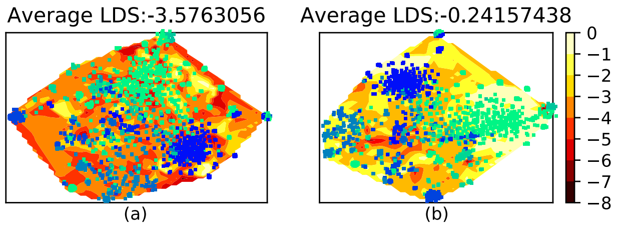

Figure 3: T-SNE visualization of the mention embedding with their LDS generated by NFETC(left) and NFETC-VAT(right) on the BBN test set.

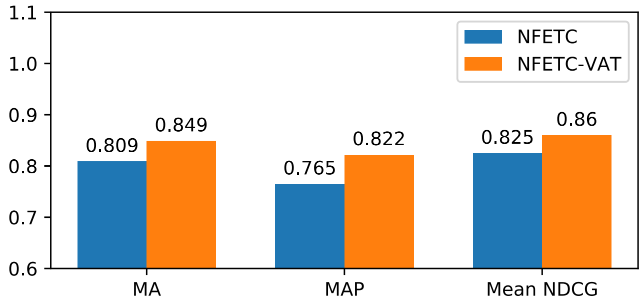

Figure 4: Quality of 20-top nearest neighbors for mentions on BBN test set.

shift problem from different perspectives, namely the robustness to dataset shift and dataset shift reduction.

5.5 Case Study and Visualization

Does masked VAT really regularize the model to predict smoothly. As illustrated in Figure 3, the model's average LDS on the sample points of test set has been significantly improved, although we do not touch the samples of the test set during the training phase. It demonstrates that the prediction of our model is effectively smoothed, and the final model is robust to the dataset shifts constructed by virtual adversarial perturbation. Further more, the robustness of the model to dataset shifts indicates that the parameters of our model have converged to the flat minimum θ^* in Figure 1.

Does masked VAT have effect on mention representation. We measure the quality of the mention representation by the quality of the k-NN (k=20) ranking of mentions. We use three commonly used metrics to evalute the quality of the model's k-nearest neighbors, namely mean accuracy (MA), mean average precision (MAP) and mean Normalized Discounted Cumulative Gain (mean NDCG). As shown in Figure 4, NFETC-VAT consistently outperforms NFETC by a significant margin. Masked VAT constrains that the type of a mention cannot be inverted via a small perturbation on representation, so the mentions with similar initial representations but different types become separated.

6 Conclusion

In this work, we propose to alleviate dataset shift problem in FET by combining the proposed masked VAT with denoising methods. Experiment results demonstrate the proposed method consistently outperforms SOTA models by a significant margin. The proposed method is general and can be used in other domains. As a part of future work, we plan to explore the proposed method on other tasks with sever dataset shift problem, such as relation extraction.

Acknowledgements

This work has been supported in part by National Key Research and Development Program of China (2018AAA010010), NSFC (U1611461, 61572425, 61751209), Key Technologies and Systems of Humanoid Intelligence Based on Big Data (Phase II), Zhejiang University iFLYTEK Joint Research Center, University-Tongdun Technology Joint Laboratory of Artificial Intelligence, Chinese Knowledge Center of Engineering Science and Technology (CKCEST), Engineering Research Center of Digital Library, Ministry of Education, the Fundamental Research Funds for the Central Universities. Xiang Ren's research was funded by the Defense Advanced Research Projects Agency with award W911NF-19-20271.

References

[Chen *et al.*, 2019] Bo Chen, Xiaotao Gu, Yufeng Hu, Siliang Tang, Guoping Hu, Yueting Zhuang, and Xiang Ren. Improving distantly-supervised entity typing with compact latent space clustering. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pages 2862–2872, 2019.

[Choi *et al.*, 2018] Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 87–96, 2018.

[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 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pages 4171–4186, 2019.

[Ding *et al.*, 2015] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Deep learning for event-driven stock prediction. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[Gillick *et al.*, 2014] Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. Context-dependent fine-grained entity type tagging. *arXiv preprint arXiv:1412.1820*, 2014.

[Goodfellow *et al.*, 2014] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[Keskar *et al.*, 2016] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[Lin and Ji, 2019] Ying Lin and Heng Ji. An attentive fine-grained entity typing model with latent type representation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 6198–6203, 2019.

[Ling and Weld, 2012] Xiao Ling and Daniel S Weld. Fine-grained entity recognition. In *AAAI*, volume 12, pages 94–100, 2012.

[Miyato *et al.*, 2018] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

[Moreno-Torres *et al.*, 2012] Jose G Moreno-Torres, Troy Raeder, RocíO Alaiz-RodríGuez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, 2012.

[Onoe and Durrett, 2019] Yasumasa Onoe and Greg Durrett. Learning to denoise distantly-labeled data for entity typing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2407–2417, 2019.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[Ren *et al.*, 2016a] Xiang Ren, Wenqi He, Meng Qu, Lifu Huang, Heng Ji, and Jiawei Han. Afet: Automatic fine-grained entity typing by hierarchical partial-label embedding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1369–1378, 2016.

[Ren *et al.*, 2016b] Xiang Ren, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, and Jiawei Han. Label noise reduction in entity typing by heterogeneous partial-label embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1825–1834. ACM, 2016.

[Xin *et al.*, 2018] Ji Xin, Hao Zhu, Xu Han, Zhiyuan Liu, and Maosong Sun. Put it back: Entity typing with language model enhancement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 993–998, 2018.

[Xiong *et al.*, 2019] Wenhan Xiong, Jiawei Wu, Deren Lei, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. Imposing label-relational inductive bias for extremely fine-grained entity typing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pages 773–784, 2019.

[Xu and Barbosa, 2018] Peng Xu and Denilson Barbosa. Neural fine-grained entity type classification with hierarchy-aware loss. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pages 16–25, 2018.

[Yang *et al.*, 2019] Xiyuan Yang, Xiaotao Gu, Sheng Lin, Siliang Tang, Yueting Zhuang, Fei Wu, Zhigang Chen, Guoping Hu, and Xiang Ren. Learning dynamic context augmentation for global entity linking. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 271–281, 2019.

[Yogatama *et al.*, 2015] Dani Yogatama, Daniel Gillick, and Nevena Lazic. Embedding methods for fine grained entity type classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 2, pages 291–296, 2015.