Multi-objective Privacy-preserving Text Representation Learning

Huixin Zhan
huixin.zhan@ttu.edu
Texas Tech University
Lubbock, TX, USA

Kun Zhang
kzhang@xula.edu
Xavier University of Louisiana
New Orleans, LA, USA

Chenyi Hu
chu@uca.edu
University of Central Arkansas
Conway, AR, USA

Victor S. Sheng*
victor.sheng@ttu.edu
Texas Tech University
Lubbock, TX, USA

ABSTRACT

Private information can either take the form of key phrases that are explicitly contained in the text or be implicit. For example, demographic information about the author of a text can be predicted with above-chance accuracy from linguistic cues in the text itself. Letting alone its explicitness, some of the private information correlates with the output labels and therefore can be learned by a neural network. In such a case, there is a tradeoff between the utility of the representation (measured by the accuracy of the classification network) and its privacy. This problem is inherently a multi-objective problem because these two objectives may conflict, necessitating a trade-off. Thus, we explicitly cast this problem as multi-objective optimization (MOO) with the overall objective of finding a Pareto stationary solution. We, therefore, propose a multiple-gradient descent algorithm (MGDA) that enables the efficient application of the Frank-Wolfe algorithm [10] using the line search. Experimental results on sentiment analysis and part-of-speech (POS) tagging show that MGDA produces higher-performing models than most recent proxy objective approaches, and performs as well as single objective baselines.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Security and privacy** → **Human and societal aspects of security and privacy**.

KEYWORDS

multi-objective; representation learning; privacy; text

ACM Reference Format:
Huixin Zhan, Kun Zhang, Chenyi Hu, and Victor S. Sheng. 2021. Multi-objective Privacy-preserving Text Representation Learning. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3459637.3482147

*Victor S. Sheng is the corresponding author.

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482147

1 INTRODUCTION

Textual information is one of the most significant portions of data that users generate by participating in different online activities such as leaving online reviews and posting tweets. On one hand, textual data consists of abundant information about users' behavior preferences for data consumers to study. On the other hand, publishing complete and intact users' textual data risks exposing the privacy information to an adversary. This scenario usually arises when the computation of a neural network is shared across multiple devices, e.g., some hidden representations are computed locally and send to a cloud-based model. In this case, the hidden representations are easy to be obtained by the adversary during uploading the data [4].

Private information can take the form of key phrases explicitly contained in the text. However, it can also be implicit. For example, demographic information about the author of a text can be predicted with above-chance accuracy from linguistic cues in the text itself [16, 17]. Some of the private information correlates with the output labels and therefore can be learned by a neural network as the saying "*you are what you write*" goes. Therefore, there is a tradeoff between the **utility** of the representation (measured by the accuracy of the neural network) and its **privacy**. This problem is inherently a multi-objective optimization problem because these two objectives, i.e., utility and privacy, are conflict, necessitating a trade-off. Thus, we explicitly cast this problem as multi-objective optimization (MOO), with the overall objective of finding a Pareto stationary solution. Unlike the current privacy preserving works optimizing a proxy objective to minimize a weighted linear combination of per-objective losses [1, 4, 12], we propose a multiple-gradient descent algorithm (MGDA) that enables the efficient application of the Frank-Wolfe algorithm [10] using the line search.

Our contributions are summarized as: **1)** We explicitly cast the privacy-preserving text representation learning problem as MOO, with the overall objective of finding a Pareto stationary solution. **2)** Our MGDA enables the efficient application of the Frank-Wolfe algorithm [10] using the line search. **3)** The experimental results show that MGDA converges to a point on the Pareto set that preserves the users' private information while retaining the utility by solving an optimization problem to decide the update over the shared parameters.

2 RELATED WORKS

The users' privacy concerns mandate data publishers to protect privacy by anonymizing the data before sharing it with data consumers. Currently, various protection methods for **structured** data

have been developed over the years such as k-anonymity [18] and differential privacy [7]. However, these methods are insufficient for user-generated textual data because (1) the data is weakly structured, noisy, and informal, (2) these methods may impose a significant utility loss for the designed objective.

For current privacy-preserving text representation learing approaches, ADV [12] trains a deep model with adversarial learning to explicitly obscure individuals' private information, Multidetasking [4] focuses on defending the adversarial classification. This method modifies the objective of the main classifier to incorporate a penalty when the adversarial classifier is good at reconstructing the private information. DPText [1] is proposed to learn a differentially private representation by minimizing the chance of attacker to infer whether target text representation is in the database with the weighted sum objective. All the aforementioned approaches haven't leverage the MOO setting but they all minimize a proxy objective, which is a weighted linear combination of per-objective losses. Although the linear-combination formulation is appealing, it typically either requires an expensive grid search [13] over various scalings or the use of a heuristic [3] to achieve a good performance. However, it is worth to notice that the linear-combination formulation is only sensible when there is a parameter set that is effective across all objectives, which is rarely the case. In order to find the Pareto stationary solutions, we cast this problem as MOO. A variety of algorithms for MOO exist. One such approach is the multiple-gradient descent algorithm (MGDA), which uses gradient-based optimization and provably converges to a point on the Pareto front [5]. In this work, we further propose a MGDA approach that enables the efficient application of the Frank-Wolfe algorithm [10] via the line search.

3 METHOD

In this section, we introduce the adversarial scenario, the adversarial attacks, and the defenses against adversarial attacks as MOO.

3.1 The Adversarial Scenario

We propose to frame the training of the adversarial scenario via a two-agent process: (1) the main agent and (2) an adversary. They are exploiting a setting similar to Generative Adversarial Networks (GAN) [8], where the adversary learns to extract the privacy information from the hidden representation, whereas the main agent learns to perform its main task and to make the adversary difficult in privacy extraction. Each example consists of a triple $(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)$, where $\mathbf{x}_i$ is a natural language text for each data instance, $\mathbf{y}_i$ is a single label, e.g. topic or sentiment, and $\mathbf{z}_i$ is a vector of private information contained in $\mathbf{x}_i$. In our two agents setting, (1) the adversary learns to predict $\mathbf{z}_i$ from the hidden representation $r(\mathbf{x}_i)$ of $\mathbf{x}_i$ used by the main agent, and (2) the main agent learns to predict $\mathbf{y}_i$ from $\mathbf{x}_i$. In order to evaluate the utility (accuracy) and privacy of a specific model, we proceed in four steps (shown in Figure 1): **1)** Training of the main classifier on $(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)$ and evaluation of its accuracy; **2)** Generation of a dataset of pairs $(r(\mathbf{x}_i), \mathbf{z}_i)$ for the adversary, where r is the representation function of the main classifier; **3)** Training of the adversary's network on $(r(\mathbf{x}_i), \mathbf{z}_i)$; **4)** Evaluation of the adversary's performance for measuring privacy.

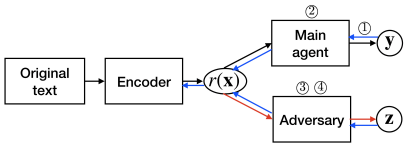


Figure 1: General setting illustration. The main agent predicts a label $\mathbf{y}_i$ from a text $\mathbf{x}_i$, the adversary tries to recover some private information $\mathbf{z}_i$ contained in $\mathbf{x}_i$ from the hidden representation (in red). All back propagations are shown in blue.

3.2 Adversarial Attack

A classifier predicts the attribute $z(\mathbf{x}_i)$ used as a proxy for recovering the privacy information of $\mathbf{x}_i$. In the adversarial attack, we generalize a dataset made of pairs $(r(\mathbf{x}_i), \mathbf{z}(\mathbf{x}_i))$, where $r(\mathbf{x}_i)$ is the hidden representation from the main classifier and $\mathbf{z}(\mathbf{x}_i)$ is a vector of private categorical variables in practice. Formally, for a single data point $(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)$, the adversarial classifier optimizes:

$$\mathcal{L}_a(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i; \boldsymbol{\Theta_a}, \Theta_r) = -\log P(\mathbf{z}_i | r(\mathbf{x}_i; \Theta_r); \boldsymbol{\Theta_a}), \tag{1}$$

where Θ_r represents the parameters for encoding the hidden representation $r(\mathbf{x}_i)$. Once the main model has been trained, the parameters Θ_r are fixed. $\boldsymbol{\Theta}_a$ represents the parameters for the adversary.

3.3 The Defense Strategy as MOO

The Basic Defense. The basic defense applies a weighted sum of per-objective losses and then optimizes the LSTM:

$$\begin{aligned}\mathcal{L}_m(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i; \Theta_r, \boldsymbol{\Theta_p}) = &-\alpha \log P(\mathbf{y}_i | r(\mathbf{x}_i; \Theta_r); \boldsymbol{\Theta_p}) \\ &- \beta \log P(\neg \mathbf{z}_i | r(\mathbf{x}_i; \Theta_r); \boldsymbol{\Theta_p}).\end{aligned} \tag{2}$$

Both $\alpha > 0$ and $\beta > 0$ control the relative weight of both terms for improving the classification accuracy and deceiving the adversary. As in a GAN, the losses of both classifiers are interdependent, but their parameters are distinct: the adversary can only update $\boldsymbol{\Theta_a}$ or $\boldsymbol{\Theta}'_{\boldsymbol{a}}$ and the main classifier can only update Θ_r and $\boldsymbol{\Theta_p}$. We simply use $\Theta = \{\Theta_r, \boldsymbol{\Theta_p}\}$ because Θ_r and $\boldsymbol{\Theta_p}$ are trained end-to-end.

The MOO Formulation. Consider the input space $\mathcal{X} = \{\mathbf{x}_i\}_{i \in [N]}$ and a collection of objective spaces $\{\mathcal{Y}_k\}_{k \in [K]}$, where $K = 2$, N is the number of data points, $\{\mathcal{Y}_1\} = \{\mathbf{y}_i\}_{i \in [N]}$, and $\{\mathcal{Y}_2\} = \{\neg \mathbf{z}_i\}_{i \in [N]}$. We consider a parametric hypothesis class per-objective as as $f^k(\mathbf{x}; \Theta)$, such that some parameters (Θ^{sh}) are shared between objectives and some (Θ^k) are objective-specific. We also consider objective-specific loss functions $\mathcal{L}^k_m(\cdot) : \mathcal{X}_k \to \mathbb{R}$, where $\mathcal{L}^k_m(\cdot)$ is the empirical loss of the objective k, defined as $\mathcal{L}^k_m(\cdot) = -\frac{1}{N} \sum_i \mathcal{L}(\log P(\mathbf{y}_i | \mathbf{x}_i))$, and $\mathbf{y}_i \in \mathcal{Y}_k$. In order to optimize two possibly conflicting objectives independently, we specify the formulation as a vector-valued loss $\mathbf{L}_m$: $\min_{\Theta^{sh}, \Theta^1, \Theta^2} \mathbf{L}_m(\Theta^{sh}, \Theta^1, \Theta^2) = \min_{\Theta^{sh}, \Theta^1, \Theta^2} (\mathcal{L}^1_m(\Theta^{sh}, \Theta^1), \mathcal{L}^2_m(\Theta^{sh}, \Theta^2))^T$ as typical MOO suggested. In a MOO case, if solution Θ is better for the first objective whereas Θ' is better for the second objective, it is not possible to compare which solution is better. In order to analyze which solution is better, we here introduce the definitions for **dominate** and **Pareto front**.

Definition 3.1. A solution Θ **dominates** a solution Θ' if $\mathcal{L}(\Theta) \leq \mathcal{L}(\Theta')$ for all objectives k and $\mathbf{L}(\Theta) \neq \mathbf{L}(\Theta')$. A solution $\Theta^{\star}$ is called Pareto optimal if there exists no solution Θ that dominates $\Theta^{\star}$. A variety of Pareto optimal solutions ($\mathcal{V}_{\Theta}$) distributed in the so-called **Pareto front** $\mathcal{V}_{\mathrm{L}} = \{\mathbf{L}(\Theta)\}_{\Theta \in \mathcal{V}_{\Theta}}$[1].

MGDA can solve the MOO problem to the local stationary points via gradient descent. The Pareto stationary points leverage the Karush-Kuhn-Tucker (KKT) conditions [6] as follows: **1)** There exist $w_1, w_2 \geq 0$ such that $\sum_{k=1}^{K} w_k = 1$ and $\sum_{k=1}^{K} w_k \nabla_{\Theta^{sh}} \mathcal{L}_m^k(\Theta^{sh}, \Theta^k)$, where $\mathbf{w} = (w_1, w_2)$ represents the set of weights. **2)** For all objectives, $\nabla_{\Theta^k} \mathcal{L}_m^k(\Theta^{sh}, \Theta^k) = 0$.

Solving the optimization problem. Considering the optimization problem:

$$\min_{\Theta^{sh}, \Theta^1, \Theta^2} \{ \left\| \sum_{k=1}^{K} w_k \mathcal{L}_m^k(\Theta^{sh}, \Theta^k) \right\|_2^2 \Big| \sum_{k=1}^{K} w_k = 1, w_k \geq 0, \forall k \}. \quad (3)$$

Désidéri [5] showed that either the resulting point satisfies the KKT conditions and the solution to this optimization problem is 0, or the solution gives a descent direction that improves all objectives. When $K = 2$, the optimization problem can be defined as $\min_{w_1 \in [0,1]} \left\| w_1 \nabla_{\Theta^{sh}} \mathcal{L}_m^1(\Theta^{sh}, \Theta^1) + (1 - w_1) \nabla_{\Theta^{sh}} \mathcal{L}_m^2(\Theta^{sh}, \Theta^2) \right\|_2^2$, which is a one-dimensional quadratic function of w_1 with an analytical solution: $w_1 = [\frac{(\nabla_{\Theta^{sh}} \mathcal{L}^2(\Theta^{sh}, \Theta^2) - \nabla_{\Theta^{sh}} \mathcal{L}^1(\Theta^{sh}, \Theta^1))^T \nabla_{\Theta^{sh}} \mathcal{L}^2(\Theta^{sh}, \Theta^2)}{\left\| \nabla_{\Theta^{sh}} \mathcal{L}^1(\Theta^{sh}, \Theta^1) - \nabla_{\Theta^{sh}} \mathcal{L}^2(\Theta^{sh}, \Theta^2) \right\|_2^2}]_+$, where $[\cdot]_+$ represents clipping $\cdot$ to as $\max(\min(\cdot, 1), 0)$. We use g and $\bar{g}$ to represent the gradients $\nabla_{\Theta^{sh}} \mathcal{L}_m^1(\Theta^{sh}, \Theta^1)$ and $\nabla_{\Theta^{sh}} \mathcal{L}_m^2(\Theta^{sh}, \Theta^2)$, and further show the solution in Algorithm 1 and Figure 2. We can see that the solution is either a vector itself or a perpendicular vector [2]. Although this is only applicable when $K = 2$, this enables the efficient application of the Frank-Wolfe algorithm [10] since the line search can be solved analytically. We give all the update equations for the Frank-Wolfe solver in Algorithm 2. The overall Θ parameter update is shown in Algorithm 2 as well, where we update the objective-specific parameters Θ^k independently, and we update the shared parameters Θ^{sh} based on MGDA.

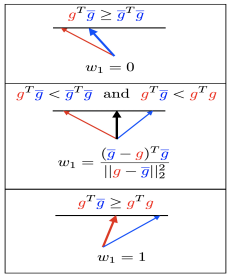


Figure 2: Visualisation of solving $\min_{w_1 \in [0,1]} \left\| w_1 g + (1 - w_1)\bar{g} \right\|_2^2$. The solution is either a vector itself or a perpendicular vector (in bold).

Algorithm 1: Computing $\min_{w_1 \in [0,1]} \left\| w_1 g + (1 - w_1)\bar{g} \right\|_2^2$

if $g^T\bar{g} \geq g^Tg$ **then**
  $w_1 = 1$
else if $g^T\bar{g} \geq \bar{g}^T\bar{g}$ **then**
  $w_1 = 0$
else
  $w_1 = \frac{(\bar{g}-g)^T\bar{g}}{\left\|g-\bar{g}\right\|_2^2}$
end if

Algorithm 2 Update equations for the MGDA algorithm

for $t \leftarrow 0$ to T **do**
  $\Theta^k = \Theta^k - \eta \nabla_{\Theta^k} \mathcal{L}_m^k(\Theta^{sh}, \Theta^k)$ {Gradient descent on the objective-specific parameters}
end for
Initialize $\mathbf{w} = (w_1, w_2) = (\frac{1}{K}, \frac{1}{K})$, $\mathbf{s}^- = \mathbf{w}$ {Here starts the Frank-Wolfe solver}
for $t \leftarrow 0$ to T **do**
  Precompute $\mathbf{X}$ st. $\mathbf{X}_{i,j} = (\nabla_{\Theta^{sh}} \mathcal{L}_m^k(\Theta^{sh}, \Theta^i))^T (\nabla_{\Theta^{sh}} \mathcal{L}_m^k(\Theta^{sh}, \Theta^j))$
  while Number of Iterations **do**
    for k **do**
      Find $\mathbf{s}$ s.t. $\langle \mathbf{s}, \mathbf{X}_{,k} \rangle < \langle \mathbf{s}^-, \mathbf{X}_{,k} \rangle + \epsilon'$
      $\gamma = \arg\min_{\gamma} ((1 - \gamma)\mathbf{w} + \gamma s_k)^T \mathbf{X} ((1 - \gamma)\mathbf{w} + \gamma s_k)$ {Here updates γ via Algorithm 1}
      $\mathbf{w} = (1 - \gamma)(e^{\gamma} - 1)\mathbf{w} + \gamma(e^{\gamma} - 1)s_k$ {Here updates $\mathbf{w}$ via the line search and ends the Frank-Wolfe solver}
    end for
  end while
  $\Theta^{sh} = \Theta^{sh} - \eta \sum_{k=1}^{K} \nabla_{\Theta^{sh}} w_k \mathcal{L}_m^k(\Theta^{sh}, \Theta^k)$ {Gradient descent on shared parameters}
end for

4 EXPERIMENTS

In this section, we discuss the tasks and datasets, the settings, and the results for the Trustpilot dataset for both tasks.

4.1 Tasks and dataset

We conduct our experiment on two types of tasks: sentiment analysis and part-of-speech (POS) tagging. For the **sentiment analysis** task, we use the Trustpilot dataset [9] for sentiment analysis. This corpus contains reviews associated with a sentiment score on a scale of five points. We use the five subcorpora corresponding to five areas, i.e., Denmark, France, Germany, United Kingdom (UK), and United States (US). We extract examples containing both the birth year and gender of the author of the review and use these as the private information. For the **POS Tagging** task, we use the TrustPilot English POS tagged dataset [9], which consists of 600 sentences, each labelled with both the gender and age of the author, and manually POS tagged based on the Google Universal POS tagset [15].

For both tasks, we classify the age of the author into two categories ('under 35' and 'over 45') based on a previous work [9]. We randomly split each subcorpus into a training set (80%), a validation set (10%) and a test set (10%). The task for the main classifier predicts the sentiment analysis results or the grammatical tagging $\mathbf{y}_i$ from the training example $\mathbf{x}_i$. $\mathbf{z}_i$ is a vector of binary variables, representing, e.g., age or gender information about the author. The attacker predicts $\mathbf{z}_i$.

4.2 Settings

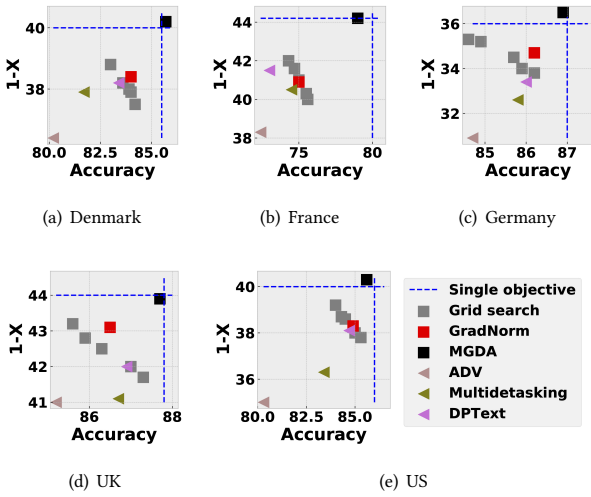
Implementation details. We implement our model via Dynet [14]. Both of the main model and the attacker have a single hidden layer of 64 units with a ReLU activation. The word embeddings have 32 units. We used the Adam optimizer [11] with the default learning rate, and 0.2 dropout rate for the LSTM. For each dataset, and the LSTM state dimension 128, we train the main model for 8 epochs (sentiment analysis) or 16 epochs (POS tagging). Then, the adversarial model is trained based on the manipulated embeddings for 16 epochs.

[1] Every Pareto optimal point is Pareto stationary

Table 1: The utility and privacy results for the POS tagging

Corpora	Denmark			France			Germany			UK			US			Overall		
Metrics	**Acc**	**Age**	**Gen**	**Acc**	**Age**	**Gen**	**Acc**	**Age**	**Gen**	**Acc**	**Age**	**Gen**	**Acc**	**Age**	**Gen**	**Acc**	**Age**	**Gen**
Single objective (utility)	0.92	0.18	0.33	0.88	0.15	0.30	0.95	0.18	0.36	0.95	0.20	0.38	0.94	0.21	0.36	0.94	0.20	0.35
Single objective (privacy)	0.68	0.85	0.78	0.64	0.79	0.74	0.62	0.90	0.84	0.68	0.91	0.85	0.70	0.90	0.84	0.69	0.89	0.83
Grid search [13]	0.88	0.75	0.57	0.87	0.74	0.53	0.93	0.80	0.64	0.94	0.83	0.65	0.91	0.80	0.62	0.91	0.79	0.60
GradNorm [3]	0.91	0.77	0.59	0.88	0.75	0.56	0.93	0.83	0.65	0.93	0.89	0.67	0.93	0.81	0.64	0.93	0.79	0.61
ADV [12]	0.87	0.63	0.47	0.84	0.60	0.43	0.89	0.66	0.57	0.89	0.70	0.53	0.87	0.70	0.60	0.89	0.65	0.50
Multidetasking [4]	0.89	0.65	0.50	0.84	0.63	0.47	0.91	0.73	0.60	0.91	0.73	0.59	0.90	0.72	0.61	0.90	0.68	0.58
DPText [1]	0.90	0.69	0.54	0.85	0.69	0.52	0.93	0.80	0.61	0.93	0.80	0.64	0.93	0.78	0.63	0.92	0.78	0.61
MGDA	**0.92**	**0.80**	**0.65**	**0.88**	**0.76**	**0.69**	**0.93**	**0.83**	**0.71**	**0.94**	**0.84**	**0.72**	**0.94**	**0.84**	**0.71**	**0.93**	**0.83**	**0.70**

Methods for comparison. We compare our experimental results with **Single objective** baselines, and some SOTA privacy-preserving representation learing approaches, i.e., **ADV** [12], **Multidetasking** [4], and **DPText** [1]. We also compare our methods with some common multi-objective baselines, i.e.,**Grid search** [13] and **GradNorm** [3]. For the Single objective model, we train the main classifier without defense. We select the model with the best accuracy and privacy with $\alpha = 1$, $\beta = 0$ (only focusing on utility), and $\alpha = 0$, $\beta = 1$ (only focusing on privacy), respectively. For the Multidetasking method, each LSTM state dimension ($\{8, 16, 32, 64, 128\}$), we train the main model for 8 epochs (sentiment analysis) or 16 epochs (POS tagging), and select the model with the best accuracy on the development set with $\alpha = \beta = 1$ when the privacy is above a certain threshold. For methods that require additional parameters, such as ADV and DPText, all parameters follow the original paper.



(a) Denmark (b) France (c) Germany

(d) UK (e) US

Figure 3: The utility and privacy results for the sentiment analysis (%)

Evaluation metrics. For the main classifier, we report a single accuracy metric. For measuring the privacy of a representation, we use the following two metrics. For the **sentiment analysis**: we use $1 - X$ as the privacy preserving ability, where X is the average of the accuracy of the attacker on the prediction of gender and age. For the **POS tagging** task, we use $1 - F$, where F is the F-score computed over the set of binary variables in $\mathbf{z}_i$. These metrics are **the higher the better** since a higher value of $1-X$ or $1-F$ indicates a higher error rate for the adversary, and therefore it indicates that the privacy-preserving ability of the main agent is higher.

4.3 Experimental results for both tasks

For the sentiment analysis task, as shown in Figure 3, any weighted sum methods (**ADV**, **Multidetasking**, and **DPText**) obtain lower accuracy than the more advanced normalization algorithms, i.e., **Grid search**, and **GradNorm**. All these five algorithms obtain lower accuracy than solving each objective separately (the **Single objective** approach). The two objectives appear to compete for model capacity since an increase in the accuracy of one task results in a decrease in the accuracy of the other. The **GradNorm** finds the solution that are slightly better than **Grid search** but it is distinctly worse than the **Single objective** approach, i.e., the solution is dominated by the **Single objective** approach. In contrast, our **MGDA** method finds a solution that efficiently utilizes the model capacity and yields accuracies that are competitive with the **Single objective** solutions. As an example, in the Denmark subcorpora, our MGDA results dominate the extreme values of the **Single objective** solutions. our MGDA results lie on a Pareto front with the **Single objective** solutions for the France, Germany, and US subcorpora. All results are show in percentage (%).

Table 1 shows the experimental results for the POS tagging. All accuracies are denoted as **"Acc"** in the Table 1) and the privacy results for age and gender are shown in $1-F$ (denoted as **"Age"** and **"Gen"** in the Table 1). We can conclude that all weighted sum methods (**ADV**, **Multidetasking**, and **DPText**) obtain lower accuracy than the more advanced multi-objective normalization algorithms such as **Grid search**, and **GradNorm**. All these five algorithms obtain lower accuracy than solving each objective separately (the **Single objective** approach). However, our MGDA results (in bold) are as good as the **Single objective** solutions.

5 CONCLUSION

For privacy-preserving text representation learning, the ultimate goal is to preserve user privacy while ensuring the utility of the published data for future tasks and usages. In order to address the trade-off, we cast this problem as multi-objective optimization (MOO) with the overall objective of finding a Pareto stationary solution. A multiple-gradient descent algorithm (MGDA) is proposed to enable the efficient application of the Frank-Wolfe algorithm via the line search. For both sentiment analysis and part-of-speech tagging, MGDA produces higher-performing models than most recent proxy objective approaches. Besides, our results lie on a Pareto front with the single objective baselines.

REFERENCES

[1] Ghazaleh Beigi, Kai Shu, Ruocheng Guo, Suhang Wang, and Huan Liu. 2019. I am not what i write: Privacy preserving text representation learning. *arXiv preprint arXiv:1907.03189* (2019).

[2] Yongcan Cao and Huixin Zhan. 2021. Efficient Multi-objective Reinforcement Learning via Multiple-gradient Descent with Iteratively Discovered Weight-Vector Sets. *Journal of Artificial Intelligence Research* 70 (2021), 319–349.

[3] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*. PMLR, 794–803.

[4] Maximin Coavoux, Shashi Narayan, and Shay B Cohen. 2018. Privacy-preserving neural representations of text. *arXiv preprint arXiv:1808.09408* (2018).

[5] Jean-Antoine Désidéri. 2012. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus Mathematique* 350, 5-6 (2012), 313–318.

[6] Axel Dreves, Francisco Facchinei, Christian Kanzow, and Simone Sagratella. 2011. On the solution of the KKT conditions of generalized Nash equilibrium problems. *SIAM Journal on Optimization* 21, 3 (2011), 1082–1108.

[7] Cynthia Dwork. 2008. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*. Springer, 1–19.

[8] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. *arXiv preprint arXiv:1406.2661* (2014).

[9] Dirk Hovy, Anders Johannsen, and Anders Søgaard. 2015. User review sites as a resource for large-scale sociolinguistic studies. In *Proceedings of the 24th international conference on World Wide Web*. 452–461.

[10] Martin Jaggi. 2013. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*. PMLR, 427–435.

[11] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[12] Yitong Li, Timothy Baldwin, and Trevor Cohn. 2018. Towards robust and privacy-preserving text representations. *arXiv preprint arXiv:1805.06093* (2018).

[13] Antonio J Nebro, Enrique Alba, and Francisco Luna. 2007. Multi-objective optimization using grid computing. *Soft Computing* 11, 6 (2007), 531–540.

[14] Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, et al. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980* (2017).

[15] Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086* (2011).

[16] Daniel Preoţiuc-Pietro, Vasileios Lampos, and Nikolaos Aletras. 2015. An analysis of the user occupational class through Twitter content. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 1754–1764.

[17] Sara Rosenthal and Kathleen McKeown. 2011. Age prediction in blogs: A study of style, content, and online behavior in pre-and post-social media generations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 763–772.

[18] Latanya Sweeney. 2002. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 05 (2002), 571–588.