An Investigation towards Differentially Private Sequence Tagging in a Federated Framework

Abhik Jana[1] and **Chris Biemann**[1]

[1] Language Technology Group, Dept. of Informatics, Universität Hamburg, Germany
jana@informatik.uni-hamburg.de, biemann@informatik.uni-hamburg.de

Abstract

To build machine learning-based applications for sensitive domains like medical, legal, etc. where the digitized text contains private information, anonymization of text is required for preserving privacy. Sequence tagging, e.g. as used for Named Entity Recognition (NER), can help to detect private information. However, to train sequence tagging models, a sufficient amount of labeled data are required but for privacy-sensitive domains, such labeled data also can not be shared directly. In this paper, we investigate the applicability of a privacy-preserving framework for sequence tagging tasks, specifically NER. Hence, we analyze a framework for the NER task, which incorporates two levels of privacy protection. Firstly, we deploy a federated learning (FL) framework where the labeled data are neither shared with the centralized server nor with the peer clients. Secondly, we apply differential privacy (DP) while the models are being trained in each client instance. While both privacy measures are suitable for privacy-aware models, their combination results in unstable models. To our knowledge, this is the first study of its kind on privacy-aware sequence tagging models.

1 Introduction

The emergence of substantial amounts of digitized unstructured text gives rise to train machine learning models for various downstream applications. But for sensitive domains like medical, legal, etc., the text documents contain private sensitive information, which is supposed to be anonymized for preserving privacy. The first step for anonymizing text is to detect the span of the private information and identify the type of information. Sequence tagging is the kind of task that can help in anonymization. However, to train such a sequence tagging model, a significant amount of labeled data are required. But the labeled data contains – by definition – private sensitive information and are often divided across different data silos. There are legal regulatory policies as well by the US Health Insurance Portability and Accountability Act (HIPAA)[1] and EU General Data Protection Regulation (GDPR) (Agencia-Espanola-Proteccion-Datos, 2019), which restricts such sensitive data access. On the other hand, to train a centralized machine learning model, there is a requirement of aggregating such distributed data in a central server, which can cause privacy breaches. Hence, there is a requirement for a privacy-preserving sequence tagging framework that complies with the data protection policies.

Federated learning (FL) (McMahan et al., 2017) is one such paradigm that provides a framework to train a centralized machine learning model without sharing the distributed data from different data silos. Since the raw data from different data silos are not being shared in this framework, the primary level of privacy is maintained. However, the FL framework is also vulnerable to several inference attacks (Bagdasaryan et al., 2020; Bonawitz et al., 2017; Geyer et al., 2017) in some scenarios. To mitigate such inference attacks, differential privacy (Dwork et al., 2006) was developed, which comes with a theoretically guaranteed measurement of privacy. There have been many recent research works on deploying the FL framework in several applications like image classification (Wang et al., 2019), emotion detection (Chhikara et al., 2021), anonymization (Choudhury et al., 2020), robotics (Imteaj and Amini, 2020), etc. and also in medical domain (Rajendran et al., 2021; Kerkouche et al., 2021; Choudhury et al., 2019; Ge et al., 2020). Researchers also investigated the scope of the differentially private algorithm in several applications (Zhao et al., 2020; Koda et al., 2020; Hu et al., 2020; Chen et al., 2018). However, for sequence

[1] https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 30–35

https://doi.org/10.26615/978-954-452-056-4_004

tagging tasks, the applicability of the FL framework along with differential privacy (DP) is yet to be explored. For our study, we focus on one such sequence tagging task, namely named entity recognition (NER). We argue that this is quite close to a sequence-tagging based anonymization task and it allows us to study the effects of privatization measures in the task performance.

In this paper, we prepare an FL framework following the FederatedAveraging (McMahan et al., 2017) approach and deploy a differentially private stochastic gradient descent (Abadi et al., 2016) optimizer while training to obtain differential privacy for the NER task. As base NER models, we consider two variants of one of the state-of-the-art approaches for sequence labeling, that uses the LSTM variant (Hochreiter and Schmidhuber, 1997) of bidirectional recurrent neural networks (BiLSTMs). Note that the aim of our work is not to produce a state-of-the-art performance for NER tasks. Rather, our focus is to analyze how the performance of NER models varies in a differentially private federated learning framework. Therefore, we do a comprehensive analysis using CoNLL 2003 English NER dataset and investigate the effect of different levels of privacy on the performance of NER models.

2 Methodology

As we deal with the named entity recognition (NER) task, we attempt with two variations of BiLSTM based models. In one variant we keep TimeDistributed Dense layer (TDDL) as the final layer with activation function – 'softmax' and in another variant, we keep the conditional random field (CRF) (Huang et al., 2015) as the final layer. In a nutshell, both the variants have three layers: Input embedding layer, Bidirectional LSTM layer, and either TDDL or CRF as the final layer. We refer to those model variants 'BI-LSTM-TDDL' and 'BI-LSTM-CRF' respectively for our study. Next, we will discuss the two privacy mechanisms: Federated Learning and Differential Privacy.

Federated Learning: The objective of the Federated Learning framework is to train a centralized model from data distributed across multiple client data silos, eliminating the need for raw data sharing. We adapt the well-accepted FederatedAveraging algorithm proposed by McMahan et al. (2017). As per the process, first, a global model from the server site is shared across n client sites. Next, each client site trains the model based on its local data. After the training is complete, the parameter updates of the local models are then subsequently sent to the central aggregation server. Next, the central server computes the average of all the model parameters over n clients and updates the global model accordingly. The whole process continues for specified number of rounds (t) and in each round a random set of m clients ($m <= n$) participate. For this study, in each round we allow all the n client sites to participate in the process.

Differential Privacy: The objective of differential privacy is to provide a strong criterion for privacy preservation of distributed data processing systems. This is a widely used privacy-preserving mechanism due to its strong information-theoretic guarantees (Dwork and Roth, 2014), algorithmic simplicity, and relatively small systems overhead. By definition, any randomized algorithm $A(D)$ satisfies ϵ - differential privacy if for all datasets D and D', that differ by a single record, and for all sets $S \in R$, where R is the range of A,

$$Pr[A(D) \in S] \leq e^{\epsilon} Pr[A(D') \in S]$$

where ϵ, is a non-negative number. ϵ measures the strength of the privacy guarantee of the algorithm. It gives a bound on how much the probability of particular model output can vary by including (or removing) a single training example. The lower the value of ϵ, the higher the privacy. There are several methods for incorporating differential privacy in an algorithm. For this study, we adopt the approach that relies on Differentially Private Stochastic Gradient Descent (DP-SGD) optimizer proposed by Abadi et al. (2016). As per this approach, after sampling a micro-batch of training points we need to compute the loss and gradient of the loss. Thereafter, we need to clip gradients, per the training example included in the micro-batch. Next, we need to add random noise to the clipped gradients and multiply these clipped and noised gradients by the learning rate and apply the product to update model parameters.

3 Experiments and Discussions

Since the objective of our work is to investigate the effect of federated learning (**FL**) framework and differential privacy (**DP**) on the state-of-the-art NER model, we design our analysis in four different phases. First, we observe the basic NER model performance without using FL as well as DP. In the second phase, we incorporate DP in the optimizer

of the NER model and analyze its behavior. Third, we analyze the performance of the NER model in an FL framework, but none of the optimizers of the clients are DP enabled. In the fourth phase, we investigate the performance of the NER model by deploying the FL framework and incorporating DP into the client-side optimizers. For our analysis, we use CoNLL 2003 English NER dataset, having a training set size of 14987, validation set size of 3466, and test set size of 3684. Each token in the dataset is tagged with other (O) or one of the four entity types: Person (PER), Location (LOC), Organization (ORG), Miscellaneous (MISC). Considering the BIO2 annotation standard, the number of possible labels for each token is 9.

Model	Precision	Recall	F-measure
BI-LSTM-TDDL	0.916	0.922	0.916
BI-LSTM-CRF	0.892	0.862	0.864

Table 1: Performance of Basic NER models.

Noise-multiplier	0.5	1	5	10	50
ϵ	3.73	0.688	0.05	0.024	0.022
BI-LSTM-TDDL					
Precision	0.915	0.911	0.917	0.914	0.911
Recall	0.919	0.911	0.924	0.912	0.913
F-measure	0.913	0.906	0.916	0.908	0.908
BI-LSTM-CRF					
Precision	0.869	0.727	0.825	0.889	0.839
Recall	0.847	0.809	0.786	0.834	0.816
F-measure	0.839	0.75	0.802	0.840	0.815

Table 2: Performance of NER models after incorporating ϵ differentially Private SGD.

$(n) \rightarrow$	2	5	10	15	20
Precision	0.905	0.881	0.824	0.792	0.786
Recall	0.903	0.885	0.830	0.814	0.803
F-measure	0.897	0.874	0.805	0.784	0.766

Table 3: Performance of BI-LSTM-TDDL model in the FL setup, with increasing number of clients (n).

Training Data	Precision		Recall		F-measure	
	mean	sd	mean	sd	mean	sd
20 %	0.756	0.002	0.798	0.000	0.762	0.000
40 %	0.814	0.006	0.824	0.001	0.797	0.001
60 %	0.832	0.001	0.842	0.000	0.820	0.000
80 %	0.845	0.009	0.854	0.000	0.835	0.001
100 %	0.861	0.005	0.863	0.000	0.846	0.001

Table 4: Performance of BI-LSTM-TDDL model in the FL setup, with increasing % of training data in each client. Number of clients = 5. Mean and standard deviation (sd) are computed over 5 different simulations.

Without DP, Without FL: We experiment with two different NER models described in Section 2[2]. The vector embeddings used is GloVe (trained on Common Crawl corpus) from spaCy library[3], the dimension of which is 300. The performance of these two models (BI-LSTM-TDDL, BI-LSTM-CRF) are presented in Table 1. Note that, since our objective is not to produce a state-of-the-art performance for NER tasks, we do not attempt to tune the hyper-parameters to obtain the best possible performance. However, the F-measure obtained by BI-LSTM-TDDL is comparable to the state-of-the-art (Akbik et al., 2018) F-measure of 0.9309. On the other hand, we put our effort into analyzing the behavior of these NER models' performance in privacy preserving framework.

With DP, Without FL: Next, we incorporate a Differentially Private Stochastic gradient descent (DP-SGD) optimizer for training[4]. The performance of both the models with varying noise-multiplier (a hyperparameter to add noise) are presented in Table 2. Note that, by increasing the value of noise-multiplier we add more privacy (lower ϵ) to the NER models. We observe, even if the privacy increases, the BI-LSTM-TDDL model produces stable performance and does not deteriorate much compared to its performance while DP is not present. On the other hand, using DP-SGD, BI-LSTM-CRF architecture performs poorly and fluctuates significantly. We observe a negative signal towards incorporating DP, where model architecture has CRF in the final layer, for tasks like NER. Finding out the reason behind such behavior of CRF-based model and mitigating the problem would be immediate future work. However, for the next two phases of analysis, we continue only with the robust BI-LSTM-TDDL model architecture.

$(n) \rightarrow$	2	5	10	15	20
Precision	0.835	0.804	0.752	0.731	0.740
Recall	0.867	0.845	0.815	0.810	0.801
F-measure	0.834	0.810	0.772	0.763	0.758

Table 5: Performance of BI-LSTM-TDDL model in the FL setup along with DP, with increasing number of clients (n). Noise multiplier = 1, $\epsilon = 0.688$

Without DP, With FL: Next, we analyze the FL setup for NER, where we have one centralized server and n number of the client sites. The training and validation data are divided equally among all the clients. We analyze the framework in two ways. First, we observe the performance variation of the aggregated model on test data with the number of clients, which is presented in Table 3. The

[2]The hyperparameter settings to train those models are as follows: epochs- 10, batch size - 32, learning rate - 0.15, optimizer - Stochastic gradient descent (SGD)

[3]https://spacy.io/models/en#en_core_web_lg

[4]we use Tensorflow Privacy library https://github.com/tensorflow/privacy/blob/master/tensorflow_privacy. The hyperparameter settings: micro-batch size - 1, normalization clip - 1.5

Training Data	Precision		Recall		F-measure	
	mean	sd	mean	sd	mean	sd
20 %	0.767	0.008	0.745	0.033	0.744	0.013
40 %	0.773	0.006	0.738	0.029	0.744	0.013
60 %	0.707	0.102	0.123	0.063	0.171	0.104
80 %	0.812	0.052	0.113	0.172	0.106	0.231
100 %	0.585	0.556	0.137	0.228	0.123	0.269

Table 6: Performance of BI-LSTM-TDDL model in the FL setup along with DP, with increasing % of training data used in each client. Mean and standard deviation (sd) are computed over 5 different simulation. Number of clients = 5, Noise multiplier = 1, $\epsilon = 0.688$.

results are reported only after one communication round, i.e. all the client models are trained with their respective training data (full) and the client models' parameters are transferred to the central server once for aggregation. From the result in Table 3, we see with the increasing number of clients the performance of the centralized NER model decreases. Given that the number of the training sample is constant and those are divided among the clients equally, the observed performance fits our intuitions. As the number of clients increases, the amount of training data per client decreases, and the client models are trained with a smaller amount of training data[5]. Secondly, we observe how the central model behaves if model parameters are shared from clients after training the client models with $x\%$ of the local training data. For our study we keep $x = 10$, which implies client model aggregation and global model update cycle is being executed after each pass of model training with 10 % of local data on the client-side. Since the $x\%$ of data is picked randomly, we simulate independently for 5 times and report the mean and standard deviation (sd) of all metrics. From the results presented in Table 4, we observe that even in such an incremental training scenario we achieve an F-measure of 0.846 with 100% training data, which is comparable to the non-incremental version of training (0.874 as per Table 3). The FL setup for the NER model looks promising in the incremental approach as well, making it suitable even for the scenario while on the client-side a large amount of training data is not available at once.

With DP, With FL: We attempt two different analyses following the same experimental setup as 'Without DP, With FL', but during client-side training, we use DP-SGD instead of SGD as the optimizer. In Table 5, we see that the trend of decreasing F-measure with the increasing number of clients, which is the same as the non-private performance. Note that, when we incorporate DP in FL setup it reduces the F-measure about 1-6 %. The result for the second type of analysis is presented in Table 6. We note that with increasing percentage (%) of training data, the F-measure does not improve, and after some point (60 %) the F-measure drops significantly. DP does not seem promising for incremental setup, while in each phase the training data used is sufficiently small. Note that, in each phase, only $x = 10\%$ training data (approx. 300 data points) is being used for training.

[5]The hyperparameter settings for training client model: epochs - 10, batch size - 32, learning rate - 0.15, optimizer - Stochastic gradient descent (SGD).

4 Conclusion

In this work, we presented an analysis of the behavior of one sequence tagging task namely NER in a federated learning framework along with differential privacy, which is the first-ever attempt of its kind. From the investigation, we observed that with DP-SGD optimizer, the performance of CRF based model tends to decrease significantly in our current experimental setup. In the federated framework, we observed that with the increase of the number of clients with smaller training data, the performance of aggregated models decrease significantly, whereas the performance shoots up when client models are trained in an incremental approach and the incremental models are communicated to the server after each training phase. On the other hand, when a DP-SGD optimizer is deployed in each client training phase, even the incremental training policy works only till 60% of the training data is used and thereafter performance drops. Immediate future work would be to find out the root cause of such a phenomenon and mitigate it, as a combination of privacy measures would be very much desired in privacy-aware application scenarios.

To extend this study, we plan to explore several other NER datasets (even low-resource datasets) to find out more general behavior. Hyper-parameter tuning is another direction of our future work to find out the best performance for each set-up. The broader goal is to build a differentially private federated framework for sequence tagging tasks with compromising performance as little as possible.

Acknowledgements

This research was funded by the German Federal Ministry of Education and Research (BMBF) as part of the HILANO project, ID 01IS18085C.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, Vienna, Austria.

Agencia-Espanola-Proteccion-Datos. 2019. K-anonymity as a privacy measure. https://www.aepd.es/sites/default/files/2019-09/nota-tecnica-kanonimidad-en.pdf.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th international conference on computational linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA.

Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2938–2948, Palermo, Sicily, Italy.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, Dallas, Texas, USA.

Qingrong Chen, Chong Xiang, Minhui Xue, Bo Li, Nikita Borisov, Dali Kaarfar, and Haojin Zhu. 2018. Differentially private data generative models. *arXiv preprint arXiv:1812.02274*.

Prateek Chhikara, Prabhjot Singh, Rajkumar Tekchandani, Neeraj Kumar, and Mohsen Guizani. 2021. Federated learning meets human emotions: A decentralized framework for human–computer interaction for iot applications. *IEEE Internet of Things Journal*, 8(8):6949–6962.

Olivia Choudhury, Aris Gkoulalas-Divanis, Theodoros Salonidis, Issa Sylla, Yoonyoung Park, Grace Hsu, and Amar Das. 2019. Differential privacy-enabled federated learning for sensitive health data. *arXiv preprint arXiv:1910.02578*.

Olivia Choudhury, Aris Gkoulalas-Divanis, Theodoros Salonidis, Issa Sylla, Yoonyoung Park, Grace Hsu, and Amar Das. 2020. Anonymizing data for privacy-preserving federated learning. *arXiv preprint arXiv:2002.09096*.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. 2006. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503, Saint Petersburg, Russia.

Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407.

Suyu Ge, Fangzhao Wu, Chuhan Wu, Tao Qi, Yongfeng Huang, and Xing Xie. 2020. Fedner: Privacy-preserving medical named entity recognition with federated learning. *arXiv preprint arXiv:2003.09288*.

Robin C Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Rui Hu, Yuanxiong Guo, E Paul Ratazzi, and Yanmin Gong. 2020. Differentially private federated learning for resource-constrained internet of things. *arXiv preprint arXiv:2003.12705*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Ahmed Imteaj and M Hadi Amini. 2020. Fedar: Activity and resource-aware federated learning model for distributed mobile robots. *arXiv preprint arXiv:2101.03705*.

Raouf Kerkouche, Gergely Acs, Claude Castelluccia, and Pierre Genevès. 2021. Privacy-preserving and bandwidth-efficient federated learning: An application to in-hospital mortality prediction. In *ACM Conference on Health, Inference, and Learning*, page 25–35, Virtual Event, USA.

Yusuke Koda, Koji Yamamoto, Takayuki Nishio, and Masahiro Morikura. 2020. Differentially private aircomp federated learning with power adaptation harnessing receiver noise. *arXiv preprint arXiv:2004.06337*.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 1273–1282, Fort Lauderdale, Florida, USA.

Suraj Rajendran, Jihad S Obeid, Hamidullah Binol, Ralph D Agostino Jr, Kristie Foley, Wei Zhang, Philip Austin, Joey Brakefield, Metin N Gurcan, and Umit Topaloglu. 2021. Cloud-based federated learning implementation across medical centers. *JCO clinical cancer informatics*, 5:1–11.

Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. 2019. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6):1205–1221.

Yang Zhao, Jun Zhao, Mengmeng Yang, Teng Wang, Ning Wang, Lingjuan Lyu, Dusit Niyato, and Kwok-Yan Lam. 2020. Local differential privacy based federated learning for internet of things. *IEEE Internet of Things Journal*, Early Access.