FedParsing: a Semi-Supervised Federated Learning Model on Semantic Parsing

Anonymous ACL submission

Abstract

Although many semantic parsing models have been proven to work effectively on “NL-to-SQL”, the limitation of annotated datasets remains a great challenge. In many semi-supervised models, while they use unlabeled data to greatly improve the model accuracy, they fail to take data privacy of users into account . In this work, we focus on improving the performance of the semantic parsing model and protecting the users’ data privacy without increasing the size of the labeled dataset. Our new model, which is named FedParsing, is a semi-supervised Federated Learning model. In order to solve the difficulty on convergence of traditional semi-supervised Federated Learning model, we incorporate the Mean Teacher algorithm and apply the Exponential Moving Average algorithm to update model parameters. Experiments on WikiSQL show that with extra unlabeled data, our model performs better than supervised training model and traditional semi-supervised Federated Learning model, which proves the effectiveness of FedParsing model.

1 Introduction

During the developments of artificial intelligence, the interactions between human and machine in daily life becomes more frequent. Semantic parsing serves as a key technology in helping the machine to understand human’s languages by translating a natural language query to logic forms, which is known as “NL-to-SQl”(Pal et al., 2020; Brunner and Stockinger, 2021) or “Text-to-SQL”(Elgohary et al., 2020; Yu et al., 2018). Many important works have emerged in this field(Jia and Liang, 2016; Dong and Lapata, 2016). There exists a long-standing problem, i.e., available labeled data is always limited, which brings great difficulties to improve the accuracy of semantic parsing. In order to solve this problem, the idea of semi-supervision by using the unlabeled data has been introduced(Yin et al., 2018; Qi et al., 2020; Jia et al., 2020).

However, most of the unlabeled data come from users. Due to user privacy, it is impossible to obtain data directly from users. Federated Learning(McMahan et al., 2017) is proposed to enable server provider to train models separately and also protect user privacy. Without obtaining users’ private unlabeled data, it allows the server and clients to train the same model jointly by exchanging model parameters or gradients. Thus, the integration of Federated Learning and semi-supervised learning becomes the key to solve the aforementioned problem.

In previous work(Bettini et al., 2021; Bian et al., 2021), the server is trained in a supervised manner, while the clients are usually trained in an unsupervised manner. Therefore, the problem of model convergence remains great challenges. Firstly, since each client has different usage pattern, the data generated between clients follow a different distribution(Zhao et al., 2018; Sattler et al., 2019), which incurs biases to model updates. Aggregating these divergent models can slow down the global model convergence. Secondly, the objective function difference makes the model much more hard to converge. One way to solve the above problem is to unify the objective function of the client and server.

Aiming at this problem, we introduce Mean Teacher algorithm(Tarvainen and Valpola, 2017) into Federated Learning, called FedParsing Model. The main idea of FedParsing Model is to set up two models on the client side, one is the student model, the other is the teacher model. The student model conducts semi-supervised training with the prediction of the teacher model as the standard labels, instead of unsupervised training. Also, we apply Exponential Moving Average (EMA) algorithm(Haynes et al., 2012) to update the parameters of the teacher model, so as to control the gradient deviation of the student model in a limited range. We train our model on WikiSQL(Zhong et al., 2017), and the results show that the accuracy of our model is improved compared with the baseline.

To conclude, there are three innovations in this work:

- We apply Federated Learning to semantic parsing which is called FedParsing Model. We complete the model training under the premise of protecting user privacy.
- We integrate Mean Teacher model and Federated Learning to realize semi-supervised training on the client side, so as to solve the problem of inconsistent goals.
- We further propose the EMA algorithm to constrain the gradient deviation of clients and verify the effectiveness of this algorithm.

2 Problem Formulation

Our goal is to improve the performance of the semantic parsing model by fully exploiting the unlabeled dataset without increasing the size of the labeled dataset or violating the clients' data privacy. For the labeled data, we combine a sentence $q = \{q^1, q^2, ... , q^n\}$ and its corresponding logical form $c = \{c^1, c^2, ..., c^n\}$ into input sentence pair of form $Concat(q, c)$, where $Concat()$ is simply merging text and separating it with commas. These input sentence pairs are constructed into Labeled Semantic Parsing Datasets $L = [q^n, c^n]_{n=1}^{N}$ which is held by the server. For data without labels, we simply use question texts $q = \{q^1, q^2, ... , q^m\}$ as the input sentences which constitute Unlabeled Datasets $UL = [q^m]_{m=1}^{M}$ held by the clients.

3 Methods

3.1 Mean Teacher and EMA algorithm

The traditional unsupervised learning on the client side aims to transfer the sentences(words) into sentences(words), for instance, Encoder-Decoder Model or Masked Language Model(Salazar et al., 2020). It is inconsistent with the goal of supervised learning on the server side which aims to transfer sentences into SQL. Subsequently, this inconsistency leads to the difference of objective function, which makes the model much harder to converge. Aiming at this problem, we propose to use two models for training on the client side, one named Student Model and the other named Teacher Model. In the initial stage, these two models are given the same parameters. However, they are trained separately. Teacher Model generates the target annotation $p = [p_1, p_2, ..., p_n]$ for each unlabeled natural language statement x, where p_i are the logical predicates. Student Model progressively generats logical forms $y^* = [y^*_1, y^*_2, ..., y^*_n]$ by treating p as the annotation result, where y^*_i are logical predicates. Then the Student Model updates its weight by using the gradient descent algorithm. We define the consistency loss $L(\theta)$ as the expected distance between p_i and y^*_i:

$$L(\theta_i) = E_{x_i}\left[||p_i(x) - y^*_i(x)||^2 \right] \quad (1)$$

As for the Teacher Model, we propose Exponential Moving Average algorithm to modify the parameters in each epoch. The formula is:

$$\theta'_t = \lambda\theta'_{t-1} + (1 - \lambda)\theta_t \quad (2)$$

whereas θ_t and θ'_t represent the weight of Student model and Teacher model at training step t. λ is an attenuation hyperparameter which controls the updating speed of the model, usually 0.9 or 0.99. Therefore, when Student Model updates the weights, Teacher Model improves in turn so that it can continuously produce better target prediction. Besides, using Mean Teacher algorithm to conduct semi-supervision on the client side, aligns the objective function with the server side. This consistency effectively promotes the convergence of the model.

3.2 Semi-Supervised Federated Learning on Semantic Parsing

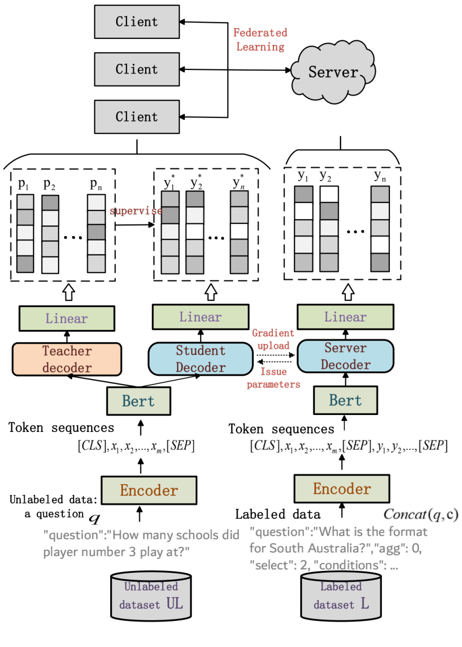

Figure 1: The Framework of FedParsing Model.

As shown in Figure1, the FedParsing Model is composed of an Encoder, Bert pretraining layer, three decoders based on attention mechanism and linear layer. The decoders include Teacher Decoder, Student Decoder and Server Decoder.

Combining the Mean Teacher and EMA algorithm, the server and clients train their models as the following steps.

Step 1: At communication round t, we train Server Decoder in a normal supervised manner.The loss function is defined by the following formula:

$$L_S(\theta_E, \theta_{SD}) = \frac{1}{N}\sum_{n=1}^{N} -logP(y^n|x^n, \theta_E, \theta_{SD}) \tag{3}$$

where θ_E is the parameter of the Encoder and θ_{SD} is the parameter of Server Decoder. After training, the global parameter w_s^t is generated and distributed to K selected clients.

Step 2: Each client has a Student Decoder and a Teacher Decoder.Teacher Decoder is trained by the global parameter w_s^t while Student Decoder is not. Student Decoder is updated according to the MSEloss in Eq.1. Instead of sharing the weights with Student Decoder, Teacher Decoder uses the EMA weights of Student Decoder according to Eq.2. In this way, Teacher Decoder together with Server Decoder, plays the role of teacher model to limit the divergence of Student Decoder. Once the training is completed, the parameter $w_{k,st}$ of client k's Student Decoder will be sent to the server.

Step 3: Upon receiving K clients' parameters, the server aggregates the gradient changes using the global parameter w_s^t and clients' parameters $w_{k,st}$.

$$w_{avg}^{t+1} = (w_s^t + \sum_{k=1}^{K} w_{k,st}^t)/(K+1) \tag{4}$$

Then Server Decoder will be trained in a supervised way using w_{avg}^{t+1}. After re-selecting K clients, the new generated w_s^{t+1} is distributed to the clients again. Subsequently, Step 2 is repeated.

The FedParsing Model+EMA algorithm can make the model converge more quickly than the original Federated Learning model. The reason is that accurate target labels generated by Teacher Decoder will lead to a faster feedback loop between the clients and server, which results in a better test accuracy. In the mean time, the FedParsing Model only transmits the gradients between server and clients, which fully protects users' data privacy.

4 Experiments

4.1 Experimental Setup

Our model is evaluated on the dataset WikiSQL, which was first presented by Zhong et al. (2017). We use RoBERTa for pre-training in a batch of 10. The inputs generation is the same as Lyu et al. (2020). Eventually, we get 36063 data for training, 10107 data for testing and 5340 data for development. Then we allocate the training data proportionally to one server and 1,000 clients. The training data S obtained by server is chosen from the set {500,1000,10000}. Each client will be given a number of training data in the range of [20,40] randomly. The communication round between the server and clients is R. In each round, K clients are selected and trained for E epochs.In this experiment, we set $E = 3$. The exponential moving average decay value in each epoch is set to be $\lambda = 0.99$. Both the server and the clients use SGD as the model optimizer.

Four groups of experiments are carried out. In the first group, the server is trained in a supervised manner without clients, called HydraNet(Lyu et al., 2020). In the second group, compared to HydraNet, the clients are trained in unsupervised Mask Language Model(MLM). The other two groups use FedParsing Model and FedParsing Model+EMA. The accuracies of this task are evaluated by seven indexes,i.e.,$Overall$,Agg,Sel,Wn,Wc,Op and Val. The index $Overall$ is the proportion that all the predictions are right. The indexes Agg, Sel, Wn, Wc, Op and Val stand for tasks of aggregation operator, predicting select column, number of conditions, where columns, where operators and where values, respectively.

4.2 Experimental Results

Review. The results of Development and Test accuracy are shown in Table 1. The accuracy of the Test set is on the left while the Development set on the right. Compared to HydraNet, the accuracies of HydraNet+MLM significantly reduce, which validates the problem of model divergence caused by inconsistent objectives. Overall, the FedParsing+EMA model performs the best. When S=1000, its accuracy is up to 1.9% higher than HydraNet and 17.6% higher than HydraNet+MLM. This shows the effectiveness of FedParsing Model. Moreover, the accuracy of the FedParsing+EMA model is up to 0.8% above the FedParsing Model, which verifies the effectiveness of EMA algorithm.

Model	**Accuracy**						
S=500	Overall	Agg	Sel	Wn	Wc	Op	Val
HydraNet	69.8 70.1	87.3 87.4	95.3 95.7	96.4 96.9	87.9 88.4	97.5 97.6	91.3 91.6
HydraNet+MLM	50.1 50.5	72.2 72.5	94.5 94.8	87.3 87.8	77.6 77.3	95.5 95.7	89.4 89.6
FedParsing	70.2 70.4	88.2 88.0	95.2 95.7	96.4 96.9	87.2 88.0	97.9 97.8	91.4 91.2
FedParsing+EMA	70.7 70.4	**88.7 88.8**	95.9 96.2	96.9 96.9	88.0 88.0	97.9 97.8	91.6 91.5
S=1,000	Overall	Agg	Sel	Wn	Wc	Op	Val
HydraNet 1	72.6 72.3	88.6 88.2	96.3 96.3	96.0 96.7	89.1 89.7	98.2 98.1	92.5 92.3
HydraNet+MLM	56.6 57.3	77.9 76.6	95.5 95.7	92.7 93.4	83.9 84.5	94.7 95.0	89.6 89.0
FedParsing	73.4 73.7	88.5 88.3	96.2 96.2	96.7 97.3	89.4 90.1	98.3 98.3	93.5 93.5
FedParsing+EMA	**74.2 74.2**	89.2 89.1	96.3 96.2	96.9 97.4	89.8 90.3	98.3 98.3	93.6 93.7
S=10,000	Overall	Agg	Sel	Wn	Wc	Op	Val
HydraNet 1	79.1 79.2	90.2 90.2	97.1 97.4	97.6 98.0	93.1 93.3	98.6 98.6	95.4 95.6
HydraNet+MLM	79.1 79.2	90.2 90.2	97.1 97.4	97.6 98.0	93.1 93.3	98.6 98.6	95.4 95.6
FedParsing	79.1 79.2	90.2 90.2	97.1 97.4	97.6 98.0	93.1 93.4	98.6 98.6	95.4 95.7
FedParsing+EMA	**79.2 79.3**	90.2 90.2	97.1 97.4	97.6 98.0	93.1 93.4	98.6 98.6	95.5 95.7

Table 1: The Development and Test accuracy when server gets 500, 1,000 and 10,000 data respectively.

Impact of S when K=10. As can be seen from Table 1, FedParsing+EMA model performs the best when S=1000. When S=500 or 10000, FedParsing+EMA model does not perform outstandingly but still well. The reason may be that when S is too small, the server has the same weight as the clients. So the gradient of supervised model does not affect the average gradient effectively. When S is too large, supervised training on server is enough to get good results. The role of the clients becomes trivial. Therefore, it is important to keep a suitable balance between server and clients data.

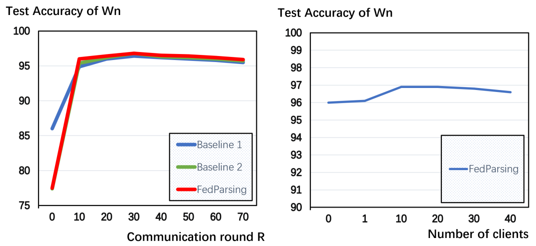

Figure 2: Diagram of how the accuracy index Wn varies with R and K.

Impact of R on the accuracy index Wn. We study the impact of the communication round R when the server gets 500 data. Figure2 shows that, in the initial stage, the accuracy rate will increase across the communication rounds. However, when the model converges, the accuracy rate will slightly decrease since the model may over-fit when R is too large.

Impact of K on the accuracy index Wn. We study the impact of the selected clients number K when the server gets 1,000 data. Figure2 shows that the accuracy index Wn rises steadily until it reaches a peak at $K = 10$ and then it drops slightly. It is speculated that when there are too many clients, their divergences will affect the average gradient, resulting in a decline in accuracy.

5 Related Work

Semi-supervised Federal Learning has been proven valid in multiple scenarios. Bettini et al. (2021) verified its effectiveness on action recognition. Wang et al. (2020a) proposed GraphFL for semi-supervised node classification on graphs. Itahara et al. (2020) proposed a distillation-based semi-supervised FL algorithm which achieved higher classification accuracy. Different from these work, semantic parsing is a generation problem, which has a larger problem space than classification problems. Thus, the convergence of model is more difficult.

6 Conclusion

The integration of Federated Learning and semi-supervision is an effective method to solve the problem of user data privacy leakage on semantic parsing task. However, traditional semi-supervised Federated Learning has the problem of model convergence. Aimed at this problem, we propose FedParsing Model by using Mean Teacher and EMA algorithm. FedParsing Model is tested on WikiSQL dataset and experimental results prove the effectiveness of this model.

References

Claudio Bettini, Gabriele Civitarese, and Riccardo Presotto. 2021. Personalized semi-supervised federated learning for human activity recognition. *CoRR*, abs/2104.08094.

Jieming Bian, Zhu Fu, and Jie Xu. 2021. Fedseal: Semi-supervised federated learning with self-ensemble learning and negative learning. *CoRR*, abs/2110.07829.

Ursin Brunner and Kurt Stockinger. 2021. Valuenet: A natural language-to-sql system that learns from database information. In *37th IEEE International Conference on Data Engineering, ICDE 2021, Chania, Greece, April 19-22, 2021*, pages 2177–2182. IEEE.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Ahmed Elgohary, Saghar Hosseini, and Ahmed Hassan Awadallah. 2020. Speak to your parser: Interactive text-to-sql with natural language feedback. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2065–2077. Association for Computational Linguistics.

Alireza Fallah, Aryan Mokhtari, and Asuman E. Ozdaglar. 2020. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

David Haynes, Steven M. Corns, and Ganesh Kumar Venayagamoorthy. 2012. An exponential moving average algorithm. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2012, Brisbane, Australia, June 10-15, 2012*, pages 1–8. IEEE.

Sohei Itahara, Takayuki Nishio, Yusuke Koda, Masahiro Morikura, and Koji Yamamoto. 2020. Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. *CoRR*, abs/2008.06180.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Zixia Jia, Youmi Ma, Jiong Cai, and Kewei Tu. 2020. Semi-supervised semantic dependency parsing using CRF autoencoders. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6795–6805. Association for Computational Linguistics.

Qin Lyu, Kaushik Chakrabarti, Shobhit Hathi, Souvik Kundu, Jianwen Zhang, and Zheng Chen. 2020. Hybrid ranking network for text-to-sql. *CoRR*, abs/2008.04759.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR.

Debaditya Pal, Harsh Sharma, and Kaustubh Chaudhari. 2020. Data agnostic roberta-based natural language to SQL query generation. *CoRR*, abs/2010.05243.

Qi Qi, Xiaolu Wang, Haifeng Sun, Jingyu Wang, Xiao Liang, and Jianxin Liao. 2020. A novel multi-task learning framework for semi-supervised semantic parsing. *IEEE ACM Trans. Audio Speech Lang. Process.*, 28:2552–2560.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2699–2712. Association for Computational Linguistics.

Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. 2019. Robust and communication-efficient federated learning from non-iid data. *CoRR*, abs/1903.02891.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

Binghui Wang, Ang Li, Hai Li, and Yiran Chen. 2020a. Graphfl: A federated learning framework for semi-supervised node classification on graphs. *CoRR*, abs/2012.04187.

Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li. 2020b. Optimizing federated learning on non-iid data with reinforcement learning. In *39th IEEE Conference on Computer Communications, INFOCOM 2020, Toronto, ON, Canada, July 6-9, 2020*, pages 1698–1707. IEEE.

Pengcheng Yin, Chunting Zhou, Junxian He, and Graham Neubig. 2018. Structvae: Tree-structured latent variable models for semi-supervised semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 754–765. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir R. Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3911–3921. Association for Computational Linguistics.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated learning with non-iid data. *CoRR*, abs/1806.00582.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103.

A Related Work

The phenomenon that the distribution of each client can be totally different, is known as non-IID (not identically and independently distributed) problem, which can cause severe model divergence. For non-IID problem, Zhao et al. (2018) proposed the Data Sharing method by building a globally shared dataset G for clients; Fallah et al. (2020) proposed the method of a shared initial model in a distributed manner; Wang et al. (2020b) proposed the FAVOR model, which utilizes a Deep Q-network to intelligently select clients to participate in training in each round of communication. Although the above method solves the non-IID problem significantly, it also leads to the leakage of users' privacy to some extent.

B Method

For labeled data, tokens take the form of: $[CLS], x_1, x_2, ..., x_m, [SLP], y_1, y_2, ..., y_n, [SLP]$, where $x_1, x_2, ..., x_n$ is the token sequence of question q and $y_1, y_2, ..., y_n$ is the token sequence of c. For unlabeled data, tokens take the form of: $[CLS], x_1, x_2, ..., x_m, [SLP]$. Subsequently, these token sequences will be decoded by a pre-trained Transformer model, such as Bert or Roberta, to obtain vectors as the final input of the model.

C Experiment

When S=500, compared to the HydraNet, the accuracies of FedParsing+EMA model mostly increase $0\% \sim 1.4\%$. Occasionally, some accuracies are worse than the HydraNet, such as Val and Wc. The reason may be that the data volume of server is almost the same as that of clients. Therefore, the weight of server is not dominant in gradient aggregating. Thus, the gradient of supervised model will not affect the average gradient effectively.

If the server is allocated 10,000 data for training, the Overall accuracy of FedParsing+EMA is 8.9% higher than the situation where the server gets 500 data. Other accuracies increase by 1.2%∼5.2%. These results are reasonable since the increase in the amount of labeled data can help the model trained by the server becoming more accurate and the weight of server becoming larger in gradient aggregating. Thus, the overall accuracy becomes higher. However, when the server gets enough data, the accuracy of our model is closer to the HydraNet. The reason may be that the gradient changes generated by the client are not enough to affect the average gradient. In the extreme case, when the server has the entire labeled training dataset, the situation will transfer to a supervised learning model.