PTab: Using the Pre-trained Language Model for Modeling Tabular Data

Guang Liu [1*], **Jie Yang** [1 2 *], **Ledell Wu** [1 †]

[1] Beijing Academy of Artificial Intelligence, Beijing, China
[2] Beijing Jiaotong University
liuguang@baai.ac.cn, 21120149@bjtu.edu.cn, wuyu@baai.ac.cn

Abstract

Tabular data is the foundation of the information age and has been extensively studied. Recent studies show that neural-based models are effective in learning contextual representation for tabular data. The learning of an effective contextual representation requires meaningful features and a large amount of data. However, current methods often fail to properly learn a contextual representation from the features without semantic information. In addition, it's intractable to enlarge the training set through mixed tabular datasets due to the difference between datasets. To address these problems, we propose a novel framework PTab, using the Pre-trained language model to model Tabular data. PTab learns a contextual representation of tabular data through a three-stages processing: Modality Transformation(MT), Masked-Language Fine-tuning(MF), and Classification Fine-tuning(CF). We initialize our model with a pre-trained Model (PTM) which contains semantic information learned from the large-scale language data. Consequently, the contextual representation can be learned effectively during the fine-tuning stages. In addition, we can naturally mix the textualized tabular data to enlarge the training set to further improve the representation learning. We evaluate PTab on eight popular tabular classification datasets. Experimental results show that our method has achieved a better average AUC score in supervised settings compared to the state-of-the-art baselines(e.g. XGBoost), and outperforms counterpart methods under semi-supervised settings. We present visualization results that show PTab has well instance-based interpretability.

1 Introduction

Tabular data is essential for modern daily life and has been widely studied. It's the core component of many industrial applications, e.g., risk management of credit cards and insurance underwriting. In the last decades, researchers from various fields have studied effective ways of modeling tabular data. As far as we knew, the tree-based methods, e.g. XGBoost (Chen and Guestrin 2016), have dominated this field due to their superior performance on feature extraction.

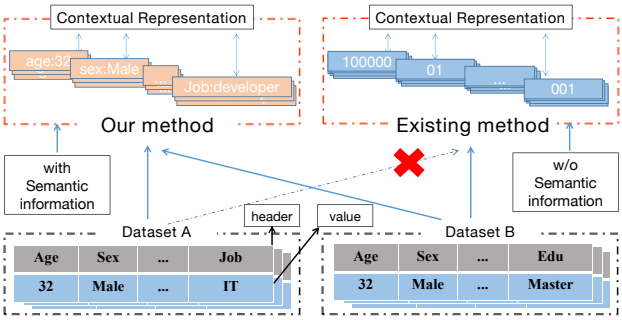


Figure 1: The illustration of methods on tabular modeling. The red folk means the mixing of datasets is not feasible. Each column in the examples is a field, and it contains a header and its values.

Recently, many neural-based methods have been proposed to learn contextual representations from tabular data (Huang et al. 2020; Yoon et al. 2020; Arik and Pfister 2021). Similar to the popular Masked Language Model (MLM) in Nature Language Processing (Devlin et al. 2018), TabTransfomer (Huang et al. 2020) tries to learn a contextual representation by masking/corrupting the values of some random fields and predicting/recovering them based on the context. It uses a multi-layer transformer to handle the value of categorical fields and a Multi-Layer Perceptron(MLP) to handle the value of numerical fields. SAINT (Somepalli et al. 2021) proposes projecting the value of numerical fields into an embedding, and uses a unified transformer to handle both numerical and categorical fields. In addition, it adopts multiple self-supervised learning tasks and data augmentation to further improve the contextual representation. These methods have achieved promising improvement compared to tree-based methods in semi-supervised settings. Intuitively, to lean an effective contextual representation requires a large amount of data and meaningful features.

However, current methods often fail to properly learn a contextual representation from the features without semantic information. Besides, it's intractable to enlarge the training set through mixed tabular datasets due to the difference between datasets. Existing methods, such as XGBoost (Chen and Guestrin 2016) and TabTransfomer (Huang et al. 2020), often lead to information loss during the feature transfor-

*These authors contributed equally.
†corresponding author

mation. As shown in Fig 1, Information technology (IT) is semantically close to engineers before the transformation but may be very distant after the transformation. Under such a condition, the current neural-based method often achieves under-performance compared to the tree-based methods (Grinsztajn, Oyallon, and Varoquaux 2022). Additionally, the values of numeric fields tend to be in an ambiguous situation. For example, the value of age and shoe size both can be 32, and they may be transformed into identical representations. The transformation process ignores the headers for the values which may help to improve the contextual representation. Without the information in the header, we need more data to learn a precise contextual representation. Noticeably, the mixing of different datasets to enlarge the size of the training set is not practical in the tabular domain. The difference in headers and values between different datasets makes it harder for the neural-based methods to learn a proper contextual representation. So, the technique that can model the tabular data with semantical information and mixed datasets is needed to be explored.

To address these problems, we propose a novel framework, using the Pre-trained language model to model Tabular data (PTab), to enhance the contextual representation of tabular data with semantic information and makes the training on mixed datasets feasible. PTab consists of three stages: Modality Transformation(MT), Masked-language fine-tuning(MF) and Classification fine-tuning(CF). MT stage is fundamental to the framework. It textualizes the tabular data to enrich its semantical feature and diminish the difference between different tabular datasets. To this end, it uses a simple lexicon and an ordered syntax to describe the headers and values in a sample of tabular data. The information that describes the meanings of the field can be used easily, e.g., the header information of each fields. Consequently, we can utilize the knowledge in the PTM to enhance the learned representation through the MF stage. CF stage adjusts the learned contextual representation for the downstream task. It's worth mentioning that textualization diminishes the difference between datasets. By doing so, the model training on mixed textualized tabular datasets can be realized smoothly. The tabular datasets from the similar domains can be used to improve representation learning for the corresponding downstream task. The major contributions of this paper are summarized as follows:

- We propose a novel framework PTab, using the Pre-trained language model to model Tabular data. PTab can improve the contextual representation learned from tabular data by incorperate extra semantical information.
- We evaluate PTab on eight binary classification datasets. The extensive experimental results shows that PTab outperforms the state-of-the-art baselines, including strong tree-based baseline XGBoost and GBDT, in terms of average AUC score in both supervised/semi-supervised settings.
- We firstly propose a feasible way for training on mixed tabular datasets. The results of mixed dataset training demonstrate that PTab can get higher classification performance by training on mixed textualized tabular datasets in similar domains, and vice versa.
- Interestingly, the visualization results demonstrate that PTab trained model has well instance-based interpretability for feature importance and feature semantical similarities, while the tree-based method is hard to achieve.

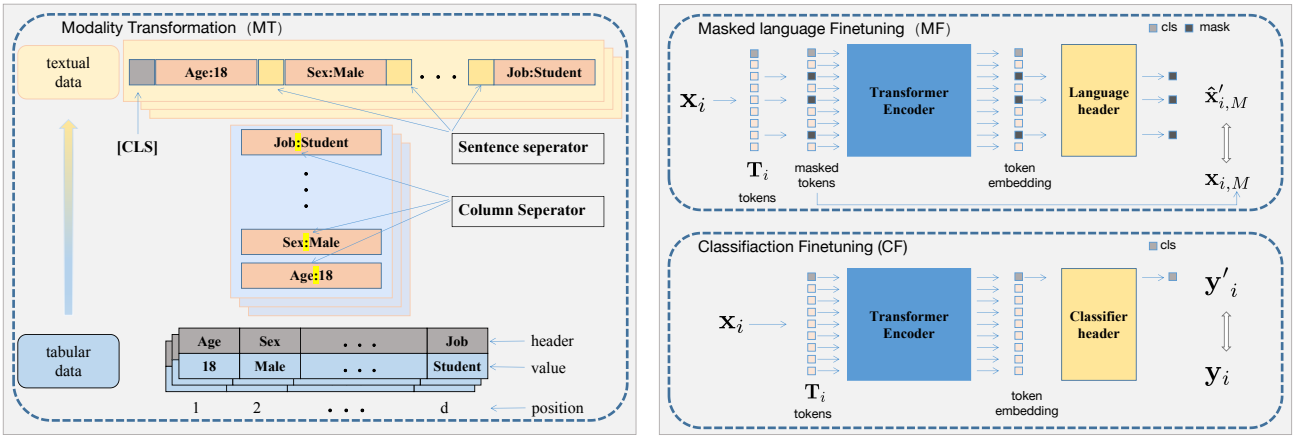
2 Related works

Tabular data modeling is the fundamental task in financial and insurance domains. Many researchers focus on extracting features and building models from tabular data (Gorishniy et al. 2021; Borisov et al. 2021; Shwartz-Ziv and Armon 2022; Gorishniy et al. 2021). These studies can be divided into numerical-based methods and text-based methods.

Numerical-based methods mainly focus on using neural-based methods to learn contextual representations from numerical tabular data (Gorishniy et al. 2021; Shwartz-Ziv and Armon 2022; Yoon et al. 2020). TabNet (Arik and Pfister 2021) firstly proposes using self-supervised learning to learn the contextual representation. SAINT (Somepalli et al. 2021) uses data augmentation and multi-task learning to enhance the learning process. Subtab (Ucar, Hajiramezanali, and Edwards 2021) turns the task of learning from tabular data into a multi-view representation learning problem by dividing the input features into multiple subsets. This line of research usually uses contrastive learning to learn a representation from unlabeled numerical data and then fine-tuned on labeled numerical data. However, such methods often suffer from a shortage of labeled data (Xu et al. 2019). Without enough labeled data, it's intractable to learn an effective representation.

Textual-based methods mainly focus on learning table representation for table-related tasks (Iida et al. 2021; Deng et al. 2022; Chen et al. 2019). Different from numerical-based methods, textual-based method tries to encode multi-rows of tabular data into one representation. Table2vec (Zhang, Zhang, and Balog 2019) considers different table elements, such as caption, field headers and values, for training word and entity embeddings. Tabert (Yin et al. 2020) jointly learns representations for natural language sentences and semi-structured tables. TURL (Deng et al. 2022) captures factual knowledge in relational tables. TUTA (Wang et al. 2021) designs a bi-dimensional tree to define value coordinates and value distance in generally structured tables. These methods mainly focus on specific downstream tasks in Natural Language Processing (NLP). Few works have explored the way to model one row of tabular data as numerical-based methods.

Accordingly, both numerical- and textual-based methods learn contextual representation contrastively: mask/corrupt some elements in a sample and predict/recover them by their context. For example, TabTransfomer (Huang et al. 2020), TabNet(Arik and Pfister 2021), and SAINT(Somepalli et al. 2021) all learns tabular representation by applying the Masked Language Model (MLM) to categorical fields. Tabbie (Iida et al. 2021) and TUTA (Wang et al. 2021) also use MLM for textual tabular data representation learning. And MLM-like method works well on textual tabular data and often not on numerical tabular.



(a) Modality transformation (b) Training process

Figure 2: The framework of our proposed method

3 Task formulation

For a given n samples tabular dataset, $\{\mathbf{X}_i, y_i\}, \mathbf{X}_i \in \mathbb{R}^D, i \in [1, n]$, we formulate the tabular data modeling as follow,

$$f_\theta(\mathbf{X}_i; \mathbf{H}) = y_i \,. \tag{1}$$

Here, f_θ is a model with parameter θ, the i_{th} row of tabular data $\mathbf{X}_i$ contains D columns, $\mathbf{X}_i \sim \{x_0, x_1, \cdots, x_D\}$,$\mathbf{H} \sim \{h_0, h_1, \cdots, h_D\}$ represent the shared header descriptions for all $\mathbf{X}$, y_i is the category label. Noticeable, the H provides extra information for tabular data modeling.

4 Methodology

The proposed PTab is a three-stage framework to model textualized tabular data with the pre-trained model, as illustrated in Fig. 2. The first stage is Modality Transformation (MT) which textualizes the tabular data with a simple lexicon and ordered syntax. The second stage is Masked language fine-tuning(MF) which performs a randomly masking and predicting process to learn a contextual representation from textualized tabular data. The third stage is Classification fine-tuning(CF) which adjusts the learned contextual representation for a specific task.

4.1 Modality transformation

As shown in Fig 2(a), the MT stage textualizes the tabular dataset with a simple lexicon and ordered syntax. This is a simple and effective way to pull different tables into the same semantic space for representation learning (Deng et al. 2022). This transformation has two major steps, phrase and sentence construction.

For phrase construction, we simply join the header h_i and value x_i of i_{th} column with a column separator : as a phrase $p_i = \{h_i : x_i\}$. For sentence construction, we join the phrases with sentence separator $[SEP]$ by their appearance order in the row. In the end, we get the sentence $S_i = \{p_0, p_1, \cdots, p_d\}$ for the tabular data $\{X_i, H\}$. Recently, textual-based method unified-SKG (Xie et al. 2022) uses " " to as column seperator and | as columns separator. We find out that the choice of column separator makes an insignificant difference in the final performance.

4.2 Masked language fine-tuning

In the MF stage, the sentence S of textualized X is fed into a pre-trained model for contextual representation learning. For example, the tabular data $X = \{32, Male, driver\}$ with headers $H = \{Age, Sex, Job\}$ can be transformed into a textual sentence. So, the X is translated into a sentence $S = \{Age : 32[SEP]Sex : Male[SEP]Job : driver\}$. Before contextual learning, we tokenize the sentence S_i into tokens $T = \{t_0, t_1, \cdot, t_k\}$. For the final task tuning, we add the $[CLS]$ token at the beginning of each sentence. Following, we randomly mask a certain ratio of tokens with $[MASK]$. Then, we train the model to predict the masked

Table 1: The statistics of all dataset.

Dataset	BM	AD	IN	OS	ST	BC	SB	QB
Num of examples	45211	34190	32561	12330	10000	7043	2583	1055
Num of Columns	16	24	14	17	11	20	18	41
Num of Textual Category[1]	8	0	8	3	3	16	4	0
Domain	business	income	income	shopping	business	business	geology	chemistry
Posivie ratio(%)	11.7	85.4	24.1	15.5	20.4	26.5	6.6	33.7

[1]The textual category means the value of the feature is a textual phrase or sentence.

token. The learning target can be formalized as follows,

$$L_{mf}(X_M|X_{-M}) = \frac{1}{K}\sum_{k=1}^{K}\log p(x_{m_i}|X_{-M};\theta)\,. \quad (2)$$

Here, $M = \{m_0, m_1, \cdots, m_K\}$ denotes the masked tokens indexes in sentence X, K is the number of masked tokens. Similar to BERT (Devlin et al. 2018), the MF stage aims to predict the masked token depend on corresponding contents. By doing so, the contextual representation can be properly learned.

Training on mixed datasets Noticeably, the MT stage can be performed on mixed textualized tabular datasets. Different from the current methods in modeling tabular data, we perform contextual representation learning on textualized tabular data. Similar to textual-based methods, different datasets can be fed into a PTM for contextual representation learning (Yin et al. 2020). In textual modality, the difference between tabular datasets degenerates as the difference between sentences. We can treat the mix of data from the different datasets as the mix of sentences from different domains. Intuitively, the sentence from the same domain will benefit the representation learning(see Section 6.3).

4.3 Classification fine-tuning

In CF stage, we fine-tuning the model on classification tasks. This stage can adjust the learned contextual representation for the specific task. The model is initialized by the model with the lowest loss in MF stage, following the setting in previous work (Hu et al. 2019). We use the embedding of $[CLS]$ token to represent the sentence. On top of the sentence representation, we add a randomly initialized linear projection layer. The output of projection layer is the final predictions. The target of CF stage is to learn a function from S_i to y_i as follows,

$$f(S_i)'_\theta = y_i\,. \quad (3)$$

Here, the learning target is a cross-entropy loss on two classes $1/0$, the model f'_θ has different linear projection layers with f_θ. The semantic of the sentence representation will be changed gradually towards the target of the task.

5 Experimental settings

5.1 Data

As shown in Tab. 1, we evaluate our method on eight classification datasets, which are used in recent work (Huang et al. 2020). Bank Marketing **(BM)** is related to direct marketing campaigns (phone calls) of a Portuguese banking institution. 1995 INcome **(IN)** aims to determine whether a person makes over 50K a year. Online Shoppers **(OS)** aim to detect users who browse products online will eventually choose to buy. Seismic Bumps **(SB)** describe the problem of high energy seismic bumps forecasting in a coal mine. ShruTime **(ST)** contains details of a bank's customers and the target variable is a binary variable reflecting the fact whether the customer left the bank (closed his account) or he continues to be a customer. Qsar Bio **(QB)** aims to classify 1055 chemicals into 2 classes (ready and not ready biodegradable). BlastChar **(BC)** aims to predict behavior to retain customers. These datasets are collected from different domains. **BM**, **ST**, and **BC** are from the business domain. **SB** is from the mining industry. **BC** is from the field of chemistry. Besides, the data have different ratios of category features. Most datasets have both numeric features and category features, except **QB** and **AD**. We evaluate our method on each dataset under five-fold cross-validation. In each fold, the split ratio is 65/15/20%, following the settings in TabTransfomer for a fair comparison.

5.2 Baseline models

We compare our method with six baseline models: 1)logistic regression model (LR), the tree-based model 2)XGBoost (Chen and Guestrin 2016) and 3)GBDT (Friedman 2001)', the neural-based model 4)TabNet (Arik and Pfister 2021), 5) TabTransfomer (Huang et al. 2020),and 6)SAINT (Somepalli et al. 2021). For fair comparison, we empirically tune the key hyperparameters for the baseline model seperately and choose the best resutls. Specially, we apply a greed search for XGBoost on number of trees $\in [500, 600, \cdots, 1000]$, depth $\in [4, 8, 16]$ and learning rate $\in [0.05, 0.1, 0.15, 0.2]$.

5.3 Settings

In Modality Transformation (MT) stage, we simplify the header information and category values to avoid exceeding the maximun length of 512. The data are textualized, tokenized and padded into the same length. In Masked language Fintuning(MF) stage, we set the mask ratio to 15%, and train the model 10 epochs with batch size 16 and learning rate $4e^{-5}$. We evaluate the model on validation set at the end of each epoch and select the best model as the initial weights for Classification fine-tuning(CF). In CF stage, we train our model for 40 epoches with batch size 16 and learning rate $2e^{-5}$. The one with the best auc score on validation set is selected for the final testing. We report the mean and variation of AUC score in five-fold cross-validation. We choose the base version BERT (Devlin et al. 2018) as our initial model which has 12 attention layers, 12 attention heads and 768 hidden state size. In MF stage, the language head of BERT is added on top of the model to get the final vocabulary size predictions for contextual representation learning. In CF stage, we replace the language head with a random initialized classification head to adjust the $[CLS]$ embedding for better adaption to classification. In both supervised and semi-supervised settings, we use a linear learning rate warmup (He et al. 2016) in MF stage.

6 Results and analysis

6.1 Supervised learning

To evaluate our proposed method of contextual representation learning, we compare our model with six baseline models. As shown in Tab 2, our model outperforms all baselines in terms of average AUC scores. For example, we achieve a 4% improvement in average score compared to SAINT. Different from SAINT, our model uses a pre-trained model as

Table 2: The results on full-size dataset.

Dataset	BM	AD	IN	OS	ST	BC	SB	QB	AVG
TabNet*	0.885±0.017	0.663±0.016	0.875±0.006	0.888±0.020	0.785±0.024	0.816±0.014	0.701±0.051	0.860±0.038	0.809±0.087
LR*	0.911±0.005	0.721±0.010	0.899±0.002	0.908±0.015	0.828±0.013	0.844±0.01	0.749±0.068	0.847±0.037	0.838±0.071
TabTransfomer*	0.934±0.004	0.737±0.009	0.906±0.003	0.927±0.010	0.856±0.005	0.835±0.014	0.751±0.096	0.918±0.038	0.858±0.078
GBDT*	0.933±0.003	**0.756±0.011**	0.906±0.002	**0.930±0.008**	**0.859±0.009**	**0.847±0.016**	0.756±0.084	0.913±0.031	0.863±0.073
TabNet@	0.921±0.003	0.711±0.011	0.906±0.004	0.913±0.006	0.841±0.013	0.826±0.010	0.701±0.018	0.643±0.053	0.808±0.077
LR@	0.889±0.003	0.656±0.008	0.848±0.006	0.896±0.006	0.745±0.008	0.845±0.006	0.697±0.044	**0.921±0.025**	0.812±0.099
SAINT@	0.810±0.012	0.728±0.004	0.909±0.003	0.907±0.009	0.814±0.023	0.814±0.008	0.766±0.030	0.918±0.002	0.833±0.071
XGBoost@	0.933±0.002	0.741±0.017	0.919±0.003	**0.930±0.002**	0.854±0.012	0.842±0.004	0.753±0.026	0.914±0.040	0.861±0.078
Our	**0.939±0.005**	0.730±0.008	**0.926±0.002**	**0.930±0.003**	**0.859±0.008**	**0.847±0.002**	**0.783±0.014**	0.916±0.022	**0.866±0.077**

$*$ means cited from TabTransfomer (Huang et al. 2020). @ means our reproduced results.

an initial model and uses textualized tabular data. These two major difference brings extra semantic information for our model to learn a more precise contextual representation.

Notably, our model outperforms the state-of-the-art tree-based methods, XGBoost and GBDT, on six out of eight datasets. The neural-based model is often under-performance than the tree-based model due to the sensitivity to uninformative features (Grinsztajn, Oyallon, and Varoquaux 2022). The superior performance of our model may come from the textual modality transformation and attention mechanism, which help to enrich the features information and selectively ignore the uninformative features.

Besides, our method achieves significantly higher AUC results on datasets with textual category features, such as **BM/IN/SB**. The textual category features mean the feature value is a phrase or sentence. Tab 1 shows the number of textual category features and total features on all datasets. And our method achieves comparable results on numerical features only datasets, such as **AD/QB**. Such results indicate that our method benefits from the pre-training process which can extract information from meaningful features.

6.2 Semi-supervised Learning

To further evaluate the ability of PTab on learning contextual representation, we conduct three groups of experiments in semi-supervised settings. Specifically, we sample a fixed amount of labeled examples from the training set and remove the labels of the unselected training examples. In this way, we set up three groups of experiments with 50, 200, and 500 labeled examples, which is a commonly used settings (Huang et al. 2020; Wei and Zou 2019; Sohn et al. 2020). Five-fold cross-validation is performed for each experiment and the average and standard deviation of the AUC score are reported. We choose the state-of-the-art models on semi-supervised settings, SAINT and TabTransfomer, for comparison. We choose the popular tree-based method, XGBoost, plus Pseudo labeling as the baseline.

Tab 3 show results on 50 labeled examples. The experiment results show that our method outperforms SAINT with a significant margin in terms of average AUC score. In addition, our method achieves the best AUC score on half of the eight datasets. Our methods also achieve the second-best average AUC score. We think that the under-performance may be caused by high variance under different random seeds

Table 3: AUC score for semi-supervised learning models on all datasets with 50 data points. Values are the mean over 5 cross-validation splits.

Dataset	TabTrans	GBDT(PL)	SAINT	Our
BM	0.735±0.040	0.688±0.057	0.677±0.074	**0.769±0.008**
AD	0.613±0.014	0.519±0.024	0.543±0.023	**0.617±0.021**
IN	**0.862±0.018**	0.685±0.084	0.685±0.08	0.828±0.006
OS	0.780±0.024	**0.818±0.032**	0.807±0.023	0.755±0.018
ST	**0.741±0.019**	0.651±0.093	0.662±0.034	0.722±0.025
BC	**0.822±0.009**	0.729±0.053	0.757±0.021	0.787±0.041
SB	0.738±0.068	0.601±0.071	0.720±0.029	**0.749±0.046**
QB	0.869±0.036	0.804±0.057	0.861±0.031	**0.876±0.034**
AVG	**0.770±0.084**	0.687±0.100	0.714±0.097	*0.763±0.076*

Table 4: AUC score for semi-supervised learning models on all datasets with 200 fifine-tune data points. Values are the mean over 5 cross-validation splits.

Dataset	TabTrans	GBDT(PL)	SAINT	Our
BM	0.838±0.010	0.802±0.012	0.835±0.017	**0.856±0.030**
AD	0.614±0.012	0.572±0.04	0.587±0.016	**0.631±0.016**
IN	**0.875±0.011**	0.822±0.02	0.814±0.04	0.862±0.004
OS	0.838±0.024	0.846±0.019	0.869±0.010	**0.897±0.011**
ST	**0.783±0.024**	0.750±0.050	0.643±0.048	0.782±0.026
BC	**0.841±0.014**	0.783±0.017	0.804±0.006	0.812±0.016
SB	0.708±0.083	0.603±0.023	0.758±0.022	**0.761±0.013**
QB	**0.889±0.030**	0.855±0.035	0.885±0.023	**0.889±0.032**
AVG	*0.798±0.094*	0.754±0.108	0.774±0.107	**0.811±0.088**

Table 5: AUC score for semi-supervised learning models on all datasets with 500 fifine-tune data points. Values are the mean over 5 cross-validation splits.

Dataset	TabTrans	GBDT(PL)	SAINT	Our
BM	0.860±0.016	0.838±0.019	0.853±0.025	**0.884±0.003**
AD	0.647±0.008	0.647±0.030	0.651±0.024	**0.672±0.032**
IN	**0.880±0.007**	0.839±0.013	0.875±0.018	0.875±0.006
OS	0.861±0.014	0.865±0.011	0.895±0.008	**0.918±0.008**
ST	0.815±0.004	0.788±0.019	0.772±0.020	**0.834±0.031**
BC	**0.839±0.015**	0.795±0.021	0.815±0.005	0.836±0.011
SB	0.729±0.069	0.666±0.063	0.760±0.025	**0.766±0.021**
QB	0.889±0.038	**0.908±0.024**	0.896±0.030	0.902±0.030:
AVG	*0.815±0.085*	0.793±0.093	0.815±0.084	**0.836±0.082**

Table 6: Results of ablation study

Dataset	BM	AD	IN	OS	ST	BC	SB	QB
our	0.933±0.012	0.730±0.008	0.926±0.002	0.930±0.003	0.859±0.008	0.846±0.004	0.783±0.014	0.916±0.022
$-sep$	0.886±0.024	0.674±0.021	0.915±0.016	0.902±0.007	0.848±0.018	0.826±0.011	0.754±0.017	0.875±0.012
$-mlm$	0.782±0.061	0.596±0.044	0.912±0.009	0.863±0.034	0.843±0.009	0.839±0.006	0.750±0.022	0.838±0.043

when the labeled data is few.

As illustrated in Tab 4 and 5, our method achieves the highest AUC score on 5 out 8 datasets and the highest average AUC score. The experimental results show that our method outperforms counterpart methods by a significant margin with more than 50 labeled examples in semi-supervised settings. For example, our method achieves a 0.836 average AUC score which is 2.6% higher than TabTransfomer/SAINT and 5.4% higher than GBDT(PL). Besides, our method achieves stable improvements in AUC score with the increase of labeled examples, while TabTransfomer(200) achieves lower AUC than TabTransfomer(50) on SB, and SAINT(200) achieves lower AUC than SAINT(50) on ST.

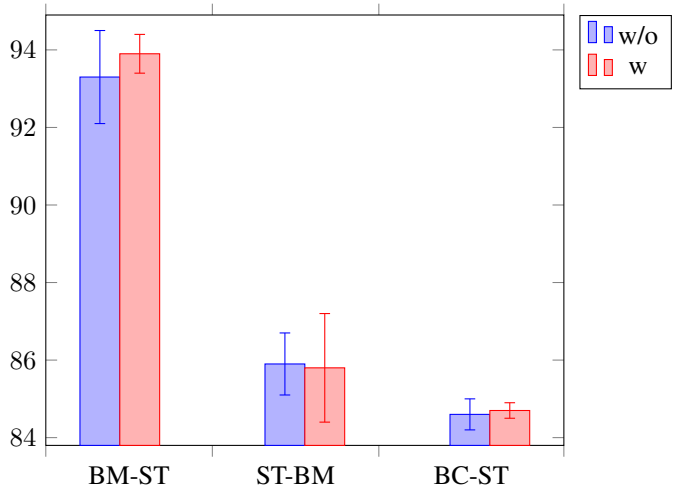


Figure 3: Resutls of training on mixed in-domain datasets.

6.3 Mixed dataset pre-training

To evaluate our proposed method for mixed datasets training, we set up groups of experiments. Mixing datasets in the similar domain is a commonly used method to enlarge the training dataset (Li, Jiang, and Jiang 2021; Li et al. 2022). Existing models are specifically designed for a single dataset and are not extensible to others. Our method can tackle this problem by unsupervised pre-training on related task data and fine-tuning on target task data.

1) We train our model on mixed dataset (**BM**,**ST** and **BC**) from the business domain, which is mentioned in the dataset description in Tab 1. As shown in Fig 3, training on mixed in-domain datasets will help to improve the performance of tabular data modeling. For detail, **BM** aims to predict if the client will subscribe (yes/no) to a term deposit. **ST** aims to predict the fact whether the customer left the bank (closed his account) or continues to be a customer. **BC** aims to predict behavior to retain customers. Besides, they have some feature overlaps, e.g., **BC** and **ST** both contain the time of service ordered and the sex of custom. There is some overlap in feature names and values between these datasets. By textualizing tabular data, our model can learn correlations between features through attention mechanisms. The experimental results show that our model can benefit from the mixed in-domain datasets.

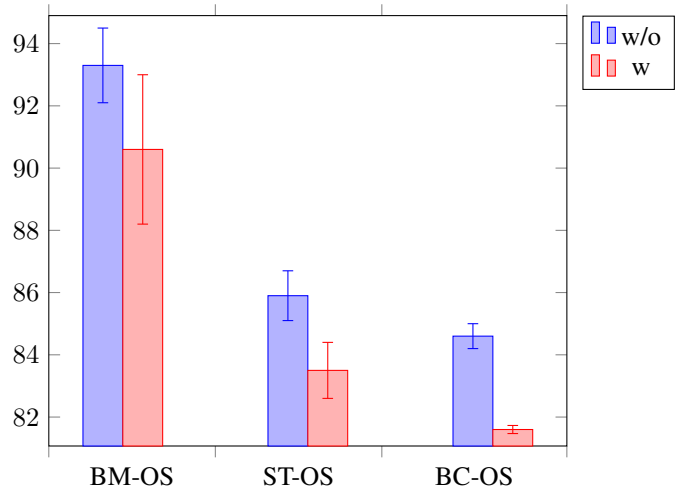


Figure 4: Resutls of training on mixed out-of-domain datasets.

2)For comparison, we also train our model on mixed out-of-domain datasets, as illustrated in Fig. 4. The unrelated task data will significantly reduce the performance of the target task. For example, the **BM** and **OS** has feature overlap,e.g., education and job, but very different task definition. In this situation, the training on mixed datasets may lead to performance degradation.

6.4 Ablation Study

To analyze the effects of each stage in our proposed PTab, we performed ablation experiments on all 8 datasets. As shown in Tab 6, $-sep$ represents the results of removal of sentence separator in the MT stage, and $-mlm$ indicates the results of removal of MF stage. We cannot apply classification without the CF stage, so we do not test the performance with the removal of this component. $-sep$ and $-mlm$ both lead to a decrease in performance significantly. Without the sentence separator, we get significant performance degradation on all datasets. That indicates the MT stage plays an

important role in our methods. Moreover, the removal of the MF stage decreases the performance of our model further. Specifically, $-mlm$ has lower performance than $-sep$. The experiments show that the MT stage the is the key component in our proposed framework.

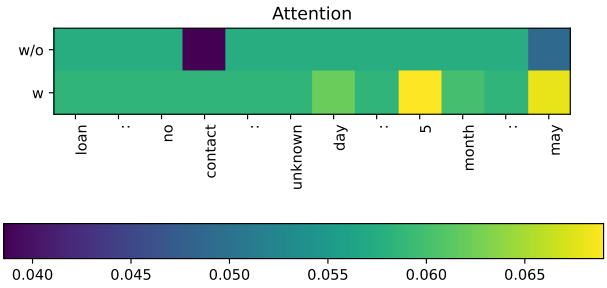


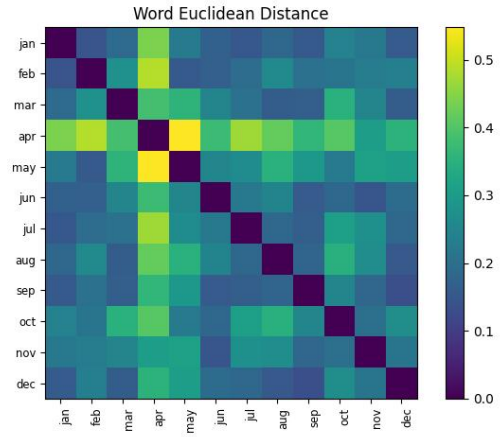
Figure 5: Visualization of attention weights for selected tokens of a randomly picked example from BM.

6.5 Visualization of Attention weights

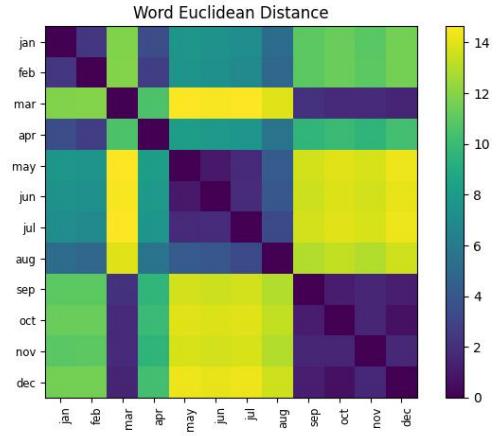
We visualize the attention score to analyze whether the attention scores can learn effective features from textualized tabular data. The $[CLS]$ token is used for final classification. So the attention score of $[CLS]$ token reflect the importance of each features. The visualization of attention scores can help us to explain instant-based feature importance. We draw the attention scores of the $[CLS]$ of a random select example from BM dataset. The attention scores have changed significantly after fine-tuning. As shown in Fig. 5, w/o denotes the attention scores of $[CLS]$ token without Masked language Fine-tuning(MF), w denotes the attention scores of $[CLS]$ token with MF. In w We can see the tokens 5 and may have higher attention scores and the tokens of 'loan' and 'contract' feature have uniformly low attention scores. That shows the ability of our method in selecting features, which addresses the problem of the neural-based method sensitive to the informative features (Grinsztajn, Oyallon, and Varoquaux 2022).

6.6 Effects on modeling feature

To demonstrate that our method can learn contextual representations from textual features, we visualize the similarity between features. As we all know that a good representation of tabular data is sensitive and meaningful to the change of feature value. For example, two data point has very similar feature values except one feature should have different representation that reflects the difference between these feature value. For example, we enumerate the value of "job" and get the embedding of $[CLS]$ token with one randomly example from BM dataset. We use the Euclidean distance between each pair of $[CLS]$ embeddings to measure the similarity. As show in Fig. 6(a), the feature with lower importance, the most unsimilar "job" change from "unemployed"-"housemaid" to "admin"-"retired". Interesting, as show in Fig. 6(b), the month of "last contact" has forms three clusters: $[1, 4], [5, 8], [9, 12]$. That shows the effectiveness of PTab in learning proper contextual representations for tabular data.



(a) w/o Masked language Fine-tuning(MF)



(b) w/ Masked language Fine-tuning(MF)

Figure 6: The Euclidean Distance of examples with different month value.

7 Conclusion and futher works

To model tabular data in textual modality, we propose a novel framework, the Pre-trained language model to model Tabular data (PTab). It enhances the representation of tabular data with semantic information and makes the training on mixed datasets feasible. The experiments shows that PTab outperform the-state-of-the-art baslines on both supervised and semi-supervised settings in terms of average AUC scores. The visulizations analysis show that PTab can effectively select features by attention mechanism and learn contextual representation from textualized tabular data.

In the future, we can improve self-supervised learning in PTab. We have empirically tested the different mask ratios in MF stage and got results with an insignificant difference. The way to effectively learn representation under a self-supervised setting still needs to be explored. Besides, the precise representation of numbers is hard in textual modality. The numbers have often been tokenized into multi-tokens and that may lead to major affection for the learning of contextual representation. But this direction is beyond the topic of this paper.

8 Acknowledgements

This work was supported by the National Key R&D Program of China (2020AAA0105200).

References

Arik, S. Ö.; and Pfister, T. 2021. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 6679–6687.

Borisov, V.; Leemann, T.; Seßler, K.; Haug, J.; Pawelczyk, M.; and Kasneci, G. 2021. Deep neural networks and tabular data: A survey. *arXiv preprint arXiv:2110.01889*.

Chen, J.; Jiménez-Ruiz, E.; Horrocks, I.; and Sutton, C. 2019. Learning semantic annotations for tabular data. *arXiv preprint arXiv:1906.00781*.

Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Deng, X.; Sun, H.; Lees, A.; Wu, Y.; and Yu, C. 2022. Turl: Table understanding through representation learning. *ACM SIGMOD Record*, 51(1): 33–40.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Friedman, J. H. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

Gorishniy, Y.; Rubachev, I.; Khrulkov, V.; and Babenko, A. 2021. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34: 18932–18943.

Grinsztajn, L.; Oyallon, E.; and Varoquaux, G. 2022. Why do tree-based models still outperform deep learning on tabular data? *arXiv preprint arXiv:2207.08815*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hu, Z.; Tan, B.; Salakhutdinov, R. R.; Mitchell, T. M.; and Xing, E. P. 2019. Learning data manipulation for augmentation and weighting. *Advances in Neural Information Processing Systems*, 32.

Huang, X.; Khetan, A.; Cvitkovic, M.; and Karnin, Z. 2020. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*.

Iida, H.; Thai, D.; Manjunatha, V.; and Iyyer, M. 2021. Tabbie: Pretrained representations of tabular data. *arXiv preprint arXiv:2105.02584*.

Li, D.; Jiang, T.; and Jiang, M. 2021. Unified quality assessment of in-the-wild videos with mixed datasets training. *International Journal of Computer Vision*, 129(4): 1238–1257.

Li, J.; Li, D.; Xiong, C.; and Hoi, S. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*.

Shwartz-Ziv, R.; and Armon, A. 2022. Tabular data: Deep learning is not all you need. *Information Fusion*, 81: 84–90.

Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C. A.; Cubuk, E. D.; Kurakin, A.; and Li, C.-L. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33: 596–608.

Somepalli, G.; Goldblum, M.; Schwarzschild, A.; Bruss, C. B.; and Goldstein, T. 2021. Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. *arXiv preprint arXiv:2106.01342*.

Ucar, T.; Hajiramezanali, E.; and Edwards, L. 2021. Subtab: Subsetting features of tabular data for self-supervised representation learning. *Advances in Neural Information Processing Systems*, 34: 18853–18865.

Wang, Z.; Dong, H.; Jia, R.; Li, J.; Fu, Z.; Han, S.; and Zhang, D. 2021. TUTA: tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 1780–1790.

Wei, J.; and Zou, K. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Xie, T.; Wu, C. H.; Shi, P.; Zhong, R.; Scholak, T.; Yasunaga, M.; Wu, C.-S.; Zhong, M.; Yin, P.; Wang, S. I.; et al. 2022. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. *arXiv preprint arXiv:2201.05966*.

Xu, L.; Skoularidou, M.; Cuesta-Infante, A.; and Veeramachaneni, K. 2019. Modeling tabular data using conditional gan. *Advances in Neural Information Processing Systems*, 32.

Yin, P.; Neubig, G.; Yih, W.-t.; and Riedel, S. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. *arXiv preprint arXiv:2005.08314*.

Yoon, J.; Zhang, Y.; Jordon, J.; and van der Schaar, M. 2020. Vime: Extending the success of self-and semi-supervised learning to tabular domain. *Advances in Neural Information Processing Systems*, 33: 11033–11043.

Zhang, L.; Zhang, S.; and Balog, K. 2019. Table2vec: Neural word and entity embeddings for table population and retrieval. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1029–1032.