External Knowledge Infusion for Tabular Pre-training Models with Dual-adapters

Can Qin
Northeastern University
qin.ca@northeastern.edu

Sungchul Kim
Adobe Research
sukim@adobe.com

Handong Zhao*
Adobe Research
hazhao@adobe.com

Tong Yu
Adobe Research
tyu@adobe.com

Ryan A. Rossi
Adobe Research
ryrossi@adobe.com

Yun Fu
Northeastern University
yunfu@ece.neu.edu

ABSTRACT

Tabular pre-training models have received increasing attention due to the wide-ranging applications for tabular data analysis. However, most of the existing solutions are directly built upon the tabular data with a mixture of non-semantic and semantic contents. According to the statistics, only 30% of tabular data in wikitables are semantic entities that are surrounded and isolated by enormous irregular characters such as numbers, strings, symbols, etc. Despite the small portion, such semantic entities are crucial for table understanding. This paper attempts to enhance the existing tabular pre-training model by injecting common-sense knowledge from external sources. Compared with the knowledge injection in the natural language pre-training models, the tabular model naturally requires overcoming the domain gaps between external knowledge and tabular data with significant differences in both *structures* and *contents*. To this end, we propose the dual-adapters inserted within the pre-trained tabular model for flexible and efficient knowledge injection. The two parallel adapters are trained by the knowledge graph triplets and semantically augmented tables respectively for infusion and alignment with the tabular data. In addition, a path-wise attention layer is attached below to fuse the cross-domain representation with the weighted contribution. Finally, to verify the effectiveness of our proposed knowledge injection framework, we extensively test it on 5 different application scenarios covering both zero-shot and finetuning-based tabular understanding tasks over the cell, column, and tables levels.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; **Machine learning**; • **Information systems** → **Data mining**.

KEYWORDS

tabular model, common sense; knowledge injection

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '22, August 14–18, 2022, Washington, DC, USA.
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539403

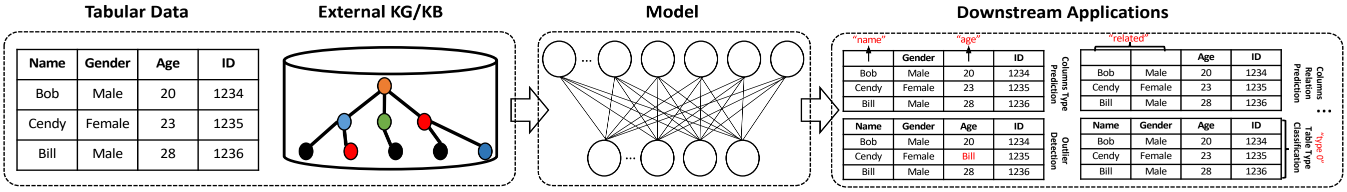
ACM Reference Format:
Can Qin, Sungchul Kim, Handong Zhao, Tong Yu, Ryan A. Rossi, and Yun Fu. 2022. External Knowledge Infusion for Tabular Pre-training Models with Dual-adapters. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3534678.3539403

1 INTRODUCTION

With the advance of large-scale pre-trained language models (LMs) [? ? ? ? ? ? ? ?], the representation learning of tabular data has recently caught increasing attention. Several tabular pre-training models have been proposed to address the tasks like table interpretation, augmentation, and question answering [? ? ? ? ? ?]. However, most existing works have been focusing on pre-training itself, which only rely on the general distribution information of corpora without considering the difference between semantical and irregular characters. Unlike the natural language, most of the tabular data are organized by non-semantical items, including numbers, strings, or symbols, which approximately remain 70% of the tabular pre-training datasets like Wikitables and Common Crawl Tables [? ?]. Therefore, the semantical table entities, including both headers and cells, take the remaining 30% but play an important role in high-level table understanding such as column type prediction or table classification based on the semantical attributes. The equal consideration of these imbalanced data will potentially result in the bias of the model towards the non-semantical items.

Thus, injecting structured knowledge from the external database [? ? ? ? ? ? ?] into LMs is a natural idea to enhance the semantical dependency among the entities. Such similar ideas have been applied in the natural language pre-training models [? ? ? ? ?]. For instance, CN-ADAPT [?] inserts bottlenecks layers into the transformer modules for knowledge fusion. Such bottlenecks layers are regarded as the external parameters optimized by the masked language modeling (MLM) loss over the synthetic corpus by unwrapping ConceptNet [?] into sentences. [?] applies the phrase strategy and entity strategy masking to augment the word-level mask. Nevertheless, there are three main challenges to injecting external structural knowledge into tabular data pre-training models. First of all, the tabular pre-training models are built upon the various tables containing many irregular characters and out-of-vocab (OOV) strings with unique meanings. The semantical entities only take a small part of the whole corpora, which requires the model to overcome the severe noise of irregular characters to learn

[*] Corresponding Author.



Name	Gender	Age	ID
Bob	Male	20	1234
Cendy	Female	23	1235
Bill	Male	28	1236

	Gender		ID
Bob	Male	20	1234
Cendy	Female	23	1235
Bill	Male	28	1236

		Age	ID
Bob	Male	20	1234
Cendy	Female	23	1235
Bill	Male	28	1236

Name	Gender	Age	ID
Bob	Male	20	1234
Cendy	Female	Bill	1235
Bill	Male	28	1236

Name	Gender	Age	ID
Bob	Male	20	1234
Cendy	Female	23	1235
Bill	Male	28	1236

Figure 1: Overview of our model. The proposed model is pretrained by the tabular corpora [?] and external well strctured data such as knowledge graph [?]. Then the model will be applied to multiple downstream tasks by either zero-shot learning or direct finetuning.

semantical dependencies. Secondly, there are severe domain gaps between the external knowledge and tabular data in both data structure and distribution. The tabular data are organized in rigid grids. However, the knowledge base is mainly represented by triples or graphs, where most linked entities are spread across tables without any connections in corpora. Moreover, most of the popular entities in the knowledge base do not have a high frequency in the tabular corpora, which cannot have a significant impact on tabular representation learning. Last but not least, the finetuning of large-scale pre-training models has a high computation cost due to the large amount of parameters. Flexibility is also a crucial issue to be considered to fulfill our objectives.

To address these challenges, this paper invents a novel framework to efficiently embed external knowledge into well pre-trained tabular models for semantical representation enhancement as shown in Fig. 1. The keys of external knowledge infusion can be concluded as three aspects, including 1) external structure design, 2) cross-domain alignment, and 3) knowledge injection loss. In this paper, we select the ConceptNet [?], covering most of useful entities as well as their underlying relations in tabular corpora, as the source of external knowledge. However, such a database cannot be directly fused with the tabular data due to the domain gap. Previous solution [?] has unwrapped the knowledge graph into natural sentences for alignment with pre-training data, which is unreasonable here due to the mismatch in structure between the knowledge graph and tabular data. Instead, this paper treats each entity/relation in the external knowledge graph as an independent cell whose data-processing pipeline can be referred to Fig. 2. A pre-trained tabular model can easily obtain the embeddings of such cell-like entities/relations organized as the tabular. To better infuse the cell-like triplets with existing models, we propose the dual-path adapter inserted within transformer layers. Inside each proposed module, the KG adapter is trained by the triplet-like cells collected externally with the help of TransE loss [?]. The other tabular adapter is applied to enhance the tabular-embedded semantics for alignment of the external knowledge with the tabular corpora mixed by irregular characters. In specific, the semantical-dense tables are selected to train such an adapter. Within these tables, some entity cells are randomly replaced by either linked or unlinked entities according to the external knowledge base to enforce learning the semantical dependencies among the tables. In addition, to bridge the domain gap of knowledge triplets and tabular data, we further devised the path-wise attention layers for feature fusion of the two different adapters with the weighted contribution where the semantical entities and OOVs are expected to have different weights accordingly.

The main contributions of this paper can be summarized as:

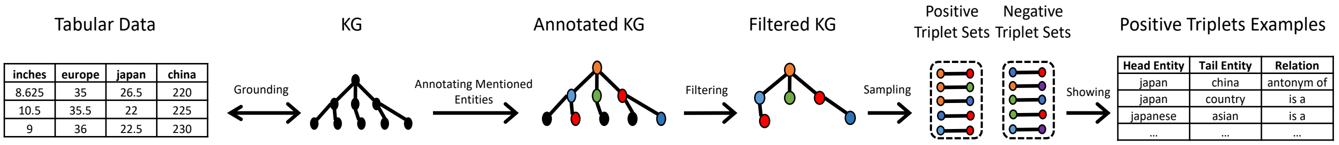
- To the best of our knowledge, this is the first work to introduce external common-sense knowledge into the tabular pre-training models with the post-hoc fine-tuning strategy.
- To align the cross-domain representation of tabular data and external knowledge, we have devised the dual-path architecture adapters with path-wise attention layer for contribution weighting.
- To collect the useful knowledge from the external database, we link the mentioned entities of the training corpora, i.e., Wikipedia and Common Crawl tables [?], with the KG dataset, i.e., ConceptNet [?], and re-organize them as over a half million head-relation-tail triplets for infusion.
- Finally, the proposed knowledge-embedded model has been evaluated across multiple downstream tasks over the tabular datasets, including header type prediction, corrected cell detection, etc.

2 RELATED WORKS

In this section, we review the related works in terms of both Knowledge Graph (KG) injected Language Models (LMs) and tabular LMs.

2.1 KG Injected LMs

Similar ideas have been applied in the natural language pre-training models due to the weakness of LMs for capturing factual knowledge [? ?]. CN-ADAPT [?] takes the BERT [?] as the pre-training models and inserts bottlenecks layers as the adapter [?] into the transformer modules for knowledge fusion. Such bottlenecks layers are regarded as the external parameters optimized by the MLM loss over the synthetic corpus by unwrapping ConceptNet into sentences. Such unwrapping strategy of CN-ADAPT is obviously not reliable for tabular knowledge injection due to the large domain gap between the natural sentences and tabular cells. K-ADAPTER [?] has injected the factual and linguistic knowledge into the pre-trained LM model with the help of multi-task training. To this end, two adapter models are devised for relations classification and dependency relation prediction to learn the multiple kinds of knowledge. ERNIE [?] applies the phrase strategy and entity strategy masking to augment the word-level mask. Such additional masks can help to enhance the representations with the long-term semantic dependencies among the corpus. Instead of unwrapping the KG



inches	europe	japan	china
8.625	35	26.5	220
10.5	35.5	22	225
9	36	22.5	230

Head Entity	Tail Entity	Relation
japan	china	antonym of
japan	country	is a
japanese	asian	is a
...	...	...

Figure 2: KG dataset processing pipeline. The raw tabular data will be aligned with the KG data in the beginning. Then, the mentioned entities and their relations will be selected from the KG dataset. This paper treats each entity/relation in the external knowledge graph as an independent cell as the input to the tabular model. More details can be referred to Sec. 4.1.

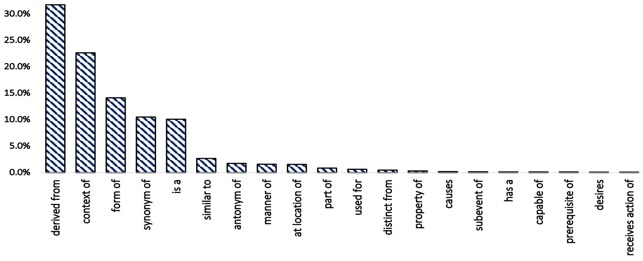
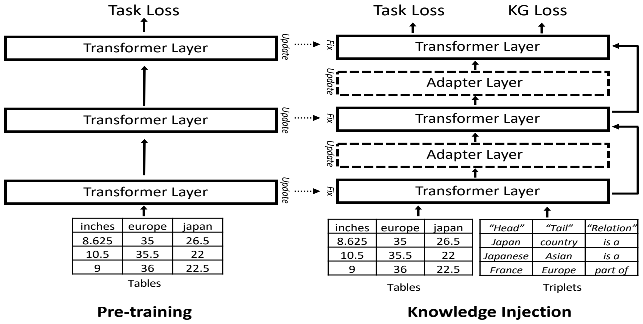


Figure 3: Histogram of top 20 relations in processed KG triplets set. These items follow a long-tail distribution.

as sentences, K-BERT [?] inserted the KG in the natural language with the help of positional indexing. Based on the precise alignment between the training corpora and KG, each sequential sentence will be extended into the tree structure with the leaf branches from KG appended aside. A mask-self-attention strategy is further applied to enforce the model focusing on the visible leaf branches due to the irrelevance between the rest KG leaf branches. BERT-MK [?] integrates the domain knowledge for medical data understanding. Similar to K-BERT, BERT-MK has transformed the raw data as the graph, which will be flattened as the node sequence for input. To ensure the consistency between the training corpora and injected domain knowledge, BERT-MK takes an adjacent matrix to mask the irrelevant nodes during training. KEPLER [?] considers the knowledge injection into LM and embedding learning of KG as a joint task with multiple objects to fulfill.

2.2 Tabular LM

Tabular pre-training has received growing attention due to its high potential for table understanding, which mainly focuses on pre-training itself without considering external common-sense knowledge. TaBert [?] is proposed for QA-based tabular understanding, which takes both background text and tables for model pre-training. TAPAS [?] is proposed to address the tabular-based question answering based on weakly-supervised learning. To do this, TAPAS extends the architecture of Bert with additional embeddings to represent the structure information of tabular data. The final representation of the transformer model will be applied to predict the selecting cells or aggregation operators for semantical understanding. TUTA [?] has devised a tree-structure transformer model to encode the tabular data in various formats. Such the proposed bi-dimensional coordinate tree enables the joint encoding of both positional and hierarchical information. Moreover, tree-based attention is applied for knowledge fusion of the surrounding cells based on tabular structure. Tabbie [?] is also a pioneering approach for pure and general-purpose tabular pre-training without additional text-based input such as titles, captions, premise (QA-based), and it is also the direct baseline for our implementation of pre-training models. The semantical entities are deludedly embedded in Tabbie due to their sparse distribution around the OOV strings. TURL [?] is designed to address the representation learning on relational Web tables whose cells and tables are linked through the internet. Specifically, TURL proposed the Masked Entity Recovery strategy for model pre-training and finetuned the pre-trained model on multiple downstream tasks such as relation prediction and cell filling on the relational tabular data. We take the isolated/independent tables as the inputs, which is more general and is the essential difference with TURL.



inches	europe	japan
8.625	35	26.5
10.5	35.5	22
9	36	22.5

“Head”	*“Tail”*	*“Relation”*
Japan	*country*	*is a*
Japanese	*Asian*	*is a*
France	*Europe*	*part of*

Figure 4: The proposed tabular adapter layers are plugged into the well trained transformer layers with different loss functions to optimize.

3 APPROACH

The final goal of this paper is to inject external common-sense knowledge into a well pre-trained model for tabular data understanding. Such a model should be verified on some downstream tasks as shown in Fig. 1 with the help of finetuning. The keys of valid external knowledge injection can be summarized as three aspects, including 1) external structure design, 2) cross-domain alignment, and 3) knowledge injection loss. As shown in Fig. 4, our proposed knowledge injection employs the tabular adapter layers plugged inside the well-trained transformer layers, which is also model-agnostic and can be applied for most of the tabular data pre-training models. In this paper, we pick the general-purpose tabular pre-training model, i.e., Tabbie [?], as the example for implementation. More details of our solution are given below.

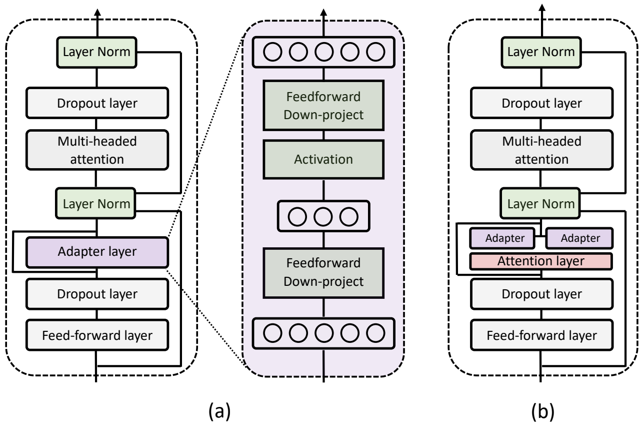


Figure 5: (a): Detailed architecture of the enhanced transformers and the adapter layers. (b): Dual-path adapters with path-wise attention. Inside the figure, the rectangle box with rings indicate a neural layer.

3.1 Base Pre-training Model

In the Tabbie [?] model, two different transformers are applied for row and column representation learning to collect row-wise embedding set $\mathcal{R} = \{r_{i,1}, r_{i,2}, ..., r_{i,N}\}$ and column-wise embedding set $C = \{c_{i,1}, c_{i,2}, ..., c_{i,M}\}$. The tabular pre-training model takes an $M \times N$ table as input and outputs embeddings $\mathbf{X} = \{x_{ij} | i = 1, .., M, j = 1, ..., N\}$ for each cell. Specifically, the contextualized cell embedding is the average of row embedding and column embedding:

$$r_{i,j}^{L} = \phi_{\theta_r}(x_{i,j}^{L}), \tag{1}$$

$$c_{i,j}^{L} = \phi_{\theta_c}(x_{i,j}^{L}), \tag{2}$$

$$x_{i,j}^{L+1} = (r_{i,j}^{L} + c_{i,j}^{L})/2, \tag{3}$$

where L denotes the index of transformer layer, and θ_r and θ_c represent the parameters of row transformer and column transformer respectively. The subscripts i and j denote the coordinates of the cell at the i-th column and j-th row. The base model adopts corruption loss by predicting if the cell is corrupted or not:

$$p_{i,j} = \sigma\left(w^T x_{i,j}^{L}\right), \tag{4}$$

where $\sigma(\cdot)$ denotes a Sigmoid function and w represents the projection matrix for outlier cell prediction. Such outlier cells can be self-supervisedly generated by automatically swapping and removing some cells with the labels as either 0 or 1 to represent polluted or not. Therefore, the pre-training object is a binary cross entropy loss:

$$\mathcal{L}_{task} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} y_{i,j} \log p_{i,j} + (1 - y_{i,j}) \log(1 - p_{i,j}), \tag{5}$$

where $y_{i,j}$ means the cell-wise corruption label.

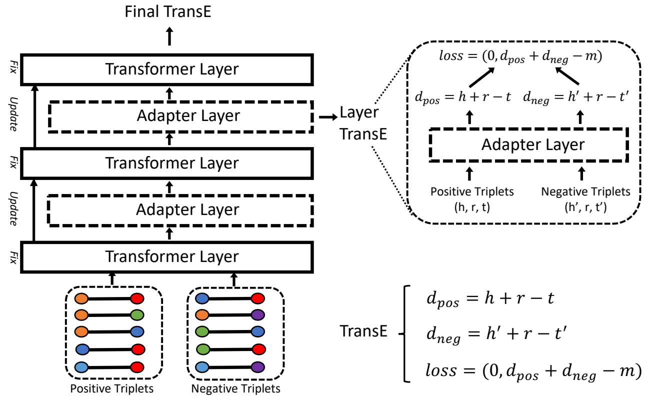


Figure 6: Multilayer external knowledge injection.

3.2 Dual-path Tabular Adapters

Adapter was proposed in the natural language pre-training models for efficient adaptation to downstream applications [? ? ? ? ?]. It is only required to update the parameters of adapter layers based on finetuning loss, with the majority of parameters fixed. The basic idea behind this comes from the transfer learning that the low/mid-level representation are shared across similar tasks [? ?]. Therefore, it is reasonable to have the similar assumption on the tabular pre-training models whose general representations can be enhanced by the external knowledge.

As shown in Fig. 5, there are two different types of adapters proposed for tabular pre-training models. The vanilla version is simply plugging the adapter between the dropout and layer-norm layers, which can be formulated as:

$$\phi_{\theta_{ad}}(h) = h + w_u^T f(w_d^T h + b_d) + b_u, \tag{6}$$

where h is the embedding of previous layer and w_d and w_u represent the downscale and upscale projection matrices with the corresponding bias weights as b_d and b_u. $f(\cdot)$ is the activation function such as ReLU.

Due to the gap between tabular data and external knowledge, the alignment between such two domains are necessary. To this end, we have devised the dual-path tabular adapter with a knowledge adapter and a tabular adapter parameterized by θ_k and θ_t, respectively, given different input data. Accordingly, the knowledge adapter $\phi(\cdot)_{\theta_k}$ will only be trained by the external knowledge and tabular adapter $\phi(\cdot)_{\theta_t}$ will be trained by the semantically augmented tabular data. During downstream finetuning, both adapter models need to be updated with an attention layer to weight the contributions from two paths:

$$Adapter(h) = w_k \phi_{\theta_k}(h) + w_t \phi_{\theta_t}(h), \tag{7}$$

where the path-wise weights w_k and w_t are computed by the MLP layer as:

$$[w_t, w_k] = MLP_{\theta_{att}}(h), \tag{8}$$

where $h \in \mathbb{R}^d$ denotes a cell embedding.

Table 1: Column Type Classification

	TaBert			Tabbie-F			Tabbie-M			Ours-NoKG-F			Ours-20K-F			Ours-Full-F		
#Data	Acc@1	Acc@3	F1	Acc@1	Acc@3	F1	Acc@1	Acc@3	F1	Acc@1	ACC@3	F1	Acc@1	Acc@3	F1	Acc@1	Acc@3	F1
1K	-	-	84.7	85.1	92.7	82.0	84.8	91.9	81.6	85.4	92.6	82.1	86.0	92.8	83.1	85.9	93.0	82.8
10K	-	-	93.5	93.4	96.9	91.8	92.2	96.4	90.7	93.2	97.1	92.0	93.6	97.8	92.7	93.8	97.8	92.8
56K	-	-	97.2	96.4	98.4	95.5	95.7	98.3	94.2	96.5	98.7	95.4	96.9	98.1	95.8	97.0	98.0	95.7
Avg	-	-	*91.8*	91.6	96.0	89.8	90.9	95.5	88.8	91.7	96.1	89.8	**92.2**	*96.2*	**90.5**	**92.2**	**96.3**	*90.4*

3.3 External Knowledge Injection

The external knowledge expected to be injected into the pre-trained model come from the raw knowledge graph dataset. In specific, the knowledge graphs are commonly denoted as $\mathcal{KG} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E} = \{e_1, ..., e_N\}$ is the set of entities and $\mathcal{R} = \{r_1, ..., r_P\}$ is the relation set. $\mathcal{T} = \{(e_{t_i^1}, r_{t_i^2}, e_{t_i^3}) | 1 \leqslant i \leqslant T, e_{t_i^1}, e_{t_i^3} \in \mathcal{E}, r_{t_i^2} \in \mathcal{R}\}$ represents the head-relation-tail triplet set. $N_v = \{(r, u) | (v, r, u) \in \mathcal{T}\}$ represents the set of neighboring relations and entities of an entity v which is also considered as the positive/correct data. This paper has applied the ConceptNet [?] as the source of external KG.

TransE [?] is a classic method for knowledge representation learning. In TransE, the tail entity is represented as the sum of head entity embedding and relation embedding: $\overrightarrow{h} + \overrightarrow{r} = \overrightarrow{t}$ where $(\overrightarrow{h}, \overrightarrow{r}, \overrightarrow{t}) \in S$. The negative triples, i.e., $(\overrightarrow{h'}, \overrightarrow{r'}, \overrightarrow{t'}) \in S'$ cannot satisfy such constraint. To this end, the TransE loss is defined as:

$$\mathcal{L}_{TransE} = \sum_{(h,r,t) \in S} \sum_{(h',r,t') \in S'} [\gamma + d(h + r, t) - d(h' + r, t')], \quad (9)$$

to maximize the difference between positive and negative triplets. To ensure the dense knowledge injection, as shown in Fig. 6, we have employed the multilayer training where the TransE loss can be computed in both final layer and the higher adapter layers.

3.4 Joint Training of KG and Tabular Data

The domain gap between the two modalities is crucial for structural knowledge injection into tabular pre-training models, i.e., triples and tabular forms. The vanilla tabular adapters are directly optimized by the TransE loss from raw knowledge triplets, which cannot address the mismatch issue. Instead, our proposed dual-path adapter considers the domain gap and applies two different adapters for feature fusion.

Here, we share the details about training the dual-path tabular adapters with the iterative optimization between the task loss and knowledge loss on different inputs. The final training losses are:

$$\hat{\theta}_{att}, \hat{\theta}_t, \hat{\theta}_k = \underset{\theta_{att}, \theta_t, \theta_k}{\arg\min}\ \mathcal{L}_{task}(\mathbf{X}) + \mathcal{L}_{TransE}(S, S'), \quad (10)$$

where the two losses are iteratively optimized in our implementation, and $\hat{\theta}_{att}, \hat{\theta}_t, \hat{\theta}_k$ denote the updated parameters of path-wise attention network, tabular adapter and knowledge adapter, respectively.

Algorithm 1: External Knowledge Injection for Dual-path Tabular Adapters

```
Input: External Knowledge Graph $\mathcal{KG} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$.
  Semantically Augmented Tabular Forms
  $\mathcal{X} = \{(x_{ij}, y_{ij}) | i = 1, .., M, j = 1, ..., N\}$ .
Initialize: Tabular Adapter $\Theta_k = \{\theta_k^i | i = 1, ..., L\}$.
  Knowledge Adapter $\Theta_t = \{\theta_t^i | i = 1, ..., L\}$. Path-wise
  Attention Network $\Theta_{att} = \{\theta_{att}^i | i = 1, ..., L\}$.
Sample the positive and negative knowledge graph triples,
  i.e., $S$ and $S'$, from $\mathcal{KG}$.
for $t = 1 \sim T$ do
    Achieve embeddings of tabular data $h_t = \phi(X)$ ;
    Achieve embeddings of knowledge triplets $h_k = \phi(S)$
      and $h_k^{'} = \phi(S')$ ;
    Compute the task loss $\mathcal{L}_{task}$ as Eq. (5) ;
    Compute the triplet loss $\mathcal{L}_{TransE}$ as Eq. (9) ;
    Back-propagate gradients and update $\Theta_t$, $\Theta_k$ and $\Theta_{att}$.
end
Output: $\Theta_t^*$, $\Theta_k^*$, $\Theta_{att}^*$ .
```

4 EXPERIMENTS

4.1 Datasets Processing

There are multiple challenges to overcome to obtain high-quality external knowledge for representation enhancement. The topmost one is the data processing to transform and align the external knowledge with our tabular data. To do this, we first extract the named entities from the ConceptNet dataset [?], covering most of useful entities as well as their underlying relations in tabular corpora. Then, a pre-defined parsing rule is applied to match the similarity of tabular cells with the entities in the semantic space. Finally, the filtered entities are selected to retrieve the existing triplets as the aligned external knowledge for our purpose. More details of the whole pipeline are illustrated in Fig. 2.

We also provide more details about the processed KG triplets and the statistics of semantical entities in the tabular data. Tab. 2 shows the whole picture of tabular pre-training data as well as the extracted triplets well aligned with entity cells. Moreover, Tab. 4 reveals that the over 30% tabular cells can be regarded as the semantical entities mixed with 70% non-semantical cells, including numbers, out-of-vocabulary strings, symbols, etc. Such statistics support our arguments that the tabular models should not equally consider the semantical and non-semantical cells since the former carries most of the information, which, however, only takes a small part of the whole tabular data. Our knowledge injection is based on the extracted triplets whose semantical entities are already presented in the tabular data. However, due to the nature of the Wikitable corpora, the entities, as well as their relations, follow the long-tail distribution as shown in Fig. 3.

Table 2: Statistics of the Entities of the Wikipedia Tabular Data

Tables	Headers	Cells	Entities	Unique Entities	Triplets	Relations	Headers/Table	Cells/Table	Entities/Table
3 M	15.62 M	250.38 M	80.59 M	91,744	654,873	30	5.21	83.47	26.86

Table 3: Statistic of Parameters

	Tabbie	Adapter	Dual-adapters
Parameters	279.6 M	3.578 M	4.466 M

Table 4: Statistics of the Entities and Their Ratios, i.e., (Quantity/Percentage), of the Tabular Data

Overall Entities	Headers Entities	Non-Headers Entities
80.59 M (32.19%)	10.09 M (64.58%)	70.51 M (30.03%)

4.2 Experiments Setup

Implementation. Our proposed adapter-based knowledge injection is agnostic to most of the existing tabular pre-training models. Here we choose Tabbie [?] as our base model, which a general-purpose tabular pre-training model. The vanilla Tabbie model is composed of 12 layers with a hidden dimensionality of 768 for both row and column Transformers. Each layer includes a row and a column transformer, respectively. A pre-trained Bert embedder initializes the input to Tabbie. Our adapter takes a bottleneck architecture with the downscale projection from the dimensionality of 768 to 48 and the upscale projection from 48 to 768. The path-wise attention network applies a two-layer MLP with the output in the dimensionality of 2. Both adapters in the dual-path model take the identical architecture trained by different data. Therefore, there are 48 adapters inserted within the whole pre-trained model for knowledge injection. Despite the large number, the adapter layers only take a small portion of the whole model whose quantity of parameters can be referred to Tab. 3. For model optimization, we take the Adam with decoupled weight decay (AdamW) [?] as the optimizer on PyTorch [?]. The learning rate is assigned as 0.0002 in both knowledge injection and downstream tasks.

Baselines. There are multiple methods to compare. The two-version Tabbie, including Mix and Freq with different training strategies, can be regarded as the direct baselines. The Tabbie-Freq only uses frequency-based cell sampling, and Tabbie-Mix is trained by the 50/50 mixture of frequency-based cell sampling and intra-table cell swapping. TaBert [?] is another baseline designed for Tabular QA. Variant ablations, such as the adapter w/o kg (No-KG), or w/o tabular adapter, are also necessary to be included to evaluate the effects of knowledge injection. The quantity of external knowledge will also impact the performance. We have selected 20K triplets or the full version (i.e., 654,873 triplets) for comparison in most tasks to explore its detailed influence.

Evaluation metrics. The evaluation tasks can be classified as binary/multiclass classification and 1-to-N retrieval, whose matrices are slightly different. In the outlier cells or tables detection, we take the precision, recall, and f1 score as the evaluation metrics. Other tasks, such as column classification, column relation classification, and eve relation retrieval, can be considered as a multiclass-based classification or retrieval problem. Accuracy at k (acc@k) is a widely used metric in multiclass-based problems. acc@k represents the ratio of any top k prediction results that match the ground truth. We select different k to demonstrate the accuracy over various scales according to the number of classes. Among most of the tables for quantitative comparison, the best results are marked in **bold**, and the second-best ones are highlighted in *italic*.

Table 5: Relation Classification of Columns

Data	1K			10K		
Methods	@1	@3	@5	@1	@3	@5
Tabbie-M	85.3	95.2	97.5	92.3	97.6	99.2
Tabbie-F	86.4	95.3	97.1	92.2	98.1	99.0
Ours-NoKG-M	85.3	94.1	96.4	92.5	98.0	98.6
Ours-NoKG-F	86.1	94.7	**97.8**	92.2	*98.6*	*99.3*
Ours-20K-M	86.0	**95.6**	*97.6*	*92.7*	98.4	98.8
Ours-Full-M	86.1	95.4	96.8	**92.8**	97.9	98.8
Ours-20K-F	**87.5**	95.1	96.8	92.4	**98.7**	*99.3*
Ours-Full-F	*86.7*	*95.5*	97.0	92.5	**98.7**	**99.4**

Benchmarks. For a fair comparison with Tabbie and TaBert, this paper takes the Wikipedia tables mixed with preprocessed Common Crawl [?] as the benchmarks for table-based tasks. To demonstrate the effectiveness of knowledge injection, we also conducted the results on the knowledge level in which the model intends to predict the relations of two entities. We randomly sample the 10,000 triplets from the ConceptNet [?] as the benchmark for evaluation.

4.3 Column Classification

The column classification is the task that predicts a high-level type of a particular column (e.g., name, age, etc.) without access to its header. This task applies when processing the tables with missing or unreliable headers, which often happens in practice. With help from accurate header classification, these missing or incorrect headers will be recovered for further analysis. Moreover, these high-level predictions potentially cluster the columns into several groups that are helpful for content summarization. The experimental results can be referred to in Tab. 1 and Tab. 8 that most of the reported scores are collected from the same benchmarks, including 1k, 10k, and 57600 tables for finetuning. Both tasks have the shared set of testing tables for a fair comparison. From the tables, we can easily notice that the proposed knowledge injection has boosted the performance in most of the scenarios. Especially considering the ablation between w/o or w KG injection, the experimental results show that the proposed injection effectively improves the column type classification beyond the help of extra parameters, which also brings certain benefits for this downstream task.

Table 6: Outlier Cell Classification on Hybrid Data

	Intra-table swap									Random swap								
	Non-header			Header			Overall			Non-header			Header			Overall		
Methods	Pre	Recall	F1	Pre	Recall	F1	Pre	Recall	F1	Pre	Recall	F1	Pre	Recall	F1	Pre	Recall	F1
TaBert	-	-	-	-	-	-	81.2	69.5	74.9	-	-	-	-	-	-	86.7	87.0	86.8
Tabbie-F	98.1	70.4	81.9	99.3	*98.9*	99.1	98.2	73.8	84.3	99.3	**97.4**	**98.4**	*99.3*	**99.0**	**99.2**	99.3	**97.5**	**98.5**
Tabbie-M	97.7	**82.2**	**89.4**	*99.5*	**99.1**	**99.3**	97.9	**84.3**	**90.6**	99.2	97.3	*98.3*	**99.4**	*98.9*	*99.1*	99.2	95.5	*98.3*
Ours-20K-M	*98.5*	*81.1*	*89.0*	**99.6**	98.8	*99.2*	**98.7**	*83.2*	*90.3*	99.3	96.9	98.1	*99.3*	98.8	*99.1*	99.3	97.1	98.2
Ours-Full-M	**98.6**	81.0	88.8	99.4	98.8	99.1	**98.7**	82.9	90.1	*99.3*	97.0	98.1	99.3	98.7	99.0	99.3	97.2	98.2
Ours-20K-F	98.3	68.9	81.0	99.2	98.0	99.1	98.5	72.5	83.5	*99.4*	*97.1*	98.2	99.2	*98.9*	*99.1*	*99.4*	*97.3*	*98.3*
Ours-Full-F	*98.5*	66.1	79.1	99.4	98.7	99.0	*98.6*	70.0	81.9	**99.5**	96.3	97.9	*99.3*	98.7	99.0	**99.5**	96.5	98.0

Table 7: Cell Classification on Semantical Entities

	Mix Model			Freq Model		
Methods	Pre	Recall	F1	Pre	Recall	F1
Tabbie	59.1	64.9	61.2	60.2	66.5	63.2
+ 20K KG	**74.0**	**78.2**	**76.0**	**77.0**	*81.7*	**79.3**
+ Full KG	*73.2*	*77.7*	*75.4*	*76.3*	**82.6**	**79.3**

Table 8: Column Classification on Semantical-rich Data

	Baselines		Ours	
Finetuning Data	Tabbie-F	Tabbie-M	Ours-20K	Ours-100K
5K	85.9	85.7	*86.2*	**86.6**
33K	89.1	88.8	*89.9*	**90.0**

4.4 Relation Prediction

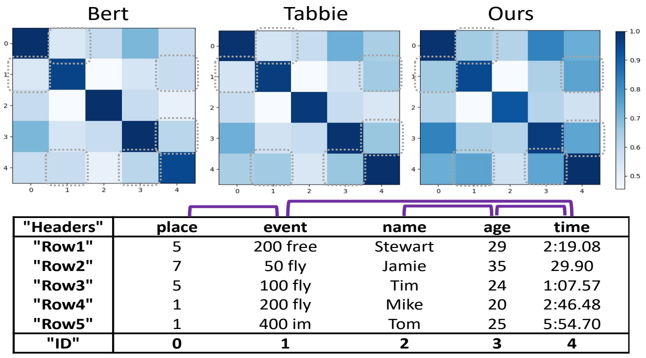
Column relation prediction is a unique task applied in this paper that aims to classify the specific relations of two columns when masked headers. Ground-truth relations are obtained based on the reasoning over the header entities in the KG database. For instance, the headers of two columns could be "California" and "US", whose relation is annotated as "part of" in the ConceptNet. The goal of this task is to predict the desired relation, i.e., "part of" by masking the headers "California" and "US". This downstream task is useful when it is required to infer the content of tables with missing headers. This technique also applies to infer or enrich hidden relations/schema of the tabular columns in the case of schema completion. We have conducted the experiments on two sets with 1k and 10k tables, respectively. In addition, there is a shared testing set with 1k tables. As shown in Table 5, most KG injected models have outperformed the baseline ones over acc@1, acc@3, and acc@5. Moreover, the no-KG ablations are still inferior to KG ones.

4.5 Cell Value Prediction

This task aims to evaluate the model over the cell-level tasks. We have shown the two settings here, i.e., Tab. 6 and 7, in which the former one follows the [?] with randomly corrupted cells. Another setting considers more fine-grained cell classification where some of the entity cells are replaced by the "counterpart entities + relation" according to the positive triplets in KG. The models are required to detect the replaced cells, which has practical use in the missing cell completions given a pool of candidates. Moreover, cell value prediction is also helpful to pre-processing the raw and noisy tabular data with corrupted values. The results of the first setting can be referred to Tab. 6 where the two sub-settings come from [?] with different strategies for cell corruption. Tabert's results are directly copied from [?] with some blank items since there is no open-source code for Tabert on this task. Other results of baselines are collected from our re-implementation which are similar to the reported results in [?]. We notice that the performance is comparable after knowledge injection. On Tab. 7 of semantic cells detection, the proposed methods show remarkable improvements compared with the baselines. In this setting, the model needs to

Table 9: Tables Classification

Methods	Pre	Recall	F1-score
Tabbie	**95.8**	*82.1*	88.5
+ NoKG Adapter	*95.6* (-0.2)	78.6 (-3.5)	*86.3* (-2.2)
+ KG Adapter	93.1 (-2.6)	**96.4** (+14.3)	**94.7** (+6.2)



"Headers"	place	event	name	age	time
"Row1"	5	200 free	Stewart	29	2:19.08
"Row2"	7	50 fly	Jamie	35	29.90
"Row3"	5	100 fly	Tim	24	1:07.57
"Row4"	1	200 fly	Mike	20	2:46.48
"Row5"	1	400 im	Tom	25	5:54.70
"ID"	0	1	2	3	4

Figure 7: The above figure visualizes the heatmap of the dot product between column-wise embedding collected by different models. The purple lines link the connected headers in KG, whose corresponding affinity scores are marked in gray boxes for comparison.

Table 10: Ablations Study on Zero-shot Retrieval Tasks

Architecture		KG Quantity		Column-based Retrieval				Entity-based Relation Retrieval			
KG-Ada	Entity-Ada	20K	100K	Acc@3	Acc@5	Acc@10	Acc@15	Acc@3	Acc@5	Acc@10	Acc@15
-	-	-	-	2.11	4.91	39.40	49.51	9.79	14.85	29.88	47.94
✓	-	✓	-	3.42	5.41	43.57	56.72	13.34	23.46	32.68	48.12
-	✓	-	-	2.34	5.11	39.51	50.14	10.20	15.13	29.81	48.11
✓	✓	✓	-	*4.87*	*6.23*	*45.72*	**58.43**	*21.43*	*29.91*	*41.84*	*50.61*
✓	✓	-	✓	**4.89**	**6.43**	**46.12**	*58.31*	**22.61**	**30.02**	**41.97**	**50.72**

Table 11: Entity-based Relation Retrieval with Different γ

Margins	Acc@3	Acc@5	Acc@10	Acc@15
γ=0.1	**21.71**	*30.60*	*44.09*	*52.20*
γ=0.2	*21.43*	29.91	41.84	50.61
γ=0.4	20.84	**31.12**	**44.79**	**52.41**

distinguish the original entity and the similar entity with extending meanings. Such different phenomena have demonstrated that the extra knowledge injection contributes more to the semantical-dense tasks.

4.6 Table Classification

To verify the effectiveness of our knowledge infusion for such a task, we have collected some tables with binary labels to indicate the density of semantic entities. The tables with more than half of semantic entities will be annotated as type 0, and those below 30% are marked as 1. Such a task is helpful for filtering the informative tables with rich semantics. After finetuning the models on 100 labeled tables, there are another 100 tables as the testing set. The quantitative comparison can be referred to Tab. 9 that the KG adapter has greatly boosted the performance compared with both vanilla Tabbie and adapters-inserted-Tabbie without knowledge injection.

4.7 Zero-shot Retrieval

We have applied the zero-shot relation retrieval to verify the effectiveness of representation enhancement by knowledge injection given either columns or cell entities. Compared with the natural language models, there are only limited tabular works devised to investigate zero-shot tasks. However, such zero-shot tasks are practical in many applications, such as schema matching, where the model does not need updating.

Column-based Relation Retrieval. Column-based relation retrieval is a relatively challenging task due to the heavy noise of unrelated cells. To further enhance the difficulty, we mask the headers of each column for the pure evaluation of the semantical affinity of cells. The experimental results can be referred to Tab. 10.

Entity-based Relation Retrieval. The right half of Tab. 10 shows the entity-based relation retrieval where the model is expected to retrieve the top matched relations with two input cell entities [? ? ?]. This task is only applied to evaluate the knowledge representation of the tabular model, which does not have much practical use. From the table, we can easily see that the proposed knowledge injection has boosted column and entity retrieval accuracy. In addition, more KG triplets bring more gains in the performance on most of the scenarios.

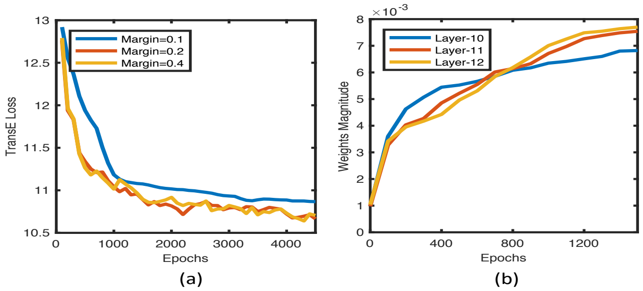


Figure 8: (a) Curve of TransE losses with different margins, i.e., γ. (b) Increase of magnitude of adapters' weights trained by the TransE loss.

4.8 Analysis

Loss Margins. The margin γ of TransE loss in Eq. 9 has a significant influence on the overall performance. Therefore, we have investigated the performance and loss curve with different γ. Tab. 11 shows that the best retrieval results can be achieved by $\gamma = 0.4$. According to Fig. 8 (a), the lower bound of $\gamma = 0.1$ is higher than the rest two which indicates that the $\gamma = 0.2$ and $\gamma = 0.4$ are better choices. We have applied γ as 0.2 in most of the experiments.

Magnitude of Adapters' Weights. Knowledge injection is highly related with the training of KG adapters. In Fig. 8 (b), we have plotted the magnitude of adapters along different epochs. It is clear to notice that the magnitude has increased in scales during training, which is a side evidence of effective knowledge injection.

Case Study. In Fig. 7, we have visualized the affinity matrix of three different models with the same input table. Some of the columns are highly related as indicated by the linked lines. The affinity scores of these linked headers have increased after knowledge infusion, which is a side evidence for the enhancement of semantic dependencies.

5 CONCLUSION

This paper attempts to improve the existing large-scale tabular pre-training models by infusing common-sense knowledge, which is flexible and easy to plug in. Compared with the knowledge infusion into the natural-language-based pre-training models, the tabular models naturally require overcoming the domain gaps between external knowledge and tabular data with the significant difference in both structures and contents. We have proposed the dual-path adapters inserted within the well pre-trained tabular models. Specifically, the dual-path adapters are trained by the knowledge triplets and semantically augmented tables for injection. A path-wise attention layer is applied to fuse the cross-modality representation of the two designs for the final object. To verify the effectiveness of our proposed knowledge injection framework, we have tested it on multiple downstream tasks ranging in cell, column and table levels under both zero-shot and finetuning-based settings. The highly semantic tasks are more beneficial from this technique.

6 ACKNOWLEDGEMENTS

This paper and the research behind it would not have been possible without the exceptional support of collaborators from Adobe Research. We would also like to show our gratitude to the Varun Manjunatha and Hiroshi Iida for their help.