Missing Value Imputation via Pre-trained Language Models with Trainable Prompt and Retrieval Augmentation

Xiang Huang, Shuang Hao*
School of Computer Science and Technology, Beijing Jiaotong University, China
Beijing Key Laboratory of Traffic Data Analysis and Mining, China
{22120382,haoshuang}@bjtu.edu.cn

ABSTRACT

Dealing with missing values in tabular training data is a challenging preparation phase before proceeding to model training. Directly removing missing data results in a loss of information. Thus, these missing values are often estimated using extensively studied imputation techniques. Recently, pre-trained language models (PLMs) have shown exceptional performance in various language processing tasks, leading to several approaches that employ PLMs to predict missing values in tabular data. However, these methods usually concatenate all attribute names and their values in a tuple as the input to PLMs and rely solely on current tabular data for missing value imputation, which cannot realize the full potential of PLMs that are pre-trained with corpus in natural language. This paper introduces PRPMI, a novel PLM-based missing value imputation framework. It combines a trainable tuple representation technique that converts each tuple into a format that can be easily understood by PLMs. Moreover, a knowledge retrieval module is designed to search for additional joinable tabular data or relevant documents from the data lake or the Internet, which are beneficial for missing value imputation tasks. Preliminary experiments on multiple datasets demonstrate the superiority of PRPMI compared to existing state-of-the-art imputation techniques and verify the effectiveness of these two techniques.

VLDB Workshop Reference Format:
Xiang Huang, Shuang Hao*. Missing Value Imputation via Pre-trained Language Models with Trainable Prompt and Retrieval Augmentation. VLDB 2024 Workshop: The 1st International Workshop on Data-driven AI (DATAI).

VLDB Workshop Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/xianghuang2/MIPLM.

1 INTRODUCTION

Missing data is common in real-world scenarios due to equipment malfunctions or mismatches during the integration of heterogeneous data, which can produce biased estimates, leading to invalid conclusions. Thus, it is imperative to fix the missing data before commencing data mining and analysis.

Existing Methods and Their Limitations. The most common strategy for dealing with missing values is reconstructing the entire dataset through data imputation techniques, which includes statistics-based [2], heuristic [3], machine learning (ML) based [9, 16], and deep learning (DL) based methods [8, 18]. Most of these techniques primarily rely on the statistical characteristics, data distribution, co-occurrence of attribute values, or symbolic similarity between tuples, which often fail to capture semantic information.

Recently, with the success of PLMs in natural language processing (NLP), new data imputation approaches based on PLMs, such as TURL [5], IPM [14] and RPT [17], have emerged. TURL utilizes PLMs to generate the representation for missing data, which is then linked with additional knowledge base entries to retrieve the actual value. IPM leverages PLMs to learn the semantic features of a tuple and treats missing value imputation as a classification task. RPT uses a transformer-based neural translation architecture to learn how to reconstruct the original tuples to support the missing value imputation. **(L1)** These methods take the tuple as a sentence for input, which concatenates all attribute names and their values. This "unnatural" approach to building sentences makes it difficult to fully utilize PLMs that are pre-trained with corpus in natural language. **(L2)** Furthermore, most of these approaches rely solely on current tabular data for missing value imputation. When there is insufficient information in the other attributes to infer the missing value, these methods will fail.

Our Proposal. We propose a novel PLM-based data imputation framework named PRPMI to address the aforementioned limitations. PRPMI models the imputation task as the multiclass classification problem, which takes the tuple with missing values as input and predicts their original values, where the candidates are from the domain of corresponding attributes. Moreover, to tackle the first limitation **(L1)**, we concatenate trainable continuous prompt embeddings with discrete tokens in a tuple based on P-tuning [12], feeding them together as the input to the PLMs and refining through model training to optimize the task objective. This way, the language model can better capture semantic information within the tuple. To address the second limitation **(L2)**, we borrow the idea from the retrieval-augmented generation (RAG) [10] that searches for additional joinable tabular data or relevant documents from the data lake or the Internet to provide extra knowledge for missing value imputation. Note that these techniques can also apply to other data preprocessing tasks, such as PLM-based error detection and data repair.

It has to be mentioned that there is also some work on fine-tuning large language models (LLMs) such as GPT and LLaMa through prompt engineering, applying LLMs to several table tasks, including missing value imputation [11, 19]. This branch of studies typically

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
* Shuang Hao is the corresponding author.

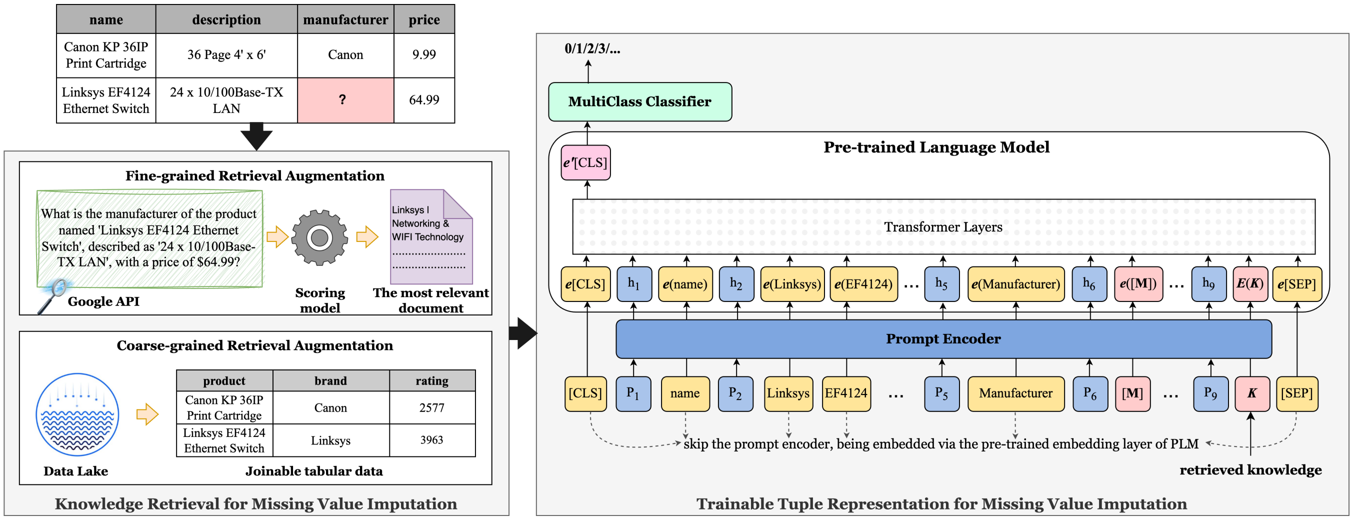
name
description
manufacturer
price
Canon KP 36IP Print Cartridge
36 Page 4' x 6'
Canon
9.99
Linksys EF4124 Ethernet Switch
24 x 10/100Base-TX LAN
?
64.99
Fine-grained Retrieval Augmentation
What is the manufacturer of the product named 'Linksys EF4124 Ethernet Switch', described as '24 x 10/100Base-TX LAN', with a price of $64.99?
Google API
Scoring model
Linksys I Networking & WIFI Technology
The most relevant document
Coarse-grained Retrieval Augmentation
Data Lake
product
brand
rating
Canon KP 36IP Print Cartridge
Canon
2577
Linksys EF4124 Ethernet Switch
Linksys
3963
Joinable tabular data
Knowledge Retrieval for Missing Value Imputation
0/1/2/3/...
MultiClass Classifier
Pre-trained Language Model
e'[CLS]
Transformer Layers
e[CLS]
h1
e(name)
h2
e(Linksys)
e(EF4124)
h5
e(Manufacturer)
h6
e([M])
h9
E(K)
e[SEP]
Prompt Encoder
[CLS]
P1
name
P2
Linksys
EF4124
P5
Manufacturer
P6
[M]
P9
K
[SEP]
skip the prompt encoder, being embedded via the pre-trained embedding layer of PLM
retrieved knowledge
Trainable Tuple Representation for Missing Value Imputation

Figure 1: Framework Overview

requires a lot of money and computational resources. Therefore, we do not adopt the LLMs in this paper and find from the preliminary experimental results that our method has comparable performance in the missing value imputation task.

Contributions. We introduce a novel PLM-based missing value imputation method named PRPMI. To the best of our knowledge, we are the first to learn trainable representation for tabular data. We also retrieve additional tables or documents for the PLMs to accomplish the missing value imputation task. We conduct preliminary experiments to demonstrate the effectiveness of our method.

2 PROPOSED METHOD

2.1 The Overview of PRPMI

The framework of PRPMI is given in Figure 1. PRPMI treats the missing value imputation as a multiclass classification task, selecting a value from the attribute domain to impute the missing cell. Specifically, PRPMI first serializes the attribute names and values in a tuple into a sequence of tokens to conform to the input format of the PLM. Then, we concatenate a series of trainable continuous embeddings from an additional *Prompt Encoder* with the embeddings of attribute names and values given by the embedding layer of the PLM, and feed them as the input to the transformer layer of the PLM. Note that the continuous embeddings will be updated by back-propagation to optimize the task objective, and the details will be given in Section 2.2. We also retrieve extra knowledge for the imputation task from the data lake or the Internet through either coarse-grained or fine-grained retrieval methods. The details can be found in Section 2.3. These embeddings of knowledge are also appended to the tuple representation in a trainable manner and co-input to the transformer layer. Here, we also adopt some strategies to select the extra data that are most beneficial for missing value imputation.

In the end, we take the embedding of token [CLS] from the last layer of the PLM as the summarization of the tuple to be imputed. Then, a multiclass classifier, consisting of a multilayer perceptron (MLP) and a softmax layer, is appended and outputs the probability of each value being correctly imputation result (PRPMI can deal with more than one attribute at the same time, *i.e.,* append multiple multiclass classifiers behind the PLM). Therefore, two steps are conducted for PRPMI:

Fine-tuning. We use the complete tuple as the training data. We mask an attribute, retrieve extra knowledge for the masked attribute value, and train the model to predict the original value. In the back-propagation phase, we update the parameters of the PLM and the prompt encoder.

Imputing. After fine-tuning, PRPMI will select the value with the maximum probability as the imputation results.

2.2 Trainable Tuple Representation

Extensive studies [12] have demonstrated the critical importance of prompt design in PLMs. Meanwhile, we find that simply concatenating attribute names and values cannot fully realize the power of PLMs. Consequently, we propose to learn trainable representation for each tuple.

Token Embedding. Formally, let $\boldsymbol{T}$ denote a relational table with n tuples $\{t_1, \ldots, t_n\}$ and m attributes $\{A_1, \ldots, A_m\}$. A_i is the name of the ith attribute, and we use $t[A_i]$ to denote the value of A_i in tuple $t \in \boldsymbol{T}$ when there is no need to specify which tuple in particular. Assuming $t[A_k]$ is missing, tuple t can be initially represented as:

$$\boxed{t = \{[CLS], \boldsymbol{P}_1, A_1, \boldsymbol{P}_2, t[A_1], \ldots, \boldsymbol{P}_{2k-1}, A_k, \boldsymbol{P}_{2k}, [\mathbf{M}], \ldots, \boldsymbol{P}_{2m}, t[A_m], [SEP]\}}$$

Here, $\boldsymbol{P}_i$ is the ith trainable continuous prompt and initializes as the positional embedding, *i.e.,* indicating the position of the following attribute name or value in the sequence. The mask token $[\mathbf{M}]$ denotes the missing attribute value. [CLS] and [SEP] are the start and the end of a tuple.

Subsequently, the attribute name A_i and its value $t[A_i]$ are embedded into $\boldsymbol{e}(A_i)$ and $\boldsymbol{e}(t[A_i])$ through the pre-trained embedding layer of the PLM. Meanwhile, we leverage an additional prompt encoder to map trainable embedding $\boldsymbol{P}_i$ into $\boldsymbol{h}_i$, which has the same dimension as the embedding of a token from the PLM. Here, a lightweight neural network, such as long short-term memory (LSTM) or MLP, can be adopted as the prompt encoder. Thus, the tuple can be finally represented as:

$$\boxed{\begin{aligned}&\boldsymbol{E}(t) = \{\boldsymbol{e}([CLS]), \boldsymbol{h}_1, \boldsymbol{e}(A_1), \boldsymbol{h}_2, \boldsymbol{e}(t[A_1]), \ldots, \boldsymbol{h}_{2k-1}, \boldsymbol{e}(A_k), \boldsymbol{h}_{2k},\\ &\boldsymbol{e}([\mathbf{M}]), \ldots, \boldsymbol{h}_{2m}, \boldsymbol{e}(t[A_m]), \boldsymbol{e}([SEP])\}\end{aligned}}$$

If additional attributes or documents are obtained via knowledge retrieval introduced in Section 2.3, denoted by $\boldsymbol{K}$, we concatenate them with the tuple representation also in a trainable way:

$$\boxed{\begin{aligned}&\boldsymbol{E}(t) = \{\boldsymbol{e}([CLS]), \boldsymbol{h}_1, \boldsymbol{e}(A_1), \boldsymbol{h}_2, \boldsymbol{e}(t[A_1]), \ldots, \boldsymbol{h}_{2k-1}, \boldsymbol{e}(A_k), \boldsymbol{h}_{2k},\\ &\boldsymbol{e}([\mathbf{M}]), \ldots, \boldsymbol{h}_{2m}, \boldsymbol{e}(t[A_m]), \boldsymbol{h}_{2m+1}, \boldsymbol{E}(\boldsymbol{K}), \boldsymbol{e}([SEP])\}\end{aligned}}$$

Here, $\boldsymbol{E}(\boldsymbol{K})$ denotes the representation of extra knowledge, which is also encoded by the PLM. Note that if $\boldsymbol{K}$ is tabular data, we also adopt the trainable representation for it.

Training of Prompt Encoder. The prompt encoder is a separate model, independent of the PLM, but they work together to minimize the difference between the multiclass classifier's prediction of the missing value and its truth value. Thus, in the back-propagation phase, we optimize the parameters of the PLM and the prompt encoder successively to minimize the cross-entropy loss,

$$\mathcal{L} = -\sum_{i=1}^{n_c} log(p(y_i)) \tag{1}$$

where n_c is the number of complete tuples in $\boldsymbol{T}$ and $p(y_i)$ is the probability of belonging to the class y_i (the true label of tuple t_i) given by the multiclass classifier.

2.3 Knowledge Retrieval for Data Imputation

It has been proven by the NLP field that leveraging additional knowledge can significantly enhance the capabilities of PLMs which is also known as retrieval-augmented generation (RAG) [10]. Here, we design two ways to perform "RAG" for tabular data. The first one is to search for a joinable table and expand attributes for all tuples simultaneously, which can be referred to for predicting the missing value (*Coarse-grained Retrieval Augmentation*). The second is to individually search for the reference documents for each tuple containing the missing value (*Fine-grained Retrieval Augmentation*).

Coarse-grained Retrieval Augmentation. We take the whole table as a query and discover joinable tables from the data lake employing existing techniques [7]. Joining them to the query table provides more attributes that can be referred to for missing value imputation. Due to the possibility of the retrieved joinable tables containing numerous irrelevant attributes, we can adopt inter-attribute correlations (such as Cramér's V coefficient [4]) or powerful AI tools like GPT-4 to filter them out. The coarse-grained retrieval augmentation only needs to be executed once before the imputation task.

Fine-grained Retrieval Augmentation. Here, we transform each tuple with the missing value as a query and invoke search engines such as Google API to obtain relevant documents. These searches can come up with a lot of documents, which are not necessarily helpful for imputing the missing value. To address this issue, we train a PLM-based scoring model to evaluate the relevance of each document. Specifically, this model takes the query and each retrieved document as input and outputs their relevance score (which is set between 1 and 5, with higher indicating more relevance). To obtain training data for this model, we leverage GPT-4 to generate the labels, which have been proven effective in providing such feedback [13]. After obtaining the relevance score for each document, only the one with the highest score will be appended to the tuple as the extra knowledge. And if all documents are irrelevant (score below 2), we will not add the additional document for the imputation. It is worth noting that although we need to perform retrieval for each tuple, the overall cost is not particularly high. Calling Google API is free, and relevant documents can be returned for a tuple in about 1 second. We also find from the experiments that when there is almost 80% of the training data remaining, the scoring model can completely replace the GPT-4 (*i.e.,* documents with the highest score given by the model do contribute to the imputation task). Thus, there is no need to utilize the GPT-4 to label the training data in large quantities.

Table 1: Imputation accuracy of PRPMI and existing methods

Methods	Movie	Buy	Restaurant
HoloClean	15.0	16.2	33.1
EGG-GAE	24.4	28.8	38.7
RPT	35.6	48.9	43.3
IPM	59.1	96.5	77.2
GPT-4	85.5	**100**	62.3
PRPMI	**91.1**	**100**	**86.9**

3 PRELIMINARY RESULTS

3.1 Experimental Setting

Datasets. We employ the *Movie* dataset [1] to evaluate our method with coarse-grained retrieval augmentation. Additionally, we choose two challenging benchmark datasets, namely *Restaurants* and *Buy*, from the previous study [14] to evaluate our method with fine-grained retrieval augmentation. We remove values from several categorical attributes completely at random, and the missing rate is set to 10% by default.

Baselines. To validate the effectiveness of our method, we compare it against the state-of-the-art methods from various categories, including the statistics-based method *HoloClean* [15], the DL-based method *EGG-GAE* [18], the PLM-based methods *RPT* [17] and *IPM* [14], the powerful large language model *GPT-4* [1].

Evaluation Metrics. Following the previous work [14], we measure the accuracy of whether the imputation results are equal to the ground truth, *i.e.,* the proportion of missing values accurately imputed.

Implementation Details. We utilize BERT [6] as the PLM for our experiments to better show the improvements offered by our method. The multiclass classifier incorporates two linear layers with ReLU as the activation function. Additionally, a three-layer MLP serves as the prompt encoder. All experiments are conducted on an Intel(R) Xeon(R) Silver 4210R 2.40GHz server with an NVIDIA GeForce RTX 4090 GPU.

[1] https://alchemy.cs.washington.edu/data/

Table 2: Ablation study

Methods	Movie	Buy	Restaurant
PRPMI	91.1	100	86.9
PRPMI *w/o* Trainable	88.9	98.5	85.5
PRPMI *w/o* RAG	55.6	96.5	77.2
PRPMI *w/o* Trainable&RAG	51.1	94.2	75.8

3.2 Experimental Results

Comparison with Existing Methods. We first evaluate the performance of PRPMI and other baselines on three datasets, and the results are shown in Table 1. The best scores are highlighted in bold, and the second-best scores are underlined. The results of our method are on a gray background. From Table 1, we can see that PRPMI performs better than other baselines across all datasets. Particularly, it achieves a 100% imputation accuracy on the Buy dataset, comparable to the performance of the state-of-the-art LLM model, GPT-4. This may be largely attributed to the knowledge retrieval module in PRPMI, which allows the model to refer to additional data beyond the table at hand for missing value imputation. Moreover, the trainable tuple representation helps the model better understand the semantics of structured data. These methods such as Holoclean and EGG-GAE, which are not based on pre-trained language models, perform poorly since they fail to utilize the semantic information in the tabular data. RPT and IPM suffer from unsatisfactory performance due to improper tuple representation and a lack of additional knowledge. Stimulating the potential of large language models such as GPT-4 requires carefully designed prompts and a costly fine-tuning process.

Ablation Study. We conduct an ablation study to evaluate the effectiveness of the knowledge retrieval and trainable tuple representation modules, with the results presented in Table 2. These two modules are denoted by “RAG” and “Trainable”, respectively, and it can be seen from Table 2 that both of them enhance the performance of PRPMI. It is worth noting that removing the knowledge retrieval module decreases accuracy from 91.1% to 55.6% in the Movie dataset. This validates the effectiveness of our coarse-grained retrieval augmentation, where the appended tabular data is highly correlated with the attributes to be imputed. Additionally, the decrease in accuracy from 100% to 96.5% in the Buy dataset and from 86.9% to 77.2% in the Restaurant dataset demonstrates the efficacy of our fine-grained retrieval augmentation, which provides extra documents as the evidence for missing value imputation. Although the effect of trainable tuple representation on the model performance is not as significant as that of knowledge retrieval, it does have an impact on model performance, *e.g.,* its removal results in a 2.2% decrease in model accuracy in the Movie dataset. And the retrieved data also requires trainable prompts to be better integrated with the original tuple.

4 CONCLUSION

In this paper, we have proposed PRPMI, a missing value imputation framework based on pre-trained language models. We have utilized trainable tuple representation in PRPMI, which bridges the representation gap between tabular data and natural language text. We have designed two ways to retrieve extra knowledge from the data lake or the Internet for the missing value imputation task. Extensive experimental results have demonstrated the approach outperforms the state-of-the-art methods.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (62372034, 61902017).

REFERENCES

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] S. Van Buuren, J. P.L. Brand, C.G.M. Groothuis-Oudshoorn, and D. B.Rubin. 2006. Fully conditional specification in multivariate imputation. *Journal of Statistical Computation and Simulation* 76, 12 (2006), 1049–1064.

[3] Xu Chu, Ihab F Ilyas, and Paolo Papotti. 2013. Holistic data cleaning: Putting violations into context. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*. IEEE, 458–469.

[4] Harald Cramér. 1999. *Mathematical methods of statistics*. Vol. 26. Princeton university press.

[5] Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. TURL: table understanding through representation learning. *Proceedings of the VLDB Endowment* 14, 3 (2020), 307–319.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[7] Yuyang Dong, Kunihiro Takeoka, Chuan Xiao, and Masafumi Oyamada. 2021. Efficient joinable table discovery in data lakes: A high-dimensional similarity-based approach. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 456–467.

[8] Xuerui Hong and Shuang Hao. 2023. Imputation of Missing Values in Training Data using Variational Autoencoder. In *2023 IEEE 39th International Conference on Data Engineering Workshops (ICDEW)*. IEEE, 49–54.

[9] Julie Josse, Jérôme Pagès, and François Husson. 2011. Multiple imputation in principal component analysis. *Advances in data analysis and classification* 5 (2011), 231–246.

[10] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[11] Peng Li, Yeye He, Dror Yashar, Weiwei Cui, Song Ge, Haidong Zhang, Danielle Rifinski Fainman, Dongmei Zhang, and Surajit Chaudhuri. 2023. Table-gpt: Table-tuned gpt for diverse table tasks. *arXiv preprint arXiv:2310.09263* (2023).

[12] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. GPT understands, too. *AI Open* (2023).

[13] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634* (2023).

[14] Yinan Mei, Shaoxu Song, Chenguang Fang, Haifeng Yang, Jingyun Fang, and Jiang Long. 2021. Capturing semantics for imputation with pre-trained language models. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 61–72.

[15] Theodoros Rekatsinas, Xu Chu, Ihab F Ilyas, and Christopher Ré. 2017. HoloClean: Holistic Data Repairs with Probabilistic Inference. *Proceedings of the VLDB Endowment* 10, 11 (2017).

[16] Daniel J Stekhoven and Peter Bühlmann. 2012. MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics* 28, 1 (2012), 112–118.

[17] Nan Tang, Ju Fan, Fangyi Li, Jianhong Tu, Xiaoyong Du, Guoliang Li, Sam Madden, and Mourad Ouzzani. 2021. RPT: relational pre-trained transformer is almost all you need towards democratizing data preparation. *Proceedings of the VLDB Endowment* 14, 8 (2021), 1254–1261.

[18] Lev Telyatnikov and Simone Scardapane. 2023. EGG-GAE: scalable graph neural networks for tabular data imputation. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2661–2676.

[19] Haochen Zhang, Yuyang Dong, Chuan Xiao, and Masafumi Oyamada. 2023. Jellyfish: A Large Language Model for Data Preprocessing. *arXiv preprint arXiv:2312.01678* (2023).