ACCIO: Table Understanding Enhanced via Contrastive Learning with Aggregations

Whanhee Cho
Kahlert School of Computing
University of Utah
whanhee@cs.utah.edu

Abstract

The attention to table understanding using recent natural language models has been growing. However, most related works tend to focus on learning the structure of the table directly. Just as humans improve their understanding of sentences by comparing them, they can also enhance their understanding by comparing tables. With this idea, in this paper, we introduce **ACCIO**, t**A**ble understanding enhan**C**ed via **C**ontrastive learn**I**ng with aggregati**O**ns, a novel approach to enhancing table understanding by contrasting original tables with their pivot summaries through contrastive learning. ACCIO trains an encoder to bring these table pairs closer together. Through validation via column type annotation, ACCIO achieves competitive performance with a macro F1 score of 91.1 compared to state-of-the-art methods. This work represents the first attempt to utilize pairs of tables for table embedding, promising significant advancements in table comprehension.

1 Introduction

Leveraging the success of natural language processing techniques, the comprehension of tables has also significantly grown. Many works related to understanding tables have been helpful in applications, such as column type annotation, joining relation databases from data lakes, table to visualization, and table normalization. These efforts typically analyze tables based on the structure of the table, column relationships, or entity associations.

However, to our knowledge, no prior research has been aimed at enhancing table understanding by comparing two tables. In the realm of sentence embeddings, success has been achieved by comparing sentences using techniques like SBERT (Reimers and Gurevych, 2019) or SimCSE (Gao et al., 2021), leveraging natural language inference (NLI) datasets (Bowman et al., 2015; Nie et al., 2020). These datasets typically contain triplets consisting of a premise, an entailment, and a contradiction. The entailment sentence can be logically inferred from the premise, while the contradiction sentence directly contradicts the premise. Previous studies (Reimers and Gurevych, 2019; Gao et al., 2021) have introduced methods such as closing the entailment sentence and premise and contrasting the premise with the contradiction sentence, ultimately leading to high-quality sentence embeddings.

Year	Month	Passengers
1949	January	112
1949	February	118
1949	March	132
1949	April	129
1949	May	121

(a)

Month	1949	1950	1951
April	129	135	163
August	148	170	199
December	118	140	166
February	118	126	150
January	112	115	145

(b)

Table 1: Passenger data and pivot table. (a) is the original tabular data containing year and month passenger attributes. (b) is a pivot table with a user's query of "the average number of passengers by month and year"

Therefore, in this paper, we exploit the notion that premise and entailment sentences should be closely related by leveraging pivot tables and the original table as the tables that should be conceptually close. Pivot tables are summaries of tables using aggregation by user-defined parameters such as index, column, value, and aggregation function. For instance, Table 1a represents the original tabular data consisting of year, month, and passengers.

Users typically analyze or summarize tables by using pivot tables derived from such data. For example, when a user wants to determine "the average number of passengers by month and year," they can obtain a pivot result like Table 1b.

We present a novel approach to table understanding, called **ACCIO**, t**A**ble understanding enhan**C**ed via **C**ontrastive learn**I**ng with aggregati**O**ns. By training an encoder with original data and its corresponding pivot table using contrastive learning, we aim to bring them closer together. It's worth highlighting that this paper marks the first attempt to utilize a pair of tables for table embedding.

We validate our training method through a downstream task known as column type annotation. This task is commonly used to evaluate the quality of table embeddings by predicting the types of given columns. The performances indicate that our approach achieves comparable performance in terms of macro F1 score 91.1 for column type annotation compared to state-of-the-art baselines.

2 Related Works

Table Understanding

Table-GPT (Li et al., 2023) is a generative pretrained model (GPT) tailored for tabular data. It trains GPT through instruction tuning across 14 types of table tasks, each comprising 1000 template instances and augmented instructions and tables. Table-GPT presents only one column and candidate column type from which the model can choose. Another GPT approach relies solely on prompts to obtain column-type annotations, with prompts consisting of partial tables (Korini and Bizer, 2023).

Doduo (Jiang et al., 2023) annotates columns based on each column and relations between columns using contextualized column embeddings and learning the tables from column type annotations and relations. TURL (Deng et al., 2020) focuses on entities in tables by leveraging additional metadata such as table titles and captions from Wikipedia tables, pre-training encoder with masked entity loss. Moreover, Watchhog (Miao and Wang, 2023) employs contrastive learning within tables using a self-supervised learning approach. It augments cells, rows, and columns in various ways separately training each column. As mentioned earlier, with the exception of using GPT, all these works attempt to incorporate additional mechanisms to directly understand the structure of tabular data. In contrast, our work simplifies the training method by solely contrasting two tabular datasets.

Contrastive Learning

Contrastive learning has emerged as one of the most popular methods for data embeddings in the field of deep learning (Chen et al., 2020; Gao et al., 2021; Khosla et al., 2020). This approach aims to minimize distances between similar data samples while maximizing distances between dissimilar ones. Essentially, a model is trained to bring positive examples (similar data) closer together and push negative examples (distinct data) farther apart. This mechanism, which doesn't require manually annotated labels, proves to be efficient in learning from datasets containing positive and negative examples. In the field of table understanding, Watchog (Miao and Wang, 2023) has already employed contrastive learning, but it's generally applied within a single table augmented from a single source. In contrast, ACCIO leverages two distinct tables, recognizing that one table can be inferred from the other in various ways, offering more diverse learning scenarios compared to augmentation methods.

3 Methodoloy

3.1 Serialization

Various serialization methods exist for tabular data. Some approaches embed data column-wise and include a $[CLS]$ token for each column to facilitate understanding of the data structure using appropriate learning methods (Jiang et al., 2023; Suhara et al., 2022; Miao and Wang, 2023). Table-GPT parses the data using special characters like "|" to indicate to models that it represents a value in tabular data (Li et al., 2023). TURL (Deng et al., 2020) specifies types such as title, caption, entity, or position embeddings. However, the majority of tokenization methods primarily focus on understanding the structure of tabular data.

In our approach, we simplify table tokenization to prove the impact of contrastive learning between data and its pivot tables by linearizing the table column-wise. We serialize the table as shown in Equation 1, where h_i represents the ith column header and v_{ij} represents the value of the ith column and the jth row. The header is placed at the beginning, and the column values are linearized with spacing. The serialized table is then collected to form the serialized table representation as shown

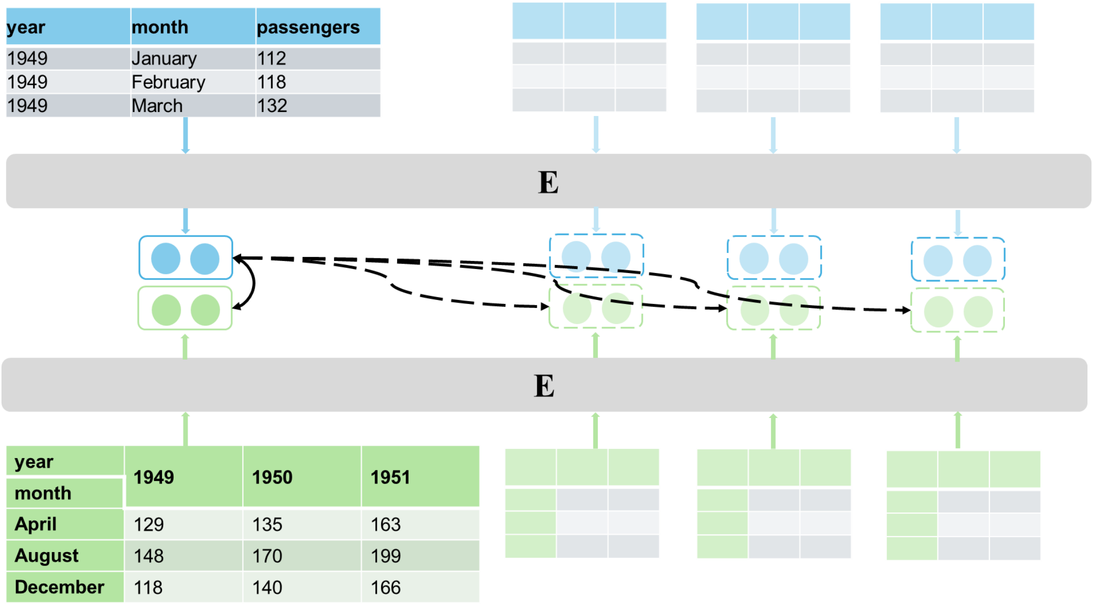

Figure 1: ACCIO training method overview. ACCIO training involves annotating original tabular data (in blue) and corresponding pivot tables (in green). The solid lines represent positive pairs, where we aim to make the embeddings closer. In contrast, the dotted lines represent in-batch negative pairs, where we aim to make the embeddings farther apart. E refers to the encoder.

in Equation 2, where N is the maximum number of columns. In this serialization, the table is appended with *[CLS]* at the beginning, and the columns are separated by *[SEP]*.

$$c_i = h_i \, v_{i1} \, v_{i2} \ldots v_{ij} \ldots \tag{1}$$

$$T = \textit{[CLS]} \, c_1 \, \textit{[SEP]} \, c_2 \, \textit{[SEP]} \ldots c_N \, \textit{[SEP]} \tag{2}$$

3.2 ACCIO

Figure 1 shows the overall contrastive learning between tabular data and pivot tables. We make embeddings closer between tabular data and its pivot tables and farther between the tabular data and the other pivot tables. We categorize pairs that need to be closer as "positive pairs" and those that need to be farther apart as "negative pairs". Specifically, in this work, we only consider negative pairs within the batch, so we refer to them as "in-batch negative pairs".

We employ h_i, the average output of last hidden state of transformer-based encoder model (Devlin et al., 2019), as done in prior works (Deng et al., 2020; Miao and Wang, 2023; Jiang et al., 2023), where the input is a table T, $i \in \{1, ..., N\}$ referred in equation 2. Each positive and in-batch negative pair of the random columns could be represented as (h_i, h_i^+), and (h_i, h_j^+), where $i \neq j$. Then, we calculate distances in pairs with a similarity function, sim, for here, sim is cosine similarity. Equation 3 shows our loss function where τ is the temperature hyperparameter.

$$l_i = -\log \frac{e^{sim(h_i, h_i^+)/\tau}}{\Sigma_{j=1} e^{sim(h_i, h_j^+)/\tau}} \tag{3}$$

4 Result

In this section, we present the experiment environment of ACCIO and the performance of the downstream task, column type annotation. We used BERT-base-uncased from huggingface [1]. NVIDIA RTX 6000 Ada 48GB.

4.1 Dataset

For training ACCIO, we utilize pairs of tabular data and pivot table datasets obtained from Auto-Suggest (Yan and He, 2020). This dataset comprises 17,189 tables along with the corresponding pivot table parameters for the Pandas function pivot table [2]. Initially, we generated pivot tables to construct our dataset consisting of tabular data and their corresponding pivot tables. Additionally, since we linearize tabular data, simply storing

[1] BERT-base-uncased: `huggingface.co/google-bert/bert-base-uncased`

[2] Pandas pivot table: `pandas.pydata.org/docs/reference/api/pandas.pivot_table.html`

the table and setting a maximum length would not account for diverse rows. Therefore, we set the number of columns to 10 and the number of rows to 10.

For the column type annotation task to validate our work, we utilize the Viznet dataset, which contains 119,360 columns preprocessed by Watchog (Miao and Wang, 2023). Watchog categorized the types of columns into 78 semantic types. Similar to prior work, we conduct experiments on multi-classification tasks by attaching a linear layer to the encoder and employing 5-fold cross-validation.

4.2 Contrastive Learning

We trained ACCIO with a learning rate of 1e-5 using the Adam optimizer, a batch size of 64, the maximum sequence length of 256, and 5 epochs, with τ set to 0.05. Our observations indicate that the training loss converges as shown in Figure 2. Therfore, we concluded that the trained model generalized enough from the contrastive learning.

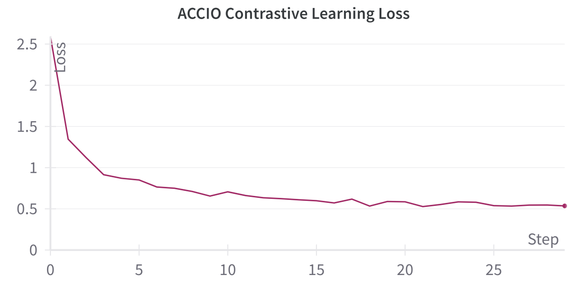

Figure 2: ACCIO contrastive learning loss.

4.3 Column Type Annotation

We appended a linear layer to the trained encoder, with a size of (768, 78) to match the 78 types and the average dimension of the last hidden state, which is 768. We trained the model with cross-entropy loss. Training the model with a maximum sequence length of 256, a batch size of 128, and 4 epochs, with a learning rate of 1e-4 using the Adam optimizer, resulted in our model achieving a macro F1 score of 91.1 and a micro F1 score of 77.3, as shown in Table 2.

ACCIO exhibits superior performance on micro F1 compared to macro F1 when contrasted with the state-of-the-art model, Watchog. This variance may stem from the unbalanced nature of the Viznet dataset, which our method appears to be sensitive to.

Method	Micro F1	Macro F1
Sherlock†	86.7	69.2
SATO†	88.4	75.6
Doduo†	94.3	84.6
Starmie†	94.0	83.6
Watchog†	**95.0**	**85.6**
ACCIO	91.1	77.3

Table 2: ACCIO performances on VizNet. †: results from Miao and Wang (2023).

5 Future Work

Quantity and Quality of Pivot Tables

Our method is limited by the number of contrastive learning training instances. With a dataset of 17,189 tables, it may struggle to generalize tables effectively. Wathhog leverages data augmentation methods within tables which doubles the training data. Additionally, due to the nature of pivot tables, some tables contain numerous null values, which poses a challenge for our approach. While our method aims to enhance table understanding by contrasting similar tables, the presence of null values can confuse the model, particularly when headers align, but most values are null. In the future, we could consider creating pivot tables from scratch and preprocessing semi-empty pivot tables to mitigate potential distractions to our work.

Structure Understanding

Since we simplified table tokenization and did not incorporate additional mechanisms for understanding the structure, we cannot guarantee that our model can fully comprehend table structures based solely on column type annotation tasks. In the future, we should include other downstream tasks, such as column relation analysis, to enhance the model's understanding of table structures.

6 Conclusion

ACCIO demonstrates competitive performance in table understanding by leveraging contrastive learning to compare tabular data and its pivot tables. This method marks the first attempt to enhance table comprehension through table comparisons. While it may not surpass state-of-the-art performance, this straightforward approach provides valuable insights for future research in the field.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 632–642. The Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. TURL: table understanding through representation learning. *Proc. VLDB Endow.*, 14(3):307–319.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910. Association for Computational Linguistics.

Zhenyu Jiang, Hanwen Jiang, and Yuke Zhu. 2023. Doduo: Learning dense visual correspondence from unsupervised semantic-aware flow. *CoRR*, abs/2309.15110.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*.

Keti Korini and Christian Bizer. 2023. Column type annotation using chatgpt. In *Joint Proceedings of Workshops at the 49th International Conference on Very Large Data Bases (VLDB 2023), Vancouver, Canada, August 28 - September 1, 2023*, volume 3462 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Peng Li, Yeye He, Dror Yashar, Weiwei Cui, Song Ge, Haidong Zhang, Danielle Rifinski Fainman, Dongmei Zhang, and Surajit Chaudhuri. 2023. Tablegpt: Table-tuned GPT for diverse table tasks. *CoRR*, abs/2310.09263.

Zhengjie Miao and Jin Wang. 2023. Watchog: A light-weight contrastive learning based framework for column annotation. *Proc. ACM Manag. Data*, 1(4):272:1–272:24.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4885–4901. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Yoshihiko Suhara, Jinfeng Li, Yuliang Li, Dan Zhang, Çagatay Demiralp, Chen Chen, and Wang-Chiew Tan. 2022. Annotating columns with pre-trained language models. In *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*, pages 1493–1503. ACM.

Cong Yan and Yeye He. 2020. Auto-suggest: Learning-to-recommend data preparation steps using data science notebooks. In *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*, pages 1539–1554. ACM.