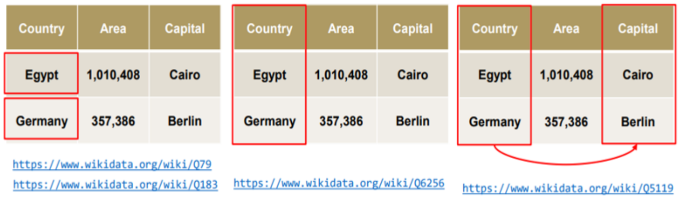
Research on Column Type Annotation Using Pre-trained Language Models

58120204 Keyu Wang

***Abstract*—This research focuses on Column Type Annotation tasks. In this research, I will consider both overall content semantics and row-column structure of the tables. And I propose a novel approach to combine two transformer models, one for capturing overall content semantics and every entity-pair relation semantics, the other for row-column structure. Finally, datasets and evaluation methods are given in this report for empiracal evaluation. The experimental results show that the proposed method can outperform existing methods based on pre-trained language models.**

I. Introduction

TABLES are crucial for data management especially for structuring large amounts of information. Nevertheless, they are hardly machine-interpretable in their raw form and thus fail to be enrolled in many automated processes. In order to better manage the tables on the web and serve various downstream applications such as Web Search [9], Question Answering [3] and Knowledge Base (KB) Construction [10], researchers begin to pay attention to how to annotate tabular data.


Country
Area
Capital
Egypt
1,010,408
Cairo
Germany
357,386
Berlin
https://www.wikidata.org/wiki/Q79
https://www.wikidata.org/wiki/Q183
https://www.wikidata.org/wiki/Q6256
https://www.wikidata.org/wiki/Q5119

(a) CEA (b) CTA (c) CPA

Fig. 1: Annotation tasks summary [1]

The annotation tasks are split into three areas, namely Cell Entity Annotation (CEA), Column Type Annotation (CTA), and Column Property Annotation (CPA). Given a data table and a target Knowledge Graph (KG), CEA links a cell to an entity within the KG (cf. Figure 1a). CTA is the task of assigning a semantic type (e.g., a class) to a column (cf. Figure 1b). Finally, CPA assigns a suitable semantic relation (predicate) from the KG to individual column pairs (cf. Figure 1c). This study mainly focuses on the Column Type Annotation task.

Recently, some studies have used the pre-training model based on transformer to CTA tasks. However, the previous SOTA methods proposed simply serialize tables and feed the serialized text directly into the pre-training model, failing to model the information of intra-column and intra-row in more detail. Therefore, more careful consideration is needed. We should not only consider how to capture the semantic relationship between each entity pair, but also focus on the structural characteristics of table rows and columns. An approach on CTA, Doduo [14], based on Transformer generally uses the method of directly serializing the table in columns, so that the table data is converted into contextual data and fed into the pre-training model, which has achieved state-of-the-art performance. This method can model the semantic relationship between any two entities in the table. However, in fact, the relationship between entities in the same column and in the same row of tables is often closer and we should pay more attention to entities which co-occur in the same row or column. TABBIE [7] encodes each cell of a table using two different Transformer models, one operating across the rows of the table and the other across columns. TURL build a visibility matrix based on the table structure and use it as an additional mask for the self-attention layer [4]. Motivated by TABBIE and TURL, two transformer models are used in the approach I designed. The first directly adopts [14], and the second applies the visibility matrix additionally, where an entity can only see other entities in the same row or column, so as to further capture the entity-pair semantics in the same row or column. Finally I combine these two transformer models for CTA tasks.

II. Related Work

Traditional CTA methods are *commercial and open source* [5] systems for data preparation and analysis, as well as prior research work on *ontology-based* [15], [16], *feature-based* approaches [12], [13].

However, these traditional methods are difficult to perform robustly on real-world web tables. In response, recent work introduced Sherlock [6], a deep learning model for semantic type detection trained on massive table corpora. Sherlock formulates semantic type detection as a multi-class classification problem where classes correspond to semantic types and extends feature-based approaches with a significantly larger set of features that includes character-level distributions, word embeddings, and paragraph vectors. However, it under-performs for types that do not have a sufficiently large number of samples in the training data and uses only the values of a column to predict its type, without considering the column's context in the table. To overcome these problem, Sato [19] combines topic modeling [2] and structured learning [8] together with single-column type prediction based on the Sherlock model. A novel pre-training/fine-tuning framework TURL are proposed for relational table understanding [4]. It consists of a structure-aware Transformer encoder to model the row-column structure as well as a new Masked Entity Recovery objective to capture the semantics and knowledge in relational Web tables during pretraining. TURL has drawbacks as it requires additional meta table information such as table headers for pretraining. [14] proposed transformer-based model Doduo which is more generic as it predicts column types and relations only relying on cell values in the table and have achieved state-of-the-art performance. However, Doduo fails to model row-column structure in the tables. This research introduce visibility matrix [19] to Doduo, aiming to better encode semantics in the tables.

I made a more detailed explanation of previous related work in Chinese in https://zhuanlan.zhihu.com/p/586867840 .

III. Method

In the model I designed, two transformer blocks are used. The architecture of first block directly adopts Doduo, and the second introduces visibility matrix on the basis of the first block to model the semantic information of rows or columns in the table. The corrsponding output vectors are concatenated, and then input into the softmax layer to do CTA multi-classification prediction.

A. Annotating Columns with Pre-trained Language Models

In this part, I adopt Doduo model and serialize the input table first. In contrast to the single-column model, Doduo is a multi-column (or table-wise) model that takes an entire table as input. Doduo serializes data entries as follows: for each table that has n columns $T = (c_i)_{i=1}^{n}$, where each column has N_m column values $c_i = (v_j^i)_{j=1}^{m}$. It lets

$$serialize(T) ::= [CLS]v_1^1 ... [CLS]v_1^n ... v_m^n [SEP]$$

Note, Doduo predicts as many labels as the number of [CLS] tokens in the input sequence. Since Doduo inserts dummy [CLS] symbols for each column, we can consider the output embeddings of the pre-trained LM for those symbols as contextualized column representations. Note that Doduo is a table-wise model, which takes the entire table as input and thus contextualized column representations take into account table context in a holistic manner. For column type prediction, Doduo attaches an additional dense layer followed by output layer with the size of $|Ctype|$. In my architecture, I modify the size of output layer to $\frac{1}{2}$ $|Ctype|$.The architacture of Doduo is shown in figure 2.

B. Applying Visibility matrix to Model

To interpret relational tables and extract the knowledge embedded in tables, it is important to model row-column structure. So I apply visibility matrix proposed in [7] to the architecture in Doduo. Visibility matrix acts as an attention mask so that each token (or entity) can only aggregate information from other structurally related tokens/entities during the self-attention calculation. M is a symmetric binary matrix with $M_{i,j} = 1$ if and only if $element_j$ is visible to $element_i$. Specifically, M is defined as follows: If $element_i$ is a token or an entity in the table and $element_j$ is a token or an entity in the same row or the same column, $M_{i,j} = 1$. *Entities and text content in the same row or the same column are visible to each other*.

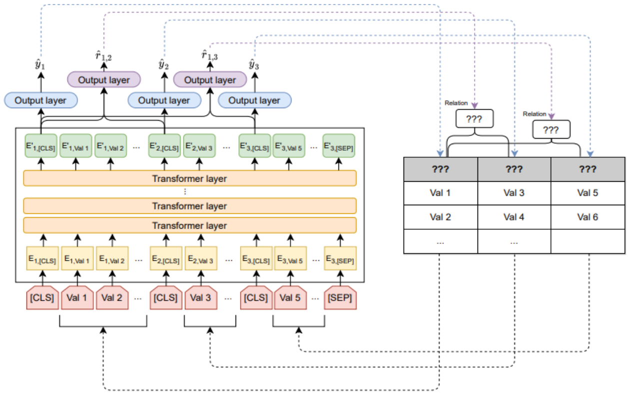


Fig. 2: Overview of Doduo's model architecture [14]

C. Combine the two transformer models

Input the table, we concatenate the output of the two models corresponding to the [CLS] output vectors in the same column, and input the concatenated vector with the size of $|Ctype|$ into the softmax layer for multi-classification task learning. The calculation formula is as follows:

$$softmax(LM_1(T) \oplus LM_2(T))$$

The $\oplus$ symbol denotes concatenation of two vectors. I then feeds the predictions and the groundtruth labels into a cross entropy loss function to update the model parameters.

IV. EVALUATION

A. Datasets

I used two benchmark datasets for evaluation. The WikiTable dataset [4] is a collection of tables collected from Wikipedia, which consists of 580,171 tables in total. The dataset provides both annotated column types and relations for training and evaluation. For column type prediction, the dataset provides 628,254 columns from 397,098 tables annotated by 255 column types. For column relations, the dataset provides 62,954 column pairs annotated with 121 relation types from 52,943 tables for training.

The VizNet dataset [19] is a collection of WebTables, which is a subset of the original VizNet corpus [17]. The dataset is for the column type prediction task. The dataset has 78,733 tables, and 119,360 columns are annotated with 78 column types.

B. Baselines

I will choose Sherlock [6], Sato [19], TURL [4]] and Doduo [14] as baselines for comparation since they are all classic models for CTA and have achieved good performance.

Sherlock [6] is a single-column prediction model that uses multiple feature sets, including character embeddings, word embeddings, paragraph embeddings, and column statistics (e.g., mean, std of numerical values.) A multi-layer "sub" neural network is applied to each column-wise feature set to calculate compact dense vectors except for the column statistics feature set, which are already continuous values. The

Method	Col type			Col rel		
	P	R	F1	P	R	F1
Sherlock	88.40	70.55	78.47	-	-	-
TURL	90.54	87.23	88.86	91.18	90.69	90.94
Doduo	92.69	92.21	92.45	91.97	91.47	91.72
Proposed Method	93.12	92.64	92.88	91.14	91.67	91.40

TABLE I: Performance on the WikiTable dataset.

Method	Col type		Col rel	
	Macro F1	Micro F1	Macro F1	Micro F1
Sherlock	69.2	86.7	64.2	87.9
TURL	75.6	88.4	73.5	92.5
Doduo	84.6	94.3	83.8	96.4
Proposed Method	83.2	87.6	81.5	93.5

TABLE II: Performance on the WikiTable dataset.

output of the subnetworks and the column statistics features are fed into the "primary" neural network that consists of two fully connected layers.

Sato [19] is a multi-column prediction model, which extends Sherlock by adding LDA features to capture table context and a CRF layer to incorporate column type dependency into prediction. Sato is the state-of-the-art column type prediction on the VizNet dataset.

TURL [4] is a recently developed pre-trained Transformer-based LM for tables. TURL further pre-trains a pre-trained LM using table data, so the model becomes more suitable for tabular data. Since TURL relies on entity-linking and meta information such as table headers and table captions, which are not available in our scenario, we used a variant of TURL pre-trained on table values for a fair comparison. Note that to perform column type/relation annotation, we fine-tuned the pre-trained TURL model on the same training sets as for Doduo and other baselines.

Doduo [14] takes as input values from multiple columns of a table after serialization and predicts column types and column relations as output. Doduo considers the table context by taking the serialized column values of all columns in the same table. This way, both intra-column (i.e., co-occurrence of tokens within the same column) and inter-column (i.e., co-occurrence of tokens in different columns) information is accounted for. Doduo appends a dummy symbol [CLS] at the beginning of each column and uses the corresponding embeddings as learned column representations for the column. The output layer on top of a column embedding (i.e., [CLS]) is used for column type prediction, whereas the output layer for the column relation prediction takes the column embeddings of each column pair.

C. Experimental Settings

Since the WikiTable dataset can have multiple labels on each column/column pair, I used Binary Cross Entropy loss to formulate as a multi-label prediction task. For the VizNet dataset, which only has a single annotation on each column, I used Cross Entropy loss to formulate as a multi-class prediction task. Models and experiments were implemented with PyTorch [11] and the Transformers library [18].

Following the previous studies [4], [19], I use micro F1 for the WikiTable dataset, and micro F1 and macro F1 for the VizNet dataset, as evaluation metrics. The micro F1 score is the weighted average of F1 values based on the sample size of each class, while the macro F1 score is the simple average of F1 values for all classes.

D. Evaluation Results

WikiTable Table 1 shows the micro F1 performance for the column type prediction and column relation prediction tasks on the WikiTable dataset. The proposed method significantly outperforms the state-of-the-art method Doduo on the task of column type prediction. In Doduo, the model uses the self-attention mechanism with the "cross-column" edges removed, which they referred to as visibility matrix . A significant difference in the model architecture between our proposed method and Doduo is whether the model uses full self-attention. From the results, the proposed method with the full self-attention performs better than TURL and Doduo, which indicates that some direct intersections between tokens in different columns and different rows are useful for the column annotation problem. However, the proposed method fails to outperform in the recall of column relation prediction. This may be because too many relationships bring heavy training burden, which leads to insufficient study of relationships.

VizNet Table 2 shows the results on the VizNet dataset. Note that the proposed method is trained only using the column prediction task for the VizNet dataset, as column relation labels are not available for the dataset. The results show that the proposed method outperforms Sherlock, Sato and Doduo, the SOTA methods for the dataset. As described in Section 2, Sato is a multi-column model that incorporates table context by using LDA features. Different from the LDA features that provide multi-dimensional vector representations for the entire table, the Transformer-based architecture enables to capture more fine-grained inter-token relationships through the self-attention mechanism. Furthermore, the table-wise design naturally helps incorporate inter-column information into the model.

V. CONCLUTION

This report presents a novel pre-training/fine-tuning framework for relational table understanding. It consists of a structure-aware Transformer encoder to model the row-column structure as well as a new visibility matrix to capture the semantics and knowledge in relational Web tables during pre-training. On compiled benchmark, I show that the method proposed in this report can be applied to a wide range of tasks with minimal fine-tuning and achieves superior performance in most scenarios. Interesting future work includes focusing on other types of knowledge such as numerical attributes in relational Web tables, in addition to entity relations and incorporating the rich information contained in an external KB into pre-training.

References

[1] Abdelmageed, N., Schindler, S.: Jentab: Matching tabular data to knowledge graphs. In: SemTab@ ISWC. pp. 40–49 (2020)

[2] Blei, D.M.: Probabilistic topic models. Communications of the ACM **55**(4), 77–84 (2012)

[3] Chakrabarti, K., Chen, Z., Shakeri, S., Cao, G.: Open domain question answering using web tables. CoRR **abs/2001.03272** (2020), https://arxiv.org/abs/2001.03272

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

[5] Ferrari, A., Russo, M.: Introducing Microsoft Power BI. Microsoft Press (2016)

[6] Hulsebos, M., Hu, K., Bakker, M., Zgraggen, E., Satyanarayan, A., Kraska, T., Demiralp, Ç., Hidalgo, C.: Sherlock: A deep learning approach to semantic data type detection. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1500–1508 (2019)

[7] Iida, H., Thai, D., Manjunatha, V., Iyyer, M.: Tabbie: Pretrained representations of tabular data. arXiv preprint arXiv:2105.02584 (2021)

[8] Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001)

[9] Nguyen, T.T., Hung, N.Q.V., Weidlich, M., Aberer, K.: Result selection and summarization for web table search. In: Gehrke, J., Lehner, W., Shim, K., Cha, S.K., Lohman, G.M. (eds.) 31st IEEE International Conference on Data Engineering, ICDE 2015, Seoul, South Korea, April 13-17, 2015. pp. 231–242. IEEE Computer Society (2015). https://doi.org/10.1109/ICDE.2015.7113287, https://doi.org/10.1109/ICDE.2015.7113287

[10] Oulabi, Y.: Augmenting cross-domain knowledge bases using web tables. Ph.D. thesis, University of Mannheim, Germany (2020), https://madoc.bib.uni-mannheim.de/55962

[11] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[12] Pham, M., Alse, S., Knoblock, C.A., Szekely, P.: Semantic labeling: a domain-independent approach. In: International Semantic Web Conference. pp. 446–462. Springer (2016)

[13] Ramnandan, S.K., Mittal, A., Knoblock, C.A., Szekely, P.: Assigning semantic labels to data sources. In: European Semantic Web Conference. pp. 403–417. Springer (2015)

[14] Suhara, Y., Li, J., Li, Y., Zhang, D., Demiralp, Ç., Chen, C., Tan, W.C.: Annotating columns with pre-trained language models. In: Proceedings of the 2022 International Conference on Management of Data. pp. 1493–1503 (2022)

[15] Syed, Z., Finin, T., Mulwad, V., Joshi, A., et al.: Exploiting a web of semantic data for interpreting tables. In: Proceedings of the Second Web Science Conference (2010)

[16] Venetis, P., Halevy, A.Y., Madhavan, J., Pasca, M., Shen, W., Wu, F., Miao, G., Wu, C.: Recovering semantics of tables on the web. pvldb 4, 9 (2011), 528–538 (2011)

[17] Wellens, Q.: Natural Language Interfaces for Data Analytics. Ph.D. thesis, Massachusetts Institute of Technology (2021)

[18] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q., Rush, A.M.: Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 38–45. Association for Computational Linguistics, Online (Oct 2020), https://www.aclweb.org/anthology/2020.emnlp-demos.6

[19] Zhang, D., Suhara, Y., Li, J., Hulsebos, M., Demiralp, Ç., Tan, W.C.: Sato: Contextual semantic type detection in tables. arXiv preprint arXiv:1911.06311 (2019)