Enhancing Table Retrieval with Dual Graph Representations

Tianyun Liu[1], Xinghua Zhang[1,2], Zhenyu Zhang[1,2], Yubin Wang[1,2], Quangang Li[1,2], Shuai Zhang[1,2], and Tingwen Liu[1,2](✉)

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{liutianyun,zhangxinghua,zhangzhenyu1996,wangyubin, liquangang,zhangshuai0612,liutingwen}@iie.ac.cn

Abstract. Table retrieval aims to rank candidate tables for answering natural language query, in which the most critical problem is how to learn informative representations for structured tables. Most previous methods roughly flatten the table and send it into a sequence encoder, ignoring the structure information of tables and the semantic interaction between table cells and contexts. In this paper, we propose a dual graph based method to perceive the semantics and structure of tables, so as to preferably support the downstream table retrieval task. Inspired by human cognition, we first decouple a table into the row view and column view, then build dual graphs from these two views with the consideration of table contexts. Afterward, intra-graph and inter-graph interactions are iteratively performed for aggregating and exchanging local row- and column-oriented features respectively, and an adaptive fusion strategy is eventually tailor-made for sophisticated table representations. In this way, the table structure and semantic information are well considered with dual-graph modeling. Consequently, the input query can match the target tables based on their full-fledged table representations and achieve the ultimate ranking results more accurately. Extensive experiments verify the superiority of our dual graphs over strong baselines on two table retrieval datasets WikiTables and WebQueryTable. Further analyses also confirm the adaptability for row-/column-oriented tables, and show the rationality and generalization of dual graphs. The source code is available at https://github.com/ty33123/DualG.

Keywords: table understanding · table retrieval · graph representation learning

1 Introduction

Table retrieval is an important task in information retrieval, which aims to rank the candidate tables extracted from the web given a natural language query. Due to valuable semi-structured information in these tabular data, table retrieval has been used in various research tasks such as knowledge graph construction [12,22], question answering [3,11], and fact verification [2,15].

D. Koutra et al. (Eds.): ECML PKDD 2023, LNAI 14172, pp. 107–123, 2023.
https://doi.org/10.1007/978-3-031-43421-1_7

Caption: Cost of Attendance

#	Dependent Student	Independent Student
Enrollment Fees	$728	$728
Food and Housing	$4392	$10980

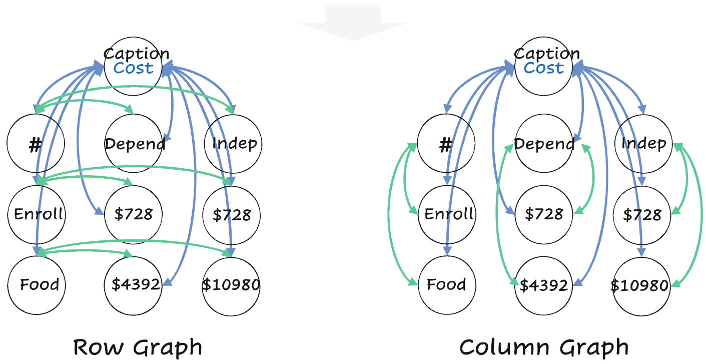

Fig. 1. Tabular graph construction for the *table* along with its context (e.g., caption), comprising *Row Graph* and *Column Graph* in our method.

Previous table retrieval methods typically treat a table as plain text by flattening the table with its contexts (e.g., page title, table caption) into a long sequence [4,13,18,19,24], in which the most direct way is arranging these table cells from left to right and from top to bottom. However, unlike natural language processing, flattened tables are still not strictly grammatical and cannot act as native text sequences, and an ideal table understanding model requires the ability of semantic comprehension and structure awareness. Taking the table in Fig. 1 as a concrete example, table cells appearing in the same column or row tend to have similar surface form and some semantic relevance, which are difficult to perceive and capture with these flat-based methods, resulting in sub-optimal performances.

Naturally, tables have a two-dimensional structure, that is, rows and columns organize the table horizontally and vertically. Therefore, there are two orthogonal ways to deliver the association among data cells and need to be accurately understood for downstream tasks like table retrieval. As shown in Fig. 1, data cells "*$4392*" and "*$10980*" share the identical fee type since they appear in the same row "*Food and Housing*", so that establishing the connections between cells and their corresponding row headers is capable of boosting the table representation. Similarly, this phenomenon also exists in columns. Moreover, table context (e.g., page title, table caption) is another information source closely related to the table topic and helps assist table understanding. From the above observations, we believe that the table should be understood from row and column, respectively, with the consideration of table contexts.

To this end, we propose a **Dual-Graph** (DualG) based table representation model for precise table retrieval. Specifically, DualG imitates human cognition and constructs row graph and column graph from horizontal and vertical perspectives, respectively. Meanwhile, table context is also specified as special nodes integrated into these graphs. Next, dual-graph representation learning is devised for the row and column graphs to obtain a full-fledged table representation, where the *intra-graph interaction* mechanism is conducted to aggregate the local row- and column-oriented information in each graph. Beside, we also propose *inter-graph interaction* to enable the exchange of heterogeneous messages between two graphs for the recombination of rows and columns, which avoids the risk of information loss in single row or column graph and facilitates the concise interaction of two views. Finally, an adaptive fusion module is introduced to dynamically fuse the unique information from row and column perspectives for achieving the holistic table representation.

In this way, the informative table representation can be involved into the final retrieval prediction procedure. Specifically, natural language query is matched with both the table representation encoded by dual-graph learning and the semantic-rich table contextual representation, and then the matching features are fed into the ultimate regression network to gain the relevance score between the query and candidate table. As the table is modeled into dual graphs, the row- and column-oriented structures can be delicately captured for enhancing table retrieval. We construct broad experiments on the WikiTables and WebQueryTable datasets to evaluate our DualG. Experimental results confirm our consistent improvements in comparison with previous state-of-the-art methods. Extensive analyses show that DualG can adaptively fuse the row- and column-aware structures, and a cross-dataset evaluation reveals the better generality.

2 Related Work

Flat table-based methods have always been a popular paradigm in the table retrieval field, which flatten the table into a text sequence for representation. Some early approaches are following unsupervised BM25 [14] or feature-based [16,18,26] table retrieval procedure. Researchers have recently explored ways to use BERT [6] for table retrieval. TaBERT [24] jointly learns representations for natural language sentences and structured tables by the content snapshot. BERT4TR [4] combines BERT and table features for joint training. Due to the length limit for BERT, only the most relevant components (rows, columns, cells) are encoded. StruBERT [19] encodes the row or column sequence of a table using horizontal and vertical self-attention. However, these flat table modeling methods cannot fully mine and explore the table structural information, which is essential for table understanding. In other ways, MTR [17] uses a gated multimodal unit (GMU) to learn a joint representation of the query and the different table modalities, and the final table-query relevance is estimated based on the query and unimodal representation.

The development of graph representation learning has brought state-of-the-art research results in many fields. There are also some researchers try to adapt these techniques into table retrieval. MGNETS [5] builds the graph based on the whole corpus, where each unique data cell, context term, and table in the corpus

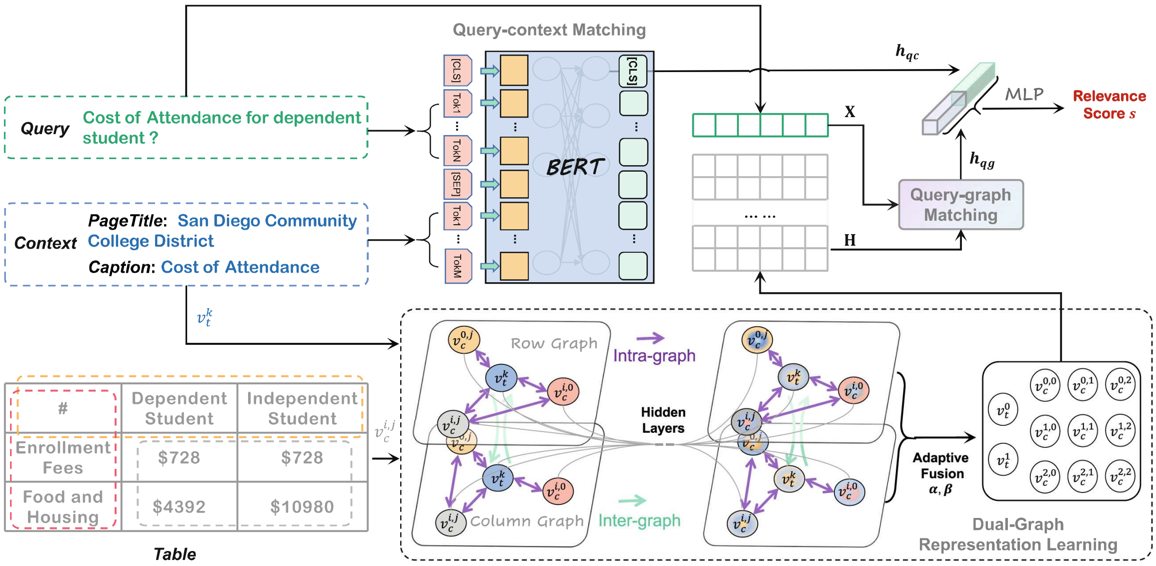

Fig. 2. Overview of DualG, we obtain the relevance score (s) by query-context matching and query-graph matching, where **H** is achieved by Dual-Graph representation learning and **X** is the initial representation of query. The horizontal purple arrows represent information propagation within the graph (Sect. 3.3), and the vertical green arrows represent an exchange of heterogeneous information between the row graph and column graph (Sect. 3.3).

serve as nodes, and edges represent the membership and co-occurrence relationship. GTR [23] converts the table into a single tabular graph with data cell, virtual row, and column nodes to capture the layout structures. Although the above graph-based algorithms can obtain the intrinsic table structure, they still face the challenges of information fuzziness and entanglement. Because rows and columns may represent different semantic relations in different tables, modeling the global natural layout into a single graph neglects the local row and column properties. In addition, table contexts also have strong indicative effects on table gist and offer vital guidance for table understanding, which prior methods often ignore for modeling the table. Different from the above attempts, we follow the design specification of tables and the expression mechanism of information, propose first to *decouple a table into row view and column view and build dual graph* with its instructive contexts, and then couple two graphs for the holistic table representation.

3 Methodology

In this section, we first introduce the studied problem (Sect. 3.1) and then describe our proposed framework DualG for table retrieval.

Overview. Figure 2 illustrates the architecture of DualG. In our framework, table retrieval is composed of tabular graph construction (Sect. 3.2), tabular graph representation learning (Sect. 3.3), and relevance prediction (Sect. 3.4). In particular, we consider the table cells and its contexts to build the tabular graphs from row and column views in tabular graph construction. As for tabular graph learning, a systematic dual-graph learning method is proposed to capture the holistic table representation, which contains intra-graph interaction, inter-graph interaction, and adaptive fusion. In prediction module, the relevance score can be obtained by matching the natural language query with the table representation and table contextual features.

3.1 Task Definition

In table retrieval task, given a query $q \in Q$, the candidate table set $\mathcal{T} = \{T_1, ..., T_p\}$ is sorted by the relevance to q in descending order, where T_i $(i = 1, .., p)$ is the i-th table in the table set. The table T_i consists of m rows and n columns, and several surrounding text (e.g., table caption). The core problem of table retrieval is the calculation of the correlation between q and T_i.

3.2 Graph Construction

In this subsection, to effectively depict the structured information for the table, we describe how to construct dual graphs: the row graph $\mathcal{G}_r = \{(\mathcal{V}, \mathcal{E}_r)\}$ and the column graph $\mathcal{G}_c = \{(\mathcal{V}, \mathcal{E}_c)\}$. From human perception, $\mathcal{G}_r$ and $\mathcal{G}_c$ should have the same nodes but different edges. So, they are both composed of cell nodes $\mathcal{V}_c$ and context nodes $\mathcal{V}_t$, i.e., $\mathcal{V} = \{\mathcal{V}_c, \mathcal{V}_t\}$. Each data cell in i-th row and j-th column of table is regarded as a node $v_c^{i,j} \in \mathcal{V}_c$ in the graph. For the merged data cell, we restore the original layout and fill in the same data as the merged cell. As table context (e.g., web page title, table caption) has rich semantic knowledge which is highly relevant to the table topic and helps assist table understanding, we view each context information as a graph node $v_t^k \in \mathcal{V}_t$.

Considering that the first node of a row/column is usually the table header which describes the main content of row/column, It is intuitive to establish the connection between each table cell node and header node in a differentiated way. Herein, when building distinct tabular graphs with row and column structures, we construct two kinds of undirected edges (cell-context edge and cell-cell edge). The specific edges of *row* and *column* graphs are as follows

Row Graph. As for the edges $\mathcal{E}_r$ in row graph, we connect each data cell node $v_c^{i,j}$ $(i \in \{0, ..., m\}, j \in \{0, ..., n\})$ in the table with the row header node $v_c^{i,0}$ and the context node v_t^k, that is $(v_c^{i,j}, v_c^{i,0}) \in \mathcal{E}_r$ and $(v_c^{i,j}, v_t^k) \in \mathcal{E}_r$, which enables the graph to aggregate information by rows.

Column Graph. When connecting $\mathcal{E}_c$, we build the relation between each cell node $v_c^{i,j}$ $(i \in \{0, ..., m\}, j \in \{0, ..., n\})$ and the column header node $v_c^{0,j}$, the context node v_t^k, respectively. That is $(v_c^{i,j}, v_c^{0,j}) \in \mathcal{E}_c$ and $(v_c^{i,j}, v_t^k) \in \mathcal{E}_c$. In this graph, the message can be passed by columns.

Figure 1 shows an example of a constructed tabular graph from the table which caption is *Cost of Attendance*. In a tabular graph, the caption as context node connects to all cell nodes, and the first cell of a row/column connects to each cell node of this row/column. To this end, correlation among different rows can be captured by the intermediate context (e.g., table caption) node in our DualG, which is modelled in a column graph. The correlation among different columns can also be captured in row graph by the similar way.

3.3 Dual-Graph Representation Learning

As shown in Fig. 2, representation learning for dual graphs (row and column graph) contains three parts: *intra-graph interaction*, *inter-graph interaction* and *adaptive fusion module*, where node information is aggregated within a graph and exchanged between dual graphs. In this way, row- and column-based structures can be finely captured respectively and then fused adaptively for holistic dual-graph representations.

Intra-Graph Interaction. Graph convolutional network [9] is a typical graph neural network model that contains a stack of convolutional layers and is employed for different representation learning tasks. To better learn the table representation, we perform it on row and column graphs to learn latent node embeddings, respectively, and learn the representation of each node by aggregating information from its neighbours within a graph. Therefore, intra-graph interactions can be used to understand different cell contents through tabular structures:

$$\mathbf{H}^{(l+1)} = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}\right) \tag{1}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix $\mathbf{A}$ of the graph $\mathcal{G}_r$ ($\mathcal{G}_c$) with added self-connections, $\mathbf{I}$ is the identity matrix, $\tilde{\mathbf{D}}$ is the diagonal node degree matrix with $\tilde{\mathbf{D}}(i,i) = \sum_j \tilde{\mathbf{A}}(i,j)$, $\mathbf{W}^{(l)}$ is the l-th layer trainable weight matrix, σ is a non-linear activation function, $\mathbf{H}^{(0)}$ is initialized with pretrained word vectors according to each node content. Thus, l-th layer hidden representations in $\mathcal{G}_r$ and $\mathcal{G}_c$ can be notated as $\mathbf{H}_r^{(l)}$ and $\mathbf{H}_c^{(l)}$ respectively.

Inter-Graph Interaction. To enable heterogeneous information (row- and column-based structures) from different graphs to be gradually fused into an accordant one, we introduce inter-graph interaction to exchange information between dual graphs, which first apply a linear transformation and layer normalization [20] to node representation following l-th layer in GCN:

$$\mathbf{H}_r^{(l)} = \mathrm{LayerNorm}(\mathbf{H}_r^{(l)}\mathbf{W}_r^{(l)} + b_r^{(l)}), \tag{2}$$

$$\mathbf{H}_c^{(l)} = \mathrm{LayerNorm}(\mathbf{H}_c^{(l)}\mathbf{W}_c^{(l)} + b_c^{(l)}), \tag{3}$$

$$\mathbf{H}^{(l)} = \mathrm{MLP}([\mathbf{H}_{\mathrm{r}}^{(1)} \| \mathbf{H}_{\mathrm{c}}^{(1)}]), \tag{4}$$

where $\mathbf{W}_r^{(l)}$, $b_r^{(l)}$, $\mathbf{W}_c^{(l)}$ and $b_c^{(l)}$ are trainable parameters. After that, the normalized node representations $\mathbf{H}_r^{(l)}$ and $\mathbf{H}_c^{(l)}$ learned from the row graph $\mathcal{G}_r$ and column graph $\mathcal{G}_c$ are concatenated, which then goes through an MLP layer to combine the heterogeneous information and obtain the final l-th layer hidden features $\mathbf{H}^{(l)}$ in dual-graph representation learning, i.e., $\mathbf{H}_r^{(l)} = \mathbf{H}_c^{(l)} = \mathbf{H}^{(l)}$.

Adaptive Fusion. A perfect fusion strategy should eliminate redundant data after combining heterogeneous information. So, we introduce a particular fusion module in the last layer of DualG for full-fledged table representation. Concretely, the fusion strategy to adaptive fuse row- and column-based unique features and the dynamic weights are computed as follows:

$$\boldsymbol{\alpha} = \mathrm{sigmoid}(\mathbf{H}_r^{(L)}\mathbf{W}_r^{(L)} + b_r^{(L)}), \tag{5}$$

$$\boldsymbol{\beta} = \mathrm{sigmoid}(\mathbf{H}_c^{(L)}\mathbf{W}_c^{(L)} + b_c^{(L)}). \tag{6}$$

where $\mathbf{H}_r^{(L)}$ and $\mathbf{H}_c^{(L)}$ are row-aware and column-aware node representations after the L-th (last) layer of GCN, which do not conduct the inter-graph interaction but direct access to fusion module. Due to the complexity and diversity of tables, we design these vector $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^{|\mathcal{V}|\times 1}$ represent the each cell row/column weight, where $\mathbf{H}_r^{(L)}$ and $\mathbf{H}_c^{(L)}$ are calculated with Eq. 1 in row graph $\mathcal{G}_r$ and column graph $\mathcal{G}_c$ respectively at the last layer of GCN. $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ represent the adaptive fusion weights of $\mathcal{G}_r$ and $\mathcal{G}_c$ respectively. To enable $\boldsymbol{\alpha} + \boldsymbol{\beta} = \mathbf{1}$, we normalize $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is:

$$\boldsymbol{\alpha} = \boldsymbol{\alpha}/(\boldsymbol{\alpha} + \boldsymbol{\beta}), \tag{7}$$

$$\boldsymbol{\beta} = \mathbf{1} - \boldsymbol{\alpha}. \tag{8}$$

Then, we can get the final full-fledged table representation $\mathbf{H}^{(L)}$ based on the calculated importance $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ for dual graphs:

$$\mathbf{H}^{(L)} = \boldsymbol{\alpha}\mathbf{H}_r^{(L)} + \boldsymbol{\beta}\mathbf{H}_c^{(L)} \tag{9}$$

Inspired by Attention [20], we use the multi-head dual-graph representation method, which allows the model jointly learn relevant information from different representation subspaces. For table representation $\mathbf{H}_h^{(L)}$ in *head* h, we can obtain it with Eq. 1–9. And then we stack the initialized table representation and the multi-head table representations $\mathbf{H}_{mh} = [\mathbf{H}^{(0)}, \mathbf{H}_1^{(L)}, ..., \mathbf{H}_h^{(L)}, .., \mathbf{H}_N^{(L)}]$, and perform mean pooling to get the final table representation $\mathbf{H}$ whose dimension is consistent with $\mathbf{H}_h^{(L)}$.

3.4 Prediction

Based on the table representation, we conduct the prediction process in three steps: *query graph matching* to understand the table structure and obtain query-related features, *query context matching* to compute the semantic similarity between the query and table context, and the final matching optimization to combine above results for the final metric.

Query Graph Matching. Given the table representation obtained by Section 3.3 and query representation encoded by the pre-trained word vectors which also encode the graph node content, we first apply the linear transformation and layer normalization similar to Eq. 2 to these representations, which maps the table and query representations into the same space for subsequent matching operations. In order to implement query-relevant table features h_{qg}. We designed a query-aware attention mechanism:

$$\begin{aligned}\mathbf{h}_{qg} &= \text{Attention}(\mathbf{q}, \mathbf{K}, \mathbf{V}) \\ &= \text{softmax}(\mathbf{q}\mathbf{K}^\top/\sqrt{d_k}) \cdot \mathbf{V}\end{aligned} \tag{10}$$

where $\mathbf{q} = \mathbf{X}\mathbf{W}_q$, $\mathbf{K} = \mathbf{H}\mathbf{W}_K$, $\mathbf{V} = \mathbf{H}\mathbf{W}_V$. $\mathbf{X}$ and $\mathbf{H}$ are normalized query, table representations, respectively.

Query Context Matching. The table contextual information (e.g., page title and table caption) has valuable semantic knowledge and can provide indicative information for the table data. Therefore, we perform the query context matching to further improve the retrieval. The sentence pairs task in the pre-trained language model (PLM) is a clever interactive semantic similarity computation model. So we concatenate the query with all contexts ($\mathbf{X}_{qc}$ = *[CLS] query [SEP] contexts [SEP]*) and input them into a pre-trained language model (e.g., BERT) to extract semantic similarity features:

$$\mathbf{h}_{qc} = \text{PLM}(\mathbf{X}_{qc}) \tag{11}$$

Training and Inference

Training. After matching the query with tabular graph and table context, respectively. We concatenate the query-related table features h_{qc} and semantic similarity features h_{qg}, and feed them into a multi-layer perceptron (MLP) to calculate the relevance score:

$$s = \text{MLP}([\mathbf{h}_{qc} \| \mathbf{h}_{qg}]) \tag{12}$$

Following BERT4TR [4], we treat the problem as a regression task and approximate point-wise ranking with a mean square error (MSE) loss as:

$$\mathcal{L}_{\text{MSE}} = \textstyle\sum_i (y_i - s_i)^2 \tag{13}$$

where y_i is the gold relevance score of table i and s_i is the predicted score by Eq. 12.

Otherwise, when the query has only one relevant table, we minimize the cross-entropy loss as training objective, following [23]:

$$\mathcal{L}_{\text{CE}} = -\textstyle\sum_i [y_i \log(s_i) + (1-y_i) \log(1-s_i)] \tag{14}$$

Inference. Given a query q, each candidate table T_i is constructed as dual tabular graphs for the table representation which is then passed into query-graph matching process, and then combined with query-context matching for an estimation of relevance score s_i.

4 Experiments

4.1 Datasets

To evaluate the effectiveness of the proposed approach, we conducted extensive experiments on two datasets for table retrieval: WikiTables [26] and WebQueryTable [18]. Query-table pairs of both datasets were collected from different sources.

WikiTables is one of the most commonly used datasets for table retrieval task. It contains 60 queries from two source query subsets [1,21], and the tables extracted from Wikipedia[1] (dump date: 2015 October). The dataset has a total of 3120 retrieval pairs (query-table pairs), and each retrieval pair is labelled with 0 (irrelevant), 1 (relevant), and 2 (highly relevant).

WebQueryTable uses the search logs of commercial search engines to obtain a list of query-table pair marks the most relevant query-table pair with 1 and the rest of the candidate list with 0, producing 21,113 query-table pairs, each query has only one relevant table. We follow previous work [16,23,26] to separate the dataset as training, validation, and testing with a 7:1:2 split.

4.2 Baselines

We compare our method with the following table retrieval baselines and group them into four types:

(1) *Feature-based methods*: **BM25** [14] calculates the score for document and each word in the query. **LTR** [26] uses 18 different discrete features for regression training using a random forest. **T2VW** [25] employ neural language modeling approaches to embed tabular data into vector spaces. **STR** [26] extends LTR by introducing additional 16 features. **Feature + NeuralNet** [18] combines word-level, phrase-level, sentence-level features and neural network architectures to measure the relevance score between query and table. **TabIESim** [16] enhances the retrieval by a combination of intrinsic and extrinsic table similarity based on BM25 and cluster hypothesis [10].

(2) *BERT-based methods*: **BERT4TR** [4] first use the BERT to encode the table, and then the retrieval performance is further improved by combining features [26]. **TaBERT** [24] jointly learns contextual representations for utterance and the structured schema of tables, implicitly capturing the mapping between them based on BERT. **StruBERT** [19] uses horizontal and vertical self-attentions to the encoded column- and row-based table sequences.

[1] https://en.wikipedia.org/.

Table 1. Main results on WikiTables. **Bold** indicates the best result, <u>underline</u> is the second best, and "-" indicates the result not reported in the original paper. The significant test p-value < 0.05 when comparing with GTR.

Method Type	Method	N@5	N@10	N@15	N@20	MAP
Feature-based	BM25 [14]	0.3196	0.3377	0.3732	0.4045	0.4260
	LTR [26]	0.5527	0.5456	0.5738	0.6031	0.4112
	T2VW [25]	0.5974	0.6096	0.6312	0.6505	0.4675
	STR [26]	0.5951	0.6293	0.6590	0.6825	0.5141
	TabIESim [16]	0.6498	0.6479	–	0.6935	0.5124
BERT-based	BERT4TR [4]	0.6361	0.6519	0.6558	0.6564	0.6311
	TaBERT [24]	0.5926	0.6108	0.6451	0.6668	0.6326
	StruBERT [19]	0.6393	–	–	–	0.6378
Multi-modal	MTR [17]	<u>0.6631</u>	<u>0.6813</u>	–	<u>0.7370</u>	0.6058
Graph-based	MGNETS [5]	0.6373	0.6490	–	–	0.6339
	GTR [23]	0.6554	0.6747	<u>0.6978</u>	0.7211	<u>0.6665</u>
	DualG (Ours)	**0.6707**	**0.6925**	**0.7259**	**0.7541**	**0.7083**

Table 2. Retrieval performance on WebQueryTable.

Method	Precision@1	MAP
BM25 [14]	0.4712	0.5823
Feature+NN [18]	0.5415	0.6718
BERT4TR [4]	–	0.7104
GTR [23]	0.6257	0.7369
DualG (Ours)	**0.6363**	**0.7466**

(3) *Multi-modal methods*: **MTR** [17] views web tables as multimodal objects, which uses gated multimodal units (GMUs) to learn joint-representation.

(4) *Graph-based methods*: **MGNETS** [5] constructs two table-term graphs by mining co-occurrence relation, and conduct GCN on both graph. **GTR** [23] transforms the table into a single tabular graph with data cell, row and column as nodes to capture multi-granular content and the layout structures.

4.3 Implementation Details

In our experiments, we use BERT-base [6] to extract semantic similarity features h_{qc}. Similar to BERT4TR [4] and GTR [23], the words inside table are initialized by FastText [8] with dimension 300 in query-graph matching. The number of GCN layers L and heads N are set to 2 and 4, respectively. During training, we set the learning rates of BERT and GCN to 1e-5 and 1e-4, respectively. The random seed number is 0. The batch size and the number of maximum epochs

Table 3. Ablation studies on WikiTables. N is the abbreviation of NDCG.

	N@5	N@10	N@15	N@20	MAP
DualG (Ours)	**0.6707**	**0.6925**	**0.7259**	**0.7541**	**0.7083**
w/o Dual graphs	0.6161	0.6610	0.6944	0.7213	0.6849
w/o Row graph	0.6403	0.6663	0.7100	0.7406	0.6989
w/o Column graph	0.6100	0.6441	0.6951	0.7202	0.6757
w/o Context node	0.6024	0.6505	0.6925	0.7173	0.6850
w/o Inter-graph interaction	0.6117	0.6538	0.6906	0.7225	0.6823
w/o Adaptive fusion	0.6365	0.6830	0.7121	0.7436	0.7034
w Single graph	0.6163	0.6663	0.6888	0.7178	0.6792
replace Adaptive fusion	0.6349	0.6749	0.7050	0.7373	0.6938
replace Inter-graph interaction	0.6247	0.6466	0.6799	0.7175	0.6891

are 16 and 5, respectively, on both datasets. Because the WikiTables dataset does not provide a data split, we follow the previous work [4,5,16,17,23,26] and conduct a 5-fold cross-validation on this dataset for evaluation. We use the originally released split for WebqueryTable evaluation. Our framework is implemented with PyTorch and DGL for graph learning.

Evaluation Metrics. Due to distinct labeling strategies between two datasets, we use different groups of metrics to evaluate the performance of different methods for WikiTables and WebQueryTable, following previous work [4,23,26]. On WikiTables, we report Normalized Discounted Cumulative Gain (NDCG@n, n={5,10,15,20}) and Mean Average Precision (MAP). Because WebQueryTable only has one positive sample for each query, we report MAP and Precision@1 (P@1). Specifically, MAP and NDCG metrics are calculated using the TREC[2].

4.4 Results

From Tables 1 and 2, it can be seen that our proposed DualG method outperforms all other baselines on both datasets. The outstanding results confirm the necessity of capturing the table structural information, and the effectiveness of decoupling the whole table layout into dual graphs (row and column graph), which contributes to the fine-grained row- and column-aware structural features for full-fledged table understanding.

On WikiTables dataset, our method outperforms prior graph-based SOTA GTR [23] by 1.53%, 2.81%, 3.30% on NDCG@{5,15,20} and 4.18% on MAP. The reason is that our approach has the ability to adaptively capture precise row- and column-aware structures based on dual-graph learning instead of the overall layout graph without row/column distinction, and then the thorough

[2] https://github.com/usnistgov/trec_eval.

and accurate table representation can be achieved for higher performance. Compared to the multimodal approach, our proposed method outperforms MTR [17] by 1.12% on NDCG@10 and 10.25% on MAP, which shows the limitation of capturing table structure based on multimodal information. It is worth noting that the significant improvement in the MAP metric indicates that DualG has higher precise discrimination on the candidate tables.

In contrast with flat table based methods, our approach outperforms the latest StruBERT [19] by 3.14% on NDCG@5 and 7.05% on MAP. The reason is that flattening tables into sequences loses the structural information. While these approaches adapt a variety of attention mechanisms to achieve an understanding of the table, it fundamentally limits the upper bound of the model. As for the WebqueryTable dataset, our DualG outperforms the existing SOTA by 1.06% on Precision@1 and 0.97% on MAP, which comes to a consistent conclusion.

4.5 Analyses

Ablation Study. To evaluate the impact of each module in our method, we perform the following ablation studies:

(1) After removing dual-graph representation learning (*w/o* Dual graphs), i.e., the query only matches with table contexts, our DualG reduces by 5.46% on NDCG@5 and 2.34% on MAP. This ablation result shows the significant efficacy of our dual graphs, which is conducive to precise table representation for an accurate matching process. Furthermore, eliminating one of the dual graphs (*w/o* Row/Column graph) leads to a 3.04% and 6.07% decline on NDCG@5, which confirms the necessity of row and column structures.

(2) When removing the context node in our dual graphs (*w/o* Context node), the performance is reduced by 6.83% on NDCG@5 and 2.33% on MAP, which demonstrates that the table context information (e.g., web page title, table caption) can significantly profit the table understanding.

(3) Meanwhile, getting rid of inter-graph interaction when performing dual-graph representation learning results in a 5.90% drop in NDCG@5. The reason is that heterogeneous information is exchanged and propagated between row and column graphs, which contributes to a thorough understanding of table layout.

(4) We replace the adaptive fusion module with an average one (setting alpha and beta to 0.5). As shown in Table 3 (*w/o* Adaptive fusion), the NDCG@5 index declined by 3.43%, which confirms the efficacy of assigning dynamic weights to dual graphs to fuse the unique information.

(5) We merge the edges in row and column graphs into a single graph (*w* Single graph), leading to remarkable declines, which demonstrates the effectiveness of capturing fine-grained row/column information by building dual graphs.

(6) Furthermore, we replace the inter-graph interaction module or the adaptive fusion module with another of these modules and observed that the retrieval performance decreased substantially. The experimental results show that these modules play different roles in DualG as described in the motivation.

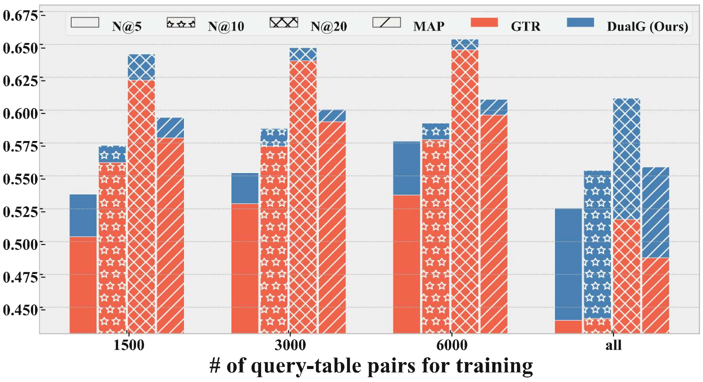

Fig. 3. Results of cross-dataset evaluation, training on WebQueryTable and testing on WikiTables. Blue blocks indicate the absolute increase of our method. (Color figure online)

Year	Award Ceremony	Category	Nominee	Result
1954	Tony Award	Best Performance by a Leading Actor in a Musical	Alfred Drake	Won
1954	Tony Award	Best Conductor and Musical Director	Louis Adrian	Won

(a) Relational Table

Area	55,673km²
State capital	Shimla
Language	Hindi
District	12
Population	Census
Literacy	77%

(b) Entity Table

	Dependent Student	Independent Student
Enrollment Fees	$728	$728
Food and Housing	$4,392	$10,980

(c) Matrix Table

Fig. 4. Example for three types of Table.

Cross-Dataset Evaluation. To study the generalization ability, we train our DualG and prior GTR on WebQueryTable, and evaluate on WikiTables. Concretely, we use *1500*, *3000*, *6000* and *all* query-table pairs from WebQueryTable, which are approximately *half*, *equal*, *double* and *quadruple* of WikiTables dataset respectively. As shown in Fig. 3, blue blocks indicate the absolute increase of our method in comparison with GTR [23]. We can observe that our DualG achieves consistent increases on all query-table pairs of different proportions. The DualG shows the prominent superiority on NDCG@5, indicating that our model can rank the related tables ahead even across datasets. In addition, training on the full WebQueryTable tends to produce the worst performance on WikiTables, the reason may be that a large number of training instances exacerbate the data distribution gap between WebQueryTable and WikiTables dataset. Still, our DualG shows the more significant advantages over GTR (e.g., 11.27% increase on NDCG@10) when training on *all* query-table pairs of WebQueryTable. Overall, our superiority comes from the dual graphs based on row view and column view, which contribute to the accurate table understanding and improve the generalization of our DualG.

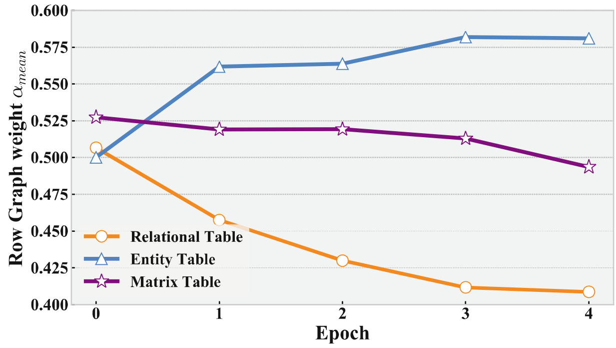

Fig. 5. Dynamic changes of row graph weight α_{mean} during training for various tables.

Case Study. As shown in Fig. 4, table types can be summarized into *Relational Table*, *Entity Table* and *Matrix Table* [7,23]. To study the adaptability of our DualG for various table types, we give 3 cases from WikiTables dataset corresponding to three table types in Fig. 4, and then record the change trend of fusion weight $\alpha_{mean} = \text{mean}([\text{mean}(\boldsymbol{\alpha}_1), ..., \text{mean}(\boldsymbol{\alpha}_N)])$ for the row graph during training process in Fig. 5. We can observe that: α_{mean} for the relational table tends to decrease with the training epoch, while the entity table shows an uptrend. As the relational table is column information-intensive and DualG properly focuses on a column-aware layout instead of a row-aware one, while the entity table has rich row information and exhibits the opposite trend. The α_{mean} weight for the matrix table converges to about 0.5, which shows DualG can capture both row- and column-aware structures correctly. Overall, our DualG has the powerful potential of adaptability for diversified table layouts.

5 Conclusion

This paper proposes a dual-graph based method DualG for enhancing table retrieval, which captures the local row- and column-aware layouts with table contexts (e.g., caption), and then acquires the thorough table representation with tailor-made interaction and fusion mechanism. Experimental results show that the approach is effective and feasible by using it in table retrieval, and the cross-dataset evaluation shows that it has a good generalization ability. Our method further enriches the table retrieval community from graph perspective, and we would like to apply it into related downstream tasks, e.g., table-based question answering and fact verification in the future.

Acknowledgment. This work is supported by the National Key Research and Development Program of China (grant No.2021YFB3100600), the Strategic Priority Research Program of Chinese Academy of Sciences (grant No.XDC02040400) and the Youth Innovation Promotion Association of CAS (Grant No. 2021153).

Ethics Statement. I understand that using technology can have ethical implications, especially in collection, processing, and privacy of form retrieval data. I acknowledge and recognize the importance of complying with ethical standards and the hazards of potential risks.

In the data collection and processing, my training data comes from two publicly available tabular search datasets. Although we do not collect or store any sensitive information, we should strictly restrict the retrieval text of users and ensure that it does not contain any dangerous information.

In addition, when the model used in police or military related applications, we should pay special attention to its use in these areas, which must conducted in a more responsible manner. To prevent models from providing inaccurate search results for police or military personnel, users are responsible for ensuring that they comply with ethical principles and laws and regulations when using model outputs, and for screening search results.

In summary, I strive to ensure that the model outputs search results in an ethical and responsible manner, and I urge my users to do the same. I will continue to adhere to ethical standards and stay abreast of emerging ethical issues in the fields of machine learning and data mining.

References

1. Cafarella, M.J., Halevy, A., Khoussainova, N.: Data integration for the relational web. Proc. VLDB Endowment **2**(1), 1090–1101 (2009)
2. Chen, W., et al.: TabFact: a large-scale dataset for table-based fact verification. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020. OpenReview.net (2020). https://openreview.net/forum?id=rkeJRhNYDH
3. Chen, W., Zha, H., Chen, Z., Xiong, W., Wang, H., Wang, W.Y.: HybridQA: a dataset of multi-hop question answering over tabular and textual data. In: Findings of the Association for Computational Linguistics: EMNLP 2020. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.findings-emnlp.91, https://aclanthology.org/2020.findings-emnlp.91
4. Chen, Z., Trabelsi, M., Heflin, J., Xu, Y., Davison, B.D.: Table search using a deep contextualized language model. In: Huang, J., et al. (eds.) Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, 25–30 July 2020. ACM (2020). https://doi.org/10.1145/3397271.3401044, https://doi.org/10.1145/3397271.3401044
5. Chen, Z., Trabelsi, M., Heflin, J., Yin, D., Davison, B.D.: MGNETS: multi-graph neural networks for table search. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 2945–2949. Association for Computing Machinery, New York, NY, USA (2021), https://doi.org/10.1145/3459637.3482140
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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. vol. 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1423, https://aclanthology.org/N19-1423

7. Eberius, J., Braunschweig, K., Hentsch, M., Thiele, M., Ahmadov, A., Lehner, W.: Building the dresden web table corpus: a classification approach. In: 2015 IEEE/ACM 2nd International Symposium on Big Data Computing (BDC) (2015). https://doi.org/10.1109/BDC.2015.30
8. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. vol. 2, Short Papers. Association for Computational Linguistics, Valencia, Spain (2017). https://aclanthology.org/E17-2068
9. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017, Conference Track Proceedings. OpenReview.net (2017). https://openreview.net/forum?id=SJU4ayYgl
10. Kurland, O.: The cluster hypothesis in information retrieval. In: Jones, G.J.F., Sheridan, P., Kelly, D., de Rijke, M., Sakai, T. (eds.) The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2013, Dublin, Ireland - July 28 - August 01, 2013. ACM (2013). https://doi.org/10.1145/2484028.2484192, https://doi.org/10.1145/2484028.2484192
11. Li, X., Sun, Y., Cheng, G.: TSQA: Tabular scenario based question answering. Proc. AAAI Conf. Artif. Intell. **35**(15), 13297–13305 (2021). https://ojs.aaai.org/index.php/AAAI/article/view/17570
12. MacDonald, E., Barbosa, D.: Neural relation extraction on wikipedia tables for augmenting knowledge graphs. In: d'Aquin, M., Dietze, S., Hauff, C., Curry, E., Cudré-Mauroux, P. (eds.) CIKM 2020: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, 19–23 October 2020. ACM (2020). https://doi.org/10.1145/3340531.3412164, https://doi.org/10.1145/3340531.3412164
13. Pan, F., Canim, M., Glass, M., Gliozzo, A., Fox, P.: CLTR: an end-to-end, transformer-based system for cell-level table retrieval and table question answering. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.acl-demo.24, https://aclanthology.org/2021.acl-demo.24
14. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M., Gatford, M., et al.: Okapi at trec-3. Nist Special Publication Sp 109 (1995)
15. Shi, Q., Zhang, Y., Yin, Q., Liu, T.: Logic-level evidence retrieval and graph-based verification network for table-based fact verification. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (2021). https://doi.org/10.18653/v1/2021.emnlp-main.16, https://aclanthology.org/2021.emnlp-main.16
16. Shraga, R., Roitman, H., Feigenblat, G., Canim, M.: Ad hoc table retrieval using intrinsic and extrinsic similarities. In: WWW 2020: The Web Conference 2020, Taipei, Taiwan, April 20–24, 2020. ACM/IW3C2 (2020). https://doi.org/10.1145/3366423.3379995, https://doi.org/10.1145/3366423.3379995

17. Shraga, R., Roitman, H., Feigenblat, G., Canim, M.: Web table retrieval using multimodal deep learning. In: Huang, J., (eds.), Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, 25–30 July 2020. ACM (2020). https://doi.org/10.1145/3397271.3401120, https://doi.org/10.1145/3397271.3401120
18. Sun, Y., Yan, Z., Tang, D., Duan, N., Qin, B.: Content-based table retrieval for web queries. Neurocomputing **349**, 183–189 (2019). https://doi.org/10.1016/j.neucom.2018.10.033, https://www.sciencedirect.com/science/article/pii/S0925231218312219
19. Trabelsi, M., Chen, Z., Zhang, S., Davison, B.D., Heflin, J.: StruBERT: structure-aware BERT for table search and matching. In: Proceedings of the ACM Web Conference 2022. WWW 2022, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3485447.3511972, https://doi.org/10.1145/3485447.3511972
20. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., (ed.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4–9, 2017, Long Beach, CA, USA (2017)
21. Venetis, P., et al.: Recovering semantics of tables on the web. Proc. VLDB Endowment **4**(9), 528–538 (2011)
22. Wang, D., Shiralkar, P., Lockard, C., Huang, B., Dong, X.L., Jiang, M.: TCN: table convolutional network for web table interpretation. In: Proceedings of the Web Conference 2021. WWW 2021, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3442381.3450090, https://doi.org/10.1145/3442381.3450090
23. Wang, F., Sun, K., Chen, M., Pujara, J., Szekely, P.: Retrieving complex tables with multi-granular graph representation learning. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3404835.3462909
24. Yin, P., Neubig, G., Yih, W.T., Riedel, S.: TaBERT: pretraining for joint understanding of textual and tabular data. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.acl-main.745, https://aclanthology.org/2020.acl-main.745
25. Zhang, L., Zhang, S., Balog, K.: Table2vec: neural word and entity embeddings for table population and retrieval. In: Piwowarski, B., Chevalier, M., Gaussier, É., Maarek, Y., Nie, J., Scholer, F. (eds.) Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, 21–25 July 2019. ACM (2019). https://doi.org/10.1145/3331184.3331333, https://doi.org/10.1145/3331184.3331333
26. Zhang, S., Balog, K.: Ad hoc table retrieval using semantic similarity. In: Champin, P., Gandon, F.L., Lalmas, M., Ipeirotis, P.G. (eds.) Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, 23–27 April 2018. ACM (2018). https://doi.org/10.1145/3178876.3186067, https://doi.org/10.1145/3178876.3186067