11711:

Yang Yang* **Yansheng Cao ***
{yangy4, yanshenc}@andrew.cmu.edu

1 Dataset and Task

For this project, our team chose to explore the task of Ad-hoc table retrieval, as we think table is one of the powerful, versatile and interesting format for working with data through our experiences as data scientist and quantitative researchers. There are a massive number of tables in real applications such as on Wikipedia, Web and as excel spreadsheets; and to deal with a large collection of tables, a powerful best-match based search model is important especially working with tabular data in the wild. The Ad-hoc table retrieval task can be described as given a query, keyword or short phrase, return a ranked list of tables from a table corpus that are relevant to the query. During our research, we discovered that the Ad-hoc table retrieval is currently an important and yet understudied field compared to other table-to-text tasks such as semantic parsing or question answering, especially given that table retrieval is a core building block in the aforementioned other table-to-text tasks. Hence, the goal of our project is to better understanding the connection between text and semi-structured table, and better leverage the semi-structured nature of tables to improve the performance of Table retrieval.

We performed extensive research on existing literature and we found only three major data sources: WikiTables (Cafarella et al., 2009), (Venetis et al., 2011), (Zhang and Balog, 2018a), WebQueryTable (Yan et al., 2017)and NQ_TABLE (Herzig et al., 2021). Given this scarcity, we choose to work on improving Table based Retrieval using WikiTables (Cafarella et al., 2009), (Venetis et al., 2011), (Zhang and Balog, 2018a) and WebQueryTable (Yan et al., 2017), which are the benchmarks that most works in the research community experiment on, and covers are from real-world search log with diverse intents. Both dataset that contains query/questions with answers embedded in the tables.

We aim to study the influence of each aspect of table: row, column, cell, row header, column header, and the interactions between these aspect for table retrieval, and develop novel ways to encode information within Table data using graphical representation and study the importance of graphical representation for table understanding for the Ad-hoc table retrieval.

2 Literature Review

Table based retrieval system have received more attention in the information retrieval in the recent years following the advances of dense retrieval for textual data. The formal definition of our task is to develop a system that given a keyword query q and the set of tables $T_{corpus} = T_1, ..., T_N$, the system automatically returns a ranked list of tables according to how likely the query q would be satisfied by the information in each table. Prior approaches before 2020, for the Ad-hoc retrieval task, follows two approaches. 1) Information retrieval approach, consider table as the same as linear textual data and apply language model using Neural Network. 2) Feature engineering approach by either considering only lexical matching between the contents of tables and queries, or shallow modelling of semantic matching using Word2Vec based embedding representation. More recently, as neural architectures evolved, there is an emerging trend to move beyond lexical information and better leverages the semantic representations from entities and categories, as well as leveraging the semi-structured nature of tables to better understanding the connection between text and semi-structured table using pre-trained language models such as BERT.

In the following sections, we will discuss the limited set of open-sourced datasets and the vari-

*Everyone Contributed Equally – Alphabetical order

ous approaches that researchers have proposed to improve the performance of table-retrieval systems.

2.1 Dataset

Compared to traditional information retrieval task, Table Retrieval is still an emerging field as the interest for Text-to-SQL and Table QA grew, hence there are much fewer relevant open domain datasets and some are derived using subset of other QA dataset. We believe that as natural language processing for semi-structured data such as Web table data becomes a more prominent field of research, there will be an increase in investment towards new dataset and new pretraining retrieval, QA as joint models. Hence in this section, we aim to provide a brief overview of the currently available public datasets and analyze their strengths, weaknesses, and key features.

WikiTables The WikiTables corpus (Bhagavatula et al., 2015) is one of the most important dataset for QA on tabular data, the dataset comprises of 1.6 million tables extracted from Wikipedia. (Zhang and Balog, 2018a) developed a table retrieval dataset using WikiTable corpus by sampling 60 test queries from two independent sources based on Web users (Cafarella et al., 2009), and query logs from Google Squared (Venetis et al., 2011). For this query subset, 3,120 candidate tables were extracted from Wikipedia and each candidate query-tables pair is labeled by annotators with one of three relevance scores with : 0 (irrelevant), 1 (relevant), and 2 (highly relevant). Each table is associated with contextual information including a caption, Wikipedia's page title and section title. Out of the 3120 query-table pairs, 377 are labeled as highly relevant, 474 as relevant, and 2269 as non-relevant, and 1800 of these tables contains some nested structure which is suitable for our goal.

WebQueryTable Chen et al(Yan et al., 2017) use search logs from a commercial search engine to get a list of queries that could be potentially answered by web tables. The dataset contains 21,113 web queries and 273,816 web tables from Wikipedia. Each query table pair is obtained from the top ranked Web page of a commercial search engine with manual evaluation, and fir each table, the table caption was also given as additional contextual information.

NQ-TABLE The NQ-TABLE dataset (Herzig et al., 2021) is originally created from the Google Natural Questions dataset(Kwiatkowski et al., 2019) (Web based open domain QA dataset), the dataset contains 11K examples where the answer resides in some table from the full QA dataset. This dataset is particularly interesting as it is part of the on-going Natural Question Open domain QA competition, and we are keenly interested in how much augmentation does semantic representation of tables can improve the result of QA model if we have enough time and resources. Each sample of NQ dataset consists of a question, an answer, and an associated Wikipedia page context triplet. The task is to output a long answer (which is typically a paragraph) and a short answer (which is an entity) given the input. Out of the 320k questions, the answer to 11k questions resides in the embedded table data in the Wikipedia article, hence the data that we will work with can be formulated as: (Q, T, A) triplets of question, table and answer. Given the collection of tables from all of NQ-TABLE, our task is to find the most relevant tables based on the unstructured natural language query.

2.2 Systems and Architectures

Ad-hoc Table retrieval is not an entirely new field, previously it has been studied as relational ranking by the database community as well as the information retrieval community. Before large pretraining model emerged, researchers focused on improving Table IR system using a rich set of lexical and semantic features. (Table 1 in the Appendix section provides an overview of different features used). As neural network based architectures evolved, recent efforts in advancing the retrieval task for table have focused on representational learning through vector semantic encoding using techniques from Word2Vec such as Table2Vec (Zhang et al., 2019), to pretrained-LM such as TAPAS (Herzig et al., 2020), TaBERT (Yin et al., 2020), that offers more precise and generalizable results for the downstream retrieval task using semantic similarities. Furthermore, given that tables may have complex structures, some of the most recent approaches leverages graph representation learning using graph transformers to perform graph to text matching for table retrieval in order to further leverage the structured natures of tables compared to plain text.

WebTables (Cafarella et al., 2008) was one of the first large-scale attempt to extract and leverage the relational information embedded in HTML tables on the Web. The authors discussed a collec-

tion of lexical table-dervied features and parsing techniques to develop a relation search engine is built on the set of recovered relations and features. (Zhang and Balog, 2018b) leverages these features and trained a linear classifier as a baseline for ad-hoc table retrieval task on their selected dataset built from two independent sources of 60 queries with the WikiTables dataset.

STR (Zhang and Balog, 2018b) summarized the set of lexical and semantic features that have been proposed and used in the Ad-hoc table retrieval tasks and introduced various fusion techniques to learn semantic (vectorized) representation of information in tables built from entities, category, words and entity graphs.

Table2Vec (Mikolov et al., 2013a) and (Mikolov et al., 2013b) proposed the Word2Vec model which drastically changed the direction of representation Learning on Text, and following the advance of dense distribution embedding of text using neural networks, researchers in the IR community adapted the Skipgram Word2Vec model (Mikolov et al., 2013b) as backbone to model table headings, entities, and all of the table data using heading embeddings, entity embeddings and word embeddings. The Table2Vec embeddings achieved significant improvement for retrieval result compared previous state of the art baselines using lexical and semantic features such as entity similarity based on similarity of relations of entities, jaccard similarity between outgoing links of entities etc.

BERT and BERT4TR Since language model pre-training has successfully improved performance on many natural language processing tasks, (Devlin et al., 2018) built the Bidirectional Encoder Representations from Transformers (BERT) model. And with the advancement of large-scale pretrain models such as BERT, research on Ad-hoc table retrieval shifted from manually crafting features/embedding based features built from Word2Vec towards learning representations for natural language sentences and semi-structured tables using weakly/semi supervised learning.

BERT for Table Retrieval (Chen et al., 2020) applied the pre-trained language model BERT (Devlin et al., 2018)to encode flattened representation of tables for the ad-hoc table retrieval task. The authors proposed an system architecture that leverages BERT to encode table information and perform relevance matching. The pipeline is separated into three components, the first is an embedding based selection process to select the most relevant rows from tables with respect to queries, followed by using BERT to encode the query and contextual information of a table and selected tabular content is flattened as a sequence and encoded by BERT. The vectorized representation generated by BERT is then concatenated with manual a query-table features using an multi-layer perceptron to compute the final relevance score. Following BERT4TR, multiple researches have tackled table retrieval by developing better pre-training models built upon BERT that better leverages all the information in table data.

TAPAS TAPAS is a model proposed by google research (Herzig et al., 2020) which is a weakly supervised QA model on tables without generating logical forms. TAPAS is extending BERT's architecture with additional embeddings that could capture the table's structure and two classification layers to select cells and predict a corresponding aggregation operator.TAPAS is trained from end to end, encoding tables crawled from Wikipedia as input.

The authors also showed that TAPAS could effectively pretrain over large scale data of text-table pairs. Furthermore, TAPAS could achieve better results comparing with previous state-of-the-art parsers by fine-tuning on semantic parsing datasets.

TaBERT TABERT is a pretrained language model that jointly learns representations for natural language sentences and (semi-)structured tables (Yin et al., 2020).

TABERT linearizes the structure of tables to be compatible with a Transformer-based BERT model. To cope with large tables, we propose content snapshots, a method to encode a subset of table content most relevant to the input utterance. This strategy is further combined with a vertical attention mechanism to share information among cell representations in different rows (§ 3.1). To capture the association between tabular data and related NL text, TABERT is pretrained on a parallel corpus of 26 million tables and English paragraphs

Dense Retrieval for Table QA With the advance of pre-trained language models, open-domain QA over a corpus of textual passages(free-text input data) have seen a lot of progress, specifically us-

ing a two-stage framework consisting of a retriever model that first selects a small subset of candidate passages relevant to the query, followed by a reader model that selects the "best" answer given the selected relevant contexts. Specifically on the retriever side, dense retrieval approaches targeted for retrieving passages (Lee et al., 2019) (Guu et al., 2020) and (Karpukhin et al., 2020) have shown strong performance on various QA dataset from Wikipedia to Reddit Forum data. Given the connection between text and table data, more recent researches in Ad-hoc table retrieval focuses on modify the retriever to better handle tabular contexts. (Herzig et al., 2021) designed a retriever model that contextually represent text and a table jointly using the TAPAS architecture which includes table specific embeddings that capture the table's structure, such as row and column ids. In DTR, we use TAPAS to represent both the query q and the table T. For efficient retrieval during inference we use two different TAPAS model for the query and the table, and learn a similarity metric.

Most pre-trained language modelling based on extending BERT to tabular data leverages row-wise and col-wise attention to model the interaction between cells in the table similar to a fully connected graph fashion based on attention mechanism, however, this not only introduces high computational cost for example fine-tuning experiment for dual TAPAS encoder for the query and the table as the retriever model took 6 hours using 32 TPUs; but also does not necessarily deliver the best performance compared to a more sparse graph representation of a table, especially in the context of the ad-hoc retrieval task.

Graph Base Table Retrieval(GTR) Compared to linear language models on plain textual data (attention over row or column), table data often have different and complex structures in terms of layouts, for example, the table structure may be relational, matrix-format, nested or contains pure entites etc. Therefore, (Wang et al., 2021) proposed a Graph-based table retrieval system that constructs an multi-granular tabular graph for any arbitrary table using cell node adjacency, and uses a pre-trained graph transformer based model architecture proposed by (Koncel-Kedziorski et al., 2019), similiar to graph attention network, to characterize both cell content and layout structures with cell nodes representations initialized using FastText(Mikolov et al., 2018), followed by a BERT as the query-context matching module and FastText as the encoder for the query for the query-graph matching module. Finally, the query-table matching representation is concatenated with the query context representation and fed to a multi-layer perceptron to calculate the relevance score between query representation and table representation. The GTR system showed significant improvement compared to linear language models based such as TABERT for complex tables and diverse queries, furthermore GTR perform better than BERT-based methods even w/o pre-training on WebQueryTable (Yan et al., 2017) and achieves better cross dataset generalization.

3 Baselines

3.1 GTR

Motivation We select GTR (Wang et al., 2021) as our baseline for mainly two reasons: First, the graph representation of tables provides more flexibility in table layouts. Second, the model uses a tablular Graph Transformer to support multi-granular feature extraction with tabular graphs as input and hence it could better capture the semantic information of a table in various subunits. Furthermore, the novelty graph based approach shows better performance in training efficiency, both general and specific queries, and complex tables than the previous BERT-based methods. The overall architecture of GTR is shown in Figure 2.

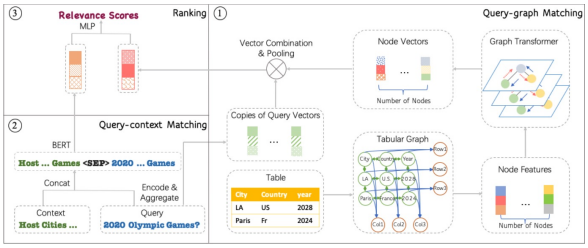

Figure 1: Overall Architecture of GTR system taken from (Wang et al., 2021)

Setting and Dataset We use the code-base released by the authors of GTR (Wang et al., 2021). To be consistent with the paper's result, we followed the same train and test procedures as described in the paper. On the WikiTable dataset, we used 5-fold cross validation to train and test the model given that the dataset contains only 60 queries, we trained the model for 10 epoches on every fold. And we tried limited grid search to tune the learning rate of Adam optimizer for BERT and for the Graph Transformer given re-

source constraint on WikiTable. On the WebQueryTable dataset, we used split the dataset into train and test set, and we used the same hyper-parameter included in the original repo of GTR and trained the model for 2 epoches compared to 5 epoches reported by the original authors due to our computational limits. For the BERT model, we used BERT_base_uncased, which is consistent with the original authors and is less hardware demanding. During our experiment, we found that training the GTR system on WikiTable with our setting 2 hours on a single Tesla V-100 GPU, while on WebQueryTable, training the model for 2 epoches takes 30 hours.

Evaluation Metric We uses Normalized Discounted Cumulative Gain(NDCG@k) at four cut of 5, 10, 15, 20 and Mean Average Precision(MAP) as the evaluation metric. Both metrics are the standard metrics used for evaluating search engine/retrieval models, and they are calculated using the TREC evaluation tool. The Mean Average Precision score measures whether all of the relevant items tend to get ranked highly; and the NDCG@k score measures whether very relevant results are ranked earlier and using normalization makes the results comparable across different queries.

Result During inference, we output the ranked list of tables for each query according the relevance score produced by the GTR system, and we use the TREC_eval software to compute the final metrics.

The result that we obtained for our best trained model on WikiTable dataset is listed in the table 1. The results is around 1 percent worse than the ranking results recorded by the original authors. We believe this is largely due to the randomness from performing cross validation given the small data size of the WikiTable dataset. We will be attempt to achieve more accurate results once AWS approves our request and run more experiments.

	WikiTable
NDCG@5	0.6432
NDCG@10	0.6621
NDCG@15	0.6868
NDCG@20	0.7102
MAP	0.6525

Table 1: The performance (accuracy) of baselines on WikiTable

The result that we obtained for the GTR model trained for two epoches on WebQueryTable dataset is listed in the table2. The MAP score is very close to the result in the original paper, and we believe that once we have the AWS limit raise approved: we can run the model training for five epoches and achieve more accurate results, similar to the results in the original paper.

	WebQueryTable
MAP	0.7289

Table 2: The performance (accuracy) of baselines on WebQueryTable

Error Analysis

- There are certain cases that the model over-evaluate the text similarity between query and table title but under-evaluate the actual content of the table. For example, for query "fast car", the model gives the table 'Fast Cars and Superstars: The Gillette Young Guns Celebrity Race' highest rank. Although the title explicitly contains the text "fast car", the table is all about racer's performance rather than the cars. Similarly, for query "infections treatment", the tables it ranks the highest contains information about the diseases and bacteria without information about treatment. We believe these errors are raised when the context is not strong enough while the query is a high-level description so that the table content has very few direct connections to the query.

- We notice that the model tends to perform better in larger sized tables and it tends to give lower correlation scores for smaller tables. First, among the top ranked tables for each query, the correctly predicted rate is higher for larger tables as shown in the figure. Additionally, for small sized table (bottom 25%), only about 64.28% of the high correlated table query pairs are predicted correctly, while for large size tables (top 25%), the number rise to 76.71 %.

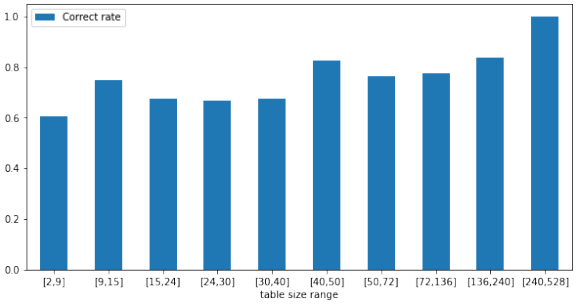

Figure 2: Correct rate of different table sizes

4 Approach and Future Plan

There are three potential improvement that we hope to implement and experiment on the WikiTable and WebQueryTable dataset.

We aim examine the robustness of UnifiedQA and BERTjoint to position bias by separating the Nat- ural Questions dataset into samples with answers in the first sentence of a context vs samples with answers in other parts of a context. Prior works(Ko et al., 2021) have shown that for LMs such as BERT pretrained on SQuaD with answers that can be found in the first sentence of the context, the correlation coefficient defined by cosine similarity is much higher in the synthetic dataset. While this outcome is not desirable, such exploitation may help boost performance on the given dataset be- cause real-application scenarios do not necessarily follow this distribution. We also hope to incorporate multi-hop reason- ing in the question and context generation phase. Instead of working with the exact question, we want to explore the possibility of adding multi-hop reasoning to iterate between finding/reading the context and generating a refined search/question query.

References

Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. 2015. Tabel: Entity linking in web tables. In *The Semantic Web - ISWC 2015*, pages 425–441, Cham. Springer International Publishing.

Michael J Cafarella, Alon Halevy, and Nodira Khoussainova. 2009. Data integration for the relational web. *Proceedings of the VLDB Endowment*, 2(1):1090–1101.

Michael J. Cafarella, Alon Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. 2008. Webtables: Exploring the power of tables on the web. *Proc. VLDB Endow.*, 1(1):538–549.

Zhiyu Chen, Mohamed Trabelsi, Jeff Heflin, Yinan Xu, and Brian D. Davison. 2020. Table search using a deep contextualized language model. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

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*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training.

Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Martin Eisenschlos. 2021. Open domain question answering over tables via dense retrieval.

Jonathan Herzig, Pawel Krzysztof Nowak, Francesco Piccinno Thomas Müller, and Julian Eisenschlos. 2020. Tapas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering.

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. 2019. Text generation from knowledge graphs with graph transformers.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality.

Petros Venetis, Alon Halevy, Jayant Madhavan, Marius Pasca, Warren Shen, Fei Wu, Gengxin Miao, and Chung Wu. 2011. Recovering semantics of tables on the web. In *37th International Conference on Very Large Data Bases (VLDB)*. Stanford InfoLab.

Fei Wang, Kexuan Sun, Muhao Chen, Jay Pujara, and Pedro Szekely. 2021. Retrieving complex tables with multi-granular graph representation learning. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Zhao Yan, Duyu Tang, Nan Duan, Junwei Bao, Yuanhua Lv, Ming Zhou, and Zhoujun Li. 2017. Content-based table retrieval for web queries.

Pengcheng Yin, Graham Neubig, Wen tau Yih, and Sebastian Riedel. 2020. Tabert: Pretraining for joint understanding of textual and tabular data.

Li Zhang, Shuo Zhang, and Krisztian Balog. 2019. Table2vec. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Shuo Zhang and Krisztian Balog. 2018a. Ad hoc table retrieval using semantic similarity. *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*.

Shuo Zhang and Krisztian Balog. 2018b. Ad hoc table retrieval using semantic similarity. *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*.

IDF_f	cell2
QLEN	cell5
n rows	cell8
n cols	cell2
n NULLs	number of empty table cells
PMI	cell8
inLinks	cell2
outLinks	cell5
tableImportance	cell8
n hitsLC	cell2
n hitsB	cell5
yRank	cell8
qInPgTitle	cell2
qInTableTitle	cell5
Entity	Encoding for if two entities are related
Category	Encoding for if entity is assigned to a Wikipedia category
Word in Tables	cell5
Graph of entities	cell8

Table 3: Your caption.

5 Appendix

5.1 Table of features used